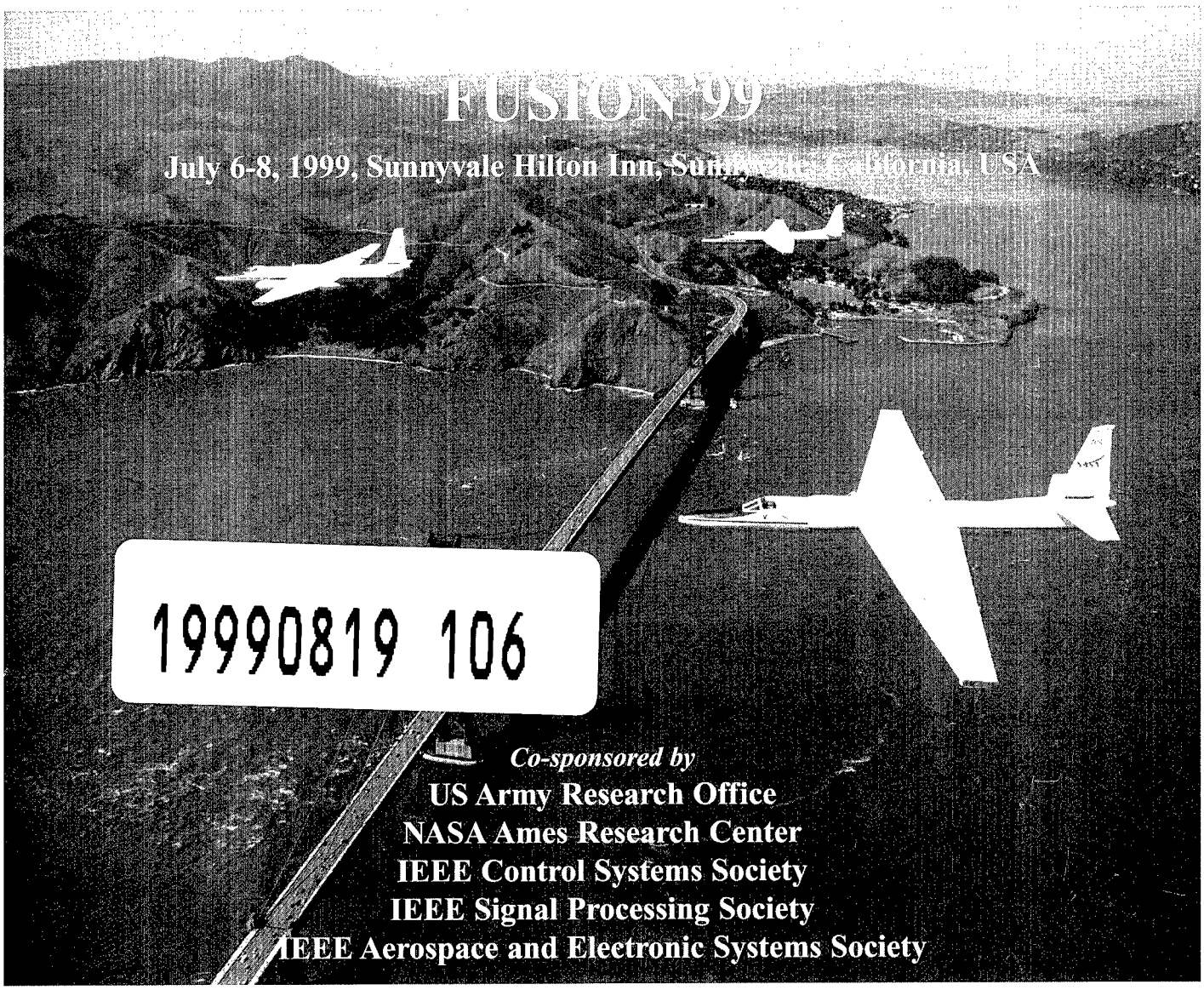
International Society of Information Fusion

PROCEEDINGS OF THE SECOND INTERNATIONAL CONFERENCE ON INFORMATION FUSION

Volume I



REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
		Final Report

4. TITLE AND SUBTITLE

FUSION'99 -- The Second International Conference on Information Fusion, Volumes I and II

5. FUNDING NUMBERS

DAAD19-99-1-0271

6. AUTHOR(S)

Dr. James Llinas, Principal Investigator

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES)

International Society of Information Fusion
Mountain Vieew, CA 94043

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

ARO 40107.1-EL-CF

11. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

NO ABSTRACT PROVIDED

14. SUBJECT TERMS

15. NUMBER IF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

International Society of Information Fusion

PROCEEDINGS OF THE SECOND INTERNATIONAL CONFERENCE ON INFORMATION FUSION

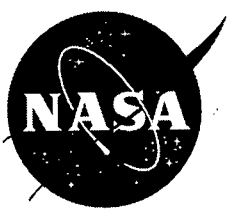
Volume I

FUSION'99

July 6-8, 1999, Sunnyvale Hilton Inn, Sunnyvale, California, USA

Co-sponsored by

US Army Research Office
NASA Ames Research Center
IEEE Control Systems Society
IEEE Signal Processing Society
IEEE Aerospace and Electronic Systems Society



This set of volumes contains technical papers presented at the Second International Conference on Information Fusion (FUSION'99). Their inclusion in this publication does not necessarily constitute endorsements by the International Society of Information Fusion, the sponsors, or by the publisher.

Distributed by

International Society of Information Fusion
415 Clyde Avenue, Suite 108
Mountain View, CA 94043 USA
Tel: 650-966-8700, Fax: 650-966-8700
Email: info@inforfusion.org, URL: www.inforfusion.org

Volume I, ISBN 0-9671429-C-3-1-1
Volume II, ISBN 0-9671429-C-3-2-X
Set, ISBN 0-9671429-C-3-3-8

Photo on the cover: The Golden Gate Bridge, San Francisco, California, viewed from the U-2 airplane. Courtesy of NASA Ames Research Center, photographer Dominic Hart.

Printed by Omnipress in the U.S.A.

Preface

Dear Members of the Information Fusion Community:

It is a pleasure to report to you that the Information Fusion community continues to mature and grow, a positive reflection on all members and especially on that subgroup of the community that persists in supporting its maturation process. Thanks are due to Dongping Daniel Zhu and X. Rong Li, Belur Dasarathy, and the members of the Transitional Board of the International Society of Information Fusion (ISIF), for the attention paid to and energy expended on the wide variety of tasks and issues involved with trying to get the ISIF established. Tasks of this sort are 'yet another thing to do' for those involved but these noble, collective efforts and their results and consequences are what give identity and substance to a community. Slowly but persistently, this community is filling in the "Infrastructure gaps" it has suffered from for some time--we hope soon to have a Society, an International Journal, and an Information Analysis Center; we already have one University Research Center, which could be expanded to a Consortium framework.

The ISIF is a particularly welcome and needed infrastructure initiative in our community, but it will only be as good as the collective efforts of its membership. Being a member of any Society results in both an opportunity and an obligation; opportunity for collegiality in its fullest sense, and obligation to contribute in its fullest sense. Being among the oldest in this community, I can tell you that I have always been proud to label myself as a member of the "fusion" community since it is a distinctive, extraordinarily interesting field of specialization, and one with great promise. We welcome and encourage you to become "official" members via the ISIF, about which we will all have considerable discussion at FUSION'99 - give us your thoughts about what ISIF should be, and give us your membership; see http://www.inforfusion.org for more information.

In recent visits I have had the opportunity to interact with and learn from Information Fusion researchers in Australia, in Spain, and in Norway, and last year I was involved in a technology planning task in Sweden. In all cases I was impressed with both the nature of the work and the talented people involved in it. I think I can say without reservation that all of the people involved in these IF efforts, as well as the cognizant organizational leaders and managers are anxious for interaction, and technology and knowledge-sharing, and for a forum to periodically share ideas. Inspired by this, I have motivated a session on "International Collaboration in IF" for this year's conference which I hope will be a standing session for future conferences, and which I hope will be one focused forum in which people can both understand what options for collaboration may exist and also to act on them. Of course the "FUSION'XX" conferences serve this purpose in the large, but offering some details on the underlying mechanics regarding programs and activities specifically tailored to international collaboration won't hurt.

Welcome to FUSION'99

Jim Llinas
President, International Society of Information Fusion

International Society of Information Fusion

Initial Board of Directors

Yaakov Bar-Shalom, University of Connecticut, CT, USA
Mark Bedworth, Defence Evaluation & Research Agency (DERA), Malvern, UK
Chee-Yee Chong, Booz-Allen & Hamilton, San Francisco, CA, USA
Belur V. Dasarathy, Dynetics, Inc., Huntsville, Alabama, USA
Alfonso Farina, Alenia Defence Systems, Italy
X. Rong Li, University of New Orleans, LA, USA
Jim Llinas, State University of New York, Buffalo, NY, USA
Daniel McMichael, CSSIP, Australia
Jane O'Brien, Defence Evaluation & Research Agency (DERA), Malvern, UK
Pramod K. Varshney, Syracuse University, NY, USA
Dongping Daniel Zhu, Zaptron Systems, Inc., Mountain View, CA USA

Initial Society Officers

President: Jim Llinas, State University of New York, Buffalo, NY, USA
Secretary: Dongping Daniel Zhu, Zaptron Systems, Inc. Mountain View, CA, USA
Treasure: Chee-Yee Chong, Booz-Allen & Hamilton, San Francisco, CA, USA

Society Address

415 Clyde Avenue, Suite 108
Mountain View, CA 94043, USA
Tel: 650-966-8700, Fax: 650-966-8780
Email: info@inforfusion.org URL: http://www.inforfusion.org

Foreword

Across Las Vegas desert land, Heat waves shimmering from the sand.
A fusion caravan comes into view, Destination – Timbuktu.

If Shakespeare is correct that "What's past is prologue," then FUSION'98 should be an good introduction that brings us together again at FUSION'99 in the Silicon Valley, exactly one year later. Clearly, data fusion follows from idea fusion and people fusion

It gives us great pleasure to introduce this collection of papers presented at the Second International Conference on Information Fusion (FUSION'99), organized by the International Society of Information Fusion (http://www.inforfusion.org) on July 6 through July 8, 1999, at Sunnyvale Hilton Inn, California, USA. These papers reflect the state-of-the-art of sensor, data and information fusion, and cover architecture, algorithms and applications in many fields, ranging from target tracking and recognition to diagnostic information fusion and image fusion to biomedical and management information fusion.

Many factors have contributed to FUSION'99. First of all, we'd like to thank the conference sponsors, without their support this conference would not have been possible. These sponsors are NASA Ames Research Center*, US Army Research Office*, IEEE Signal Processing Society, IEEE Control Systems Society, and IEEE Aerospace and Electronic Systems Society.

We are fortunate to have many renowned people to provide vision and leadership to the conference. We are especially grateful to Dr. Yaakov Bar-Shalom of University of Connecticut who serves as Honorary Chairman, Franklin White of Navy SPAWAR as Steering Committee Chairman, Dr. Kenneth Ford of NASA as Advisory Committee Chairman, Mark Bedworth of DERA, UK and Dr. X. Rong Li of University of New Orleans as General Vice Chairmen, and Dr. Pramod Varshney of Syracuse University as Technical Program Chairman. We gratefully acknowledge Dr. Bill Sanders of Army Research Office for his continued inspiration and support.

We are very grateful to the many colleagues who are experts in the field and have greatly helped organize the conference. In particular, the General Chairman would like to thank all members on the Technical Program Committee, led by Dr. Pramod Varshney and Dr. Peter Willett, for their efforts in assembling a collection of quality papers, and Dr. Robert Levinson for his tireless effort in printing and publishing the Proceedings. We like to acknowledge other Executive Committee members: Dr. Chee-yee Chong for managing logistics and finance, Captain Erick Blasch for leading a successful sponsors program, Dr. Belur Dasarathy for publicizing the conference to a wide audience, and Dr. Fa-long Luo for local arrangements. Last but not the least, Society board directors and liaisons, session chairs, authors, and many others have offered valuable assistance. They all helped make the conference a success.

We also like to thank the following persons: Deborah Jean Gamble-Ly of Creation, Janny Wu, and Mike Lee of ComStar for administrative assistance, Maylene Duenas and her staff at NASA for technical support, Bob Hamm of OmniPress for publication, and the staff at Zaptron Systems for web site support.

With the success of FUSION'99, we can expect even greater successes at FUSION'2000 in the new millennium. In the words of Sir Winston Churchill: *"This is not the end, it is not even the beginning of the end, but it is perhaps the end of the beginning."*

Dongping Daniel Zhu, General Chairman
Zaptron Systems, Inc.
Robert Levinson, Publication Chair
University of California-Santa Cruz

* The views, opinions, and/or findings contained in this proceedings are those of the authors and should not be construed as an official US government or its agency's position, policy, or decision, unless so designated by other documentation.

Technical Program Chair's Message

I am delighted to welcome you to FUSION'99. We have assembled an excellent technical program consisting of 29 contributed and invited sessions. The conference attracted about 210 submissions from 22 countries. Each submission was reviewed by the technical program committee and only worthy papers were included in the final program. I was extremely pleased with the large number of submissions and their high quality. In addition to the technical sessions, we feature three plenary talks and a luncheon talk by R. Luo (Taiwan), K. Ford (USA), G. Shaw(USA) and F. White(USA). All of these speakers are widely known and have significant experience in their areas of expertise.

It is a pleasure to acknowledge the tireless effort of Peter Willett, the Technical Program Vice Chair. He reviewed each and every submission and was instrumental in putting the sessions together. I would like to thank the members of the Technical Program Committee for their assistance with reviewing: M. Alford (USA), B. Dasarathy (USA), D. McMichael (Australia), J. O'Brien (UK), E. Shahbazian (Canada), and P. Svensson (Sweden).

The efforts of the following persons in organizing invited sessions are greatly appreciated: C. Anken, E. Blasch, R. Blum, O. Drummond, K. Goebel, M. Kokar, M. Larkin, R. Liuzzi, J. Llinas, G. Rogova, S. Shah, A. Stoica, and D. Zhu.

This is the second year for this conference and we have made great strides in this short period. I am confident that the conference will continue to grow both in terms of size and quality. Thank you all for making this conference a success.

Pramod E. Varshney
Technical Program Chair
Professor
Syracuse University
NY, USA

Fusion'99 Executive Committee

Honorary Chairman:
Yaakov Bar-Shalom, University of Connecticut, CT, USA

Steering Committee Chairman:
Franklin E. White, SPAWAR, San Diego, CA, USA

Advisory Committee Chairman:
Kenneth M. Ford, NASA Ames Research Center, Moffet Field, CA, USA

General Chairman:
Dongping Daniel Zhu, Zaptron Systems, Inc., Mountain View, CA, USA,

General Vice Chairmen:
Mark Bedworth, Defense Evaluation & Research Agency, UK
X. Rong Li, University of New Orleans, LA, USA

Technical Program Committee Chair:
Pramod K Varshney, Syracuse University, NY, USA

Technical Program Committee Vice Chair:
Peter Willett, University of Connecticut, CT, USA

Publicity Committee Chair:
Belur V. Dasarathy, Dynetics, Inc. Huntsville, AL, USA

Publication Committee Chair:
Robert Levinson, University of California at Santa Cruz, CA, USA

Sponsors Program Chair:
Erick Blasch, Air Force Research Lab/SNAT, OH, USA

Financial Committee Chair:
Chee-Yee Chong, Booz-Allen & Hamilton, San Francisco, CA, USA

Local Arrangement Committee Chair:
Fa-long Luo, Resound Corporation, Redwood City, CA, USA

Technical Program Committee

Pramod Varshney (Chair), Syracuse University, USA
Peter Willett (Vice Chair), University of Connecticut, USA
Mark Alford, Air Force Research Laboratory at Rome, USA
Belur Dasarathy, Dynetics Corporation, USA
Daniel McMichael, University of South Australia, Australia
Jane O'Brien, Defense Evaluation Research Agency, U.K.
Elisa Shahbazian, Lockheed-Martin, Canada
Per Svensson, FOA, Sweden

Fusion'99 International Committee

J. K. Aggarwal, University of Texas-Austin, USA
C. Anken, AFRL, USA
Alain Appriou, French National Establishment for Aerospace Research, Chatillon, France
Steve Bishop, Army Night Vision Lab, Virginia, USA
Rick S. Blum, Electrical Engineering and Computer Science Dept., Lehigh University, USA
George Chapline, Lawrence Livermore Labs, CA, USA
Nianyi Chen, Chinese Academy of Sciences, Shanghai, P. R. China
Oliver Drummond, Consulting Engineer, Culver City, CA, USA
Alfonso Farina, ALENIA, Italy
Mohamad Farooq, Royal Military College of Canada, Ontario, Canada
A. Fatholahzadeh, SUPELEC: Ecole Superieure d'Electricite, France
Kai Goebel, Information Technology Laboratory, General Electric, NY, USA
Chris Harris, University of Southampton, UK
M. Hinman, AFRL, USA
R.A. Hogendoorn, National Aerospace Laboratory, The Netherlands
Masatoshi Ishikawa, University of Tokyo, Japan
Vishrut Jain, National University of Singapore, Singapore
Ivan Kadar, Northrop Grumman Corp, Advanced Systems and Technology, NY, USA
T. Kirubarajan, University of Connecticut, USA
M. Kokar, Northeastern University, MA, USA
P. Korpisaari, Tampere University of Technology, Finland
Rudolf Kruse, Universitaet Magdeburg, Germany
Michael J. Larkin, Naval Undersea Warfare Center-Newport Division, Newport, RI, USA
James Llinas, State University of NY at Buffalo, Buffalo, NY, USA
R. Lynch, Naval Undersea Warfare Center, RI, USA
Ren C. Luo, National Chung Cheng Univesity, Taiwan
Rabinder N. Madan, Office of Naval Research, Arlington, VA, USA
Peter Wide, Department of Technology and Science, Oerebro University, Oerebro, Sweden
R. Liuzzi, AFRL, USA
Shozo Mori, Raytheon TI Systems, San Jose CA, USA
S. Musick, AFRL, USA
Akira Namatame, Dept.of Computer Science, National Defense Academy, Japan
Theo van Niekerk, Port Elizabeth Technikon, South Africa
Nageswara S. V. Rao, Oak Ridge National Laboratory, TN, USA
G. Rogova, Calspan/CUBRC, NY, USA
Peter Wide, Orebro University, Sweden
Bill A. Sander, Army Research Office, NC, USA
S. Shah, Wayne State University, MI, USA
E. Shahbazian, Lockheed-Martin, Canada
Alan Steinberg, ERIM, MI, USA
Adrian Stoica, NASA Jet Propulsion Laboratory, CA, USA
Kamyshnikov Vladimir, Department of Economy, Tomsk State Architectural University, Russia
Zhongtuo Wang, Business School, Dalian University of Technology, P. R. China
Vitaly Yaschenko, Institute of Mathematical Machines and Systems, Ukraine

1 Plenary Speech I: "Multisensor Fusion and Integration Issues, Approaches and Opportunities"

Dr. Ren C. Luo, Professor and Dean College of Engineering National Chung Cheng University, Taiwan and General Chair of MFI'99 - IEEE International Conference on Multisenor Fusion and Integration for Intelligent Systems

1.1 ABSTRACT

Interest has been growing in the use of multiple sensors to increase the capability of intelligent systems. In this presentation, the issues, approaches in dealing with multisensor fusion and integration (MFI) will be discussed. The applications and potential opportunities for the implementation of MFI will also be included. The issues involved in integrating multiple sensors into the operation of a system are presented in the context of the type of information these sensors can uniquely provide. The advantages gained through the synergistic use of multisensory information can be decomposed into a combination of four fundamental aspects: the redundancy, complementarily, timeliness, and cost of the information can then defined as the degree to which each of these four aspects is present in the information provided by the sensors.

In general, sensory fusion can be accomplished at different levels: data fusion, feature fusion and decision fusion. More commonly known is data fusion level, Example of this type of fusion are fusion of multiple ultrasonic data, and fusion of images from different imaging sensors. In feature fusion level, features are extracted from the raw measurements that are then combined in a quantitative or qualitative manner. For example, feature fusion can be used to fuse information from imaging and a non-imaging sensor. Decision fusion level can be employed when the sensors available are not compatible or be applicable to many pattern recognition problems.

Typical of the applications that can benefit from the use of multiple sensors are industrial tasks like assembly, military command and control for battlefield management, mobile robot navigation, multitarget tracking, and aircraft navigation. Common among all of these applications is the requirement that the systems intelligently interact with and operate in an unstructured environment without the complete control of a human operator. Advances in hardware, software and algorithm have made it possible to employ multiple data sources for information gathering and to develop more complex multisensor fusion and integration system. An example of applying MFI system in an automations mobile robot/intelligent wheelchair system with video demonstration will also be presented.

1.2 Short Biographical Sketch

Ren C. Luo (IEEE M'82 - SM'87 - F'92), is currently a Professor and Dean of College of Engineering at National Chung Cheng University, he also served as Director of Automation Technologies Program at National Science Council and Advisor of Ministry of Economics Affairs in Taiwan, R.O.C. He was a Professor in the Department of Electrical and Computer Engineering and the Director of the Center for Robotics and Intelligent Machines at North Carolina State University in Raleigh, North Carolina, USA. He received his Ph.D degrees from Technische Universitaet Berlin, Berlin, Germany in 1982.

¿From 1983 to 1984, he was an Assistant Professor in the Department of Electrical Engineering and Computer Science at the University of Illinois at Chicago. From 1984 to 1990, he was Assistant, Associate Professor and became Professor since 1991 in the Department of Electrical Computer Engineering at North Carolina State University, Raleigh, NC. From 1992 to 1993, he was Toshiba Chair Professor at University of Tokyo, Japan.

Dr. Luo's research interests include: sensor-based intelligent robotics systems, multisensor fusion and integration, computer vision, rapid prototyping and advanced manufacturing systems. Dr. Luo has published over 170 technical journals, proceedings, and patents in the above-mentioned areas. He authored a book, Multisensor Fusion and Integration (Ablex, 1995); and was editor of the book, Robotics and Vision (IEEE, 1988). Dr. Luo was also guest editors for the Journal of Robotics Systems (John Wiley and Sons. Vol. 7, 3, 1990), IEEE Transactions on Industrial Electronics in special issues on the topics of multisensor fusion and integration for intelligent machines, and editor of IEEE/ASME Transactions on Mechatronics.

2 Plenary Speech II: "AI and Space Exploration"

Dr. Kenneth M. Ford Associate Center Director for Information Technology and Director of NASA's Center of Excellence in Information Technology, NASA Ames Research Center, Moffet Field, CA, USA

2.1 ABSTRACT

Humans are quintessentially explorers and makers of things. These traits, which identify us as a species and account for our survival, are reflected with particular clarity in the mission and methods of space exploration. The romance associated with the Apollo project is being replaced with a different vision, one where we make tools to do our exploring for us. We are building computational machines that will carry our curiosity and intelligence with them as they extend the human exploration of the universe.

In order to succeed in places where humans could not possibly survive, these "remote agents" must take something of us with them. They must be self-reliant, smart, adaptable and curious. Our mechanical explorers cannot be merely passive observers or puppets dancing on tenuous radio tethers from earth. They simply will not have time to ask us what to do: the twin constraints of distance and light-speed would render them helpless while waiting for our instructions, even if we knew what to tell them. AI plays a central role in space exploration because there is, literally, no other way to make it work. Our bodies cannot fly in the tenuous Martian atmosphere, endure Jupiter's gravity or the electromagnetic turbulence of Saturn's rings; but our machines can, and we will send them there. Once at distant worlds, however, they must deal with the details themselves. The only thing we can do is to make them smart enough to cope with the tactics of survival.

How clever will these agents of human exploration need to be? Certainly, cleverer then we can currently make them. It will not be enough to be situated and autonomous: they will need to be intelligent and inquisitive and thoughtful and quick. NASA is committed to integrating intelligent systems into the very center of our long-range strategy to explore the universe.

In this talk, I will describe the current and future research directions of NASA's expanding information technology effort with a particular emphasis on intelligent systems.

2.2 Short Biographical Sketch

Kenneth M. Ford is the Associate Center Director for Information Technology at NASA Ames Research Center and Director of NASA's Center of Excellence for Information Technology. In these roles, Dr. Ford has had the honor and responsibility of helping shape NASA's IT research effort (about 200M dollars effort at Ames, but much larger Agency wide). The Ames Research Center has about 5,000 employees, of which about a third work in IT and 700 have Ph.D degrees.

Additionally, Dr. Ford is the Director and Founder of the Institute for the Interdisciplinary Study of Human and Machine Cognition (IHMC) at the University of West Florida - a multidisciplinary research unit of the State University System. Since its founding in 1990, IHMC has rapidly grown into a well-respected research institute investigating a broad range of topics related to understanding cognition in both humans and machines with a particular emphasis on building cognitive prostheses to leverage and amplify human intellectual capacities. While at the University of West Florida Professor Ford received national and local recognition for teaching excellence and in 1997 he was awarded the University's highest research distinction, the Research and Creative Activities Award. Dr. Ford has been on a leave absence from the University to NASA for the last two years.

Dr. Ford entered computer science and artificial intelligence through the back door of philosophy. After studying epistemology as an undergraduate, he joined the Navy and wound up fixing computers among other things. When his Navy stint ended, he earned his doctoral degree in computer science from Tulane University in 1988. His research interests, among others, include: artificial intelligence, knowledge-based performance support systems, computer-mediated learning, and internet-based applications. Dr. Ford is the author of well over 100 scientific papers and the author/editor of five books.

Dr. Ford is the Editor-in-Chief of AAAI/MIT Press, Executive Editor

of the International Journal of Expert Systems, Associate Editor of the Journal of Experimental and Theoretical Artificial Intelligence, and is a Behavioral and Brain Sciences (BBS) Associate.

3 Plenary Speech III: "Music Enhances Learning: Keeping Mozart in Mind"

Dr. Gordon Shaw Professor Emeritus, Elementary Particle Theory Theoretical Neurobiology Department of Physics and Center for the Neurobiology of Learning and Memory University of California - Irvine CA, USA

3.1 ABSTRACT

Theoretical studies [Leng and Shaw, 1991], the "Mozart effect," based on the trion model [Shaw et al., 1985] predicted that music would enhance spatial-temporal reasoning (the ability to mentally image and transform patterns in space and time). Recent supporting experiments involving the Mozart Sonata for Two Pianos in D Major-K.448 are: behavioral studies showed that listening to it enhanced spatial-temporal reasoning in humans [Rauscher et al., 1993, 1995; Johnson et al., 1998] and in rats [Rauscher et al., 1998]; EEG studies [Sarnthein et al., 1997] showed that listening to it results in increased coherence lasting several minutes; exposure to it reduced pathological activity in comatose epileptic patients [Hughes et al., 1998]. MRI studies [Muftuler et al., 1999] showing excitation of cortex relevant to spatial-temporal reasoning. Studies relevant to education are: We [Rauscher et al., 1997] showed that preschool children who were given 6 months of piano keyboard training improved dramatically on spatial-temporal reasoning. Second grade children (in the inner-city 95 St. School in Los Angeles) given 4 months of piano keyboard training as well as training on Peterson's math video software scored striking higher [Graziano et al., 1999] on proportional math and fractions. Support for the trion model from cortical data [Bodner et al., 1997] show families of firing patterns related by symmetries. Implications for education, basic neuroscience, clinical medicine, and technology are discussed.

3.2 Short Biographical Sketch

Professor Shaw earned his B.S. from Case Institute of Technology in 1954 and his Ph.D. in Theoretical Physics from Cornell University in 1959. He had post-doctoral positions at Indiana University and the University of California, San Diego, and a teaching position at Stanford University before joining the new UCI campus in 1965. In addition to his research in elementary particle theory, he started working on brain theory in 1974. He is a member of the UCI Center for the Neurobiology of Learning and Memory.

4. Plenary Speech IV: "International Fusion: Changes and Approaches"

Franklin E. White, Jr.
Director, Program Development, Navy SPAWAR Systems Center,
Code D101, San Diego, CA, USA, Email: whitefe@spawar.navy.mil

4.1 ABSTRACT

In the current Information age, the potential for overwhelming availability of data largely without meaning has become a reality. Everywhere individuals and organizations are drowning in data and information and starved for knowledge and understanding. This is a problem that has become apparent worldwide in developed and developing countries. One of the keys to addressing this is data and information fusion. Fusion has long been the domain of a relatively small number of practioners in a largely classified endeavors within nations. This speech will address the changes in this world view that are coming about and discuss the burgeoning exchange of information about fusion on an increasingly global basis. It will also suggest some discipline and approaches essential to making fusion tools useful, and discuss some of the needed mechanisms and pitfalls as an international community comes together.

4.2 Short Biographical Sketch

Franklin E. White Jr. has spent 30 years with Navy as an officer and scientist. He has focused on integration and fusion efforts, has worked with Navy's Command, Control and Intelligence systems and is Chairman of the Joint Directors of Laboratories, Data Fusion Group. Mr. White has long term experience with Top Level architectures, serving on the team that developed the Copernicus Architecture and spending two years on detail to the Intelligence Community Management Staff (CMS) where he chaired the working group that developed the INTELINK information sharing concept. He has long been a supporter of international cooperation serving for 2 years at RAF Brawdy Wales, UK and temporarily at many European sites and is active in many international programs. He has spoken at international CIS symposia and AFCEA meetings. He is a long time member of AFCEA , SASA, The Naval Institute and Naval Intelligence Professionals and is currently the Director of Program Development at SPAWAR Systems Center San Diego.

Tuesday July 6

Session TA1 Naval Applications
Chair: Michael Larkin, Naval Undersea Warfare Center

Session TA2 Medical Applications
Chair: Robert Levinson, University of California at Santa Cruz

Session TA3 Decentralized Detection Systems
Chair: Jane O'Brien, Defense Evaluation & Research Agency, U.K.

Session TA4 Formal Methods for Information Fusion
Chair: Mitch Kokar, Northeastern University

Session TB1 Radar and Communication Applications of Fusion
Chair: Rick Blum, Lehigh University

Session TB2 Image Fusion I
Chair: Elisa Shahbazian, Lockheed-Martin, Canada

Session TB3 Fusion for Target Tracking I
Chair: Mohamad Farooq, Royal Military College of Canada

Session TB4 Classification I
Chair: Nageswara S. V. Rao, Oak Ridge National Laboratory

Session TC1 International Collaboration in Fusion Research and Development
Chair: James Llinas, State University of New York at Buffalo

Session TC2 Diagnostic Information Fusion
Chair: Kai Goebel, Information Technology Laboratory, General Electric

Session TC3 Fusion of Fuzzy Information
Chair: Daniel McMichael, University of South Australia

Wednesday July 7

Session WA1 Image Fusion II
Chair: P. Svensson, Defence Research Establishment, Sweden

Session WA2 Fusion Architecture & Management I
Chair: Alan Steinberg, Environment Research Institute of Michigan

Session WA3 Fusion for Target Tracking II
Chair: S. Musick, US Air Force Research Laboratories

Session WA4 Biological and Linguistic Models for Fusion
Chair: George Chapline, Lawrence Livermore National Laboratory

Session WB1 Knowledge-Based Techniques for Information Fusion and Discovery
Chair: Ray Liuzzi and Craig Anken, US Air Force Research Laboratories

Session WB2 Hardware for Information Fusion
Chair: Adrian Stoica, NASA Jet Propulsion Laboratory

Session WB3 Multisensor-Multisource Fusion for Object Tracking and Recognition
Chair: Shishir Shah, Wayne State University

Session WB4 Fusion for Target Tracking III
Chairs: X. Rong Li, University of New Orleans and T. Kirubarajan, University of Connecticut

Thursday July 8

Session RA1 Image Fusion III
Chair: Robert S. Lynch, Naval Undersea Warfare Center, RI, USA

Session RA2 Fusion Architecture and Management II
Chair: Ivan Kadar, Consultant, Interlink and Northrup Grumman Corp., USA

Session RA3 Information Fusion for Decision Support
Chair: Galina Rogova, Calspan/CUBRC, NY, USA

Session RA4 Fusion for Fault Detection and Diagnosis
Chair: Mark Alford, Air Force Research Laboratory-Rome, NY, USA

Session RB1 Management and Business Information Fusion
Chair: Nianyi Chen, Chinese Academy of Sciences, Shanghai, P. R. China

Session RB2 Multisensor Target Tracking and Recognition of Small-to-Medium Sized Targets
Chair: Oliver Drummond, Consulting Engineer, Culver City, CA, USA

Session RB3 Emerging Applications I
Chair: Xue-Gong Zhang, Tsinghua University, Beijing, P. R. China

Session RC1 Emerging Applications II
Chair: Peter Wide, Orebro University, Sweden

Session RC2 Classification II
Chair: M. Hinman, Air Force Research Laboratory, USA

Session RC3 Sensor Fusion for Automatic Target Recognition
Chair: Erik Blasch, Air Force Research Laboratory, USA

* Manuscript not received or withdrawn.

Session TA1
Naval Applications
Chair: Michael Larkin
Naval Undersea Warfare Center, RI, USA

Information Fusion in Undersea Warfare

Michael J. Larkin
Naval Undersea Warfare Center (Code 3111)
1176 Howell Street
Newport, RI 02841-1708

Abstract: The shift in emphasis in undersea warfare from open ocean to shallow water has complicated the objective of threat detection. Detection and classification of enemy submarines, torpedoes, and mines is much more difficult in the littoral environment, with its adverse acoustical characteristics. In an effort to solve his problem, new sensors have been developed, both acoustic (active and passive sonar) and non-acoustic (magnetic, laser, etc.). As a result, information about a particular contact is often derived from multiple sensors. The information obtained from an individual sensor may be only partially reliable, for example, in the case of a quiet threat in a noisy and cluttered environment. Thus, novel information fusion techniques are called upon optimally combine these sensors and to best detect and classify the threat. This paper will survey recent efforts to apply such techniques to undersea warfare. These applications, in general, fall into two categories: tracking (data association and state estimation), and classification of contacts. Particular examples from the Platform Acoustic Warfare Data Fusion (PAWDF) project will be highlighted.

Fusion Of Multi-Sensor Information From An Autonomous Undersea Distributed Field Of Sensors

Mark D. Hatch
Edward R. Jahn
Joan L. Kaina
SPAWAR Systems Center Code D722
53560 Hull Street
San Diego, CA 92152-5001

Abstract: *The Deployable Autonomous Distributed System (DADS) Intra-Field Data Fusion Project is developing technology to fuse sensor information from a field of autonomous sensor nodes and dynamically control the field of autonomous nodes. The field consists of three different types of nodes in littoral waters, which operate on batteries and communicate underwater via acoustic modems. Sensor nodes contain acoustic sensors, electric field sensors, and vector magnetometers. These nodes collect and process data, fuse the acoustic and electromagnetic data available within the node, and forward contact information to a master node. The master node fuses the sensor outputs and also controls the power usage in nodes throughout the field to maximize system lifetime. Data are sent to an operator site via the gateway nodes using RF communications. This paper will concentrate on the fusion and network control methodologies being developed for the master node that are unique to operation of such an autonomous field.*

Keywords: data fusion, undersea surveillance, automated classification, multiple hypothesis tracking, optimization, fuzzy logic, dynamic control, autonomous systems

1 Introduction

The Deployable Autonomous Distributed System (DADS) Intra-Field Data Fusion Project seeks to develop technology to support a field of autonomous sensors in shallow water. Technologies under development include the fusion of data within the field and control of the communications network and other functional processes to extend the life of the field. This project, sponsored by Dr. D. H. Johnson at the Office of Naval Research, is an integral part of a broader thrust which is addressing the other technologies required for the implementation of the overall DADS concept. The concept utilizes three different types of nodes, which make up a network. *Sensor nodes* are small nodes that sit on the ocean floor and contain acoustic sensors, electric field sensors, and vector magnetometers. Data are collected from the sensors, processed, and locally fused in the node. The node then forwards contact information to a *master node*, which controls the field and fuses the data it receives from the various sensor nodes. Master nodes send their data acoustically to *gateway nodes*, which communicate with a command center via RF communications. Each of the nodes will run on battery power and communicate with each other via underwater acoustic modems.

The development of a system concept for such a field presents many unique problems and thus provides opportunities for research and technology development. Among the unique problems are the following: a) energy limitations based on using batteries to power the nodes; b) the use of micro processors in the nodes, limiting the complexity and computational demands which can be expected for near real time signal and fusion processing; c) the variability of the environment in littoral waters significantly affects the consistency of sensor data acquisition required to support state estimation by the fusion engine; d) the configuration and density (of sensor nodes) of the field will often result in a paucity of data from sensor nodes, creating large gaps (in time/ distance) between track segments; e) difficulty in correlating tracks when reports from different sensor nodes are based on different sensors detecting the target or different attributes being reported; f) demands of the system to control the reporting of false alarms, g) the need for an automated classification capability.

This paper will describe several of the areas of technology development being pursued to address some of the unique problems posed by a DADS field.

2 Correlation

Due to potentially different configurations of the DADS field, a number of correlation processes and controls were considered for the fusion performed by the master node. Target kinematic, target attribute, and environmental measurements are the prime inputs. In a sparse field, the opportunities for sensor node coverage overlap are minimal or nonexistent. The issue becomes the lack of data due to long time periods between detection reports. In a densely spaced field where sensor coverages overlap, textbook correlation algorithms can be employed. In either case, the correlation of data and tracks from nonhomogeneous sensor types reporting different target attributes also requires a careful application of correlation methods.

2.1 Correlation Methods and Strategies

Because of large uncertainties associated with underwater measurements, a multiple hypotheses tracking (MHT) approach, developed by ORINCON Corp., has been selected as the fusion core. In this well-known concept, hypotheses are formed based on the association and correlation of sensor reports. The Munkres algorithm and a geo test are used to evaluate the data associations. Each hypothesis consists of a different combination of sensor reports, an association confidence, and a tracking confidence. This methodology allows for soft decisions to be made until more data are received. Drawbacks entail the use of more memory due to potentially large combinatorics and the addition of pruning rules to manage the hypotheses. An implementation using fuzzy control to provide efficient hypothesis management (see section 3) has been incorporated into the MHT.

To significantly reduce the data transmission from each sensor node and offload some of the fusion processing at the master node, intra-node fusion (with cross-cueing between acoustic and electromagnetic sensors) will be performed at the sensor node. This results in the reporting of tracklets that provide high confidence position, course, speed, and classification attributes. From the field level fusion perspective, they provide strongly correlated sensor reports, reducing the number of uncorrelated or weakly correlated detections that occur at the sensor node.

To address correlation of dissimilar sensor types, correlation in the fuzzy-conditioned Dempster-Shafer (FCDS) target classification algorithm (see section 4) uses a sensor-target attribute database and expert system heuristics to determine the probability of correct classification. This process requires initial correlation and clustering. Described in more detail in reference [1], the basic step of the attribute correlation is matching the measured attributes to existing database composite target profiles. This *attribute probability of association* is then combined with the *kinematic probability of association* to determine the report-to-track combination.

To reduce sensor detection "holes" in sparsely spaced fields, a master node control is being designed (see section 5). When appropriate, the master node would be able to direct selected sensors at the sensor nodes to reduce their detection threshold, thereby increasing probability of detection and hence the sensor area of coverage. Though this also results in an increase in false alarms and potentially in communications, the benefit is in the fact that more detections allow for more correlation opportunities. System trade-offs between this benefit and other disadvantages are being studied through modeling and simulation.

Lastly, correlation processes can be vastly improved by using *in situ* environmental information as well as information external to the field. At the field level (master node), knowledge about the environment can be exploited to control processing at the nodes and provide the best opportunities for correlation of sensor detections. Likewise, external or INTEL information about likely targets in the area can be used to adjust correlation confidences.

2.2 Summary

Once the mission and sensor node spacing have been selected prior to deployment of a DADS field, the appropriate correlation methodologies can be selected as part of the configuration package. These methodologies, of course, must be tested in conjunction with the target state estimator for optimal fusion performance. Once deployed, to allow for the greatest flexibility in changing shallow water environments and unpredictable target movements, adaptive controls should be applied.

3 Distributed Autonomous Tracking

The selection of an MHT for the DADS fusion engine provides the project with an already developed product. A need to make it more efficient in addressing the DADS requirements dictated minimization of computational demands while maintaining a high level of performance in a fully automated environment. A study was undertaken to assess the benefits associated with maintaining large numbers of hypotheses when operating in a DADS-like environment. Results reported in a paper presented at Fusion 98 [1] indicate that a single hypothesis approach (nearest neighbor tracker) had poor performance for sparse field configurations but that a limited hypothesis tracker (3 hypotheses) often performed comparably to the full MHT. The recognition of cases where limiting the tracker to three hypotheses resulted in reduced performance suggested the development of adaptive methods for pruning hypotheses.

Independently, an effort was initiated at the Center for Multisource Information Fusion (CMIF) at SUNY, Buffalo, to study fuzzy logic methods for their applicability in addressing some of the problems associated with tracking targets in a DADS environment.

3.1 Fuzzy Control of Multiple Hypothesis Tracker Parameters

The overall performance of the fusion engine depends upon the set of parameters that are used by the MHT. Although a static set of parameters may work well over a wide range of scenarios, they will not lead to optimal performance in all cases. In an effort to improve performance of the data fusion system at a master node, a fuzzy logic controller was developed to adaptively tune some of the parameters. To date, two fuzzy logic algorithms have been developed [2] that modify the tuning parameters of the ORINCON MHT. The first parameter is a sliding window length used for cluster N-scan pruning and the second is the amount of process noise to inject into the Kalman filter for target maneuver tracking.

Cluster N-scan pruning is a technique used for efficient hypothesis management. The algorithm uses a sliding window of length N to prune away poor branching hypotheses. Cluster N-scan pruning forces a hard decision on all measurements in the (N-1)-st oldest scan. It therefore allows the MHT algorithm to carry multiple hypotheses on the most current data and make hard decisions on older data. A Fuzzy Logic Controller (FLC) to adapt the N-scan length of each individual cluster allows each cluster to carry its own window length as needed to resolve its own ambiguity. Since each cluster contains a different N-scan length, the overall number of hypotheses carried by the MHT is reduced. The reduction occurs when a cluster that contains little or no ambiguity, carries only a few hypotheses and has an N-scan length of one to three. The number of hypotheses used by the MHT algorithm is influenced by two key values: the number of system tracks (in the cluster) and the amount of contention among these existing tracks for incoming measurements. The contention value is calculated as the average normalized residual among all pairs of tracks in the cluster. This calculation is done only after all track states and covariances are predicted to the current time. Each of these two input variables uses five input membership functions: 1) the number of targets is defined as either None, Few, Some, Many, or Numerous, 2) the contention is defined as either Low, Medium-Low, Medium, Medium-High, or High. Finally, the output variable N-scan length is defined as one of the seven possible values: Very-Short, Short, Medium-Short, Medium, Medium-Long, Long, or Very-Long. Each of the 25 possible combinations of the variables is mapped into one of the seven possible values for N-scan

length by the fuzzy rule inference engine. Then, using a defuzzification procedure, a single value for N-scan length is obtained that is used as a parameter within the MHT algorithm. This process is performed each time a cluster of tracks is updated based upon new measurement reports.

The process noise of a Kalman filter is used to account for mismodeled dynamics, unmodeled modes, and noise in the system model. The motion model used in the MHT is constant course/constant speed. If a target maneuvers from its present course, the straight-line motion model is incorrect and possible track fragmentation occurs. A way to track through the maneuver is to add process noise to the Kalman filter prediction step. The amount of process noise used in the Kalman filter is acceleration-dependent for all targets. A fixed amount of process noise has ill effects on targets moving at different accelerations. A Fuzzy Logic Controller (FLC) to adapt the process noise to account for targets with different acceleration allows the filter to track through significant maneuvers without compromising the tracking accuracy for cases of minor maneuvers. A typical maneuver involves a change of velocity and hence this information is used as input into the process noise calculation using an estimated acceleration value. This value is simply calculated as the difference between the velocity of the track estimate and the velocity of track-level measurement. In this way, the process noise is able to enlarge in situations in which there are significant differences in velocity between the track and measurement states. However, in situations where the velocity terms agree, the process noise is able to remain small.

3.2 Fuzzy Logic Based α–β Tracker

A wide variety of estimator forms have been developed to deal with the target tracking problem. One of the early forms, a so-called fixed-coefficient estimator called the α–β filter, has been employed on many operational systems. In spite of its simplicity and limitations, it continues to be of interest. The work at CMIF/SUNY, Buffalo explored many aspects of developing a fuzzy logic gain modified α–β filter. The focus of recent efforts was control-theoretic based analysis of the fuzzy α–β filter. A detailed characterization of the development of this filter is provided in another paper presented at this symposium (and included in the Proceedings) [3].

Unlike the fixed gain α–β filter, the fuzzy logic based α–β filter changes the smoothing parameters, α and β, as a function of the maneuver error and error rate to provide tracking performance comparable to a Kalman filter, at least for the types of ASW scenarios evaluated. Furthermore, the computational cost is less than that of the Kalman filter.

The maneuver error can be defined as the difference between the observed position and the predicted position of the target. The error rate can likewise be defined as the difference between errors for successive observations. Singh [3] defines the input membership functions with seven error and seven error rate input sets (positive/negative large, medium and small plus zero) requiring a minimum of 49 rules. He then exploits an analogy from system control theory, the rest-to-rest maneuver of a second order system to define appropriate rules. The control law is modeled in the form of a non-linear spring-damper system. A transfer function for the spring-damper system is developed in the time domain relating the position and velocity of the mass to the undamped eigenfrequency and the damping ratio. The results of this work develop the 49 rules to relate the error and error rate to small, medium, large or zero values for the undamped eigenfrequency and the damping ratio. This results in the development of stiffness and damping control surfaces defined for variation in error and error rate. Finally, a transformation is made to provide the input-output-relationship between maneuver error and error rate and the smoothing parameters α and β. The adaptive α–β filter developed is thus capable of tracking various types of maneuvers whereas the fixed α–β filter can only be optimized for one type of maneuver and level of sensor noise.

The proposed fuzzy α–β filter was evaluated against several alternative trackers (filters) using a tracker testbed developed at CMIF/SUNY Buffalo. The tests compared five tracking filters; (1) a fixed α–β filter, (2) a fixed α–β–γ filter, (3) Chan's α–β FL filter [4], (4) the proposed FL α–β filter, and (5) a Kalman filter. Each were evaluated on four realistic benchmark target maneuvers: (1) targets moving with constant speed on a straight line, (2) targets moving with constant acceleration on a straight line, (3) targets moving with constant speed on a single gradual turn, and (4) targets moving with constant acceleration on a single gradual turn. The benchmarks were also run with three different sensor distributions (dense, medium, and sparse). Examples of the results (tracker error) obtained are shown in Table 1. While other, more dramatic maneuvers may give

Table 1: Simulation Results for a Medium Sensor Field at 60 [s] Sampling Time

Filter	Maneuv. 1		Maneuv. 2		Maneuv. 3		Maneuv. 4	
	Mean	Var	Mean	Var	Mean	Var	Mean	Var
1	381	108535	700	411456	749	541677	1523	1635908
2	1786	9770645	1135	1159524	1050	1159934	1612	2200747
3	583	307365	1172	1386552	753	562431	1249	1443696
4	378	111253	666	402310	687	522226	1548	1845937
5	405	123283	1173	990780	809	488069	1629	2054990

different results, the results for these maneuvers for the other field configurations were comparable to those in Table 1 and demonstrated the proposed FL α–β filter to be a viable tracker for such applications. In addition, a counter for floating point operations was actuated to provide an estimate of the computational costs associated with the respective filters. Table 2 shows the comparison of FLOPS used by the trackers for equivalent operations (20 scans). The proposed FL α-β filter required approximately 75% less floating point operations than the Kalman filter while providing comparable performance. It also out performed the other trackers evaluated for maneuvering targets.

Table 2: FLOPS used by the Target Tracker

	Tracker 1	Tracker 2	Tracker 3	Tracker 4	Tracker 5
FLOPS	470	974	2021	2287	8282

4 Automated Classification

Automatic target classification is a critical function of an autonomous system of sensors because of the lack of an operator in the loop. The issues to be resolved not only address the reduction of false alarm reporting from the field, but also the levels of target classification refinement and their associated uncertainties. Because of the use of multiple sensor types in DADS, the approach selected uses multi-sensor parametric attribute information reported by the individual sensor types. The centralized nature of the DADS fusion architecture drives the primary target classification process to take place in the master node.

Emulating the human thought process, the automated classification approach requires two components: 1) the data bases of sensor attributes and target characteristics, and 2) a process to combine the received information. Detailed classification databases were developed to compare the parametric data to previously collected data from a variety of targets. The fidelity of both the data and the databases determine how well a system can ultimately classify targets. The detail in the databases determines how refined a classification estimate can be produced. For this reason, DADS employed Summit Research Corporation to construct accurate, detailed databases of acoustic, magnetic, and electric field sensor attributes for classification of ASW targets.

The second component required for the target classification in DADS is the algorithm that fuses the measured sensor attributes using the data bases to determine an overall classification estimate. Lockheed Martin has developed the fuzzy conditioned Dempster-Shafer (FCDS) algorithm for this purpose. FCDS, a fully probabilistic theory consistent with Bayesian probability theory, is a Dempster-Shafer like methodology for reasoning with ambiguous, imprecise/vague, and non-disjoint evidence. Unlike Dempster-Shafer, FCDS is also capable of incorporating *a priori* knowledge of targets. The FCDS classification procedure is based on a method of mathematically modeling imprecision/vagueness in parametric

attributes as random fuzzy sets. Both the databases and the FCDS algorithm have been discussed in detail in reference [1].

5 Optimization and Control

The goal of the DADS Network Control and Optimization task is to increase field lifetime by reducing power consumption while maintaining field level detection capability. This task is divided into two parts: processing optimization, which limits the consumption of battery power by controlling the processing in the nodes, and communications network control, which attempts to reduce power by dynamically routing communications from the sensor nodes to the master node.

5.1 Processing Optimization

The basic idea behind processing optimization is to intelligently determine in which of five primary processing modes a node should be set, and to determine the detection thresholds for processing at each sensor node. The five primary modes are processing, relay-only, detect-only, sleep, and dead. Processing is when the node is processing data from some or all of its sensors, generating and sending its own reports, and relaying reports from other nodes. Relay-only mode is when the sensor node is relaying reports from other sensors, with no processing or detection of its own. A node in detect-only mode processes data from its sensors and generates and sends its own reports, but does not relay messages. Nodes in sleep mode are still alive, but only have the ability to receive wake up signals. A dead sensor has used up all of its battery power, and therefore cannot detect nor relay messages. For nodes in processing and detect-only modes, a detection threshold for each sensor in that node must be set. These thresholds are set to determine how clearly a signal (from a target of interest) must be distinguishable from noise to be reported. High detection thresholds decrease the probability of detection, and also decrease the probability of false alarms. Low detection thresholds increase both probability of detection and false alarms. Thresholds will be set to increase the probability of detecting targets in areas of the field where targets are expected to be located, while reducing the number of detections and false alarms (and therefore messages generated) in areas where targets are not expected.

To date there have been two different approaches proposed to control processing. The first is a simple heuristic based on remaining battery power, and the second seeks to limit false alarms while increasing detections in areas of special interest. Both will be discussed in the next section.

5.1.1 Processing Optimization Methods

Wagner Associates has developed a simple heuristic for first level control of the processing modes for the sensor nodes. In each node, a battery power threshold is set. When the remaining battery power for a node in the processing mode exceeds this threshold, it is switched to relay-only mode. The node remains in relay-only mode until the threshold is changed or the node dies. The master node can alter the battery power threshold, and will do so for a number of reasons. If the current threat condition is high, for example, the master node may tell a sensor node to decrease its threshold, thus keeping it in the processing mode longer. Alternately, the threshold may be increased when the threat condition is low for the field or individual node.

Another approach to processing optimization involves maintaining a constant false alarm rate (CFAR) for the field. The idea is to maximize the probability of detection while maintaining a CFAR (or alternatively a constant probability of detecting a target) throughout the field while maximizing lifetime. This will be accomplished by lowering detection thresholds for the sensors on sensor nodes that have been alerted to possible threats in the area. Lower detection thresholds increase both the probability of detection and the false alarm rate for those nodes. In order to maintain a field level CFAR, other sensor nodes will have to increase their thresholds, or switch to a different mode, such as relay-only or sleep.

This will impact power consumption in several ways. First, nodes in areas where targets are currently not expected may change to a relay-only or sleep mode. Other nodes may have their detection thresholds increased. This will lower the number of messages generated and sent from such nodes. Not only will this save on power consumption at that node, but also at other nodes along the path to the master node.

In order to develop a CFAR algorithm (or alternatively a constant probability of detection algorithm), several parameters will have to be defined and modeled. There are a number of ways to model field level probability of detection. One can determine that a detection (at a field level) is made when at least m sensors (or sensor nodes) in the field detect a target. Another possibility is to call a detection if at least one sensor has detected a target and that target is classified with a high confidence. In like manner, field level false alarm rate also needs to be modeled.

After models for probability of detection and false alarm rate are created, a method for solving the CFAR objective function will need to be established. The objective function will maximize field level probability of detection subject to the constraint that probability of false alarm remains constant [5]. Efforts to adapt this objective function to the DADS field for the purpose of maximizing field lifetime are currently in their infancy.

5.2 Communications Network Control and Optimization

While it is believed that the gain in field lifetime by processing optimization will be significant (particularly when processing control drives the amount of reporting and hence the communication requirements), the dynamic control of the communications network may in itself prove to be very significant in increasing field lifetime. Dynamic control of the DADS communications network consists of the initial assignment of routes from each sensor node to a master node, and the adjustment of these routes as time progresses.

When the DADS field is initialized, a routing table will be produced that lists every node to which an individual node can talk. This table will be stored in the master node. From this table, an initial routing of the field will take place to connect every sensor node to a master node. It is expected that as time progresses, these routes will need to change, in order to prevent some portions of the field from burning out faster than others. The master node will maintain a database with estimates of each node's remaining energy, and will periodically poll the routing algorithm to check if rerouting is in order. The algorithm will change routes only when it is determined that doing so will increase the field lifetime.

5.2.1 Control Techniques

The communications routing algorithms developed for DADS will determine the best route from each sensor node to the master node for the purpose of extending the lifetime of the field. Routes will be updated as needed in order to extend field lifetime. Two separate algorithms are under development in order to determine the optimal routing strategy. These are a one step rollout algorithm (simplified Neural Dynamic Programming) and a genetic algorithm (GA).

The rollout algorithm is an approach to stochastic control using dynamic programming [6]. The rollout algorithm seeks to minimize, over all possible control strategies, a *cost-to-go* function, which is the expected cost to termination from each state of the system. The cost-to-go is given by Bellman's equation,

$$J^*(i) = \min_u \sum_{j=1}^{n} p_{ij}(u)\left[g(i,u,j) + J^*(j)\right],$$

where

$p_{ij}(u)$ is the probability of transitioning from state i to state j given control strategy u,

$g(i,j,u)$ is the cost of transitioning from state i to state j given control strategy u,

and $J^*(j)$ is the cost-to-go from state j.

The rollout algorithm estimates the cost-to-go using a base heuristic.

In DADS, the algorithm works as follows: a simple algorithm, such as a minimum hop

algorithm, is used to determine the initial routing. When an update is requested the algorithm creates a large number of candidate routings, including the current route, and calculates the expected lifetime for each candidate routing. The lifetime is calculated assuming that the field will maintain this route for time T, and then revert to some base heuristic. There are currently two candidates for the base heuristic: 1) keep the current routing or 2) revert to the initial routing. Testing of the system will determine which heuristic yields the best result. Included in the calculation is the drain on the power in order to reroute. The routing candidate with the maximum expected lifetime is chosen, unless it fails to show significant improvement over the current route.

Much earlier in development is the Genetic Algorithm (GA). Genetic algorithms attempt to model the biological processes of natural selection, also known as "survival of the fittest", in order to reach an optimum. Work on a Genetic algorithm for optimization is currently being performed under an ONR SBIR. Once complete, the GA technique will be compared to the rollout algorithm, and the best will be chosen for use in the DADS communications network control strategy.

5.3 Summary

Minimizing the amount of energy used by the DADS field is necessary in order to maintain the lifetime of the field. At the same time, field capability must not be degraded. Maintaining a high probability of detecting targets while constraining the FAR and reducing power consumption will prove to be a valuable approach for extending the usefulness of the field.

Intelligently controlling the communications network to maximize field lifetime should also prove to be of great benefit to a DADS field. Both the rollout algorithm and the Genetic Algorithm are expected to provide good solutions to the problem. Results from simulated test cases of the rollout algorithm are due shortly. These should provide insight into both the usefulness of dynamic routing in general and the rollout algorithm in particular. The GA is still in early stages of development, and tests and results are not expected until CY2000.

6 Summary

Developing fusion technology to deal with the limitations of the DADS field results in the creation of interesting fusion and control algorithms. The methods developed in this project will be tested in a virtual environment in the near term, with at sea tests planned for the future. Eventually, these techniques will be targeted for transition into fielded Navy systems.

7 References

[1] Mark Hatch et al. *Data Fusion Methodologies in the Deployable Autonomous Distributed Systems (DADS) Project.* First International Conference on Multisource, Multisensor Information Fusion (Fusion 98), July 1998.

[2] Peter Shea, Mark Owen. *Fuzzy Control in the Deployable Autonomous Distributed System.* 13th Annual SPIE International Symposium on Aerospace/Defense Sensing, Simulation, and Controls (Aerosense 99), April 1999.

[3] Tarunraj Singh, *Formally Derived Characterization of the Performance of Alpha-Beta_Gamma Filters.* Secomd International Conference on Information Fusion (Fusion 99), July 1999.

[4] Jim Llinas, Tarunraj Singh. *Fuzzy Mathematics and Logic for Multisource Data Association and Target Tracking (Phase II).* Report, Grant N00014-97-1-0584/P00001 September 1998.

[5] C. H. Gowda et al. "Distributed CFAR Target Detection." *Journal of the Franklin Institute: Special Issue on Information/ Decision Fusion with Engineering Applications.*

[6] D. P. Bertsekas and D. A. Castanon. "Rollout Algorithms for Stochastic Scheduling Problems." *Heuristics,* 1996.

Target Detection Performance by Fusing Information From Tracks Generated by Independent Waveforms

Robert S. Lynch, Jr.*
Code 3113, Naval Undersea Warfare Center
Newport, RI 02841

Abstract *In this paper, several methods are demonstrated and compared for detecting targets by fusing information from tracks generated by independent continuous wave (CW) and frequency modulated (FM) waveforms. Performance of each method is illustrated using operating characteristic type curves that are based on an average of over 2200 pings of real active sonar data. The results of this comparison reveal that performance improves, over that of either an OR detector or a track association test, when a classification approach is adopted for information fusion. Specifically, the Bayesian Data Reduction Algorithm (BDRA) is applied to the data, which selects the features from both waveforms yielding best target detection performance.*

Keywords: Active sonar, Target tracking, Feature selection

1 Introduction

The subject of this paper is the comparison of several methods for detecting targets by fusing information from tracks generated by independent continuous wave (CW) and frequency modulated (FM) waveforms (i.e., sonar echoes whose purpose is to track and detect various surface ships and submarines). With the first of these methods, individual detection decisions of the CW and FM sequential (kinematic log likelihood ratio (KLLR)) detectors are fused by an OR detector. In the next method, information fusion is accomplished by associating the CW and FM tracks using a Chi-squared track association test. Finally, a classification approach is adopted for information fusion by using as features the KLLR and Chi-squared test statistics of the previous two methods, and Doppler information. Also, with this latter method, those features which yield best target detection performance are found using the Bayesian Data Reduction Algorithm (BDRA), [4]. Performance of each test is illustrated using plots of the total number of detected target tracks verses the number of false alerts per hour. Additionally, all results are based on an analysis of over 2200 pings of real active sonar data (obtained in several littoral environments), which represents a time duration of approximately fifteen hours. As it turns out, the BDRA shows the best performance followed, respectively, by the Chi-squared track association test and the OR detector.

The methods used to fuse information from the CW and FM tracks are detailed in the sections below. However, before describing these methods a few observations are made about the track generation process for each waveform. Note, at each ping the CW and FM waveforms are transmitted from the source as a wavetrain (a delay of approximately one half seconds exists between successive pings). Then, upon reception and prior to any track information fusion, each waveform is independently processed by a processing chain which consists of matched filtering, normalization, clustering, and shape filtering. From here, the processed CW and FM waveforms are detected and tracked by separate Automatic Detect and

*Supported by an NUWC In-House Laboratory Independent Research (ILIR) Grant, and by the Platform Acoustic Warfare Data Fusion (PAWDF) Project.

Tracking (ADT) algorithms. In this case, the ADT contains a sequential kinematic log likelihood ratio (KLLR) detector and a Kalman filter based interacting multiple model (IMM) tracker, [1]. Finally, the CW and FM track pairs are time aligned (by time index shifting) in order to be in the correct format for information fusion.[1]

2 Description of the Methods Used for Track Information Fusion

2.1 OR Detector

As mentioned in the previous section each ADT contains a sequential kinematic log likelihood ratio (KLLR) detector. The KLLR detector is based on track innovation, which is the difference between a measured track and its prediction (produced by the ADT). In this method of track information fusion the decisions of the CW and FM KLLR detectors are fused using a logical OR. Thus, this method of fusion depends on the accuracy of each individual detector, and as will be seen below performance can be substantially degraded if one of the detectors has a high false alert rate.

2.2 Chi-squared Track Association Test

In the next method of information fusion track pairs are associated by a Chi-squared track association test, which is based on the normalized (by the estimation errors) product of the difference between the individual CW and FM track state estimates (a four dimensional vector of position and velocity estimates in two dimensions). In particular, track association begins by first forming the difference between the track state estimates of CW and FM (see [1]),

$$\mathbf{\Delta}(n) = \mathbf{x}^{CW}(n) - \mathbf{x}^{FM}(n) \qquad (1)$$

and, assuming independent tracks at the n^{th} time index (ping), the sum of their estimation error covariance matrices given by

$$\mathbf{T}(n) = \mathbf{P}^{CW}(n) - \mathbf{P}^{FM}(n). \qquad (2)$$

Then, using formulas (1) and (2) the track association test is performed using the statistic

$$[\mathbf{\Delta}(n)]'\,[\mathbf{T}(n)]^{-1}\,[\mathbf{\Delta}(n)] < D_\alpha \qquad (3)$$

where the left side of formula (3) is a Chi-squared random variable with four degrees of freedom (i.e., the number of elements in the track state vector). Note, the threshold D_α is set from a Chi-squared table (for example, see [2]) for an α probability of missing a valid association.

Intuitively, it can be seen that an advantage of state association (as compared to an OR detector) based on the Chi-squared test is that more information is used in the detection decision (i.e., the four components of the target state). However, a shortcoming of this method is that the target must exist in both tracks in order for a detection to be declared. Thus, state association tends to be opportunity limited, and this is evident in the results below.

2.3 The Bayesian Data Reduction Algorithm

The Bayesian Data Reduction Algorithm (BDRA) uses the Dirichlet distribution, [3], as a noninformative prior. The Dirichlet represents all symbol probabilities as uniformly-distributed over the positive unit-hyperplane. Using this prior the algorithm works by reducing the quantization complexity, M, to a level which minimizes the average conditional probability of error, $P(e \mid X)$. The formula for $P(e \mid X)$ is fundamental to the BDRA, and for two classes it is given by (see, [4, 5])

$$P(e \mid X)$$

[1] Tracking errors for the CW and FM tracks are assumed independent in both the measurement and the state.

$$= \sum_{\mathbf{y}} \sum_{X} P(H_k)\, \mathcal{I}(z_k \le z_l)\, f(\mathbf{y}|\mathbf{x}_k, H_k) + P(H_l)\, \mathcal{I}(z_k > z_l)\, f(\mathbf{y}|\mathbf{x}_l, H_l) \quad (4)$$

where (note, k and l are exchangeable)
$z_k = f(\mathbf{y}|\mathbf{x}_k, H_k)$
$= \frac{N_{\mathbf{y}}!(N_k+M-1)!}{(N_k+N_{\mathbf{y}}+M-1)!} \prod_{i=1}^{M} \frac{(x_{k,i}+y_i)!}{x_{k,i}!y_i!}$;
$k, l \in \{\text{target, nontarget}\}$, and $k \neq l$;
$H_k : \mathbf{p}_{\mathbf{y}} = \mathbf{p}_k$;
M is the number of discrete symbols;
$X \equiv (\mathbf{x}_k, \mathbf{x}_l)$ is all training data;
$x_{k,i}$ is the number of the i^{th} symbol in the training data for class k, and $N_k \left\{ N_k = \sum_{i=1}^{M} x_{k,i} \right\}$;
y_i is the number of the i^{th} symbol in the test data, and $N_{\mathbf{y}} \left\{ N_{\mathbf{y}} = \sum_{i=1}^{M} y_i \right\}$.

Notice that only cases involving one test observation (i.e., $N_{\mathbf{y}} = 1$) are considered here so that $f(\mathbf{y}|\mathbf{x}_k, H_k)$ of formula (4) becomes

$$f(y_i = 1|\mathbf{x}_k, H_k) = \frac{x_{k,i}+1}{N_k + M}. \quad (5)$$

Given formula (4) the algorithm is implemented by using the following iterative steps.

1. Using the initial training data with quantization M, formula (4) is used to compute $P(e \mid X; M)$.

2. Beginning with an arbitrarily selected feature, sum (i.e., merge) the training data of those quantized symbols that correspond to its reduction (e.g., in the binary case, merge those quantized symbols containing a binary zero with those containing a binary one).

3. Use the newly merged training data, X', and the new quantization, M', and again compute $P\left(e \mid X'; M'\right)$.

4. Repeat items two and three for all adjacent feature quantizing levels, and all remaining features.

5. From item four select the minimum of all computed $P\left(e \mid X'; M'\right)$ (break ties arbitrarily), and choose this as the new training data configuration for each class.

6. Repeat items two through five until the probability of error decreases no further, or until $M' = 2$.

As will be seen in the results below, the motivation for using the BDRA is to select the best combination of features that simultaneously overcomes the opportunity limitations of the Chi-squared test, and the high false alert rate of the OR detector.

Note, before discussing performance results for methods shown here it is pointed out that the data first had to be correctly labeled by identifying true targets. This was accomplished manually by comparing the similarity of estimated tracks to those of the Global Positioning Satellite (GPS). Therefore, any track not identified as a true target (surface ship or submarine), by default, automatically was labeled a nontarget (this latter class is made up of background disturbances such as shipping noise and clutter).

3 Results

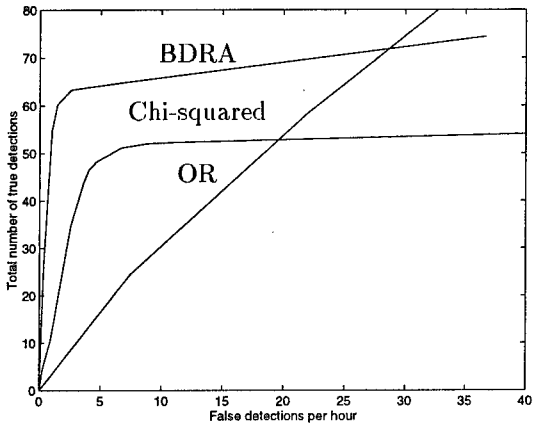


Figure 1: Target track recognition performance comparison of the OR detector, the Chi-squared test, and the BDRA.

In Figure 1, operating characteristic (OC) curves appear showing the total number of true detected targets (i.e., target track fusions) versus the number of false detections (i.e., false

track fusions) per hour for the OR detector, the Chi-squared test, and the BDRA.[2] Notice, the OC curve for the OR detector was obtained by simultaneously varying the thresholds of both the CW and FM detectors from -2.3 to 30. However, note that in the results for this detector many nontarget tracks are counted more than once because the true identity of these tracks can not be established (i.e., all nontarget detections for CW and FM are added together). Also, the OC curve for the Chi-squared test was obtained by varying the threshold D_α (see formula (3)) from 0 to 100,000. In this case, the only situation that is counted as a valid detection is when the same target is contained in the CW and FM track pair which passed association. Therefore, potential false track fusions for the Chi-squared test are all CW and FM target and nontarget track pairs (at a given ping) which do not have the same true target label. Before the results contained in Figure 1 are discussed further the application of the BDRA to the data for feature reduction (and selection) is described next.

The first step in applying the BDRA to the data was to form the following five dimensional feature vector,

{ Chi-squared statistic, CW Doppler, FM Doppler, CW KLLR, FM KLLR }

where the Chi-squared statistic and KLLR are shown above, and the additional features have the following descriptions.

CW Doppler (knots) is measured from the CW processor.

FM Doppler (knots) is estimated from range rate.

Based on this feature vector the data were then partitioned into a training set consisting of 5774 samples of which $N_{target} = 848$, and $N_{nontarget} = 4926$. Actually, the original data contains more than fifty thousand track pairs that can be considered of the nontarget category (i.e., all track pairs which can potentially be tested for association). However, a form of track pruning, or gating, was employed to substantially reduce this number by ordering all Chi-squared statistics. That is, for each track only the smallest Chi-squared statistic was accepted (the track it most closely associated with), and all other larger Chi-squared statistics involving this track were rejected (all other tracks it might also have been associated with).

[2]All results shown are based on converting detected target pings to detected target tracks using the average number of pings contained in a track. Also, results for the BDRA are determined by testing on the training data.

Table 1: Threshold Settings for Each Feature Before Applying the BDRA

level	Chi-sq.	Doppler	KLLR
1	7.78	1	2.3
2	50	5	6.8
3	100	10	20
4	100K	70	30

At this point, before actually applying the BDRA to the data it was necessary to threshold each feature into an initial set of discrete levels. This thresholding was based on experience examining the data, and as a result, four thresholds were chosen for each feature. Thus, with four discrete levels per feature the initial quantization complexity of this data was $M = 1024$. Table 1 lists these thresholds where at each discrete level the upper bound is shown, and the lower bound is defined in the next lower level (note, the Doppler and KLLR columns represent both CW and FM).

After the BDRA was applied to the data the initial quantization complexity of $M = 1024$ was reduced to a final quantization complexity of $M = 8$. With this, the computed empirical probability of error (see formula (4)) was reduced from 0.325 to 0.117. In reducing this data, it was found that the BDRA completely removed the FM features. Addi-

tionally, it reduced the Chi-squared statistic, CW Doppler, and CW KLLR to binary valued features keeping, respectively, the thresholds of 7.78, 1, and 2.3. Thus, for correct target recognition the BDRA prefers to rely mostly on CW, and it only uses FM when it associates with CW through the Chi-squared statistic. Notice, this is consistent with the fact that FM is known to perform poorly in this data.

Now, continuing with the results in Figure 1, it can be seen in this figure that for low rates of false alert (the area of most interest) the BDRA is able to improve performance over the other methods. Notice, it is apparent that the high false alert rate of FM is degrading the performance of the OR detector. Also, the opportunity limitations of the Chi-squared test are obvious because the target must exist in both waveforms in order for this test to detect it. However, the BDRA overcomes these limitations by selectively choosing those features associated with best performance.

4 Summary

In this paper, several methods were applied to fusing information from sonar echoes which were produced by independent CW and FM waveforms. It was shown that a classification approach (using the BDRA for feature selection) was more effective at correct target recognition than either a Chi-squared test, or an OR detector.

References

[1] Y. Bar-Shalom and X. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*, Course Notes, University of Connecticut, 1995.

[2] G. Casella and R. L. Berger, *Statistical Inference*, Duxbury Press, Belmont, California, 1990.

[3] G. F. Hughes, "On the Mean Accuracy of Statistical Pattern Recognizers," *IEEE Transactions on Information Theory*, vol. 14, no. 1, January 1968, pp. 55-63.

[4] R. S. Lynch, Jr., *Bayesian Classification Using Noninformative Dirichlet Priors*, Ph.D. Dissertation, Major Advisor, P. K. Willett, University of Connecticut, May, 1999.

[5] R. S. Lynch, Jr. and P. K. Willett, "Bayesian Classification and Data Driven Quantization Using Dirichlet Priors," *Proceedings of the* 32^{nd} *Annual Conference on Information Sciences and Systems*, March 1998.

Signal Estimation Using Selectably Fused Sensor Data of Varying Cost

Dana Sinno
Telecommunications Research Center
Arizona State University
Tempe, AZ 85287-7206 USA

Douglas Cochran
Department of Electrical Engineering
Arizona State University
Tempe, AZ 85287-7206 USA

Abstract *The problem of optimally estimating the state of a stochastic linear dynamical system using a library of linear sensors (observation maps) of varying costs is addressed. The role of sensor cost in determining the optimal sensing strategy is illustrated by several examples. The marginal trade-off between sensor gain and cost in estimation of a system operating near steady state leads to the notion of sensor value which serves as the basis for sensor selection in the attentive sensing strategy. This suggests the possibility of analytically quantifying sensor value for well-defined scenarios in future work, thereby allowing libraries of sensors suitable for estimation tasks to be identified prior to deploying sensor suites.*

Keywords: attentive sensing, estimation, Kalman filtering, data fusion

1 Introduction

Recent research has shown how sensor data should be chosen for optimal iterative estimation of the state of a discrete-time linear stochastic system when linear measurement maps can be selected from a pre-determined library in each iteration [1, 2]. In prior work, the library of measurement maps represents a collection of available sensors from which the best combination must be selected at each iteration time due to resource constraints (e.g., communication bandwidth or computational power) or sensor constraints (e.g., ability to operate in only one mode at any given time). The goal has been to achieve a sensing strategy that minimizes some function of the estimation error covariance matrix at each iteration of the estimator – without consideration to costs or risks associated with the sensing strategy.

This paper extends earlier work by introducing sensing cost as a factor in the selection of a sensing strategy. In this setting, some measurement maps may be more expensive to use than others in terms of risk or monetary cost and strategies that are optimal with respect to criteria incorporating both cost and estimation performance are sought.

Depending on the specific scenario involved, there are several reasonable ways in which sensing cost can be considered in optimizing a sensing strategy; e.g.,

1. Choose the strategy of lowest overall cost that will satisfy a pre-established criterion on estimator performance.

2. Choose the strategy that achieves the best estimation performance subject to a cost constraint.

3. Minimize an objective functional that includes both overall cost and estimation error at a pre-set terminal iteration.

4. Perform step-by-step minimization of such an objective functional without assumption of a pre-set terminal iteration.

Pioneering work by Athans [3] in attentive estimation addressed variation 3 in a continuous-time setting. This paper focuses on variation 4: optimization at each iteration

of the estimator with respect to an objective functional combining estimation performance and sensor costs. This is the most straightforward generalization of the problem examined in [1, 2]; previous results are subsumed by assuming uniform sensing in those presented here and a key property of earlier results (i.e., that "sensor scheduling" solutions, in which the entire optimal sequence of sensor selections is made before the onset of data collection, are possible if certain system parameters are known in advance) is preserved by this formulation.

2 Attentive Estimation with Sensor Cost

2.1 Classical Iterative Estimation

A classical discrete-time iterative estimation problem involves estimating the state x_k of a linear stochastic system

$$x_{k+1} = A_k x_k + \omega_k \tag{1}$$

in which A_k is a matrix and the ω_k are independent vectors of zero-mean gaussian noise with known covariance matrix Q_k. The estimate is to be based on noisy linear measurements of the state

$$y_k = H_k x_k + \nu_k \tag{2}$$

where H_k is a matrix and the ν_k are independent vectors of zero-mean gaussian noise which are independent of the ω_k and have known covariance matrices R_k.

The optimal estimate of x_n given y_0,...,y_n (in most commonly accepted senses) is $\hat{x}_n = \mathsf{E}[x_n | y_0, ..., y_n]$. This estimate is provided iteratively by the Kalman filter.

2.2 Formulation of the Attentive Estimation Problem with Cost

A related attentive estimation problem arises when the state of the system (1) is to be estimated using noisy measurements that are *selectable* from among a collection of linear observation maps; i.e., H_k in (2) is selectable from a collection $\mathcal{H}_k$ of observation maps representing a collection $\mathcal{V}_k$ of viable sensor configurations. The selection of a measurement map H_k entails a cost ξ_k which, in practice, may arise as a monetary cost or a risk associated with using the sensor or sensing mode that yields that particular measurement map.

The goal is to choose a sensing strategy $\{H_0, H_1, ...\}$ that minimizes the sum J_k of a cost measure C_k and an estimation performance measure E_k at each stage k. Since the system state x_k is to be estimated from measurements $y_0, ..., y_k$ by an unbiased estimator $\hat{x}_k$ at each stage k, the estimation performance measure may be taken to be a function of the estimation error covariance matrix $P_k \triangleq \mathsf{E}[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]$. Throughout the remainder of this paper, the estimation performance measure will be mean-squared error $\mathsf{E}[(x_k - \hat{x}_k)^T (x_k - \hat{x}_k)] = \mathrm{tr}\ P_k$, though the approach described is also applicable if another function of P_k is used in this role. The cost term is given by $C_k = c_{j_k}$ where j_k is the index of output map selected at stage k and c_{j_k} is its (pre-established) cost.

The Kalman filter propagates pre-measurement error covariance S_k and post-measurement error covariance P_k according to the equations

$$\begin{aligned} S_{k+1} &= A P_k A^T + Q_k \\ P_k &= (S_k^{-1} + H_k^T R_k^{-1} H_k)^{-1} \end{aligned} \tag{3}$$

Examination of these equations makes the solution to this problem evident: at each time step k, H_k should be selected to minimize $J_k = c_k + \mathrm{tr}\ P_k$.

Some previously published work on estimation with selectable sensors has emphasized the problem of calculating an observation map H_k, subject to constraints, that minimizes some given cost function of P_k [4, 5]. The form of the second equation in (3) suggests the formidability of such a calculation. Here, the collection $\mathcal{H}_k$ is assumed to be finite or parameterized in such a way to allow an optimal or nearly optimal H_k to be found by exhaustive search or perhaps some efficient search strategy on the

parameterized search space (e.g., a gradient or genetic algorithm). A pair of approaches that efficiently identify nearly optimal observation maps from a large collection have recently been proposed by Reeves [6].

It is important to note that the solution described here is an open-loop strategy. "Sensor scheduling" can be undertaken based on knowledge of the system parameters *before any data are actually collected.* Other work has shown that adaptive strategies for sensor selection are possible if certain system parameters are unknown, but that these strategies are typically closed-loop; i.e., the sensor to use in the next iteration cannot be determined until the current iteration is complete.

3 Examples of Attentive Sensing with Sensor Cost

To illustrate the role cost can play in selection of a sensing strategy, several examples are presented in this section. All of the examples are based around a stable discrete-time dynamical system with three-dimensional state space:

$$x(k+1) = \begin{bmatrix} .1 & 0 & 0 \\ 0 & .1 & 0 \\ 0 & 0 & .1 \end{bmatrix} x(k) + \omega(k)$$

The states are coupled through the zero-mean gaussian driving signal $\omega(k) = [\omega_1(k)\ \omega_2(k)\ \omega_3(k)]^T$ which has constant covariance matrix

$$Q = \begin{bmatrix} 1.8 & .8 & 0 \\ .8 & 1 & 0 \\ 0 & 0 & 1.8 \end{bmatrix}$$

and is independent from stage to stage.

3.1 Sensors with Equal Gains

Figure 1 shows results when three sensors of equal gain relative to the measurement noise variance $R = 1$ are available at each stage:

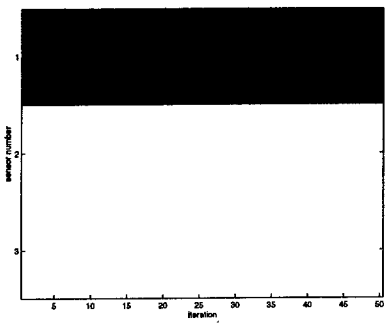
$$\begin{aligned} H_1 &= [1\ 0\ 0] \\ H_2 &= [0\ 1\ 0] \\ H_3 &= [0\ 0\ 1] \end{aligned}$$

Note that each sensor provides a noisy observation of exactly one of the three system states. Figure 1(a) shows the sensing strategy when sensor costs are ignored (i.e., $c_1 = c_2 = c_3 = 0$). Time increases along the horizontal axis and sensor number is indicated on the horizontal axis. At the horizontal position corresponding to each time increment k, the region corresponding to the sensor selected at that time is shaded black. The fact that H_1 is always chosen in this case is justified by the observation that sensor 1 provides the estimate of minimal mean-square error – the only criterion if costs are equal. For comparison, the aggregate mean-square error obtained by the attentive sensing strategy was 3.201 which was obtained (in this special case) by using only sensor 1; using only sensor 2 yielded a mean-square error of 3.768; using only sensor 3 provided mean-square error of 3.459. An aggregate mean-square error of 3.426 was obtained with a round-robin sensing strategy (i.e., using sensors 1, 2, and 3 in rotation).

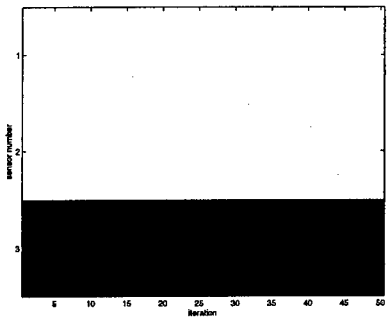
Figure 1(b) shows the sensing strategy obtained if the cost of sensor 1 is raised to 0.4 while the other two sensors' costs remain at zero. The best sensor, number 1, has become expensive enough that exclusive use of sensor 3 is favored even though the resulting mean-square error is higher (as noted above).

Figure 1(c) shows that a cost of 0.22 on sensor 1 with the other two sensors' costs remaining at zero results in a strategy that alternates between sensors 1 and 3. After each use of sensor 1, the estimation performance measure is sufficiently good that the cost-performance tradeoff favors the inferior but less expensive sensor 3 for the next measurement (i.e., the *value* of sensor 3 exceeds that of sensor 1). After sensor 3 is used, however, the estimation performance measure is degraded to the point that the additional performance of sensor 1 is worth its extra cost and it becomes the sensor of highest value for the next measurement.

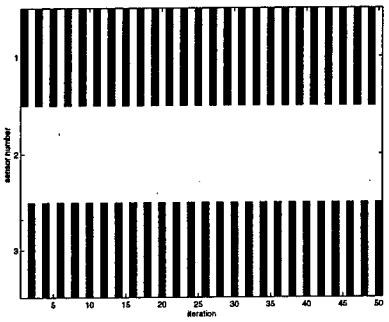
Figures 1(d) and 1(e) illustrate that the behavior observed in Figure 1(c) is preserved if both c_1 and c_2 are increased by approximately the same amount while holding c_2 constant



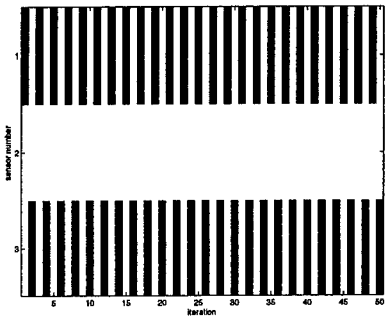
a. $c_1 = 0$
$c_2 = 0$
$c_3 = 0$



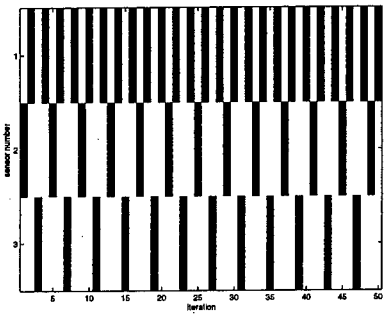
b. $c_1 = 0.4$
$c_2 = 0$
$c_3 = 0$



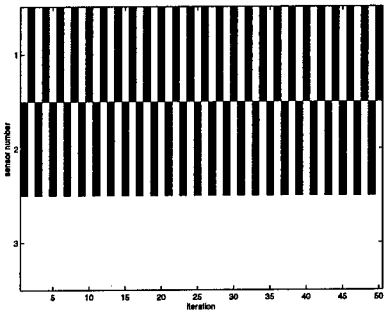
c. $c_1 = 0.22$
$c_2 = 0$
$c_3 = 0$



d. $c_1 = 0.4$
$c_2 = 0$
$c_3 = 0.18$



e. $c_1 = 0.57$
$c_2 = 0$
$c_3 = 0.3461$



f. $c_1 = 0.57$
$c_2 = 0$
$c_3 = 0.35$

Figure 1: Effect of cost on sensor selection: sensors with equal gains.

$$A = \begin{bmatrix} .1 & 0 & 0 \\ 0 & .1 & 0 \\ 0 & 0 & .1 \end{bmatrix} \qquad \begin{aligned} H_1 &= [1\,0\,0] \\ H_2 &= [0\,1\,0] \\ H_3 &= [0\,0\,1] \end{aligned} \qquad Q = \begin{bmatrix} 1.8 & .8 & 0 \\ .8 & 1 & 0 \\ 0 & 0 & 1.8 \end{bmatrix} \qquad R = 1$$

— but only up to a point. When both sensors 1 and 3 become sufficiently expensive, the attentive strategy begins to use sensor 2. The costs for which the values of the three sensors are sufficiently close to equal for all three to be used can be sensitive to small perturbations, as shown in Figure 1(f) where sensor 3 is dropped from the strategy of Figure 1(e) after only a small change in cost.

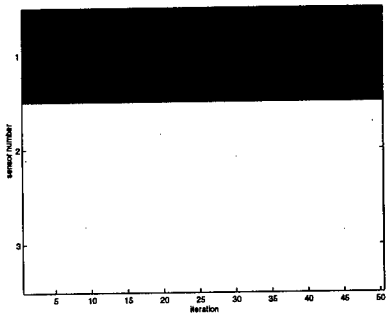
The aggregate mean-square estimation error obtained in the case depicted in Figure 1(f) is actually higher (3.556) than if a round-robin strategy were used; cost causes the strategy pictured to be favored.

3.2 Sensors with Unequal Gains

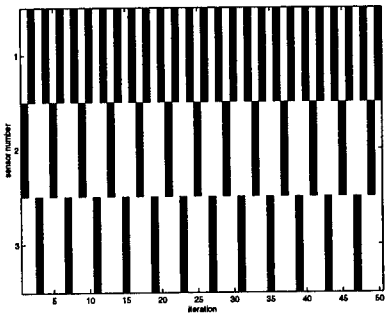
The results pictured in Figure 2 also come from a three-sensor scenario. In this case, however, the gains of the sensors relative to the measurement noise variance $R = 1$ are not all equal:

$$\begin{aligned} H_1 &= [2\ 0\ 0] \\ H_2 &= [1\ 0\ 0] \\ H_3 &= [0\ 0\ 2] \end{aligned}$$

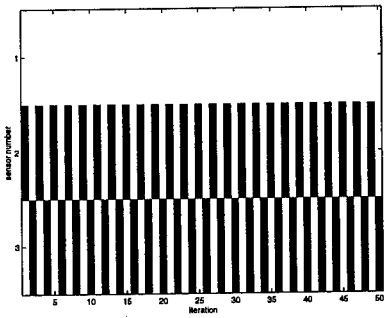
Again, each sensor provides a noisy observation of exactly one of the three system states. In the absence of cost considerations, sensor 2 is



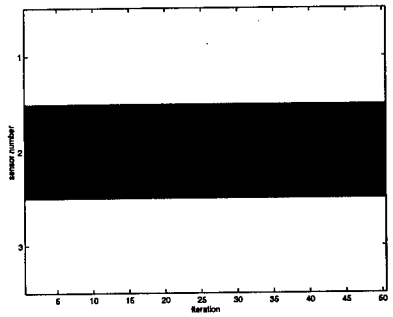
a. $c_1 = 0$
$c_2 = 0$
$c_3 = 0$



b. $c_1 = 0.5075$
$c_2 = 0$
$c_3 = 0.2103$



c. $c_1 = 0.6$
$c_2 = 0$
$c_3 = 0.2$



d. $c_1 = 0.6$
$c_2 = 0$
$c_3 = 0.21$

Figure 2: Effect of cost on sensor selection: sensors with different gains.

$$A = \begin{bmatrix} .1 & 0 & 0 \\ 0 & .1 & 0 \\ 0 & 0 & .1 \end{bmatrix} \qquad \begin{aligned} H_1 &= [200] \\ H_2 &= [100] \\ H_3 &= [002] \end{aligned} \qquad Q = \begin{bmatrix} 1.8 & .8 & 0 \\ .8 & 1 & 0 \\ 0 & 0 & 1.8 \end{bmatrix} \qquad R = 1$$

dominated by sensor 1: regardless of the state of the system, a measurement from sensor 1 will always yield a higher-quality estimate than a measurement from sensor 2.

Figure 2(a) shows that, with the costs of all sensors equal, a sensing strategy using only sensor 1 yields the lowest aggregate mean-square estimation error (2.630). A strategy involving all three sensors is obtained when the costs of sensors 1 and 3 are raised enough to make the use of sensor 3 cost effective. Figure 2(b) shows a case in which costs are chosen so that all three sensors have approximately equal value. In this example, the mean-square error is 2.909 — higher than when cost considerations are ignored, as expected.

Figures 2(c) and 2(d) illustrate that the situation in which all three sensors have approximately equal values is again sensitive to small perturbations in sensor costs. Modest variations in costs between the cases in Figures 2(b), 2(c), and 2(d) result in radical changes in sensing strategy.

4 Discussion and Conclusions

This paper has introduced considerations of sensor cost, which may arise as operating cost or risk, into the framework of attentive estimation for a discrete-time linear dynamical system. Examples illustrating the behavior of attentive sensing strategies as costs are adjusted were presented and discussed, and the observation that the sensor configuration of highest *value* is chosen by the attentive strategy was made. This suggests the possibility of theoretical analysis of the marginal tradeoffs between sensor gain and cost for a system in steady state to predict a sensor's value in particular scenarios — perhaps to identify libraries of sensors suitable for collections of estimation tasks prior to fielding of a sensor suite.

Future work should also consider costs incurred by switching between sensors.

References

[1] D. Sinno and D. Cochran. Dynamic Estimation with Selectable Linear Measurements. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 2193-2196, May 1998.

[2] D. Sinno and D. Cochran. Estimation with Configurable and Constrained Sensor Systems. *Signal Processing*, (to appear).

[3] M. Athans. On the determination of optimal costly measurement strategies for linear stochastic systems. *Automatica*, vol. 8, no. 4, pp. 397-412, July 1972.

[4] J.S. Baras and A. Bensoussan, Optimal sensor scheduling in nonlinear filtering of diffusion processes. *SIAM Journal of Control and Optimization*, vol. 27, no. 4, pp. 786-813, July 1989.

[5] Y. Oshman. Optimal sensor selection strategy for discrete-time state estimators. *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, no. 2, pp. 307-314, April 1994.

[6] S.J. Reeves and Z. Zhe. Sequential algorithms for observation selection. *IEEE Transactions on Signal Processing* (to appear).

Support Systems and Techniques for Submarine Sensor Fusion

Pailon Shar* **and X. Rong Li†**

Department of Electrical Engineering
University of New Orleans
New Orleans, LA 70148, USA
Phone: (504)280-7416, Fax: (504)280-3950, xli@uno.edu

Abstract-Some techniques and systems that support submarine sensor fusion are discussed. They include navigation system, other sensor systems like oceanographic sensor systems and own-ship performance monitoring sensor systems, own-ship steering system, sensor management, data flow arrangement, onboard data base system, and temporal and spatial information alignment. Their relationships with the fusion system, the coordination among these systems themselves and practical system design tips are also presented. Special emphases are put on system development considerations, especially on the unique requirements for submarines. The aforementioned systems are critical to the fusion system. They are also very complicated and may need fusion techniques in their own information processing, although they serve as supports to the fusion system.

Key Words: sensor, fusion, submarine.

1. Introduction

In fusion system development, much attention has been put on the basic fusion techniques, such as fusion structures and algorithms. In fact, also very important are support systems or techniques, such as coordinates selection and conversion, timing between different sensors and events, sensor management and data flow coordination. They serve as the basis for the fusion system. They are an inseparable part of the fusion system. Without a reasonable arrangement of these systems and techniques, it is impossible for a fusion system to work smoothly and effectively, and present most useful fusion results.

The supports available to different fusion systems are quite different. So is the management of these support resources. For submarine fusion systems, the support techniques are particularly important. The entire information environment of submarines has many disadvantages, such as poor information quality, mostly passive type of information, miscellaneous information patterns and an enormous amount of information [1]. In addition, similarly to other military systems there are many uncertain factors that may have an impact on the submarine information system such that the system may become very fragile and vulnerable to even minute errors. Under such conditions, a strong support from basic subsystems is vital. Not only can they help the system reach its performance climax, but enhance its robustness also. Therefore, especially in fusion system development, the importance of these support systems and techniques should never be overlooked or belittled.

2. Support Sensor Systems

There are three major categories of sensors that serve directly as information providers for command and control. They are own-ship information sensors, environmental information sensors and target information sensors. By sensor fusion many people mean fusion of target information sensors. This is the case in the literature where emphasis is mostly put on target information fusion. The fact is, however, own-ship and environmental information is also important. It is the basis of target information collecting and processing, not to mention other functions. As a matter of fact, the other two categories are also composed of many sophisticated modern sensors.

It should be emphasized that the three sensor groups are not completely separable. Some sensors are not confined in one group. Radar and the periscopes, for example, are important target search and detection sensors. Simultaneously they are also important navigation sensors. This demonstrates that navigation system and other sensors are not only conceptually important for the target sensor fusion system, they are also physically connected to the target sensor system.

2.1 Own-Ship Information Sensors

* Also named Peilun Xia, visiting scholar, on leave from Ocean University of Qingdao, P. R. China.

† Supported by ONR via Grant N00014-97-1-0570, NSF via Grant ECS-9734285, and LEQSF via Grant (1996-99)-RD-A-32.

The most important own-ship information sensor system is navigation system. It includes all the navigation sensors and the related processing devices. It provides own-ship positional and postural information which is a reference frame for the entire sensor system, including the sensor fusion system. This information includes longitude, latitude, depth, course, speed, dip angles, etc. of the own-ship.

The most important navigation sensors onboard a modern submarine are the GPS receiver and ship's inertial navigational system (SINS). GPS provides three-dimensional fixes with very high accuracy. The limitation is, however, that the mast has to be raised out of the water to obtain a fix. This is always a risky action although modern masks are usually coated with radar-absorbing material (RAM).

Inertial navigation system is very important for a modern submarine because it enables the submarine with long time submerged navigation ability. It consists of a three-gyroscope and three-accelerometer system that senses relative motion from a known starting point. Obviously the fixation error accumulates with time. Other navigation measures like GPS are needed to update the output periodically.

Of course there are many other more traditional navigation sensors such as magnetic compass, gyrocompass, radar, periscope and log. Clear enough, the navigation system is also a multisensor system that is not less complicated than the target information sensor system. So it is natural that its information be fused to achieve more concise and accurate results. The fusion of these navigation sensors cannot be expected to be easier. In fact, the own-ship information fusion is similar to the target information fusion. For example, the navigation sensors can also be divided into two groups, submerged sensors and surfaced sensors. The fusion system is accordingly divided into two parallel sectors: submerged fusion and surfaced fusion, exactly the same as for target information fusion. Some fusion techniques can also be shared.

The fused own-ship information finally should be input into target information fusion system, serving as a reference frame to target information. The navigation system here is treated as a support system to the fusion system, that is, as one of the necessary "supports" for the fusion system. In terms of e.g., military importance and system development, the navigation system and the target information sensor fusion system are equivalent - they complement and support each other.

Another important submarine sensor is the self-noise monitor sonar. It consists of several arrays located at different noise sensitive points on the submarine hull. Area around the propeller is one of such locations that need to be monitored because the propeller is the main noise source of the submarine. Self-noise level is one of the decisive factors for successful submarine operations. Propeller noise increases tremendously when cavitation occurs. Cavitation is a hazardous physical phenomenon that appears when the rate of rotation of the propeller is high enough, or equivalently, the speed of the submarine reaches a certain level. The main task of the array around the propeller is monitoring cavitation noise. Another location often monitored is around the bow sonar dome. The noise around this location is significantly harmful to the performance of the sonars with their arrays located in this area. Seemingly irrelevant to the fusion system, the monitor sonar provides early alarm for other sonars. In fact, the information provided by the monitor sonar is an important factor in underwater sensor management, a basic function of the fusion system.

2.2 Environmental Information Sensors

Environmental information sensors usually mean sensors that provide hydrographic, oceanographic and even meteorologic information. Some people argue that they also should be included in the navigation sensor category, which does not make much sense. Environmental information sensors are miscellaneous. They provide bathythermy, chemistry, magnetics, gravity, and acoustics information such as the temperature, salinity and seawater density, sound speed gradient, the basic structure and components of the sea bottom, and ambient noise. Such information is used to estimate the underwater sound speed gradient, acoustic convergence zone, propagation loss, and reverberation data. Other important information such as acoustic propagation paths, acoustic sensor range, etc, can also be estimated or predicted.

A significant difference between environmental information sensors and the other two sensor groups is that there is no strong need for real-time collection and processing of the information provided by the former group. It is not necessary to provide this information repeatedly in an engagement without a major environment and/or time change. Usually this information is measured as soon as the submarine reaches its designated position or battlefield. Environmental information is usually stored in onboard data base.

Environmental information is fundamental to sensor fusion as well as other command and control functions. Ocean environment analysis and sensor per-

formance prediction, which are important in sensor fusion, can only be achieved by using such information. Therefore, the environmental information sensors typically support the fusion system in an indirect way. Their measured information is used to calculate the basic parameters in cases such as underwater acoustics analysis and sonar performance prediction. These parameters are important fundamental elements for fusion functions such as determination of association gate size and sensor management.

3. Spatial Alignment

Although the sensors onboard a submarine are concentrated in a small space (Typhoon, the biggest submarine in this world, has a length of 171m and a beam of 24m), the work of putting information from these sensors into the same space reference frame can not be ignored.

First, since sensor transducer arrays are installed in different places onboard the submarine in a distributed fashion, the effect of this location distribution on the fusion results has to be examined. For example, the noise sonar array is mounted in the bow nose dome of the submarine. The information provided by it is most likely centered at the ship bow point. The passive ranging sonar arrays, however, are symmetrically arranged on both flanks of the ship. The reference center is usually the central point of the ship. Radars and periscopes are usually installed on the bridge somewhere between the bow and the center of the ship. Navigation sensors are also distributed. Information provided by these sensors has to be converted into a common space reference frame before fusion.

Secondly, different sensors may use different coordinate systems. Some sensors provide information under the Cartesian coordinates. Others use polar or spherical coordinates, absolute geographic coordinates, or relative coordinates. A unified coordinate system is needed when a fusion system is developed, and this is also important for spatial alignment of information.

Sometimes other factors have to be considered. For example, another coordinate system may be adopted when firing weapons. It will be much more convenient if the same system is adopted for both fusion and weapon firing purposes. If different coordinate systems are used, it is better that they can be converted easily. Processing algorithms (e.g., fusion or tracking algorithms) sometimes also have some special requirements for coordinates. A well-selected coordinate system should facilitate to satisfy these requirements.

Sometimes even units should be unified. It is common to use the navigation unit system for naval applications. In some cases, however, the international standard metric system is adopted by some sensors (e.g., some radars) and weapons (e.g., missiles). This difference has to be handled although it is a minor problem.

4. Temporal Alignment

The timing of different sensors is usually different. It is necessary to create a unified time reference for all sensors when fusion system is developed.

The frequencies being used are different. Sensors, such as passive sonars, are used for both surveillance and detection. They may be in operation all the time in the battlefield. Some other sensors, especially active or exposed sensors like radar, active sonar and periscopes, can be used only occasionally and under rigid restrictions. The exposed sensor is a name designated for submarine sensors like radar and periscope whose operation requires raising their masts out of the water, or sensors like active sonar and again radar whose operation requires sending out energy waves which can be detected. In both cases, the use of such sensors bears the risk of exposure the submarine to enemy.

The different physical field in which different sensors operate can lead to timing problems also. For example, radars and periscopes operate in light speed physical field while sonars operate in underwater acoustical speed physical field. If a target is detected at the same time instant by a sonar and a radar, the information provided by these two sensors obviously represents the target states at different time instants. While the information given by the radar may be deemed instantaneous, that is, at the time instant when the detection is made, the sonar only provides target information that is say, several minutes (or even longer) old because its takes considerable time for the acoustical wave to travel. To make things more complicated, the underwater acoustical wave path is often seriously distorted caused by the highly uneven distribution (sometimes even with sharp leap) of the transmission media. The time lag for the signals to travel from the target to the sonar array is difficult to estimate. For active sonars, the time lag is even larger due to the round trip of the acoustical pulse but can be easily determined.

The difference in the data rates of different sensors may also cause problems in timing as well as communication organization. Some modern digital sensors have very high data rates. They are usually used in environment with higher real-time requirements. Some other sensors like active sonars can not have a very high data rate. In cases where both of these sensors are involved, coordination and compromise are necessary.

There are many other special problems in sensor timing. The data flow is very complicated, especially during a real engagement. The timing of the sensors is virtually the timing of the data flow, a very difficult task. Miscellaneous requirements have been imposed on both the sender and the receiver of a signal. Some need the signal to be sent or received at particular time instants. Others have no such a requirement. Some require automatic sending or receiving of signals. Others do it upon request. Some may transmit data only when other data is available. Some need a strict synchronization. Others may transmit asynchronously.

5. Data Base

Like many other military systems, the submarine sensor fusion system deals with two groups of data or information. One is the fragile information that would become useless if not processed timely. Measurement data of a moving target belongs to this group. The other is the more robust data or information that can last for a relatively longer time. Characteristic data of a target is an example. This type of data should be stored and accessed when necessary. To manipulate and manage these data effectively, a powerful tool, such as a database management system, is needed.

Data base provides an important support for the fusion system [2]. For a submarine sensor fusion system especially, the information available is relatively poor and monotonous. For example, when the submarine is in its most probable submerged navigation state the information provided by sensors is simply acoustic measurements that are often seriously corrupted by noise and other factors. Performances of sensors onboard both the submarine and target ships should be evaluated by these data. The target can not be recognized without the help from these data. Algorithms may have to be initialized using these data. Artificial neural networks need to be trained by these data. Incidentally, collecting and processing such data is not an easy job. It is a painstaking and time-consuming effort.

The amount of data needs to be entered into a data base for fusion purposes varies, because several factors may affect it. The requirement of the user, the capability of the data base system available, the ability to handle data of this kind of the fusion system itself are some of the major factors. No matter what kind of data base is used, however, the following basic information is necessary.

1) Ocean Environmental Data

Ocean environmental data, from onboard sensors or from historical records, is important for the fusion system. A new trend these years is to put all the information on a marine chart into the data base. This is the so-called electronic marine chart. With its user-friendly interfaces, this new chart can provide rich nautical and oceanographic information in a very effective and flexible way. The ability of three-dimensional information generation and display of this chart system is especially useful in submarine applications.

2) Target Data

Facts about some possible targets are needed in many ways by the fusion system. Apart from basic data e.g., size, displacement, movability, and weapon capacity, information like acoustic features of its propeller noise and active sonar signals is also key to such missions as target recognition.

3) Decision-Making Data

This category includes information concerning human intelligence. Examples include fusion related tactical regulations, reasoning rules for artificial intelligence systems, artificial neural network training data, etc. Submarine tactics weighs heavily human and artificial intelligence in decision-making because of the usually disadvantageous information environment. Enough well-selected and well-organized decision-making data in the data base is a prerequisite for effective human and artificial intelligence systems.

6. Sensor Management

Proper management of sensor sources is quite a key to a successful battle engagement. The entire sensor system should be operated in an optimum synergic way. The basic concepts of sensor management are target assignment and target indication. Target assignment is the initiation of observation channels by assigning a specific sensor a specific target. Target indication is telling the assigned sensor where its target might be located, helping the sensor to catch the

target quickly. Indication may be given by the possible position of the target, the possible direction in which the target may move or the possible sector in which the target may stay.

Sensor management sometimes has to be concurrent with navigation and own-ship steering. Some sensors require special own-ship posture to ensure their specified performances. For example, avoiding detection blind zone of a sensor is the most basic requirement when the sensor is recommended to operate [3]. To manipulate the submarine to avoid the blind zone is the obligation of the navigation and steering systems. In addition, the time instant at which a specific sensor is used is another factor that should be considered. This is particularly true for the exposed sensors. To use them timely is critical for improving the observation and even the final results of an entire engagement.

In submarine tactics, the use of an exposed sensor is always seen as an action that needs precautions. For example, to operate a sonar in an active mode might mean to give up stealth and tactical advantages. However, in some cases it is necessary. In a submarine versus submarine engagement, for instance, it is extremely difficulty to detect a modern submarine by passive mode. Using an active sonar may expose yourself to your opponent though, you may win the critical time advantage. Sometimes it is very difficult to get an ideal fire control solution by using passive information, "ping" the target before firing weapons to get some active information to improve the solution can be a wise choice. In such cases that involve the use of the exposed sensors, target indication is especially important, because the time and the number of shots allowed for using these sensors is usually strictly limited.

Each sensor usually has its optimum frequency band. If a sensor is better at getting data on a particular signal, it is better to switch to this particular sensor. For example, targets at long distances can be detected more effectively by low frequency sonars. Small targets are better detected by high frequency sonars. Fast moving targets can be handled well by sonars with Dopplor abilities. At the same time, each sensor itself usually has several operation modes and frequency bands. Mode or frequency band recommendation sometimes is also a necessary task of sensor management.

It can be seen that sensor management in most cases is a decision-making problem. It is therefore almost impossible to handle it entirely automatically. Human interference is necessary and very important.

7. Steering System

One important task of sensor fusion is own-ship motion optimization. This is emphasized in [1, 3]. The goal of this optimization is to guide own-ship motion so that best sensor observations and/or best fusion results may be obtained. This can be achieved, however, only if the relationships between the fusion system and the steering system, among many other systems, are appropriately coordinated. There are always conflicting requirements for these systems. First of all, the recommended motion strategies by the fusion system should be realizable. One distinctive feature of a submarine is its poor movability. Strategies out of the reach of the steering system are absolutely unacceptable. To achieve such a balance is always a challenge. To make things even worse, there are many other constraints. For example, the remaining power storage of the battery arrays is another limitation for a diesel-electric submarine motion. The blind zones of sensors also impose limitations on submarine maneuver. Tactical requirements are another source of concerns. With all these factors being taken into account, the room left for the fusion system may be quite small.

Compromise strategies are necessary for the fusion system to handle these situations. For example, when a target contact is reported by a sensor, say, a noise sonar, it is usually a good strategy for the submarine to move at a low speed, because a low speed corresponds to a low level of self-noise. A low self-noise is a favorable condition for the sonar to keep the detection stable and effective and for the submarine to keep itself stealthy. It is also more power efficient, an attractive lure for a diesel submarine. Sometimes, however, some of these advantages have to yield to more urgent requirements. For example, low speed in some cases (e.g., bearings-only case) can result in much longer tracking time. Sometimes critical opportunities of tactical operations can be missed because of low speed.

Another point that should be emphasized is the fact that the procedure of carrying out a steering recommendation needs time. After a recommendation is made, the operator notices it and then reports to the commander, who then makes the decision to accept it or not. If the decision is yes, he gives the order. Personnel in charge of the operation of the steering system (e.g., planesman and helmsman) then carry out the order. Even after the human control operation, it still takes some time for the ship to finish its adjustment. At this time, the situation may have changed significantly since the recommendation was made. This implementation delay has to be taken into account when

making the recommendation. In fact, this is a problem for any machine-made decision. If this delay is large, its impact should be examined in such a decision-making process.

8. Conclusion

The support systems and techniques can be seen as an important integral part of the submarine sensor fusion system. Although known as support systems and techniques, they are actually complicated. So are their implementations. In fact, only some major issues are discussed in this paper. There are many other important aspects, such as system management, human-machine relationship and system performance evaluation that deserve attention. In fact, there are much more considerations when a real system development is to be undertaken. At the same time, submarines are experiencing modernization [4]. New technology and devices keep on pouring in [5,6]. Submarine sensor fusion, as well as its support systems and techniques, has to adjust itself to this development trend all the time.

References

1. P. Shar and X. Rong Li, "Some Considerations of Submarine Sensor Fusion," *Proc. of 1998 Int. Conf. on Information Fusion (FUSION'98)*, Vol. II, Las Vegas, Nevada. July 1998.
2. A. Jouan, L. Gagnon, E Shahbazian and P. Valin, "Fusion of Imagery Attributes with Non-Imaging Sensor Reports by Truncated Dempster-Shafer Evidential Reasoning," Ibid.
3. P. Shar and X. Rong Li, "Passive Sonar Fusion for Submarine C^2 Syatems," *Proc. of 1999 Int. Conf. on Information Fusion (FUSION'99)*, Silicon Valley, CA. July 1999.
4. J. Courter and L. Thompson, "Support the Navy's Submarine Modernization Plan," *Sea Power*, June 1995.
5. N. H. Guertin and R. W. Miller, "A-RCI—The Right Way to Submarine Superiority," *Naval Engineers Journal*, March 1998.
6. R. Hammett, M. Coakley, D. Sevigny and R. Zamojski, "Automatic Performance Monitoring Enhances Seawolf Submarine Ship Control Maintainability," *Naval Engineers Journal*, March 1998.

Session TA2
Medical Applications
Chair: Robert Levinson
University of California at Santa Cruz, USA

Discovering and Fusing Relevant Knowledge from Databases based on an Incremental Unsupervised Learning Approach

Francisco Azuaje[1], Werner Dubitzky[2], Norman Black[2], Kenny Adamson[2]

[1] *Northern Ireland Bio-engineering Centre, University of Ulster, Co. Antrim BT37 0QB, Northern Ireland, UK.*
[2] *School of Information and Software Engineering, University of Ulster, Co. Antrim BT37 0QB, Northern Ireland, UK.*

Abstract— *This paper discusses the fusion of structured and unstructured data. Information fusion methods based on a knowledge discovery model, and the case-based reasoning decision framework are implemented and evaluated. At the core of the knowledge discovery model is an unsupervised and incremental neural learning approach. Using signal data and database records from the heart disease risk estimation domain, three data fusion methods are discussed. Two of these methods combine information at the retrieval-outcome level, and one method merges data at the discovery-input level. The evaluation of such techniques demonstrates that the fusion of information at the retrieval-outcome level are significantly superior.*

Key words— Biomedical information fusion, case-based reasoning, knowledge discovery in databases, neural networks.

1. Information Fusion and Case Retrieval

The fusion of information is crucial in domains that consist of multiple variables and sources of data. With the proliferation of numerous sources of data, information fusion has become a fundamental research field inside and outside the computer science community. The term "data or information fusion" has been very recently established. The following definition has been adopted by *The European Association of Remote Sensing Laboratories, The European Space Agency and the Space Exploration Engineering Group* [1]*:*

"*Data fusion is a formal framework in which are expressed means and tools for the alliance of data originating from different sources. It aims at obtaining information of greater quality; the exact definition of 'greater quality' will depend upon the application*".

Thus information fusion is used to improve decision tasks — such as classification, estimation, and prediction — and to provide a better understanding of the phenomena under consideration.

Fusion may take place at the level of data acquisition, data pre-processing, data or knowledge representation, or at the decision-making level. Fig. 1.a illustrates a process of information fusion at the level of input data representation in a medical decision-making environment. At this level two main types of data may be effectively fused or integrated by the expert: structured and unstructured data. In this paper, the term structured data refers to standard *n*-tuple database records or attribute-value representation. It can represent medical tests or any other clinical reports. In this diagram the term unstructured data refers to text, signals, images, etc. Fig. 1.b depicts a process of information fusion at the level of decision-making. Here, the medical expert interacts with other human or/and computer-based experts that provide knowledge or hypotheses in order to support a final decision. This process can be understood as the fusion of multiple decisions. Thus Fig. 1 provides an illustration of the process of medical decision-making that can be approached as an information fusion problem.

Data Fusion based on *artificial intelligence* (AI) are becoming more and more established in areas ranging from image analysis through robotics to biomedical systems [2], [3]. The need for higher levels of reliability, emphasising at the same time clinical reasoning models, makes AI particularly attractive for those tasks that involve clinical data fusion. Assi [4] discusses some medical data fusion applications, where textual data from essays used in pharmacological practice and toxicology teaching is used, however, he focuses only on unstructured data.

One such AI technique is Case-based reasoning (CBR), [5] that views understanding and reasoning as a by-product of the underlying *memory processes* of memorising (data storage) and reminding (data retrieval). In CBR, the basic processes of solving a new problem or interpreting a new situation revolve around the *retrieval* of *relevant* cases from a case memory. This process is followed by the *adaptation* of the past to the new problem or situation. Arguably, the most crucial aspect in building effective CBR systems is the modelling of *relevance* knowledge. This knowledge is used in the retrieval stage to ensure that only those cases relevant to the current problem are retrieved. Usually, relevance, in such systems, is modelled via a *similarity measure* (computational approach) or an *indexing structure* (representational approach), or a combination of both [5].

Traditional CBR systems do not explicitly consider the dimension of fusing data which originated from different sources. Typically, a case in a CBR system is represented as a monolithic data record and the underlying retrieval and adaptation

schemes do not explicitly model the fusion of data. The limitations of this simplistic view become apparent in situations where the underlying case data is composed of different types of data; either; *structured* and or *unstructured*. In this study, the two main data sources have representatives in both categories, namely, digitised electrocardiogram signal data and medical database records (see Fig. 2). The basic decision task is that of estimating the coronary heart disease (CHD) risk of asymptomatic subjects.

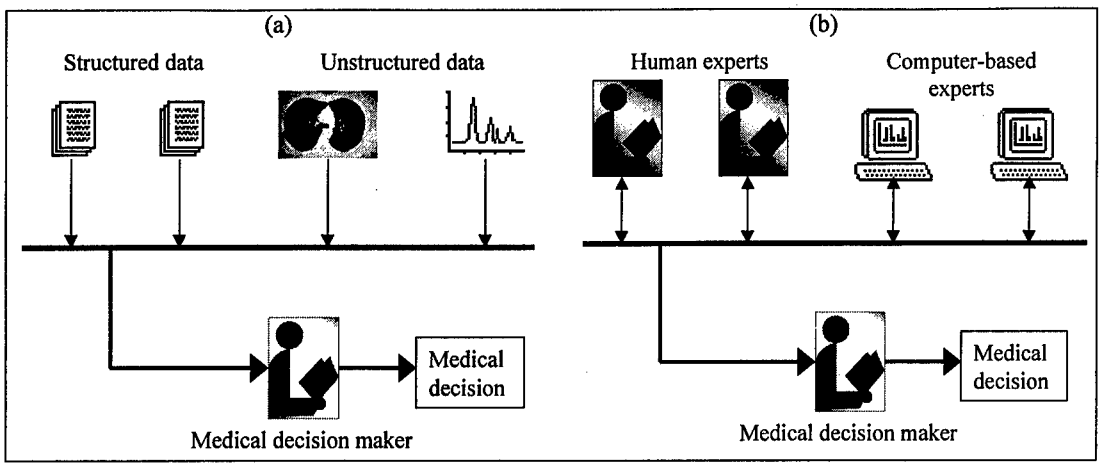


Fig. 1. (a) The process of medical reasoning seen as an information fusion problem at the level of input data representation. (b) Medical reasoning seen as an information fusion problem at the level of input data representation.

The work presented in this paper proposes a framework for information fusion based on the CBR paradigm. A knowledge discovery method permits relevance knowledge to be automatically extracted from existing structured and unstructured data. This method is based on a self-organising and incremental neural model called *growing cell structures* (GCS).

The remainder of the paper is organised as follows: Section 2 describes the medical problem under consideration. Section 3 introduces a relevance knowledge discovery model based on GCS. In Section 4, the information fusion methods are described in detail. Section 5 illustrates the implemented CHD risk estimation experiments based on the three fusion models, and compares the resulting overall systems with two single-source models. Finally, Section VI ends with some concluding remarks.

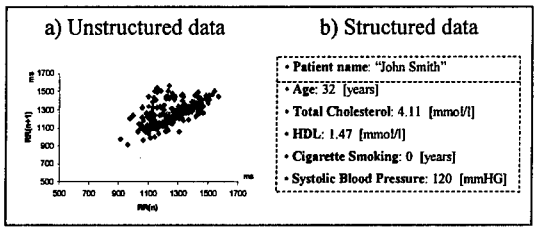


Fig. 2. a) RR interval sequence encoded as Poincaré plot; b) CHD risk factors encoded as feature vector.

2. The medical application domain

Coronary heart disease is a multi-factorial disease and it remains one of the most common causes of death in many countries [6]. A number of different approaches involving a variety of AI techniques such as chaos theory, neural networks, and based upon long-term ECG have been used to facilitate diagnosis of the condition [7], [8], [9].

The method proposed in this this paper models CHD risk estimation on the basis of input data that describes asymptomatic patients by means of short-term electrocardiograms (RR intervals) and recognised risk factors. A single RR interval reflects the length of the time period between the R-spikes of two subsequent heartbeats. In line with other research on RR intervals [10], a Poincaré plot encoding is used to represent a sequence of RR intervals. Plotting each RR interval in a sequence against the following, Poincaré plots provide an easy-to-understand visualisation of the heart's beat-to-beat behaviour. Fig. 2 illustrates the two types of data sources involved in the proposed risk estimation method.

The diagram in Fig. 2 depicts the RR interval and risk factor data of a healthy, low-risk subject. It is known that CHD risk is related low mean RR interval and low *heart rate variability*. Moving the cluster of points in such a plot from bottom-left to top-right corresponds to an increased heart rate, and a lower dispersion of the points in the cluster reflects an increased hear rate variability.

The data underlying this study was obtained from a set of standard screening tests that were performed on 75 asymptomatic, middle-aged, male subjects in order to identify subjects of high CHD risk. For each subject the CHD risk was determined by means of the Anderson scoring system [11]. This method calculates the risk score based on the following factors: Age, Sex, Total Cholesterol, High Density Lipoprotein Cholesterol, Systolic Blood

Pressure, Diastolic Blood Pressure, Smoking, Diabetes, Left Ventricular Hypertrophy. In addition to the risk factor data, the subjects also underwent a supine resting electrocardiogram at fixed respiratory frequencies. For each subject, four tests were carried out under varying respiratory conditions with regard to breathing volume and frequency encoded into a single RR interval sequence by using a Poincaré plot.

3. Relevance Knowledge Discovery Model

The notion of *relevance* is of primary concern in *information retrieval, case-based reasoning* systems, and for *multiple attribute decision making* methods [5]. Based on a query or problem description, relevance knowledge provides a means to retrieve those data items from a repository that are relevant to answering the query or to solving the problem. The explicit definition of useful similarity measures or indexing structures can be laborious and time-consuming and, as a result, less effective general measures, such as the Euclidean distance, are often used. Another less frequently encountered approach is to use machine learning methods or statistical models discover relevance knowledge from existing data. A relevance knowledge model of this sort, *indexing knowledge discover,* IKD, is at the centre of the data fusion methods proposed in this paper. The IKD model is based on the self-organising neural network approach; *growing cell structures* (GCS) [12].

The process of discovering similarity or indexing knowledge from a given set of cases, Ω, can be described in terms of partitioning the cases into a set, C, of disjoint groups or clusters such that members of the same cluster are more alike than members of different cluster [13]. A clustering algorithm, A, produces a mapping

$$A: \Omega \rightarrow C\,;\,(C \subseteq 2^{\Omega}) \wedge \bigcap_{X \in C} X = \varnothing,$$

which associates a cluster of similar cases, G, with every case, i, in the case base (where $i \in \Omega$ and $G \in C$).

In this work, a GCS neural network is employed to cluster cases. GCS neural networks constitute a variation of so-called *self-organising map* neural networks [14]. A typical GCS can be described as a two-dimensional space, where the units (cells) are organised in the form of triangles. The cells are represented as a weight vector, which is of the same dimension as the input data. The learning process in a GCS network consists of a number of input vectors or case presentations and weight vector adaptations.

In the first step of each learning cycle (i.e., presentation of a single case), the cell, c, with the smallest distance between its weight vector, w_c, and the actual input vector, x, is chosen. This cell is referred to as the winner cell. This selection process defined in equation (1) (O denotes the set of all cells in the network).

$$c: \|x - w_c\| \leq \|x - w_i\|;\ \forall i \in O \qquad (1)$$

The second step consists in the adaptation of the weight vectors of the winning cell, c, and their neighbour cells; these steps are defined by equation (2) and (3). The terms ε_c and ε_n are learning rates for the winner and its neighbours respectively; ε_c, $\varepsilon_n \in [0,1]$, and N_c denotes the set of direct neighbouring cells of c.

In the third step of the learning cycle, each cell is assigned a *signal counter*, τ, which counts how often a cell has been chosen as the winner during the learning process.

$$w_c(t+1) = w_c(t) + \varepsilon_c(x - w_c) \qquad (2)$$

$$w_n(t+1) = w_n(t) + \varepsilon_n(x - w_n);\ \forall n \in N_c \qquad (3)$$

Equations (4) and (5) specify how this counter is modified from one learning cycle at time t to the next at time $t + 1$ (the index c refers to the winning cell and i to all other cells). The parameter α is a constant rate of counter reduction, where $\alpha \in [0,1]$.

$$\tau_c(t+1) = \tau_c(t) + 1 \qquad (4)$$

$$\tau_i(t+1) = \tau_i(t) - \alpha\,\tau_i(t);\ i \neq c \qquad (5)$$

The GCS learning algorithm also performs an adaptation of the overall structure by *inserting* new cells in those regions that represent large portions of the input data. In this respect GCS neural networks differ from conventional and classic Kohonen-type neural networks. The insertion adaptation process is performed after a fixed number of learning cycles or *input presentations epochs*, λ. An input presentation epoch refers to a series of learning cycles within which the network is sequentially presented with each case, or input vector, in the training set. For example, if there are n = 100 training cases, and λ = 10, then a new cell will be inserted every 1000 learning cycles. Equations (6), (7), and (8) define the rules that govern the insertion process in a GCS network. the step of insertion of new cells. In the equations, the terms h_i and h_q reflect the *relative counter* of the corresponding cells i and q respectively.

$$h_i = \tau_i / \textstyle\sum_j \tau_j;\ \forall i, j \in O \qquad (6)$$

$$q: h_q \geq h_i;\ \forall i \in O \qquad (7)$$

$$r: \|w_r - w_q\| \geq \|w_p - w_q\|;\ \forall p \in N_q \qquad (8)$$

The cell with the highest relative counter, h_q, is symbolised by q. Now, the neighbouring cell, r, of q with the least similar weight vector is determined as defined by equation (8), and a new cell, s, is added between the cells q and r. The initial weight

vector of this new cell is equal to the mean of the two existing weight vectors. At the same time the signal counters, τ, in the neighbourhood, N_s, of the recently inserted cell, s, have to be adjusted. The new values of τ represent an approximation to a hypothetical situation where the cell s would have been existing since the beginning of the process.

After completion of an entire learning process, a number of ordered, discrete reference vectors are fitted to the distribution of the vectorial input patterns (cases). Thus, each case is assigned to the cell whose *weight* vector is closest to the case itself represented by the input vector. The resulting GCS network topology, with its cells, connections, and adapted weights, can be thought of as an partition structure for the underlying cases. Each cell in such a structure represents zero or more cognate cases that form a cluster or (extensional) concept. Generally, the similarity between cases from *direct* neighbour cells is higher than that of more distant cells. Based on weight vector differences of neighbouring cells, a quantification of

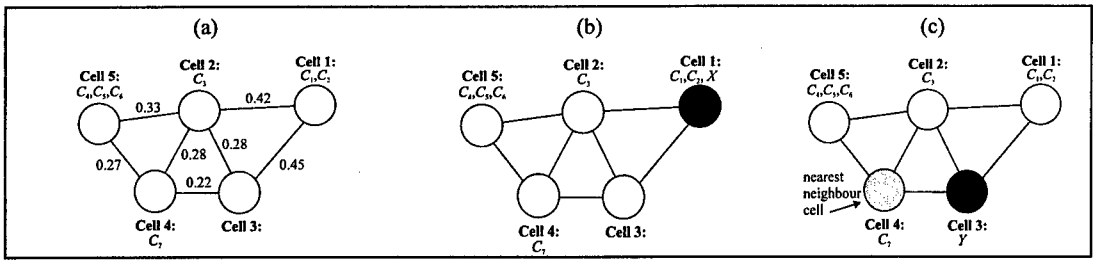


Fig. 3. (a) GCS indexing structure after learning; (b) retrieval of cases C_1 and C_2; (c) retrieval of case C_7

inter-cluster similarities is readily available. Hence, once training is completed, such a GCS network can be used to assign a new, previously unseen query case to its nearest "non-empty" cell. All cases in that cell are deemed most relevant or similar, and are retrieved for further processing.

To illustrate consider Fig.3. Fig. 3(a) depicts the GCS topology and inter-cluster distances (similarities) that have emerged after a *learning* episode based on a set of seven training cases. The resulting structure consists of five cells, each of which is associated with a subset of the training cases. For example, Cell 5 represents the cases C_4, C_5, and C_6, and Cell 3 is not associated with any case.

In the *test* mode (run-time), when the GCS is presented with a new query case, X, the GCS similarity indexing structure is used locate those cases in the case base that are most relevant or similar to X. This situation is depicted in Fig. 3(b). The new case, X, is assigned to Cell 1 based on the difference between X's value vector, $\mathbf{v}_X$, and the cell weight vectors, $\mathbf{w}_i$, using $DIS(\mathbf{v}_X,\mathbf{w}_i) = \| \mathbf{v}_X - \mathbf{w}_i \|$ (e.g., the Euclidean distance). All cases associated with the "best-match" call are retrieved, in this case the cases C_1 and C_2.

Fig. 3(c) illustrates the retrieval scenario where a query case, Y, is initially assigned to a cell (here Cell 3,) that does not represent any cases. In this situation, the algorithm selects the closest neighbouring cell (based on inter-cluster distances). If that cell is also "empty" the process will continue until a "non-empty" cell is reached. In the example, the process terminates at Cell 4, and the associated cases, in this case only a single case (C_7), are retrieved.

Many advanced adaptation techniques have been reported in the CBR literature [5]. For this study, the simple *null adaptation* was used. Null adaptation directly applies the past solution to the new case without modifying or transforming the past solution taking into account the differences between the retrieved and the new case. In this research the adaptation function for the CHD risk estimation task is defined in the following way:

Definition 1 Let X denote the new query case, and $r(X)$ the risk score that has to be estimated for X. Further, let the set, K, denote the set of most relevant cases retrieved for X, where $K = \{ C_1, C_2, \ldots, C_n \}$; $n \in \{1, 2, \ldots\}$. Then, on the basis of the previous risk scores $r(C_1)$, $r(C_2)$, ..., $r(C_n)$, the risk score $r(X)$ is estimated as follows:

$$r(X) = \frac{1}{n}\sum_{k=1}^{n} r(C_k); C_k \in K \qquad (9)$$

A diagrammatic illustration of the retrieval and adaptation model, based on the IKD method, is provided in Fig. 4(a). In the diagram, the discovered indexing structure is depicted by the bar labelled indexing, and the adaptation model, defined by equation (9), is portrayed by the bar labelled adaptation.

4. Information Fusion on Structured and Unstructured Data

The three fusion models presented in this section can be divided into two groups according the fusion level: *case representation*, and *retrieved cases* fusion. Both types of fusion rely on the IKD retrieval strategy and the solution adaptation model presented in the previous sections. Fig. 4(a) represents the basic single-source model in which a query is based on the underlying data source, and represented by the query case X. An

indexing structure is established by applying the IKD model, and an adapted solution is obtained in order to estimate the risk score $r(X)$.

4.1 The Case Representation Fusion Model

The *case-representation* fusion model combines RR interval data (source S_1) with risk factor records (source S_2) at the case representation level. This means that a single vector is created for each patient in the entire data from both data vectors (sequence of RR interval values and risk factor values). The bold-lined box labelled "fusion" in Fig. 4(b) illustrates this fusion model and the resulting transformed cases. Before risk scores can be estimated for new cases, the IKD model is applied to establish an *indexing structure* from the *transformed* cases in the case base. Also, a query is based on the fusion of the underlying data sources into the *transformed* query case, X.

Definition 2 Let the sequence $\mathbf{S}_1 = \langle f_1, \ldots, f_n \rangle$ and $\mathbf{S}_2 = \langle t_1, \ldots, t_m \rangle$ denote two distinct data sources that describe two properties of a single case. Then *case representation fusion*, $f_{CR}(\mathbf{S}_1,\mathbf{S}_2)$, of the data sources $\mathbf{S}_1$ and $\mathbf{S}_2$ is defined to be the sequence, $\mathbf{F}$, composed of the elements in $\mathbf{S}_1$ immediately followed by the elements in $\mathbf{S}_2$, as follows:

$$f_{CR}(\mathbf{S}_1,\mathbf{S}_2) = \mathbf{F} = \langle f_1, \ldots, f_n, t_1, \ldots, t_m \rangle \qquad (10)$$

In equation (2) $m, n \in N^+$, and the values f_i, and t_j are normalised, such that $f_i, t_j \in [-1,+1]$. The method can be is generalised to n sources. For the application described in this paper, source $\mathbf{S}_1$ relates to the risk factors and $\mathbf{S}_2$ represents RR interval information of the same patient; $n = 5$, and $m = 144$ (the 144 RR interval information values obtained from the Poincaré plots encoding [9]).

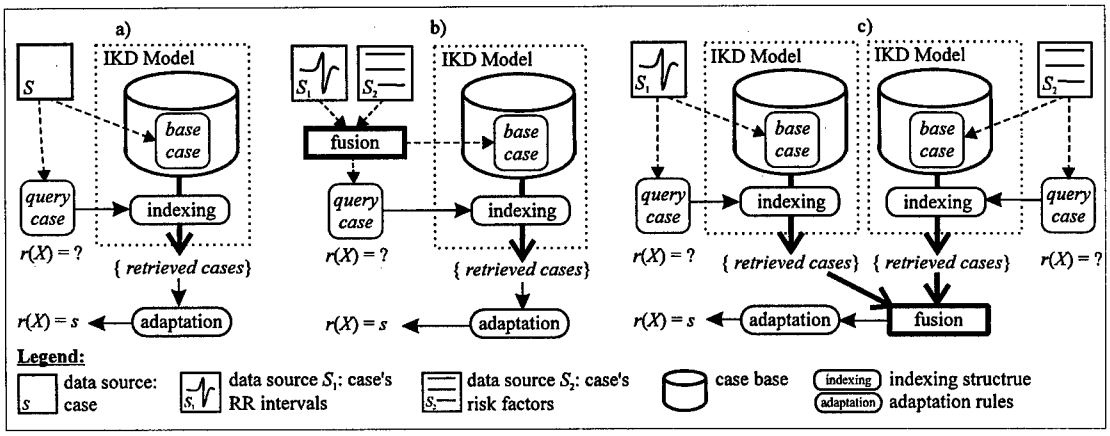


Fig. 4. The IKD case retrieval and adaptation model. a) basic, single-source model; and b) case-representation fusion; and c) retrieved retrieved-cases fusion. (rounded boxes = transformed data; dotted boxes = IKD model; bold style boxes = fusion rules; dashed arrows = data transformation; bold arrows = case data flow; thin-lined arrows = control; $r(X) = s$: risk score estimation, s, for query case X)

4.2 The Retrieved Cases Fusion Model

The *retrieved-cases* fusion model combines information based on the retrieval results of multiple single-source or *partial-case* case bases. The idea is that for each individual data source a separate case-base of *partial cases* is constructed using the IKD model. Fig. 4(c) depicts the architecture of such a system with two data sources and the corresponding partial-case case bases. A partial case reflects that part of the original case that is described by the corresponding data source. For a new query, the retrieval process is then carried out in each individual partial-case system. The result of each individual retrieval process is a set of one or more *partial* cases (depicted in the diagram by "{*retrieved cases*}"). Assuming a simple case identifier mechanism, each partial case can be linked to the overall outcome or solution (e.g., CHD risk score) of the original *complete* case. In general, the set of *complete* cases *identified* by the retrieved partial cases of one partial-case system is not identical to that of another. This raises the question: Given n sets of partial cases, which set of *complete* cases should form the basis for further processing (solution transformation)? In Fig. 4(c), this question is illustrated by the bold-lined boxes labelled "fusion", which takes as input the partial cases of the individual partial-case systems. Two fusion models are proposed to address this question, namely, *multiple-credit* and *single-credit* fusion. These fusion models are *general* fusion models, because their input is made up of the output of n individual case-based retrieval systems, which are *general* information processing engines.

4.2.1 Single-Credit Fusion

The single-credit fusion model merges complete cases referenced by the retrieved cases from the underlying partial-case systems, and it *removes* duplicate cases. This means that if the same complete case should appear more than once, the past solution that comes with this case will only be considered once.

Definition 3 Let $A = \{ P_1, \ldots, P_i, \ldots, P_n \}$ and $B = \{ Q_1, \ldots, Q_j, \ldots, Q_m \}$ denote the sets of retrieved partial cases originating from two independent partial-case systems corresponding to the data sources $\mathbf{S}_1$ and $\mathbf{S}_2$. Further, let $i(R)$ denote the complete case associated with the partial case $R \in A \cup B$. Then the fused *single-credit* set, $K_{SC}(\mathbf{S}_1,\mathbf{S}_2)$, of complete cases (solutions) processed by the adaptation module is defined as follows:

$$K_{SC}(\mathbf{S}_1,\mathbf{S}_2) = I(A) \cap I(B) \qquad (11)$$

such that $I(A) = \{ i(P_1) \ldots, i(P_i), \ldots, i(P_n) \}$, $I(B) = \{ i(Q_1), \ldots, i(Q_j), \ldots, i(Q_m) \}$, and $m, n \in N^+$.

4.2.2 Multiple-Credit Fusion

The multiple-credit fusion model also merges complete cases referenced by the retrieved cases from the underlying partial-case systems, but it does *not* remove duplicate cases. This means that if the same complete case appears more than once, the past solution that comes with this case will be included as many times as the it appears. Intuitively, this method gives increased attention or credit to those cases that are deemed relevant on the basis or multiple data sources.

Definition 4 Based on the formalism and notation in Definition 3, the multiple-credit fusion results in the *mutiple-credit* bag or multiset, $K_{MC}(\mathbf{S}_1,\mathbf{S}_2)$, which denotes complete cases (solutions) processed by the adaptation module; it is defined as follows:

$$K_{MC}(\mathbf{S}_1,\mathbf{S}_2) = I(A) \cup I(B) \qquad (12)$$

5. Results and Evaluation

All three fusion methods discussed in Section 4 have been implemented and tested using the data described in Section 2. In addition, two single-source reference experiments were carried out using only RR interval data and risk factor data respectively. The respective IKD-based fusion models discussed in Section 4 were then applied (risk estimation task) to the query cases in the test sets. The overall mean of the average absolute errors (15 query cases) for each of the five estimation models after 10 test runs are shown in Table 1.

Table 1. Two single-source, and three fusion-model results.

RF	RRI	CR	SCF	MCF
3.73	5.03	5.18	3.52	3.22

(RF: risk factor source only; RRI: RR interval source only; CR: case-representation fusion; SCF: single-credit fusion; MCF: multiple-credit fusion).

We observe that the single-credit and multiple-credit fusion models perform better than both single-source methods and the case-representation fusion model.

An analysis of means (ANOM) allows us to evaluate the significance of the difference of the proposed models [15]. In this evaluation method one computes decision lines defined as:

$$X_{..} \pm h(\alpha; I, N-I)\sqrt{MS_e}\sqrt{\frac{I-1}{nI}} \qquad (13)$$

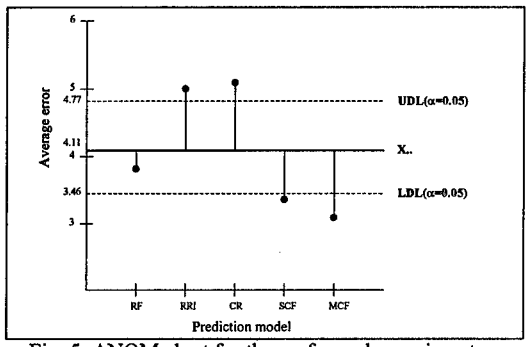


Fig. 5. ANOM chart for the performed experiments.

Where the critical values $h(\alpha; I, N-I)$ are available in [15], α is the probability level of significance, $X_{..}$ and MS_e represent the main absolute error and main variance respectively for all of the performed experiments, I is the number of models, n represents the number of absolute errors for each model and N is the total number of absolute errors under study. In this case, taking into account the averages absolute errors of the five models for each of the test runs, the ANOM is carried out graphically by computing decision lines at the $\alpha = 0.05$ level, $X_{..} = 4.11$, $MS_e = 0.75$, $I = 5$, $n = 10$ and $N = 50$. The ANOM chart (Fig. 5) compares the averages absolute errors for each model based on the lower (LDL) and upper (UDL) decision lines calculated by using Equation (13). From this chart one sees that the retrieved case fusion models fall outside the lower decision line LDL, while the single source models and the case representation model fall onto the other critical regions. Thus one finds that the means obtained from these experiments are significantly different at the $\alpha = 0.05$ level.

The statistical significance of the error differences comparing the best single source model

(RF) with the best fusion model (MCF) is given in Table 2.

The results are interpreted in the light of the following three significance methods: *Voting strategy*: The multiple-credit fusion method performed better than the single-data method in 8 out of 10 tests, and 3 of those results were statistically significant. *Total Combined SE (I)*: Taking into account the average outcomes from the 10 tests, a combined standard error (SE) equal to 2.53 was obtained. *Total Combined SE (II)*: Taking into account all the average errors of each of the 10 test runs (150 samples), a combined SE of 2.10 was obtained. This evaluation indicates that the total average error (3.22), obtained through the multiple-credit fusion method, was significantly superior to that obtained from the single-data model (3.73).

Table 2. Comparing single-data with multiple-credit fusion.

Test Run	RF	MCF	SE	Significance
1	3.80	3.74	0.10	N.S
2	3.25	2.86	0.71	N.S
3	3.43	2.84	0.58	N.S
4	3.09	3.19	0.13	N.S
5	3.32	3.24	0.09	N.S
6	3.81	4.08	0.4	N.S
7	3.69	2.51	1.98	$p<0.05$
8	5.09	3.56	1.86	$p<0.05$
9	2.71	2.60	0.19	N.S
10	5.08	3.58	1.76	$p<0.05$
avg.	3.73	3.22	—	—

N.S. = not significant; SE = combined standard error

6. Conclusions

This paper presented three data fusion models for case-base decision support and reasoning using CHD data. It was clearly demonstrated that at least two of the fusion models — single-credit and multiple-credit fusion — were superior to single-source models. Based on the best single-source model and the best fusion model, it was shown the superior performance of the fusion approach was statistical significant.

CBR is often characterised by five dimensions, namely, *representation*, *retrieval*, *adaptation*, *revision*, and *retention*. At a methodological level, the three fusion models put forth in this paper could be viewed as general models for some of these dimensions. Essentially, the *case-representation* fusion model constitutes a case representation framework that can consistently handle and integrate structured and unstructured data sources into a single unit. The two *case-retrieval* fusion approaches could be thought of as a multiple-case adaptation strategy.

The *indexing knowledge discovery* model proposed in this paper forms a crucial part in the overall fusion approach. Not only can this model handle data format diversity, high dimensionality, and relative importance of the data source, but it is also capable of incrementally updating the existing indexing structure when new cases are added to the system.

The case retrieval and fusion techniques outlined in this paper have demonstrated significant improvements of a medical decision support task. They can provide a better insight into the process of medical reasoning viewed as a multi-source and incremental data application domain.

Future work on this fusion model will have to consider intra-cluster, i.e., local, similarity processing, source selection procedures and fusion models. Another line of investigation would be to consider performance feedback within the learning stage, possibly in conjunction with genetic algorithms.

References

[1] Wald, L, 1998. "An European proposal for terms of reference in data fusion", *International Archives of Photogrammetry and Remote Sensing* vol. XXXII (7), pp. 651-654.

[2] D. McMichael, "Data fusion for vehicle-borne mine detection", in *Proc. 1st IEE Conf. on the Detection of Abandoned Land Mines*, Edinbourgh, UK, pp. 167-171, 1996.

[3] Pan H. and McMichael, "Information fusion, causal probabilistic network and Probanet", in *Proc. 1st Int'l Workshop on Image Analysis and Information Fusion*, Adelaide, Australia, pp. 445-458, 1997.

[4] A. Assi, "Data fusion for medical applications", in *Proc. of Fusion'98,* Las Vegas, USA, pp. 447-450, 1998.

[5] T.W. Liao, Z. Zhang, & C.R. Mount, "Similarity Measures for Retrieval in Case-Based Reasoning Systems", in *Applied Artificial Intelligence*, vol. 12, pp. 267-288, 1998.

[6] *Coronary heart disease: an epidemiological overview*. London: HMSO, 1994.

[7] M. Cohen, D. Hudson and P. Deedwania, "Combination of chaotic and neural network modeling for diagnosis of heart failure, in *Proc. of the Int'l Conf. on Computers and their Applications*, pp. 254-257, 1997.

[8] Z. Shen, M. Clarke, R. Jones and T. Alberti, "A neural network approach to the detection of coronary artery disease", in *Proc. of IEEE Computers in Cardiology*, vol. 20, pp. 221-224, 1993.

[9] P. Lopes, R. Mitchel and J. White, "The relationships between respiratory sinus arrhythmia and coronary heart disease risk factors in middle aged males", in *Automedica*, vol. 16, pp. 71-76, 1994.

[10] F. Azuaje, W. Dubitzky, P. Lopes, N. Black, K. Adamson, X. Wu, and J. White, "Predicting Coronary Disease risk Based on Short-Term RR Intervals Measurements: A Neural Network Approach", *Artificial Intelligence in Medicine*, 15, pp 275-297, 1999.

[11] K. Anderson, P. Wilson, P. Odell and W. Kannel, "An updated coronary risk profile: A statement for health professionals", in *Circulation*, vol. 83, pp. 356-361, 1991.

[12] B. Fritzke, "Growing cell structures-a self-organizing network for unsupervised and supervised learning", in *Neural Networks*, vol 7, pp. 1441-1460, 1994.

[13] B. Ripley, in *Pattern Recognition and Neural Networks*. Cambridge, England: Cambridge University Press, 1996, pp. 311-326.

[14] T. Kohonen, *Self-Organizing Maps*, Heidelberg, Germany: Springer, 1995.

[15] Wadsworth, H M, ed 1998. *Handbook of statistical methods for engineers and scientist*, New York: McGraw-Hill.

Three Dimensional Data Fusion for Biomedical Surface Reconstruction

J.M. Zachary and S. S. Iyengar
Department of Computer Science
Louisiana State University
Baton Rouge, LA 70803, U.S.A.
zachary@bit.csc.lsu.edu
iyengar@bit.csc.lsu.edu

Abstract - *Traditional surface reconstruction techniques have focused exclusively on contour sections in one anatomical direction. However, in certain medical situations, such as in presurgical planning and radiation treatment, medical scans are taken of the patient in three orthogonal directions to better localize pathologies. Fusion techniques must be used to register this data with respect to a surface fitting method. We explore the issues involved in fusing data from ellipsoid anatomy, such as the brain, heart, and major organs. The output of the fusion process is a set of data points that are correlated to one another to describe the surface of a single object. This data network is then used as input to a surface fitting algorithm which depends on two sampling metrics which we define. The solution to this problem is important in presurgical planning, radiation treatment, and telemedical systems.*

Key Words: data fusion, contour reconstruction, surface reconstruction, scattered data

1. Introduction

A common problem in many scientific fields is to reconstruct a three-dimensional surface from a set of planar contours. This type of data can be taken from many problems, including histological sampling of anatomy and computer aided design settings, but the most common field in which this technique is used is in clinical medicine. Data is obtained from patients by measuring serial sections of anatomy with medical imaging devices. The most well-known of these are computer axial tomography (CAT scans), magnetic resonance imaging (MRI), and ultrasound, which measure structural information in the object [1].

Most methods that reconstruct surfaces from contours only handle a set of contours along a single axis [2-4], but it is common practice to obtain medical scans from patients in three orthogonal directions. It is not possible to recover features of the surface of an object in areas where sampling is insufficient. The additional information in multiaxial contours is used to better localize anatomy; to "fill in the gaps" between contours in one direction and observe anatomy from a different perspective. Precise localization of target objects in clinical treatment is imperative in presurgical planning to minimize invasiveness and in radiation treatment to minimize exposure to surrounding tissue. Thus, our work is concerned with the data fusion issues in integrating the contours of three orthogonal axes into a coherent data set to which a surface may then be fit.

Many times, instead of segmenting sections from each scan slice, imaging techniques will instead consider the data as a sampling of a trivariate function on a cubical lattice and employ volumetric rendering techniques to create isosurfaces of the object, such as the marching cubes algorithm [5]. However, this approach assumes that the data are uniformly dense over the cubical region and thus can require the storage and transmission of a large amount of data.

We adopt the terminology of Meyers et.al. [4] and use the following definitions:

1. A *contour* is a simple polygon that results from the intersection of an object's surface with a plane.
2. A *section* is a collection of contours in the same plane. Note that, in general, a section is not necessarily composed of contours from the same object and an object may have more than one contour in a section.

The objects that we are interested in are nonbranching ellipsoidal objects, such as the brain, the heart, and other major organs.

1.1 Previous Work

Most attention to contour reconstruction has focused on fitting a smooth surface to a triangulation of a set of contours along a single axis. Meyers et. al. [4] have decomposed the problem into four subproblems: the *correspondence problem* which results from having multiple contours in a section and the solution of which determines the gross topology of the objects, the *tiling problem* which establishes a triangulation between adjacent contours (where adjacency is determined by the solution to the correspondence problem) based on some optimality metric, the *branching problem*

which handles the case where x contours are merged to y contours in adjacent sections and $x \neq y$, and the *surface fitting problem* which parametrically fits a smooth surface to the triangulation determined by the solution to the tiling and branching problem. Boissonnat [2] and Fuchs et.al. [3] are additional references to Meyers et.al. [4] for this case.

The aforementioned approach is inappropriate for the set of contour data from three different directions, since it makes the assumption that the data points all lie on the same closed manifold in three-dimensional space (i.e., is error intolerant) and it depends on the structured nature of parallel planar contours to perform the triangulation. However, a drawback of this assumption is that connectivity information is lost outside of the planes of intersection, hence the correspondence and branching problem mentioned above. Payne and Toga [6] proposed a possible solution based on a dense sampling of parallel planes to form a volume of data upon which resampling along any other intersecting plane may be performed. While this method allows arbitrary intersecting planes, it also leads to data inconsistencies since contours on separate planes of intersection are required to agree where the planes intersect. Also, this approach is computationally expensive. Finally, this method requires that the data is dense enough to construct a volume of data. However, one of our goals is to not require a dense sampling of patients. By minimizing the amount of data required to accurately reconstruct the object we also minimize the amount of possible radiation exposure to the patient and the time needed to transmit the contour data over low-bandwidth networks in telemedicine applications.

1.2 Motivation

According to Luo and Kay [7], multisensory fusion "refers to any stage in the integration process where there is an actual combination of different sources of sensory information into one representation format". For example, electrocardiogram sensor nodes are placed at various points on the surface of a patient to measure electrical cardiac activity and are combined into a single reading of spikes and valleys on a CRT display or paper tape. The definition of multisensor fusion is also applicable to the setting where data acquired from a single sensory device over an extended period of time is to be fused together into a single representation format [8]. The advantages of multiple axes contour data in surface reconstruction over the uniaxial approach can be summarized in terms of two of the advantages of multisensor fusion presented by Luo and Kay.

1. *Complementarity* The use of multiple sets of contour data taken from different directions collects more detail than contour data taken along a single axis. For example, features of the object in between two contours of a single set of contour sequences can never be reconstructed since there is no data of the feature. However, it is more likely that the multiple contour approach, particularly if the directions of sampling are mutually orthogonal, will sample data from these features and represent them in the final reconstruction. Another example is the possible disambiguation of the branching and correspondence problems mentioned previously that are a result of the structure of the data in uniaxial contour reconstruction.
2. *Redundancy* Errors in either collecting the sampled contours or in triangulating the sequence of contours in the uniaxial approach results in erroneous surfaces over a wide segment of the object. The redundancy provided by sampling the same object from different directions reduces uncertainty and contributes to a more accurate surface representation of the object.

1.3 Our Approach

The approach to surface reconstruction from contour information in this paper is motivated by the shortcomings of the uniaxial contour reconstruction methods. Additionally, we wish to use the standard set of image data collected from patients in the transaxial, saggital, and coronal orientations and not require special orientations not normally performed. Note that we do not require that data be collected by the same sensor arrangement. The only restriction is that the type of the sensor used, structural versus functional, be consistent. We also wish to minimize the amount of data needed to capture salient anatomy features, thus making our approach applicable in low bandwidth telemedical applications.

Our method consists of three computational components in which the output from each stage feeds into a subsequent stage (Figure 1). The first stage is to perform a segmentation of the object of interest in the medical images into sets of points that form contours. This stage is typically performed with trained human intervention. The second stage takes the three sets of contour data

sets and fuses them into a coherent data network which defines a single object. This network is then fed into the third stage in which a smooth surface should be fit. Our contribution and the focus of this paper is primarily with the second stage of data fusion (Section 2) and secondarily with the third stage of surface fitting (Section 3). We utilize an existing method of surface fitting to scattered data points by deriving two sampling metrics required by the algorithm.

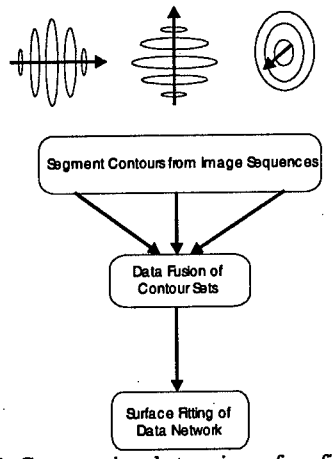


Figure 1. Computational steps in surface fitting process

2. Contour Fusion

The first component of the reconstruction process is a fusion of the sets of contour data from each axis to ensure that the axes along which the data are situated are orthogonal and the contour data points describe the same manifold. In this work, the assumption is made that the object from which the data is taken is ellipsoid in nature and, thus, has only one representative contour in each section taken. While this excludes certain types of anatomy, specifically anatomy with branching properties such as vascular networks, this assumption simplifies the method. Also, it should be noted that most anatomical features that are imaged in this manner are ellipsoid in nature.

The input to the data fusion stage is assumed to be three sets of contour sequences where each set is composed of contours that lie in perpendicular planes to some axis. The contours are sequences of points that have been segmented from the images of the medical scanning process, such as MRI or CAT scans. Usually, some human intervention is involved in the segmentation of points from interesting anatomical regions. In effect, the input can be thought of as three sets of points each of which lies on or near the surface of the object and the fusion problem is to correlate the points so that they define the same object.

The first step in the fusion process is to align the three contour axes such that each is orthogonal to the other two. Let the three contour sequences be $S_1 = \{c_{1,1}, c_{1,2}, \ldots, c_{1,n1}\}$, $S_2 = \{c_{2,1}, c_{2,2}, \ldots, c_{2,n2}\}$ and $S_3 = \{c_{3,1}, c_{3,2}, \ldots, c_{3,n3}\}$ where each contour $c_{i,j} = \{p_{i,j,k}\}$ for $i = 1,2,3$, $j = 1,2,\ldots,n_i$, and k is a subscript of the number of points in $c_{i,j}$. Since for a given S_i, the contours are assumed to be planar and parallel, one ordinate is fully specified by the imaging geometry and serves as the axis along which the contours are aligned parallel to one another. Additionally, the points are assumed to be given relative to the same global image coordinate system (I_x, I_y, I_z) for which the 2-D subspace (I_x, I_y) is illustrated in Figure 2. Given the imaging geometry for all three sets, the axes of the sets are assumed to be mutually orthogonal. The points that are a result of the segmentation process are in the coordinate system of the images and each series of images from the three differing directions are separate coordinate systems. In order to correlate the axes along which the contour sequences lie, we must transform them to a common coordinate system (Figure 2). For each S_i, the transformation $\tau : c_{i,j} \rightarrow c'_{i,j}$ takes the points of contour j from the image coordinate system to the coordinate system centered at the origin and bounded by the cube [-1...1, -1...1, -1...1] which may be considered the object coordinate system. The procedure for this step is

1. Determine the maximum and minimum ordinates of all points in S_i. Denote these as 3-vectors *max* and *min* on which indices *x,y,z* exist.
2. For all points (x, y, z) in S_i, make the following transformation:

$$x' = \frac{2}{\max_x - \min_x}(x - \min_x) - 1$$

$$y' = \frac{2}{\max_y - \min_y}(y - \min_y) - 1$$

$$z' = \frac{2}{\max_z - \min_z}(z - \min_z) - 1$$

This transformation is τ as given above.

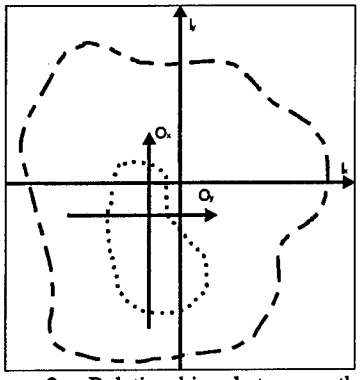


Figure 2. Relationship between the different coordinate sytems. The object has its own coordinate system denoted by O_x, O_y as well as being described in the coordinate system of the image I_x, I_y.

When this procedure is applied to S_1, S_2, and S_3, the transformed sets share the same coordinate system. In this way, the three axes along which the contours are aligned are correlated and fit to the cube [-1...1, -1...1, -1...1].

The second step of the fusion process is to adjust the contours for scaling. In practice, images taken from a patient using the same medical imaging device in a single session do not undergo scaling differences between different directions. The reason for this is that the scanning device, which is composed of an energy source and sensors on the other side of the body, is mounted on a circular platform and rotated concentrically during the scanning process [1]. However, scaling errors may occur in the segmentation phase when points are sampled from images.

If scaling is to be resolved, it can be assumed that it exists only between the sets S_1, S_2, and/or S_3. That is, there is no difference in scaling between contours of the same contour sequence. This assumption is reasonable when one considers that the contours that make up a particular sequence *Si* are sampled at the same sampling session. There is no combination of contours along an axis from different scanning sessions. If this were the case, the resulting data would be useless to the clinician since temporal and spatial differences could not be resolved.

Scaling of contours is either uniform or non-uniform. The general transformation from a set of points to a scaled set of points is given by the homogeneous matrix

$$\begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

When $S_x=S_y=S_z$, the scaling transformation is uniform. We determine and handle uniformly scaled contours by comparing the ratios of the width of the central contours in each of the three sets in a pairwise manner. If the ratio is greater than $1\pm\varepsilon$, we scale the offending contour set by using the ratio as the scaling factor. It is not sufficient to compare only two sets since it leaves the question of which contour set to scale undetermined. Currently, we do not consider the case of non-uniformly scaled contour sets.

The reader may notice that we have dealt with translation and scaling but have not discussed rotation. The reason for this omission is that it is assumed that the imaging geometry of the medical scanning device takes care of the element of rotation. In fact, in real world settings, the three directions that we have discussed, transaxial, saggital, and coronal, are mutually orthogonal directions that are "hardwired" into the scanning geometry of the hardware systems [1]. This condition may be violated when the patient moves while the device is in the process of scanning, but usually patient movement will induce other serious errors such as ghost images and blurring before there is enough movement to grossly violate the orthogonality.

The output of the fusion process is a set of data points that are correlated to one another to define a single object's surface. This set of points is then used as input to a surface fitting process.

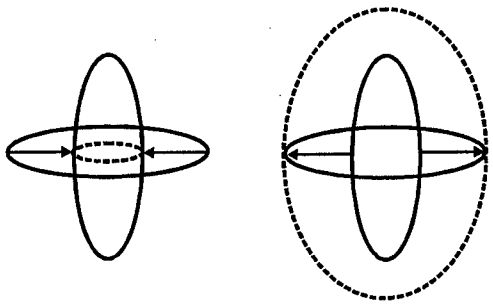


Figure 3. Scaling contours either by shrinking or growing.

3. Surface Determination

The output from the data fusion step is a network of points with a certain structure and the final task is to fit a smooth surface to this network of data points. We assume our surface to be compact, connected, orientable in $\Re^3$, and closed. This problem has been examined from many different perspectives in the computer graphics and vision community.

3.1 Surface Fitting to Scattered Data

Hoppe et. al. [9] present an algorithm to reconstruct the surface of an object from an unorganized collection of scattered data sampled from the surface. While our data network is not unorganized - it has the structure of three sets of points where each set is a sequence of planar contours and the three axes are orthogonal – we choose to use this algorithm because it produces good results when the sampling density of points is sufficient. A generalized approach is also preferable since it is applicable to the case of two sets of contour sequences instead of three. In fact, Hoppe et. al. [9] demonstrate their algorithm on the uniaxial reconstruction problem.

Prior to applying the scattered data reconstruction algorithm, two metrics describing the sampling error and sampling density must be defined. Consider the set $X = x_1, x_2, \ldots, x_n$ of sampled data points, i.e. $X = S_1 \cup S_2 \cup S_3$, on or near the unknown surface. We assume that $x_i = y_i + e_i$ where y_i is a point on the surface and $e_i \in \Re^3$ is an error term. Then, X is δ-noisy if $\|e_i\| \leq \delta, \forall i$. Features on the surface of the object that are smaller in magnitude than δ are not captured in the reconstruction. We estimate δ by recognizing that the most significant source of error associated with points not on the surface of the object is the segmentation by human intervention. Thus, the resolution of the image on the computer screen and the size in pixels of the image are important factors in estimating δ.

The other metric to be defined is the sampling density. As mentioned, features in those regions on the object's surface that have been insufficiently sampled cannot be reconstructed. Let $Y = y_1, y_2, \ldots, y_n$ be defined as above and S be a sphere of radius ρ. If $\|y_i - y_c\| < \delta$, $\forall y_i \in Y$ where y_c is any point on the surface of the object representing the center of S, then Y is said to be ρ-dense. We can provide an estimate of ρ by noting the structure in Figure 4(a). When combined and scaled, the sets of contours intersect to form a series of patches that approximates the surface. We assume that the inter-contour distance is constant along a given axis and that the distance between contour planes is greater than the distance between adjacent points on the same contour. Define d_i as the distance between contours $c_{i,j}$ and $c_{i,j+1}$ along axis i, and d_k as the distance between contours $c_{k,l}$ and $c_{k,l+1}$ along axis k. Define a sphere S centered between $c_{i,j}$, $c_{i,j+1}$, $c_{k,l}$, and $c_{k,l+1}$ with radius

$$\rho_{i,k} = \frac{\max(d_i, d_k)}{\sqrt{2}}$$

S is the sphere circumscribed around the square sharing the same center and with sides of length $\max(d_i, d_k)$ (Figure 4(b)) and it contains at least one point in the set $S_i \cup S_k$ when placed near some patch.

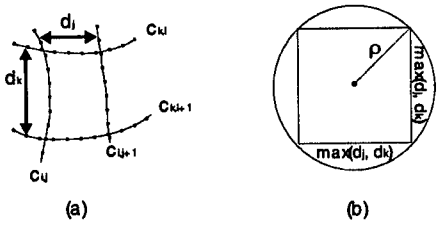


Figure 4. Configuration of contours in estimating the sampling density ρ

We outline the algorithm for surface reconstruction of scattered data below and refer the reader to Hoppe et. al. [9] for the details.

1. Define a scalar valued *signed distance function*

$$f : D \rightarrow \Re, D \subset \Re^3$$

which estimates the signed distance from a point to the unknown surface. Take the zero set of f as the estimate of the unknown surface.

A. *Estimate the tangent planes*

Let $N_k(x_i)$ be the k points of X closest to x_i, otherwise known as the *k-neighborhood* of x_i. A tangent plane which is the least-squares best fitting plane to $N_k(x_i)$ can be determined in the following way. Compute the centroid of $N_k(x_i)$ as

$$o_i = \frac{\sum_{j=1}^{k} x_j}{k}$$

and let r_i be the eigenvector associated with the smallest eigenvalue of the symmetric 3x3 semidefinite matrix

$$\sum_{y \in N_k(x_i)} (y - o_i) \otimes (y - o_i)$$

This eigenvector corresponds to the normal vector of the tangent plane, and thus, the tangent plane is given by (o_i, r_i). While (o_i, r_i) for each $x_i \in X$ forms a local linear approximation to the surface at each x_i, the set of all (o_i, r_i) cannot be used as the approximation for the surface since the resulting union may not be a manifold.

B. *Make the tangent plane orientations consistent*

Determining the tangent planes (o_i, r_i) is relatively straightforward; however, in order to be useful, the set of all (o_i, r_i) must be consistently oriented. Note that two points x_i and x_j are geometrically close if $x_i \in N_k(x_j)$ or

$x_j \in N_k(x_i)$, and their tangent planes are consistently oriented if $r_i \dot{r}_j \approx 1$. The problem of making the tangent planes for geometrically close points consistently oriented over all points in X can be cast as an NP-complete graph optimization problem, so Hoppe et. al. choose an approximation scheme. The basic idea is to choose an orientation arbitrarily (such as the x_i with the largest value of the z ordinate) and propagate the orientation that favors nearly parallel tangents planes by constructing the MST of the Riemannian graph.

C. *Construct f*

The signed distance function is created in the following way. Let $p \in \Re^3$ be an arbitrary point.

a $i \leftarrow$ index of (o_i, r_i) whose centroid is closest to p

b $z \leftarrow o_i - ((p-o_i)\dot{r}_i) r_i$ as the projection of p onto (o_i, r_i).

c if $d(z,X) < \rho + \delta$ then $f(p) \leftarrow (p-o_i)\dot{r}_i$ else $f(p)$ is undefined

2. Perform contouring of the zero set of f

The zero set of f is linear yet discontinuous. Thus, contouring methods such as the marching cubes algorithm [5] are used to extract an isosurface which is piecewise linear and continuous.

4. Concluding Remarks and Future Work

A method for integrating and correlating contour data sequences from three orthogonal imaging directions for surface reconstruction algorithms is developed in the context of data fusion by establishing a common coordinate system and adjusting for scaling discrepancies between the contour sets. Metrics for sampling density and error used in a scattered data reconstruction algorithm are derived for the fused data network.

4.1 Implementation

We have implemented our data fusion approach for three types of objects: spheres, ellipsoids, and a rough segmentation of medical data. The first two objects are artificially generated with a small perturbation from the idealized surface added at random intervals during the point generation process. Also, for one instance of the sphere, we uniformly scaled the contours along each of the three axes. Although we have not focused on nonuniform contour scaling in this paper, we generate data for spheres with a "bulge" in one set of contours. Each contour sequence in an object was generated from three distinct offsets to simulate segmentation of objects in the image coordinate systems. The ability to control the error in the data is an important factor at this stage of development.

The results of the contour alignment and uniform scaling fusion steps were favorable for the case of the sphere (Figure 5) and the ellipsoid. We have found that small errors are induced by the effects of non-uniformly scaled contours which are "warped" by the error, as predicted above. These errors in turn induce dimpled surfaces because of the segments of the contour which are non-uniformly warped outside of the surface. We are currently working on the registration of non-uniformly scaled contours. We are also gathering other sources of real digitized medical data as our attempts at digitizing MRI films produced digital images of low contrast which made subsequent segmentation by hand difficult, if at all possible.

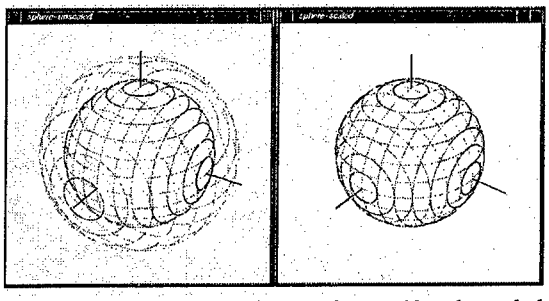


Figure 5. A sphere with a random uniformly scaled contour sequence (scaled by 1.34134398702) and the results of our uniform scaling fusion technique.

One area for exploration is designing surface reconstruction algorithms that build simplicial surfaces by exploiting the structure of the fused data network. There are two possible approaches to the triangulation of our data network. The first approach is to use the two extra sets to guide or correct the triangulation of a single contour sequence. This approach, as an extension of the methods described in [3,4], places constraints on the triangulation based on a geometric optimality criterion. Another approach is to extend the traditional methods of triangulation by considering all three sequences simultaneously. A different approach to the surface fitting stage that we are currently investigating is the use of deformable models as a 3-D surface representation. Further investigation to determine whether the additional information provided by the three sets of contour sequences is helpful to disambiguate the branching and correspondence

problems posed by Meyers et. al. [4] would be useful.

5. Acknowledgements

This work was supported in part by support from Oak Ridge National Laboratory. We gratefully acknowledge Dr. Deepak Awasthi of the LSU School of Medicine Neurosurgery Department for the MRI data sets used in this research. We also acknowledge Lei Huang for his help in preparing this manuscript.

References

[1] J.T. Bushberg, J.A. Seibert, E.M. Leidholdt Jr., and J.M. Boone, *The Essential Physics of Medical Imaging*, Williams and Wilkins, 1994.

[2] J.D. Boissonnat, "Shape reconstruction from planar cross-sections," *Computer Vision, Graphics, and Image Processing* 44(1), pp. 1-29, 1988.

[3] H. Fuchs, Z.M. Kedem, and S.P. Uselton, "Optimal surface reconstruction from planar contours," *Communications of the ACM* 20(10), pp. 693-702, 1977.

[4] D. Meyers, S. Skinner, and K. Sloan, "Surfaces from contours," *ACM Transactions on Graphics* 11(3), pp. 228-258, 1992.

[5] W.E. Lorensen and H.E. Cline, "Marching cubes: A high resolution 3d surface reconstruction algorithm," *Computer Graphics* 21(4), pp. 163-169, 1987.

[6] B.A. Payne and A.W. Toga, "Surface reconstruction by multiaxial triangulation," *IEEE Computer Graphics and Applications* 14(6), pp. 28-35, 1994.

[7] R.C. Luo and M.G. Kay, "Multisensor integration and fusion in intelligent systems," *IEEE Transactions on Systems, Man, and Cybernetics* 19(5), pp. 901-927, 1989.

[8] R.R. Brooks and S.S. Iyengar, *Multi-Sensor Fusion: Fundamentals and Applications with Software*, Prentice Hall, 1997.

[9] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," *Computer Graphics (SIGGRAPH '92 Proceedings)* 26(2), pp. 71-78, 1992.

Discovering Relevant Knowledge for Clustering through Incremental Growing Cell Structures

Xinyu W. Wu

School of Computing Science
Middlesex University
The Burroughs London
NW4 4BT
Telephone: 0181 3625000-2254
Fax: 0181 3626943
Email w.wu@mdx.ac.uk

Werner Dubitzky
***Francisco J Azuaje**
School of Information & Software Engineering
***Northern Ireland Bio-Engineering Centre**
University of Ulster at Jordanstown
BT 37 0QB
Telephone: 01232 368330
Fax: 01232 366068
Email w.dubitzky, fj.azuaje@ulst.ac.uk

Abstract - Artificial neural networks have been shown to be a useful computational model for a wide range of applications such as machine learning, pattern recognition, and pattern clustering. However, they received criticisms of being too rigid-structured models; their performance relies too heavily on a large number of free parameters; and most importantly, there were no explanations for their reasoning processes. They are considered by some not well suited for knowledge discovery tasks. In this paper, an attribute-pruning algorithm is presented and applied to a self-organised *growing cell structures* network in an attempt to discover knowledge that is most relevant for pattern clustering. Instead of using a predefined, fixed structure, the network topology is generated gradually during the incremental self-learning process and is determined entirely by the problem in hand. The results of this work demonstrate that by excluding irrelevant or less significant information, the network performance can be improved. More importantly, the extracted knowledge that is relevant to clustering can provide meaningful explanations for the clustering process and useful insight into the underlying domain.

1. Introduction

There have been many successful examples in the use of *artificial neural networks* for pattern clustering or other complex machine learning tasks. However, artificial neural networks suffer difficulties describing or explaining their behaviours. There is no simple mechanism, so far, that can be equipped to a neural network to help with the explanations of the knowledge learned in the network. Some hybrid approaches that integrate neural learning mechanisms and symbolic rule-based systems have been proposed to address this important issue [Healy & Caudell, 1997; Setionon & Liu, 1996; Sun & Bookman 1995; Tan, 1997] for the purpose of knowledge discovery [Weiss & Indurkhya, 1998]. The most popular hybrid approaches — the so called knowledge-based neural networks [Fu, 1993; Towell & Shavlik, 1993] — rely on some initial domain knowledge for network construction. In these models the learned knowledge, embedded in the large number of numeric connection weights of the trained network, is extracted by performing some complex rule extraction algorithms. The explanation of the reasoning process of such networks depends on the set of extracted, simple symbolic *if-then* rules. In case of weak domain knowledge in some real life applications, (e.g., the DNA promoter recognition problem [Barbara *et al.*, 1998] which has imperfect domain knowledge) a fully connected fat network would have to be constructed. This makes the rule extraction procedure even more difficult, and in some cases impossible [Wu, 1998].

Another problem that prevents artificial neural networks from being a main stream technique for problem-solving is that users (even some experienced users) often find it difficult to determine a network structure of suitable size and topology. That is, the number of hidden layers, hidden units, connection links between any two layers, and some other free learning parameters such as the learning rates.

Within the context of the discussed shortcomings of conventional neural networks, this paper proposes a promising approach to discovering and explaining knowledge relevant for pattern clustering. The approach taken is based on an incremental neural network model called *growing cell structures* (GCS). A key advantage of the proposed method is that it allows the shape as well as the size of the network to be determined during the simulation in an incremental fashion. Thus, the resultant

network has a structure that is intimately linked with the underlying problem-solving situation. For clustering tasks, the network is able to capture and represent the semantic similarity of the *n*-dimensional input patterns through the corresponding network topological structure (all input patterns are represented by *n* attributes). Moreover, this knowledge can be easily conveyed to and understood by humans via readily available visualisation techniques.

This paper proposes a powerful extension to GCS networks that makes it possible to provide *explanations* of the network learning process by using an *attribute pruning* algorithm. The algorithm is capable of determining those attributes in the underlying patterns whose contribution to the clustering task most significant. Identifying such attributes constitutes the key to explaining the network's clustering process [Agrawal *et. al.*, 1998; Mitra, *et. al.*, 1997]. Irrelevant or redundant attributes will be discarded. In other words, the initial high-dimensional input data is reduced by the algorithm to a pattern of lower dimensionality, and the knowledge most relevant to the clustering task is discovered. Additional advantages of attribute-pruning include better generalisation capability and lower cost of future data collection. By pruning attributes that are of little or no relevance for the clustering process, a better clustering accuracy on unseen patterns can often be achieved. Furthermore, a lower dimensionality of patterns means that only values of high-impact attributes need to be collected, and therefore, the cost of data collection can be reduced. As a consequence, the time required to cluster new patterns can also be reduced. By performing the proposed attribute-pruning algorithm on the GCS for pattern clustering, it is possible to display not only the semantic similarity of high-dimensional patterns, but also highlight the relevant knowledge for each cluster. Above all, the reasoning process in the GCS can be explained in terms of the most significant attributes for clustering.

The remainder of the paper is organised as follows: Section 2 explains the basics of the GCS — a growing and splitting artificial neural network. Section 3 presents the details of the attribute-pruning algorithm, and experimental results are reported in Section 4. Finally, conclusions are drawn in Section 5.

2. Growing Cell Structures

GCS neural networks [Fritzky, 1996] constitute an extension to Kohonen's *self-organising maps* [Kohonen, 1995], and are only one member in the family of self-organising, incremental models. Other family members include *growing neural gas* [Martunetz & Schulten, 1994], *growing grid* [Blackmore & Miikkulainen, 1993], and *dynamic cell structures* [Bruske & Sommer, 1995]. These models are not very different at all from an architectural point of view. Some properties shared by all models are described first in Section 2.1, followed by a concise description of the GCS which was used in the experiments carried out for this work.

2.1 Common Properties

Self-Organising networks have no predefined network topology, i.e., their structure emerges during learning. The structure of a network after learning is a graph consisting of a number of cells (also referred to as units or nodes) and edges connecting the cells. Each cell, c, "owns" a weight vector, w_c, which is of the same dimension as the input data vector. The basic learning procedure of a network is characterised by repeated input vector, x, presentations and weight vector adaptations. The purpose of the adaptation of the weight vector, w_c, is to reduce the distance between w_c together with its direct topological neighbours and the input vector, x.

At each adaptation step, local error information is accumulated, which is then used to determine where to insert new cells in the network after a fixed number of adaptation steps. When an insertion is done, the error is re-distributed locally. This increases the probability that the next insertion will be somewhere else. The local error will be reduced in a particular area of the input space by inserting new cells in exactly the same area. The local error variables can be thought of as a kind of memory which lasts over several insertion cycles and indicates where most errors have occurred and new cells are required.

2.2 Growing Cell Structures Networks

A typical GCS neural network can be described as a two-dimensional output matrix, where the cells are organised in the form of triangles. The network starts with only three connected cells each assigned with an *n*-dimensional weight vector with small random values. The first step of each learning cycle selects the cell, c, with the smallest distance between its weight vector, w_c, and the actual input vector, x. This cell is known as the winner (best-matching) cell for the current input pattern. The selection process is succinctly defined by using the Euclidean distance measure as indicated in expression (1) where O denotes the set of cells within the structure at a given point in time.

$$c : \|x - w_c\| \leq \|x - w_i\|; \forall i \in O \qquad (1)$$

The second step of the learning process consists of the adaptation of the weight vector, w_c, of the winning cell, and the weight vectors, w_n, of its directly connected neighbouring cells, N_c; see equations (2) and (3).

$$w_c(t+1) = w_c(t) + \varepsilon_c(x - w_c) \quad (2)$$

$$w_n(t+1) = w_n(t) + \varepsilon_n(x - w_n); \forall n \in N_c \quad (3)$$

The symbols ε_c and ε_n are the learning rates for the winner and its neighbours respectively, such that ε_c, $\varepsilon_n \in [0,1]$, and N_c represents the set of direct neighbour cells of the winning cell, c.

In the third step of the learning cycle, each cell is assigned a *signal counter*, τ, which represents the number of times a cell has been chosen as the winner. Equation (4) and (5) define how the signal counter is updated (symbol c still refers to the winning cell).

$$\tau_c(t+1) = \tau_c(t) + 1 \quad (4)$$

$$\tau_i(t+1) = \tau_i(t) - \alpha\,\tau_i(t); i \neq c \quad (5)$$

The parameter α in equation (5) reflects a constant rate of counter reduction for the rest of the cells at the current learning cycle, t.

Apart from weight vector adaptation, cell insertion is another important operation of the learning process for GCS. Pragmatically speaking, new cells are inserted into those regions of the output space that represent large portions of the input data to reduce local errors. Also, in some cases, a better modelling can be obtained by removing cells that do not contribute to the input data representation. Cell deletion may split the output space into several disconnected areas, each of which representing a set of highly similar input patterns. The adaptation process is performed after a fixed number of learning cycles (or epoches) of input presentations. Therefore, the overall structure of a GCS network is modified through the learning process by performing cell insertion and/or deletion. Equations (6), (7) and (8) define the rules that govern the insertion behaviour of the network.

$$h_i = \tau_i / \sum_j \tau_j; \forall i, j \in O \quad (6)$$

$$q : h_q \geq h_i; \forall i \in O \quad (7)$$

$$r : \|w_r - w_q\| \geq \|w_p - w_q\|; \forall p \in N_q \quad (8)$$

Insertion starts with selecting the cell, which served the most often as the winner, on the basis of the signal counter, τ. The cell, q, with the highest relative counter value, h, is selected. The neighbouring cell, r, of q with the most dissimilar weight vector is determined using expression (8). In the expression, N_q denotes the set of neighbouring cells of q. A new cell, s, is inserted between the cells q and r, and the initial weight vector, w_s, of this new cell is set to the mean of the two existing weight vectors, w_q and w_r, as follows: $w_s = (w_q + w_r) / 2$.

Finally, the signal counters, τ, in the neighbourhood, N_s, of the newly inserted cell, s, are adjusted. The new signal counter values represent an approximation to a hypothetical situation where s would have been existing since the beginning of the process. A simplified growing process of a GCS network is shown in Figure 1.

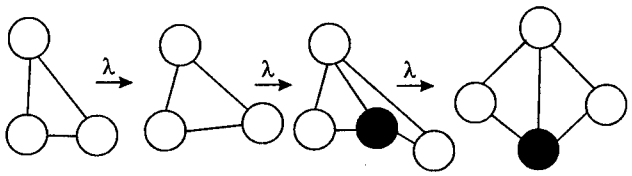


Figure 1. An example of the GCS network growing process.

The initial structure is a triangle of cells with randomly initialised weight vectors. The structure is reorganised (with or without insertion) after a constant number, λ, of input pattern presentations. When a new cell (black circle) is inserted, it is connected to other cells in the local neighbourhood in such a way that again a structure of triangles emerges.

3. Attribute-Pruning

A self-organising neural network based on the GCS approach described above has the advantage of being able to automatically construct a network, and to support easy visualisation of semantic similarity in high-dimensional data. There is, however, no explanation of the clustering process carried out by the network. This section presents an attribute-pruning algorithm, which is designed to exclude as many clustering-irrelevant attributes as possible, and to lower the dimensionality (complexity) of the data in each cluster. The most significant attributes can then be drawn upon for the explanation of the network's clustering process. The proposed pruning algorithm is inspired by some previous work on neural network pruning [Castellano *et al.*, 1997; Setiono & Liu, 1997], and especially Setiono and Liu's work on feedforward networks.

3.1 General Descriptions

Given a trained GCS network with each cell associated with an n-dimensional weight vector, which corresponds to the n input attributes, the purpose of the pruning mechanism proposed in this paper is to find the smallest subset of the attributes that still represents the characteristics of the patterns for clustering. After pruning, it is important that only those weights corresponding to the significant attributes have large magnitudes in the weight vector. To achieve this goal, a penalty function approach [Setiono, 1997] is adopted to take part in the modification of each weight at each processing cycle during the pruning operation. The penalty function that was used is defined in equation (9) below:

$$P(w) = \xi_1 \beta w^2 / (1+\beta w^2) + \xi_2 w^2 \qquad (9)$$

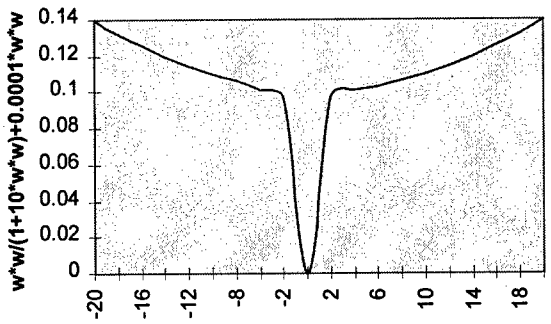


Figure 2. The penalty function with $w \in [-20,20]$.

Figure 2 illustrates the characteristics of the function with w in the range of –20 to 20, $\xi_1 = 0.1$, $\xi_2 = 10^{-4}$, and $\beta = 10$. The diagram shows that the penalty function encourages weights of small magnitude to converge to zero. Also, as reflected by the quadratic component, weights are prevented from taking too large values.

At the beginning of the pruning process, the values of the penalty parameters ξ_1 and ξ_2 are set equal for all weights, since it is not clear which attributes are the most significant and which are irrelevant ones. Each time the pruning process starts (with one weight, w_i, in each weight vector in a cluster set to zero), the clustering accuracy rate is computed on both the training and testing patterns respectively. A high accuracy rate suggests that the particular input attribute, a_i, which corresponds to the weight w_i does not contribute much information to this cluster, and may be removed from the input attribute set. The values of the penalty function parameters are then updated for all the remaining weights based on the accuracy rate of these networks. Larger penalty parameters are given to the networks with higher accuracy rate in order to keep the values of the weights smaller after the networks are retrained. It is therefore expected that the corresponding attributes are removed in the next iteration of the pruning algorithm. On the other hand, a lower accuracy rate indicates that the attribute, a_i, provides information important for clustering the underlying data and should therefore not be removed. In this case, small penalty parameters are assigned within the retraining process. This pruning operation is repeated for all weights in the n-dimensional weight vector until no more attributes can be removed. The detailed algorithm is outlined below.

3.2 The Attribute-Pruning Algorithm

A. Partition all the patterns into training and testing sets, T_1 and T_2 respectively.

B. Perform GCS-based clustering on T_1 to obtain the network topology (see Section 2). After training, each cell owns a weight vector, w_i, corresponding to the n-dimensional input attribute, a_i ($i = 1, 2, \ldots, N$).

C. Initialise penalty function parameters: $\xi_1 = 0.1$, $\xi_2 = 10^{-4}$; set lowest acceptable clustering accuracy rate: $\delta = 70\%$. (these settings were used in the experiments)

D. Use both T_1 and T_2 for attribute-pruning per cluster. Based on the clustering accuracy rates T_1 and T_2, the algorithm decides whether to continue or stop pruning.

- for $i = 1$ to N
  - Set all w_i except w_k ($k = 1, 2, \ldots, N$) equal to the trained weights; set $w_k = 0$
  - Thus, N networks are obtained each of which has one weight equal to zero in the weight vector w_{ij}; i and j count the numbers for the networks and weights in the weight vectors.
  - Compute the clustering accuracy rate R_i of each of the N networks on T_1 and T_2 respectively.
  - R_{avg} is the average of these rates. It is calculated only once in the first iteration of the algorithm and is then used as a constant for the rest of the pruning process.
- Rank the networks according to accuracy rates: $R_1 > R_2 > \ldots > R_N$. Each time, consider network N_i with the best accuracy rate R_{best} until the network with R_N is considered or the other recursive condition is met:
  - If $R_{best} < \delta$, terminate the pruning process.
  - Otherwise, retrain network N_i by updating the penalty parameters as follows:

    If $R_{best} \geq R_{avg}$,

    Let $\xi_{1(i)} = 1.1\,\xi_{1(i)}$, and $\xi_{2(i)} = 1.1\,\xi_{2(i)}$

If $R_{bes} < R_{avg}$,

Let $\xi_{1(i)} = \xi_{1(i)} / 1.1$, and $\xi_{2(i)} = \xi_{2(i)} / 1.1$

$w_{ij}(t+1) = w_{ij}(t) - P(w_{ij})$; for: $i \neq j$ and $P(w_{ij})$ is computed as shown in equation (9)

- Reset the input attribute a to $a - a_i$, and set $N = N - 1$, go back to D.

4. Experimental Results

Two real-world problems from the medical prognosis domain and an artificially generated animal taxonomy or clustering problem, borrowed from Ritter and Kohonen's [1989] early work on semantic maps, were used to test the pruning algorithm. The following two subsections report the results of the attribute-pruning experiments.

4.1 Clustering of Animals

The problem is concerned with clustering 16 animals. Each of the 16 animals is represented by a 29-dimensional vector consisting of 13 semantic or symbolic attributes and a 1-out-of-n coding of the animal's species name. The 13 semantic attributes are *size-small, size-medium, size-big, has-2-leg, has-4-leg, has-hair, has-hooves, has-mane, has-feathers, likes-run, likes-hunt,* and *likes-swim*. The GCS-based clustering process was performed during self-organisation of the network. The clustering results are shown in Figure 3.

The diagram in Figure 3 illustrates that four clusters were generated automatically. The GCS-based clustering provides a clear visualisation of the semantic similarity among different input patterns (e.g., horse, zebra, and cow are very similar). However, no straightforward explanations of the clustering process or results can be drawn from the resulting network topology. For example: What (which attributes) are the important characteristics that make cows and zebras similar? In an attempt to overcome this explanation problem, the attribute-pruning algorithm was applied to the data in a series of experiments. In the experiments, the entire data set of 160 patterns was randomly partitioned 10 times into a disjoint training set and a testing set, each training set containing 150 patterns, and each testing set containing 10 patterns. The clustering accuracy rates on training and testing sets before and after pruning were calculated. The results are summarised in Table 1.

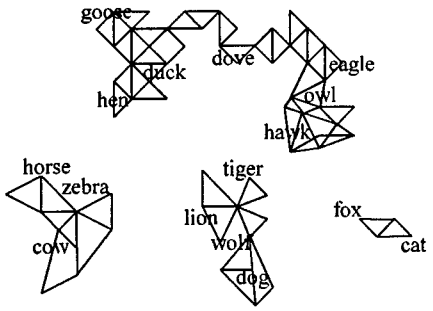


Figure 3. GCS based clustering results for animal attributes.

They show that clustering accuracy rates on testing sets are improved after the irrelevant attributes are pruned from the original attribute set. For example, consider Cluster 1 (Bird), when only the five most significant attributes (*size-small, 2-leg, has-feathers, like-fly, like-swim*) are used for the clustering task, the accuracy rate on the testing set is 94.5%. This is a significant improvement over the clustering accuracy before pruning (91.5%). Also, the clustering carried out by the GCS network may be explained by the first row in Table 1. For instance, consider Cluster 1. An explanation for the reasoning process can be stated as follows: *If an animal is small-sized, has two legs, feathers, and likes to fly or swim, then it is a bird.*

Table 1. Results of animal clustering; underlined attributes are pruned for some patterns in the cluster and kept for others.

Columns: Clusters and Clustering Attributes	**C 1: Bird** *small-size; 2-leg; has-feathers; like-fly; like-swim*	**C 2: Gentle Mammal** *size-big, has-hooves, has-mane, like-run*	**C 3: Hunter** *size-big, size-medium, has-hair, has-mane, like-hunt, like-run*	**C 4: Small Mammal** *size-small, has-hair, like-hunt*
Average Percentage of Relevant Attributes	17.2	13.8	20.7	10.3
CAR on Training Set *before* Pruning (%)	100	89.4	80.6	96.2
CAR on training Set *after* Pruning (%)	96.2	83.6	73.9	90.5
CAR on Testing Set *before* Pruning (%)	91.5	78.1	76.3	92.2
CAR on Testing Set *after* Pruning (%)	94.5	76.5	77.5	97.1

Legend: C n: Cluster n; CAR: clustering accuracy rate

4.2 CHD and CC Problems

The pruning algorithm was also tested on two medical data sets within a medical prognosis task context. The prognosis tasks were to predict coronary heart disease (CHD) *risk change*, and the *survival time* of colorectal cancer (CC) patients after surgery. The algorithm was applied to both data sets based on a GCS clustering process ($\varepsilon_c = 0.095$, $\varepsilon_n = 0.01$, $\alpha = 0.095$, $\lambda = 10$); Table 2 summarises the corresponding data. The same training and testing patterns were used for both problems. To test the pruning algorithm, the attribute numbers were expanded by taking each possible value of some of the attributes as a single new attribute.

For example, in the CC problem, attribute *pathological type*, which could take four values *tubular*, *mucinous*, *papillary*, and *signet* would be taken as four different attributes. The total numbers of attributes used in the pruning experiments for the CHD and CC problems are therefore 20 and 64 respectively. The pruning results are summarised in Table 3. Only three clusters are included in the results table for each of the two problems because of the large number clusters obtained from the GCS experiments. It should be noted that coherent results were obtained for most clusters.

Table 2. GCS clustering data for CHD and CC data.

	Attribute No.	Train. Sample	Testi. Sample	Cluster No.
CHD	5	71	12	12
CC	15	158	30	20

Table 3. Attribute-pruning results on CHD and CC problems.

Relevant Attributes for the Clusters	CHD-C1	CHD-C2	CHD-C3	CC-C1	CC-C2	CC-C3
Avg. Percentage of Relevant Attributes	25.0	40.0	55.0	39.1	21.7	23.9
CAR on Training Set before Pruning (%)	90.2	92.7	89.0	98.1	85.5	91.0
CAR on Training Set after Pruning (%)	83.1	80.6	75.9	91.8	78.3	80.9
CAR on Testing Set before Pruning (%)	84.5	87.1	73.9	92.0	75.3	79.4
CAR on Testing Set after Pruning (%)	86.5	89.5	76.5	95.1	79.0	84.8

Legend: CAR: clustering accuracy rate; C *n*: cluster number *n*

5. Conclusions

Using self-organising GCS networks to meaningfully cluster data has a number of appealing features over more conventional neural network models. For example, incremental self-construction, and easy visualisation of semantic relationships among the input data. However, a severe shortcoming of this model is that it cannot provide explanations of the clustering process. To address this problem, an attribute-pruning algorithm is proposed in this paper. It is designed to extract those attributes that are most relevant for pattern clustering. The most relevant knowledge for each cluster can be highlighted, and provide meaningful explanations about the clustering rocess and useful insight into the underlying problem and data.

The key idea of the pruning algorithm is to distinguish relevant and irrelevant attributes by determining how their corresponding weights in the weight vectors of the trained GCS network influence the network performance. Irrelevant attributes are identified by the small magnitude of their respective weights and are excluded from the original input attribute set. A penalty function approach serves as a basis to update all the weights in the weight vectors during retraining of the networks. The algorithm has been implemented and tested on two real-world medical data sets and one artificially generated data set. The experimental results show that with only a small subset of those relevant attributes used, the performance of the networks in terms of the clustering accuracy rates on unseen data can be improved. Although focus has primarily been on applying the attribute-pruning algorithm to self-organised GCS networks, the approach is naturally applicable to networks of arbitrary topology as pruning operates on both nodes and connections respectively.

6. References

[Agrawal *et.al.*, 1998] R. Agrawal, J. Gehrke, D. Gunopilos, & P. Raghavan: "Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications", Proceedings of the ACM SIGMOD International Conference on Management of Data, Seattle, Washington, June, 1998

[Barbara *et al.*, 1998] A. Barbara, J.S. Eckman, J.A. Borkowski, W.J. Bailey, K.O. Elliston, A.R. Williamson, & R.A. Blevins: "The Merck Gene Index Browser: an Extensible Data Integration System for Gene Finding, Gene Characterization and EST Data Mining", *Bioinformatics*, 14(1), pp.2-13, 1998

[Blackmore & Mikkulainen, 1993] J. Blackmore and R. Miikkulainen: "Incremental Grid Growing: Encoding High-Dimensional Structure into Two-Dimensional Feature Map", *Proceedings of IEEE Int. Conference on Neural Networks*, pp.450-455, 1993

[Bruske & Sommer, 1995] J. Bruske and G. Sommer: "Dynamic Cell Structure Learns Perfectly Topology Preserving Map", *Neural Computation*, 7(4), pp.845-865, 1995

[Castellano *et al.*, 1997] G. Castellano, A.M. Fanelli, & M. Pelillo: "An Iterative Pruning Algorithm for Feedforward Neural Networks", *IEEE Transactions on Neural Networks*, 8(3), pp.519-531, 1997.

[Fritzke, 1996] B. Fritzke: "Growing Self-organising Networks — Why", *European Symposium on Artificial Neural Networks*, pp.61-72, 1996.

[Fu, 1993] L. Fu: "Knowledge-Based Connectionism for Revising Domain Theories", *IEEE Transactions on System, Man, and Cybernetics*, 23(1), pp.173-182, 1993.

[Healy & Caudell, 1997] M.J. Healy and T.P. Caudell: "Acquiring Rule Sets as a Product of Learning in a Logical Neural Architecture", *IEEE Transactions on Neural Networks*, 8(3), pp.461-473, 1997.

[Kohonen, 1995] T. Kohonen: Self-Organising Maps, Springer-Verlag, 1995.

[Martunetz & Schulten, 1994] T.M. Martunetz and K.J. Schulten: "Topology Representing Networks", *Neural Networks*, 7(3), pp.507-522, 1994

[Mitra *et.al.*, 1997] S. Mitra, R.K. De, and S.K. Pal: "Knowledge-Based Fuzzy MLP for Classification and Rule Generation", IEEE Transactions on Neural Networks, 8(6), pp.1338-1350, 1997.

[Ritter & Kohonen, 1989] H.J. Ritter and T. Kohonen: "Self-organising Semantic Maps", *Biological Cybernetics*, 61, pp.241-254, 1989

[Setiono & Liu, 1996] R. Setiono and H. Liu: "Symbolic Representation of Neural Networks", *Computer*, 29(3), pp.71-77, 1996.

[Setiono & Liu, 1997] R. Setiono and H. Liu: "Neural-Network Feature Selector", *IEEE Transactions on Neural Networks*, 8(3), pp.654-661, 1997.

[Setiono, 1997] R. Setiono: "A Penalty-Function approach for Pruning Feedforward neural networks", *Neural Computation*, 9(1), pp.185-204, 1997.

[Sun & Bookman, 1995] R. Sun and L.A. Bookman (Eds): *Computational Architectures Integrating Neural and Symbolic Processes, A Perspective on the State of the Art*, Kluwer Academic Pub., 1995.

[Tan, 1997] A.H. Tan: "Cascade ARTMAP: Integrating Neural Computation and Symbolic Knowledge Processing", *IEEE Transactions on Neural Networks*, 8(2), pp.237-250, 1997

[Towell & Shavlik, 1993] G.G. Towell and J.W. Shavlik: "Extracting Refined Rules from Knowledge-based Neural Networks", *Machine Learning*, 13(1), pp.71-101, 1993.

[Weiss & Indurkhya, 1998] S.M. Weiss and N. Indurkhya: *Predictive Data Mining: A Practical Guide*, Morgan Kaufmann Pub., 1998.

[Wu, 1998] X. Wu: "A Hybrid Intelligent Framework for Explanation in Connectionist Networks", *Proc. of 31st Hawaii Int. Conf. on System Science*, pp.152-163, 1998.

Proved Segmentation From Pictures Sequence By Evidence Theory, Application IRM Pictures

[1]L. GAUTIER, [1]A. TALEB-AHMED, [2]M. ROMBAUT , [3]J-G. POSTAIRE, [4]H. LECLET

[1]LASL, Universite du Littoral Cote d'Opale, Calais FRANCE
[2]LM2S, Universite Technologique de Troyes, Troyes FRANCE
[3]I3D, Universite de Lille, Lille FRANCE
[4]Institut CALOT, Berck-sur-Mer FRANCE

Abstract *Our goal is to reconstruct the human rachis in order to observe the growth of some pathologies on patient suffering from scoliose. We present the first results on segmentation step. So we use an active contour method of segmentation. To minimize the uncertainty and the inaccuracy of the information, we use a data fusion method based on Dempster-Shafer theory. The originality of our contribution, consist to work with pictures sequence and not slice by slice. From three distincts sources, we search to detect the position of error on the slices. For each slice we use the information content in preceding and following slice. We define for each pair of slice's , a distribution of mass. The decision is taked from maximum credibility-plausibility criterion. We show endependently of the position of error that our method give to doctor a decision of good or bad classification of each part of the slices sequence.*

Keywords: Dempster-Shafer theory, picture sequence, proved segmentation, data fusion, IRM.

1 Introduction

The work presented in this article is keeping with the "Institut Calot de Berck sur Mer". It is done in order to help the doctors for spinal diseases. The studies of the pathologies are made from MRI images. The objective is reconstructing each vertebra of the lumbar spine from a serial parallel sections. From a initial segmentation, we're looking for parts which represents as better as possible the vertebra anatomical contour, in order to give to the doctors a belief degree on each part of this segmentation, and to show clearly the parts for which it is impossible to conclude. Generally, the slices present some imperfections, it is not always possible to define exactly the anatomic contour. We propose to use the adjacent sections, in order to get more information and to affirm or invalidate the taken decision.

The methodology is based on the belief theory using in order to fusion the information. This methode introducts a doubt notion between the differents elements.

2 The data

Several parallel views of spinal are used. On each of them, a spinal segmentation is realised with the snake method. We consider there is no junction. The objective is to make a segmentation of the vertebra of the lumbar spine. Each vertebra is delimited by a thin area with low signal intensity surrounded the vertebral body. We now study only one vertebra. For the reconstruction of the spinal, the same approach is repeated for each vertebra. The spinal is observed by a dozen of slices. The study bringing at the vertebral body segmentation which a parallelepiped form. The thickness of the IRM acquisition slice is small than the size of vertebra. For a done vertebra, the first and the last views are ignored because they are tangent to the vertebra extremities.

3 The frame of discernement

Each view is constitued by K elements which represent K different sets (or K organs) defined by Ω. In each view then is a lot of elements, and a lot of them are not separable to each other. This problem of diferentiation comes from the working principle of the MRI sensor.

$$\Omega = \{skin,\ vertebral\ body,\ cortex\ muscle,\ air,\ fat,\ fluid\} \quad (1)$$

The new Ω set is defined by the K=2 following elements :

$$\Omega = \{S, \overline{S}\} \tag{2}$$

with S is the substance familly which gives signal intensity (vertebral body, muscle, fat), and $\overline{S}$, the substance familly wich gives a low signal intensity (cortex, air, fluid).

Several methods of segmentation have been tested in order to extract from the picture the information of cortex. The segmentation by active contour has been choosen for the good detection results and its always closed contour.

The segmentation of cortex gives the contour toward the low signal intensity. By considering the thickness of the section and the knowledge of the vertebra anatomic body, we can deduce the following hypothesis : for two consecutives slices, importante variation of the spinal form leads to, at least, a mistake on one of them. The vertebral body is defined by a contour on all views. Each contour is sampling with the same number of points. The segmentation is defined to converge toward the good area.

$$\begin{aligned} Segmentation &= \{Q_i\} \\ &with\ i \in [1..N] \end{aligned} \tag{3}$$

$$Q_i \in \overline{S} \qquad with\ i \in [1..N] \tag{4}$$

The goal is to give an opinion on the Q_i elements in order to determinate if they are really a part of the $\overline{S}$ area.

The MRI introduces some artefacts during the acquisition of the data due to the partial volume effect and signal noise. So it's not possible to affirm that the obtained segmentation converges perfectly toward the good contour.

4 The expert mass sets

4.1 The expert

An expert gives an opinion on one or several elements of the frame of discernement. But, sometimes the expert can't differentiate several hypotheses and his opinion is distributed on the recovered familly. The belief theory is enable to introduce the doubt by passing from K elements frame discerning to the 2^K elements.

Then, the expert gives an opinion on the set of proposition of $2^\Omega = \{S, \overline{S}, \Omega\}$.

If the detected contour isn't entirely false, we call that we have high form variation when the distance between two parts of two consecutive slices become important compared to the mean distance separated all the points of the slices.

The expert or the original information, used here is based on the knowledge of two segmentations of successives slice, and particulary of the separated distance of two matched points.

We consider the slices two by two, in a same space. Each obtained contour is sampled with N points. Each points is matched at a point of the next slice. The matching is realised by the correlation two consecutive contours. We keep the combination of points which minimize the mean distance between P and Q.

4.2 Mass set

The mass set quantify all the expert opinions on the different elements of the discerning frame. When two points belong at two successives contours (P_1 and Q_1) and are closed, they are surelly a part of the same organ. But if this distance is more important, it would be a mistake of segmentation on one of the two contours.

We don't determine if the two points belong at S or $\overline{S}$, but these points have the same nature. We define a mass distribution for the expert opinion to a x point has the same nature, different nature, or uncertainty compared to its matched point. The expert opinion is modeled by the three following mass : $m_x\left(\left|\begin{matrix}S_p\\S_q\end{matrix}\right| \cup \left|\begin{matrix}\overline{S_p}\\\overline{S_q}\end{matrix}\right|\right)$, $m_x\left(\left|\begin{matrix}S_p\\\overline{S_q}\end{matrix}\right| \cup \left|\begin{matrix}\overline{S_p}\\S_q\end{matrix}\right|\right)$, $m_x(\Omega)$, with $x = Q_i$ a point from the sampled segmentation of cortex.

The single mass $m_x\left(\left|\begin{matrix}S_p\\S_q\end{matrix}\right| \cup \left|\begin{matrix}\overline{S_p}\\\overline{S_q}\end{matrix}\right|\right)$ and $m_x\left(\left|\begin{matrix}S_p\\\overline{S_q}\end{matrix}\right| \cup \left|\begin{matrix}\overline{S_p}\\S_q\end{matrix}\right|\right)$ give the belief degree of the expert to the x element has an identical (respectively different) nature compared to its matched point.

The composed mass $m_x(\Omega)$ gives the doubt that the expert has on the membership of x point. The proposed mass distributions are the following :

$$\begin{aligned} m_x\left(\left|\begin{matrix}S_p\\S_q\end{matrix}\right| \cup \left|\begin{matrix}\overline{S_p}\\\overline{S_q}\end{matrix}\right|\right) &= 1 - e^{-\eta.|d_{pq}-\alpha|} \\ &, d_{pq} \in [\varepsilon..\alpha] \\ &= 0 \quad , \text{otherwise} \end{aligned} \tag{5}$$

$$m_x(\Omega) = e^{-\eta.|d_{pq}-\alpha|} \quad , \forall d_{pq} \tag{6}$$

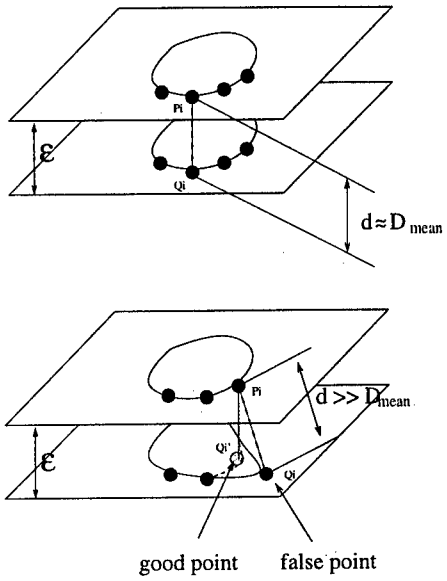


Figure 1: two slices

$$m_x\left(\left|\frac{S_p}{S_q}\right| \cup \left|\frac{\overline{S_p}}{S_q}\right|\right) = 1 - e^{-\eta.|d_{pq}-\alpha|} \quad , d_{pq} \in [\alpha..\infty[\quad = 0 \quad , \text{otherwise} \tag{7}$$

with $\eta = f(\varepsilon, \alpha)$.

Where α is the maximum expert doubt on the distance d_{pq}, α is function of D_{mean}. d_{pq} is the distance between two points, and D_{mean} is the mean distance between each point of two slices.

5 Combination of two sources in three consecutives slices

5.1 Distribution fusion

Each expert has his own frame of discernement, that must be extended to a common frame in order to process the fusion. We take three consecutive slices (P Q R) for which we calculate the mass set. For the couple (P Q) we define the m^1 mass. For the following couple (Q R) we affect m^2. The following slice (R) and the preceding (P) allow at the current slice (Q) to dispose relationship (with m^1 and m^2 mass) between (P_i, Q_i) and (Q_i, R_i).

The combinaion result shows some information on the continuity relationship between P_i Q_i R_i points. This gives a better information on the points belong to the anatomic contour. The Dempster rule is used to combine these distributions of mass.

$$m^{1,2}_{Q_i} = m^1_{Q_i} \oplus m^2_{Q_i} \tag{8}$$

We interpret the combination to extract many informations. We obtain the following relation on the error position :

- no error on slices P Q R,
- error on slice P and no error on slice Q R,
- error on slice Q and no error on slice P R,
- error on slice R and no error on slice P Q.

Futhermore we extract information of uncertainty between all these cases.

5.2 Decision

We find several choice to make the decision following the maximum plausibility, the maximum credibility, the interval credibily-plausibility, or the maximum evidence. We choose to use the interval credibily-plausibility to extract information to evaluate the point belonged because we can calculate them for each case. The credibility and the plausibility are defined :

$$Cr(A) = \sum_{A \subseteq B} m(B) \tag{9}$$

$$Pl(A) = 1 - Cr(\overline{A}) \tag{10}$$

The interval allows to exclude any cases where we can't differentiate the plausibilities or the credibilities.

6 Results

At the end of the fusion step, a segmentation solution is proposed to the doctor. This step gives an opinion for the P_i Q_i R_i points belonging at the S, $\overline{S}$, or Ω elements. The synoptic fig. 2 represents the synthesis data.

We tested our modelling on several examples of syntheses in order to verify its evolution. On the figure 3 we simulated no mistake on slices, the result is validated by the corresponding interval credibility-plausibility that is maximal. On the figure 4 (mistake on P), figure 5 (mistake on Q), et figure 6 (mistake on R), we simulated only one mistake on one of the three slice. The interval credibility-plausibility is maximal where the mistake is located. On the figure 7 we simulated a

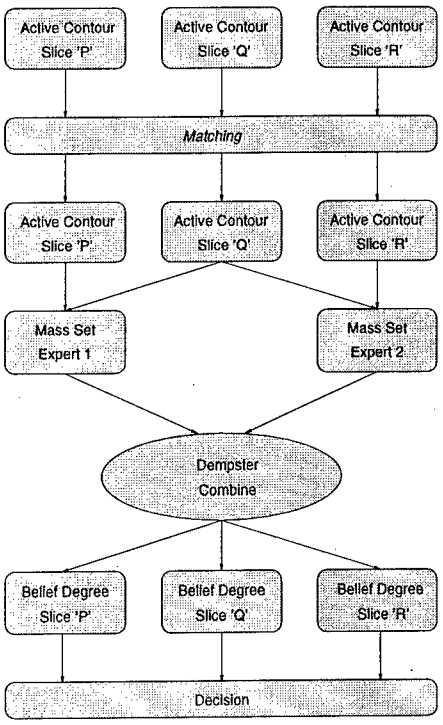


Figure 2: Synoptic fusion

mistake on the P and Q slices. The interval on R appears major because the two other slices are in the same way nature (identical mistake). Therefore the R slice appearing different of the two another, the mistake will be supposed on this last contour. That is compliant to the logic of our modelling. On the figure 8 we simulated an identical mistake on the three slices. That results in one maximal interval credibility-plausibility of not of mistake. They appear according to our modelling as being in the same way nature. This method allows to detect the abrupt variation of the spinal contour. The expert's criterion is defective on the example of the figure 9. There are errors on the two extremes slices and no error on the central slice. The fusion of the expert gives an error on the central slice Q. The expert criterion need to be completed to solve this case.

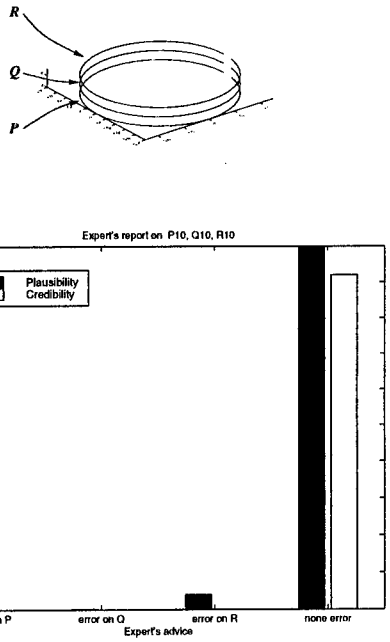


Figure 3: no error

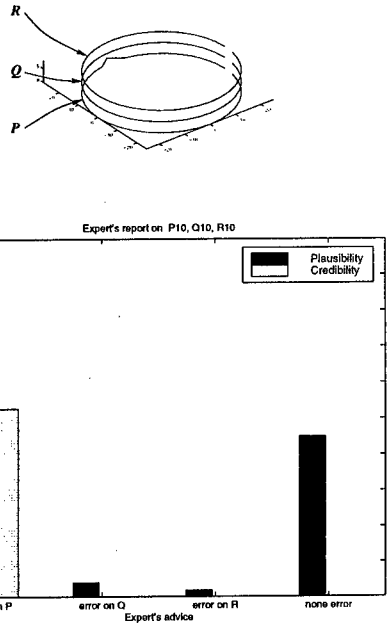


Figure 4: error on P

7 Conclusion

The results are interesting because they take out the correct decisions in most cases. Moreover the result of the decision is a belief degree. We have created a new expert, that from three consecutive slices, gives us an opinion on each of these. It let us consider in future works to use this expert in a new process of fusion. This expert doesn't allow to solve all cases, but combine with others,it will permit to correct invalide decision of segmentation.

We envisage in the future, to modify the modelization to solve this problem, the expert appraisal could have been completed by other informations as a low level fusion on the pixel intensity.

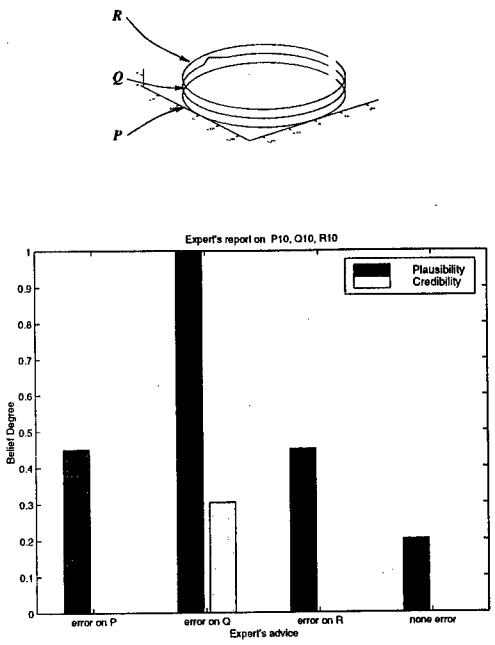


Figure 5: error on Q

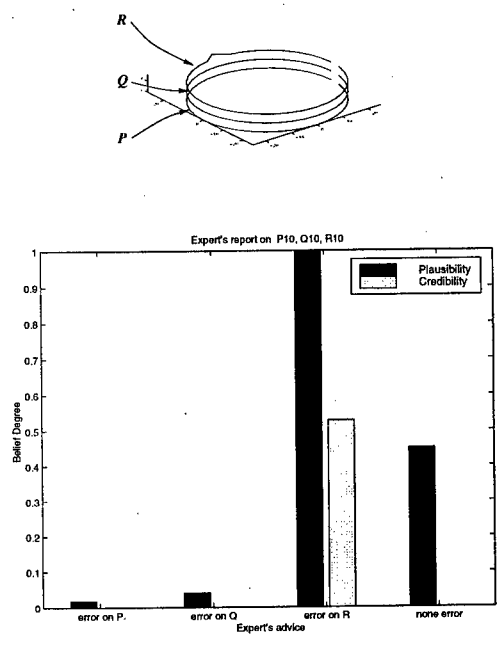
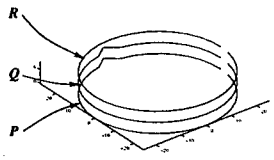


Figure 6: error on R



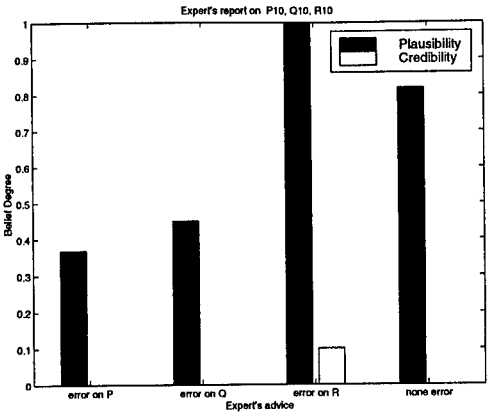
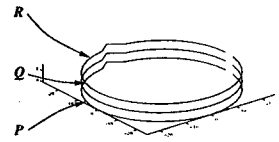


Figure 7: two errors



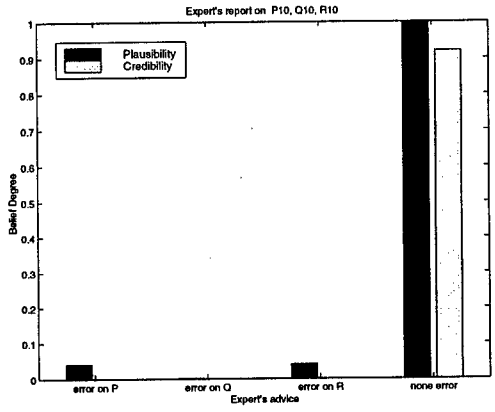


Figure 8: three errors

References

[1] A. Alaux *L'Image par Résonnance Magnétique.* Sauramps Médical, 1994.

[2] G. Shafer *A mathematical theory of evidence.* Princeton University Press, 1976.

[3] A. Dromigny-Badin *Fusion d'images par la théorie de l'évidence en vue d'applications médicales et industrielles.* PhD thesis, Institut Nationnal des Sciences Appliquées de Lyon, 1998.

[4] M. Rombaut and V. Cherfaoui. Decision mak-

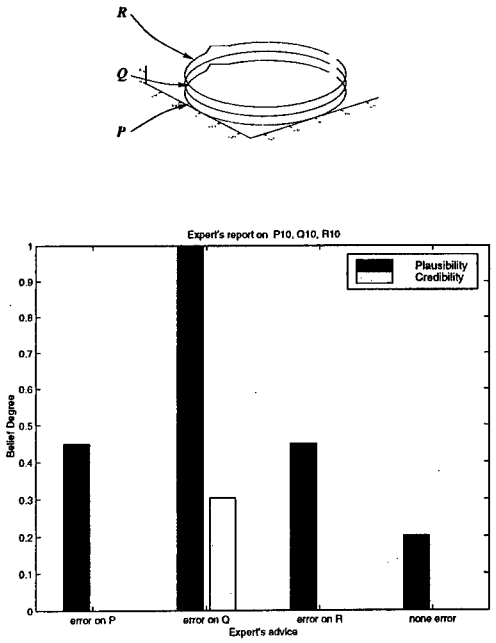


Figure 9: two extremes errors

ing in data fusion using Dempster-Shafer's theory. *3rd IFAC Symposium on Intelligent Components and Instruments for Control Applications*, 375–379, 1997.

[5] T. Denoeux. Decision making in data fusion using Dempster-Shafer's theory. *IEEE Transactions on systems, mans and cybernetics*, (25)804–813, 1995.

Aorta detection in ultrasound medical image sequences using Hough transform and data fusion

R. DEBON[1], B. SOLAIMAN[1], J-M. CAUVIN[2], L. PEYRONNY[3], C. ROUX[1]

[1]**École Nationale Supérieure des Télécommunications de Bretagne,** Dépt. ITI, Laboratoire LATIM, BP 832
29285 Brest, France

[2]**Département d'information médicale du CHRU de Brest,** LATIM, CHRU de Brest
29285 Brest, France

[3]**École Nationale Supérieure des Télécommunications de Bretagne,** Dépt ITI, BP 832
29285 Brest, France

{Debon.Renaud, Basel.Solaiman, Laurent.Peyronny, Christian.Roux} @enst-bretagne.fr
Jean-Michel.Cauvin@univ-brest.fr

Abstract- *Nowadays, information fusion constitutes a challenging research topic. Our study proposes to achieve the fusion of several knowledge sources, in order to detect the aorta artery, in ultrasound slices of the esophagus area. After a brief description of information fusion concepts, we propose a system architecture including both model and data fusion. Two primary models compose the algorithm: a fuzzy model, based on data fusion of three different information sources extracted from slices, and a Hough Transform (HT) model, which is often employed for pattern recognition. A global fusion model combines their complementary aspects and advantages. Along the sequence, spatial aorta matching is achieved by parameters propagation and controlled using a 3D-trajectory model. Simulation results, obtained from echo-endoscopic sequences, are presented.*

Key Words: *Hough Transform, knowledge sources fusion, echo-endoscopic image sequences, spatial matching.*

I. INTRODUCTION

Many engineering research domains use imaging processing architectures that often include fusion modules. In medical imaging and particularly in ultrasound imaging, data fusion is a must. Original data image usually contains insufficient information to develop robust segmentation algorithms, because of noise and distortion introduced by the acquisition system. Unfortunately, numerous medical imaging systems are not based on a multi physical sensor architecture that offers complementary information, to improve the efficiency of *a posteriori* numerical computation.

In this study, our particular interest is the detection of the aorta position and shape, estimated on a sequence of ultrasound transversal slices of the esophagus area. This detection is a module within a larger project intended to achieve a realist esophagus 3D reconstruction based on anatomical context information, in order to use the whole reconstruction as a diagnosis aid tool to evaluate digestive system pathologies [1].

As numerous anatomical human objects, an ellipse can first approximate the general shape of the esophagus wall. On ultrasound slices, the aorta has significant shape and position variations during all the sequence acquisition, as a consequence the condition of continuity is not always satisfied. Despite the small distance between two consecutive slices (1mm), strong shape and position variations can be produced by the patient breath activity, blood stream, natural anatomical orientation, and sensor displacements. Otherwise the contour is imprecise, noisy and is usually opened.

After the section II, which presents general fusion concepts, we propose in section III an aorta detection system based on data fusion, model fusion and on the Hough Transform (HT). In this section are first discussed the different knowledges which are able to complete the poor numerical information of ultrasound images. In a second time, method to combine their complementary aspects is precised. At the end of this part, model fusion to improve HT efficiency is proposed and ways to include knowledge at different level of HT are presented. In the section IV, results from a sequence acquired in real conditions of a medical exam are presented and commented. The conclusion evokes possible work perspectives.

II. INFORMATION FUSION

A definition of data fusion, given in [2] can be generalized to information fusion as follows: *'A multilevel, multifaceted process dealing with*

automatic detection, association, correlation, estimation, and combination of information from single and multiple sources'.

II.1 Information fusion concepts

Information fusion appeared when researchers have had the necessity to solve problem classes requiring to imitate the human intelligence. A possible classification of the fusion [3] introduces three conceptual levels corresponding to the three kinds of information:

-Data Fusion- is the first conceptual level. It usually consists in the merging of low level information, as primitives, in order to deduce a decision less noisy than with only one information source.

-Decision fusion- acts at the decision space level. Decision fusion achieves the combination of elaborated information as decision hypothesis, or results issue from a data fusion.

-Model fusion- is the case where information to be merged are strategies, processing methods or reasoning modes. A model fusion uses complementary aspects of two or more approaches in the case that just one isn't able to lead to the solution of a given problem. In [4], edge detection problem and model fusion are considered through the use of the Canny-Derich algorithm.

II.2 General fusion system architecture

This subsection intends to summarize the two major fusion system architectures. Due to some historical reasons, the first available scheme that we have when discussing information fusion systems, is that of a multi-sensor system. This scheme constitutes a partial view of the reality where several "physical" sensors are needed, in order to access several information issues of an object from the real world scene. In fact, two main architectures of information fusion systems can be distinguished:

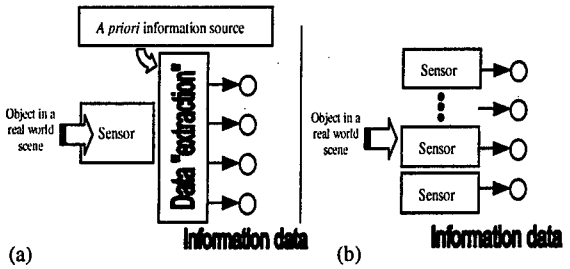


Figure 1: Information fusion systems architecture Mono sensor (a) and multi-sensor architectures (b).

The first, Figure 1.a, (referred to as the mono-sensor architecture) is based on the use of a single sensor and, the application of *a priori* knowledge, to obtain a new set of information data. The use of the probability set theory or the fuzzy set theory is generally performed through this step. The second system architecture (Figure 1.b) corresponds to the intuitive multi-sensor situation, where the "analyzed" object is observed through different physical sensors (or the same sensor, but with different geometric observation positions as is in the case of stereovision). The first system architecture has not been considered, for a long time, as being a real information fusion system. Anyhow, this is the main architecture used in several applications where the use of different sensors remains an obstacle and where an important amount of knowledge can be formulated as *a priori* knowledge sources of information. This is the case, for instance, in medical applications where the processing system can use a huge amount of *a priori* anatomical and expert-based sources of knowledge, to analyze medical images.

III. AORTA DETECTION

As previously mentioned, the aim of this study is to accomplish the aorta detection, using an ultrasound image sequence, acquired by an echo-endoscopic system. The sensor, called endoscope, is introduced through the mouth in the patient digestive system. Generally, a doctor assumes the sensor control but, in our particular case, endoscope progress through the esophagus lumen is entirely controlled by a mechanical system [1]. The obtained precision on z coordinate, which is about one millimeter, is enough to acquire all structures useful for a diagnosis elaboration (esophagus, aorta artery, and ganglions...).

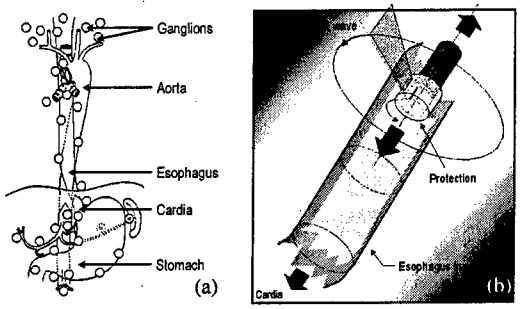


Figure 2: (a) Position of aorta in anatomical general scheme: aorta artery is always in contact with the esophagus. (b) Echo-endocopic imaging system: endoscope progresses along the esophagus lumen and, thanks to ultrasound waves, an esophagus area image can be computed.

An echo-endoscopic image is shown in Figure 3. Detailed analysis shows that the image quality depends mainly on two phenomena: speckle noise (due to the ultrasound imaging acquisition approach) and a concentric wave reflections network created

by the protection surrounding the ultrasound transducer. These different factors show the extreme difficulties encountered in the detection of the aorta section [1][5][6][7][8].

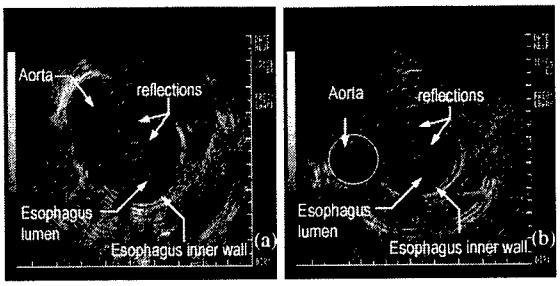


Figure 3: Echo-endoscopic 2D-slice views of the esophagus area. Aorta lumen is uniformly black and contour is clearly visible (a). Aorta lumen becomes noisy and the contour is practically invisible (b).

III.1 Global architecture

Concerning the numerical images processing, we propose a mono-sensor information fusion system based on the use of echo-endoscopic image slices of the esophagus and of *a priori* knowledge to detected the aorta section.

We have taken into account the following constraints: (i) it is necessary to preserve medical information contained in the slices. (ii) Numerical information is completed by means of models and *a priori* knowledge to make algorithms more robust. (iii) A slice by slice processing is applied, given the characteristics of the acquisition system. (iv) Aorta's shape uncertainty is handled knowing that a, b and γ are set according to a variation Δ. Considering the above constraints, two different approaches are used:

FUZZY LOGIC: allows integrating knowledge from different sources, simplifying data fusion thanks to fuzzy operators properties.

HT: detects parameterized shapes, handling uncertainty. This transformation can also include *a priori* knowledge at different levels of its implementation. Finally, HT is robust on noisy images because it is based on co-operative vote and on the notion of Accumulation Kernel (AK) [9].

III.2 Considered knowledge

Aorta visual appearance: A doctor easily denotes aorta presence in echo-endoscopic slices, but he can't precisely draw its contour. In fact, on ultrasound slices, aorta contour is very noisy. The following scheme shows elements, which perturb the artery detection.

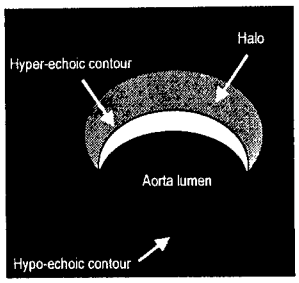


Figure 4: Pixels of interest for the aorta detection are contained in the hyper-echoic contour. Pixel within the halo must be eradicated to achieve a right detection.

The part of the aorta ellipse called hyper-echoic contour, which is opposite to the ultrasound sensor, is the primary knowledge we use. Otherwise, independently to the approach of detection that will be adopted, it is necessary to privilege hyper-echoic pixels, and contribution of the halo pixel must be less important.

Aorta position: Specialists usually consider that the aorta artery is invariant in term of position relatively to the anatomical context. Real medical exams and sequences observation leads to think that the aorta has to be searched in a region surrounding the esophagus. It is a precious information, which avoid to confuse aorta with others encountered elliptical anatomical structures (harmonics, ganglions, artery...).

Scalable trajectory model: Aorta 3D shape can be considered to be invariant with a linear transformation. Both, doctors considered a 3D model as an information we have to take in account to perform a good detection. As, for the moment, a slice by slice processing was retained, only the 3D-model projection is of great interest to be use as knowledge.

III.3 Fuzzy model

The fuzzy set theory pioneered by L. Zadeh [10][11] provides us with a powerful mathematical tool for modeling the human ability to reach conclusions when the information available is imprecise, incomplete, and not totally reliable. The major characteristic that distinguishes fuzzy set theory from traditional crisp set theory is that it allows intermediate grades of membership. A fuzzy set A over Ω is defined as the set of ordered pairs $A=\{(X,\ \mu_A(X)),\ X\in\Omega\}$, where $\mu_A(X)$ $(\in[0,1])$ is termed the grade of membership, or simply the membership value, of the element X to the fuzzy set A.

Let first introduce, the useful concept of fuzzy images. A fuzzy image is defined as the

transformation of an original image (considered as a MxN array of gray level associated with each pixel) into an image with the same dimensions but where each pixel is associated with a value denoting the degree of possessing a fuzzy property:

$$A: \quad M \times N \rightarrow [0,1]$$
$$P(x,y) \rightarrow \mu_A(P) \qquad (1)$$

where, $\mu_A(P)$ reflects the appropriateness or the validity of the fact that the pixel P possesses the fuzzy property "A". Concerning the application of the esophagus inner wall detection, four fuzzy images are defined:

Fuzzy position image: fuzzy image representing the more reliable position of the aorta section
Fuzzy intensity image: fuzzy image representing the "brightness" of different pixels.
Fuzzy gradient image: fuzzy image representing the gradient computed at each pixel.
Fuzzy region image: fuzzy image representing the contrast of each pixel relatively to the dark region of the esophagus light.

Fuzzy Position Image: In a sequence, reliable aorta position information is introduced through a manually built fuzzy image μ_p(P) where pixels gray level traduces the membership value to the aorta contour. More a pixel is far from the center i.e. the location of the esophagus, more its membership value is important (Figure 5.a).

Fuzzy Intensity Image: Physical consideration on the tissue nature lead to conclude that a large part of the aorta contour is generally hyper-echoic. Therefore, contour pixels have a high intensity. The S-shape function is applied over the gray level values in order to construct the fuzzy intensity image, μ_i(P). The S-shape parameters selection method is considered as a normalization process of the image brightness values and, thus, the visualization parameters tuning has no influence on the fuzzy intensity image (Figure 5.b).

Fuzzy Gradient Image: The edge information constitutes an important element in the determination of the aorta contour. Therefore, the fuzzy gradient image, μ_g(P), (representing the degree of membership of each pixel P to the "ill-defined" or ambiguous concept of an edge) is of great interest. For this purpose, a 5x5-gradient filter, similar to the Sobel operator, is used. The horizontal and the vertical masks of this filter are given as follows:

$$G_x = \begin{pmatrix} 0 & -1 & 0 & 1 & 0 \\ -1 & -2 & 0 & 2 & 1 \\ -1 & -2 & 0 & 2 & 1 \\ -1 & -2 & 0 & 2 & 1 \\ 0 & -1 & 0 & 1 & 0 \end{pmatrix} \quad G_y = \begin{pmatrix} 0 & -1 & -1 & -1 & 0 \\ -1 & -2 & -2 & -2 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 2 & 2 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \qquad (2)$$

Therefore, the x-y gradient of an image $I(x,y)$ is given through the following expressions :

$$\frac{\partial I}{\partial x}(x,y) = G_x * I(x,y)$$
$$\frac{\partial I}{\partial y}(x,y) = G_y * I(x,y) \qquad (3)$$

The module of the gradient is given by:

$$G_I(x,y) = \sqrt{(G_x * I)^2(x,y) + (G_x * I)^2(x,y)} \qquad (4)$$

Finally, we use the S-shape function to perform the *'fuzzification'* operation (Figure 5.c).

Fuzzy Region image: aorta halo introduces imprecision in the contour detection cause of its hyper-echoicity and area. Also, the use of a Π-function, which performs a progressive threshold as well as the *'fuzzification'*, seems to be adapted to limit influence of pixels corresponding to this region. In a fuzzy region image $\mu_R(P)$, each pixel is represented by a coefficient (i.e. membership value) denoting the degree of possessing the property: Do not be in "touch" with the region of the aorta halo (*Figure 5.d*).

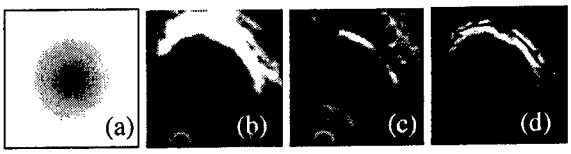


Figure 5: An example of knowledge sources extraction: position fuzzy image (a), intensity fuzzy image (b), fuzzy region image (c) and fuzzy gradient image (d).

Fuzzy Reasoning: The fuzzy reasoning step aims at the concentration of all the information previously mentioned in order to produce a single membership value, for each pixel in the analyzed image, to the aorta inner wall. The wide range of combination operators proposed in fuzzy set literature (see, for instance, [12]) reflects the power as well as the flexibility of the use of fuzzy concepts. In this study, the simple fuzzy intersection operator (i.e. a "conjunctive-type" combination operator) is used:

$$\mu_W(P) = Min\,(\mu_p, \mu_i, \mu_g, \mu_r) \qquad (5)$$

where $\mu_W(P)$ denotes the "global" degree of membership of the pixel P to the aorta wall. On the Figure 6, an example of fuzzy decision is presented. We can notice that the halo has disappeared, as well as harmonics and speckle noise.

Figure 6: Decision fusion obtained from a given image of an echo-endoscopic sequence. We note that the information is less noisy and that only pixels belonging to the aorta contour has significant membership degree.

III.4 HT Model

Hough has introduced a detection method (HT) in 1962 for identification of straight lines [13]. Duda and Hart have extended the same method to extract parameterized curves in general [14].

General idea: Let $f(X, V)=0$ be an analytic expression defining a parameterized curve, where $X=(x, y)$ define a pixel coordinate and V a parameter vector. The HT is accomplished in two steps:

The first aims to the definition of the parameter V and the quantification of the parameter space into rectangular n-dimensional cells called Hough Space. The last expression signifies that if we are given a parameter vector V, then, the curve of interest is formed by pixels in the image plane satisfying the analytic curve expression. The application of an HT consists in considering the inverse situation where we have a contour pixel E_k included in a parameterized curve and we are looking for the set of parameter vectors V that pass through this considered pixel which verify the following expression: $f(E_k, V)=0$. The locus of these vectors in the Hough Space (HS) is called Accumulation Kernel (AK) as in [9]. Let consider for a set of pixel of the same contour the set of the parameter-associated vector. In theory, as these pixels are members of the same parameterized curve, among the set of parameter associated vector, only one is in common. This vector entirely defines the search curve. Given that, each pixel can be considered as an elementary expert, which contributes to the global object detection.

Particular ellipse case: In the ellipse case, five parameters are necessary to entirely define the curve. On each slice, aorta section can be modeled as an elliptic shape according to these parameters: ellipse center coordinates (x_0, y_0), semi-major a, semi-minor b and orientation γ.

We don't directly discuss in five dimensions HS. In a first time, only a restriction space of two dimensions, corresponding to ellipse center position space, is considered.

As previously mentioned, HT needs to know curve parameters. Thus, an initialization of these parameters is required. This problem will be discussed in the next sub-section.

This operation achieved, the slice gradient is computed using a large convolution kernel. From the gradient image, two informations are extracted: the gradient magnitude, which is a criterion to accomplish a first selection of pixels implicated in the algorithm (using a threshold), and the gradient direction, which is exploited to limit the search space.

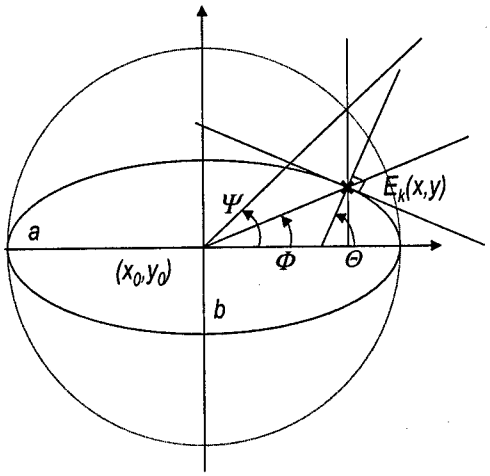


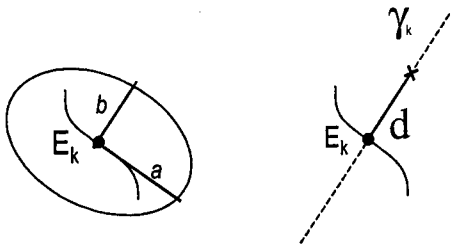
Figure 7: (x_0, y_0) is the center of the ellipse, Φ the angle to the ellipses center, Θ the direction of the gradient.

In the case where the searched ellipse parameters are a, b, γ, the geometric relation between the gradient angle and the angle relative to the center is given by the following relation (Figure 7):

$$\Phi = \gamma + \arctan((b/a)^2 \tan(\Theta) \qquad (6)$$

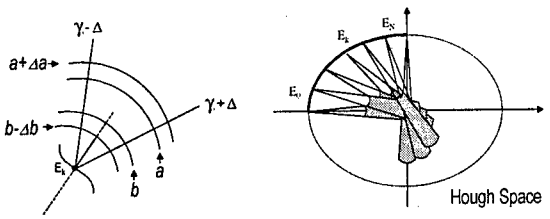
Imprecision handle: Handle of imprecision on direction radius introduced in [9] was useful to take in account fluctuation of aorta shape.

In the case of total ignorance accumulation, from a pixel edge E_k, the whole space is explored (Figure 8.a).



(a) (b)

Figure 8: Total ignorance accumulation (a), and total knowledge accumulation (a).



(a) (b)

Figure 9: Imprecise direction/radius accumulation (a). Accumulation Kernel accumulation (b).

In total knowledge accumulation, a precise direction for given distance d is observed (Figure 8.b). In the case of imprecise direction/radius accumulation, imprecision on the definition of semi-minor and semi-major axis is introduced as well as in the direction exploration (Figure 9.a). For each contour pixel, an area so called AK, corresponding to a set of parameter vectors, is computed (Figure 9.b).

The aorta center estimation is obtained computing the max of the accumulation. Finally, considering the set of pixels $S=\{M_1, M_2, ..., M_n\}$, which have contributed to this estimation, a and b parameters are re-estimate with the following method:

Given b member of the interval $[b-\Delta b, b+\Delta b]$, a is computed from each pixels of S. The b which corresponds to the minimal standard deviation of a can be considered as a good estimation of the semi-minor axis. The retained semi-major axis is given by the mean of the obtained a. Given x, y, a, b, γ, ellipse is entirely defined. Then parameters are propagated to the next slices assuming that a continuity hypothesis satisfied.

III.5 HT and Information fusion

The proposed general architecture is presented in Figure 10. We can see that others knowledges as numerical information have been introduced at three levels of the HT implementation to improve the efficiency of our method:

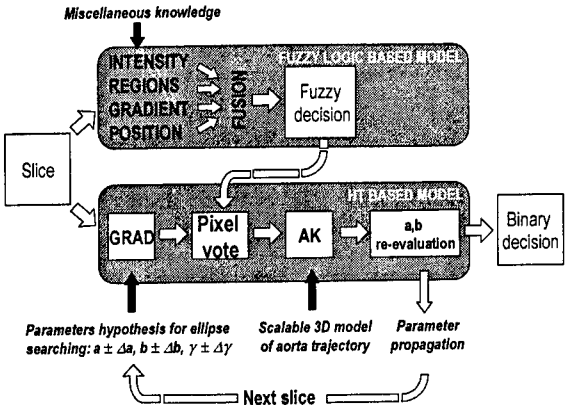


Figure 10: Aorta detection architecture based on fuzzy logic and HT. A fuzzy model based on data fusion of information extracted from a priori knowledge is merged with a HT based model.

Initialization of parameters: For the moment, initialization of ellipse searched parameters is assumed considering typical anatomical measures. But we can easily imagine a human-assist tool, which is able to assume this task for the first slice of the sequence. Once the first ellipse detected i.e. parameters evaluated, these ones are propagated to initialize the detection on the next slice.

Fuzzy decision fusion: At the level of accumulation elaboration, fuzzy decision image is used to weigh pixel vote (see [15] for a ponderation by the gradient). This method has the advantage to take in account, in the HT, both the numerical information contained in a slice, elaborated considerations as the halo problem and *a priori* knowledge on the aorta position.

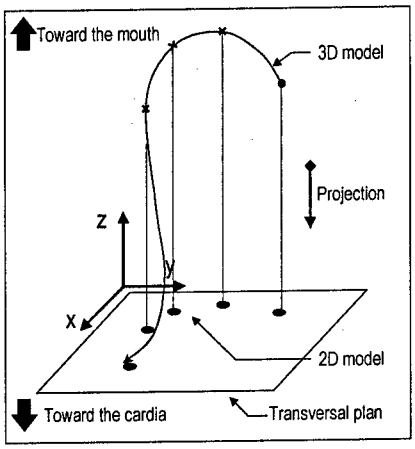


Figure 11: From aorta trajectory 3D model, we just consider the projection on the slice plan. The model

is first adjusted to the data and then used as knowledge.

Scalable 3D model: Recall the proposed solution is based on a slice by slice processing, the coefficients derive problem must be considered. Even if a continuity hypothesis is considered by introducing parameters propagation, it is a relatively local consideration, which is insufficient to assure a correct detection along the sequence. The proposed solution is based on the use of a 3D model of trajectory to assure the global coherence of the elaborated reconstruction. Cause of the 2D nature of processing, only the 2D-model projection seems useful (Figure 11). The model is first adapted to the data considering a linear transformation (in fact a similitude). Then, when the error is inferior to a given threshold, the model can be fully considered as real knowledge source.

IV. RESULTS

Images given in Figure 12 and Figure 13 come from a real sequence acquired in a medical context. We can notice the aorta contour vanishing in several parts. Finally, it is worthwhile to notice the important problem due to the harmonics presence, which can introduce error on the detection of aorta.

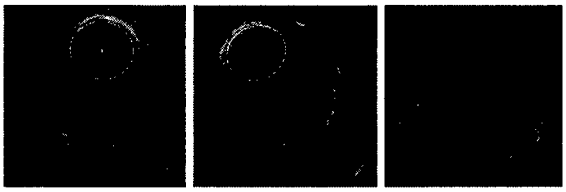


Figure 12: Detected aorta at different levels of a sequence acquired from a real patient.

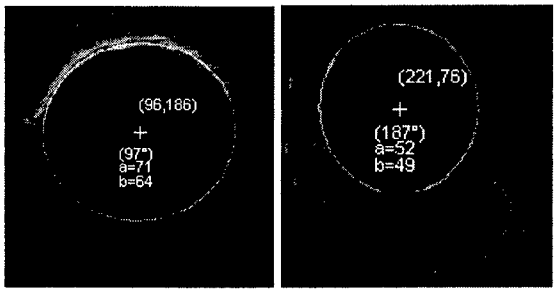


Figure 13: Obtained results on two slices of an echo-endoscopic sequence. On each frame, coordinates of ellipse center (x_0, y_0), semi-major (a) and semi-minor (b), and orientation (γ) are precised.

On Figure 12, we can see the detected aorta at different levels of the sequence. The stability of the detection is due to trajectory 3D model. This knowledge source adds a fundamental information on the aorta global shape, which fully compensates defects of a slice by slice processing sequence.

On Figure 13, two magnified views prove that the use an elliptical model is judicious to approximate the aorta contour. Such a model assures a correct precision despite its relative simplicity.

V. CONCLUSION

Obtained results are very encouraging. Simulations have shown that the processing sequence is robust enough against the noise (Figure 13), thanks to Hough Transform. Imprecision on *a* and *b* estimation at the level of the aorta bend, should be compensated by the introduction of a full 3D model taking in account both the trajectory of the center, semi-minor and semi-major axis.

In term of image processing, ultrasound slices relative positions can be corrected from this reconstruction considering shape regularity conditions.

Actual studies are conducted in order to generalize the proposed HT based information fusion system to the case of spherical anatomical structures 3D detection as ganglions.

The whole system will be soon integrated in a blackboard architecture that seems to be promising.

REFERENCES

[1] F.Pipelier, "Analyse d'images écho-endoscopiques de l'œsophage", PHD dissertation thesis, ENST-Bretagne – Université de Rennes I, 1997.

[2] Lawrence A. Klein. "Sensor and data fusion concepts and applications". Spie Optical Engineering Press Volume TT14, pp. 48-54, 1993.

[3] B. Solaiman, R. Debon, C. Roux, "Information fusion: application to data and model fusion for ultrasound image segmentation", accepted for the special issue of IEEE-Trans. on biomedical Engineering on the "Biomedical Data Fusion", 1999.

[4] B. Solaiman, R. Koffi, M.C. Mouchot and A. Hillion, "An Information Fusion Method for Multispectral Image Classification Post-processing", IEEE Trans. on Geoscience and Remote Sensing, Vol.36, No.2, pp. 395-406, March 1998.

[5] J.S. Lee. "Speckle analysis and image smoothing of synthetic aperture radar images. Computer Graphics and Image Processing", (17) 24-32, 1981.

[6] P. Wilinski, B. Solaiman, C. Roux and M. Robaszkiewicz, "Texture analysis of ultrasonographic endoscopy images using the Master Classifier Method", VII European Signal

Processing Conference, EUSIPCO-94, 13-16 Sep, Edinburgh, Scotland, 1994.
[7] F. Pipelier, P. Danduran, B. Solaiman, C. Roux and M. Robaszkiewicz, "Esophagus inner wall detection on echo-endoscopic images sequence", IEEE Workshop on Nonlinear Signal and Image Processing, pp. 799-802, 1994.
[8] B. Solaiman, F. Pipelier, C. Roux and M. Robaszkiewicz, "Esophageal wall detection using endosonographic imaging systems", IEEE Engineering in Medicine and Biology Society Conference, EMBS 95 pp. 611-612, 1995.
[9] B. Solaiman, B. Burdsall, C. Roux, "Hough transform and uncertainty handling", IEEE Transaction computer 27(8), pp. 777-781, 1997.
[10] L.A.Zadeh. "Fuzzy sets. Inform". *Inform. Contr.*, vol. 8, pp. 338-353, 1965.
[11] L.A. Zadeh, "Fuzzy algorithm, "Inform. *Contr.*, vol. 12, pp. 94-102, 1968.
[12]I. Bloch, " Information combination operators for data fusion: a comparative review with classification," IEEE Trans. Sys. Man Cyber. A 26 (1), pp. 52-67, 1996.
[13]Hough, "method and means for recognising complex pattern", PVC, US Patent, 1962.
[14]R.0. Duda, P.E. Hart, Using the Hough transforms to detect lines and curves in pictures", Comm. of the ACM, 15(1), pp. 11-15, 1972.
[15]D. H. Ballard, "Generalizing the Hough Transform to detect arbitrary shapes", Pattern Recognition 13 (2), pp. 111-112, 1981.

Session TA3
Decentralized Detection Systems
Chair: Jane O'Brien
Defense Evaluation & Research Agency, U.K.

Minimum Energy Information Fusion in Sensor Networks

George Chapline
Lawrence Livermore National Laboratory

Abstract

In this paper we consider how to organize the sharing of information in a distributed network of sensors and data processors so as to provide explanations for sensor readings with minimal expenditure of energy. We point out that the Minimum Description Length principle provides an approach to information fusion that is more naturally suited to energy minimization than traditional Bayesian approaches. In addition we show that for networks consisting of a large number of identical sensors Kohonen self-organization provides an exact solution to the problem of combing the sensor outputs into minimal description length explanations.

Key Words: self-organization, fusion

1. Introduction

One of the grand challenges of cognitive science is to understand how, at least in principle, a network of sensors and simple data processors might be configured to "understand" what is going on its environment. In general forming perceptions from sensor outputs will require a network of sensors because noise or insufficient selectivity may not allow individual sensors to provide unambiguos signals regarding the environment. It should be kept in mind in this connection that increasing the sensitivity of an individual detector will not lead to an increase in the signal to noise ratio for the signatures of interest unless some scheme for background subtraction is available. The upshot is that even in networks where the individual detectors are very sensitive, it will in general be desirable to correlate or "fuse" the signals from different kinds of sensors or sensors in different spatial locations.

When one is considering the problem of combing information from different sensors it is tempting to use Bayesian probabilistic reasoning [1] or its Demster-Shafer generalization [2]. One of the attractive features of a Bayesian approach to information fusion is its adaptability to incremental computational schemes [1], which allow one to pool the evidence from different sensors hierarchically using a tree-like network. In particular each node of a Bayesian data fusion tree combines the conditional probabilities for the units which proceed it in some ordering to form a new set of conditional probabilities. These Bayesian networks often incorporate unobserved latent variables known as hidden variables, and such networks have been successfully used for some quite difficult real world pattern recognition problems such as speech recognition. Bayesian-type networks are also attractive for combining the outputs of neural network classifiers [2]. On the other hand when applied to the problem of information fusion in an autonomous network of sensors and associated data processors neither Bayesian probabilistic reasoning nor the Dempster-Shafer method seem by themselves to offer any particular insights into the important problem of how to minimize overall energy usage in the network. In this paper we will argue that in contrast with Bayesian techniques the Minimum Description Length (MDL) principle [3] appears to be an ideal statistical inference methodology to use when energy usage in the network is an important constraint.

The MDL principle has been gaining in popularity as a fundamental alternative to Bayesian reasoning for statistical inference for several reasons. Two well-known problems with Bayesian reasoning are a) a priori probability distributions may not be known or even exist, and b) Bayesian methods are impractical when there are many possible explanations for a given instance of environmental data. While the Neyman-Pearson likelihood ratio test is uncontested as the best thing to use when there is just one hypothesis to test, there is no similarly canonical

method when there are many approximately equally likely explanations for the environmental data. Indeed, not only does keeping track of the conditional probabilities for a possibly exponentially large number of hypotheses make hierarchical Bayesian fusion schemes difficult to implement, but choosing the single largest conditional probability to select a particular hypothesis could give the wrong answer. On the other hand the MDL principle was the inspiration for the Helmholtz machine [4], which is a promising approach to dealing with the combinatorial complexities associated with data whose explanation is ambiguous.

A third general problem with Bayesian methods is that they don't by themselves address the important question of minimizing the complexity of coded representations. A corollary of this second point is that Bayesian methods don't seem to be particularly well suited to the problem of optimizing the energy usage in an sensor network. However by focusing on the simplest possible way to explain environmental data the MDL principle appears to be very well suited to minimizing energy usage in a sensor network. In the following section we briefly review the MDL approach to pattern recognition. The basic idea here is that overall description costs are minimized when the probabilities for various explanations are related to their description costs by the Boltzmann distribution. In section 3 we show that MDL explanations for the outputs of a large number of identical sensors can obtained using Kohonen's algorithm for self-organization. In section 4 we compare the energy requirements for sensor fusion using distributed self-organization with the energy requirements for sensor fusion using a hierarchical Bayesian network.

2. Minimal description length approach to pattern recognition

It has been understood for some time that pattern recognition systems are in essence machines that utilize either preconceived probability distributions or empirically determined posterior probabilities to classify patterns [5]. In the ideal case where the a priori probability distribution $p(\alpha)$ for the occurrence of various classes α of feature vectors and probability densities $p(x|\alpha)$ for the distribution of data sets x within each class are known, then the best possible classification procedure would be to simply choose the class α for which the posterior probability

$$P(\alpha \mid x) = \frac{p(\alpha)p(x \mid \alpha)}{\sum_{\beta} p(\beta)p(x \mid \beta)} \qquad (1)$$

is largest. Unfortunately in the real world one is typically faced with the situation that neither the class probabilities $p(\alpha)$ nor class densities $p(x|\alpha)$ are precisely known, so that one must rely on empirical information to estimate the conditional probabilities $P(\alpha \mid x)$ needed to classify data sets. In practice this means that one must adopt a parametric model for the class probabilities and densities, and then use empirical data to fix the parameters θ of the probability model. Once values for the model parameters have been fixed, then sensory data can be classified by simply substituting values for the model probabilities $p(\alpha; \theta)$ and $p(x|\alpha; \theta)$ into equation (1).

Unfortunately determining values for the model parameters from empirical data is itself a computationally intractable problem. This means that in practice one is usually limited to using models of relatively modest complexity, and consequently one is always faced with the issue of choosing the best possible values for the model parameters. A very elegant approach to this problem was suggested some time ago by Rissanen [3], who suggested choosing a model such that the length of binary code needed to represent the model is minimized. The description length for a binary variable s_i is:

$$E(s_i) = -s_i \log p - (1-s_i)\log(i-p_i), \qquad (2)$$

This leads us to define the description cost or "energy" of a classification α to be

$$E_\alpha = \sum_i E(s_i) + \sum_i E(x), \qquad (3)$$

where the x are the variables needed to describe the input data and the s_i are the variables needed to represent the interpretation of the input data. One might think that the pattern recognition algorithm should be chosen to minimize E_α, but this is incorrect because it is possible [3] to devise coding schemes that take advantage of the entropy of alternative explanations for the input data. The effective cost $F(x)$ for describing a data set x with explanations $\alpha=\{s_i\}$ is

$$F(x) = \sum_\alpha \{E_\alpha P(\alpha) - (-P(\alpha)\log P(\alpha))\}. \quad (4)$$

The quantity $\sum_\alpha P_\theta(\alpha)\log[P_\theta(\alpha)/P(\alpha)]$ in the second term in equation (4) is always positive and measures of the difference in bits between the model distribution $P_\theta(\alpha)$ and the true distribution $P(\alpha)$. This distance measure, known as the Kullback- Leibler divergence, is the basis for the Maximum Likelihood estimator that is widely used by statisticians to measure how well a given set of model probabilities reproduces the empirical data [5]. As in physics $F(x)$ is minimized when the probabilities of alternative explanations are exponentially related to their costs by the canonical Boltzmann distribution:

$$P(\alpha \mid x) = \frac{e^{-E_\alpha}}{\sum_\alpha e^{-E_\alpha}}. \quad (5)$$

Thus a minimal cost recognition model should produce a probability distribution Q(α) that is as similar as possible to the Boltzmann distribution (5).

Of course we are still left with the problem of how to generate explanations and conditional probabilities $P(\alpha)$ that satisfy equation (5). An ingenious approach to generating explanations {α} for which the posterior probabilities P(α| x; θ) are naturally represented in the canonical Boltzmann distribution form (5) was introduced in 1985 by Ackley, Hinton, and Sejnowski [6]. In this model, known as the *Boltzmann machine*, environmental data and their "explanations" are represented by configurations of binary units with activation levels a_i= 0 or 1. The energy function for the assembly of binary units is assumed to have the same form as that used by physicists to describe a system of interacting spins in a magnet:

$$E(\mathbf{a}) = -\frac{1}{2}\sum_{i\neq j} w_{ij}a_i a_j + \sum_i \theta_i a_i, \quad (6)$$

where $\mathbf{a} = \{a_i\}$ denotes the set of activation levels and the weight w_{ij} describes the interaction strength between binary units i and j In the original version of the Boltzmann machine these interactions are assumed to be symmetric; i.e. $w_{ij} = w_{ji}$. However layered versions of the Boltzmann machine with asymmetric weights; i.e. $w_{ij} \neq w_{ji}$, are also of interest because they are equivalent to Bayesian decision networks [7]. In both the symmetric and asymmetric Boltzmann machines the probability distribution $P_\theta(\alpha)$ will be the probability distribution for the activation levels in a certain subset, referred to as the hidden units, of all binary units. The remaining binary units, referred to as the visible units, represent the environmental data x. The model parameters θ for the Boltzmann machine are the connection strengths w_{ij} and biases θ_i for the binary units. These parameters are determined by minimizing the Kullbach-Leibler divergence between the probability distribution $P_\theta(\alpha)$ with the visible unit activation levels fixed and the probability distribution for classifications with the activation levels of the visible units allowed to vary freely. Used as a pattern recognition device the Boltzmann machine has the virtue that high order correlations between different instances of environmental data can be represented and used in the classification of data sets. This means that the classifications provided by the Boltzmann machine take into account more information than just the relationship between a class and its feature vectors. Unfortunately Boltzmann machines have not found many practical applications because determination of the connection strengths and biases for realistic data sets is very slow because of the necessity for repetitive Monte Carlo sampling of a joint probability distribution

for the activation levels of the hidden units.

A quasi-deterministic version of the Boltzmann machine, known as the Helmholtz machine [4, 8], assumes the binary nodes are organized into layers and that there is Markov transition probability for going between layers of the form

$$p(a_i(n+1) \mid a(n)) = \sigma[\beta(1-2a_i(n+1))\sum_j w_{ij}a_j], \quad (7)$$

where $\sigma(x) = 1/[1+\exp(-x)]$ and for each node of the activation level a_i =0 or 1. The vector $a(n) = \{a_i(n)\}$ in equation (7) denotes the set of activation levels at layer n of the network. One can also think of the way activation levels vary from layer to layer as describing the time evolution of a system of binary units [9]. If one assumes that the activities of the binary units within a given layer are independent, then the probability of a particular explanation α, which we identify as the "time history" {a(n), n>1} of activations will be given by [8]:

$$Q(\alpha)=\prod_{n>1}\prod_j [p(a_j)]^{a_j}[1-p(a_j)]^{1-a_j} \quad (8)$$

so that the binary units that are turned on contribute with weight $p_j(x)$ while the units that are turned off contribute with weight $1-p_j(x)$. In order to determine the recognition weights w_{ij} Hinton et. al. employ a parallel "fantasy" generation network to generate a model distribution $P_\theta(\alpha)$. The weights of the fantasy generation network are chosen so as to minimize the Kullback- Leibler divergence between the model distribution and the probability values for training the recognition connection weights w_{ij} using standard neural network training algorithms.

By restricting its attention to distributions of the form (8) the Helmholtz machine finesses the combinatorial problem associated many hypotheses. Therefore organizing a network of sensors and data processors as a Helmholtz machine, as was previously recommended by the author [10], might seem like a good idea. However, two aspects of the Helmholtz machine architecture seem problematical in connection with the problem of energy minimization. The first is that even though the Helmholtz machine attempts to minimize the free energy $F(x)$, by restricting attention to distributions of the form (8) it is not clear how close one can approach to the ideal Boltzmann distribution (5). A second problem is that each node in a given layer will in general be connected to every node in the previous layer. Compared with a hierarchical Bayesian network this would increase the number of communication links in a network of N total nodes by a factor on the order of N/L, where L is the number of layers. However, replacing the fully interconnected network used in the Boltzmann machine with the quasi-deterministic evolution of a string of bits does point us in the direction of the exact model for MDL information fusion described in the next section.

3. Self-organization approach to MDL information fusion

Let us suppose that our sensor network consists of N feature detectors, and that each feature detector can communicate with three neighboring feature detectors. The assumption of three communication links per node is made for convenience since models where the feature detectors are allowed to connect to a larger (but fixed) number of neighbors lead to similar results. Also for simplicity we will assume that the feature being looked for can be characterized by a single continuous variable w such that $0 \le w \le 2\pi$; leaving for the future the more typical case where the features are characterized by a vector in a higher dimensional space. In addition we assume that every sensor is looking at the same environment. As an initial condition for the network we assign to each sensor a value ϕ of the feature that is randomly chosen from a probability distribution for the occurrences of various features in the environment. Now intuitively it seems clear that since in principle nearby sensors ought to have the similar outputs, a minimal description of the sensor outputs ought to involve just

giving the parameters of a smooth curve for w vs. location **r** of the sensor nodes. Therefore we will guess that the data processing required for minimum description information fusion can be modeled by assuming that the maps of sensor locations into feature space are "self-organizing. If we follow Kohonen's prescription for self-organization [11], this means that the orientation of the feature detector located at r will evolve according to a rule of the form

$$w(r,t+1) = w(r,t) + h(r-s)[\phi - w(r,t)], \quad (9)$$

where h(r) is typically assumed to be a Gaussian function peaked at r=0. For our purposes the function h(r-s) can be replaced by the rule that each feature detector is connected to just three of its nearest neighbors. The location s in (9) corresponds to the feature detector whose orientation w(s) is closest to ϕ. Thus the data fusion process is modeled as a Markov process whose states are the sets {w(r)} of possible states of the feature detectors, and where the transition probabilities are determined b y probabilities of occurrence in the environment of various orientations ϕ. In order to construct an analytical model of this evolution process it will be useful to introduce an energy functional E[w] that satisfies

$$< P(\phi)\delta w >= -grad_w E \quad (10)$$

where $\delta w = w(r,t+1) - w(r,t)$ and P(ϕ) is the a priori probability distribution for the orientations of the environmental stimuli. Neglecting certain mathematical subtleties, the required energy functional is [12]

$$E[w] = \frac{1}{2}\sum_{<r,s>}\sum_{\phi \in R} P(\phi)|\phi - w(r,t)|^2 \quad (11)$$

where the sum over <r,s> runs over nearest neighbor connections and *R* (r) is the receptive field of the feature detector located at r; i.e. the union of all environmental stimuli that are closer to w(r, t) than any other w(s,t), where $s \neq r$.

Given an energy functional that satisfies (10) there are standard techniques that one can use to describe the stochastic evolution of the organization of our neural network. However here we will limit our interest in how the organization of feature detectors evolves with time to noting that under the influence of the random variable ϕ(t) the system relaxes to an asymptotic state characterized by a stationary probability distribution for various final configurations of feature vectors {w(r)} . Given the existence of an energy functional satisfying (19) the statistical properties of the set {w(r)} in this stationary state can be derived from a "partition function" Z = exp[-*F* (*x*)] which is a sum over all possible stationary state configurations weighted with the Boltzmann factor exp(-E[w]). If we assume that the stochastic evolution of network is governed by an energy functional of the form (11) then this partition function has the form [13]:

$$Z = \sum_L \kappa^F \prod_{i=1}^{F} \int_o^{2\pi} dw(r_i) e^{-\frac{K}{2}\sum_{<i,j>}|w(r_i)-w(r_j)|^2} \quad (12)$$

where κ and K are constants, the sum over*L* means a sum over triangular lattices, and the indices i and j refer to orientation sensitive neurons located at the centers of the triangles in this lattice (note that N is the number of faces of the triangulation *L*). For large numbers of faces the triangulation*L* can be thought of as approximating a smooth 2-dimensional surfaces, and in the limit N $\rightarrow \infty$ the sum over triangular lattices in eq.12 becomes a sum over smooth 2-dimensional surfaces. In this limit the partition function (12) becomes

$$Z = \int Dw(\sigma)\exp(-S), \quad (13)$$

where (σ_1, σ_2) are the coordinates of a point on the smooth surface and the continuum action S is given by

$$S = \frac{K}{2}\int d^2\sigma \partial_\alpha w \partial_\alpha w + \lambda \quad . \quad (14)$$

The constant λ in (14) replaces the constant κ and plays the role of an energy per node. It turns out that the partition function (13) has an interesting physical interpretation [13]; namely, it represents the quantum theory of a "string" moving on a 2-dimensional surface - in mathematical terms this means random holomorphic mappings from a 2-dimensional manifold onto a fixed 2-dimensional manifold. In this string interpretation the angle variable w becomes a complex variable by the addition of a second real variable representing the local magnification of the mapping. It is worth noting that this result is consistent with the theorem [14] that for maps of 2-dimensional surfaces onto 2-dimensional surfaces the stationary state of Kohonen's algorithm is a holomorphic (or anti-holomorphic) map. Thus we have the general result that the feature vector will be a smooth analytic function of position, in accordance with our initial expectations. As noted in ref. 13 this formalism also determines the topological connectivity of the 2-dimensional surfaces involved; therefore in contrast with other approaches to information fusion the network topology is not an extra ad hoc assumption, but follows from the MDL principle.

We can now see why somewhat miraculously Kohonen self-organization provides an exact solution to the problem of finding minimally complex explanations for the outputs of a large number of sensors. In the large N limit explanations are represented by smooth mappings of a 2-dimensional surface representing the physical layout and connectivity of the network into feature space. The information cost of any particular explanation is just an exponential of minus the quantized area of the surface in feature space given in eq. 14. The natural unit of quantization, i.e. the area equivalent to 1 bit, is determined by the inverse of the constant K in eq. 14. The cost averaged over environmental inputs is just the negative logarithm of the partition function Z defined in eq. 13.

4. Hierarchical versus distributed information fusion

It is self-evident that other things being equal generation of a minimal binary representation for feature vectors and explanations for feature vectors will minimize the energy usage in any sensor network. A remaining question though is how to compare the energy costs of hierarchical Bayesian network with those of network that fuses sensor outputs via Kohonen self-organization. In a self-organized network of sensors and data processors the information fusion processes are distributed throughout the network. However if, as we have been implicitly assuming, the different sensors in the network are physically separated then some means must be provided for these nodes to communicate with each other. In a Bayesian network the communication support must be capable of relaying the conditional probabilities at one decision level of the network to the data fusion units in the next level of the network within a relevancy time interval. Thus an interesting question is how the data processing and communication energy costs for information fusion in a network using distributed self-organization compare with these costs in a network using hierarchical Bayesian reasoning.

If one uses a conventional hierarchical data fusion strategy [see e.g. 15] where separate data fusion nodes collect information from sensor nodes, then every data fusion node in the system must incorporate a Bayesian inference engine which calculates conditional probabilities for all the relevant hypotheses. In a Bayesian tree-like network of data fusion and sensor nodes with a total of N nodes, these conditional probabilities must be calculated at each of the N nodes and communicated to the node in the next higher level. If every data fusion node receives information from say 3 nodes in the next layer down, there will be approximately lnN layers and 2N/3 communication links in the network for large N. The total amount of information that needs to be transmitted from one layer of the network to the next will be on the order of (N/ lnN) $\sum_{\alpha}$ (E_α +(-ln $P(\alpha)$)

where E_α is the description cost for

hypothesis α. On the other hand, in a network of M sensors using a self-organization scheme of the type discussed in the previous section for data fusion, the conditional probabilities are not directly calculated; instead they are coded into the description length for the feature vectors. This is a tremendous advantage because initially one can choose the most likely feature vector for every sensor node. Furthermore, after a certain number of iterations of Kohonen's algorithm the M feature vectors are compressed into a smooth function. Therefore in a self-organizing network the amount of information that must be processed during the data fusion process is enormously reduced because one isn't carrying along conditional probabilities for every possible hypothesis.

If every sensor node in a self-organizing network communicates with 3 of is neighbors, the number of communication links that must be established to implement Kohonen self-organization will be approximately equal to 3M/2. If we assume every data fusion node in a hierarchical Bayesian network also functions as a sensor node so N=M, we see that the minimum number of communication links required in a self-organizing network with the same number of sensors will be approximately 9/4 the required number of links in a hierarchical Bayesian network. However as one moves from one step of the data fusion process to the next the amount of information that must be transmitted in a self-organizing network will be vastly smaller when the number of hypotheses to be tested is very large. As discussed in the last section one can initiate the self-organization process by independently choosing feature vectors for each sensor according to the probability that they occur in the environment. However in reality the sensor readings are not independent, and it would make more sense to initially replace each sensor output with the most likely explanation for the sensor output. In this case the total amount of information transmitted between sensor nodes for each iteration of the Kohonen algorithm will be on the order of $(3M/2)\,F(x)$, where $F(x)$ is the average description cost for the explanations.

Since $F(x)$ will be on the order of $H\ E_\alpha$, where H is the number of hypotheses, we see that the communication costs in a hierarchical Bayesian network will be larger than those in a self-organizing network with the same number of sensors by a factor on the order of H. If we assume that each data fusion node in a hierarchical Bayesian network combines the conditional probabilities from three nodes then the computational costs for each hypothesis are similar in the Bayesian and self-organizing networks. However in the Bayesian network the conditional probabilities must be updated for each hypothesis; therefore the computational cost per node will be on the order of H larger in the Bayesian network. The end result is that the relative energy costs of moving from one step of the data fusion process to the next in a distributed self-organizing versus a hierarchical Bayesian network will be on the order of

$$\text{Bayesian / Self-organization energy cost} \approx \frac{H}{\ln N} \qquad (15)$$

5. Conclusion

We see that when there is only one hypothesis to test the energy cost for fusing sensor outputs using a hierarchical Bayesian network is not significantly different form the MDL energy costs of a self-organizing network. However, when the number of hypotheses to be tested is very large, then the energy costs of using distributed self-organization to fuse sensor outputs will be very much smaller. We should also note that because the fusion process in a self-organizing network is distributed throughout the network, using self-organization for information fusion also has the advantage of greater reliability. In addition we have seen that distributed self-organization may be better able to deal with the combinatorial complexities associated with ambiguous explanations. It is of course tempting to speculate that the energy saving and reliability features of distributed self-organization, as well as the ability to cope with ambiguous environmental stimuli, are principal reasons why

biological evolution has favored self-organization and complete decentralization of the cognitive processes in the mammalian brain.

Acknowledgments: The author is grateful for discussions with Bob Bryant and Chris Cunningham. Work performed under the auspices of the U.S. Dept. of Energy under contract W-7505 -Eng-48.

References

1. 8. J. Pearl, *Probabilistic Inference in Intellegent Systems* (Morgan Kaufmann 1988).
2. G. Rogova and R. Menon, "Decision Fusion for Learning in Pattern Recognition", in *Proceedings of the First International Conference on Multisource-Multisensor Fusion* (CREA Press Athens, Georgia 1998).
3. J. Rissanen, *Stochastic Complexity in Statistical Inquiry* (World Scientific 1998).
4. G. E. Hinton, et al , Science 268, 1158 (1995).
5. B. D. Ripley, *Pattern Recognition and Neural Networks* (Cambridge U. Press, 1996).
6. G. E. Hinton and T. J. Sejnowski, Cognitive Science 9, 147 (1985).
7. R. M. Neal, Neural Comp. 4, 832 (1992).
8. P. Dayan, G. Hinton, and R. Neal, Neural Comp. 7, 889 (1995).
9. W. A. Little, Math. Biosci. 19, 101 (1974); G. L. Shaw and R. Vasudevan, Math. Biosci. 21, 207 (1974).
10. G. Chapline, "Sentient networks", in *Proceedings of the First International Conference on Multisource-Multisensor Fusion* (CREA Press Athens, Georgia 1998).
11. T.Kohonen, *Self-Organization and Associative Memory* (Springer-Verlag1987).
12. Ritter, H. and Schulten, K. , *IEEE International Conference on Neural Networks* I 109 (1988).
13. G. Chapline, Network: Comp. Neural Syst. 8, 185 (1997).
14. Ritter, H. and Schulten, K., *Biol. Cybern.* 54 99 (1986).
15. R. Lobbia and M. Rakijas, SPIE 3081 110 (1997).

FUSE – Fusion Utility Sequence Estimator

Belur V. Dasarathy
Dynetics, Inc.
P. O. Box 5500
Huntsville, AL 35814-5500
belur.d@dynetics.com

Sean D. Townsend
Dynetics, Inc.
P. O. Box 5500
Huntsville, AL 35814-5500
sean.townsend@dynetics.com

Abstract - *An expert system GIFTS (a Guide to Intelligent Fusion Technology* Selection) *developed to aid sensor fusion system design, was presented at Fusion 98 as an on-going project with additional support tools under development. In this paper, a simulation tool, FUSE, that exercises a decisions in - decision out (DEI-DEO) fusion model to estimate the benefits (utility) along a sequence of multiple observations, is presented. This can be employed either independently or as one of the tools supporting GIFTS. FUSE permits the simulation of different Boolean fusion logic functions in the context of sensor suites with two independent sensors. The inputs to FUSE are the sensor performance characteristics in terms of the probabilities of correct and incorrect decisions for target and decoy classes along with parameters that define the fusion logic and duration. The outputs of the system are the fused system performance expressed in terms of probabilities of correct-, incorrect- and non- decisions over the specified range of observations. Whenever any of these input parameters are altered, FUSE responds instantaneously by updating the fused system performance. In order to further aid the user in the comparison of the different fusion logic alternatives and to assess the benefits of temporal fusion through multiple independent observations, FUSE provides several graphic visualization options.*

Key Words: decision fusion, fusion benefits, fusion logic, temporal fusion

1. Introduction

A common question that arises, especially from outside the sensor fusion community, is why fuse the sensors at all. An often heard comment is: "Why should I fuse my two sensors when sensor number one has superior performance? I will just be degrading its performance by mixing in less reliable information." Of course fusing sensors can be beneficial, but it is often hard to convey this message without showing hard data to the skeptic.

Even assuming that one is convinced of the advantages of pursuing fusion, a second question is often how the fusion should be accomplished. This is the challenge of determining what to fuse, when to fuse [1], and how to fuse. There is abundant literature offering different methods of accomplishing fusion [2,3], but few universal rules to follow. Instead, each scenario has been individually analyzed and all methods have to be considered in the appropriate context.

FUSE (Fusion Utility Sequence Estimator) is designed to address both of these questions, albeit in a limited fashion. To help with the first problem (addressing the utility of fusion), FUSE can be used as a stand-alone fusion simulator. A user simply inputs appropriate values for the individual sensor characteristics through the user interface and fused decision probabilities are immediately updated. Hence, the advantages of fusion can be instantaneously gleaned. To further aid in examining the fusion benefits, graphical representations can be displayed.

FUSE, when used in conjunction with GIFTS (a Guide to Intelligent Fusion Technology Selection) [4], addresses the larger problem as well. GIFTS guides a user through an interactive query session that defines a fusion system architecture that is appropriate to the problem environment under consideration. Included within the GIFTS architecture, are several support tools that provide assistance to the designer in developing and assessing the detailed fusion system design corresponding to the chosen architecture. FUSE is an additional assessment tool that can be included in the GIFTS architecture or can be operated as a stand-alone simulation. FUSE employs a decisions in -decision out (DEI-DEO) fusion model to estimate the benefits along a sequence of multiple observations using a two-sensor suite under AND and OR boolean logic.

In this paper, the initial version of FUSE will be introduced. In Section 2, an overview of the basic

fusion concepts underlying the estimation techniques is presented along with a summary of the GIFTS architecture. Section 3 is a description of the FUSE simulation and section 4 presents how FUSE can be used as a realistic analysis tool. Section 5 offers some closing comments and outlines the scope for further development.

2. Background

This section contains the basics of the methods by which FUSE estimates fusion benefits and the GIFTS architecture. For more details on the estimation techniques see [5]. Those interested in GIFTS should look up [4].

2.1 Fused Probability Estimation

There are two basic fusion strategies that are used in FUSE. The two strategies are OR and AND boolean logic. Both strategies operate in an environment where two sensors are operating in parallel, have a provision for multiple looks, and have a non-decision option as well as the normal binary decisions. OR logic fuses decisions by making a binary decision if the two sensors are not contradictory and a non-decision otherwise. AND logic, on the other hand, requires that the two sensors make concurring decisions to obtain a binary decision and a non-decision otherwise.

Let c_{ij}, w_{ij}, and u_{ij} correspondingly represent the probabilities of correct, incorrect, and non-decision of objects j ={Target (T), Decoy (D)} by the sensors i = {1,2}, where both sensors are deemed independent. Similarly, p_{fj}^{k}, q_{fj}^{k}, and r_{fj}^{k} represent the fused probabilities of correct, incorrect, and non-decision of object j after the k^{th} fusion attempt under logic f = {OR (o), AND (a)}. Using these definitions we note that

$$c_{ij} + w_{ij} + u_{ij} = 1. \tag{1}$$

It can then be shown for OR logic that

$$p_{oj}^{1} = c_{1j}c_{2j} + c_{1j}u_{2j} + u_{1j}c_{2j} \tag{2}$$

$$q_{oj}^{1} = w_{1j}w_{2j} + w_{1j}u_{2j} + u_{1j}w_{2j} \tag{3}$$

$$r_{oj}^{1} = u_{1j}u_{2j} + c_{1j}w_{2j} + w_{1j}c_{2j}. \tag{4}$$

Similarly, the following equations can be developed for AND logic.

$$p_{aj}^{1} = c_{1j}c_{2j} \tag{5}$$

$$q_{aj}^{1} = w_{1j}w_{2j} \tag{6}$$

$$r_{aj}^{1} = u_{1j}u_{2j} + c_{1j}w_{2j} + w_{1j}c_{2j} + c_{1j}u_{2j} + u_{1j}c_{2j} + w_{1j}u_{2j} + u_{1j}w_{2j} \tag{7}$$

The k^{th} probabilities can thus be written as

$$P_{fj}^{k} = \sum_{i=1}^{k} p_{fj}^{i} = p_{fj}^{1}\frac{1-[r_{fj}^{1}]^{k}}{1-r_{fj}^{1}} \tag{8}$$

$$Q_{fj}^{k} = \sum_{i=1}^{k} q_{fj}^{i} = q_{fj}^{1}\frac{1-[r_{fj}^{1}]^{k}}{1-r_{fj}^{1}} \tag{9}$$

$$r_{fj}^{k} = 1 - P_{fj}^{k} - Q_{fj}^{k} \tag{10}$$

$$r_{fj}^{1} = 1 - p_{fj}^{1} - q_{fj}^{1}. \tag{11}$$

These eleven equations form the basis of the fusion benefit analysis in FUSE.

Three types of fusion benefits will be defined. These are with respect to the probability of correct decision, probability of incorrect decision, and both. A fusion benefit exists with respect to the probability of a correct decision when

$$P_{fj}^{k} > \max(c_{1j}, c_{2j}). \tag{12}$$

Similarly, the fusion benefit with respect to the probability of an incorrect decision would be

$$Q_{fj}^{k} < \min(w_{1j}, w_{2j}). \tag{13}$$

A joint fusion benefit is thus when both (12) and (13) simultaneously hold true.

2.2 GIFTS

GIFTS is currently composed of four modules. The primary component is the architecture selection process that determines the relevant architecture. The second piece is a reference database that can be used to help answer problem specific questions. The third part is an FEI-DEO fusion selector. It provides a means of choosing an implementation of FEI-DEO fusion. The final component is FUSE that is discussed in this paper to simulate DEI-DEO fusion.

Through the use of all the modules, GIFTS can provide aid to the fusion system architecture designer during the different phases of development. The user, who knows the specifics of a fusion problem, uses GIFTS to determine a proposed architecture. This is accomplished by responding to problem spe-

cific questions posed by the primary component of GIFTS. Once the proposed architecture has been created, the user will begin a problem specific refining process that will produce a fusion solution. During this time, the goal is to determine the optimal means of implementing the different fusion modes in the proposed architecture. (Of course the option of not utilizing a fusion mode in the proposed architecture is always available. It may be that even though fusion is practical in this mode, there is no reasonable means of implementing it in the user's application, or a restraint outside the realm of GIFTS could be a limiting factor.)

It is at this point in the development process that the remaining modules of GIFTS will be useful. The reference tool can be used to provide sources of information on different fusion levels. The reference tool will provide a list of references that are related to the fusion modes in the proposed architecture. Thus, the reference tool makes use of the knowledge gained by the primary component. Similarly, if the user has not previously determined methods for performing FEI-DEO fusion or making local decisions, then the FEI-DEO fusion selector will be helpful. In this tool, the user is asked application specific questions to determine the most appropriate implementation method. The FUSE tool would be used to investigate DEI-DEO fusion as discussed in this paper.

3. The FUSE Tool

FUSE is currently implemented on a PC using Visual C++ [6]. It will thus run with no alterations on Windows 95, Windows 98, or Windows NT. The user can alter the sensor characteristics by choosing "Fusion Inputs" from the Fusion menu, or by clicking on the fusion characteristics button on the toolbar. Figure 1 shows the user interface with the Fusion menu activated. After "Fusion Inputs" has been chosen, the Data Definition Dialog Box (DDDB) will appear. It is through this dialog box that the fusion characteristics can be altered. Figure 2 is the default setting of the DDDB.

The DDDB consists of two group boxes labeled "Inputs" and "Fused Decision Probabilities", respectively along with a button label "OK."

The "Inputs" group is where the fusion characteristics are controlled. The inputs that can be altered are: decision probabilities for sensor 1, decision probabilities for sensor 2, the type of fusion logic, and the number of looks permitted for each sensor. Note that the user can have control over the correct and incorrect decision probabilities for both of the possible binary decisions (T and D). This allows for the maximum flexibility in the definition of a sensor. These probabilities can be entered by directly typing in the desired number of by using the adjacent sliders. It should also be pointed out that from equation (1), the four non-decision probabilities are uniquely defined by the inputs. The fusion logic is selected by a simple check box (checked for AND logic and unchecked for OR logic). The number of looks are entered by typing the appropriate integer in the box labeled "Number of Looks."

The "Fused Decision Probabilities" group is where the fused results, based on the above inputs, are displayed. The fused correct, incorrect, and non-decision probabilities are displayed for both the T and D binary decision.

The calculation of fused probabilities is only one aspect of FUSE. FUSE also provides a collection of visualization tools to help analyze the results. Each of these options are accessed through the "Fusion" menu. (Note that to activate the "Fusion" menu the input dialog cannot be open. Hence, if the dialog box is open then the "OK" button needs to be selected to exit the dialog box.) The graphical options are:

"Plot Target"- plots the fused correct, incorrect, and non-decision probabilities, for the T object, against the number of looks,

"Plot Decoy" - plots the fused correct, incorrect, and non-decision probabilities, for the D object, against the number of looks,

"Plot Target And/OR" - plots the fused correct probability, for the T object, under both AND and OR logic against the number of looks,

"Plot Decoy And/OR" - plots the fused correct probability, for the D object, under both AND and OR logic against the number of looks,

"Plot Target Fusion Benefits" - plots the fused correct and incorrect probabilities, for the T object, against the number of looks while shading the domain of fusion benefit for each and marking the domain of joint benefit, and

"Plot Decoy Fusion Benefits" - plots the fused correct and incorrect probabilities, for the D object, against the number of looks while shading the domain of fusion benefit for each and marking the domain of joint benefit.

The definition of a domain of joint fusion benefit is the intersection of the domains of correct and incorrect fusion benefit. The domain of correct (incorrect) fusion benefit is the domain where the correct (incorrect) fusion benefit exists. The correct and incorrect fusion benefit domain can be thought of as the domain bounded by a fused probability curve (with respect to the number of looks) and the best performance of a single sensor. As an example, if the probability of correct, incorrect, and non-decision for sen-

sor 1 are 71%, 16%, and 13% respectively and 45%, 15%, and 40% for sensor 2, then the best performance for a single sensor would be a correct-decision probability of 71%. Thus the fused correct probability curve and a horizontal line would bound the domain of fusion benefit for the probability of a correct decision at 71%.

4. FUSE Usage Illustration

For FUSE to be of practical value, one needs to be able to exercise it in a realistic scenario. It is meant to be a utility to assist a fusion system designer. To demonstrate its utility, consider the following scenario. A fusion system is being designed that employs two sources of decisions (or sensors) that are independent and capable of multiple looks. (An example of such a system would be a target acquisition system that employs an active X-band radar and a passive IR sensor.)

The goal is to balance the performance requirements of the system against the costs. Often this balance is obtained while using individual sensors that make decisions below system specifications and obtain decision probabilities that meet specifications through fusion. For example, the system in this scenario requires that the probability of correct and incorrect decisions for the target object are 95% and 1% respectively, while these probabilities are 90% and 5% for the decoy object. From a cost-benefit analysis, it was determined that each sensor will be manufactured to produce at best a 65% - 70% probability of a correct decision and 7% - 10% probability of incorrect decision. Also, the maximum number of looks desired should be between 5 and 8.

Initial values are first chosen in the analysis. In this case, sensor one has probabilities of correct (T), correct (D), incorrect (T) and incorrect (D) of 0.650, 0.549, 0.070, and 0.098 respectively. Similarly, sensor two has values of 0.700, 0.647, 0.075, and 0.137. OR logic will be examined with the number of looks at five. The DDDB with these values is displayed in Figure 3.

Immediately, it can be seen that the results will not be satisfactory because the fused probabilities are just shy of the specifications and the non-decision probability has been driven down to 0 at five looks leaving no room for further gains. Hence, additional looks will not help. Likewise, fewer looks will degrade performance. Both of these conclusions can be seen by examining the "Plot Target" and "Plot Decoy" graphs. (See figures 4 and 5.)

One possibility, yet to be considered for these sensor inputs is the use of AND as opposed to OR logic. By examining the "Plot Target And/OR" and "Plot Decoy And/OR", it can be seen that AND logic shows increased fused correct probabilities for greater than five looks. The "Plot Decoy And/OR" graph is displayed in figure 6. By checking the AND logic on the DDDB, AND logic results can be more investigated further. An examination of the "Plot Decoy" graph, which can be found in figure 7, shows that a minimum of 6 looks will be needed, but unfortunately for 6 or more looks the fused probabilities for the target object do not meet the specifications. Hence, the basic sensor characteristics need to be tweaked. The probability of correct (T) decision will be increased to 0.6600.

With this new value, the "Plot Target And/OR" and "Plot Decoy And/OR" show that for five looks, OR logic is preferable but for 6 or more looks AND logic will provide better fused results. The "Plot Target And/OR" is shown in figure 8.

Unfortunately, an inspection of either the "Plot Target" or "Plot Decoy" (which is shown in figure 9) charts show that five looks will produce a probability of incorrect decision that is larger than acceptable. Hence, AND logic will be considered with six to eight looks. With six looks, the system requirements can be met.

Of course because this analysis is being done in the design phase, another useful fact is to know how many looks are required for fusion to be beneficial. An examination of the "Plot Target Fusion Benefits" and "Plot Decoy Fusion Benefits" shows that fusion benefits can be obtained in the range of three to eight looks. These plots are displayed in figures 10 and 11 respectively.

5. Concluding Comments

This work represents a continuing effort to further the application of fusion technologies by developing tools to aid in fusion system development. As a continuation of this effort, the same logic that was used to develop the theoretical foundations for determining fusion benefits for two sensors should be expanded to include three or more sensors and incorporated into FUSE. Furthermore, additional modules (similar to FUSE) covering other fusion modes should be added to increase the utility of the GIFTS architecture.

6. References

[1] Rao, N.S.V., "To fuse or not to fuse: fuser versus best classifier," *Proceedings of SPIE - Sensor Fusion: Architectures, Algorithms, and Applications II*, 3376: 25-34, April 1998.

[2] Dasarathy, B.V. Decision Fusion. IEEE Computer Society Press. 1994.

[3] Varshney, P.K. Distributed Detection and Data Fusion. Springer Verlag. New York. 1997.

[4] Dasarathy, B.V. and Townsend S.D., "GIFTS – A Guide to Intelligent Fusion Technology Selection," *Proceedings of the First International Conference of Multisource-Multisensor Information Fusion*, 65-72, July 1998.

[5] Dasarathy, B.V., "Fusion strategies for enhancing decision reliability in multisensor environments," *Optical Engineering*, 35 (3): 603-616, March 1996.

[6] Kruglinski, Shepherd, and Wingo. Programming Microsoft Visual C++ Fifth Edition. Microsoft Press. 1998.

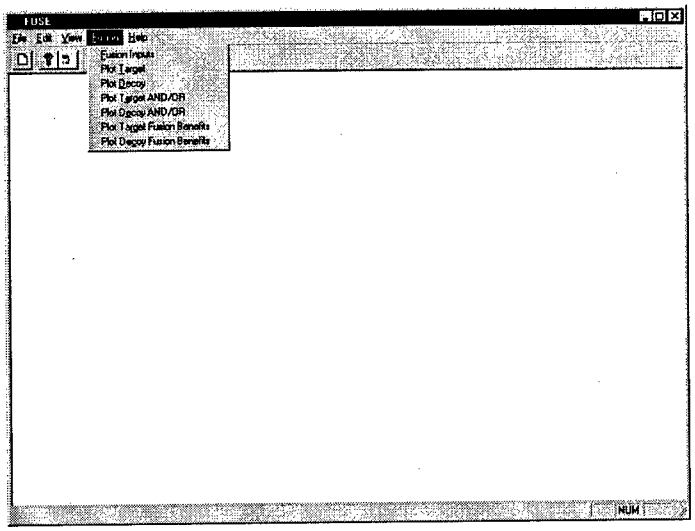


Figure 1: FUSE menu options

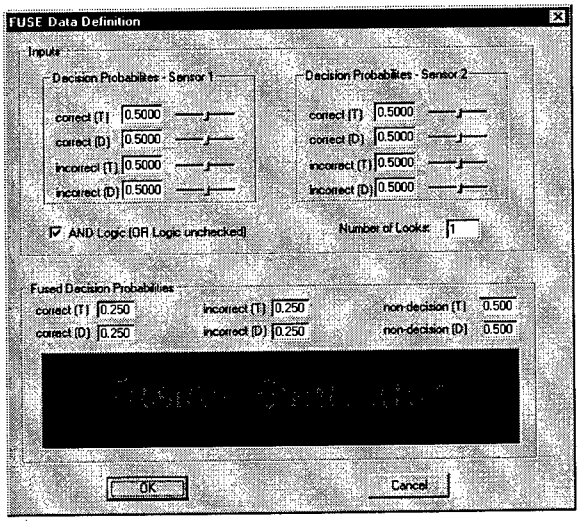


Figure 2: Data Definition Dialog Box with default values.

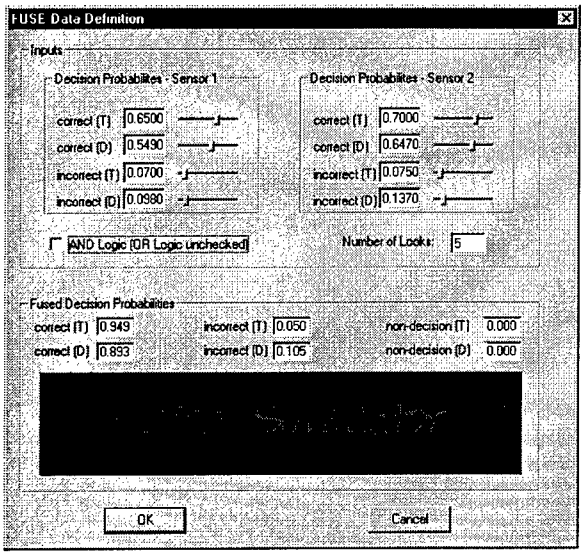


Figure 3: Initial analysis inputs

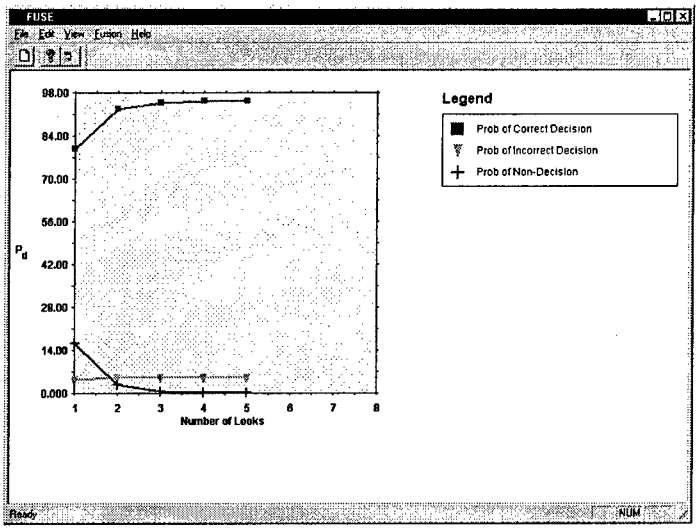


Figure 4: Plot of fused probabilities for the target object using initial inputs

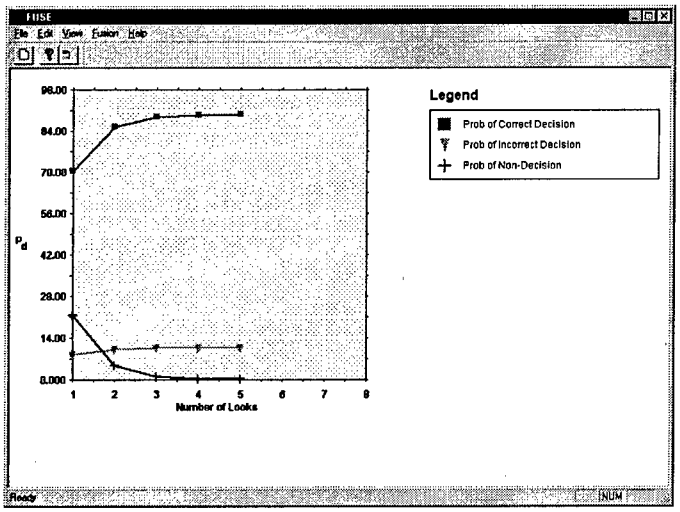


Figure 5: Plot of fused probabilities for the decoy object using initial inputs

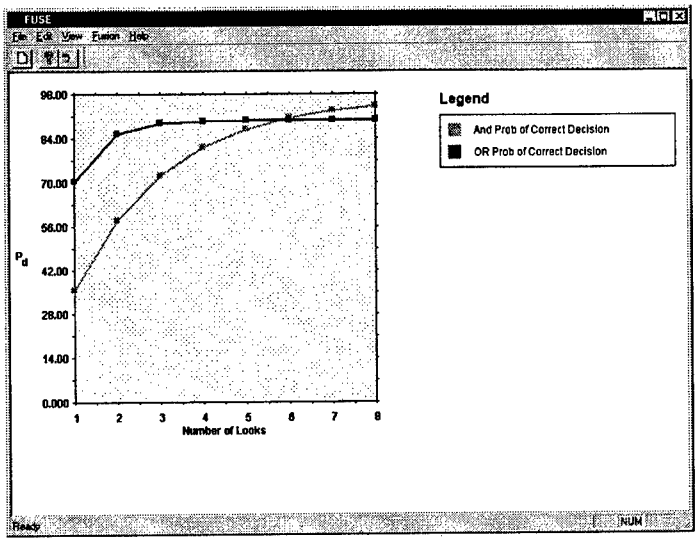


Figure 6: Comparison of probability of correct decision for decoy object using initial inputs under AND and OR logic

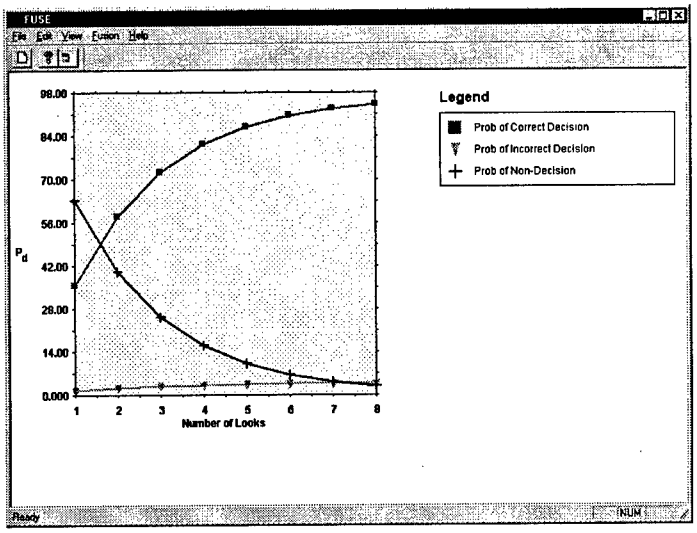


Figure 7: Plot of fused probabilities for decoy object using AND logic

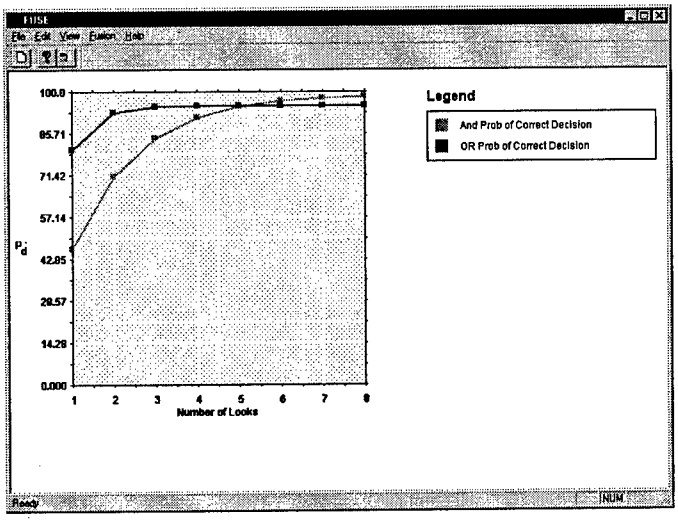


Figure 8: Comparison of probability of correct decision for target object using revised inputs under AND and OR logic

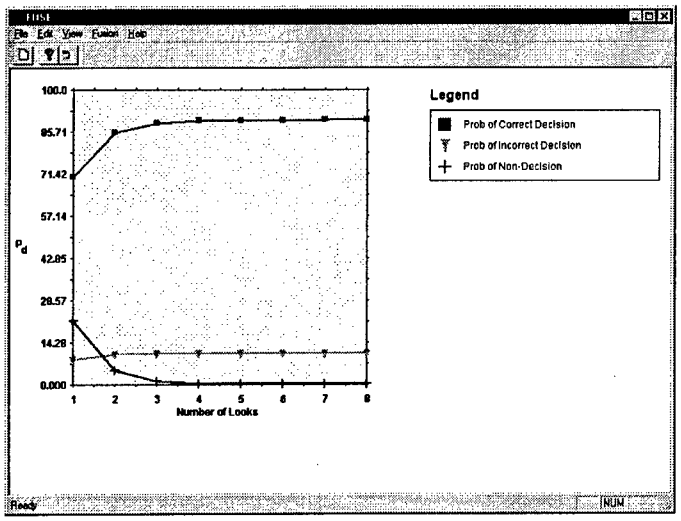


Figure 9: Plot of fused probabilities for decoy object using OR logic and revised inputs

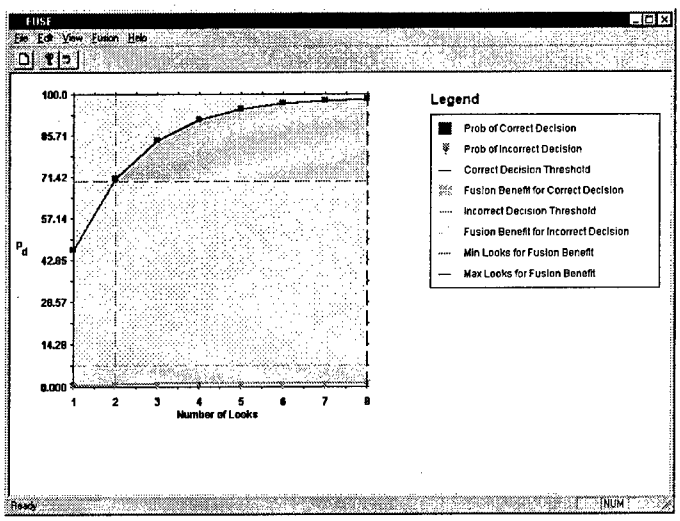


Figure 10: Fusion benefits for target object using revised inputs and AND logic

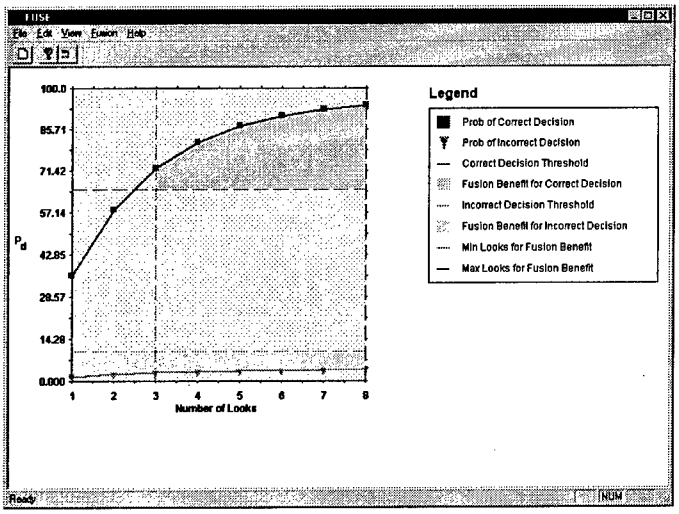


Figure 11: Fusion benefits for decoy object using revised inputs and AND logic

Optimal Decision Fusion Given Sensor Rules

Yunmin Zhu *
Department of Mathematics
Sichuan University
Chengdu, Sichuan 610064, P. R. China
ymzhu@scuu.edu.cn

X. Rong Li † (Corresponding Author)
University of New Orleans
New Orleans, LA 70148
Phone: 504-280-7416, Fax: 504-280-3950
xli@uno.edu

Abstract—Optimal distributed fusion assuming that sensor decision rules are given is considered. A general and computationally tractable optimal fusion rule is presented, which relies only on the joint conditional probability densities of all sensor observations and all local decision rules. It is valid for general decision systems with any sensor observations and sensor decision rules, regardless of their interdependence, and any network structure. It is also valid for M-ary Bayesian decision problems and binary problems under the Neyman-Pearson criterion. Local decision rules of a sensor that are optimal for the sensor itself are also presented, which take the form of a generalized likelihood ratio test. Numerical examples are given, which reveal some interesting phenomena.

Key words: distributed decision, optimal fusion, likelihood ratio test, sensor rule

1 Introduction

The multisensor distributed decision problem continues to attract much research interest in recent years, as evidenced by recent publications, e.g., [1–25]. A system with multiple sensors offers many advantages over one with a single sensor in terms of e.g., survivability, reliability, and robustness [12, 25].

Consider a decision system with a distributed sensor network. Each local sensor observes data and may receive messages from other sensors simultaneously. It locally fuses/compresses all its available information to a communicable message and transmitted it to a fusion center and/or other sensors. The fusion center makes a final decision using all received messages by some fusion rule. Communications are possibly permitted not only between the sensors and the fusion center, but also among sensors themselves.

The best distributed decision system uses a fusion rule and a set of local sensor decision rules that globally optimize the system's performance given a communication pattern of the system. The optimum fusion rule and the corresponding set of sensor rules are highly coupled in this framework. Early work along this line were reported in e.g., [1–3]. Some new ideas and results for a class of distributed decision systems that are quite general have been presented recently in [4–7], where finding the optimum fusion rule is reduced to determination of the sensor rules that yield optimum system performance.

On the other hand, a sensor decision rule is determined (optimally) in many practical situations based only upon all information available to it, regardless of the whole system's performance (without knowledge of the fusion rule). The fusion center only makes a final decision that is optimal subject to the fixed sensor rules. For example, in a decision process of a dynamic system, it is impossible sometimes for a decision maker to wait and make intermediate decisions until the final decision is known. Another example involves decision systems in a war situation. In order to enhance the survivability of the whole decision system, every local sensor must make a locally optimal decision upon all information available to it and then transmit the decision out. In so doing, even if the fusion center or some local sensors are destroyed, other local sensors can still make decisions. In short, while optimal fusion with already fixed sensor rules does not yield globally optimum performance, it is of strong practical significance and interest.

There are two classes of distributed decision systems in which local (sensor) rules are fixed, depending on whether *every* local rule is known to the fusion center (or the whole decision system) or not. We will call the first class "fusion with given local rules" and the second "fusion subject to fixed local rules." Note that in the first class either the fusion center or every sensor and the fusion center have complete knowledge of *every* local rules. The latter enables us to determine not only the optimal fusion rule with given local rules but also all sensor rules that are locally optimal. The first class is more often encountered in practice, but the second is not rare, either. A typical example of the second class is a decision system involving partners who do not want to share all intimate details of their own systems. Optimal fusion for the second class is clearly more difficult and the authors are not aware of any relevant result.

*Supported in part by The National Key Project and Natural Science Foundation of China.

†Supported in part by ONR via grant N00014-97-1-0570, NSF via Grant ECS-9734285, LEQSF via Grant (1996–99)-RD-A-32.

Previous work on distributed decision fusion with given local rules has been reported in [8–10]. Chair and Varshney [8] presented an optimal fusion rule as a linear combination of local decisions for distributed binary decision with independent local decisions, where the weight for each decision is a ratio of correct-decision probability to incorrect-decision probability. In [9], Drakopoulos and Lee extended the result of [8] to cases with dependent local decisions. They used correlation coefficients to express the joint conditional decision probabilities. Following a similar idea, Kam, Zhu and Gray in [10] normalized local decisions first and then employed the so-called *Bahadur-Lazarsfeld polynomial* and the normalized correlation coefficients to express the optimal fusion of correlated local decisions for distributed binary decision. In fact, these two expressions of the likelihood ratios are equivalent, because for zero-one binary random variables, conditional probabilities can be easily expressed as conditional expectations. It is hard to extend these results for coupled local decisions to more general cases, e.g., M-ary decision systems.

The main contribution of this paper lies in optimal fusion with given sensor rules for general decision systems, in particular those with dependent sensor decisions. There are three cases that lead to dependent sensor decisions: **I)** sensors with coupled observations but without mutual communications; **II)** sensors with independent observations but with communications among sensors; **III)** sensors with coupled observations and mutual communications. Note that sensor observations of a random signal are coupled even if observation errors are independent.

This paper presents a general and computationally tractable optimal decision fusion rule with given sensor rules in terms of the joint conditional probability densities and the sensor rules given. The optimal fusion rule is completely general in that it is valid for all sensor decisions, dependent or not, and all sensor network structures, with or without communications between any two sensors, provided that the joint conditional probability densities of all sensor observations and the sensor rules are known. It is also valid for both M-ary Bayesian decision problems and binary problems under the Neyman-Pearson criterion. Under the same optimality criterion as for the entire system, sensor decision rules are also presented that are optimal based on all information available to them individually, including their own observations and the received messages from other sensors. Thus, combining the optimal fusion rule and the locally optimal sensor rules, the optimal performance of a distributed decision system with given locally optimal sensor rules can be obtained. Finally, three numerical examples are given. They not only support the analytic results presented but also demonstrate some interesting properties of a distributed decision system with given sensor rules.

The paper is organized as follows. The problem is formulated in Sec. 2. In Sec. 3, we present, analyze, and show how to compute the optimal fusion rule for a general decision system given local sensor rules. Sec. 4 describes locally optimal local sensor decision rules. In Sec. 5, we extend the above results to several more general decision systems. Numerical examples are provided in Sec. 6. Finally, concluding remarks are given in Sec. 7.

2 Problem Formulation

Consider a distributed decision problem of M hypotheses $H_0, H_1, \ldots, H_{M-1}$ and l sensors with multi-dimensional observation data $\mathbf{y}_1, ..., \mathbf{y}_l$, where $\mathbf{y}_i \in R^{n_i}$. Each local sensor i makes a local M-ary decision u_i based upon the information available to it first and then transmits its decision out. If communications between sensors are allowed, the information available to a sensor includes not only its own observation but also messages received of some other sensor decisions. Finally, a fusion center (which may also observe data itself) makes a final M-ary decision F based upon all the received messages of local sensor decisions.

Obviously, this is a very general formulation of a distributed decision system. For example, it allows feedback among sensors. However, for notational simplicity, we consider a two-level Bayesian binary decision system first, which consists of only one level local sensors and a fusion center. Then, we show in Sec. 5 that all results presented for this simpler case can be extended to the more general decision systems described above.

At the fusion center, a final decision is made using a nonrandomized fusion rule F. Let $p(\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_l|H_1)$ and $p(\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_l|H_0)$ be the known conditional probability density functions (pdfs) of the observations under the two hypotheses, respectively, and let $(u_1, u_2, ..., u_l)$ be the observations of the fusion center. The Bayesian cost is

$$\begin{aligned} C(u_1, u_2, ..., u_l; F) = {} & c_{00}P_0P(F=0|H_0) \\ & +c_{01}P_1P(F=0|H_1) \\ & +c_{10}P_0P(F=1|H_0) \\ & +c_{11}P_1P(F=1|H_1) \end{aligned} \quad (1)$$

where c_{ij} are some suitable cost coefficients, P_i is the *prior* probability of hypothesis H_i, and $P(F=i|H_j)$ is the probability that the fusion center decides on hypothesis H_i while hypothesis H_j is true.

Substituting identity $P(F=1|H_j) = 1 - P(F=0|H_j)$ into (1) and simplifying yield

$$\begin{aligned} C(u_1, u_2, ..., u_l; F) = {} & P_0c_{10} + P_1c_{11} \\ & +P_1(c_{01}-c_{11})P(F=0|H_1) \\ & -P_0(c_{10}-c_{00})P(F=0|H_0) \end{aligned}$$

Denote the set for H_0 decision (a finite point set) by

$$\mathcal{F}_0 = \{(u_1, u_2, ..., u_l) : F = 0\} \quad (2)$$

Hence,

$$P_1(c_{01}-c_{11})P(F=0|H_1)-P_0(c_{10}-c_{00})P(F=0|H_0)$$

$$=\sum_{\mathcal{F}_0}[P_1(c_{01}-c_{11})P(u_1,u_2,...,u_l|H_1) \\ -P_0(c_{10}-c_{00})P(u_1,u_2,...,u_l|H_0)]$$

Using the above three equations, minimizing the cost function is equivalent to defining $\mathcal{F}_0$ as follows:

$$\mathcal{F}_0=\left\{(u_1,u_2,...,u_l):\frac{P(u_1,...,u_l|H_1)}{P(u_1,...,u_l|H_0)}<\frac{P_0(c_{10}-c_{00})}{P_1(c_{01}-c_{11})}\right\} \quad (3)$$

This is the optimal fusion rule mentioned in [8].

3 Computation of Likelihood Ratios

To have optimal fusion performance given the sensor rules, (3) indicates that all we need to do is computing the required ratio of likelihoods or conditional joint sensor decision probabilities. The contribution of [8] was in essence the simplification of the above likelihood ratio to a product of the ratios of two conditional decision probabilities of every sensor when sensor observations are independent and there are no communications among sensors. The sensor decision probabilities can be calculated easily from the given conditional probability densities $p(\mathbf{y}_1,\mathbf{y}_2,...,\mathbf{y}_l|H_1)$ and $p(\mathbf{y}_1,\mathbf{y}_2,...,\mathbf{y}_l|H_0)$. In practice, if the above two conditional probability densities of a sensor are not known, the two unknown conditional decision probabilities of the sensor may be replaced by their approximate (e.g., empirical) average values obtained from the historical data. The formula in [8] thus is still applicable but is, of course, no longer optimal. In [9, 10], although two alternative formulas were given, the computation of the desired correlation coefficients in the two alternatives is exactly just the computation of all possible conditional joint sensor decision probabilities.

In this paper, we point out that when the two conditional probability density functions of the observations, $p(\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_l|H_1)$ and $p(\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_l|H_0)$, as well as all sensor decision rules are known to the fusion center, we can compute all conditional joint sensor decision probabilities via the probabilities of subsets in the product space of all sensor observations, no matter how complicated the sensor decision rules are. This is computationally tractable. For example, in the case of no communication among sensors, the computational burden is the same as for the case with independent local decisions.

We state the above precisely as a proposition.

Proposition 3.1. A given set of all local nonrandomized decision rules defines a given 1-1 mapping between 2^l subsets $\mathcal{U}_1,\ldots,\mathcal{U}_{2^l}$ of the product space $R^{n_1}\times\cdots\times R^{n_l}$ of all sensor observations and points of the set of an l-tuple $(u_1,u_2,...,u_l)_i$ $(i\le 2^l)$ of (0,1) binary elements. These 2^l subsets $\mathcal{U}_1,\ldots,\mathcal{U}_{2^l}$ form a partition of the product space $R^{n_1}\times\cdots\times R^{n_l}$.

Proof. Given a set of all sensor decision rules, no matter how complicated they are, a mapping from the product space $R^{n_1}\times\cdots\times R^{n_l}$ of all sensor observations onto an l-tuple $(u_1,u_2,...,u_l)$ of (0,1) binary elements is defined. Let $\mathbf{u}_i=(u_1,u_2,...,u_l)_i$ be the ith point in the range of this mapping, that is, the ith possible value of this l-tuple or the ith possible set of sensor decisions. Clearly, there are 2^l distinct $\mathbf{u}_i$'s and corresponding to each $\mathbf{u}_i$ there is a unique subset $\mathcal{U}_i$ of the product space $R^{n_1}\times\cdots\times R^{n_l}$ (if more than one subset is mapped to the same $\mathbf{u}_i$, then $\mathcal{U}_i$ is their union). The proposition thus follows from the fact that these $\mathcal{U}_i$'s are disjoint and exhaustive. ■

In Sec. 4, we illustrate this proposition by some quite general examples. Using Proposition 3.1, the conditional joint sensor decision probabilities $P(u_1,u_2,...,u_l|H_0)$ and $P(u_1,u_2,...,u_l|H_1)$ can be computed easily in principle from the conditional pdfs of the sensor observations as follows:

$$P(u_1,...,u_l|H_1)=\int_{\mathcal{U}_i}p(\mathbf{y}_1,\mathbf{y}_2,...,\mathbf{y}_l|H_1)d\mathbf{y}_1d\mathbf{y}_2\cdots d\mathbf{y}_l$$

$$P(u_1,...,u_l|H_0)=\int_{\mathcal{U}_i}p(\mathbf{y}_1,\mathbf{y}_2,...,\mathbf{y}_l|H_0)d\mathbf{y}_1d\mathbf{y}_2\cdots d\mathbf{y}_l$$

where $\mathcal{U}_i=\{(\mathbf{y}_1,...,\mathbf{y}_l):(u_1,u_2,...,u_l)_i\}$.

To partition the whole observation space $R^{n_1}\times\cdots\times R^{n_1}$ into two decision regions, we need to compute 2^{l+1} such integrals of $\sum_{i=1}^{l}n_i$ folds because the optimal fusion (3) requires the computation of probabilities over all possible points $(u_1,u_2,...,u_l)_i$ under both hypotheses. If no communications exist among sensors, each sensor makes a decision based only on its own observation. Thus, each region $\mathcal{U}_i$ can be decomposed into a product of l regions of lower dimensions. This implies that the above $(\sum_{i=1}^{l}n_i)$-fold integrals can be reduced to the product of l integrals of $n_1,\ldots,n_l$ folds, respectively. Hence, although the local decisions may be coupled in this case through the interdependent sensor observations, the computation of the likelihood ratio is the same as for the case with independent local decisions.

4 Locally Optimal Sensor Decision Rules

All the above results assume that the local decision rules are known. Locally optimal sensor decision rules are presented in this section. By a "locally optimal sensor rule" of a local sensor we mean a sensor rule that is optimal using all information it received under the same optimality criterion as the one used for the whole decision system, that is, the Bayesian criterion with the same parameters P_0, P_1, and c_{ij}. Such a sensor rule is not to be confused with one based on the so-called "locally optimal test."

Clearly, in the case of no communications among sensors, regardless the dependence among sensor observations, the decision rule of a sensor relies only on its own observations. Thus, the locally optimal sensor rule is given by the following marginal-likelihood ratio test, for all $i \leq l$:

$$\frac{p(\mathbf{y}_i|H_1)}{p(\mathbf{y}_i|H_0)} \underset{u_i=0}{\overset{u_i=1}{\gtrless}} \frac{P_0(c_{10}-c_{00})}{P_1(c_{01}-c_{11})} \tag{4}$$

If a sensor receives some other sensors' decisions, along with knowledge of their decision rules, its locally optimal decision rule is more complicated but can be obtained as follows.

Suppose the ith sensor can receive j local decisions from other sensors along with knowledge of their decision rules. Without loss of generality, denote the received j local decisions by $(u_1, u_2, ..., u_j)$ and assume $i > j$ for the convenience of presentation. The general decision rule at this sensor is defined by the following mapping:

$$u_i(u_1, ..., u_j, \mathbf{y}_i) : \{0,1\}^j \times R^{n_i} \longmapsto \{0,1\} \tag{5}$$

To define this mapping, we need to determine the values of $u_i(\,)$ for every possible value of the j-tuple $(u_1, u_2, ..., u_j)$ and $\mathbf{y}_i$. As these j sensors may also receive local decisions from other sensors, each point of the j-tuple $(u_1, u_2, ..., u_j)$ of binary elements is mapped from a subset of $R^{n_1} \times \cdots \times R^{n_{i-1}} \times R^{n_{i+1}} \times \cdots \times R^{n_l}$. Since we consider nonrandomized decisions only, these 2^j subsets are disjoint. Denote them by $\{\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_{2^j}\}$. Since all sensor rules are known, we know exactly what every subset $\mathcal{U}_k$ is. Thus, similarly as for the case without communications, the locally optimal sensor rule at the ith sensor is given by

$$\frac{\int_{\mathcal{U}_k} p(\mathbf{y}_1, ..., \mathbf{y}_l|H_1) d\mathbf{y}_1 \cdots d\mathbf{y}_{i-1} d\mathbf{y}_{i+1} \cdots d\mathbf{y}_l}{\int_{\mathcal{U}_k} p(\mathbf{y}_1, ..., \mathbf{y}_l|H_0) d\mathbf{y}_1 \cdots d\mathbf{y}_{i-1} d\mathbf{y}_{i+1} \cdots d\mathbf{y}_l} \underset{u_i=0}{\overset{u_i=1}{\gtrless}} \frac{P_0(c_{10}-c_{00})}{P_1(c_{01}-c_{11})}, \quad \forall k \leq 2^j \tag{6}$$

Note that all the integrals in the above rule are functions of $\mathbf{y}_i$ and this rule consists of 2^j sub-rules corresponding to different values of $(u_1, u_2, ..., u_j)$ so that the mapping (5) is uniquely defined.

When there is no communication between the i sensor and any other sensor, $j = 0$ and thus the only partition $\mathcal{U}_1$ of the product space $R^{n_1} \times \cdots \times R^{n_{i-1}} \times R^{n_{i+1}} \times \cdots \times R^{n_l}$ is the product space itself. As such,

$$p(\mathbf{y}_i|H_i) = \int_{\mathcal{U}_1} p(\mathbf{y}_1, ..., \mathbf{y}_l|H_i) d\mathbf{y}_1 \cdots d\mathbf{y}_{i-1} d\mathbf{y}_{i+1} \cdots d\mathbf{y}_l$$

That is, rule (6) reduces to rule (4).

The above result can be extended to the more general case with feedback from the fusion center to the local sensors. Suppose that after the fusion center makes a final decision based upon all the received local decisions at the first stage, the fusion center communicates its decision to some local sensors.

Suppose that the ith sensor can receive the fusion center's decision at the first stage and j other local decisions, along with knowledge of their decision rules, at the second stage. Without loss of generality, denote the received fusion center's decision and j other local decisions by $(F^{(1)}, u_1^{(2)}, u_2^{(2)}, ..., u_j^{(2)})$. Note that other sensors may also receive $F^{(1)}$. Assume the two joint conditional probability densities $p(Y^{(1)}, \mathbf{y}_1^{(2)}, ..., \mathbf{y}_l^{(2)}|H_1)$ and $p(Y^{(1)}, \mathbf{y}_1^{(2)}, ..., \mathbf{y}_l^{(2)}|H_0)$ are known, where $Y^{(1)} \equiv (\mathbf{y}_1^{(1)}, ..., \mathbf{y}_l^{(1)})$.

Note that $F^{(1)}$ actually defines a partition $\{\mathcal{F}_0^{(1)}, \mathcal{F}_1^{(1)}\}$ of $R^{n_1} \times R^{n_2} \times \cdots \times R^{n_l}$. Note also the following analogy:

$$(u_1, u_2, ..., u_j) \leftrightarrow (F^{(1)}, u_1^{(2)}, u_2^{(2)}, ..., u_j^{(2)})$$

$$p(\mathbf{y}_1, ..., \mathbf{y}_l|H_i) \leftrightarrow p(Y^{(1)}, \mathbf{y}_1^{(2)}, ..., \mathbf{y}_l^{(2)}|H_i)$$

$$\mathcal{U}_k \leftrightarrow \{\mathcal{F}_0^{(1)} \times \mathcal{U}_k^{(2)}\} \cup \{\mathcal{F}_1^{(1)} \times \mathcal{U}_k^{(2)}\}$$

By an analysis similar to the one that led to (5)–(6), we can derive the locally optimal sensor i's rule at the second stage, given by

$$\frac{\int_{\{\mathcal{F}_0^{(1)} \times \mathcal{U}_k^{(2)}\}} p(Y^{(1)}, \mathbf{y}_1^{(2)}, ..., \mathbf{y}_l^{(2)}|H_1) dY^{(1)} d\mathbf{y}_1^{(2)} \cdots d\mathbf{y}_{i-1}^{(2)} d\mathbf{y}_{i+1}^{(2)} \cdots d\mathbf{y}_l^{(2)}}{\int_{\{\mathcal{F}_0^{(1)} \times \mathcal{U}_k^{(2)}\}} p(Y^{(1)}, \mathbf{y}_1^{(2)}, ..., \mathbf{y}_l^{(2)}|H_0) dY^{(1)} d\mathbf{y}_1^{(2)} \cdots d\mathbf{y}_{i-1}^{(2)} d\mathbf{y}_{i+1}^{(2)} \cdots d\mathbf{y}_l^{(2)}} \underset{u_i^{(2)}=0}{\overset{u_i^{(2)}=1}{\gtrless}} \frac{P_0(c_{10}-c_{00})}{P_1(c_{01}-c_{11})}, \quad \forall k \leq 2^j$$

$$\frac{\int_{\{\mathcal{F}_1^{(1)} \times \mathcal{U}_k^{(2)}\}} p(Y^{(1)}, \mathbf{y}_1^{(2)}, ..., \mathbf{y}_l^{(2)}|H_1) dY^{(1)} d\mathbf{y}_1^{(2)} \cdots d\mathbf{y}_{i-1}^{(2)} d\mathbf{y}_{i+1}^{(2)} \cdots d\mathbf{y}_l^{(2)}}{\int_{\{\mathcal{F}_1^{(1)} \times \mathcal{U}_k^{(2)}\}} p(Y^{(1)}, \mathbf{y}_1^{(2)}, ..., \mathbf{y}_l^{(2)}|H_0) dY^{(1)} d\mathbf{y}_1^{(2)} \cdots d\mathbf{y}_{i-1}^{(2)} d\mathbf{y}_{i+1}^{(2)} \cdots d\mathbf{y}_l^{(2)}} \underset{u_i^{(2)}=0}{\overset{u_i^{(2)}=1}{\gtrless}} \frac{P_0(c_{10}-c_{00})}{P_1(c_{01}-c_{11})}, \quad \forall k \leq 2^j$$

Clearly, the above integrals are all functions of $\mathbf{y}_i^{(2)}$ and can be called *generalized likelihood functions*. As such, the locally optimal sensor rules in the general setting still take the form of a likelihood ratio test.

When the sensor observations is a strictly stationary independent sequence, the above decision rule reduces to

$$\frac{\int_{\mathcal{U}_k^{(2)}} p(\mathbf{y}_1, ..., \mathbf{y}_l|H_1) d\mathbf{y}_1 \cdots d\mathbf{y}_{i-1} d\mathbf{y}_{i+1} \cdots d\mathbf{y}_l}{\int_{\mathcal{U}_k^{(2)}} p(\mathbf{y}_1, ..., \mathbf{y}_l|H_0) d\mathbf{y}_1 \cdots d\mathbf{y}_{i-1} d\mathbf{y}_{i+1} \cdots d\mathbf{y}_l} \underset{u_i^{(2)}=0}{\overset{u_i^{(2)}=1}{\gtrless}} \frac{P(\mathcal{F}_0^{(1)}|H_0) P_0(c_{10}-c_{00})}{P(\mathcal{F}_0^{(1)}|H_1) P_1(c_{01}-c_{11})}, \quad \forall k \leq 2^j$$

$$\frac{\int_{\mathcal{U}_k^{(2)}} p(\mathbf{y}_1, ..., \mathbf{y}_l|H_1) d\mathbf{y}_1 \cdots d\mathbf{y}_{i-1} d\mathbf{y}_{i+1} \cdots d\mathbf{y}_l}{\int_{\mathcal{U}_k^{(2)}} p(\mathbf{y}_1, ..., \mathbf{y}_l|H_0) d\mathbf{y}_1 \cdots d\mathbf{y}_{i-1} d\mathbf{y}_{i+1} \cdots d\mathbf{y}_l} \underset{u_i^{(2)}=0}{\overset{u_i^{(2)}=1}{\gtrless}} \frac{P(\mathcal{F}_1^{(1)}|H_0) P_0(c_{10}-c_{00})}{P(\mathcal{F}_1^{(1)}|H_1) P_1(c_{01}-c_{11})}, \quad \forall k \leq 2^j$$

5 Extensions to More General Systems

The results in Sec. 4 indicate that the locally optimal sensor rules as well as the optimal fusion rule given these local rules depend only on the conditional probability densities in a form well known as the likelihood ratio test. In view of this, the optimal fusion rules can be extended to a variety of very general distributed decision systems.

5.1 Extension to Sophisticated Network Structures

A multi-level decision system, such as a *tandem* or a *tree* network system, can be viewed as the above two-level decision system with possible communications among sensors and between sensors and the fusion center. Sensors at a higher level in the multi-level system may be treated by fictitious sensors that receive new messages at a new stage in the above system. This should be the case since a two-level system that allows communications between any two sensors and between any sensor and the fusion center is actually a system of a general structure. Note that the optimal fusion rule and the locally optimal sensor rules presented in the above sections are valid for this general system.

5.2 Extension to M-ary Decision Systems

The above results can be easily extended to an M-ary decision system because the optimal decision rule for a centralized M-ary decision problem can be reduced to a set of likelihood ratio tests (see, e.g., [26]).

For an M-ary decision system, the Bayesian cost in (1) can be extended to

$$C(u_1,u_2,...,u_l;F)=\sum_{i=0,j=0}^{M-1}c_{ij}P_jP(F=i|H_j)$$

$$=\sum_{i=0}^{M-1}\sum_{\mathcal{F}_i}\sum_{j=0}^{M-1}c_{ij}P_jP(u_1,...,u_l|H_j)$$

where each c_{ij} is some suitable cost coefficient; P_j is *a priori* probability of hypothesis H_j; and each $P(F=i|H_j)$ denotes the conditional probability that the fusion center decides on H_i while in fact H_j is true, $i,j=0,1,...,M-1$. Similarly, the optimal decision region $\mathcal{F}_i$ for H_i is defined as

$$\mathcal{F}_i=\{(u_1,...,u_l):$$
$$\sum_{j=0}^{M-1}c_{ij}P_jP(u_1,...,u_l|H_j)\le\sum_{j=0}^{M-1}c_{kj}P_jP(u_1,...,u_l|H_j),$$
$$\forall k\neq i\} \qquad (7)$$

where those points $(u_1,...,u_l)$ satisfying multiple decision regions $\mathcal{F}_i$ can be defined to belong to anyone of them.

5.3 Extension to Neyman-Pearson Decision Systems

For a distributed Neyman-Pearson decision system, the major task for its optimal decision rules is still the computation of the conditional joint sensor decision probabilities $P(u_1,u_2,...,u_l|H_i)$, $i=0,1$. The only thing that differs from the Bayesian decision in this case is that $P(u_1,u_2,...,u_l|H_i)$, $i=0,1$ are in general nonzero over the region

$$\Big\{(\mathbf{y}_1,...,\mathbf{y}_l):$$
$$\frac{\int_{\mathcal{U}_k}p(\mathbf{y}_1,...,\mathbf{y}_l|H_1)d\mathbf{y}_1\cdots d\mathbf{y}_{i-1}d\mathbf{y}_{i+1}\cdots d\mathbf{y}_l}{\int_{\mathcal{U}_k}p(\mathbf{y}_1,...,\mathbf{y}_l|H_0)d\mathbf{y}_1\cdots d\mathbf{y}_{i-1}d\mathbf{y}_{i+1}\cdots d\mathbf{y}_l}$$
$$=\frac{P_0(c_{10}-c_{00})}{P_1(c_{01}-c_{11})}\Big\}$$

An appropriate parameter λ ($0\le\lambda\le1$) for the probability of making H_1 decision while observation falls into the above region is required in order for the actual type I error (false-alarm) probability P_f to best approximate (but not exceed) its maximum allowable value (see, e.g., [9]).

6 Numerical Examples

In the following simulations, we consider distributed systems of 2 and 3 sensors, respectively, for detecting Gaussian signals in Gaussian noise.

6.1 Two-sensor Neyman-Pearson Decision System

The two hypotheses are

$$\begin{aligned}H_0:&\quad y_1=\nu_1,&\quad y_2=\nu_2\\H_1:&\quad y_1=s+\nu_1,&\quad y_2=s+\nu_2\end{aligned}$$

where the signal s and the two sensor-observation noises ν_1, and ν_2 are Gaussian and all mutually independent:

$$s\sim N(2,2),\ \ \nu_1\sim N(0,0.3),\ \ \nu_2\sim N(0,0.2).$$

Thus, the two conditional pdfs under H_0 and H_1, respectively, are

$$p(y_1,y_2|H_0)\sim N\left(\begin{pmatrix}0\\0\end{pmatrix},\begin{bmatrix}0.3&0\\0&0.2\end{bmatrix}\right)$$
$$p(y_1,y_2|H_1)\sim N\left(\begin{pmatrix}2\\2\end{pmatrix},\begin{bmatrix}2.3&2\\2&2.2\end{bmatrix}\right)$$

Example 6.1

Consider Neyman-Pearson detection with false-alarm probability $P_f\le0.092$. Table 1 gives the detection probabilities, false-alarm probabilities and the thresholds of the two-sensor centralized decision, single-sensor decisions, and two-sensor distributed decision with given two sensor decision rules, where the step size used for the discretized algorithm was 0.05.

It is observed that the distributed decision system outperforms the single sensor decision systems but of course

Table 1: Performance Comparison of N-P Systems

	P_f	P_d	λ
Centralized	0.0913	0.8805	0.375
Sensor 1	0.0919	0.8087	0.656625
Sensor 2	0.0919	0.8437	0.51
Distributed	0.0919	0.8584	0.65

is worse than the centralized decision system. Sensor 2 with a greater signal-to-noise ratio (SNR) performs better than sensor 1. In this numerical example, it turned out that randomized decision was not carrier out because $P(u_1, u_2|H_1) = \lambda P(u_1, u_2|H_0)$ case never happened.

6.2 Three-sensor Bayesian Decision System

It was set in all the simulations below for Bayesian decision systems that $c_{ij} = 1$ for $i \neq j$, $c_{ii} = 0$, $P_0 = 1/2$, $P_1 = P_2 = 1/4$. In this case, the Bayesian cost functional, denoted as P_e, is actually a weighted sum of *decision error probabilities*.

The hypotheses are

$$\begin{array}{lllll} H_0: & y_1 = \nu_1, & y_2 = \nu_2, & y_3 = \nu_3 \\ H_1: & y_1 = s_1 + \nu_1, & y_2 = s_1 + \nu_2, & y_3 = s_1 + \nu_3 \\ H_2: & y_1 = s_2 + \nu_1, & y_2 = s_2 + \nu_2, & y_3 = s_2 + \nu_3 \end{array}$$

where the two signals s_1, s_2 and the three sensor observation noises ν_1, ν_2 and ν_3 are all Gaussian and mutually independent:

$$\begin{array}{l} s_1 \sim N(2,3), \;\; s_2 \sim N(-2,3), \\ \nu_1 \sim N(0,3), \; \nu_2 \sim N(0,2), \;\; \nu_3 \sim N(0,1) \end{array}$$

Therefore, the three conditional pdfs under H_0, H_1 and H_2, respectively, are

$$p(y_1, y_2, y_3|H_0) \sim N\left(\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right)$$

$$p(y_1, y_2, y_3|H_1) \sim N\left(\begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}, \begin{bmatrix} 6 & 3 & 3 \\ 3 & 5 & 3 \\ 3 & 3 & 4 \end{bmatrix} \right)$$

$$p(y_1, y_2, y_3|H_2) \sim N\left(\begin{pmatrix} -2 \\ -2 \\ -2 \end{pmatrix}, \begin{bmatrix} 6 & 3 & 3 \\ 3 & 5 & 3 \\ 3 & 3 & 4 \end{bmatrix} \right)$$

Example 6.2

Consider a parallel Bayesian decision system with the above ternary hypotheses without communications among sensors. According to (4), the locally optimal fusion rule at each sensor can be derived. Table 2 gives the decision error probabilities of the centralized decision, single sensor decisions, and distributed decision with given sensor decision rules.

Again, the distributed decision system outperforms all single sensor decision systems but of course performs slightly worse than the centralized decision system. Among the three single sensor decisions, the greater the SNR of a local sensor is, the better the performance is.

Table 2: Performance Comparison of Bayesian Systems

	Centr.	Sensor 1	Senor 2	Senor 3	Distr.
P_e	0.2157	0.3642	0.3274	0.2645	0.2475

Example 6.3

Consider again the above three-sensor decision system, but with one extra communication channel from sensor i to sensor j, denoted as "Sensor i-j," $i, j = 1, 2, 3, i \neq j$, in addition to transmitting all local decisions to the fusion center.

Sensor decision rules can be obtained by (7). For example, for "Sensor 1-2," the three local decision rules (regions) for the sensor 1 are given by

$$\mathcal{H}_0^{(1)} = \left\{ \mathbf{y}_1 : \left\{ \begin{array}{l} \frac{1}{4}(p(\mathbf{y}_1|H_1) + p(\mathbf{y}_1|H_2)) \\ \leq \frac{1}{2}p(\mathbf{y}_1|H_0) + \frac{1}{4}p(\mathbf{y}_1|H_2), \\ \frac{1}{4}(p(\mathbf{y}_1|H_1) + p(\mathbf{y}_1|H_2)) \\ \leq \frac{1}{2}p(\mathbf{y}_1|H_0) + \frac{1}{4}p(\mathbf{y}_1|H_1) \end{array} \right\} \right\}$$

$$\mathcal{H}_1^{(1)} = \left\{ \mathbf{y}_1 : \left\{ \begin{array}{l} \frac{1}{2}p(\mathbf{y}_1|H_0) + \frac{1}{4}p(\mathbf{y}_1|H_2) \\ < \frac{1}{4}(p(\mathbf{y}_1|H_1) + p(\mathbf{y}_1|H_2)) \\ \frac{1}{2}p(\mathbf{y}_1|H_0) + \frac{1}{4}p(\mathbf{y}_1|H_2) \\ \leq \frac{1}{2}p(\mathbf{y}_1|H_0) + \frac{1}{4}p(\mathbf{y}_1|H_1) \end{array} \right\} \right\}$$

$$\mathcal{H}_2^{(1)} = \left\{ \mathbf{y}_1 : \left\{ \begin{array}{l} \frac{1}{2}p(\mathbf{y}_1|H_0) + \frac{1}{4}p(\mathbf{y}_1|H_1) \\ < \frac{1}{4}(p(\mathbf{y}_1|H_1) + p(\mathbf{y}_1|H_2)) \\ \frac{1}{2}p(\mathbf{y}_1|H_0) + \frac{1}{4}p(\mathbf{y}_1|H_1) \\ < \frac{1}{2}p(\mathbf{y}_1|H_0) + \frac{1}{4}p(\mathbf{y}_1|H_2) \end{array} \right\} \right\}$$

By an extension of (6) (see the likelihood ratio test given in [26] for a ternary decision system), the locally optimal sensor 2 decision rules are defined by the following 9 regions of $\mathbf{y}_2$, where $\mathcal{H}_i^{(2)}(\mathcal{H}_j^{(1)})$, $i, j = 1, 2, 3$, denotes the region for sensor 2 to decide on H_i while the received sensor 1's decision is H_j.

$$\mathcal{H}_0^{(2)}(\mathcal{H}_j^{(1)}) = \left\{ \mathbf{y}_2 : \left\{ \begin{array}{l} \frac{1}{4}(\int_{\mathcal{H}_j^{(1)}} p(\mathbf{y}_1, \mathbf{y}_2|H_1)d\mathbf{y}_1 \\ + \int_{\mathcal{H}_j^{(1)}} p(\mathbf{y}_1, \mathbf{y}_2|H_2)d\mathbf{y}_1) \\ \leq \frac{1}{2}\int_{\mathcal{H}_j^{(1)}} p(\mathbf{y}_1, \mathbf{y}_2|H_0)d\mathbf{y}_1 \\ + \frac{1}{4}\int_{\mathcal{H}_j^{(1)}} p(\mathbf{y}_1, \mathbf{y}_2|H_2)d\mathbf{y}_1, \\ \frac{1}{4}(\int_{\mathcal{H}_j^{(1)}} p(\mathbf{y}_1, \mathbf{y}_2|H_1)d\mathbf{y}_1 \\ + \int_{\mathcal{H}_j^{(1)}} p(\mathbf{y}_1, \mathbf{y}_2|H_2)d\mathbf{y}_1) \\ \leq \frac{1}{2}\int_{\mathcal{H}_j^{(1)}} p(\mathbf{y}_1, \mathbf{y}_2|H_0)d\mathbf{y}_1 \\ + \frac{1}{4}\int_{\mathcal{H}_j^{(1)}} p(\mathbf{y}_1, \mathbf{y}_2|H_1)d\mathbf{y}_1 \end{array} \right\} \right\}$$

$$\mathcal{H}_1^{(2)}(\mathcal{H}_j^{(1)}) = \left\{ \mathbf{y}_2 : \left\{ \begin{array}{l} \frac{1}{2}\int_{\mathcal{H}_j^{(1)}} p(\mathbf{y}_1,\mathbf{y}_2|H_0)d\mathbf{y}_1 \\ +\frac{1}{4}\int_{\mathcal{H}_j^{(1)}} p(\mathbf{y}_1,\mathbf{y}_2|H_2)d\mathbf{y}_1 \\ \le \frac{1}{2}\int_{\mathcal{H}_j^{(1)}} p(\mathbf{y}_1,\mathbf{y}_2|H_0)d\mathbf{y}_1 \\ +\frac{1}{4}\int_{\mathcal{H}_j^{(1)}} p(\mathbf{y}_1,\mathbf{y}_2|H_1)d\mathbf{y}_1, \\ \frac{1}{2}\int_{\mathcal{H}_j^{(1)}} p(\mathbf{y}_1,\mathbf{y}_2|H_0)d\mathbf{y}_1 \\ +\frac{1}{4}\int_{\mathcal{H}_j^{(1)}} p(\mathbf{y}_1,\mathbf{y}_2|H_2)d\mathbf{y}_1 \\ < \frac{1}{4}(\int_{\mathcal{H}_j^{(1)}} p(\mathbf{y}_1,\mathbf{y}_2|H_1)d\mathbf{y}_1 \\ +\int_{\mathcal{H}_j^{(1)}} p(\mathbf{y}_1,\mathbf{y}_2|H_2)d\mathbf{y}_1) \end{array} \right\} \right\}$$

$$\mathcal{H}_2^{(2)}(\mathcal{H}_j^{(1)}) = \left\{ \mathbf{y}_2 : \left\{ \begin{array}{l} \frac{1}{2}\int_{\mathcal{H}_j^{(1)}} p(\mathbf{y}_1,\mathbf{y}_2|H_0)d\mathbf{y}_1 \\ +\frac{1}{4}\int_{\mathcal{H}_j^{(1)}} p(\mathbf{y}_1,\mathbf{y}_2|H_1)d\mathbf{y}_1 \\ < \frac{1}{2}\int_{\mathcal{H}_j^{(1)}} p(\mathbf{y}_1,\mathbf{y}_2|H_0)d\mathbf{y}_1 \\ +\frac{1}{4}\int_{\mathcal{H}_j^{(1)}} p(\mathbf{y}_1,\mathbf{y}_2|H_2)d\mathbf{y}_1, \\ \frac{1}{2}\int_{\mathcal{H}_j^{(1)}} p(\mathbf{y}_1,\mathbf{y}_2|H_0)d\mathbf{y}_1 \\ +\frac{1}{4}\int_{\mathcal{H}_j^{(1)}} p(\mathbf{y}_1,\mathbf{y}_2|H_1)d\mathbf{y}_1 \\ < \frac{1}{4}(\int_{\mathcal{H}_j^{(1)}} p(\mathbf{y}_1,\mathbf{y}_2|H_1)d\mathbf{y}_1 \\ +\int_{\mathcal{H}_j^{(1)}} p(\mathbf{y}_1,\mathbf{y}_2|H_2)d\mathbf{y}_1) \end{array} \right\} \right\}$$

Table 3 gives the performances of the distributed decision fusion with given sensor rules for the systems with all possible Sensor i-j, respectively.

Table 3: Performance Comparison of Distributed Bayesian Decision Fusion with Different Single Additional Communications

Distr.	Sensor 1-2	Sensor 1-3	Sensor 2-3
P_e	0.2518	0.2433	0.2381
Distr.	Sensor 3-2	Sensor 3-1	Sensor 2-1
P_e	0.2441	0.2509	0.2577

Table 3 carries very interesting information. Comparing it with the results in Example 6.2, we have the following **observations**:

- *Fusion does improve performance*: All distributed decision systems still outperform the three single sensor decision systems. This makes sense since more information is used in the fused decision than in any single-sensor decision.
- *Communication direction matters*: The direction of communication affects the performance of the distributed decision system significantly. For two given sensors, communication from the one with a smaller SNR to the one having a greater SNR leads to better performance than the other way round. This is understandable from the following perspective. It can be seen from a comparison between sensor 1 (with three decision regions) and sensor 2 (with nine regions) that in terms of decision rules, a sensor receiving information from another sensor is equivalent to having a more refined partition of its observation space. Thus it can be expected that communication to a more reliable sensor will result in better performance than if the communication direction is reversed.
- *Communication does not necessarily improve performance*: Not all the distributed decision systems with communication between sensors outperform the corresponding distributed decision systems without communication between sensors. While communication between sensors with a greater SNR improves performance, communication between sensors with a smaller SNR degrades the performance. This is somewhat counter-intuitive at first glance. Note, however, that what we considered is distributed fusion with given sensor rules. Communication involving a less reliable sensor either forces it to make more refined decisions, leading to increased decision errors, or forces the other sensor to make its decision based on these less reliable decisions.

These observations give an insight into the problem of distributed decision with sensor-wise communications and provide guidelines for the design of communications between sensors in practice. Further analysis and more numerical examples will be reported in near future.

7 Conclusions

We have developed the optimal distributed decision fusion for general distributed systems in which local decision rules are given. An optimal fusion rule has been presented in a general and computationally tractable way based on the joint probability densities of all sensor observations conditioned on the hypotheses and all local sensor decision rules given. It is valid for any sensor observations and any given local decision rules (whether they are dependent or not) and any network structure (with or without communications between any two sensors). We have also shown that the decision rules of a sensor that are locally optimal — in the sense that all information available to it is used optimally — are of the form of a generalized likelihood ratio test. Numerical examples have been given. They provide not only additional support to the analytic results presented, but also useful information for the design of communications among sensors in practice.

References

[1] S. C. A. Thomopoulos, R. Viswanathan, and D. P. Bougoulias, "Optimal Decision Fusion in Multiple Sensor Systems," *IEEE Trans. Aerospace and Electronic Systems*, 23(5): 644–653, 1987.

[2] A. R. Reibman, and L. W. Nolte, "Optimal Detection and Performance of Distributed Sensor Sys-

tems," *IEEE Trans. Aerospace and Electronic Systems,* 23(1): 24–30, 1987.

[3] Z. B. Tang, K. R. Pattipati and D. K. Kleinman, "An Algorithm for Determining the Decision Threshold in a Distributed Detection Problem," *IEEE Trans. Systems, Man, and Cybernetics*, 21: 231–237, 1991.

[4] Y. M. Zhu, R. Blum, Z. Q. Luo and M. K. Wong, "Optimum Distributed Sensor Detectors for Dependent Observation Cases and Unexpected Properties of Fusion Rules," to appear as a regular paper in *IEEE Trans. Automatic Control*, 1999.

[5] Y. M. Zhu, "Optimum Fusion in Distributed Multisensor Network Decision Systems," *Proc. 14th World Congress of International Federation of Automatic Control*, Beijing, July 1999.

[6] Y.M. Zhu, C.Y. Liu and Y. Gan, "Multisensor Distributed Neyman-Pearson decision with Correlated sensor observations", *Proc. 1998 International Conf. on Information Fusion (FUSION'98)*, 1998, pp. 35–39.

[7] Y. M. Zhu, Y. Gan, K. S. Zhang and C. Y. Liu, "Multisensor Neyman-Pearson Type Sequential decision with Correlated sensor observations", *Proc. 1998 International Conf. on Information Fusion (FUSION'98)*, 1998, pp. 787–793.

[8] Z. Chair and P. K. Varshney, "Optimal Data Fusion In Multiple Sensor Detection Systems," *IEEE Trans. Aerospace and Electronic Systems*, 22(1): 98–101, 1986.

[9] E. Drakopoulos and C. C. Lee, "Optimum Multisensor Fusion of Correlated Local Decisions," *IEEE Trans. Aerospace and Electronic Systems*, 27(4): 593–606, 1991.

[10] M. Kam, Q. Zhu, and W. S. Gray, "Optimal Data Fusion of Correlated Local Decisions in Multiple Sensor Detection Systems," *IEEE Trans. Aerospace and Electronic Systems*, 28(3): 916–920, 1992.

[11] R. R. Tenney and N. R. Sandell, "Detection with Distributed Sensors", *IEEE Trans. Aerospace and Electronic Systems*, 17(4): 501–510, 1981.

[12] L. A. Klein, *Sensor and Data Fusion Concepts and Applications*, SPIE Press, 1993.

[13] S. Chaudhuri, A. Crandall and D. Reidy, "Multisensor Data Fusion for Mine Detection," *Proc. SPIE–Sensor Fusion III: 3–D Perception and Recognition*, vol. 1306, SPIE, Bellingham, Washington, pp. 187–204, 1990.

[14] J. Han, P. K. Varshney and V. C. Vannicola, "Distributed Detection of Moving Optical Objects," *Proc. SPIE–Sensor Fusion III: 3–D Perception and Recognition*, vol. 1306, SPIE, Bellingham, Washington, pp. 115–124, 1990.

[15] D. D. Freedman and P. A. Smyton, "Overview of Data Fusion Activities", *Proc. American Control Conf.*, San Francisco, June, 1993.

[16] R. C. Luo and M. G. Kay, "Multisensor Integration and Fusion in Intelligent Systems," *IEEE Trans. Systems, Man, and Cybernetics*, 19: 901–931, 1989.

[17] H. Delic, Papantoni-Kazakos and D. Kazakos, "Fundamental Structures and Asymptotic Performance Criteria in Decentralized Binary Hypothesis Testing," *IEEE Trans. Communications,* 43: 32–43, 1995.

[18] J. D. Papastavrou and M. Athans, "Distributed Detection by a Large Team of Sensors in Tandem," *IEEE Trans. Aerospace and Electronic Systems*, 28: 639–652, 1992.

[19] J. N. Tsitsiklis, "Decentralized Detection," *Advances in Statistical Signal Processing, Vol. 2: Signal Detection,* H. V. Poor and J. B. Thomas, Ed., Greenwich, CT: JAI Press, 1993.

[20] J. N. Tsitsiklis and M. Athans, "On the Complexity of Decentralized Decision Making and Detection Problems," *IEEE Trans. Automatic Control*, 30: 440–446, 1985.

[21] V. Venugopal, V. Veeravalli, T. Basar and H. V. Poor, "Decentralized Sequential Detection with a Fusion Center Performing the Sequential Test," *IEEE Trans. Information Theory*, 37: 433-442, 1993.

[22] V. Veeravalli, T. Basar and H. V. Poor, "Minimax Robust Decentralized Detection," *IEEE Trans. Information Theory*, 38: 35-40, 1994.

[23] R. Vismanathan and P. K. Varshney, "Distributed Detection with Multiple Sensors: Part I–Fundamentals," *Proceedings of the IEEE*, 85: 54–63, 1997.

[24] R. S. Blum, S. A. Kassam and H. V. Poor, "Distributed Detection with Multiple Sensors: Part II–Advanced Topics," *Proceedings of the IEEE*, 85: 64–79, 1997.

[25] P. K. Varshney, *Distributed Detection and Data Fusion*, New York: Springer-Verlag, 1997.

[26] H. L. Van Tree, *Detection, Estimation and Modulation Theory*, Vol. 1, New York: Wiley, 1968.

A Neural-Network Learning Method for Sequential Detection with Correlated Observations

Chengan Guo & Anthony Kuh *
Department of Electrical Engineering
University of Hawaii at Manoa
Honolulu, Hawaii 96822

Abstract *This paper proposes a neural-network sequential detection method for correlated observations drawn from an AR(1) model. We examine Wald's optimal sequential probability test (SPRT) when observation data are correlated. We focus on developing neural network methods to implement the SPRT procedure under the condition where parameters of the data model are unknown. In the paper, an optimal neural network model is designed to represent the ideal target functions – the conditional posterior probabilities of the observation data with which an SPRT procedure can be realized. A reinforcement learning method is then proposed to train the neural network using the temporal difference (TD) learning algorithm. Simulation results show that the proposed neural-network sequential detector can successfully learn the unknown ideal target functions and is able to give the same detection level performance as the parametric SPRT.*

Keywords: Neural networks, reinforcement learning, sequential detection, learning detection.

1 Introduction

Most research in sequential detection has been restricted to tests with independent observation data [1, 2, 3]. This is because, in general, the theory of sequential tests, such as the optimum property of the *sequential probability ratio test* (SPRT) [1] is limited when the observations are not independent. In engineering applications, however, there are many situations where it is natural to consider sequential tests with correlated observations. For example, in decentralized sequential detection problems, when there is feedback from fusion center to local detectors [4], the input data to the fusion center are highly coupled even if the original observations of each local detector are independent, identically distributed (i.i.d.) sequences. In digital communications, the received signal samples are often contaminated with intersymbol interference during channel transmission and hence are correlated although the original signals sent out from source transmitter might be independent. For nonindependent observation data, as pointed by Ghosh [2], it is an open problem whether the SPRT is really better than other detection procedures.

Note that the optimum SPRT property does not require the observation data $\{X_k\}$ to be necessarily independent [2, 3]. In fact, it only requires the log-likelihood ratios Z_n ($Z_n = \sum_{k=1}^{n} Y_k$) should be composed of independent components Y_k, i.e., $\{Y_k\}$ must be independent [2, 3]. When $\{X_k\}$ are i.i.d., $\{Y_k\}$ are also i.i.d.. Therefore, for the optimum property of the SPRT, the i.i.d. condition on $\{X_k\}$ is only a sufficient condition, but not a necessary one since $\{Y_k\}$ may turn out to be i.i.d. even when $\{X_k\}$ are not so.

In this paper we consider the sequential detection problem with correlated observation data $\{X_k\}$ which are a first order autoregressive ($AR(1)$) sequence (also a Markov process). We will show that, in this case, although

*This work was supported in part by NSF grant ECS9625557.

$\{X_k\}$ are correlated, $\{Y_k\}$ can be represented as independent variables and, more specifically, $\{Y_k; k \geq 2\}$ are i.i.d. sequence. Therefore, the optimum property is still achievable when these observation data are used in an SPRT procedure.

We then focus our discussion on corresponding neural-network approaches with this data model. Our objective is to develop a neural network method to realize the SPRT procedure for correlated data under the condition where accurate knowledge on the data model is not available.

The conventional SPRT method is a parametric approach where complete statistical knowledge about the observation data is given in advance. In practice, however, this statistical knowledge may not be available, or is only partially known. In order to overcome this difficulty, we study neural network based learning sequential detection methods that will not have access to statistical information, but will learn the sufficient statistics from observation data. Once the neural network is trained, it will operate as a sequential detector: as the neural network receives input observations sequentially until one of two output units of the network exceeds a certain specified boundary, a final decision will be made by accepting the hypothesis associated with that output neuron.

In previous work, the authors proposed a neural network sequential detection method [5] for independent observation data. In this paper we extend the work to the case of correlated observations. We first derive an equivalent SPRT algorithm that uses the conditional posterior probability functions to make sequential decisions instead of using the likelihood ratio function. The posterior probability functions are then chosen as the ideal target functions of the neural network sequential detector. A suitable neural network architecture is obtained through examining the ideal target functions and taking advantage of the Markov property. The proposed network architecture is shown to be optimal and there exists a set of ideal weights where the outputs of the neural network are equal to the ideal target functions. A reinforcement learning algorithm is then designed to train the neural network to approach the ideal target functions. The learning algorithm uses a binary label function as a reinforcement signal and uses the temporal difference (TD) learning method [6] as an updating rule. It learns the desired outputs from the labeled training data without needing statistical information about the data model. Simulation results conducted in the paper show that the proposed neural network sequential detector can successfully approach the ideal target functions and give detection performance that is comparable to the parametric SPRT.

2 Data Model and Optimality Analysis

Consider the following signal model:

$$X_k = \begin{cases} S_0 + N(k), & \text{if } X_k \in H_0 \\ S_1 + N(k), & \text{if } X_k \in H_1 \end{cases} \quad (1)$$

where S_i (i=0,1) are constants (signals), and $\{N(k)\}$ are first order autoregressive ($AR(1)$) noise sequence satisfying

$$N(k) = \rho N(k-1) + e_k, \quad (2)$$

where $\{e_k\}$ are i.i.d. zero mean Gaussian sequences, i.e., $e_k \sim N(0, \sigma^2)$, and parameter ρ satisfies $-1 < \rho < 1$.

Our goal is to develop neural-network sequential detection methods using the data $\{X_k\}$ drawn from this $AR(1)$ model with unknown parameters, S_0, S_1, ρ and σ. Before discussing the neural network method, a question arises about this detection problem: *whether the optimum property of the SPRT is achievable if these parameters are known?*

We will show that, although $\{X_k\}$ are correlated, the log-likelihood ratios, Z_n can be represented as the summation of an i.i.d. sequence, $\{Y_k; k \geq 2\}$ and an independent variable Y_1 with the given data model. Then according to [3], the SPRT property still holds.

Let $\mathbf{x}_n = (X_1, \cdots, X_n)$ be the vector of the

observation data. By using the Markov property of the AR process, we have the following expression for the log-likelihood ratio Z_n:

$$\begin{aligned} Z_n &= \log \frac{f_{\mathbf{X}}(\mathbf{x}_n \mid H_1)}{f_{\mathbf{X}}(\mathbf{x}_n \mid H_0)} \\ &= \log \frac{f(X_1 \mid H_1)}{f(X_1 \mid H_0)} + \sum_{k=2}^{n} \log \frac{f(X_k|X_{k-1}, H_1)}{f(X_k|X_{k-1}, H_0)} \\ &= \sum_{k=1}^{n} Y_k \end{aligned} \quad (3)$$

where

$$\begin{aligned} Y_1 &= \log \frac{f(X_1 \mid H_1)}{f(X_1 \mid H_0)}, \quad \textit{and} \\ Y_k &= \log \frac{f(X_k|X_{k-1}, H_1)}{f(X_k|X_{k-1}, H_0)}, \ \textit{for } k \geq 2 \end{aligned} \quad (4)$$

and $f(x|H_j)$ is the conditional density function of x given H_j.

From (1) and (2), we have that

$$\begin{aligned} X_k &= \begin{cases} S_0 + \rho N(k-1) + e_k \\ S_1 + \rho N(k-1) + e_k \end{cases} \\ &= \begin{cases} m_0 + \rho X_{k-1} + e_k, & \text{if } X_k \in H_0 \\ m_1 + \rho X_{k-1} + e_k, & \text{if } X_k \in H_1 \end{cases} \end{aligned} \quad (5)$$

where $m_i = (1-\rho)S_i$ for $i = 0, 1$.

Hence, for $k \geq 2$

$$\begin{aligned} Y_k &= \log \frac{f(X_k|X_{k-1}, H_1)}{f(X_k|X_{k-1}, H_0)} \\ &= \log \frac{g_e(X_k - m_1 - \rho X_{k-1})}{g_e(X_k - m_0 - \rho X_{k-1})} \\ &\triangleq \psi(e_k; m_0, m_1, \rho) \end{aligned} \quad (6)$$

where $g_e(\cdot)$ is the density function of e_k (Gaussian), and $\psi(e_k; m_0, m_1, \rho)$ is the function of e_k and parameters m_0, m_1 and ρ.

From (6), we see that $\{Y_k, k \geq 2\}$ is an i.i.d. sequence conditioned on each hypothesis since $\{e_k\}$ are i.i.d. .

We next show that Y_1 and $\{Y_k, k \geq 2\}$ are independent of each other. This can be done by showing that $N(1)$ and $\{e_k, k \geq 2\}$ are independent since $X_1 = S_i + N(1)$ and $Y_1 = \log \frac{f(X_1|H_1)}{f(X_1|H_0)} \triangleq \psi(N(1); S_0, S_1)$.

From [7], the $AR(1)$ sequence $\{N(k)\}$ can also be represented as a moving average process, i.e.,

$$N(k) = \rho N(k-1) + e_k = \sum_{j=0}^{\infty} \rho^j e_{k-j} \quad (7)$$

This says that $N(1) = \sum_{j=0}^{\infty} \rho^j e_{1-j}$ and hence $N(1)$ is independent of $\{e_k; k \geq 2\}$.

Then, by examining the optimality analysis conducted in [3] one can see that the average sample size of the SPRT procedure in this case is minimized among all other tests under the same error detection bounds.

3 Neural-Network Architecture Design

3.1 Ideal Target Functions and An Equivalent SPRT Algorithm

In order to design a neural-network model that can approach the optimal performance of the parametric SPRT, we first need to find a function that is both able to match the SPRT and also suitable for a neural network to learn. We call this function the *ideal target function.*

Let us consider the conditional posterior probability, $Q_j(\mathbf{x}_t)$ $(j = 0, 1)$ defined by

$$Q_j(\mathbf{x}_t) = P(H_j \ \textit{true} \mid \mathbf{x}_t), \ \textit{for } j = 0, 1 \quad (8)$$

where $\mathbf{x}_t = (X_1, \cdots, X_t)$ is the observation sequence up to time t.

We show that $Q_j(\mathbf{x}_t)$ is well suited to serve as the ideal target function: Let $\pi_i = P(H = H_i)$ $(i = 0, 1)$ be the prior hypothesis probabilities, $A = 1/(1 + \frac{\pi_1}{\pi_0} e^a)$ and $B = 1/(1 + \frac{\pi_0}{\pi_1} e^{-b})$ with a and b the Wald's SPRT detection boundaries [1]. We then have the following sequential detection algorithm using $Q_j(\mathbf{x}_t)$:

(1) *compute* $Q_0(\mathbf{x}_t)$ *and* $Q_1(\mathbf{x}_t)$;
(2) *accept* H_0 *and stop, if* $Q_0(\mathbf{x}_t) \geq A$;
(3) *accept* H_1 *and stop, if* $Q_1(\mathbf{x}_t) \geq B$; (9)
(4) *continue observing* $X_{t+1}, t \leftarrow t+1$, *and go to* (1), *otherwise.*

In [5] we proved that the above sequential detection algorithm is equivalent to the original SPRT method.

In Section 3.3 and Section 4, we will discuss how to learn the ideal target functions.

3.2 Neural-Network Sequential Detection Scheme

We are now ready to give a neural-network sequential detection scheme that is shown in Figure 1. In this scheme, as the observation data, X_t are fed into the neural network successively, the neural network gives two outputs, $y_0(t)$ and $y_1(t)$ that are used to learn the ideal target functions, $Q_0(\mathbf{x}_t)$ and $Q_1(\mathbf{x}_t)$ with

$$y_0(t) \simeq Q_0(\mathbf{x}_t) \text{ and } y_1(t) \simeq Q_1(\mathbf{x}_t). \quad (10)$$

After the training phase is finished, these two outputs are used to replace $Q_0(\mathbf{x}_t)$ and $Q_1(\mathbf{x}_t)$ to make sequential decisions by using the equivalent SPRT algorithm given in (9).

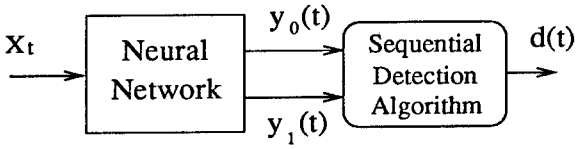


Figure 1: Block diagram of neural-network sequential detection scheme.

3.3 Neural-Network Architecture

In order to obtain a proper network architecture, let us further examine the log-likelihood ratio function Z_t under the data model of (1) and (2) for $t \geq 2$:

$$\begin{aligned} Z_t &= \log \frac{f(X_1 \mid H_1)}{f(X_1 \mid H_0)} + \sum_{k=2}^{t} \log \frac{f(X_k|X_{k-1}, H_1)}{f(X_k|X_{k-1}, H_0)} \\ &= U_0(\theta) + U_1(\theta)X_1 + V_0(\theta)(t-1) \\ &\quad + V_1(\theta)\sum_{k=2}^{t} X_k + V_2(\theta)\sum_{k=2}^{t} X_{k-1} \end{aligned} \quad (11)$$

where $\theta = (S_0, S_1, \rho, \sigma)$ is the vector of parameters of the data model defined by (1) and (2),

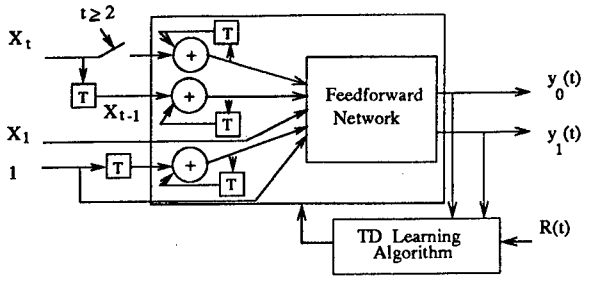


Figure 2: Neural network architecture for sequential detection for AR(1) data

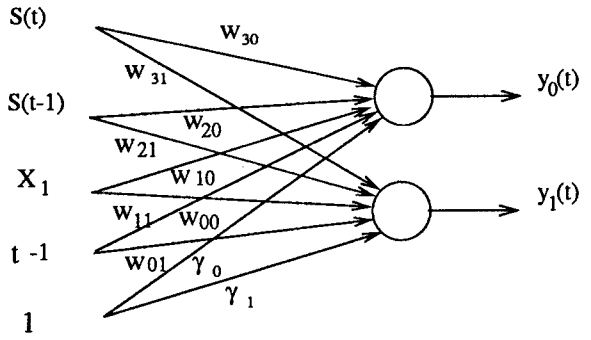


Figure 3: Realization of the feedforward network of Figure 2 ($S(t) = \sum_{k=1}^{t} X_k$)

$U_i(\theta)$ (i=0,1) and $V_j(\theta)$ (j=0,1,2) are functions of the parameter θ with

$$\begin{cases} U_0(\theta) &= (1-\rho^2)(S_0^2 - S_1^2)/(2\sigma^2) \\ U_1(\theta) &= (1-\rho^2)(S_1 - S_0)/\sigma^2 \\ V_0(\theta) &= (1-\rho)^2(S_0^2 - S_1^2)/(2\sigma^2) \\ V_1(\theta) &= (1-\rho)(S_1 - S_0)/\sigma^2 \\ V_2(\theta) &= \rho(1-\rho)(S_0 - S_1)/\sigma^2 \end{cases} \quad (12)$$

Above expressions are useful for our neural network method not for the purpose of parametric computation but because we can take $U_i(\theta)$ and $V_i(\theta)$ as unknown weights to learn by using neural-network techniques.

Based on these expressions, we construct a neural network architecture shown in Figure 2 and Figure 3. In Figure 2, the units marked with " T " represent a unit time delay operation and the input nodes marked with "+" are linear accumulative units (context units). The two output units of Figure 3 take linear weighted sums of their inputs and then pass these values through a sigmoidal nonlinearity

$\sigma(z) = 1/(1+e^{-z})$ to get outputs $y_0(t)$ and $y_1(t)$.

Similar to the i.i.d. case discussed in [5], for the $AR(1)$ data, we have Theorem 1 that discusses how the proposed neural network architecture matches the ideal target functions.

Theorem 1 : *For the neural network model shown in Figure 2 and Figure 3, when the input data X_t ($t = 1, 2, \cdots$) satisfy the data model defined by (1) and (2), there exists a set of ideal weights, $w^o_{i,j}$ and γ^o_j ($i = 0, 1, 2, 3$; $j = 0, 1$), that makes the outputs of the network, $y_j(t)$ equal to the ideal target functions $Q_j(\mathbf{x}_t)$, i.e.,*

$$y_0(t) = Q_0(\mathbf{x}_t) \text{ and } y_1(t) = Q_1(\mathbf{x}_t), \quad (13)$$

where

$$\begin{cases} \gamma^o_1 = U_1(\theta) - \log\frac{\pi_0}{\pi_1}, \ \gamma^o_0 = -\gamma^o_1 \\ w^o_{0,1} = V_0(\theta), \ w^o_{1,1} = U_1(\theta) \\ w^o_{2,1} = V_1(\theta), \ w^o_{3,1} = V_2(\theta) \\ w^o_{i,0} = -w^o_{i,1} \quad for \ i = 0, 1, 2, 3 \end{cases} \quad (14)$$

Theorem 1 can be proved by first representing the ideal target functions $Q_j(\mathbf{x}_t)$ as sigmoidal functions, i. e.,

$$Q_0(\mathbf{x}_t) = \frac{1}{1 + e^{Z_t - \log\frac{\pi_0}{\pi_1}}},$$

$$Q_1(\mathbf{x}_t) = \frac{1}{1 + e^{-(Z_t - \log\frac{\pi_0}{\pi_1})}}$$

and then using the neural network model and the ideal weights given by (14) to express the outputs of the neural network $y_j(t)$.

Theorem 1 implies that the neural-network architecture given in Figure 2 and 3 is an optimal neural-network model for the $AR(1)$ observations since it gives the ideal target functions $Q_j(\mathbf{x}_t)$ ($j = 0, 1$) whenever the ideal weight values have been learned.

4 Reinforcement Learning Algorithm

This section discusses the learning algorithm design for the neural network detector to learn the ideal target functions. The condition for this learning problem is that the parameters of observation model are unknown and the desired values for ideal target function $Q_j(\mathbf{x}_t)$ are unavailable. Under this condition, it is clear that commonly used supervised learning methods, such as the *error back propagation* (BP) algorithm [8], are not suitable. Note that the problem we are facing is similar to the reinforcement learning problem discussed in [5] except that the observation data here are not independent. We extend the reinforcement learning approach of [5] to this case:

- Assume that a binary label signal $D_j(\mathbf{x}_t)$ ($j = 0, 1$) is available for each training sequence $\mathbf{x}_n$ at the termination time of the sequence,

$$D_j(\mathbf{x}_t) = \begin{cases} unavailable, & t < n \\ 1, & \text{if } \mathbf{x}_t \in H_j \text{ and t=n} \\ 0, & \text{if } \mathbf{x}_t \notin H_j \text{ and t=n} \end{cases}$$

  Note that, in practice, we may have access to some labeled training data, but not to the statistical characteristics of the data. For example, in radar and sonar problems, for each set of the experimental data, we know whether or not there is a target in the experiment, but accurate statistical information about the data is not readily available.

- Construct the reinforcement signal using $D_j(\mathbf{x}_t)$

$$R(t) = \begin{cases} unavailable, & t < n \\ (D_0(\mathbf{x}_n), D_1(\mathbf{x}_n)), & t = n \end{cases}$$

- Take $R(t)$ as nominal targets to train the neural network using the TD learning algorithm [6]:

$$\Delta \mathbf{w}^{(m)}_j(t) = e_j(t) \sum_{k=1}^{t} \lambda^{t-k} \nabla_{\mathbf{w}} y_j(k)$$

$$\mathbf{w}^{(m+1)}_j = \mathbf{w}^{(m)}_j + \mu \sum_{t=1}^{n} \Delta \mathbf{w}^{(m)}_j(t)$$

where $E(t) = (e_0(t), e_1(t)) = Y(t+1) - Y(t)$ $(t = 1, 2, \cdots, n-1)$, $E(n) = R(n) - Y(n)$, $Y(t) = (y_0(t), y_1(t))$, $\mathbf{w}_j$ is the weight vector consisting of the weights connected to the output $y_j(t)$, λ is the parameter of TD learning algorithm, μ is the learning rate, and $\nabla_{\mathbf{w}} y_j(k)$ is the gradient of $y_j(k)$ with respect to $\mathbf{w}_j^{(m)}$.

With correlated observations, we will not have the same theoretical results to guarantee the convergence of the learning algorithm as we did in [5] with i.i.d. data. This is because the Law of Large Number (LLN) [9] used in the convergence analysis in [5] is not applicable to this case though LLN does hold for some dependent data under certain conditions.

But in practical applications, many learning algorithms, such as the BP algorithm [8], have frequently been used and been found to be successful when input data are not independent. From our simulation results, we see that the reinforcement learning algorithm works well for all the simulations conducted with the $AR(1)$ data.

5 Numerical Results

Simulation experiments are conducted for the neural-network sequential detection method using the correlated data drawn from AR(1) models with a variety of different parameters. Simulations are also conducted for the parametric SPRT method using the same data. In the experiments, the data parameters are given to the SPRT detectors. These values are not given to the neural-network reinforcement learning (NNRL) detectors as they must learn the ideal target functions from labeled training data without this information.

Table 1 shows the simulation results with three different $AR(1)$ models, corresponding to the cases of small, medium, and large correlation values. Four performance measures are used in the simulations to evaluate the correct detection rate $\overline{R}$, the average sample number $\overline{N}$, the false alarm probability $\overline{\alpha}$, and

Table 1: Detection performance of SPRT and NNRL method for AR(1) data

	AR(1)−1 $S_{0,1}$:−1.0,1.0 σ=1.0,ρ=0.1		AR(1)−2 $S_{0,1}$:−1.0,1.0 σ=1.0,ρ=0.5		AR(1)−3 $S_{0,1}$:−3.0,3.0 σ=1.0,ρ=0.9	
	SPRT	NNRL	SPRT	NNRL	SPRT	NNRL
$\overline{R}$	.9952	.9946	.9942	.9936	.9946	.9938
$\overline{N}$	3.52	3.42	8.69	8.71	11.83	11.59
$\overline{\alpha}$	.0053	.0059	.0066	.0091	.0069	.0083
$\overline{\beta}$	.0043	.0049	.0050	.0038	.0040	.0042

miss probability $\overline{\beta}$ for the two different detection methods. All the performance values are obtained by averaging over 10,000 testing sequences. From the simulation results, we see that the neural-network method has achieved the same performance level both in detection rate and in average sample number as the parametric SPRT.

The two error detection probability bounds were preset at $\alpha = \beta = 0.01$ in the simulations that were used to determine the detection boundaries with $A = B = 0.99$. Both the SPRT and NNRL methods have kept within these bounds in the experiments.

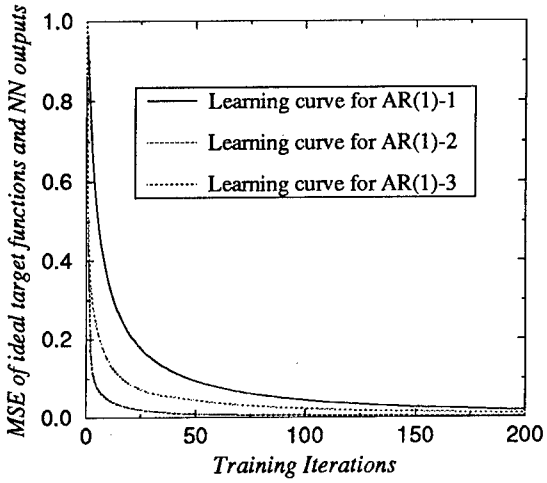


Figure 4: Learning curves of neural-network detector for AR(1) observation data.

Figure 4 shows the learning curves of the neural-network sequential detector under the three different AR(1) observation models during the training procedure. These learning curves are the square roots of the normalized

sample mean-squared errors (MSE) between the ideal target functions $Q_j(\mathbf{x}_n)$ and the neural network outputs $y_j(n)$ in different training stages. From these learning curves, one sees that the neural-network detector is able to learn the unknown ideal target functions with rapidly decreasing mean-squared error. The learning results also indicate that the proposed neural-network sequential detector can approach the optimal SPRT performance.

6 Summary

In this paper, we first studied the realizability of the optimal SPRT property with $AR(1)$ observation data. It is shown that the optimum property of the SPRT method still holds under correlated observations when the parameters of the data model are known.

We then studied using a neural network learning method to implement the sequential detection procedure for the case where statistical information about data is unknown. An optimal neural network architecture is obtained that can give the ideal target functions for realizing the SPRT procedure. Then a reinforcement learning method is used to train the neural network to learn the ideal target functions from a set of labeled training data with the TD learning algorithm. Simulation experiments conducted in the paper show that the proposed neural-network sequential detector can successfully learn the unknown ideal target functions and can give the same detection level performance as given by the parametric SPRT.

Further directions for this work include extending this work to other correlated data models and extending the work to decentralized sequential detection problems.

References

[1] A. Wald, *Sequential Analysis,* New York: John Wiley & Sons, 1947.

[2] B. K. Ghosh, " *Sequential Tests of Statistical Hypotheses,* " Addison-Wesley, Cambridge, MA, 1970.

[3] B. Eisenberg and B.K. Ghosh," Sequential probability ratio test," B.K.Ghosh and P.K.Sen Editors, *Handbook of Sequential Analysis,* Marcel Dekker, New York, 1991.

[4] V. V. Veeravalli, T. Basar and H. V. Poor, "Decentralized sequential detection with a fusion center performing the sequential test," *IEEE Trans. Inform. Theory,* vol. 39, pp. 433-442, March, 1993.

[5] C. Guo and A. Kuh, "Temporal difference learning applied to sequential detection," *IEEE Trans. Neural Networks,* vol. 8, No.2 pp. 278-287, March. 1997.

[6] R. S. Sutton, "Learning to predict by the methods of temporal differences, " *Machine Learning,* 3, pp. 9-44, 1988.

[7] G. E. P. Box and G. M. Jenkins, " *Time Series Analysis: Forecasting and Control,* " Holden-Day, San Francisco, 1970.

[8] D. E. Rumelhart, J.L.McClelland and the PDP Research Group, "*Parallel Distributed Processing,* " Volume 1, MIT Press, Cambridge, Mass., 1986.

[9] H. Stark and J. W. Woods, " *Probability, Random Processes, and Estimation Theory for Engineers,*" Second Edition, Prentice Hall, Englewood Cliff, 1994.

Adaptive Coordination and Integration of Local Decisions

Akira Namatame
Dept. of Computer Science
National Defense Academy
Yokosuka, 239-0811, JAPAN
E-mail: nama@cc.nda.ac.jp

Abstract

The fact that "Two heads are better than one" can bring many usually work together to solve many complex problems. Problem solving in a group, for instance, is a dynamic process and the action of each member must be coordinated in order to achieve globally consistent and good solutions. Cooperative works, if it is by a group of decision-makers, or by a team of experts, also require a proper consensus formation. Consensus formation is necessary, to ensure that the work is carried out in time and that no conflicts arise.

The decentralized decision is also an important consideration, and it should be deployed in a manner parallel to the consensus formation. In order to increase the efficiency of complex decision-makings through proper consensus formation, they have to exchange and integrate their knowledge. If they argue same problem, there happen to reach different conclusion especially when each decision-maker perceives different environments or they have different viewpoints or backgrounds.

In this paper, we aim at investigating the knowledge integration, the high level of information fusion, in the problem domains of the decentralized decision-making environments. In order to investigate this type of problem, we consider the team of decision-makers with their own preferences and the levels of adaptation. We then consider the situation where a team of decision-makers, as a whole, should make the group decision based on the consensus formation. Each decision-maker modifies his own preference by reflecting other decision-makers' preferences. Each decision-maker is modeled to adapt toward the group preference based on his own adaptation level.

We then provide the adaptive mechanism for knowledge integration and coordination. We also consider how heterogeneity of the group formed by different types of personality and sociality make effects on the speed of consensus formation and the quality of group decision-making.

Keyword: *adaptive coordination, decision-making with knowledge integration, preference fusion ,consensus formation,*

1 Introduction

Cooperative works, if it is by a team of engineers, or by a group of expert, for instance, require a proper coordination. Coordination is necessary, to ensure that the work is carried out in time and that no conflicts arise. Problem solving in a group is a dynamic process and the action of each member must be coordinated in order to achieve globally consistent and good solutions. The group decision is always an important consideration, and it should be deployed in a manner parallel to the coordination [3][8].

Adaptive behavior of each member in a group is an important issue when considering the problem of coordination [6]. The fact that "Two heads are better than one" can bring many peoples usually work together. However, in order to increase the efficiency of the co-works, they need coordination through the consensus formation.

The general impossible theorem by Arrow indicates that the impossibility of obtaining the overall preference of a group that consists of more than two peoples, each of them has his own preference satisfying the weak-ordering [7]. However, in a conference, for instance,we analyze the problem based on the individual preference, and express our own opinion considering what is most suitable for both an individual and a group. Each menber also considers the general group opinion and that consideration provides new base of each member to reconsider his own preference. Therefore, there is a great success of getting agreement in a group by self modifying his own preference by reflecting the position in a group [1][5].

If they argue same problem, there happen to reach different consensus as a group when the adaptiveness of each agent differs. How will these phenomena happen? To analyze this issue, we also

consider both the degree of personality and sociality. In this paper, we consider the problem of consensus formation in a group of adaptive agents. Adaptive agent has its own preference with different level of adaptation. Here, we define an adaptive agent as an autonomous agent with the self-modification capability of its own preference. We introduce the indexing method so that each agent's preference can be expressed by the proper index, and group preference is obtained by aggregating those indices. Each agent is modeled to adapt toward the group preference based on its own adaptation level. We show that we can avoid the problem of impossibility that encounter in group decision-making with the adaptive model of consensus formation. We also consider how heterogeneity of the group formed by different types of personality and sociality provides the effects on the speed of consensus formation and the quality of group decision-making.

We also illustrate the prototype model developed as the group decision-aid in order to allocate internet resources.

2 The indexing method of individual preference and their aggregation

In this research, we aim at investigating the emergent coordination in the problem domain of the group decision-making. In order to investigate this type of problem, we consider the group of adaptive agents with their own preferences and the adaptive capability. We then consider the situation where a group of those adaptive agents, as a whole, should make the group decision based on the consensus formation. The coordination among autonomous agents with their own preference, possess a difficult issue if the priority is given toward each agent's rationality, and they seek their own individual preference. Precursor to a group decision, they have to exchange and share their preference. Each agent is then required to modify its own preference by reflecting other agents' preferences.

When each agent's preference relation satisfies weak-ordering, it can be represented as linear-ordering. But Arrow's general impossibility theorem indicates that it is impossible to derive the group consensus by aggregating individual preference The condition of weak-ordering is quite strong condition and there exist many cases in where any two alternatives can not be compared and that violates one of the conditions of weak-ordering.

In this research, we consider semi-ordering which satisfies the condition of only *reflexivity* and *transitivity*. This excludes the condition of *connectedness* from weak-ordering. The tree expression can be used for representing the preference with semi-ordering. The advantage of the tree expression is that the judgement based on natural feeling, such as comparing each two alternatives can be also expressed easily.

We introduce the indexing method of the semi-ordering preference. Here, we consider a set of agents $G=\{A_i\ ;\ 1\le i\le n\}$ and a set of alternatives of $W=\{O_i\ ;\ 1\le i\le k\}$.

Step 1: Setting of local code

Each element of W is locally coded. That is, we provide the local code $C(O_i)$ to each alternative $O_i \in W$ as follows: the i-th element of $C(O_i)$ is 1 and set 0 for the other elements.

Step 2: Inheritance of upper index code

For the immediate descent alternatives O_j of O_i, i.e. if they satisfy the relation $O_j \prec O_i$, the index code of O_i is inherited to the index code of O_j as follows:

$$C(O_j)=C(O_i)\oplus C(O_j) \qquad (2.1)$$

where $\oplus$ represents the bit OR of each element of the two row vectors. For instance,we consider the following preference order,

$$R: O_1 \succ O_2 \succ O_4,\ O_1 \succ O_3, O_2 \succ O_5\ .$$

Then, this preference order can be indexed as follows:

$$C(O_1)=(1\,0\,0\,0\,0),\ C(O_2)=(1\,1\,0\,0\,0),$$
$$C(O_3)=(1\,0\,1\,0\,0),\ C(O_4)=(1\,1\,0\,1\,0),$$
$$C(O_5)=(1\,1\,0\,0\,1) \qquad (2.2)$$

We define group preference by aggregating the preference of each $A_i \in G$. When any two alternatives $O_\alpha, O_\beta \in W$ satisfies the following condition,

$$\sum_{i=1}^{n}\left\|C^{(i)}(O_\alpha)\right\| < \sum_{i=1}^{n}\left\|C^{(i)}(O_\beta)\right\| \qquad (2.3)$$

where $\|\ \|$ represents the sum of the

elements of each preference index which expressed bit vector. We then define the alternative O_α is preferable to O_β by the whole group G, and denote the group preference defined by (2.3) as $O_\alpha \succ_G O_\beta$. We have to remark that, from the definition of the index code in (2.1), the most preferred alternatives has the lowest norm.

3 The adaptive consensus formation

We define adaptation of each agent as the modification process of its own preference order. Each agent, considering a harmony with others, puts emphasis on both its own preference and the group preference. Since each agent adapts its own preference to the group preference, the group preference is also modified through each agent's adaptation. The group preference can be considered as evolutionary coordination by considering the modification of each agent preference as the adaptation process. The adaptive mechanism of each agent is defined as follows:

$$C_{t+1}^{(i)}(O) = \alpha_i(G_t(O) - C_t^{(i)}(O)) + C_t^{(i)}(O) \tag{3.1}$$

where $G_t(O)$ represent the index of group preference for the alternative $O \in W$, $C_t^{(i)}(O)$ represents the index of agent $A_i \in G$ preference at the step t, respectively. The factor α_i represents adaptive speed of agent $A_i \in G$ which takes the value between 0 to 1. If agent has the value that closes to 0, it insists its own preference. In other word, such an agent is said to be rational, and to be faithful to its own original preference. On the other hand, if it has the value that closes to 1, it can be said to be a sympathetic agent. The following property can be derived from the adaptation process of each individual defined in (3.1)

.

$$\textit{If} \quad G_t(O) \succ C_t^{(i)}(O)$$
$$\textit{then} \quad C_{t+1}^{(i)}(O) \succ C_t^{(i)}(O) \tag{3.2}$$

That is, if the individual preference index is lower by comparing with the group preference index, it increases its own preference index. On the other hand, if it is much higher, it may decreases it. From this property, the adaptive mechanism in (3.1) reflects the fact that each agent revises its own preference in accordance with the group preference.

4 Simulation Results

4.1 Consensus formation of a homogeneous group

We consider a group of agents of five $G = \{A_1, A_2, A_3, A_4, A_5\}$, each of them has its own preference over the five alternatives $W = \{O_1, O_2, O_3, O_4, O_5\}$ as follows:

$$\begin{aligned} R_1 &: O_1 \succ O_2 \succ O_3 \succ O_4 \succ O_5 \\ R_2 &: O_2 \succ O_3 \succ O_4 \succ O_5 \succ O_1 \\ R_3 &: O_3 \succ O_4 \succ O_5 \succ O_1 \succ O_2 \\ R_4 &: O_4 \succ O_5 \succ O_1 \succ O_2 \succ O_3 \\ R_5 &: O_5 \succ O_1 \succ O_2 \succ O_3 \succ O_4 \end{aligned} \tag{4.1}$$

It is impossible to derive consensus from such preference relations, and which is known as "paradox of voting"We define a homogeneous group as a group of agents with the same value of α, i.e., the speed of adaptation.

We also classify homogeneous group into the following two categories.

Case 1: Each agent has the high value of α (α=0.9)
These agents put high emphasis on the group preference.

Case 2: Each agent has the low value of α (α=0.1)
These agents, on the other hand, insist their own preference.

The simulation results of Case1 and Case2 are shown in Fig.1 and Fig.2. In Case1, such a group can make the consensus formation very quickly. In this case the consensus formation can be made even if they encounter the circulation order of preference, so called, 'paradox of voting'. Since each agent adapts its preference to group preference quickly, such a group can be also said to be harmonious.

On the other hand, in Case2, that group preference does not converge. In Case2, each agent is rational and it puts emphasis on its own preference. And, they can not derive consensus formation. This also implies that individual rationality is not always rational as a member of a group.

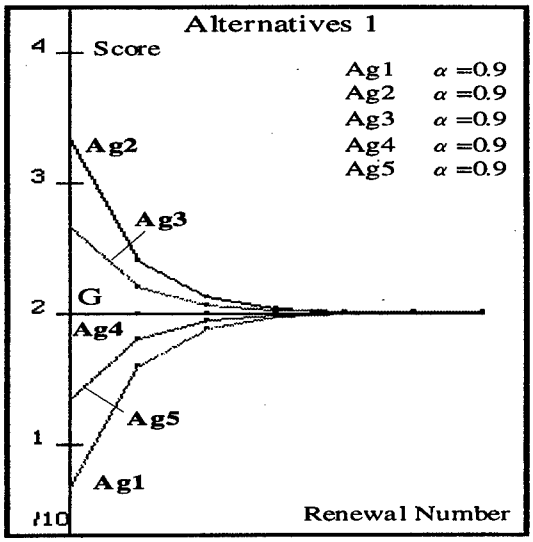


Fig 1 The simulation result (Case1)

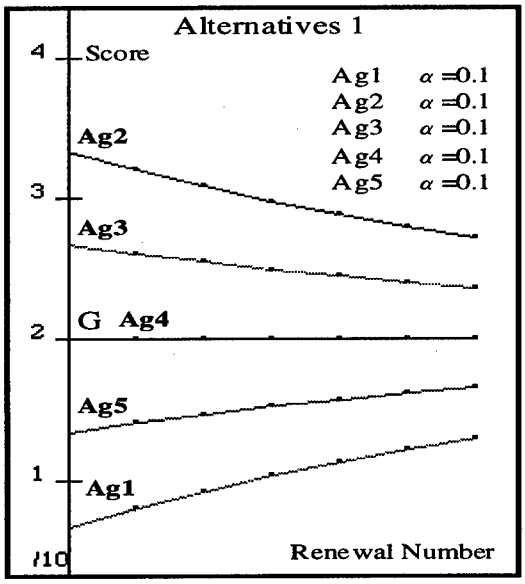


Fig.2 The simulation result (Case2)

4.2 Consensus formation of a heterogeneous group

We define a heterogeneous group in which each agent has the different adaptive speed α. In a heterogeneous group, different consensus formation can be emerged even if each agent has the same preference in the two groups. We consider the combination of adaptive speeds as follows.

Case 3:

$\alpha_1 = 0.9,\ \alpha_2 = 0.7,\ \alpha_3 = 0.5,\ \alpha_4 = 0.3,\ \alpha_5 = 0.1$

Case 4:

$\alpha_1 = 0.1,\ \alpha_2 = 0.3,\ \alpha_3 = 0.5,\ \alpha_4 = 0.7,\ \alpha_5 = 0.9$

We also consider the same preference order of each individual as given in (4.1). The simulation result of Case 3 is shown in Fig.3.

In heterogeneous group consists with many types of agents, in terms of their adaptive capability, it is also possible to get consensus formation. These simulation results show that heterogeneity of the group gives influences such as the promotion of deriving consensus and the emergence of new idea which nobody can expect.

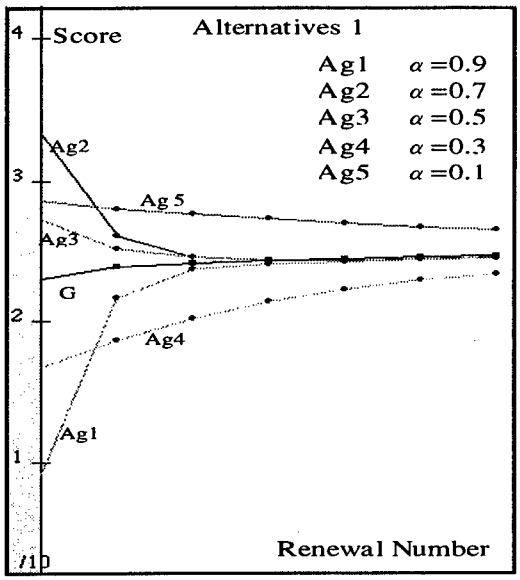


Fig.3 The simulation result (Case3)

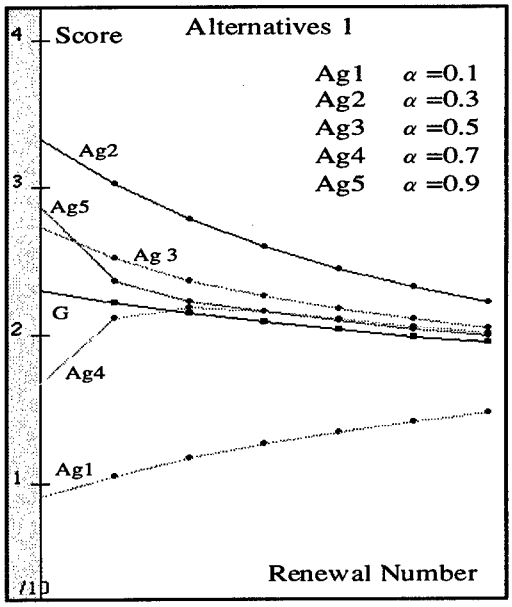


Fig.4 The simulation result (Case4)

5 Application

In this section, we illustrate the prototype model developed as the group decision-aid in order to allocate the internet resources. The concept of the prototype model is depicted in Fig.3. This model works with the following steps.

(1) Presentation the set of alternatives from the server to each user.

(2) Each user provides his preference and the level of adaptation to his user agent by GUI.

(3) The server aggregates all users' preferences, and the determine group preference. It is the presented to each user agent.

(4) The user agents adapts user's preference, and the modified preference is sent to the server.

(5) After several iterative process, each user agent presents the result of the negotiation as group decision to each user.

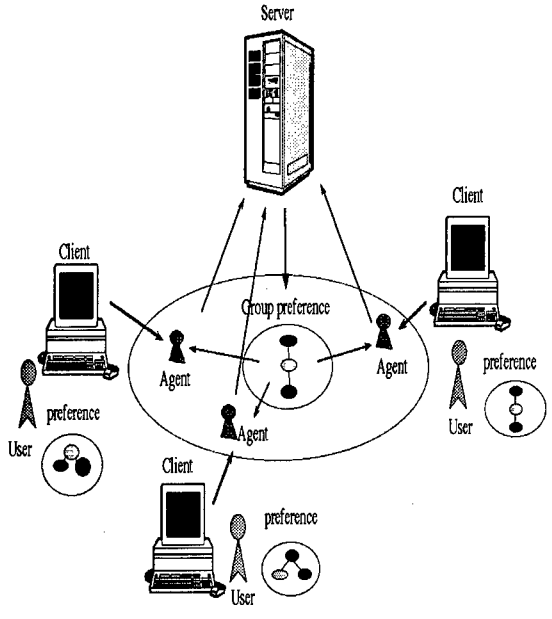


Fig.5 Application for the adaptive consensus formation on the internet.

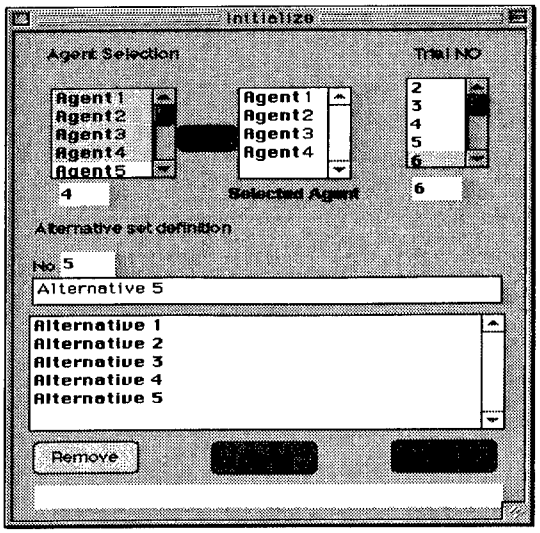


Fig.6(a) User interface and the screen image (1)

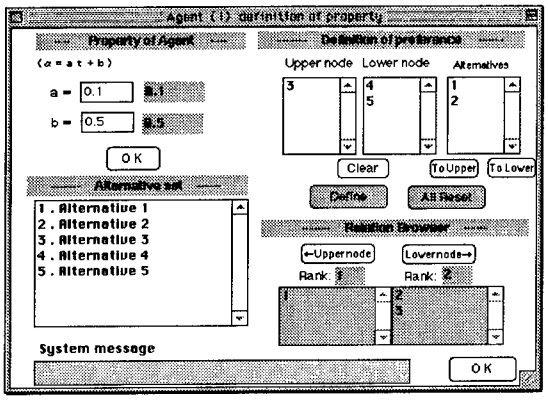


Fig.6(b) User interface and the screen image(2)

We have developed the group decision support system over the internet. Some of the screen images of the prototype model are shown in Fig.6.

We applied this prototype model in the domain of the resource allocation of the internet to many potential users. We could derive many useful properties of the evolutionary approach for the group decision-making. The concept of the internet resources allocation using this prototype model is shown in Fig7.

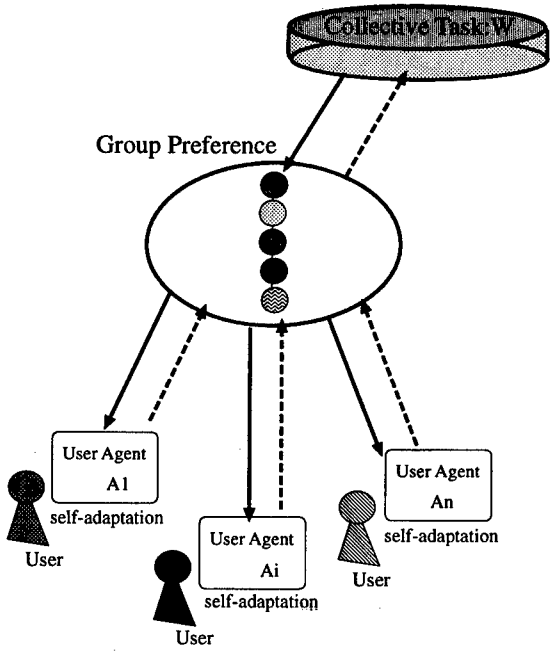


Fig.7 Application for the allocation of the internet resources

6 Conclusions

We considered the emergent adaptive behavior in the problem domain of group decision-making. We formulated the adaptive mechanism of each agent who has its own preference. Each agent, not only sticking to its own initial preference, it also cares other agents' preferences. And we also developed the prototype model in order to evaluate the evolutional approach for consensus formation. And we showed that the benefit what group can evolutional decision-making without they coming together on same place.

References

[1] Cohen M. (Eds):"*Organizational Learning*", Sage Press , (1996)

[2] Giampiero E.G. Beroggi, William A. Wallance, :"*A Decision Logic for Operational Risk Management*", Computational Organization Theory, IEA,pp.289-308, (1994)

[3] Grosz K. & Kraus S. : "*Collaborative plans for complex action*", Artificial Intelligence, Vol.86, pp195-244, (1996).

[4] Holland G. :"*Hidden Order*", Addison-Wisely Publishing, (1995)

[5] Kirman A. (Eds) :"*Learning and Rationality in Economics*", Blackewell,(1995)

[6] Rosenschein J.S., Zlotkin G.:*Rules of Encounter, Designing Conventions for Automated Negotiation among Computer*", The MIT Press, (1994).

[7] Winterfeldt D., Edwards W. :"*Decision Analysis And Behavioral Research*", Cambridge University Press,(1986)

[8] Zlotkin G. & Rosenschein J.S. : "*Mechanism design for automated negotiation*", Artificial Intelligence, Vol.86, pp195-244, (1996).

[3] Demazeau,Y. and Müller, J.P(eds) "Decentralized Artificial Intelligence",in *Decentralized AI*, Elsevier, pp.3-31, 1990.

[4] Detlof von Winterfeldt, Ward Edwards, "Decision Analysis And Behavioral Research", Cambridge University Press,1986

[5] Fudenberg Drew, "Game Theory", The MIT Press, 1991

[6] Grosz,K, and Kraus,S, "Collaborative plans for complex action, Artificial Intelligence, Vol.86, pp195-244 (1996).

[7] Howard Raiffa: "Decision Analysis, Introductory Lectures on Choices under Uncertainty", Addison-Wesley, July 1970.

[8] Gasser, L: "Computational Organization Research", ICMAS'95 Proceedings, pp414-415,1995

[9] Rosenschein,J.& Zlotkin,G: Rules of Encounter, Designing Conventions for Automated Negotiation among Computers,MIT Press (1994).

[10] Zlotkin, G, Rosenschein, J .S. and: Mechanism design for automated negotiation, Artificial Intelligence, Vol.86, pp195-244 (1996).

Session TA4
Formal Methods for Information Fusion
Chair: Mitch Kokar
Northeastern University, MA, USA

An Approach to Automation of Fusion Using Specware

Hongge Gao
Department of Electrical and Computer Engineering
Northeastern University
Boston, MA 02115, U.S.A.
hgao@ece.neu.edu

Mieczyslaw M. Kokar
Department of Electrical and Computer Engineering
Northeastern University
Boston, MA 02115, U.S.A.
kokar@coe.neu.edu

Jerzy Weyman
Department of Mathematics
Northeastern University
Boston, MA 02115, U.S.A.
weyman@neu.edu

Abstract *This paper introduces an approach to the automation of fusion using category theory based formal method. Category theory has a rich and rigorous mathematical language to manipulate complex systems via the relations between various kinds of objects. Specware, a category theory based formal development system, is used as a platform. This approach has the following advantages. First of all, fusion systems designed using this approach are easy to reuse, extend and maintain under evolution. Secondly, it provides a formal support to represent and state human knowledge explicitly. Finally, with the support of Specware, we can refine the formal specification into final executable code by stepwise refinement. Specware guarantees that the final executable code is provably correct.*

Keywords: information fusion, formal method, category theory.

1 Introduction

A number of information fusion architectures, models and techniques have been proposed, but there are few systematic approaches to representing, implementing and maintaining fusion systems. For instance, it is not possible to guarantee that a system designed using specific architecture actually implements the architecture and its requirements. It is also hard to reuse, extend or evolve such systems.

To deal with these kinds of issues, we use a formal method approach to the development of fusion systems. In our approach, we follow a software engineering paradigm, i.e., we first specify requirements for a fusion system, and then we develop code through progressive refinement of specifications. Our approach has the following advantages. First of all, fusion systems designed using this approach are easy to reuse, expand and manage under changes. Secondly, it enables us to represent and state human knowledge explicitly in the specification. Finally, with the support of Specware, we can refine the formal specification into final executable code by stepwise refinement. Specware guarantee that the final executable code is provably correct.

The main problem that we are addressing in this paper is how to guide the process of fusion of specifications into a final specification of the system. In our approach, we use Specware, a formal method tool that is based on category theory. Since category theory provides us with the rigorous mathematical language and rich operations to represent and manipulate complex information structures, we can assemble fusion system specifications modularly and incrementally by using category theory operators, such as colimit and interpretation, to the

particular basic specifications.

While Specware provides a formal specification language, the specification developer has to decide which specifications to combine and how. Our goal is to automate this process. Towards this goal, we investigated the Planware approach [2] to developing specifications. Planware is a process developed at Kestrel Institute for the domain of scheduling. We are investigating a similar approach to developing fusion systems. Basically, our process is as follows.

First, we develop a library of formal specifications of various goals, sensor theories, background theories and fusion theories. The relations among these theories are represented by specification morphisms.

Second, we assemble an abstract specification of a fusion system from the library developed in the first step. Fusion is then considered as an operation of combining those various specifications into a specification of a fusion system. In other words, fusion is an operation on these specifications. This differs from other views of fusion, where it is considered as an operation on data or decisions.

Third, we refine the abstract specification into a concrete specification using the information provided by the user. For any individual specification, we refine it to a more concrete specification via sequential composition of interpretations. For structured specification, we use parallel composition operator to automatically construct the refinement.

Finally, we generate code for the concrete specification.

The rest of the paper will explain how to implement the above procedure. Section 2 provides a brief introduction to category theory and Specware. In Section 3, we describe our approach to automation of fusion using Specware. A specific multisensor fusion example will be given in Section 4 followed by summary in Section 5.

2 Background

Category theory was originally invented as an abstract mathematical language to describe the passage from one type of mathematical structure to another. Specware supports the modular construction of formal specifications. It also supports stepwise and componentwise refinement of structured specification into executable code.

2.1 Category Theory

Category theory is an abstract language for describing external properties of objects. In category theory, an object is described by its interaction with all other objects via *morphisms*. This unique feature of abstract, high-level description makes category theory an ideal mathematical tool for the information fusion problem. In information fusion, we need to know the *relations* or *interactions* between disparate sources (*information*) in order to combine them together (*fusion*).

A good review of category theory related to fusion can be found in [3]. Interested reader can find more information about category theory in [8, 5, 1].

2.2 Specware

In this section, we will introduce Specware concepts which we used to automate the fusion process.

Specware is a system which aims to provide a formal support for specification and development of software [9]. The foundations of Specware are category theory, sheaf theory, algebraic specification and general logics. Using Specware, one can construct formal specifications modularly and refine such specifications into executable code through progressive refinement. The underlying basic concepts of Specware are described below.

A specification (*spec* or *theory*) is a collection of sorts, operations and axioms that define a theory via higher-order logic. An example of specification of image is shown in Figure 1.

```
spec IMAGE is
    sorts Image, E
    op make-image : Nat, Nat, E → Image
    op xsize : Image → Nat
    definition of xsize is
        axiom xsize(make-image(m,n,e)) = m
    end-definition

    other operations and axioms ...

end-spec
```

Figure 1: Image specification

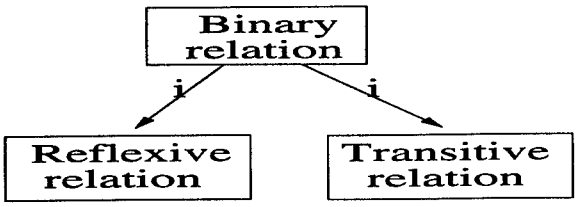


Figure 2: A specification diagram

Specifications can be developed from scratch or can be constructed from other specs via the specification-constructing operations – *import*, *translate* and *colimit*. Spec A has a copy of spec B if A imports B. Translate is similar to import except some elements of the copy of spec B are renamed according to the given renaming rules. The colimit operation takes a *specification diagram* as input and produces a specification called the colimit of the diagram.

A *specification morphism* is a mapping from *source specification* S to *target specification* T such that the signatures of the operations are translated compatibly and theorems are preserved.

A *specification diagram* (or simply *diagram*) is a directed multigraph whose nodes are labeled with specs and whose arcs are labeled with morphisms. So a diagram shows the relations between specifications. A diagram example is shown in Figure 2. In this diagram, both reflexive relation spec and transitive relation spec import binary relation spec. Therefore the morphisms are import-morphisms.

The definition of *interpretation* is as follows [10]: An *interpretation* $\rho : A \Rightarrow B$ from a specification A (called *domain* or *source*) to a specification B (called *codomain* or *target*) is a pair of morphisms $A \rightarrow A - as - B \leftarrow B$ with common codomain $A - as - B$ (called *mediating* specification or simply *mediator*), such that the morphism from B to $A - as - B$ is a definitional extension. Interpretation is also called *refinement.*

A morphism $S \rightarrow T$ is a *strict definitional extension* if it is injective and if every element of T which is outside the image of the morphism is either a defined sort or a defined operation. A *definitional extension* is a strict definitional extension optionally composed with a specification isomorphism.

Sequential (Vertical) composition of interpretations allows us to connect interpretations together so that we can refine a specification progressively. If ρ_1 and ρ_2 are two interpretations such that $\rho_1 : S \Rightarrow R$ and $\rho_2 : R \Rightarrow T$ then their sequential composition $\rho_1;\rho_2$ is an interpretation from S to T. That is, $\rho_1;\rho_2 : S \Rightarrow T$.

Parallel composition allows us to put interpretations together like specification-constructing operations allow us to put specifications together. Suppose we have interpretations for each of the specifications in a given diagram, we can compose them to obtain an interpretation whose domain is their colimit. The codomain of the composed interpretation will be the colimit of a diagram whose nodes are codomain of the component specification interpretations.

All the above concepts are expressed and implemented in *Slang*, Specware language. The specification example in Figure 1 is written in Slang.

3 Automation of Fusion

In the last section, we reviewed the basic concepts of Specware. Next, we are going to show our approach to automation of information fusion using Specware.

3.1 Information Fusion Problem

Basically, *information fusion* or *fusion* is defined as *the process of acquisition, filtering, correlation and integration of relevant information from various sources, like sensors, databases, knowledge bases and humans, into one representational format that is appropriate for deriving decisions regarding the interpretation of the information, system goals (like recognition, tracking or situation assessment), sensor management, or system control* [6]. A typical multisensor fusion scenario is depicted in Figure 3. In the figure, N sensors observe the region of interest, *World*, and send information to the fusion system. *Human* sends *queries* or *goals* to the fusion system. Based on the information received from sensors and queries or goals from the human, the fusion system computes the *solution* and returns an answer to the human (in the situations such as *detection*, *automatic target recognition*) or sends instructions to sensors (in the situation of *sensor management*).

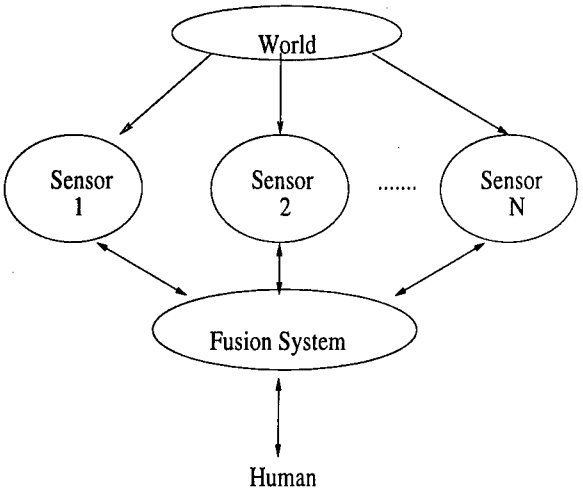


Figure 3: A multisensor fusion scenario

3.2 Abstract Fusion Specification

Based on the analysis of the fusion problem, we represent the abstract fusion problem as a structured specification as shown in Figure 4. Without loss of generality, we use two image sensors as an example. Fusion systems with more than two sensors or different types of sensors have similar structures.

Basically, there are two subdiagrams in this structure. The first subdiagram, COMB-SPECS-DIAGRAM, consists of specs IMAGE, SENSOR, SENSOR1, SENSOR2, ABS-PROB-THY 1 and ABS-PROB-THY 2 where ABS-PROB-THY 1 and ABS-PROB-THY 2 represent fusion problems, such as *detection*, expressed in terms of respective sensor theories. Specification COMB-THY is the colimit of this diagram. The second subdiagram, FUSION-DIAGRAM, consists of specs ABS-PROB, ABS-FUSION-THY and COMB-THY. ABS-REQ-THY, the abstract requirement specification of the multisensor fusion system, is the colimit of FUSION-DIAGRAM. Here we use ABS-PROB to glue ABS-FUSION-THY and COMB-THY together to get the final ABS-REQ-THY.

The above structure provides us with following advantages:

- Represent and state human knowledge explicitly in the specification. For instance, sensor theories, fusion theories are represented by specs SENSOR and ABS-FUSION-THY. Other human knowledge, from geometry to statistics, can also be represented as specifications.

- Reuse, extend and maintain fusion systems relatively easily. The structured specification gives us a clear roadmap of the whole fusion system. The relations between different parts are clearly represented by specification morphisms. This fusion system can be used repeatedly for a class of problems. Also building a larger

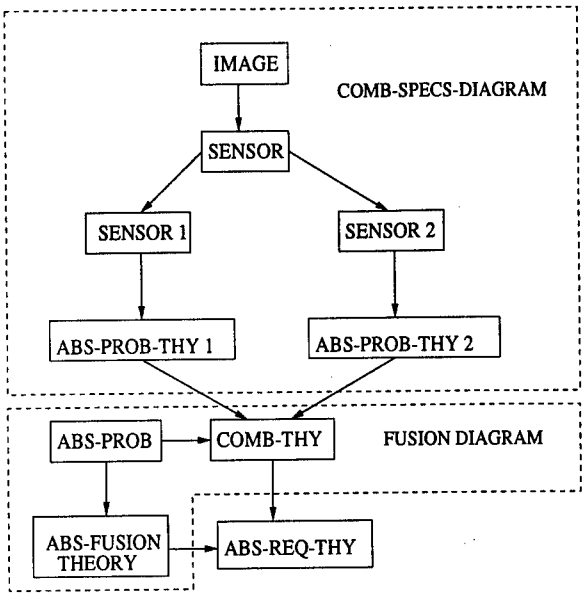


Figure 4: Abstract Fusion Specification Structure

system from this simple one is fairly easy.

- Refine the formal specification into final executable code by stepwise refinement with the support of Specware. As we discussed in Section 2, Specware supports both sequential and parallel compositions of refinement. Once we have such a structured specification, we can refine it incrementally into a *sufficiently refined* specification using sequential and parallel composition. The sufficiently refined specification is such a specification that every sort and operation of it are represented by the built-in *abstract target language* (ATL) [7]. ATL describes the constructs of the target language. Currently, Specware supports two kinds of target language, *C++* and *Lisp*.

Some of the component specifications are shown in Figure 5. Here we modeled the abstract sensor as a function. Similarly, both problem theories(ABS-PROB-THY1 and ABS-PROB-THY2) are also represented as functions. These theories will be refined into the concrete problem theories later on via user's selection. Finally, the fusion theory is represented as a function from outputs of two problem theories (*Q1* and *Q2*) to the final output *Q*. Notice we didn't specify Q1, Q2 and Q at this moment because they could be *Nat*, *Boolean* or any other sort in the refinement.

```
spec SENSOR is
    import IMAGE
    sort Sensor
    op sense : Image, Sensor → Image
end-spec

spec ABS-PROB-THY1 is
    import SENSOR1
    sort Q1
    op p1: Image1, Sensor1 → Q1
end-spec

spec ABS-PROB-THY2 is
    import SENSOR2
    sort Q2
    op p2: Image2, Sensor2 → Q2
end-spec

spec ABS-PROB is
    sorts Image1, Image2, Sensor1,
        Sensor2, Q1, Q2
    op p1 : Image1, Sensor1 → Q1
    op p2 : Image2, Sensor2 → Q2
end-spec

spec ABS-FUSION-THY is
    import ABS-PROB
    sort Q
    op fuse : Q1, Q2 → Q
end-spec
```

Figure 5: Fusion Specifications

3.3 Refining to a Domain-specific Specification

The user can refine the abstract fusion specification into a domain-specific specification by choosing concrete domain theories from the knowledge base. The basic procedure for refinement is as follows:

- First, choose two image sensors as SENSOR1 and SENSOR2 from a library of sensor theories.
- Second, choose a concrete fusion problem from a hierarchy of fusion problems. We will discuss details of this step in the next section.
- Then, choose a corresponding fusion theory.
- Finally, after refining each component theory of abstract specification to its corresponding concrete theory, compute the final domain-specific requirement theory via parallel composition of interpretations. The final domain-specific requirement theory is the refinement of ABS-REQ-THY.

we have analyzed the information fusion problem and introduced a Specware based approach to constructing fusion specification and refining it into final code. We showed that a formal system can be represented as a structured specification and one can develop such a fusion system formally through sequential and parallel composition of refinement.

4 A Fusion Example

In this section, we will show how to apply the refinement procedure described in last section to a particular fusion problem.

4.1 Subdomains of Fusion Problem

Goodman [4] described subdomains of data fusion as follows:

- *Sensor fusion.* In this kind of fusion, evidence from two or more sensors of similar type is combined in order to get more precise information which can not be deduced from each piece of evidence alone.
- *Multisource integration.* This type of fusion includes *Detection*, *Classification*(or *Automatic target recognition*), *Tracking* and *Correlation.*
- *Sensor management.* This refers to the process of adaptively allocating the dwells of each re-allocatable member of a suite of sensors.
- *Situation/threat assessment.* This is to provide an overall picture of the military significance of the data collected by the previous two kinds of fusion.
- *Response management.* This is the process of deciding upon courses of action which are appropriate response to current and evolving military situations.

Based on the above information, we can draw a hierarchy of fusion problems as below (Figure 6).

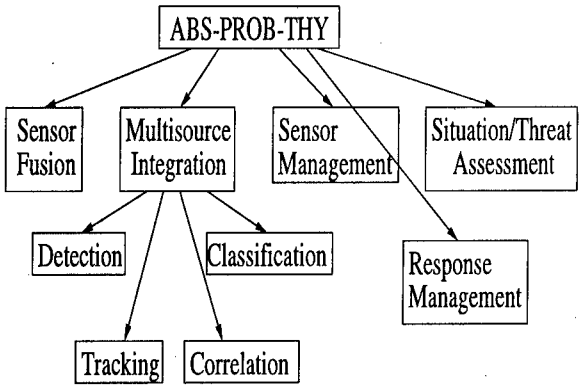


Figure 6: Hierarchy of fusion problems

Next, we will show how to refine an abstract fusion specification to a concrete specification based on this hierarchy.

```
spec DETECTION-THY is
    import LABELING
    op target? : Image, Sensor → Boolean
    definition of target? is
        axiom target?(img,s) ⇔
                gt(max-lab(img,s), zero)
    end-definition
end-spec
```

Figure 7: Detection Theory

4.2 Refining to a Detection Subdomain

To particularize the abstract fusion specification, the user has to select a concrete fusion subproblem from the hierarchy of fusion problem. The (composed) arrow from ABS-PROB-THY to the selected problem theory is the arrow used for refinement.

Suppose the user has chosen a detection theory (Figure 7). It imports LABELING theory which contains operation *max-lab*. The max-lab returns the maximum number of labels of the image being detected.

Then part of the refinement arrow is:

$$Q1 \mapsto Boolean$$

$$p1 \mapsto target?$$

$$Sensor1 \mapsto Sensor$$

Next, the user has to choose a corresponding fusion theory. In this case, the user should select DETECTION-FUSION-THY (see Figure 8).

So the refinement arrow is:

$$p1 \mapsto d1$$

$$p2 \mapsto d2$$

$$Q1 \mapsto Boolean$$

$$Q2 \mapsto Boolean$$

$$Q \mapsto Boolean$$

$$fuse \mapsto final - decision$$

```
spec DETECTION-FUSION-THY is
    sorts Image1, Image2, Sensor1, Sensor2
    const confidence1 : Nat
    const confidence2 : Nat
    op d1 : Image1, Sensor1 → Boolean
    op d2 : Image2, Sensor2 → Boolean
    op final-decision :
        Image1, Image2, Sensor1, Sensor2 → Boolean
    definition of final-decision is
        axiom d1(i1,s1) = d2(i2,s2) ⇒
            final-decision(i1,i2,s1,s2) = d1(i1,s1)
        axiom not(d1(i1,s1) = d2(i2,s2)) ∧
            gt(confidence1, confidence2) ⇒
            final-decision(i1,i2,s1,s2) = d1(i1,s1)
        ...
    end-definition
end-spec
```

Figure 8: Detection Fusion Theory

After having refined each component of the abstract fusion specification, the multisensor detection specification can be computed as described in the last section.

This section has shown how to refine the abstract fusion specification to a particular fusion problem theory. We have chosen a simple example and artificial theories to make the process clear. It needs careful, hard work to develop specifications for real applications.

5 Summary

We have shown in this paper a first step towards automation of information fusion using category theory based formal method. Specifically, we discussed the construction and refinement of fusion specifications using Specware. This approach enables us to represent human knowledge explicitly so that we can utilize this knowledge repeatedly and expand and manage it with ease in a changing environment. This formal approach also provides a way to produce provably-correct code through stepwise refine-

ment.

References

[1] M. Barr and C. Wells. *Category Theory for Computing Science.* Prentice-Hall, 1990.

[2] L. Blaine, et. al. *Planware - Domain-Specific Synthesis of High-Performance Schedulers.* Proceedings of the Thirteenth IEEE International Automated Software Engineering Conference, pp. 270-279, Honolulu, Hawaii, October 13-16, 1998.

[3] S. A. DeLoach and M. M. Kokar. *Category Theory Approach to Fusion of Wavelet-Based Features.* To appear in this conference.

[4] I. R. Goodman, R. P. S. Mahler and H. T. Nguyen. *Mathematics of Data Fusion.* Kluwer Academic Publishers, 1997.

[5] D. Rydeheard and R. M. Burstall. *Computational Category Theory.* Prentice-Hall, 1988.

[6] W. A. III. Sander. *Information Fusion.* In T. N. Dupuy, F. D. Margiotta, C. Johnson, J. Motley, and D. L. Bongard, editors, *International Military and Defense Encyclopedia*, Vol. 3, G-L, pp. 1259-1265. Brassey's, Inc., 1993.

[7] Kestrel Institute. *SPECWARE LANGUAGE MANUAL.* Specware 2.0.3, March 1998.

[8] Y. V. Srinivas. *Category Theory Definitions and Examples.* Technical Report, Department of Information and Computer Science, University of California, Irvine, February 1990. TR-90-14.

[9] Y. V. Srinivas and J. L. McDonald. *The Architecture of SPECWARE, a Formal Software Development System.* Technical report, Kestrel Institute, 1996.

[10] Y. V. Srinivas and R. Jüllig. *SPECWARE: Formal Support for Composing Software.* Proc. of Conference on Mathematics of Program Construction, Kloster Irsee, Germany, July 1995.

Category Theory Approach to Fusion of Wavelet-Based Features

Scott A. DeLoach
Air Force Institute of Technology
Department of Electrical and Computer Engineering
Wright-Patterson AFB, Ohio 45433
Scott.DeLoach@afit.af.mil

Mieczyslaw M. Kokar
Northeastern University
Department of Electrical and Computer Engineering
Boston, Massachusetts 02115
kokar@coe.neu.edu

Abstract

This paper discusses the application of category theory as a unifying concept for formally developed information fusion systems. Category theory is a mathematically sound technique used to capture the commonalties and relationships between objects. This feature makes category theory a very elegant language for describing information fusion systems and the information fusion process itself. After an initial overview of category theory, the paper investigates the application of category theory to a wavelet based multisensor target recognition system, the Automatic Multisensor Feature-based Recognition System (AMFRS), which was originally developed using formal methods.

1. Introduction

The goal of information fusion is to combine multiple pieces of data in a way so we can infer more information than what is contained in the individual pieces of data alone. This requires us to be able to determine how the individual pieces of data are related. It would also be nice if we could describe this relationship between data in a formal way so that we can automatically reason over the process without the use of unreliable and brittle heuristics. In this paper we present category theory as a unifying concept for formally defining information fusion systems. The goal of category theory is to define the relationships between objects in a category of related objects. Category theory also provides operators that allow us to reason over these relationships. In previous research we have shown category theory to be useful for defining relationships between object classes in object-oriented systems [1] and now we do the same for information fusion systems.

The first section of the paper is a tutorial on algebraic specifications and category theory. Next we describe a formally defined fusion system, the Automatic Multisensor Feature-based Recognition System (AMFRS), and describe how we could incorporate category theory constructs to provide a provably correct technique for implementing the system.

2. Theories and Specifications

The notation generally used to capture the formal definitions of systems is a *formal specification.* There are two types of formal specifications commonly used to describe the behavior of software: operational and definitional. An operational specification is a "recipe" for an implementation that satisfies the requirements while a definitional specification describes behavior by listing the properties that an implementation must posses. Definitional specifications have several advantages over operational specifications because they are generally shorter and clearer than operational specifications, easier to modularize and combine, and easier to reason about, which is the key reason they are used in automated systems.

It is recognized that creating correct, understandable formal specifications is difficult, if not impossible, without the use of some structuring technique or methodology. Algebraic theories provide the advantages of definitional specifications along with the desired structuring techniques. Algebraic theories are defined in terms of collections of values called *sorts*, *operations* defined over the sorts, and *axioms* defining the semantics of the sorts and operations. The structuring of algebraic theories is provided by category theory operations and provides an elegant way in which to combine smaller algebraic theories into larger, more complex theories.

Categories are an abstract mathematical construct consisting of *category objects* and *category arrows.* In general, category objects are the objects in the category of interest while category arrows define a mapping from the internal structure of one category object to another. In our research, the category objects of interest are algebraic specifications and the category arrows are specification morphisms. In this category, *Spec*, specification morphisms map the sorts and operations of one algebraic specification into the sorts and operations of a second algebraic specification such that the axioms in the first

specification become provable theorems in the second specification. Thus, in essence, a specification morphism defines an embedding of one specification into a second specification.

2.1. Algebraic Specification

In this section, we define the important aspects of algebraic specifications and how to combine them using category theory operations to create new, more complex specifications. As described above, category theory is an abstract mathematical theory used to describe the external structure of various mathematical systems. Before showing its use in relation to algebraic specifications, we give a formal definition [6].

Category. *A category C is comprised of*

- *a collection of things called C-objects;*
- *a collection of things called C-arrows;*
- *operations assigning to each C-arrow f a C-object dom f (the domain of f) and a C-object cod f (the "codomain" of f). If a = dom f and b = cod f this is displayed as*

$$f: a \to b \quad or \quad a \xrightarrow{f} b$$

- *an operation, "$\circ$", called composition, assigning to each pair $\langle g, f \rangle$ of C-arrows with dom g = cod f, a C-arrow $g \circ f$: dom f $\to$ cod g, the composite of f and g such that the Associative Law holds: Given the configuration*

$$a \xrightarrow{f} b \xrightarrow{g} c \xrightarrow{h} d$$

of C-objects and C-arrows, then

$$h \circ (g \circ f) \;=\; (h \circ g) \circ f.$$

- *an assignment to each C-object, b, a C-arrow, $id_{b:\, b \to b}$, called the identity arrow on b, such that the Identity Law holds: For any C-arrows f: a $\to$ b and g: b $\to$ c*

$$id_b \circ f = f \text{ and } \quad g \circ id_b = g.$$

2.1.1. The Category of Signatures

In algebraic specifications, the structure of a specification is defined in terms of an abstract collection of values, called *sorts* and operations over those sorts. This structure is called a *signature* [7]. A signature describes the structure that an implementation must have to satisfy the associated specification; however, a signature does not specify the semantics of the specification. The semantics are added later via axiomatic definitions.

Signature. *A signature $\Sigma = \langle S, \Omega \rangle$, consists of a set S of sorts and a set Ω of operation symbols defined over S. Associated with each operation symbol is a sequence of sorts called its rank. For example, $f: s_1, s_2, \ldots, s_n \to s$ indicates that f is the name of an n-ary function, taking arguments of sorts $s_1, s_2, \ldots, s_n$ and producing a result of sort s. A nullary operation symbol, c: $\to$ s, is called a constant of sort s.*

An example of a signature is shown in Figure 1. In the signature RING there is one sort, ANY, and five operations defined on the sort.

signature Ring is
sorts ANY
operations

plus	:	ANY × ANY	→ ANY
times	:	ANY × ANY	→ ANY
inv	:	ANY	→ ANY
zero	:		→ ANY
one	:		→ ANY

end

Figure 1. Ring Signature

In our research, signatures define the required structure for formally describing wavelet-based models. Signatures provide the ability to define the internal structure of a specification; however, they do not provide a method to reason about relationships between specifications. To create a theory of information fusion using algebraic specifications, operations to define relations between specifications must be available. There must be a well-defined theory about how specifications relate to one another.

As might be expected, signatures (as the "C-objects") with the correct "C-arrows" form a category that is of great interest in our research. For signatures, the C-arrows are called *signature morphisms* [7]. Signatures and their associated signature morphisms form the category, *Sign*.

Signature Morphism. *Given two signatures $\Sigma = \langle S, \Omega \rangle$ and $\Sigma' = \langle S', \Omega' \rangle$, a signature morphism $\sigma : \Sigma \to \Sigma'$ is a pair of functions $\langle \sigma_S : S \to S', \sigma_\Omega : \Omega \to \Omega' \rangle$, mapping sorts to sorts and operations to operations such that the sort map is compatible with the ranks of the operations, i.e., for all operation symbols $f: s_1, s_2, \ldots, s_n \to s$ in Ω, the operation symbol $\sigma_\Omega(f): \sigma_S(s_1), \sigma_S(s_2), \ldots, \sigma_S(s_n) \to \sigma_S(s)$ is in Ω'. The composition of two signature morphisms, obtained by composing the functions comprising the signature morphisms, is also a signature morphism. The identity signature morphism on a signature maps each sort and each operation onto itself. Signatures and signature morphisms form a category,* ***Sign****, where the signatures are the C-objects and signature morphisms are the C-arrows.*

Given the signatures RING from Figure 1 and RINGINT from Figure 2, a signature morphism σ : RING $\to$ RINGINT, is shown in Figure 3. As required by the definition of a signature morphism, σ consists of two functions, σ_S and σ_Ω as shown. σ_S maps the sort ANY to Integer while σ_Ω maps each operation to an operation with a compatible rank.

Signature morphisms map sorts and operations from one signature into another and allow the restriction of sorts as well as the restriction of the domain and

range of operations. However, to build up more complex signatures, introduction of new sorts and operations into a signature is required. This is accomplished via a signature *extension.*

```
Spec RingInt is
sorts Integer
operations
    +  : Integer × Integer   → Integer
    ×  : Integer × Integer   → Integer
    -  : Integer             → Integer
    0  :                     → Integer
    1  :                     → Integer
end
```

Figure 2. Integer Ring Signature

$$\sigma_S = \{\text{ANY} \mapsto \text{Integer}\}$$

$$\sigma_\Omega = \{\text{plus} \mapsto +, \text{times} \mapsto \times, \text{inv} \mapsto -, \text{zero} \mapsto 0, \text{one} \mapsto 1\}$$

Figure 3. Signature Morphism

Extension. *A signature $\Sigma_2 = \langle S_2, \Omega_2 \rangle$ extends a signature $\Sigma_1 = \langle S_1, \Omega_1 \rangle$ if $S_1 \subseteq S_2$ and $\Omega_1 \subseteq \Omega_2$.*

Signature extensions allow the definition of entirely new signatures and the growth of complex signatures from existing signatures.

2.1.2. The Category of Specifications

To model semantics, signatures are extended with *axioms* that define the intended semantics of the signature operations. A signature with associated axioms is called a *specification* [7].

Specification. *A specification SP is a pair $\langle \Sigma, \Phi \rangle$ consisting of a signature $\Sigma = \langle S, \Omega \rangle$ and a collection Φ of Σ-sentences (axioms).*

Although a specification includes semantics, it does not implement a program nor does it define an implementation. A specification only defines the semantics required of a valid implementation. In fact, for most specifications, there are a number of implementations that satisfy the specification. Implementations that satisfy all axioms of a specification are called models of the specification [7]. To formally define a model, we first define a Σ-algebra [7].

Σ-algebra or Σ-model. *Given a signature $\Sigma = \langle S, \Omega \rangle$, a Σ-algebra $A = \langle A_S, F_A \rangle$ consists of two families:*

- *a collection of sets, called the carriers of the algebra, $A_S = \{A_s \mid s \in S\}$; and*
- *a collection of total functions, $F_A = \{f_A \mid f \in \Omega\}$ such that if the rank of f is $s_1, s_2, \ldots, s_n \rightarrow s$, then f_A is a function from $A_{s1} \times A_{s2} \times \ldots \times A_{sn}$ to A_s. (The symbol $\times$ indicates the Cartesian product of sets here.)*

Model. *A model of a specification $SP = \langle \Sigma, \Phi \rangle$ is a Σ-algebra, M, such that M satisfies each Σ-sentence (axiom) in Φ. The collection of all such models M is denoted by Mod[SP]. The sub-category of **Mod**(Σ) induced by Mod[SP] is also denoted by Mod[SP].*

An example of a specification is shown in Figure 4. This specification is the original RING signature of Figure 1 enhanced with the axioms that define the semantics of the operations. Valid models of this specification include the set of all integers, Z, with addition and multiplication as well as the set of integers modulo 2, $Z_2 = \{0, 1\}$, with the inverse operation (-) defined to be the identity operation.

As signatures have signature morphisms, specifications also have specification morphisms. Specification morphisms are signature morphisms that ensure that the axioms in the source specification are theorems (are provable from the axioms) in the target specification. Showing that the axioms of the source specification are theorems in the target specification is a proof obligation that must be shown for each specification morphism. Specifications and specification morphisms enable the creation and modification of specifications that correspond to valid signatures within the category *Sign*. However, before we can formally define a specification morphism, we must first define a *reduct* [7].

```
spec Ring is
sorts ANY
operations
    as defined in Figure 1
axioms
    ∀a,b,c ∈ ANY
        a plus (b plus c) = (a plus b) plus c
        a plus b = b plus a
        a plus zero = a
        a plus(inv a) = zero
        a times (b times c) = (a times b) times c
        a times one = a
        one times a = a
        a times (b plus c) = (a times b) plus (a times c)
        (a plus b) times c = (a times c) plus (b times c)
end
```

Figure 4. Ring Specification

Reduct. *Given a signature morphism $\sigma: \Sigma \rightarrow \Sigma'$ and a Σ'-algebra A', the σ-reduct of A', denoted $A'|_\sigma$ is the Σ-algebra $A = \langle A_S, F_A \rangle$ defined as follows (with $\Sigma = \langle S, \Omega \rangle$):*

$$A_S = A_{\sigma(s)}' \text{ for } s \in S, \text{ and}$$
$$f_A = (\sigma(f))_{A'} \text{ for } f \in \Omega$$

A reduct defines a new Σ-algebra (or Σ-model) from an existing Σ'-algebra. It accomplishes this by selecting a set or functions for each sort or operation in Σ based on the signature morphism from Σ to Σ'. Thus if we have a signature, Σ', and a Σ'-model, we can create a Σ-model for a second signature, Σ, by defining a signature morphism between them and calculate the associated reduct. A reduct is now used to extend the concept of a signature morphism to form a *specification morphism* [7].

Specification Morphism. *A specification morphism from a specification SP = ⟨Σ, Φ⟩ to a specification SP' = ⟨Σ', Φ'⟩ is a signature morphism σ: Σ → Σ' such that for every model M ∈ Mod[SP'], M|σ ∈ Mod[SP]. The specification morphism is also denoted by the same symbol, σ: Σ → Σ'.*

We now turn to the definition of theories and theory presentations. Basically a *theory* is the set of all theorems that logically follow from a given set of axioms [6]. A *theory presentation* is a specification whose axioms are sufficient to prove all the theorems in a desired theory but nothing more. Put succinctly, a theory presentation is a finite representation of a possibly infinite theory. To formally define a theory and theory presentation we must first define *logical consequence* and *closure* [6].

Logical Consequence. *Given a signature Σ, a Σ-sentence φ is said to be a logical consequence of the Σ-sentences $\varphi_1,\ldots,\varphi_n$, written $\varphi_1,\ldots,\varphi_n \models \varphi$, if each Σ-algebra that satisfies the sentences $\varphi_1, \ldots, \varphi_n$ also satisfies φ.*

Closure, Closed. *Given a signature Σ, the closure, closure(Φ), of a set of Σ-sentences Φ is the set of all Σ-sentences which are the logical consequence of Φ, i.e., closure(Φ) = {φ | Φ ⊨ φ}. A set of Σ-sentences Φ is said to be closed if and only if Φ = closure(Φ).*

Theory, presentation. *A theory T is a pair ⟨Σ, closure(Φ)⟩ consisting of a signature Σ and a closed set of Σ-sentences, closure(Φ). A specification ⟨Σ, Φ⟩ is said to be a presentation for a theory ⟨Σ, closure(Φ)⟩. A model of a theory is defined just as for specifications; the collection of all models of a theory T is denoted Mod[T]. Theory morphisms are defined analogous to specification morphisms.*

Specification morphisms complete the basic tool set required for defining and refining specifications. This tool set can now be extended to allow the *combination*, or composition, of existing specifications to create new specifications. This is where category theory is extremely useful in information fusion. Often two specifications that were originally extensions from the same ancestor need to be combined. Therefore, the desired combined specification consists of the unique parts of two specifications and some "shared part" that is common to both specifications (the part defined in the shared ancestor specification). This combining operation is called a *colimit* [6]. The colimit operation creates a new specification from a set of existing specifications. This new specification has all the sorts and operations of the original set of specifications without duplicating the "shared" sorts and operators. To formally define a colimit, we must first define a cone (or *cocone*) [6].

Cone. *Given a diagram D in a category C and a C-object c, a cone from the base D to the vertex c is a collection of C-arrows $\{f_i: d_i \rightarrow c \mid d_i \in D\}$, one for each object d_i in the diagram D, such that for any arrow $g: d_i \rightarrow d_j$ in D, the diagram shown in Figure 5 commutes i.e., $g \circ f_j = f_i$.*

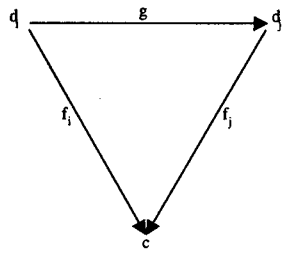


Figure 5. Cone Diagram

Colimit. *A colimit for a diagram D in a category C is a C-object c along with a cone $\{f_i: d_i \rightarrow c \mid d_i \in D\}$ from D to c such that for any other cone $\{f_i': d_i \rightarrow c' \mid d_i \in D\}$ from D to a vertex c', there is a unique C-arrow $f: c \rightarrow c'$ such that for every object d_i in D, the diagram shown in Figure 6 commutes (i.e., $f \circ f_i = f_i'$).*

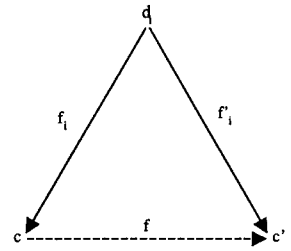


Figure 6. Colimit Diagram

Conceptually, the colimit of a set of specifications is the "shared union" of those specifications based on the morphisms between the specifications. These morphisms define equivalence classes of sorts and operations. For example, if a morphism for specification A to specification B maps sort α to sort β, then α and β are in the same equivalence class and thus is a single sort in the colimit specification of A, B, and the morphism between them. Therefore, the colimit operation creates a new specification, the colimit specification, and a cone morphism from each specification to the colimit specification. These cone morphisms satisfy the condition that the translation of any sort or operation along any of the morphisms in the diagram leading to the colimit specification is equivalent. An example of the colimit operation is shown in Figure 7 and Figure 8. Given the BIN-REL, REFLEXIVE, and TRANSITIVE specifications in Figure 7, the "colimit specification" would be the PRE-ORDER specification as shown in the diagram in Figure 8. Notice that the sorts E, X, and T belong to the same equivalence class in PRE-ORDER. Likewise, the operations •, =, and < also form an equivalence class in PRE-ORDER. Thus PRE-ORDER defines a specification with one sort, denoted by {E, X, T} and one operation, denoted by {•, =, <}, which is both

transitive and reflexive. The specification BIN-REL defines the "shared" parts of the colimit but adds nothing to the final specification.

```
spec Bin-Rel is
sorts E
operations
    • : E, E → Boolean
end

spec Reflexive is
sorts X
operations
    = : X, X → Boolean
axioms
    ∀ x ∈ X  x = x
end

spec Transitive is
sorts T
operations
    < : T, T → Boolean
axioms
    ∀ x, y, z ∈ T  (x < y ∧ y < z) ⇒ x < z
end

spec Pre-Order is
sorts {E, X, T}
operations
    {•, =, <} : {E, X, T}, {E, X, T} → Boolean
axioms
    ∀ x, y, z ∈ {E, X, T}
        x {•, =, <} x
        (x {•, =, <} y ∧ y {•, =, <} z) ⇒ x {•, =, <} z
end
```

Figure 7. Specification Colimit Example

A category in which the colimit of all possible C-objects and C-arrows exists is called *cocomplete*. As shown by Goguen and Burstall [2], the category *Sign* and *Spec* are both cocomplete; therefore, the colimit operation may be used freely within the category *Spec* to define the construction of complex theories from a group of simpler theories.

Using morphisms, extensions, and colimits as a basic tool set, there are a number of ways that specifications can be constructed [7]:

1. Build a specification from a signature and a set of axioms;
2. Form the union of a collection of specifications;
3. Translate a specification via a signature morphism;
4. Hide some details of a specification while preserving its models;
5. Constrain the models of a specification to be minimal;
6. Parameterize a specification; and
7. Implement a specification using features provided by others.

Many of these methods are useful in specifying and implementing information fusion systems. For instance, if we can define the shared part of two types of data, we can formally combine them using a colimit.

2.2. Functors

The previous sections defined the basic categories and construction techniques used to build large-scale software specifications. In this section, we extend these concepts further to define models of specifications and how they are related to the construction techniques used to create their specifications. Before describing this relationship, we define the concept of a *functor* that maps C-objects and C-arrows from one category to another in such a way that the identity and composition properties are preserved [5].

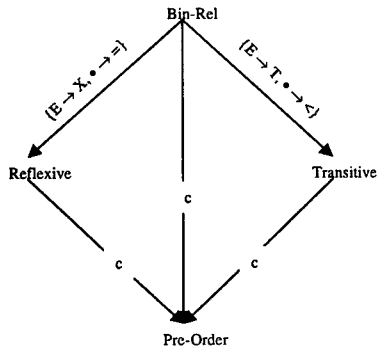


Figure 8. Example Colimit Diagram

Functor. *Given two categories **A** and **B**, a functor F: **A** → **B** is a pair of functions, an object function and a mapping function. The object function assigns to each object X of category **A** an object F(X) of **B**; the mapping function assigns to each arrow f: X → Y of category **A** an arrow F(f) : F(X) → F(Y) of category **B**. These functions satisfy the two requirements:*

$F(1_X) = 1_{F(X)}$
*for each identity 1_x of **A***

$F(g \circ f) = F(g) \circ F(f)$
*for each composite $g \circ f$ defined in **A***

Basically a functor is a morphism of categories. Actually, we have already presented two functors: the *reduct* functor that maps models of one specification (in the category $Mod[X_1]$) into models of a second specification (in the category $Mod[X_2]$) and the *models* functor that maps specifications in the category *Spec* to their category of models, *Mod[X]*, in *Cat*, the category of all sufficiently small categories.

3. AMFRS

To show applicability of the category theoretic notions described above to information fusion systems, we will discuss a case study of Automatic Multisensor Feature-based Recognition System (AMFRS) [4], which was originally developed using a model-based approach. In this case study, we transform the AMFRS framework into an equivalent system using a category theoretic approach. First we will discuss the original system and then show its equivalent structure using algebraic specifications and category theory.

3.1. Model-Theory Based Framework

In the original model-based development approach, wavelet-based models were developed for integration into the AMFRS to help recognize targets. AMFRS uses a model-based framework to describe how to combine information contained in the wavelets for use in the system. Within this framework, models were developed to help recognize targets based on wavelet coefficients that could be interpreted as meaningful features of the target.

In this framework, models were developed based on a *language* and its associated *theory* that described the semantics of the language. To combine languages and theories, three operators are used: reduction, expansion, and union. In general, the *reduction* operator removes symbols from a language along with all the sentences in which it exists in its associated theory. Expansion is the opposite. *Expansion* allows us to add symbols and new sentences about those symbols to the language. Finally, the *union* operator combines the symbols and sentences from two different language/theory pairs into a single language and a single theory.

Using these operators, Korona created a framework for combining languages and theories about two different types of sensor data into a single fused language and theory. This framework is shown in Figure 9. In Figure 9, we show only the language composition process. The theory fusion process is identical. In this example, we assume there are two sensors whose data is described by two languages L_r and L_i. These languages are extended to the languages L_r^e and L_i^e by adding symbols denoting operations on a subset of the wavelet coefficients used to describe the sensor data. These subsets of coefficients represent those coefficients that will be part of the final fused language. The coefficients are selected by the designer based on knowledge of the wavelet coefficients and their relationship to features in targets of interest.

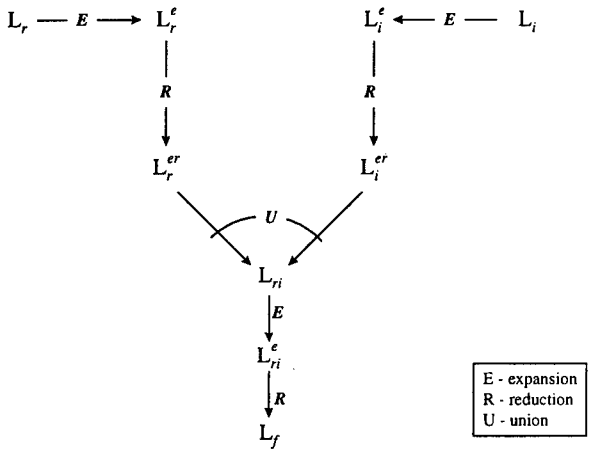


Figure 9. Model-Theory Based Framework

After the necessary symbols have been added to the languages, L_r^e and L_i^e are reduced by removing all the symbols not related to the coefficients selected for use in the final fused language. The new reduced languages, L_r^{er} and L_i^{er}, are then combined into a single language, L_{ri}, by the union operation. This language contains all the symbols representing the coefficients and operations on them required to construct the final fused language.

The last two steps in the process create our final fused language, L_f. First, L_{ri} is extended to L_{ri}^e by adding symbols denoting operations that combine the coefficients from L_r^{er} and L_i^{er}. Then, we create L_f by removing the symbols denoting those operations that do not work on the fused set of coefficients.

3.2. An Equivalent Categoric Framework

Before we convert the AMFRS model-based framework into a categoric framework, a few observations are necessary. First, the language and theory combination used in AMFRS is basically equivalent to an algebraic specification. An algebraic specification defines a set of sorts, operations over those sorts, and axioms that define the semantics of the operations. Constants, relations and functions defined via language symbols are defined as *operations* in an algebraic specification. Sentences of a theory translate to *axioms* in an algebraic specification. Algebraic *sorts* define a collection of values used in the operations.

The model-based expansion, reduction, and union operators also have counterparts in category theory. The basic operator in category theory is the *morphism*. In the category of *Spec*, which includes all possible algebraic specifications, these morphisms are *specification morphisms* that define how one specification is embedded in a second specification. That is, it defines a mapping from the sorts and

operations of the first specification into the sorts and operations of the second specification in such a way as to ensure the axioms of the first specification are theorems of the second specification (i.e., the axioms hold in the second specification under the defined mapping of sorts and operations). Thus a specification morphism can be used to define an expansion as well as a reduction (they are basically inverses of each other). If we have an expansion of specification *A* into specification *B*, in effect we have a morphism from *A* to *B*. Likewise, a reduction of specification *A* to specification *B*, indicates morphism from *B* to *A*. The language union operator can also be modeled easily using the category theory *colimit* operation. The colimit operation combines two (or more) specifications, automatically creating a morphism between the original specifications and the resulting colimit specification. If two specifications being combined using a colimit operation share common parts (e.g., they both use integers), these parts can be specified as common by defining morphisms from the common, or shared, specification to the individual specifications. This shared specification, along with the associated morphisms, are included in the colimit operation. The result of this is that the shared parts of the two specifications are not duplicated.

The conversion of the model-based framework into a category theoretic framework is shown in Figure 10. In this framework, the languages and their associated theories are converted to algebraic specifications (or *theory presentations*) and reductions and extensions are converted to morphisms. Note that a reduction from *A* to *B* results in a morphism from *B* to *A*. The union operation is converted to a colimit operation. The *S* specification denotes any shared part of specifications T_r^{er} and T_i^{er}. In this case it might include domain information about wavelets, targets, etc.

Figure 11 represents a simplification of the category theoretic setting shown in Figure 10. Basically, the morphisms σ_3, σ_4, and σ_8 from Figure 10 have been combined into morphism σ_{15} of Figure 11. This is possible since all the sorts, operations, and axioms removed by σ_3 and σ_4 can be carried along without changing the semantics. As we see when we get to the model creation phase, carrying along these extra sorts, operations, and axioms is an advantage.

Figure 12 is an even further simplification of the category theoretic setting of Figure 10. In Figure 12, the morphisms σ_1, σ_2 and σ_7 from Figure 10 have been combined into morphism σ_{14}. In this framework, we combine the two basic specifications together via the colimit operation *before* we insert any knowledge about which wavelet coefficients correspond to which interpretable features.

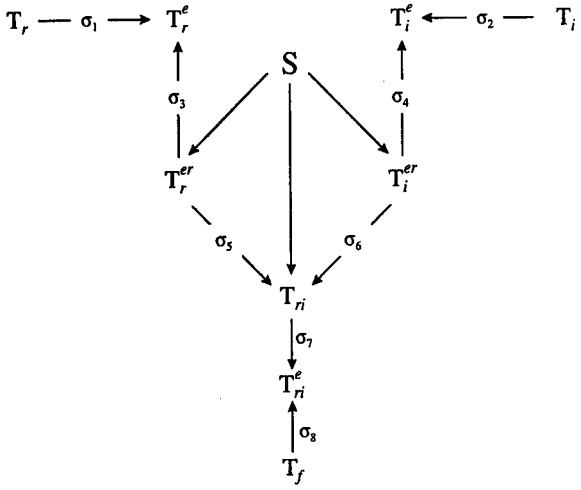


Figure 10. Categorical Framework

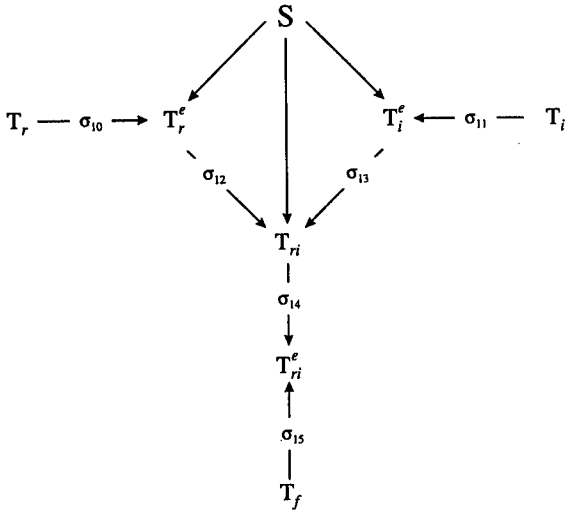


Figure 11. Simplified Categorical Setting

Since all the operations used to expand the basic specifications have a well defined interpretation in the expanded specifications (cf. [4]), the morphism σ_{14} becomes a *definitional extension* and the subdiagram contained in the dotted box becomes an *interpretation*. An interpretation basically says that we can build a model of T_f from a model of T_{ri}. This is a powerful construct in category theoretic software development tools such as Specware [3].

Finally Figure 13 describes how we create models in our category theoretic framework. In Figure 13, *MOD* represents the *model functor*, which takes specifications from the category *Spec* and maps them to a valid category of models, denoted *MOD[Spec]*, in the category *Cat* (the category of all sufficiently small categories). The nice part about the category

theoretic framework we have come up with is that each morphism, σ: $A \rightarrow B$, induces a *reduct functor*, $|_\sigma$, that automatically maps models of B to models of A. Therefore if we create a valid model for B, we automatically get a valid model for A! Following the flows of reduct functors in Figure 13, we now see that if we can create a valid model of T_f-*as*-T_{ri} (M_{ri}^e as pointed at by the large arrow in Figure 13) we can automatically create the valid models M_r, M_i, M_{ri}, and M_f from T_r, T_i, T_{ri}, and T_f respectively. Not only are these models consistent with their individual theories, but since all the models are based on a single initital model, they are consistent with each other as well.

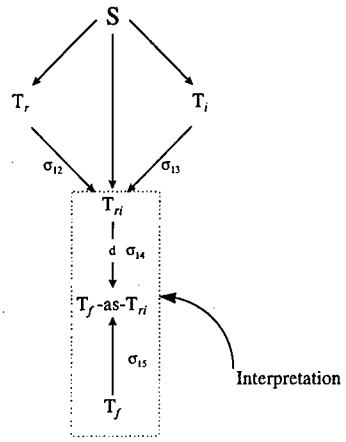


Figure 12. Theory Interpretation

4. Implications

There are many positive implications of putting the AMFRS design into a category theoretic setting. First, there is no information loss in translating languages and theories into algebraic specifications. In fact, we gain modeling ability by adding the notion of a sort. By using sorts, we can precisely define operation signatures. Also, the notions of morphisms, definitional extensions, colimits, and interpretations give us a wide variety of tools with well-defined meanings. We can prove when morphisms and definitional extensions exist as well as *construct* the resulting colimit specification based on a set of specifications and morphisms. All in all, category theory provides us a much greater capability to prove *relationships* between specifications. Finally, the categorical setting allows us to construct, in a provably correct manner, consistent sets of models required by the AMFRS system. All we have to do is construct one specific model and the models required by AMFRS can be generated automatically. The bottom line is, you lose nothing and gain a lot by using category theory in the development of formal information fusion systems such as AMFRS.

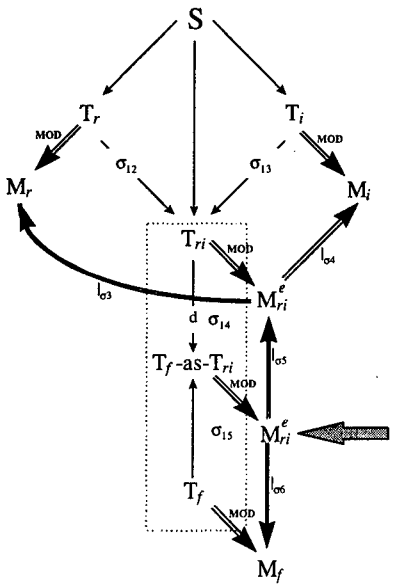


Figure 13. Model Creation using Theory Interpretation

5. References

1. DeLoach, Scott A. and Thomas C. Hartrum. "A Theory-Based Representation for Object-Oriented Domain Models," to appear in *IEEE Transactions on Software Engineering*.
2. Goguen, J. A. and R. M. Burstall. "Some Fundamental Algebraic Tools for the Semantics of Computation Part I: Comma Categories, Colimits, Signatures and Theories," *Theoretical Computer Science*, 31:175-209 (1984).
3. Jullig, Richard and Yellamraju V. Srinivas. "Diagrams for Software Synthesis." *Proceedings of the Knowledge Based Software Engineering Conference*. IEEE 1993.
4. Korona, Z. *Model-Theory Based Feature Selection for Multisensor Recognition*. Ph.D. Thesis, Northeastern University, 1996.
5. MacLane, Saunders and Birkhoff. *Algebra*. New York, NY: Chelsea Publishing Company, 1993.
6. Srinivas, Yellamraju V. *Category Theory Definitions and Examples*. Technical Report, Department of Information and Computer Science, University of California, Irvine, February 1990. TR 90-14.
7. Srinivas, Yellamraju V. *Algebraic Specification: Syntax, Semantics, Structure*. Technical Report, Department of Information and Computer Science, University of California, Irvine, June 1990. TR 90-15.

Incorporating Uncertainty into the Formal Development of the Fusion Operator

Jingsong Li
Department of Electrical and Computer Engineering
Northeastern University
Boston, MA 02115, U.S.A.

Mieczyslaw M. Kokar
Department of Electrical and Computer Engineering
Northeastern University
Boston, MA 02115, U.S.A.

Jerzy Weyman
Department of Mathematics
Northeastern University
Boston, MA 02115, U.S.A.

Abstract *This paper uses a formal approach to incorporating uncertainty of input information into the fusion process and decision making. Fuzzy set theory (fuzzy numbers, and fuzzy operators) is used to characterize and then manipulate (reason about) uncertainty. A library of specifications of fuzzy set theory is developed using category theory and Specware, a tool that supports category theory based algebraic specification of software. The library is then used to construct specifications of fuzzy information processing systems. The main construction in this process is composition. Category theory operators of limits and colimits are used for composition. As an example, a fuzzy edge detection algorithm is shown, which uses fuzzy operations in its processing. One of the advantages of this approach is that every aspect of the fusion process is specified formally, which allows us to reason about the uncertainty associated with the sensors and the processing.*

Keywords: fuzzy set, category theory, colimit

1 Introduction

In information fusion systems, uncertainty of information comes into the picture for a number of reasons: incompleteness of the coverage of the environment, inaccuracy of the sensors (e.g., limited resolution of sensors), background noise in the environment, and others. There are many ways of dealing with uncertainty. Statistical methods and efficient filtering algorithms have been applied to this area using mathematical tools, such as FFT or wavelets, but none in a completely formal way, i.e., these mathematical formalisms have been used to derive algorithms by humans, but not by computing machines (computers).

Why is a formal method so important? We know that in order to design a fusion system, we need to be able to reason about the impact of the uncertainty of the input information on the outcome of the fusion system, before the system is built. In other words, we need to be able to predict the performance of the fusion system for any given level of uncertainty and guarantee that it will give satisfactory solutions provided that the uncertainty of incoming information is within some prespecified bounds. With conventional methods, reasoning about the performance of the system cannot be done automatically, but even humans might draw different conclusions about a specific system due to the lack of full mathematical specification of the system.

In this paper, we describe the process by which uncertainty is formally incorporated into the fusion system design, so that it allows us to reason about the uncertainty of the deci-

sions of the fusion system while in the design phase. Section 2 describes how a fuzzy set theory library is built using category theory and Specware, and how the library is used to construct specifications of fuzzy information processing systems. This is the main part of the paper. Section 3 describes a simple conventional edge detection algorithm, and then maps this algorithm into a corresponding fuzzy edge detection algorithm in which all the operations are replaced by fuzzy operations. This part serves as an example of the application of our approach to reasoning about the uncertainty in information fusion. Section 4 concludes the paper and gives directions for future research.

2 Fuzzy Information Processing

Before fuzzy set theory was introduced by Zadeh in 1965, uncertainty was solely treated by probability theory. But there are some situations where uncertainty is non-probabilistic. In information processing systems, for instance, we cannot guarantee that the input data are precise numbers; instead they are often referred to as *approximately x*, or *around x*. The reason for this uncertainty is not that we measure the values with some error, but simply because we do not know what it should be. This uncertainty of imprecision can be modeled by using fuzzy set theory. Another example is evident in linguistic expressions, such as *tall, big, hot,* or *likely, unlikely,* etc. This linguistic uncertainty, of *vagueness* or *fuzziness*, can be well described by appropriate fuzzy sets.

In this paper we use fuzzy set theory to handle uncertainty in information processing systems. We show how fuzzy information processing systems can be specified by using category theory and Specware. Category theory is a mathematical technique that is suitable for representing relations between various types of objects [5]. Specifically, we are interested in relations between (algebraic) specifications. Specware is a tool that supports category theory based algebraic specifications of software [10]. This section will talk about the construction of a fuzzy set theory library and fuzzy information processing specifications.

2.1 Construction of Fuzzy Set Theory Library

The fuzzy set theory library is composed of specifications (also called *specs*) of the main concepts of fuzzy set theory: *fuzzy sets, fuzzy numbers, α-cuts*, and *fuzzy arithmetic operations.* These specs are useful in composing formal specifications of fuzzy information processing systems.

2.1.1 Fuzzy Sets

There are a number of definitions for fuzzy sets. Two most popularly used definitions are listed here for comparison, out of which we chose the second one.

Definition 1 [4]: Fuzzy set A is a set of ordered pairs

$$A = \{(x, \mu_A(x)) | x \in X\}$$

where X is a collection of objects (called *universe of discourse*), and $\mu_A(x)$ is the *membership function.* This function takes real values between 0 and 1.

Definition 2 [3]: Fuzzy set A is a function

$$A : X \rightarrow [0, 1],$$

where X is the universe of discourse.

The difference between the two definitions is that the former defines a function that is not necessarily total on X, while the latter requires that the function be total. Since Specware requires that all functions be total, we chose the second definition of fuzzy set for building specifications. The diagram of the specification of fuzzy set is shown in Figure 1.

The spec UNI-INTVL *imports* REAL and introduces a new *sort*: *Uni_intvl* = *Real* | *between_zero_one*?. FUZZY-SET is a *definitional extension* [5] of the *colimit* of UNI-INTVL and SET; it defines a *function sort*: *Fuzzy_set* = $E \rightarrow$ *Uni_intvl*, where E is the

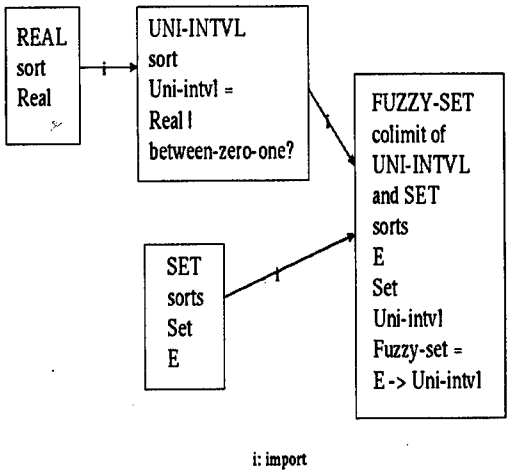


Figure 1: Diagram for Fuzzy-set

type of all elements in Set. In the FUZZY-SET spec, α-cut and height are defined as

$$op\ alpha\_cut : Fuzzy\_set, Uni\_intvl \rightarrow Set$$
$$op\ height : Fuzzy\_set \rightarrow Uni\_intvl$$

The α-cut is a powerful concepts that links fuzzy sets with sets. The application of the α-cut to a fuzzy set results in a set, and thus all operations and relations of sets can be applied to the α-cuts of the fuzzy set, or to α-levels.

2.1.2 Fuzzy Numbers

Fuzzy numbers are one specific type of fuzzy set. The universe od discourse for fuzzy numbers is real numbers. Fuzzy number A has the form: $A : Real \rightarrow [0,1]$. It has the following properties:

- A must be a normal fuzzy set. That is, the height of the fuzzy set A should be 1:

$$height(A) = \sup_{x \in X} A(x) = 1$$

- A must be a convex fuzzy set. The property of convexity is captured by the following theorem:

  Theorem: A fuzzy set A on *Real* is convex iff

  $$A(\lambda x_1 + (1-\lambda)x_2) \geq min[A(x_1), A(x_2)]$$

  for all $x_1, x_2 \in Real$ and all $\lambda \in [1,0]$, where *min* denotes the minimum operator.

- α-cut of the fuzzy set A should be a closed interval for every $\alpha \in (0,1]$.

These properties are intuitively obvious. A fuzzy number is normal since our concept of a fuzzy number "approximately x" means that it is fully satisfied by x itself. We require that the shape of the fuzzy number be monotonicly increasing on the left and monotonicly decreasing on the right, so α-cuts of any fuzzy number should be closed intervals, which leads to the property that fuzzy numbers are convex.

Fuzzy number is specified in the spec FUZZY-NUMBER, which *imports* FUZZY-SET and adds one *sort axiom*: $E = Real$. It also adds two axioms: *normality* and *convexity*.

2.1.3 Fuzzy Operations

In [3], two methods have been presented for developing fuzzy arithmetic. One method is based on interval arithmetic. Let A, B denote two fuzzy numbers, * denote any of the four basic arithmetic operations, $+, -, \times, and \div$. Then $A * B$ is a fuzzy number, which can be represented by

$$A * B = \bigcup_{\alpha \in [0,1]} ({}^{\alpha}A *^{\alpha} B) \times \alpha$$

This method requires using α-cuts of fuzzy numbers. The second method represents fuzzy number A*B in the following way:

$$(A * B)(z) = \sup_{z=x*y} min[A(x), B(y)]$$

for all $z \in Real$. We chose the latter one because it is more explicitly expressed, thus more convenient to be specified in Specware.

Fuzzy arithmetic operations are specified in the spec FUZZY-ARITHM, which is a *definitional extension* of FUZZY-NUMBER, with fuzzy operations being of the following types.

$$op\ f\_add : Fuzzy\_number, Fuzzy\_number \rightarrow Fuzzy\_number$$
$$op\ f\_sub : Fuzzy\_number, Fuzzy\_number \rightarrow Fuzzy\_number$$
$$op\ f\_mult : Fuzzy\_number, Fuzzy\_number \rightarrow Fuzzy\_number$$
$$op\ f\_div : Fuzzy\_number, Fuzzy\_number \rightarrow Fuzzy\_number$$

2.2 Fuzzy Information Processing

There are three stages in fuzzy information processing: fuzzification, fuzzy reasoning, and defuzzification. They are covered in the following three subsections.

2.2.1 Fuzzification

The first step in fuzzy information processing is to fuzzify input data. There are many ways to do this. We chose the one in which a triangular membership function is involved. For a given value c, we define the triangular fuzzy number A, such that for all $x \in Real$, $A(x)$ satisfies the equation

$$A(x) = \begin{cases} 0 & \text{if } x < c - \delta, \\ & \text{or } x > c + \delta \\ (x - c + \delta)/\delta & \text{if } c - \delta \leq x \leq c \\ (c + \delta - x)/\delta & \text{if } c \leq x \leq c + \delta \end{cases}$$

In this equation, δ represents the uncertainty level. The larger the δ, the more uncertain the input data.

One kind of typical input data for an information fusion system is image, which is generally sampled into a rectangular array of pixels. Each pixel has an x-y coordinate that corresponds to its location within the image, and an intensity value representing brightness. The spec IMAGE *imports* INTEGER and REAL, and defines a *function sort*: $Image = Integer, Integer \rightarrow Real$. The spec FUZZIFICATION is generated by taking the *colimit* of IMAGE and FUZZY-ARITHM, and defining another *function sort*: $Fuzzy\_image = Integer, Integer \rightarrow Fuzzy\_number$. The diagram for this specification is shown in Figure 2. FUZZIFICATION maps *Image* to *Fuzzy_image*, so that each pixel has a corresponding fuzzy triangular number instead of a crisp number. Also in this spec, two operations are defined:

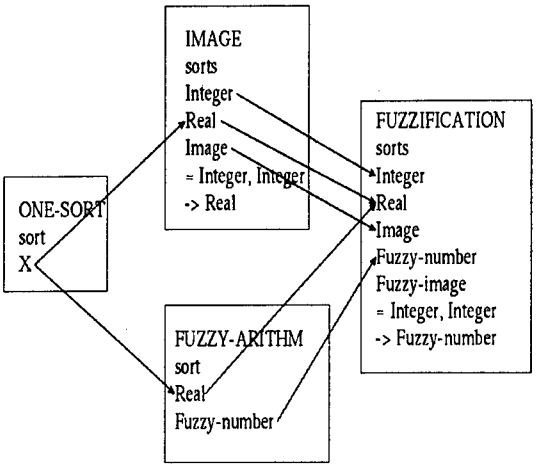


Figure 2: Diagram for Fuzzification

$$op\ fuzzify : Real, Nonzero \rightarrow Fuzzy\_number$$
$$op\ fuzzify\_2 : Real \rightarrow Fuzzy\_number$$

where $fuzzify$ takes a crisp number and some uncertainty level, and generates a fuzzy triangular number. The operation $fuzzify\_2$ deals with the situation when the uncertainty level is zero, which means there is no fuzziness about the result. The latter operation is specified so that a crisp number can also be regarded as a fuzzy number.

2.2.2 Fuzzy reasoning

Fuzzy reasoning takes fuzzified inputs and applies fuzzy arithmetic operations on them. For instance, as we discussed above, the input can be a fuzzy image in which each pixel corresponds to a fuzzy triangular number. While for crisp numbers we apply some arithmetic operations, like $+, -, \times, and \div$, for fuzzy numbers we will apply f_add, f_sub, f_mult, and f_div, as specified in FUZZY-ARITHM. Some additional fuzzy operations are specified there too, which will be useful in our applications. One is fuzzy minimum($fmin$), another is fuzzy maximum($fmax$). Let A, B denote two fuzzy numbers, then

$$fmin(A,B)(z) = \sup_{z=min(x,y)} min[A(x), B(y)]$$
$$fmax(A,B)(z) = \sup_{z=max(x,y)} min[A(x), B(y)]$$

for all $z \in Real$. The results of these two operations are fuzzy numbers. These two operations introduce partial ordering of fuzzy numbers.

Corresponding logic operations such as fuzzy equal($fequal$) and fuzzy less than(flt) are also specified here. There are many ways to define such operations. Here we have chosen the following:

$$op\ fequal : Fuzzy\_number, Fuzzy\_number \rightarrow Fuzzy\_number$$
$$op\ flt : Fuzzy\_number, Fuzzy\_number \rightarrow Fuzzy\_number$$

The operation $fequal$ takes two fuzzy numbers, defuzzifies them and compares the difference of the result. If the difference is less than a threshold, $fequal$ will return a $fone$, which is generated by $fuzzify(one, \alpha)$. α is the value where the two membership functions intersect and α will be zero if there is no intersection. If the difference is larger than the threshold, $fequal$ will return a $fzero$, which is generated by $fuzzify(zero, \alpha)$. The intersection of the two membership functions are taken to generate α, the same way as in $fuzzify(one, \alpha)$. The result of $fequal$ and flt is either $fone$ or $fzero$. This is the fuzzy equivalent of boolean values true and false. They are not limited to stating whether something is a fact or not, but in addition to this, they give the value of the uncertainty associated with such a statement.

2.2.3 Defuzzification

The input to the defuzzification process is a fuzzy number, and the output is a crisp number. There are several defuzzification methods - *centroid calculation* that returns the center of the area under the curve of the fuzzy number, *middle of maximum* that returns the average of the maximum value of the fuzzy number, *largest of maximum*, and *smallest of maximum*. We chose the *largest of maximum* method to implement the defuzzification process.

Defuzzification is implemented in DEFUZZIFICATION, which is a *definitional extension* of FUZZY-NUMBER. This spec defines the *defuzzify* operation as: $op\ defuzzify : Fuzzy\_number \rightarrow Real$. It takes a fuzzy number, finds the largest of maximum of its membership function, and returns the real number as defuzzification result. In our situation we fuzzify the input data using triangular membership function, so after fuzzy operations are applied to these fuzzy triangular numbers, the result will always have only one peak value. Therefore the largest of maximum of its membership function will always return only one value. There are situations where other types of fuzzification are used, and then the defuzzification spec should be more complex.

3 An Example: Fuzzy Edge Detection

In this section, we will show how to use fuzzy information processing specifications to translate a standard detection algorithm into a fuzzy detection algorithm, and see how uncertainty of input data propagates during the process and influences the final decision.

3.1 Edge Detection Algorithm

An edge in an image could be considered as a boundary at which a significant change of *intensity*, I, occurs. Detecting an edge is very useful in object identification, because edges represent shapes of objects. There are many algorithms for edge detection. The objective of an edge detection algorithm is to locate the regions where the *intensity* is changing rapidly. So we can decompose the whole process into two steps, the first is to derive *edge points* in an image, the second is to apply edge detection method only to these points.

We use the *Laplacian-based* method to derive *edge points*. *Edge points* are where the second-order derivatives of the points are zero, *zero crossing*. So *edge points* can be searched by looking for *zero crossing points* of $\nabla^2 I(x,y)$, which can be calculated by the equation

$$\nabla^2 I(x,y) = I(x+1,y) + I(x-1,y) + I(x,y+1) + I(x,y-1) - 4I(x,y)$$

In order to avoid false edge points, *local variance* is estimated and compared with a threshold. The local variance can be estimated by

$$\sigma^2(x,y) = \frac{1}{(2M+1)^2} \sum_{k_1=x-M}^{x+M} \sum_{k_2=y-M}^{y+M} [I(k_1,k_2) - m(k_1,k_2)]^2$$

where

$$m(x,y) = \frac{1}{(2M+1)^2} \sum_{k_1=x-M}^{x+M} \sum_{k_2=y-M}^{y+M} I(k_1,k_2)$$

with M typically chosen around 2. Since $\sigma^2(x,y)$ is compared with a threshold, the scaling factor $\frac{1}{(2M+1)^2}$ can be eliminated.

The spec EDGE-POINT *imports* IMAGE and defines a *sort* and some *ops*:

$$\begin{aligned} &sort\_axiom\ Edge\_point = \\ &(Integer, Integer) | edge\_point? \\ &op\ edge\_point? : Integer, Integer \\ &\quad \rightarrow Boolean \\ &op\ grad : Integer, Integer \rightarrow Real \\ &op\ var : Integer, Integer \rightarrow Real \end{aligned}$$

where *grad* and *var* represent gradient and local variance respectively, and for all Integers x, y:

$$edge\_point?(x,y) \iff grad(x,y) = 0 \wedge var(x,y) < thrd$$

Therefore a pixel at (x,y) satisfies an edge point if and only if the gradient equals zero and the local variance is less than the threshold. Otherwise the pixel is not an edge point.

3.2 Fuzzy Edge Detection

Now we will use fuzzy information processing specifications and translate the above edge detection algorithm into a fuzzy edge detection algorithm.

Fuzzy edge detection is specified in FUZZY-EDGE-POINT, which *imports* FUZZIFICATION, and defines a *function sort*:

$$Fuzzy\_edge\_point = Integer, Integer \rightarrow Fuzzy\_number$$

which maps each pixel to a fuzzy number representing the level at which the pixel satisfies an edge point. This fuzzy number represents *fuzzy boolean*. Instead of making the decision that a pixel *is* an edge point or *is not* an edge point, a $fone$ or a $fzero$ is given. A $fone$ states that the pixel satisfies an edge point with uncertainty as described by the fuzziness of this $fone$. A $fzero$, on the other hand, states that the pixel does not satisfy an edge point with uncertainty that is described by the fuzziness of this $fzero$. The following constants and operations are specified:

$$\begin{aligned} &const\ delta : Nonzero \\ &const\ thrd : Real \\ &op\ fgrad : Integer, Integer \rightarrow Fuzzy\_number \\ &op\ fvar : Integer, Integer \rightarrow Fuzzy\_number \end{aligned}$$

where $fgrad$ and $fvar$ represent fuzzy gradient and fuzzy local variance respectively. Calculation of $fgrad$ and $fvar$ requires fuzzy arithmetic operations that have been specified before. The operations $fequal$, flt and $fmin$ are

also needed here to realize fuzzy edge detection. The operation $fequal$ takes two fuzzy numbers and returns a $fzero$ or a $fone$, representing how similar these two fuzzy numbers are. The operation flt takes two fuzzy numbers and returns a $fzero$ or a $fone$, representing how much the first one is less than the second one. For all Integers x, y:

$$Fuzzy\_edge\_point(x,y) = fuzzy\_min[fequal(fgrad(x,y), fuzzify(one, delta), flt(fvar(x,y), fuzzify(thrd, delta))]$$

Thus the likelihood that one pixel satisfies an edge point depends on both the likelihood that the fuzzy gradient is close to zero and the likelihood that the fuzzy local variance is less than a threshold. The more the fuzzy gradient is near zero and the fuzzy local variance is far less than the threshold, the more likely this pixel is an edge point. Then $fequal(fgrad(x,y), fuzzify(zero, delta))$ should return a $fone$ with less uncertainty, and $flt(fvar(x,y), fuzzify(thrd, delta))$ should also return a $fone$ with less uncertainty. Therefore $Fuzzy\_edge\_point(x,y)$ corresponds to a $fone$ with less uncertainty.

If $fequal(fgrad(x,y), fuzzify(zero, delta))$ returns a $fzero$, which means fuzzy gradient of the pixel (x,y) is not close to zero with some uncertainty, and if $flt(fvar(x,y), fuzzify(thrd, delta))$ also returns a $fzero$, which means fuzzy local variance of the pixel (x,y) is not less than a fuzzy threshold, then $Fuzzy\_edge\_point(x,y)$ should return a $fzero$, which is the fuzzy minimum of the two results and which shows that the pixel is not an edge point with some uncertainty.

If one of these two operations($fequal$ and flt) returns a $fzero$, and the other returns a $fone$, then $Fuzzy\_edge\_point(x,y)$ should return a $fzero$ which is the fuzzy minimum of the two results. It shows that the pixel is not an edge point with some uncertainty.

3.3 Results and Analysis

In order to show that with this approach we can reason about the influence of uncertainty of input information on the final decision before the system is built, we specify a GOAL spec, which *imports* FUZZY-EDGE-POINT and introduces a theorem:

$$\forall \delta_1, \delta_2 \in Real, \delta_1 \leq \delta_2 \Longrightarrow \alpha_1 \leq \alpha_2$$

where δ_1 and δ_2 are two different values chosen to fuzzify the input data and represent the uncertainty levels of the input information, and α_1 and α_2 are the generated uncertainty values for deriving the results of $Fuzzy\_edge\_point(x,y)$ for the two different fuzzified images. These α_1 and α_2 represent the uncertainty levels in decision making. They are influenced by the result of the *fuzzy gradient* and the *fuzzy local variance*. It is natural that the more uncertain the input data the more uncertain the decision. Depending on the values δ_1, δ_2, α_1 and α_2, the theorem prover [10] returns either a "yes" or a "no".

In the above example we have applied fuzzy information processing specifications on a standard edge point derivation algorithm and the results show that the uncertainty of input data propagates through the whole process and influences the uncertainty level of the decision. The uncertainty of input data influences the fuzzy gradient and the fuzzy local variance results, which in turn influence the uncertainty of the decision. So instead of giving a crisp decision (true or false), a fuzzy decision is given: true with some uncertainty or false with some uncertainty. The relation between the uncertainty levels in the final decision and in the input information can be proved in this specification stage.

4 Conclusions and Future work

In this paper we have introduced a formal approach to characterize and manipulate uncer-

tainty in information processing systems. We chose fuzzy set theory to represent uncertainty. We have shown how to specify basic elements of fuzzy set theory in Specware. As an example, fuzzy information processing specifications were applied to an edge detection algorithm. We showed how the uncertainty of input information propagates and influences the final decision.

In our future work, we will enrich the fuzzy set theory library by putting in more specifications for fuzzy set theory. α-cut is a powerful link between fuzzy set and crisp set, so more specs for α-cut will be built. We will also put more specs in the fuzzy information processing system. For instance, various fuzzification methods other than triangular will be specified. Trapezoidal, Gaussian, and bell fuzzification methods are three most popularly used. They can represent different levels and kinds of uncertainty among input data or decision making. Fuzzy reasoning will be enriched by defining different versions of fuzzy equal and fuzzy less than. Other defuzzification methods will also be specified.

Also in our future work, we will generalize this uncertainty topic by using *random set* instead of *fuzzy set* to characterize and manipulate uncertainty. We will also specify random processing and formally introduce *randomness* to some typical information processing problems.

References

[1] G. J. Klir. *On the Alleged Superiority of Probabilistic Representation of Uncertainty.* IEEE Transactions on Fuzzy Systems, Vol.2, No.1, 1994.

[2] G. J. Klir, U. S. Clair and B. Yuan. *Fuzzy Set Theory Foundations and Applications.* Prentice Hall PTR, 1997.

[3] G. J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic Theory and Applications.* Prentice Hall PTR, 1995.

[4] J. R. Jang and C. Sun. *Neuro-Fuzzy Modeling and Control.* IEEE Transactions, 1995.

[5] S. A. DeLoach and M. M. Kokar *Category Theory Approach to Fusion of Wavelet-Based Features.* To be published on Fusion'99, 1999.

[6] J. S. Lim. *Two-Dimensional Signal and Image Processing.* Prentice Hall, Inc, 1990.

[7] G. A. Baxes. *Digital Image Processing.* John Wiley & Sons, Inc, 1994.

[8] B. C. Pierce. *Basic Category Theory for Computer Scientists.* MIT Press, 1991.

[9] Y. V. Srinivas *Category Theory Definitions and Examples.* Technical report, Kestrel Institute, Palo Alto, California, 1990.

[10] *Specware Language Manual.* Kestrel Institute, Palo Alto, California, 1998.

[11] *Specware User's Guide.* Kestrel Institute, Palo Alto, California, 1998.

A Formal Approach to Information Fusion

Mieczyslaw M. Kokar
Department of Electrical and Computer Engineering
Northeastern University
Boston, MA 02115, USA

Jerzy A.Tomasik
Universite Blaise Pascal
LLAIC1, BP86 63172 AUBIERE
France

Jerzy Weyman
Department of Mathematics
Northeastern University
Boston, MA 02115, USA

Abstract *This paper proposes a formalization of the notion of "information fusion" within the framework of formal logic and category theory. Within this framework information fusion systems can be specified in precise mathematical terms allowing in this way to formally reason about such specifications, designs and implementations. The notion of fusion proposed in this paper differs from other approaches, where either data or decisions are fused. Here, the structures that represent the meaning of information (theories and models) are fused, while data are then simply processed using these structures (filtered through these structures). Within this framework the requirement of consistency of representations is formally and explicitly specified and then can be manipulated by the computer using automatic reasoning techniques.*

Keywords: information fusion, formal methods, category theory, model theory

1 Introduction

An *information fusion system (IFS)* (see Figure 1) may receive inputs from various sources: sensors, data bases, knowledge bases, and other systems (over communication lines). In our discussion we will focus on inputs from sensors, since other sources of information can be considered as special kinds of sensors. Sensors provide measurements of a number of interrelated variables (*n-tuples*). In mathematical sense, sensors output either functions or relations. In general, the *goal* of an IFS is to interpret data received through sensors. It is expressed in a prespecified *goal language* understandable to either the user or another system.

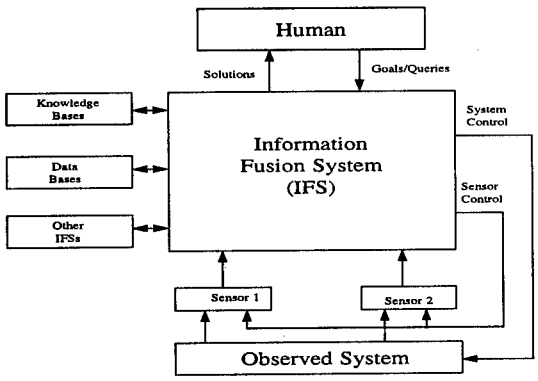


Figure 1: Information Fusion System (IFS)

A natural requirement for an information fusion system is that the interpretation of the data be "correct". Intuitively, this means that the objects identified by the IFS really exist in the world, that these objects have the features as identified by the IFS, that the relations recognized by the IFS really exist in the world, and that the interpretation does not violate the constraints that the world is known to obey, e.g., the laws of physics. In order to main-

tain the truthfulness of the interpretation, the system must maintain *consistency* of its representation.

To deal with the issue of correctness of interpretations we use the framework of model theory [1]. In particular, we make use of *formal languages* to describe the world and the sensing process and *models* to represent sensor data, operations on data, and relations among the data. Models consist of *carriers* of different sorts (usually sets) and many-sorted *operations*, and *relations* among the elements of different carriers. We use *theories* to represent symbolic knowledge about the world and about the sensors.

Fusion is then treated as a goal-driven operation of combining a fixed number of languages, theories and classes of models related to the goal, the sensors and the background knowledge, into one combined language, one combined theory and one combined class of models of the world. Therefore, fusion is a *formal system operator* that has multiple languages, theories and classes of models for inputs and a single language, a theory, and a class of models as the output.

This understanding of fusion differs from more traditional approaches [2, 3], where issues like consistency are not dealt with explicitly. Rather, there is an underlying presumption that the operations of fusion are implemented in a consistent way by the human. In our approach, on the other hand, a framework is provided in which the requirement of consistency of representations can be formally and explicitly specified and then can be manipulated by the computer using automatic reasoning techniques.

Although there are several definitions of "fusion" in the subject literature, there does not seem to be an agreement on what is and what is not fusion. In Section 2 we argue that the issue of fusion must be addressed in the specification phase. Then in Section 3, we provide our formal definition of fusion. In Section 4 we identify two parts of the fusion problem: syntactic fusion and semantic fusion. Section 5 puts the problem of fusion in the category theory framework and discusses fusion operators. We present and example of a specification developed according to our approach in Section 6. Finally, in Section 7 we provide conclusions.

2 Decomposition of the IFS

In this presentation we follow a top-down approach by progressively decomposing the problem of development of an IFS into simpler subproblems. In the first cut we decompose the IFS into three subsystems, as shown in Figure 2. This decomposition follows the formal approach to software development, where code is developed in the process of progressive refinement of a formal software specification. Information Processing represents the actual running system that takes inputs from all the sources and produces outputs in real time. The main fusion problem, as presented in this paper, is solved in Specification Synthesis. This is essentially the only block where expertise of sensors and scenarios is needed. Code Generation can be performed independently of such expertise.

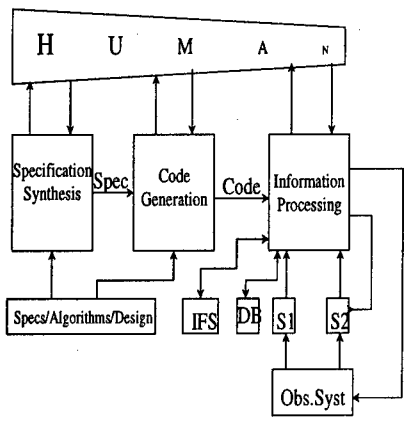


Figure 2: Information Fusion System: First-level Decomposition

3 The Fusion Problem

In this section we consider the main fusion block, i.e., the Specification Synthesis block. By specifications we mean signatures (languages), theories over the signatures, and classes of models of the theories. We show the first decomposition of the Specification Synthesis block in Figure 3. As we said in Section 1, the goal of information fusion is to develop a fused theory T_f and a fused class of models $\{M_f\}$. The inputs to this fusion process, as shown in Figure 3, are some or all of the following:

(1) Σ_G, Σ_i, Σ_j, $\Sigma_b, \ldots$ – signatures associated with the goal of the fusion system, corresponding sensors and background knowledge theories. These signatures include variables and constant symbols of different sorts as well as many-sorted operation and relation symbols.

(2) T_G, T_i, T_j, $T_b, \ldots$ – formal theories describing the *goal theory*, knowledge about the sensors, and theories of the world (background knowledge) expressed in terms of the above described signatures. Background knowledge contains constraints on possible interpretation of the received data and/or special theories like Theory of Reals (Real Closed Field Theory), Random Sets Theory, Elementary Theory of Boolean Algebras that can be utilized in the process of constructing the fused theory.

(3) $\mathcal{G}$ – goal. These are queries about the world that cannot be answered in general by using only one of the sensors (information sources) but can be answered by using many (all) sensors. They are formulas expressed in terms of the signature Σ_G of the goal theory T_G.

(4) $\{M_G\}$, $\{M_i\}$, $\{M_j\}$, $\{M_b\}, \ldots$ – classes of models associated with the theories T_G, T_i, T_j, $T_b, \ldots$, respectively.

The Fusion Problem

Given the knowledge described above, construct a *fused theory* T and an appropriate class of *fused models* $\{M\}$, such that for any model M in $\{M\}$:

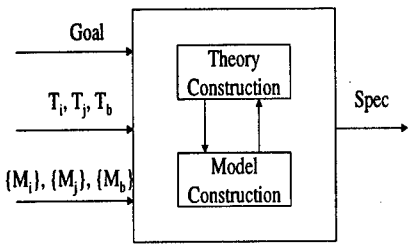


Figure 3: Specification Synthesis

1. $M \models G$
2. $M \models T$
3. $M \models T_b$

In some cases we might be given specific models M_i, M_j, instead of classes of models. Depending on which of the above are available, and depending on some other preferences, the process of developing the fused theory and class of models may be arranged on many different ways. For instance, we might first develop a fused theory T_f, and then find a class of models associated with this theory.

4 Syntactic/Semantic Fusion

One way to achieve the fusion goal is to split the inputs to the Specification Synthesis block (Figure 3) between the two tasks, so that purely semantic information (theories) are input into the Theory Construction task and the semantic inputs (models) are input into the Model Construction task. We denote the syntactic task by ∇_T, and the semantic task by ∇_M.

To be consistent with the formulation of the fusion problem in Section 3, the diagram represented in Figure 4 must commute. This can be described by the following relations.

$$M \models \nabla_T(T_G, T_i, T_j, T_b)$$

$$M = \nabla_M(M_G, M_i, M_j, M_b)$$

$$M_G \models T_G, M_i \models T_i, M_j \models T_j, M_b \models T_b$$

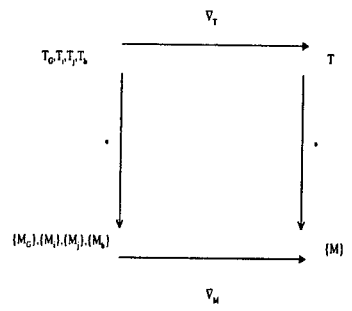


Figure 4: Syntactic and Semantic Levels of Specification Synthesis

4.1 Syntactic Theory Construction

The Theory Construction task can be considered as a goal-driven process that starts with the goal theory T_G. This theory contains a goal sentence G. The intent is to prove that the goal is true. This theory has to be combined with (extended by) other theories in order to make such a proof possible. It is natural to use the theories of the sensors in the first place. If the goal still cannot be proved, other background theories T_b need to be added. Various standard mathematical theories are added also in this step. The signatures of the goal theory and of the sensor theories, as well as some non-logical symbols appearing in these theories, can be used in the search for theories to add. As a result, we obtain a sufficiently rich theory T (specification) in which all sorts and operations from the goal theory T_G should be definable. In other words, the transition from the goal and sensor theories to the fused theory T can be achieved by appropriate definitional extensions of these theories using the background theories T_b.

4.2 Semantic Model Construction

In the Model Construction task we need to combine structures (classes of models of the particular theories fused in the syntactic task) into one class of structures. Since as a net result, this operation should produce such a class of structures $\{M\}$ that each one of them is a model of the fused theory T, the semantic model construction operation ∇_M must be so chosen that this property holds.

5 The Fusion Operator

In Section 4 we presented fusion as consisting of two operators, ∇_T and ∇_M. What can these operators be? In this section we propose a category theory based approach to this problem, similar to the one taken in the Specware approach [4]. In this approach theories are represented as specifications. They are objects in the Small Categories (Cat). Relationships among them are morphisms. Composition of theories is done using the *colimit* operation. Models of the theories are objects of another category (Mod).

According to this paradigm, Figure 4 can be represented as in Figure 5. In this diagram, corners of the diagram represent objects (or collection of objects). The arrows represent morphisms. The operators became:

$$\nabla_T(T_G, T_i, T_j, T_b) = Col(T_G, T_i, T_j, T_b)$$

$$\nabla_T(\{M_G\}, \{M_i\}, \{M_j\}, \{M_b)\} =$$

$$Lim(\{M_G\}, \{M_i\}, \{M_j\}, \{M_b)\}$$

where Col represents the colimit operator and Lim represents the limit operator. Note that, since Lim and Col are two contravariant operators, the morphism arrows point in opposite directions.

According to this diagram, fusion is accomplished by two operators: colimit and limit. The colimit operation combines (glues) two theories (specifications) along the common part. It is a *shared union* of two theories. In other words, first, common parts are identified in the languages associated with particular theories, then these common parts are renamed so that they have the same symbols in both theories, then the renaming is reflected in the axioms of the theories, and finally, the theories are put together into one structure (one theory).

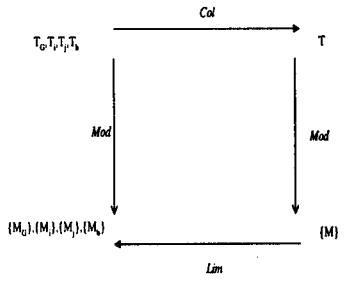


Figure 5: Category Theory Fusion Operators

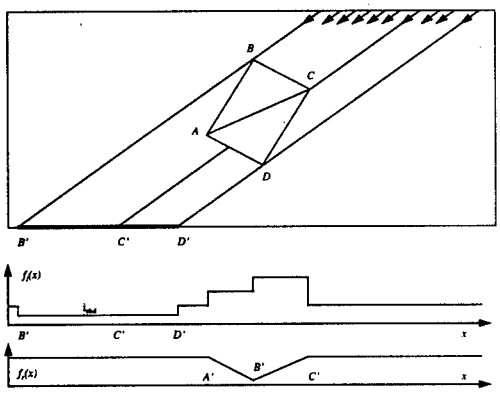


Figure 6: A scenario for sensor data fusion

Note also that the arrows from theories to models are of a different kind - they are *functors* that map objects of one category into the objects of another category.

6 Example

In this section we discuss a simple and idealized fusion scenario. In this example (see Figure 6) we consider a world which is a two dimensional plane with two kinds of objects possible: rectangles and triangles (with one right angle). The objects are illuminated with parallel light; the light direction is denoted by the angle α, as indicated in the figure. The world is measured through two sensors: a one-dimensional vision sensor, and a one dimensional range sensor.

The goal is object recognition. In some cases the range sensor is sufficient for the classification of an object into one of the three classes. E.g., when an acute angle is at the sensor side, the range sensor gives enough information to classify the object as either a right triangle or as an illegal object. Nevertheless, in some other cases, the information provided by the range sensor is not sufficient to make such a distinction. The advantage of the vision sensor stems from its ability to see shadows. In some configurations (sizes of an object and its rotational location), the size of the shadow and its location can provide the extra information that can be used to decide if the object is a triangle or a rectangle (as shown in Figure 6).

To understand this example the reader has to possess some knowledge of geometry and physics. We cannot expect that a computer has this kind of capabilities. Our goal is to understand the mechanisms involved in the above example, formalize these mechanisms, and then implement them in the computer so that this knowledge can be incorporated in the processing automatically.

6.1 Formalization of Knowledge

In the following we list the theories involved in the recognition process, show examples of the theories, and describe how they are fused. A complete presentation would include the description of appropriate classes of models. We implemented these theories in the specification language Slang, used by Specware [4], a formal method tool. For readability, however, the theories are presented here using common mathematical notation.

Theory $\mathcal{T}_r$: Range Sensor. The theory of the range sensor, $\mathcal{T}_r$, consists of the following two axioms:

1. $f_r(x) = y \wedge y < 1 \Rightarrow O_r(x, y)$

2. $f_r(x) = y \wedge y = 1 \Rightarrow \neg O_r(x, y)$

where 1 is a constant symbol, f_r is a symbol denoting a one-placed function (sensor measurement function), O_r is a symbol denoting a two-placed relation (detection).

Theory $\mathcal{T}_i$: Intensity Sensor. The theory $\mathcal{T}_i$ contains the knowledge about the intensity sensor. It consists of the following single axiom:

$$f_i(x) = i_{shd} \Rightarrow S(x)$$

where i_{shd} - is a constant symbol, f_i is a symbol denoting a one-placed (measurement) function, S is a symbol denoting a one-placed relation (detection of shadow). We have selected a very simple theory of the intensity sensor, since in this example, we use this sensor solely to identify shadows. We extract other relevant information from the range sensor.

Theory $\mathcal{T}_{rt}$: Rectangles and Right Triangles. This theory contains knowledge to distinguish rectangles from right triangles. It includes the following predicates: *segment, constant, length, projection, angle, right-angle, acute-angle, triangle, rectangle.* This knowledge is just a subset of geometry, and thus is not specific to any sensor or a specific scenario. As an example, the *segment* predicate is defined as follows:

$$SEG(x_1, y_1, x_2, y_2) \leftrightarrow \forall_{x_1 \le x \le x_2} O(x, y) \wedge$$

$$y = \frac{y_2 - y_1}{x_2 - x_1} \cdot x + y_1 \wedge \forall_{x_2 \le x \le x_1} \neg O(x, y)$$

Theory $\mathcal{T}_{sh}$: Shadows. This theory contains two axioms:

$$SHD(x_1, x_2) \Leftrightarrow \forall_{x_1 \le x \le x_2} S(x) \wedge \forall_{x_2 \le x \le x_1} \neg S(x)$$

$$TRN(x_1, y_1, x_2, y_2, x_3, y_3) \wedge$$
$$RAN(x_1, y_1, x_2, y_2, x_3, y_3) \wedge$$
$$PRJ(x_2, y_2) = x_l \wedge PRJ(x_3, y_3) = x_r \wedge$$
$$SHD(x_l, x_r) \Rightarrow TSH$$

where SHD is the symbol for a two-placed relation (end points of the shadow), TSH – constant representing "shadow of a triangle", and S is part of the language of the theory $\mathcal{T}_r$. The first axiom states that shadows are continuous, and the second axiom defines conditions for when a shadow can be TSH, it is the shadow of a triangle. The idea behind this axiom is the essence of this fusion problem. It can be understood by analyzing the scenario in Figure 6.

Theory $\mathcal{T}_w$: The World. In our example we presume that our world can be in three possible states: either it includes a rectangle, or a right triangle, or is empty.

$$\neg(TRN \wedge REC)$$

$$(TRN \vee REC) \wedge \neg TSH \Rightarrow REC$$

Goal: $\mathcal{G}$ The goal of the system is to find our which of the following four situations is the case in the world: (1) there is only a rectangle in the world ($\neg TRN \wedge REC$), (2) there is only a right triangle in the world ($TRN \wedge \neg REC$), (3) there is either a single rectangle or a single triangle in the world ($\neg TRN \wedge \neg REC$) (4) the world contains no objects ($TRN \vee REC$).

6.2 Formalization of Fusion

The specification of the rectangle/triangle recognition system was developed in Slang, the language used by the formal method tool, Specware. Both Specware and the underlying its implementation category theory are described in [5]. The structure of the resulting specification is shown in Figure 7.

The specification was developed in a bottom-up fashion. In the first step we developed the specification XREAL. This is an extension of the theory of real numbers (REAL). REAL is one of the theories that is provided with the library of Specware. We needed some additional functions and thus we needed to extend this theory. The arrow from REAL to XREAL is called *import* in Specware. It is an

extension of the category theory concept of *colimit* described in [5].

In the next step, theories of the range sensor, $\mathcal{T}_r$, and of the intensity sensor, $\mathcal{T}_i$, described in Section 6.1, are encoded in Slang. Both theories need to import XREAL. In the Specware implementation they are called RANGE-SENSOR and INTENSITY-SENSOR, respectively. In a similar manner, the RECT-TRIAN specification is created; it imports RANGE-SENSOR and encodes the axioms of the theory $\mathcal{T}_{rt}$. SHADOW imports INTENSITY-SENSOR and encodes the theory $\mathcal{T}_{sh}$.

The next level specification, RECT-TRIAN-SHADOW, is the main fusion block in this whole specification. Here the two theories, RECT-TRIAN and SHADOW, are "glued" together along the common part - the real numbers. In the diagram of Figure 7 this common part is shown as the ONE-SORT specification. The sort defined in this specification serves as the common base that unifies the real numbers from the other two specifications. At the same time all the axioms of the two component specifications are mapped into one set of axioms. Then the sort and the operations of this specification are used to extend the colimit by adding additional axioms specified by the theory $\mathcal{T}_{sh}$.

6.3 Reasoning about the Fusion System

The specification described above can be used for reasoning about the fusion system being specified. For instance, we can reason about the goals of the system. Towards this end, we would have to submit candidate theorems (queries) to a theorem prover and ask whether they could be proven within the theory presented by the specification. In the stage of specification development, such queries could be submitted by either the users (customer side) or by the specification developers (developer side). First, one would need to choose one of the goals from $\mathcal{G}$. The preferable goals are $\{\neg TRN \wedge REC\}$ and $\{TRN \wedge \neg REC\}$, since

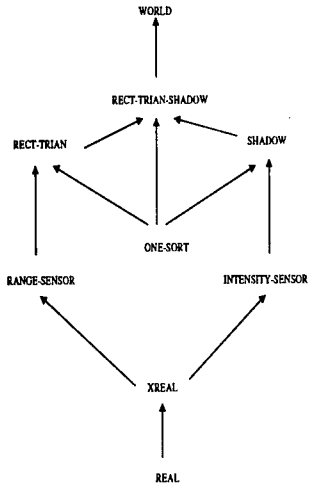


Figure 7: Diagram of the Fusion System

the success of one of these goals means a precise classification of the object in the scene. The goal $\{\neg TRN \wedge \neg REC\}$ is at the same level of detail. The goal $\{TRN \vee REC\}$ is less specific, since its success means that there is an object in the world, but it is not clear whether it is a rectangle or a triangle. In addition to goals, some information about the inputs, or ranges of inputs, would need to be entered into the system, in order for the prover to resolve the validity of a theorem. The goal is posted to the top level system, WORLD. Since this specification (theory) uses terms from the imported specification, the query is propagated down to that specification, and the process continues until all the truth values can be resolved.

Another application of such a specification is to use it for implementing the system. This can be achieved in the process called *refinement*. In this process, the specification goes through a number of refinement steps (called *interpretations*), the final step being translation into a programming language. Specware supports such a software development process.

Once the system is implemented, its operation can be understood as *model checking* (in logical terminology, (cf. [1]). If the system is implemented according to such a rigorous

methodology, as can be easily checked, it will always derive correct decisions, i.e., it will be always right whether the world contains a triangle, a rectangle, one of them, or nothing. The system is not perfect, in the sense that in some situations it will not be able to recognize whether it is a rectangle or a triangle (it will simply say that there is an object in the world: rectangle or triangle). Nevertheless, it can be seen that the fused system will be more powerful than a system with only a range sensor, since it will be able to distinguish between a rectangle and a triangle in all the situations similar to the one shown in Figure 6.

7 Conclusions

In this paper we provided a formal definition of *fusion.* Fusion is treated as a formal operator that is applied to two families of objects, *theories* and their *classes of models* and returns a pair – a *fused theory* and a *a class of fused models.* The general fusion procedure consists of two parallel tasks one of the syntactical nature and the second of the semantical nature. Syntactic Theory Construction inputs a goal theory (with a goal formula in it), theories of sensors and background theories and constructs one fused theory for the whole system in which the goal sentence can be proved. Semantic Model Construction inputs models of the theories utilized in the Syntactic Theory Construction task and generates a class of models for the fused theory.

The goal of our research is to find various schemes for performing fusion and to find computationally efficient algorithms to achieve this goal. In this paper we showed an example of developing a fused theory, i.e., of the Syntactic Theory Construction. Since we used category theory as our mathematical basis, and Specware as our implementation tool, the correctness of the resulting specification and of the existence of the properties of the specification are guaranteed by the formal semantics of Specware and of the Specware theorem prover.

Formal specifications of fusion systems, like the one described in this paper, can serve two purposes. For one, we can reason about various properties of such specifications when we are specifying such systems. This is a very valuable feature, since errors discovered in the specification phase of system development are much cheaper to eliminate than in the later stages. For instance, the same error discovered after deployment of a system can cost hundreds of thousands times more. The other purpose is that such specifications can be transformed into code through the process of refinement. This process guarantees that the specification is implemented correctly. This does not imply, however, that the specification is correct, since this decision depends on the specifier and the user to make. However, having a formally defined specification certainly makes such a process much more reliable and robust.

References

[1] C. C. Chang and H. J. Keisler. *Model Theory.* North Holland, Amsterdam, New York, Oxford, Tokyo, 1992.

[2] D. L. Hall. *Mathematical Techniques in Multisensor Data Fusion.* Artech House, Boston - London, 1992.

[3] R. C. Luo and M. G. Kay. Multisensor integration and fusion in intelligent systems. *IEEE Transactions on Systems, Man and Cybernetics*, 19-5:901–931, 1989.

[4] Y. V. Srinivas and R. Jullig. Specware^{TM}: Formal support for composing software. Technical Report KES.U.94.5, Kestrel Institute, 1994.

[5] S. A. DeLoach and M. M. Kokar. Category theory approach to fusion of wavelet-based features. In *Proceedings of the Second International Conference on Information Fusion*, 1999.

Formally Derived Characterization of the Performance of α-β-(γ) Filters

D. Tenne * T. Singh †
Center for Multisource Information Fusion
SUNY at Buffalo
Buffalo, NY 14260, U.S.A

Abstract *This paper discusses in detail the α-β-γ filter which is a sampled data target tracker which can asymptotically track a constant acceleration target. The α-β-γ parameters are studied to characterize the stability of the filter and its performance viz its noise ratio. A closed form equation for the mean square response of the system to white noise is derived for an α-β-γ filter. It is also shown that the results of the noise ratio for the α-β filter are different from those presented by other researchers.*

Keywords: Trackers, Stability, Noise ratio

1 Introduction

There exists a significant body of literature which addresses the problem of *track-while-scan systems* [1] [2] [3] and [4]. Sklansky [1] in his seminal paper analyzed the behavior of an $\alpha - \beta$ filter. His analysis of the range of values of the $\alpha - \beta$ smoothing parameters which resulted in a stable filter constrained the parameters to lie within a stability triangle. He also derived closed form equations to relate the smoothing parameters for critically damped transient response and the ability of the filter to smooth white noise, using a figure of demerit which was referred to as the *noise ratio*. Finally he proposed, via a numerical example, a procedure to optimally select the $\alpha-\beta$ parameters to minimize a performance index, which is a function of the noise-ratio and the tracking error for a specific maneuver. Following his work, Benedict and Bordner [3] used calculus of variations to solve for an optimal filter which minimizes a cost function, which is a weighted function of the noise smoothing and the transient (maneuver following) response. They show that the optimal filter is coincident with an α-β filter with the constraint that $\beta = \alpha^2/(2-\alpha)$.

Numerous researchers using assumptions of the noise characteristics develop optimal filters [5], [6] and [7] which are commonly called Kalman Filters. Those filters were first introduced in the 60's by Kalman and Bucy [8], [9].

In this paper, a detailed analysis of the α-β-γ filter is carried out. Section 2 discusses the bounds on the smoothing parameters for a stable filter. This is followed by a closed form derivation of the noise ratio for the α-β-γ filter in Section 3. The results of this paper can be used to solve for optimal filter parameters for specific maneuvers given the measurement characteristic of the sensors. It also provides bounds on the smoothing parameters which can be used in adaptive filters. The paper concludes with some remarks in Section 4.

*Graduate Student, Department of Mechanical and Aerospace Engineering
†Assistant Professor, Department of Mechanical and Aerospace Engineering

2 Stability Analysis

2.1 α-β-γ Tracker

The α-β-γ tracker is an one-step ahead position predictor that uses the current error called the innovation to predict the next position. The innovation is weighted by the smoothing parameter α β, and γ . These parameters influence the behavior of the system in terms of stability and ability to track the target. Therefore, it is important to analyze the system using control theoretic aspects to gauge stability and performance. The prediction equation of the α-β-γ filters are

$$x_p(k+1) = x_s(k) + Tv_s(k) \quad (1)$$

and

$$v_p(k+1) = v_s(k) + Ta_s(k)\ , \quad (2)$$

where the smoothed kinematic variables are calculated by weighting the innovation as follows:

$$x_s(k) = x_p(k) + \alpha(x_o(k) - x_p(k)) \quad (3)$$

$$v_s(k) = v_p(k) + \frac{\beta}{T}(x_o(k) - x_p(k)) \quad (4)$$

$$a_s(k) = a_s(k-1) + \frac{\gamma}{2T^2}(x_o(k) - x_p(k)) \quad (5)$$

The transfer function of the α-β-γ tracker can now be represented in the z-domain as

$$G(z) = \frac{\alpha + (-2\alpha - \beta + \frac{1}{4}\gamma)z + (\alpha + \beta + \frac{1}{4}\gamma)z^2}{z^3 + (\alpha + \beta + \frac{1}{4}\gamma - 3)z^2 + (-2\alpha - \beta + \frac{1}{4}\gamma + 3)z + \alpha - 1} \quad (6)$$

which reduces to the popular α-β tracker when γ is zero resulting in the transfer function

$$G(z) = \frac{x_p}{x_o} = \frac{\alpha(z-1) + \beta z}{z^2 + (\alpha + \beta - 2)z + (1-\alpha)}. \quad (7)$$

To determine the bounds on α, β and γ which guarantee stability, we exploit the Jury's stability test which is described next.

2.2 Jury's Stability Test

The Jury's Stability Test can be used to analyze the stability of the system without explicitly solving for the poles of the system. Therefore, it is used to determine the bounds on the parameters which result in a stable transfer function in the z-domain.

For a system with a characteristic equation P(z) = 0, where

$$P(z) = a_0 z^n + a_1 z^{n-1} + ... + a_{n-1} z + a_n \quad (8)$$

and $a_0 > 0$, we construct the table where the first row consists of the elements of the polynomial P(z) in ascending order and the second row consists of the parameters in descending order [10] as shown below where

Table 1: General Form of Jury's Stability Table

Row	z^0	z^1	$\cdots$	z^{n-1}	z^n
1	a_n	a_{n-1}	$\cdots$	a_1	a_0
2	a_0	a_1	$\cdots$	a_{n-1}	a_n
3	b_{n-1}	b_{n-2}	$\cdots$	b_0	
4	b_0	b_1	$\cdots$	b_{n-1}	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
2n-5	p_3	p_2	$\cdots$		
2n-4	p_0	p_1	$\cdots$		
2n-3	q_2	q_1	q_0		

$$b_k = \begin{vmatrix} a_n & a_{n-1-k} \\ a_0 & a_{k+1} \end{vmatrix}, \text{ k = 0, 1, 2, ..., n-1} \quad (9)$$

(10)

(11)

$$c_k = \begin{vmatrix} b_{n-1} & b_{n-2-k} \\ b_0 & b_{k+1} \end{vmatrix}, \text{ k = 0, 1, 2, ..., n-2} \quad (12)$$

$$q_k = \begin{vmatrix} p_3 & p_{2-k} \\ p_0 & p_{k+1} \end{vmatrix}. \text{ k = 0, 1, 2} \quad (13)$$

Note that the last row of the table contains only three elements. The Jury's test states that a system is stable if all of the following conditions are satisfied:

$$|a_n| < a_0 \quad (14)$$

$$P(z)|_{z=1} > 0 \quad (15)$$

$$P(z)|_{z=-1} \begin{cases} > 0 \text{ for even n} \\ < 0 \text{ for odd n} \end{cases} \quad (16)$$

$$\begin{array}{rcl} |b_{n-1}| & > & |b_0| \\ |c_{n-2}| & > & |c_0| \\ & \cdots & \\ |q_2| & > & |q_0| \end{array} \qquad (17)$$

2.3 Stability Bounds

Equation 6 can now be used to determine the bounds of α, β and γ for stability. For this complex system, the *Jury's Stability Test* is used as described in Section 2.2, to determine the region of stability.

Writing the coefficients of the characteristic polynomial in Jury's Table, and calculating the determinants b_2, b_1 and b_0 (Equation 9) yield the following table. The condition $a_0 > 0$ is satisfied since $a_0 = 1$. To satisfy the constraint $|a_n| < a_0$, the coefficients require $|\alpha - 1| < 1$, which is equivalent to

$$0 < \alpha < 2. \qquad (18)$$

Substituting $z = 1$ and applying the constraint $P(z)|_{z=1} > 0$, requires satisfaction of the inequality

$$1 + (\alpha + \beta + \frac{1}{4}\gamma - 3) + (-2\alpha - \beta + \frac{1}{4}\gamma + 3) + \alpha - 1$$

which can be rewritten as

$$\gamma > 0 \qquad (19)$$

Satisfying the constraint $P(z)|_{z=-1} < 0$, for odd n, yields

$$2\alpha + \beta < 4, \qquad (20)$$

which is the same constraint for α and β as in the α-β tracker. The last condition which states $|b_2| > |b_0|$, requires

$$|\alpha(\alpha-2)| > |\alpha(\alpha-2) + \alpha(\beta + \frac{1}{4}\gamma) - \frac{1}{2}\gamma| \qquad (21)$$

Observing Equation (21) and knowing the fact that $\alpha(\alpha-2)$ is always negative within the stability area, we have:

$$\alpha(\beta + \frac{1}{4}\gamma) - \frac{1}{2}\gamma > 0. \qquad (22)$$

This statement leads to the constraint on γ for which the α-β-γ tracker is stable which is

$$\gamma = \frac{4\alpha\beta}{2-\alpha} \qquad (23)$$

Figure (1) illustrates the bounding surfaces which include the stable volume in the $\alpha-\beta-\gamma$ space based on Equation (18), (19), (20) and (23).

Table 2: Jury's Stability Table of the α-β-γ Filter

Row	z^0	z^1	z^2	z^3
1	$\alpha - 1$	$-2\alpha - \beta + \frac{1}{4}\gamma + 3$	$\alpha + \beta + \frac{1}{4}\gamma - 3$	1
2	1	$\alpha + \beta + \frac{1}{4}\gamma - 3$	$-2\alpha - \beta + \frac{1}{4}\gamma + 3$	$\alpha - 1$
3	$\alpha(\alpha-2)$	$\alpha(4 - 2\alpha - \beta + \frac{1}{4}\gamma) - \frac{1}{2}$	$\alpha(\alpha + \beta - 2 + \frac{1}{4}\gamma) - \frac{1}{2}$	

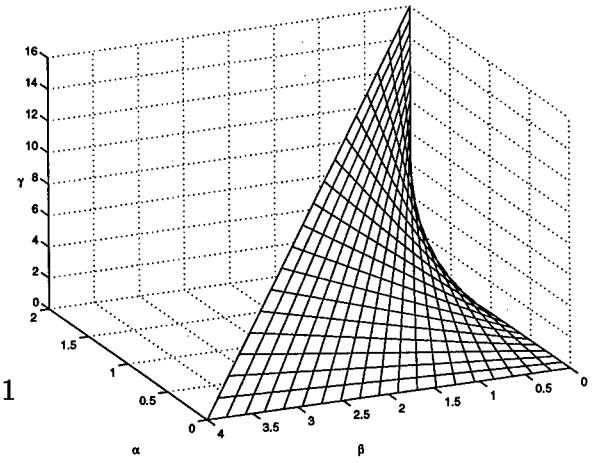


Figure 1: Stability Area of the α-β-γ Tracker

It is desirous to divide the stability volume into regions which are characterized by specific class of transient responses such as, underdamped, overdamped, and critically damped.

However, the difficulty of factorizing the characteristic polynomial of the transfer function in the α-β-γ space prompt us to conceive of a new space which we refer to as the $a-b-c$ space. In this space, the characteristic polynomial is represented as

$$(z+c)(z^2+(a+b-2)z+1-a), \qquad (24)$$

where the second order factor has a form which is identical to the characteristic equation of the $\alpha-\beta$ filter and the third pole is real and is located at $-c$. Comparing the denominator of Equation (6) with Equation (24) the following transformation is derived:

$$\begin{aligned} \alpha &= 1+c(1-a) \\ \beta &= a(1+c)+\frac{1}{2}b(1-c) \qquad (25) \\ \gamma &= 2b(1+c). \end{aligned}$$

The usefulness of this transformation, becomes evident when one derives the stability volume of the α-β-γ filter. Since, c is constrained to lie within -1 and 1, and the $a-b$ space resembles the $\alpha-\beta$ space, the stability volume in the $a-b-c$ space is a prism (Figure (2)) with a triangular cross-section which is derived from the $\alpha-\beta$ filter. Mapping the stability prism in the $a-b-c$ space to the α-β-γ space using Equation (25), we rederive the stability volume illustrated in Figure (1).

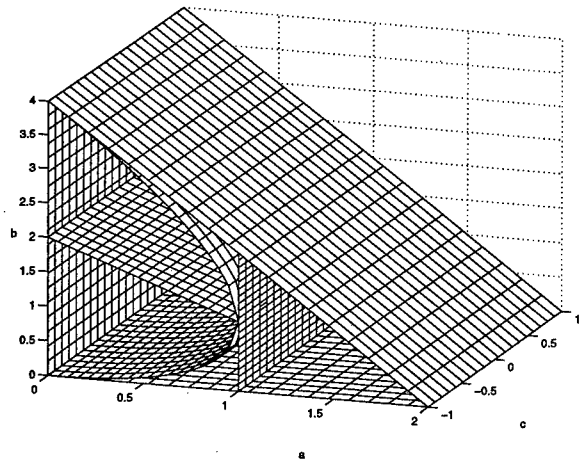


Figure 2: Stability Prism in the a-b-c Space

Since, the pair of poles of Equation (24) which are functions of a and b are responsible for oscillation of the system, the a-b-c space is divided by extruding the lines which divide the stability triangle of the $\alpha-\beta$ filter, in the c dimension. These surfaces, shown in Figure (2), are transformed using Equation (25) to the α-β-γ space. Figure (3) shows the surfaces in the α-β-γ space corresponding to each critically damped surface of the a-b-c space. Figure (4) shows the transformation of the two surfaces dividing the stability area at $a=1$ and $b=2-a$. Observing Figures (3) and

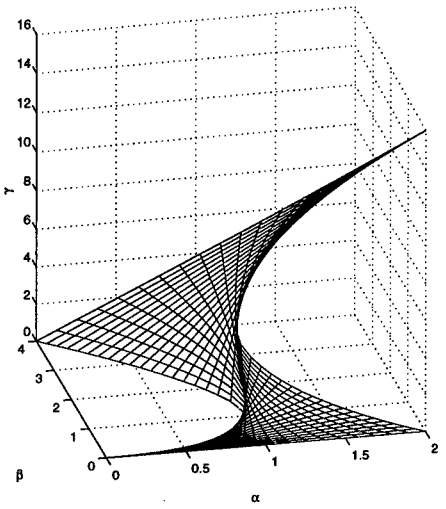


Figure 3: Critical Damped Surfaces of the α-β-γ Space

(4), illustrates the fact that for $\gamma=0$, the third order tracker reduces to the α-β tracker. Substituting $\gamma=0$ in the transfer function (Equation (7)), results in a pole zero cancellation at $z=1$, resulting in a second order tracker. From Equation (25), we can infer that c equals -1 when $\gamma=0$, and furthermore a and b degenerate to α and β. The cross-section at $c=-1$ therefore corresponds to the α-β tracker. Note that $c=0$ does not result in the $\alpha-\beta-\gamma$ degenerating to the $\alpha-\beta$ filter.

3 Noise Ratio

Measurement noise significantly effects the performance of target trackers. It is therefore,

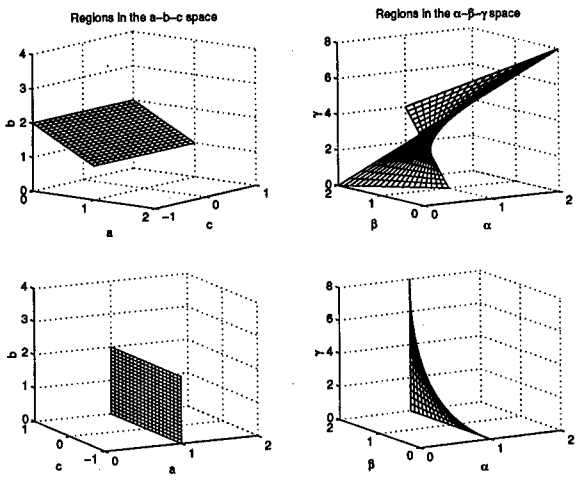


Figure 4: Mapping between a-b-c Space and α-β-γ Space

of interest to characterize the noise filtering strengths of the trackers. In this section, we derive a closed form expression for the noise ratio parameterized in terms of the a-b-c parameters whose relationship to α, β and γ is uniquely known.

Studying the effect of noisy signals requires a metric which measures the influence of the noise on the system. Since, the response of the system to a noisy input, can reflect this influence, the *noise ratio* is defined as the ratio of the root mean square value (RMS) of the system response to the RMS value of the noisy input. The noise ratio is defined as follows.

$$\rho \equiv \sqrt{\frac{\overline{x_p^2(t)}}{\overline{x_o^2(t)}}} \tag{26}$$

Since, we require the tracker to reject measurement noise, a small value of ρ implies an excellent filtering of noise. The mean-square of $x_p(t)$ can be derived in the time domain using the standard integral over time from $-\infty$ to $+\infty$.

$$\overline{x_p^2(t)} = \lim_{T\to\infty} \frac{1}{T}\int_0^T x_p^2(t)dt \tag{27}$$

Assuming the input is known, the response $x_p(t)$ of the system can be evaluated by using the transfer function, $G(s)$.

$$x_p^2(t) = \left[\mathcal{L}^{-1}\{G(s)x_o(s)\}\right]^2, \tag{28}$$

where $\mathcal{L}^{-1}$ represents the inverse Laplace Transformation and $x_o(s)$ is the Laplace transformation of the input. The input noise is assumed to be white noise, so that the value of the noise input at any time is independent of previous values. Therefore, the sampled noise can be evaluated as a train of independent impulses, where the $*$ indicates the sampling.

$$x_o^*(t) = \sum_{n=0}^{N} x_o(nT)\delta(t-nT) \tag{29}$$

Equation (28) yields the following response to the impulse train:

$$x_p^2(t) = \left[\sum_{n=0}^{N} x_o(nT)\delta(t-nT)\right]^2 \left[\mathcal{L}^{-1}\{G(s)\}\right]^2, \tag{30}$$

since, the impulse responses are uncorrelated. Since the sum of the mean square value of the system response to each impulse equals to the mean square value of the system response corresponding to the complete impulse train, the RMS value of $x_p(t)$ can be calculated by first deriving the response to each impulse and then determining the ensemble average.

$$\overline{x_p^2(t)} = \sum_{n=0}^{N} \lim_{T\to\infty} \frac{1}{T}\int_0^T \left[x_o\delta(t-nT)(nT)\mathcal{L}^{-1}\{G(s)\}\right]^2 dt. \tag{31}$$

The averaged impulse value is taken over an arbitrary time interval $nT < t < (n+1)T$ to later derive the ensemble average. Since this interval is one sampling time long, Equation (26) can now be derived as follows:

$$\rho^2 = \frac{1}{T}\lim_{T\to\infty}\int_0^{+\infty} \left[\mathcal{L}^{-1}\{G(s)\}\right]^2 dt \tag{32}$$

Fortunately, the definition of the noise ratio reduces to finding the integral of the inverse Laplace transform of the transfer function $G(s)$, of the tracker.

Equation (32) could be solved in the time domain by finding the Laplace inverse of $G(s)$ [1], or by integrating Equation (32) in the discrete domain (z-domain) by rewriting the continuous time integral (Equation (32)) in the

discrete domain as :

$$\rho^2 = \frac{1}{T}\sum_{n=0}^{+\infty} Tg^2(n) = \sum_{n=0}^{+\infty} g^2(n) \qquad (33)$$

Applying Parseval's Theorem [10] and the Residue Theorem to the sum of Equation (33) yield the final expression in the discrete domain.

$$\sum_{n=0}^{+\infty} g^2(n) = \frac{1}{2\pi j}\oint_C G(z)G(z^{-1})z^{-1}\mathrm{d}z$$

$$= \sum_{\nu=1}^{p} \operatorname*{Res}_{z_\nu}\left[G(z)G(z^{-1})z^{-1}\right], \qquad (34)$$

where p is the number of poles on or inside the unit circle. The discrete transfer functions for the α-β-γ tracker (Equation (6)) can be used to solve for the noise ratio. Note, that these transfer functions differ from those used by Benedict and Bordner [3] and by Simpson [11] respectively. Since the transfer function of the α-β-γ tracker can always be reduced to the α-β filter by setting γ to zero, only the noise-ratio of the three parameter filter are derived in the following section.

The derivations of the residues of the α-β-γ filter become difficult since the poles within the unit circle contain three roots. Therefore, it is convenient to solve the residues in the a-b-c space introduced in Equation (25). The transfer function, Equation (6), is rewritten in the a-b-c space as

$$G(z) = \frac{(1+c+a+b)z^2 + (-2+bc-2c+ca-a)z + 1 + c - ca}{(z+c)(z^2+(a+b-2)z+1-a)} \qquad (35)$$

where the poles are decomposed into a set of second order poles and one first order pole such that

$$z_1 = -c \qquad (36)$$

$$z_{2,3} = -\frac{1}{2}(a+b-2) \pm \sqrt{(a+b-2)^2 - 4(1-a)} \qquad (37)$$

The line integral in Equation (34) is carried out along the unit circle, guaranteeing that all stable poles of $G(z)$ lie within and the poles of $G(z^{-1})$ lie outside the unit circle. The three residues lying within the unit circle can be derived as follows:

$$\operatorname*{Res}_{z_\nu} f(z) = \lim_{z\to z_\nu}(z - z_\nu)\cdot f(z). \qquad (38)$$

From Equation (34), the noise-ratio can now be calculated by using the Equations (37) and (38) which leads to the following result:

$$\begin{aligned}
\rho^2 &= \frac{-(4ba^2+2ab^2+4b^2) - 4b(1+c)k_1}{2a(b+2a-4)b + a(1+c)k_2} \qquad (39)\\
k_1 &= k_{11}c^2 + k_{12}c + k_{13}\\
k_{11} &= 4a + 2a^3 - ab + ba^2 - 6a^2\\
k_{12} &= ba^2 + ab^2 - 6ab - 2a^3 + 8a\\
k_{13} &= 6a^2 - 2ab^2 - 4ba^2 + 4a - 2b^2 + 7ab\\
k_2 &= (b+2a-4)(c^2a - ca - c^2 - 2b + bc + 1)
\end{aligned}$$

Constant noise-ratio surfaces are obtained by solving Equation (39) for either a, b or c. A simple solution for $b = f(a, c, \rho)$ exists, which consists of two solutions of b, where one is always outside the stability prism. A typical constant noise-ratio ($\rho^2 = 10$) surface is shown in Figure (5). Applying the transformation of Equation (25) to each point of the constant noise-ratio surface in the a-b-c space yields the constant noise-ratio surface in the α-β-γ space shown in Figure (6).

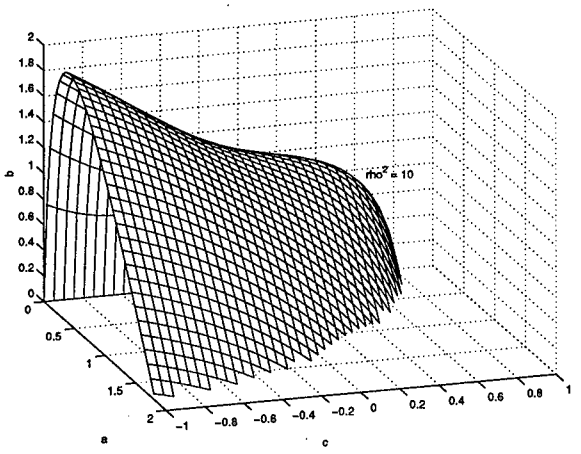


Figure 5: Constant ρ Surface in the a-b-c Space

As mentioned in the section about stability, the α-β-γ filter reduces to a two parameter tracker if γ becomes zero, which corresponds

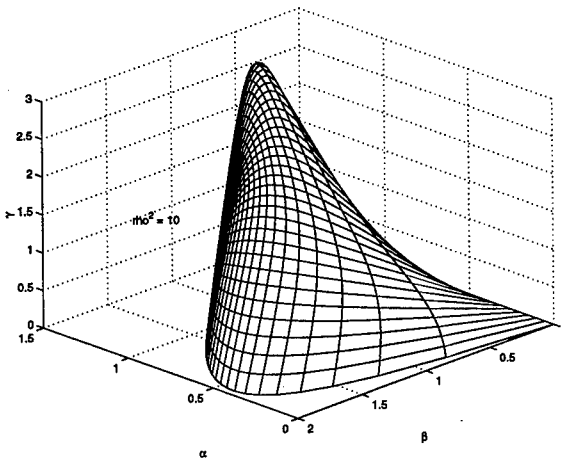


Figure 6: Constant ρ Surface in the α-β-Space

to $c = -1$; and a and b then degenerate to α and β. Thus, the noise-ratio of the α-β-γ filter reduces to the α-β filter by applying these conditions to Equation (39), which is

$$\rho^2 = \frac{2\alpha^2 + \alpha\beta + 2\beta}{\alpha(4 - \beta - 2\alpha)} \qquad (40)$$

The line of constant noise-ratio for the α-β filter is also included in the Figures 5 and 6 by setting $c = -1$ and $\gamma = 0$ respectively. However for greater clarity, the plots of constant noise ratio curves are plotted in the α-β space for various noise ratios as shown in Figure 7. Equation (40) is different from those derived by Sklansky [1] and Benedict and Bordner [3] where, Benedict and Bordner use a different transfer function $G(z)$, to represent the impulse response of the filter.

Table 3: Noise-ratio from different Approaches

Simulation	Sklansky	Benedict & Bordner	Proposed (Eq. 40)
1.9540	1.2058	0.7391	1.9565

To prove the veracity of Equation (40), numerical simulations are carried out. Results of simulating an α-β filter for normally distributed white noise are used to calculate the

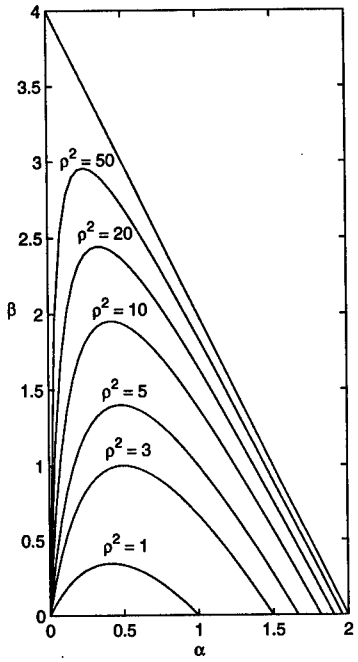


Figure 7: Constant ρ curves in the α-β Space

noise ratio by calculating the ratio of the root-mean-square value of the output and input. The following table displays the simulation result for the parameter set $\alpha = 0.5$ and $\beta = 0.7$. As is clear from the table, the solutions of Sklansky and Benedict and Bordner do not match the results of the simulation, while Equation (40) matches the simulated results.

4 Conclusions

This paper focuses on the design of α-β-γ filters. The issue of determination of stability volume is first addressed. A simple technique to simplify the procedure to determine the stability bounds on the α, β and γ filters is proposed. This includes parameterizing the characteristic equation of the α-β-γ filter via a nonlinear transformation to what is referred to as the $a-b-c$ space. In this space, the characteristic equation appears to be the product of the characteristic equation of an α-β filter and a first order pole which is only a function of

the parameter c. One can now easily determine the bounds on the parameters with knowledge of the bounds on the parameters of an α-β filter. Therefore the stability volume in the a-b-c space is a prism which can be transformed into the α-β-γ space. To quantify the performance of α-β-γ filters, a metric referred to as the noise ratio which is a figure of demerit to represent the noise filtering capability of the tracker is calculated. A closed form solution to the noise ratio is arrived at in the a-b-c space which reduces to the noise ratio for the α-β filter when γ is equated to zero. The resulting solution is shown to be different from that derived in the literature. Numerical simulations are carried out to evaluate the veracity of the derived solution. The information about the stability volume and the noise ratio can be used in the design of $\alpha - \beta - \gamma$ filters.

Acknowledgment

This research was funded by ONR/SPAWAR System Center grant number 150-1284A The authors are grateful to Mr Ed. Jahn and Dr. James Llinas for their input.

References

[1] Jack Sklansky. Optimizing the dynamic parameter of a track-while-scan system. *RCA Laboratories, Princton, N.J.*, June 1957.

[2] Paul R. Kalata. $\alpha - \beta$ target tracking systems: A survey. In *American Control Conference/WM12.* ECE Department, Drexel Univeristy Philadelphia, Pennsylvania, 1992.

[3] T. R. Benedict and G. W. Bordner. Synthesis of an optimal set of radar track-while-scan smoothing equations. In *IRE Transaction on Automatic Control*, volume AC-1, July 1962.

[4] Yaakov Bar-Shalom and Thomas E. Fortmann. *Tracking Data and Association.* Academic Press, Inc., 1988.

[5] Allen J Kanyuck. Transient response of tracking filters with randomly interrupted data. *IEEE Transaction on Aerospace and Electronic Systems*, AES-6(3), May 1970.

[6] C.C. Schooler. Optiaml $\alpha - \beta$ filters for systems with modeling inaccuracies. In *IEEE Transactions on Aerospace and Electronic System*, volume AES-11, November 1975.

[7] Bernard Friedland. Optimal steady-state position and velocity estimation using noisy sampled data. *IEEE Transaction on Aerospace and Electronic Systems*, AES-9(6), November 1973.

[8] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82, March 1960.

[9] R. E. Kalman and R. Bucy. New results in linear filtering and prediction theory. *Journal of Basic Engineering*, 83D, March 1961.

[10] Katsuhiko Ogata. *Discrete-Time Control Systems.* Prentice-Hall,Engelwood Cliffs, New Jersey, University of Minnesota, 1987.

[11] H. R. Simpson. Performance measures and optimization condition for a third order sampled-data tracker. In *IEEE Transaction on Automatic Control*, volume AC-12, June 1962.

Towards a Goal-Driven Autonomous Fusion System

Jerzy A.Tomasik
Universite Blaise Pascal
LLAIC1, BP86 63172 AUBIERE
France

Mieczyslaw M. Kokar
Department of Electrical and Computer Engineering
Northeastern University
Boston, MA 02115, USA

Abstract *This paper addresses the issue of decision fusion when two (or more) sensors and the fusion center have a common language to represent queries and decisions, while each of the sensors has its own interpretation of the formulas of the language. Fusion is achieved through the model-theoretic operation of direct product of models. Since not all (most) formulas are not preserved under the product we need an decision procedure that tell us how to combine decisions from particular sensors into one fused decision. Towards this aim the notion of Galvin system is used. The operation of a decision procedure based on this approach is explained on simple examples. The validity of the solution is formally defined and proved in an appropriate theorem. The main advantages of the approach proposed in this paper are that the decision mechanism is generic, i.e., it can check the validity of any goal formula, and that it is provably correct.*

Keywords: information fusion, formal methods, category theory, model theory

1 Introduction

In this paper we consider a case of decision fusion in which all sensors (two or more) derive decisions that are expressed in a language common to all the sensors. Even though it may seem like a very simple case, it is not quite so, because each of the sensors has its own interpretation of the terms of the language. In other words, for each sensor, there is a (different) model associated with the language. Consequently, the process of fusion (cf. [1]) requires that these different interpretations be taken into account when decisions from different sensors are fused.

We address this problem by fusing the interpretation structures (models) rather than just the decisions. In this paper we use the operation of *product* to combine structures [2]. Unfortunately, in such a case, even if both sensors derive the same decision, it is not necessarily preserved in the product of two models. For instance, the formula

$$\alpha(x,y) \equiv x \cdot y = 0 \Rightarrow (x = 0 \vee y = 0)$$

most typically does not hold in the product. To be more specific, consider two structures A and B such that $A = B = R$, i.e., both are real numbers with two operations - addition and multiplication under usual interpretation. The formula $\alpha(x,y)$ holds in both A and B, since either x or y must be 0 in order for $x \cdot y$ to be zero. We can say then that $A \models \alpha$ and $B \models \alpha$. In the product $A \times B$, however this is not the case. Note first that in the direct product $A \times B$, the zero element, 0, is represented by the pair $0(0,0)$ and if $x = (x_1, x_2)$ and $y = (y_1, y_2)$ are any elements of $A \times B$ we have $x \cdot y = ((x_1, x_2) \cdot (y_1, Y_2)) = (x_1 \cdot y_1, x_2 \cdot y_2)$. It is easy to see that for $x = (0,3)$ and $y = (5,0)$ we have $x \cdot y = (0,0)4$, but neither x or y are equal 0.

Horn formulas [3], on the other hand, are

preserved under products. Even more, Keisler [4] proved that any formula that is preserved under reduced products is equivalent to a Horn formula. Some applications of Horn formulas can be found in [5, 6, 7].

Since, as we stated above, we are interested in fusion by products, our goal in this paper is to deal with more general types of formulas, not necessarily Horn formulas, and thus we need a decision procedure which will allow us to decide when a given formula is preserved in the product of two models. Moreover, our objective is to show how such a decision procedure can be derived automatically, i.e., how to construct *autonomous fusion systems* of this sort, given that the system knows the language in which decisions are expressed.

2 Problem Formulation

We are addressing here the problem of decision fusion. We assume that the goal of the fusion system is to derive a decision φ based upon decisions $\varphi_1, \ldots, \varphi_n$ obtained from n sensors (n decisions based on inputs from sensors). It is assumed that all sensors have the same language and that they interpret information in structures of the same kind of structure. In our example we assume even more - that carriers of models are the same, although in general it is not important. However, the semantical interpretations of the information can be very different. Our goal is to construct a decision procedure which will assert a formula whenever all the sensors report that some witness formulas holds.

We envision a hierarchical scenario in which there is a central fusion unit that collects inputs from all subordinate units (we call them sensors) and then the cenral unit makes a decision. The central unit can send various questions (queries) to the sensors. It is possible since both the central unit and the sensors speak the same language.

To better explain the problem we are addressing, we consider the follwing example scenario (see Figure 1). In this scenario, the goal is to recognize whether a detected object (house) has a gable roof oriented in the East-West direction. Two sensors, N and W (north and west) provide reports to the fusion unit. Suppose one of the terms in the language is $GableEW(x)$, which is one of the *goal formulas* of the system. The fusion center, F, can then send the query to the two sensors, W and N. Both sensors will interpret this formula in their own manner. Sensor W will reply "yes" (or "true") when it sees a triangle. Sensor N, on the other hand, will reply "yes" when it sees a rectangle. The fusion center, F, will conclude GableEW(roof) holds if both sensors say so.

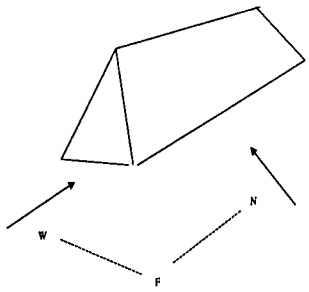


Figure 1: Decision Fusion Scenario

Notice that, as we mentioned above, in a genearal case such a decision procedure is not correct. Our goal in this paper is to propose a solution to this problem. More specifically, we will show how to decide about the truthfulness of such formulas. Since the procedure will allow for automatic answering of such queries, we call this procedure an *autonomous fusion system.*

3 Outline of the Solution

To construct an autonomous fusion system, we use the notion of *Galvin system* (cf. [8, 6]).

In the framework of Galvin systems, we show an algorithm for deriving the goal decision φ. In the first step, for any given goal formula φ, our system constructs a set of formulas S and an operation $\Pi : S^n \rightarrow S$. The operation Π asserts which of the goal formulas hold in the fused structure (in the product), given that formulas $\varphi_1, \ldots \varphi_n$ hold for each of the sensors. The algorithm then computes a set $T \subset S$, such that the goal formula φ is equivalent to the disjunction of all the formulas from T. In the next step, the algorithm computes $\bar{\varphi} = \Pi(\varphi_1, \ldots, \varphi_n)$ and checks whether this formula is in T. If it is, then the decision φ holds, otherwise it does not. It is clear that since $\bar{\varphi}$ is a disjunct in the set T, $\bar{\varphi}$ implies the goal formula φ.

Popular wisdom has it that when two sensors derive the same decision, then the decision must hold in reality. In this paper we show how to rationalize such a rule, i.e., when to accept such a rule in decision fusion and when not to. We have already showed simple examples of decision fusion for both when this popular wisdom rule should be used and when it should not be used. In the following we show how the Galvin system approach works on the scenario of Figure 1.

First of all, the fusion center F must select a query, or a *goal formula.* It is obviously related to its higher-level goals. In the next step it analyzes the syntax of the formula. Among others, it identifies *atomic formulas* within the goal formula. (Note, for now we are dealing only with open formulas, i.e., formulas without quantifiers.) In the case of $\varphi \equiv GableEW$, the goal formula consists of one atom. Based upon this analysis, F constructs the set S. In this case $S = \{GableEW, \neg GableEW\}$. In the case of open formulas, S contains all atomic formulas, all conjunctions, and their negations. In the next step, F constructs the mapping Π. In this example it is defined as:

$$\Pi(\varphi, \varphi) = \varphi$$

$$\Pi(\neg\varphi, \varphi) = \Pi(\varphi, \neg\varphi) = \Pi(\neg\varphi, \neg\varphi) = \neg\varphi$$

The set T in this case consists of only one element, $T = \{GableEW\}$. To make a specific decision, the system takes the answers to the query, computes the value of Π and checks whether the result is in T. In this example, the only case that a result is in T is when both sensors say "yes".

4 Proof of the Solution

In this section we present a formal definition of the Galvin information fusion system [8, 6] that was informally described above. We also provide a proof that a decision can be reached for any goal formula.

Definition 4.1 *A* Galvin Information Fusion System IFS *(with variables $v_1, \ldots v_n$) is a pair (S, Π), where:*

(i) S is a finite set of formulas with variables in $\{v_1, \ldots v_n\}$.

(ii) Π is a commutative and associative operation on S (i.e. (S, Π) is a commutative semigroup).

(iii) For any structure $\mathfrak{A}$ and $a_1, \ldots a_n \in \mathfrak{A}$, there is exactly one formula $\alpha \in S$ such that $\mathfrak{A} \models \alpha[a_1, \ldots a_n]$.

(iv) For any structures $\mathfrak{A}$ and $\mathfrak{B}$ and any elements $a_1, \ldots a_n \in \mathfrak{A}$ and $b_1, \ldots b_n \in \mathfrak{B}$ if for some $\alpha, \beta \in S$ we have $\mathfrak{A} \models \alpha[a_1, \ldots a_n]$ and $\mathfrak{B} \models \beta[b_1, \ldots b_n]$ then $\mathfrak{A} \times \mathfrak{B} \models \Pi(\alpha, \beta)[\langle a_1, b_1\rangle, \ldots \langle a_n, b_n\rangle]$.

Theorem 4.2 — *For any goal formula $\varphi = \varphi(x_1 \ldots x_n)$ we can effectively construct a Galvin IFS (S, Π) with variables $x_1, \ldots x_n$ and a set $T \subseteq S$ such that $\bigvee T \leftrightarrow \varphi$ is a tautology.*

Proof: We will proceed by induction. Let φ be an open formula and $\varphi_1, \ldots, \varphi_k$ be all atomic formulas occurring in φ. Let S consists of $\mathbf{2}^k$ formulas of the form $\psi_1 \wedge \ldots \wedge \psi_k$, where each ψ_i is either φ_i or $\neg\varphi_i$. Suppose $\alpha = \psi'_1 \wedge \ldots \wedge \psi'_k$, $\beta = \psi'' \wedge \ldots \wedge \psi''_k$. Take $\Pi(\alpha, \beta) = \psi_1 \wedge \ldots \wedge \psi_k$ where $\psi_i = \varphi_i$ if $\psi'_i = \psi''_i = \varphi_i$ or $\psi_i = \neg\varphi_i$ otherwise. Now Π satisfies all of our requirements and it is easy to see that (S, Π)

is then a Galvin IFS and we can find the set $T \subseteq S$ such that $\bigvee T \leftrightarrow \varphi$ is a tautology.

Let us remark now that, if (S, Π) and T satisfy our theorem for φ, then (S, Π) and $S - T$ satisfy our theorem for $\neg\varphi$.

Let $\langle S_1, \Pi_1\rangle$ and T_1 satisfy the theorem for φ_1 and let $\langle S_2, \Pi_2\rangle$ and T_2 do the same for φ_2. Let us define $S = \{\alpha \wedge \beta : \alpha \in S_1 \text{ and } \beta \in S_2\}$, $T = \{\alpha \wedge \beta : \alpha \in T_1 \text{ or } \beta \in T_2\}$, $\Pi(\alpha_1 \wedge \beta_1, \alpha_2 \wedge \beta_2) = \Pi_1(\alpha_1, \alpha_2) \wedge \Pi_2(\beta_1, \beta_2)$. Then it is easy to check that (S, Π) and T satisfy the theorem for $\varphi_1 \vee \varphi_2$.

Finally let $\varphi = \exists x_0 \varphi_1$, and let (S_1, Π_1) and T_1 satisfy the theorem for φ_1. We will find (S, Π) and T for φ. Let $S = \{\alpha_X : X \subseteq S_1\}$ where $\alpha_X = \bigwedge\{\exists x_0\gamma : \gamma \in X\} \wedge \bigwedge\{\neg\exists x_0\gamma : \gamma \in S_1 - X\}$. Then it is easy to see that S satisfies the conditions (i) and (iii).

Let $\Pi(\alpha_X, \alpha_Y) = \alpha_Z$, where $Z = \{\Pi_1(\gamma, \delta) : \gamma \in X \text{ and } \delta \in Y\}$. Obviously Π satisfies (ii). We will show that Π satisfies (iv).

Indeed, suppose $\mathfrak{A} \models \alpha_X[a_1, \ldots, a_n]$ and $\mathfrak{B} \models \alpha_Y[b_1, \ldots, b_n]$. To prove (iv), we will show that for any $\beta \in S_1$, $\mathfrak{A} \times \mathfrak{B} \models (\exists x_0\beta)[\langle a_1, b_1\rangle, \ldots \langle a_n, b_n\rangle]$ iff $\beta \in Z$.

Let $\beta \in Z$, then there is $\gamma \in X$ and $\delta \in Y$ such that $\beta = \Pi_1(\gamma, \delta)$. Moreover since $\mathfrak{A} \models \alpha_X[a_1, \ldots, a_n]$, we have $\mathfrak{A} \models (\exists x_0\delta)[a_1, \ldots, a_n]$ and in the same way $\mathfrak{B} \models (\exists x_0\delta)[b_1, \ldots, b_n]$. Thus there are $a_0 \in A$ and $b_0 \in B$ such that $\mathfrak{A} \models \delta[a_0, \ldots, a_n]$ and $\mathfrak{B} \models \delta[b_0, \ldots, b_n]$. Consequently $\mathfrak{A} \times \mathfrak{B} \models \Pi_1(\gamma, \delta)[\langle a_0, b_0\rangle, \ldots, \langle a_n, b_n\rangle]$, but $\beta = \Pi_1(\gamma, \delta)$,
whence $\mathfrak{A} \times \mathfrak{B} \models \beta[\langle a_0, b_0\rangle, \ldots, \langle a_n, b_n\rangle]$ and $\mathfrak{A} \times \mathfrak{B} \models (\exists x_0\beta)[\langle a_1, b_1\rangle, \ldots, \langle a_n, b_n\rangle]$.

Conversely, let $\mathfrak{A} \times \mathfrak{B} \models (\exists x_0\beta)[\langle a_1, b_1\rangle, \ldots, \langle a_n, b_n\rangle]$. Then there is $\langle a_0, b_0\rangle \in A \times B$ such that $\mathfrak{A} \times \mathfrak{B} \models \beta[\langle a_0, b_0\rangle, \ldots, \langle a_n, b_n\rangle]$. Let $\gamma \in S_1$ be such that $\mathfrak{A} \models \gamma[a_0, \ldots a_n]$ and $\delta \in S_1$ be such that $\mathfrak{B} \models \delta[b_1, \ldots, b_n]$. Then $\mathfrak{A} \models (\exists x_0\gamma)[a_1, \ldots, a_n]$ and by definition of α_X, $\gamma \in X$. In the same way $\delta \in Y$. Moreover $\Pi_1(\gamma, \delta) = \beta$, thus $\beta \in Z$ and Π satisfies (iv).

This completes the proof.

5 Conclusions

In this paper we addressed the issue of decision fusion. We assumed the situation in which various sensors and a central decision fusion system share a common language. More precisely, the syntax of the language is common to all parties. But the interpretation of particular symbols is different for each sensor. The fusion process is based on the model-theoretic operation of direct product of models. Accordingly, the fusion process takes all the decisions from the particular sensors and then derives a positive decision only if all the sensors agree. In this scenario, the fusion center must be able to send queries to the sensors. This query is dependent on its goal represented by a goal formula. In case of a complex query, the fusion center analyzes the structure of the goal formula and derives a decision derivation procedure. This decision process is based upon the notion of Galvin system. In this paper we showed simple examples of how such a system works. We also showed the formal derivation of the correctness of such procedure.

Decision fusion is important in many applications. The scenario in which particular sensors have their own interpretations is quite typical. For instance, a radar "sees" an object in a different way than a vision camera, an infrared camera or an ultrasound. All of these sensors, however, are used in target detection. Decisions from such sensors need to be fused. Typically, fusion algorithms are constructed around a specific set of goal formulas. If an additional formula needs to be added, the system must be redesigned and reimplemented. The approach presented in this paper resolves such a problem by proposing a generic decision mechanism that can check the validity of any goal formula. The main advantage of this mechanism is that, unlike heuristic rule-based approaches, it is provably correct.

References

[1] M. M. Kokar, J. A. Tomasik, and J. Weyman. A formal approach to information fusion. In *Proceedings of the Second International Conference on Information Fusion*, 1999.

[2] C. C. Chang and H. J. Keisler. *Model Theory*. North Holland, Amsterdam, New York, Oxford, Tokyo, 1992.

[3] A. Horn. On sentences which are true of direct unions of algebras. *Journal of Symbolic Logic*, 16:14–21, 1951.

[4] H. J. Keisler. Reduced products and horn classes. *Trans. Amer. Math. Soc. Ser.2*, 81:307–328, 1965.

[5] J. A. Makovsky. Why horn formulas matter in computer science: Initial structure and generic examples. In *CAAP'85 Arbres en Algebre et Programmation 10*, pages 374–385, 1985.

[6] J. Tomasik. On products of neat structures. *Annals of Mathematical Logic*, 36:12–16, 1976.

[7] E. A. Palyutin, J. Saffe, and S. S. Starchenko. Models of superstable horn theories. *Algebra i Logika*, 24:278–326, 1985.

[8] F. Galvin. Horn sentences. *Annals of Mathematical Logic*, 1:389–422, 1970.

Session TB1
Radar and Communication Applications of Fusion
Chair: Rick Blum
Lehigh University, PA, USA

Clarifying the Conditions for Neyman-Pearson Optimum Distributed Signal Detection

Qing Yan and Rick S. Blum *
EECS Department
Lehigh University
Bethlehem, PA 18015

Abstract - *A procedure for finding the optimum distributed sensor detectors for cases with statistically dependent observations is described. This procedure is based on a theorem proven in this paper. These results clarify and correct a number of possibly misleading discussions in the existing literature.*

Keywords: distributed signal detection, decentralized detection, Neyman-Pearson criterion, multisensor, correlated observations.

1 Introduction

Consider the design of an N-sensor distributed detection scheme, which is to decide between a simple signal-present alternative hypothesis H_1 and a simple null hypothesis H_0. Each sensor has an associated processor which makes a decision based only on the observations obtained from the sensor. The sensor processors transmit their decisions to a single central fusion center where an overall decision is made. A particular value $\mathbf{x_k}$ of the random vector $\mathbf{X_k}$ is observed at the k^{th} sensor, $k = 1, \ldots, N$, where $\mathbf{x_k}$ consists of a set of m_k real scalar observations. We consider the case where the $\mathbf{X_1}, \ldots, \mathbf{X_N}$ may not be independent. The final binary decision in our distributed detection scheme is denoted by the random variable U_0, with a particular realization of U_0 denoted by u_0 and where $u_0 = 0$ corresponds to a decision for H_0 and $u_0 = 1$ corresponds to a decision for H_1. U_k is the random variable which describes the decision made at the k^{th} sensor. A particular value for U_k is denoted by u_k which may take on only the values 0 or 1 (binary sensor decisions). We let $\gamma_0(\mathbf{u})$ denote the probability that we decide for $U_0 = 1$ for a given set of sensor decisions $\mathbf{u} = (u_1, \ldots, u_N)$. We let $\gamma_k(\mathbf{x_k})$ denote the probability we decide for $U_k = 1$ for a given observation $\mathbf{x_k}$. A complete set of sensor rules and fusion rule are described by $\gamma = (\gamma_0, \gamma_1, \ldots, \gamma_N)$.

Let us focus on the Neyman-Pearson criterion. Specifically, denote the problem of interest as NP which is defined as finding a γ that satisfies

$$NP: \quad \max_{\gamma} P_d(\gamma) \quad \text{subject to the constraint } P_f(\gamma) = \alpha$$

where $P_d(\gamma) = Prob(U_0 = 1|H_1)$ is the probability of detection obtained when γ is used, $P_f(\gamma) = Prob(U_0 = 1|H_0)$ is the probability of false alarm obtained when γ is used, and $0 \leq \alpha \leq 1$. Specifying the forms of NP optimum distributed detection schemes can be extremely difficult [1], especially for cases with dependent observations from sensor to sensor where the optimum sensor test statistics are not generally likelihood ratios.

*This paper is based on work supported by the Office of Naval Research under Grant No. N00014-97-1-0774 and by the National Science Foundation under Grant No. MIP-9703730

2 Optimum Sensor Tests

Let us assume the $\mathbf{X_k}$, $k = 1,\ldots,N$ each have probability density functions (pdfs) $f_{\mathbf{X_k}}(\mathbf{x_k}|H_j)$, $j = 0,1$. Define

$$D_{jk}(\mathbf{x_k}) = f_{\mathbf{X_k}}(\mathbf{x_k}|H_j)$$
$$\sum_{\widetilde{u_k}}\Big[Prob(U_0 = 1|\widetilde{U_k} = \widetilde{u_k}, U_k = 1) -$$
$$Prob(U_0 = 1|\widetilde{U_k} = \widetilde{u_k}, U_k = 0)\Big]$$
$$Prob(\widetilde{U_k} = \widetilde{u_k}|\mathbf{X_k} = \mathbf{x_k}, H_j), \qquad (2.1)$$

for $j = 0,1$ and $k = 1,\ldots,N$ where $\widetilde{u_k}$ stands for a specific value of the random vector $\widetilde{U_k}$ of sensor decisions excluding the k^{th} so that $\widetilde{U_k} = (U_1, U_2, \ldots, U_{k-1}, U_{k+1}, \ldots U_N)$ and $Prob(U_0 = 1|\widetilde{U_k} = \widetilde{u_k}, U_k = u_k) = Prob(U_0 = 1|\underline{U} = \underline{u})$ describes the fusion rule γ_0. The sum in (2.1) is over all values of $\widetilde{u_k}$ (for example, if $N = 2$ and $k = 1$, then $\widetilde{u_k} = u_2$ and the sum is over $u_2 = 0,1$). Note that the conditional probability $Prob(\widetilde{U_k} = \widetilde{u_k}|\mathbf{X_k} = \mathbf{x_k}, H_j)$ is defined as a limit as the conditioning event shrinks to a point.

Using these definitions, we present Theorem 1, proven in [2], which gives a set of necessary conditions for an optimum sensor rule given the fusion rule and other sensor rules are fixed.

Theorem 2.1 *Given a fusion rule and a set of sensor processor rules at all but the kth sensor and a statistical description for* $\mathbf{X}_1, ..., \mathbf{X}_N$ *under* $\mathbf{H}_0$ *and* $\mathbf{H}_1$ *such that*
1) $\mathbf{X}_k$ *is an* m_k*-dimensional random vector with a probability density function* $f_{\mathbf{X_k}}(\mathbf{x_k}|H_j)$ *with no point masses of probability under either hypothesis* $j = 0,1$*;*
2) $D_{1k}(\mathbf{X_k})/D_{0k}(\mathbf{X_k})$ *is a continuous scalar random variable with a probability density function with no point masses of probability under either hypothesis;*
3) $D_{0k}(\mathbf{x_k}) = 0$ *only if* $f_{\mathbf{X_k}}(\mathbf{x_k}|H_0) = 0$.
Then
1) A γ_k *of the form*

$$\gamma_k(\mathbf{x_k}) = Prob(U_k = 1|\mathbf{X_k} = \mathbf{x_k})$$
$$= \begin{cases} 1, & \text{if } D_{1k}(\mathbf{x_k}) > \lambda_k D_{0k}(\mathbf{x_k}) \\ 0, & \text{if } D_{1k}(\mathbf{x_k}) < \lambda_k D_{0k}(\mathbf{x_k}) \end{cases} \qquad (2.2)$$

will satisfy NP for the given fusion rule and the given set of sensor processor rules provided there exists some rule γ_k' *that will provide the required overall false alarm probability* α *for the given fusion rule and the given set of sensor processor rules. The event* $D_{1k}(\mathbf{x_k}) = \lambda_k D_{0k}(\mathbf{x_k})$*, which occurs with zero probability, can be assigned* $\gamma_k = 0$ *or* $\gamma_k = 1$.
2) Any rule that satisfies NP for the given fusion rule and the given set of sensor processor rules must be of this form except possibly on a set having zero probability under $\mathbf{H}_0$ *and* $\mathbf{H}_1$.

Theorem 2.1 gives the best form of any sensor detector, given all the other sensors and the fusion rule are fixed. Thus it gives conditions for person-by-person optimality. No better rule can be found by changing one sensor at a time. However, if two sensors are changed at the same time, it is possible that performance can be improved. Next, we will show that by considering changes in two sensors at the same time we can put further restrictions on the conditions produced by Theorem 2.1 such that $\lambda_1 = \lambda_2 = \cdots = \lambda_N$ will produce an optimum solution.

Theorem 2.2 *Under the same assumptions in Theorem 2.1 and if* $f_{\mathbf{X_k}}(\mathbf{x_k}|H_j) > 0 \forall \mathbf{x_k}, j = 0,1$*, the best performance can only be obtained with a set of sensor rules* γ_k *described in Theorem 2.1 with* $\lambda_1 = \lambda_2 = \cdots = \lambda_N$*. Thus, under these conditions only a set of sensor processor rules* $(\gamma_1, \gamma_2, \cdots, \gamma_N)$ *of the form*

$$\gamma_k(\mathbf{x_k}) = Prob(U_k = 1|\mathbf{X_k} = \mathbf{x_k})$$
$$= \begin{cases} 1, & \text{if } D_{1k}(\mathbf{x_k}) > \lambda D_{0k}(\mathbf{x_k}) \\ 0, & \text{if } D_{1k}(\mathbf{x_k}) < \lambda D_{0k}(\mathbf{x_k}) \end{cases} \qquad (2.3)$$

will satisfy NP for the given fusion rule. The event $D_{1k}(\mathbf{x_k}) = \lambda D_{0k}(\mathbf{x_k})$*, which occurs with zero probability, can be assigned* $\gamma_k = 0$ *or* $\gamma_k = 1$*. In nonsingular detection cases with* $f_{\mathbf{X_k}}(\mathbf{x_k}|H_j) = 0$ *for some* $\mathbf{x_k}$*, there can be*

other solutions which appear to be different which are also optimum.

Outline of the Proof: First assume that $f_{X_k}(x_k|H_j) > 0$ for all x_k, $k = 1,\ldots,N$ and $j = 0,1$. Then, we only need to show that for a fixed fusion rule, the set of sensor processor rules $(\gamma_1,\gamma_2,\cdots,\gamma_N)$ can not be optimum if any two of them, γ_m and γ_n $(1 \le m,n \le N, m \ne n)$, take different parameters λ_m and λ_n. We prove this by contradiction.

Let A_i denote the decision region of sensor i. Thus $u_i = 1$ if $x_i \in A_i$. Let $\Omega(A_1,\cdots,A_n,\cdots,A_m,\cdots,A_N)$ denote the scheme using $A_1,\cdots,A_n,\cdots,A_m,\cdots,A_N$. Let $(\gamma_1,\gamma_2,\cdots,\gamma_N)$ denote a set of rules for which γ_m and γ_n $(1 \le m,n \le N, m \ne n)$ take different parameters λ_m and λ_n. A set of rules which is better than $(\gamma_1,\gamma_2,\cdots,\gamma_N)$ in NP sense could be found by using the following steps. We assume $\lambda_m > \lambda_n$, as the proof for the opposite case $\lambda_m < \lambda_n$ is in fact the same.

Define $\Delta A_k^{(a,b)} = \{x_k | aD_{0k}(x_k) \le D_{1k}(x_k) < bD_{0k}(x_k), D_{0k}(x_k) > 0\}$, and $\Delta B_k^{(a,b)} = \{x_k | bD_{0k}(x_k) \le D_{1k}(x_k) < aD_{0k}(x_k), D_{0k}(x_k) < 0\}$.

First we change the parameter λ_m of the decision rule of sensor m by a small amount, i.e. $\lambda_m^* = \lambda_m - \varepsilon$, $\varepsilon > 0$, thus the decision region of sensor m will be A_m^*, where $A_m^* = (A_m \cup \Delta A_m^{(\lambda_m-\varepsilon,\lambda_m)}) \cap (\Delta B_m^{(\lambda_m-\varepsilon,\lambda_m)})^c$. Consequently, P_f^* with $\Omega(A_1,\cdots,A_m^*,\cdots,A_N)$ is given by (see (6)-(9) in [2])

$$P_f^* = P_f + \int_{\Delta A_m^{(\lambda_m-\varepsilon,\lambda_m)}} D_{0m}(x_m)dx_m - \int_{\Delta B_m^{(\lambda_m-\varepsilon,\lambda_m)}} D_{0m}(x_m)dx_m. \quad (2.4)$$

From the definition we see that $D_{1n}(x_n)/D_{0n}(x_n)$ is the combination of continuous functions of λ_m, thus $D_{1n}(x_n)/D_{0n}(x_n)$ itself is a continuous function of λ_m.

Therefore, there exists a minimum $\delta > 0$ so that

$$|D_{1n}^*(x_n)/D_{0n}^*(x_n) - D_{1n}(x_n)/D_{0n}(x_n)| < \delta \quad for\ all\ x_n \in \Delta A_n^{(\lambda_n,\lambda_m)} \cup \Delta B_n^{(\lambda_n,\lambda_m)} \quad (2.5)$$

Define $A_n^* = (A_n \cup \Delta B_n^{(\lambda_n+\delta,\lambda_n+\delta+\xi)}) \cap (\Delta A_n^{(\lambda_n+\delta,\lambda_n+\delta+\xi)})^c$. Then, there exists an $\xi > 0$ so that P_f^{**} with the decision region $\Omega(A_1,\cdots,A_m^*,\cdots,A_n^*,\cdots,A_N)$ is equal to P_f with $\Omega(A_1,\cdots,A_m,\cdots,A_n,\cdots,A_N)$, or $P_f^* - P_f = P_f^* - P_f^{**}$ so that

$$\int_{\Delta A_m^{(\lambda_m-\varepsilon,\lambda_m)}} D_{0m}(x_m)dx_m - \int_{\Delta B_m^{(\lambda_m-\varepsilon,\lambda_m)}} D_{0m}(x_m)dx_m = \int_{\Delta A_n^{(\lambda_n+\delta,\lambda_n+\delta+\xi)}} D_{0n}^*(x_n)dx_n - \int_{\Delta B_n^{(\lambda_n+\delta,\lambda_n+\delta+\xi)}} D_{0n}^*(x_n)dx_n. \quad (2.6)$$

Using ideas similar to the ones used to develop (2.4) [2] we obtain

$$P_d^* = P_d + \int_{\Delta A_m^{(\lambda_m-\varepsilon,\lambda_m)}} D_{1m}(x_m)dx_m - \int_{\Delta B_m^{(\lambda_m-\varepsilon,\lambda_m)}} D_{1m}(x_m)dx_m \quad (2.7)$$

and

$$P_d^{**} = P_d^* - \int_{\Delta A_n^{(\lambda_n+\delta,\lambda_n+\delta+\xi)}} D_{1n}^*(x_n)dx_n + \int_{\Delta B_n^{(\lambda_n+\delta,\lambda_n+\delta+\xi)}} D_{1n}^*(x_n)dx_n. \quad (2.8)$$

From the definition of ΔA_n and ΔB_n, we know that

$$\int_{\Delta A_m^{(\lambda_m-\varepsilon,\lambda_m)}} D_{1m}(x_m)dx_m \ge (\lambda_m - \varepsilon)\int_{\Delta A_m^{(\lambda_m-\varepsilon,\lambda_m)}} D_{0m}(x_m)dx_m \quad (2.9)$$

and

$$-\int_{\Delta B_m^{(\lambda_m-\varepsilon,\lambda_m)}} D_{1m}(x_m)dx_m \ge -(\lambda_m - \varepsilon)\int_{\Delta B_m^{(\lambda_m-\varepsilon,\lambda_m)}} D_{0m}(x_m)dx_m \quad (2.10)$$

Using these same ideas we can also show

$$\int_{\Delta A_n^{(\lambda_n+\delta,\lambda_n+\delta+\xi)}} D_{1n}^*(x_n)dx_n < (\lambda_n + \delta + \xi)\int_{\Delta A_n^{(\lambda_n+\delta,\lambda_n+\delta+\xi)}} D_{0n}^*(x_n)dx_n \quad (2.11)$$

and

$$-\int_{\Delta B_n^{(\lambda_n+\delta,\lambda_n+\delta+\xi)}} D_{1n}^*(x_n)dx_n$$
$$< -(\lambda_n+\delta+\xi)\int_{\Delta B_n^{(\lambda_n+\delta,\lambda_n+\delta+\xi)}} D_{0n}^*(x_n)dx_n \quad (2.12)$$

Combine this with (2.6) to obtain

$$\int_{\Delta A_m^{(\lambda_m-\varepsilon,\lambda_m)}} D_{1m}(x_m)dx_m$$
$$-\int_{\Delta B_m^{(\lambda_m-\varepsilon,\lambda_m)}} D_{1m}(x_m)dx_m$$
$$> \frac{\lambda_m-\varepsilon}{\lambda_n+\delta+\xi}\Big(\int_{\Delta A_n^{(\lambda_n+\delta,\lambda_n+\delta+\xi)}} D_{1n}^*(x_n)dx_n$$
$$-\int_{\Delta B_n^{(\lambda_n+\delta,\lambda_n+\delta+\xi)}} D_{1n}^*(x_n)dx_n\Big). \quad (2.13)$$

Since δ and ξ will get monotonically smaller when ε is made smaller, we can choose ε small enough so that $\lambda_n+\delta+\xi < \lambda_m-\varepsilon$, thus (2.13) becomes

$$\int_{\Delta A_m^{(\lambda_m-\varepsilon,\lambda_m)}} D_{1m}(x_m)dx_m$$
$$-\int_{\Delta B_m^{(\lambda_m-\varepsilon,\lambda_m)}} D_{1m}(x_m)dx_m$$
$$> \int_{\Delta A_n^{(\lambda_n+\delta,\lambda_n+\delta+\xi)}} D_{1n}^*(x_n)dx_n$$
$$-\int_{\Delta B_n^{(\lambda_n+\delta,\lambda_n+\delta+\xi)}} D_{1n}^*(x_n)dx_n \quad (2.14)$$

which can be rewritten as

$$P_d^* - P_d > -(P_d^{**} - P_d^*) \quad (2.15)$$

or

$$P_d^{**} > P_d \quad (2.16)$$

This means that the rules defined by $\Omega(A_1,\cdots,A_m^*,\cdots,A_n^*,\cdots,A_N)$ achieve a larger detection probability while maintaining the same level of false alarm. This contradicts the assumption that a scheme without $\lambda_1=\lambda_2=\cdots=\lambda_N$ can be optimum.

If λ_m is taken to be at $\pm\infty$ while $\lambda_n, n = 1,\ldots,N, n\neq m$ are taken to be finite, then a similar argument to that made above shows that performance can be improved by choosing a finite λ_m.

Now suppose that $f_{X_k}(x_k|H_j)=0$ for some x_k, $j=0,1$ and $k=1,\ldots,N$. We do not consider cases where only one of $f_{X_k}(x_k|H_0)$ and $f_{X_k}(x_k|H_1)=0$ and the other is not since this describes a singular detection problem. In this case if λ_k is in any interval where this is true then we can clearly move λ_k to any point in this interval without any change in performance. Of course, such changes are not really of any significance and if these changes are ignored, the above results still hold. Except, of course this possibility introduces cases without $\lambda_1=\lambda_2=\cdots=\lambda_N$ which can produce exactly the same performance, instead of $\lambda_1=\lambda_2=\cdots=\lambda_N$ being strictly better. Note that this possibility is incorporated in the wording of the Theorem. □

3 Discussion

Conditions for the optimum sensor detectors for NP have been studied in a few previous papers, but the derivations provided in these papers have been questioned by a number of respected authors [1, 3]. The questions they raised appear to be justified based on some of the derivations provided. Our derivations do not leave any questions. We clearly show our conditions which allow $\lambda_1\neq\lambda_2\neq\cdots\neq\lambda_N$ are necessary to solve NP. Then we show that constraining $\lambda_1=\lambda_2=\cdots=\lambda_N$ is necessary if the pdfs of the sensor observations have infinite support and in other cases it will not sacrifice optimality (we ignore singular detection cases).

In [1] the author demonstrates that attempting to solve NP in a distributed case by maximizing $P_d-\lambda P_f$ without constraints, which was the approach taken in some previous papers, is not generally correct. In particular, he demonstrates that this procedure will fail if the overall ROC is not concave. This is significant since no one yet has proven (even for cases with a fixed fusion rule and no point masses in the sensor test distributions) that the overall ROC must be concave. In fact, a counter example is given for the case of a fixed fusion rule in

[4] for a case with point masses in the sensor test distributions. In [5] we present a counter example for a case with a fixed fusion rule and no point masses in the sensor test distribution. This is the first example of this type we have seen. Clearly the ROC can be non-concave if the fusion rule is not fixed. For an example see [6]. Note that our derivation did not rely on the overall receiver operating curve (ROC) being concave since we don't attempt to maximize $P_d - \lambda P_f$.

Interestingly enough, even though we don't attempt to maximize $P_d - \lambda P_f$ the conditions we provide through Theorem 2.1 and Theorem 2.2 in this paper are similar in form to those produced in some previous papers that attempted to produce necessary conditions for maximizing $P_d - \lambda P_f$. Although we found this very confusing initially, there is a simple explanation. Basically, it can be viewed as a coincidence that the necessary conditions for maximizing $P_d - \lambda P_f$ just happen to look like our correct conditions. To see this, let us illustrate a similar circumstance for a different problem.

Consider an optimization problem where one is attempting to find a vector $\mathbf{t}$ that maximizes a quantity P_d under the constraint $P_f = \alpha$. For this example assume P_d and P_f are continuous functions of each component of $\mathbf{t}$. Necessary conditions for this case are well known. For example, restating a Theorem from page 224 of [7] we obtain the following Theorem.

Theorem 3.1 *Let P_d be a real-valued function of $\mathbf{t} = (t_1, \ldots, t_N)$ where $-\infty \leq t_i \leq \infty$ for $i = 1, \ldots, N$. Let P_f be another real-valued function of $\mathbf{t}$. Let $\mathbf{t}_o$ be a local extremum of P_d under the constraint $P_f = \alpha$ and assume $\nabla P_f \neq 0$ at $\mathbf{t}_o$ where ∇P_f denotes the gradient of P_f with respect to t. Then there exists a real-valued λ such that at $\mathbf{t}_o$ we have $\nabla P_d - \lambda \nabla P_f = 0$.*

The conditions provided in Theorem 3.1, $\nabla P_d - \lambda \nabla P_f = 0$, are called first-order necessary conditions and λ in Theorem 3.1 is generally called a Lagrange multiplier. $\nabla P_d - \lambda \nabla P_f = 0$ says that the normal vectors to the tangent planes of the P_d and P_f surfaces must point in the same direction at the extremum (see the proof in [7]). Theorem 3.1 **does not** generally imply $\mathbf{t}_o$ will be at an unconstrained extremum of $P_d - \lambda P_f$. In fact, a few counter examples are presented in [8] which show this is not generally the case. However, the conditions $\nabla P_d - \lambda \nabla P_f = 0$ would also be obtained as necessary conditions to find extrema of $P_d - \lambda P_f$ without constraints for the correct λ, the Lagrange multiplier. So in fact, the correct form of the necessary conditions are obtained using a possibly inappropriate procedure (attempting to find extrema of $P_d - \lambda P_f$ without constraints instead of using Theorem 3.1). Further, in the case where there is only one solution to $\nabla P_d - \lambda \nabla P_f = 0$ then this will be the solution to the problem posed in Theorem 3.1 and it must also be an unconstrained extrema of $P_d - \lambda P_f$. This situation will occur with the appropriate type of convexity condition as described for example in [7].

Our situation is similar, in one important way. Our correct conditions are produced by combining Theorem 2.1 and Theorem 2.2, not by attempting to maximize $P_d - \lambda P_f$. However these conditions are identical to those which have been obtained as necessary conditions for maximizing $P_d - \lambda P_f$. In summary attempting to maximize $P_d - \lambda P_f$ is generally not the correct way to solve NP, but the necessary conditions to maximize $P_d - \lambda P_f$ just happen, by coincidence, to take the same form as a set of conditions which will lead to an optimum solution.

For completeness we note some other interesting, but less important comparisons between our problem and the one considered in Theorem 3.1. First, of course, our problem is different. For example, instead of a vector optimization, we have a functional optimization where γ, a set of functions, replaces the vector $\mathbf{t}$ [1]. We produce our necessary conditions, from Theorem 2.1 as the best form of a given sensor detector when the other sensor detec-

[1] For an approach to defining a gradient which allows a generalization to Theorem 3.1 to functional optimization problems see for example [9].

tors and the fusion rule are fixed. We can view these person-by-person optimum necessary conditions as taking the place of the "first-order" necessary conditions in Theorem 3.1. In our case performance can't be improved by changing only one sensor and in the case of Theorem 3.1 performance can't be improved by moving $\mathbf{t}$ by an incrementally small distance in any direction. We emphasize again, that our conditions, like those in Theorem 3.1, don't attempt to maximize $P_d - \lambda P_f$. However, the conditions produced by combining Theorem 2.1 and Theorem 2.2 are identical to those which have been obtained as necessary conditions for maximizing $P_d - \lambda P_f$.

In the previous papers, take [10, 11] for example, that produced conditions that look similar to what we obtain from Theorem 2.2 (they use $\lambda_1 = \lambda_2 = \cdots = \lambda_N$), it was frequently implied that these conditions are always necessary for NP. In fact our derivation shows this is not always true and further we give a counter example later in this paper.

The discussions in [1, 3] led to a number of publications which stated that they produced counter examples in which conditions similar to those produced by the combination of Theorem 2.1 and Theorem 2.2 do not work. Since these conditions were originally produced using an incorrect methodology, we understand the motivation of these authors. However, in checking these counter examples, we found the results in Theorem 2.1 and Theorem 2.2 did work. Next we describe our findings in more detail.

First, consider a three-sensor scheme for detecting a Rayleigh fading signal in Gaussian noise for the case where the observations are independent from sensor to sensor [3]. In this case, likelihood ratio tests are optimum at the sensors. Let ϵ_i be the signal-to-noise ratio at sensor i, P_{Fi}, P_{Di} be the false alarm probability and detection probability at sensor i, and t_i be the threshold at sensor i that the likelihood ratio should be compared to. From [12], for this problem

$$P_{Fi} = (t_i(1+\epsilon_i))^{-1-(1/\epsilon_i)} \quad \text{if } t_i > \frac{1}{1+\epsilon_i} \tag{3.17}$$

Note that for any t_i such that $t_i < 1/(1+\epsilon_i)$ then $P_{Fi} = 1$. It appears that [3] assumed that the left-hand side of (3.17) was valid for all t_i. The discussion in [3] implies that for this problem Theorem 2.1 and Theorem 2.2 do not work. When we correctly apply (3.17) we find these conditions work. We explain this in more detail next.

For this case the conditions from Theorem 2.1 and Theorem 2.2 reduce to the following comparisons

$$\begin{aligned} t_1 &= \lambda \frac{P_{F2}(1-P_{F3})}{P_{D2}(1-P_{D3})} \\ t_2 &= \lambda \frac{1-(1-P_{F1})(1-P_{F3})}{1-(1-P_{D1})(1-P_{D3})} \\ t_3 &= \lambda \frac{(1-P_{F1})P_{F2}}{(1-P_{D1})P_{D2}} \end{aligned} \tag{3.18}$$

In [3], the authors consider the case of $\epsilon_1 = \epsilon_2 = \epsilon_3 = \epsilon$. They show a direct optimumization for this case implies that $P_{F2} = P_{D2} = 1$. Then the authors divide the equation in (3.18) for t_2 by the equation for t_3 from (3.18) and claim that after using (3.17) with algebra they obtain an equation only in terms of P_{F1}. However, they don't recognize that $P_{F2} = P_{D2} = 1$ is satisfied by any $t_2 \leq 1/(1+\epsilon)$ so that the value of t_2 can't be determined by (3.17). There is a whole range of possible t_2 that satisfy (3.17). Thus an expression of the type they say they obtain, can't be valid. In fact t_2 takes on the value necessary to satisfy the second equation in (3.18) and it can be seen easily that this will yield $t_2 < 1/(1+\epsilon)$ so the conditions in (3.18) do give the same solution as the direct optimumization does.

For example, let $\epsilon = 1$ and use a fact found from the direct optimumization, that is $P_{f1} = P_{f3}$. Define $\beta^2 = P_{f1} = P_{f3}$. Then from (3.17) we find $\beta^2 = (2t_1)^{-2}$ and so the first equation of (3.18) yields

$$\lambda = \frac{1}{2\beta(1+\beta)} \tag{3.19}$$

and this with the other equations in (3.18) yield

$$\begin{aligned} t_1 &= \frac{1}{2\beta} \\ t_2 &= \frac{2-\beta^2}{2(2-\beta^2+\beta)} \\ t_3 &= \frac{1}{2\beta} \end{aligned} \tag{3.20}$$

and from the fusion rule we find $\beta = (1 - \sqrt{1-\alpha})^{\frac{1}{2}}$, and α is the false alarm rate. From (3.20), we see that $t_2 \leq \frac{1}{2}$, and thus $P_{F2} = P_{D2} = 1$, which agrees with the results given in the direct optimumization.

A similar two-sensor example for the same Rayleigh fading signal in Gaussian noise problem has been given in [13] where the same mistake appears to have been made. Here, for the AND rule case with $\epsilon_1 \neq \epsilon_2$, there is said to be no solution to the conditions produced by Theorem 2.1 and Theorem 2.2. However, when (3.17) is applied properly a solution does exist. For example, for the AND rule case with $\epsilon_1 = 1, \epsilon_2 = 2$ and $P_F = 10^{-6}$, the optimum solution should be $t_1 \leq \frac{1}{2}$ and $t_2 = \frac{1}{3} \times 10^4$, which is satisfied by the formulation given by Theorem 2.1 and Theorem 2.2 with $\lambda = \frac{1}{3} \times 10^4$, $t_1 = \frac{1}{3}$ and $t_2 = \frac{1}{3} \times 10^4$. It should be noted that $\lambda = \frac{1}{3} \times 10^4$, $t_1 = \frac{1}{3}$ and $t_2 = \frac{1}{3} \times 10^4$ are not the only values which can solve NP. In fact, sensor one should use $P_{f1} = 1$ and as we have already stated any t_1 such that $t_1 < 1/(1+\epsilon_1) = 1/2$ will give $P_{f1} = 1$ so there are many other optimum schemes besides the one produced by Theorem 2.1 and Theorem 2.2. Of course this possibility is discussed in Theorem 2.2 as a case without infinite support. This shows that the conditions in Theorem 2.1 and Theorem 2.2 are not always necessary conditions, as some have stated. This shows that, exactly as we state in Theorem 2.1 and Theorem 2.2, we can obtain an optimum solution using Theorem 2.1 and Theorem 2.2, but in cases with $f_{X_k}(x_k|H_j) = 0, j = 0, 1$, there may be other schemes which are also optimum.

A second two sensor example is also considered in [13]. In this case a known signal (m_1, m_2) is to be detected in additive zero-mean unit-variance Gaussian noise by a two-sensor parallel distributed detection system. For the case of $m_2 = 0.5$ when either the AND or the OR rule is used, the authors state that an optimum solution to the conditions from Theorem 2.1 and Theorem 2.2 can only be found for a finite range of m_1. We found that the conditions in Theorem 2.1 and Theorem 2.2 did produce an optimum solution outside the ranges given for every case we tried. We note that in each case we tried we found the t_1 and t_2 produced by Theorem 2.1 and Theorem 2.2 were always finite values and that the optimum thresholds found by a direct optimumization also always took on the same finite values.

Numerical Results

By using Theorem 2.1,Theorem 2.2 we can employ a Gauss-Seidel type of iterative algorithm to solve a wide range of optimum distributed detection problems under the Neyman-Pearson criterion. Our approach employs the technique used in fixed fusion rule Bayesian optimization of distributed detection schemes [14] with a slight twist. The twist [5] is to find the best λ and then to apply the Gauss-Seidel procedures given in [14].

As an example, let's consider a two-sensor problem with a binary hypotheses

$$H_0 : x_1, x_2 \sim N(0, 0, 1, 1, \rho)$$
$$H_1 : x_1, x_2 \sim N(s_1, s_2, 1, 1, \rho)$$

where $N(a, b, c, d, e)$ denotes a bivariate Gaussian distribution with $E[(x_1, x_2)^T] = (a, b)^T$, $Var(x_1) = c$, $Var(x_2) = d$ and $E[x_1 x_2] = e\sqrt{cd}$. Assume the fixed fusion rule is the AND rule and consider the case of $s_1 = 1, s_2 = 2$ and $\rho = 0.2$. In this case we find the overall receiver operating characteristic is concave and the false alarm probability is monotonic with respect to the value of λ. However, we show in [5] that the curve of P_f versus λ will not always be monotonic. Further for a given value of λ, several different converged solutions may result from the Gauss-Seidel procedure. Illustrative examples and extensions to cases with multiple bit sensor decisions and other topologies are given in [5].

4 Conclusion

This paper has focused on optimum Neyman-Pearson distributed signal detection. We have presented two key Theorems which we believe clarify the conditions for NP optimum sensor detectors under a fixed fusion rule. The Theorems appear to be similar to some previous results that were obtained using an inappropriate procedure. Our Theorems, however, state requirements under which our conditions are necessary. Such conditions have been lacking in previous research and we demonstrate that these conditions are needed. Our focus here was on cases with binary sensor decisions and for a parallel architecture, but we have already extended our results in both of these regards.

References

[1] J. N. Tsitsiklis, "Decentralized Detection," *Advances in Statistical Signal Processing, Vol. 2: Signal Detection,* H.V. Poor and J.B. Thomas, Ed. Greenwich, CT: JAI Press, 1990.

[2] R. S. Blum, "Necessary conditions for optimum distributed sensor detectors under the Neyman-Pearson criterion," *IEEE Transactions on Information Theory*, IT-42, pp. 990-994, May 1996.

[3] S. C. Thomopoulos, R. Viswanathan and D. K. Bougoulias, "Optimal Decision Fusion in Multiple Sensor System", *IEEE Transaction on Aerospace and Electronic Systems*, vol. AES-23, pp. 644-653, 1987.

[4] M. Cherikh, "Optimal decision and detection in the decentralized case," Ph.D. dissertation, Department of Operations Research, Case Western Reserve University, May 1989, pp. 117-120.

[5] Qing Yan and R. S. Blum, "Distributed Signal Detection under the Neyman-Pearson Criteion," submitted to *IEEE Transactions on Information Theory.*

[6] P. K. Willett and D. J. Warren, "The suboptimality of randomized tests in distributed and quantized detection Systems," *IEEE Transaction on Information Theory* vol. 38, No. 2, pp. 355-362, March 1992.

[7] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*, Reading, Massachusetts: Addison-Wesley, 1973.

[8] M. R. Hestenes, *Optimization Theory: The Finite Dimensional Case*, New York: John-Wiley & Sons, New York, 1975.

[9] N. Dorney, *Optimization by Vector Space Methods*, Reading, Massachusetts: Addison-Wesley, 1973.

[10] R. Srinivasan, "Distributed radar detection theory," *IEE Proceedings*, vol. 133, Part F, No. 1, pp. 55-60, Feb. 1986.

[11] R. Srinivasan, "A theory of distributed detection," *Signal Processing*, vol. 11, No. 1, pp. 319-327, Dec. 1986.

[12] A. D. Whalen *Detection of Signals in Noise* , Orlando, Florida: Academic Press. Inc., 1971.

[13] P. K. Varshney, *Distributed Detection and Data Fusion*, New York: Springer-Verlag, 1997.

[14] P.F.Swaszek, P.Willett and R.S.Blum, "Distributed detection of dependent data - the two sensor problem", *29th Annual Conference on Information Sciences and Systems*, Princeton University, pp.1077-1082, Princeton, N.J., March 1996.

An Algorithm to Enhance Coordinate Registration by Fusing Over-the-Horizon Radar Sensors

William C. Torrez
SPAWAR SYSCEN CODE D7212
53560 HULL STREET
SAN DIEGO, CA 92152

William J. Yssel
SPAWAR SYSCEN CODE D7212
53560 HULL STREET
SAN DIEGO, CA 92152

Abstract *Equivalent data from independent radar sensor systems can increase track accuracy by providing diverse "looks" at a common target area. With proper geometry, complementary sensor systems can aid in resolving uncertainties in the coordinate registration process associated with the various ionospheric modes. Systematic positional differences between tracks seen from the separate radar sites can be used to improve the estimation of ionospheric parameters. In operational systems, targets are tracked by multiple over-the-horizon (OTH) radar systems in overlapping coverage areas. In this paper, we consider the case of two overlapping OTH radars. If an ensemble of targets were in the coverage area, an error in the ionospheric model parameters would manifest itself similarly in each of the tracks, and a correction to this error would improve the accuracy of all target positions in the region. This approach presumes correct or at least consistent mode assignments.*

Keywords: over the horizon radar, generalized least squares estimation

1 Introduction

In a previous paper [1], a data set was used which consisted of two independent OTH radar systems covering a common surveillance region at a range about 1500 nmi from both OTH sites. A ground-based microwave radar provided truth data in the region. During the two-hour data period, eleven ground targets were concurrently held by both OTH radar systems and by the ground-based microwave radar. The OTH radar systems, running in their standard manner, detected the targets, formed tracks in radar coordinates, identified tracks belonging to the same target, selected and assigned ionospheric modes to be used, brought each of the radar tracks to ground coordinates using the appropriate coordinate registration tables, and fused the collection into common target states. This was done for each minute in which the OTH radar held contact on the target. Using the microwave radar to provide the true target position, the range and cross-range errors for each of the targets were calculated. The range errors and cross-range errors were plotted as a function of time and it was shown that a significant range bias was present and persisted over time. In this paper, an algorithm is developed to calculate and hand-off a range correction to OTH radar 1 based on the observation of the cross-range positions observed by OTH radar 2. Likewise, OTH radar 1 will provide its cross-range bearings on common targets by mode for correction of any range bias experienced by OTH radar 2. It is shown that if this is done for the ensemble of targets in the surveillance region and these corrections are used to update the coordinate registration tables, positions of targets in the area being detected by only one of the OTH radars will also be improved.

2 Analysis Approach

We consider the problem of estimating the position Z of a target from multiple measure-

ments provided by a system of two spatially distributed OTH radar sensors. At the central tracking processor, the track plots from the multiple radars are used to update existing system tracks or initiate new system tracks as appropriate. Specifically, the central tracking processor must perform the following five functions:

1. Coordinate Registration: Transformation of the radar plots from local radar (or slant) coordinates to system coordinates, which are latitude and longitude (or ground coordinates).

2. Correlation or association of the radar plots with the appropriate system tracks.

3. Initiation of new tracks with the uncorrelated plots and rejection of clutter plots.

4. Track filtering and track prediction.

5. Track monitoring and system track management.

Functions 2 and 4 represent the heart of the traditional data association and tracking problem. However, before either of these processes can occur successfully, function 1 must be performed; that is, the individual radar data must be expressed in a common coordinate system in which the errors due to site uncertainties, antenna orientation, and improper calibration of range and time (usually due to ionospheric uncertainties) have been minimized so they do not cause a significant degradation of the system operation. The process of ensuring the requisite "error free" coordinate conversion of radar data is called *coordinate registration* (CR). Thus, CR is an absolute prerequisite for multiple radar tracking or sensor fusion in general.

The type of measurements provided by the systems consists of radar slant coordinates (bearing, θ, and range, r, from a radar sensor to the target). Following Wax[2], and using the notation in Dana[5], we formulate the difference ΔP in the reported positions as a function of the set of measured variables Z (i.e., observations) and the set of bearing and range biases β (i.e., parameters) to be estimated:

$$\Delta P = F(Z, \beta)$$

Following the usual linearization technique, but with the roles of the actual values and estimators reversed, the vector equation or position difference can be transformed in the classical Gauss-Markov *generalized linear least-squares estimation* (GLSE) model:

$$X\beta + \xi = Y$$

where X is a matrix of known parameters, ξ is the vector of measurement errors, and Y is the measurement vector.

The solution of the GLSE problem is simply

$$\beta^* = \Sigma^* X^T {\Sigma_\xi}^{-1} Y$$

where

$$\Sigma^* = \left(X^T {\Sigma_\xi}^{-1} X\right)^{-1}$$

is the covariance matrix for the estimate β^* of the vector of biases β.

The GLSE approach was developed for two range and two azimuth offset biases. To assess quantitatively the performance of the GLSE approach, the algorithm will be evaluated in detail considering both simulation and real OTHR data analyses.

For this application, we consider the case of two overlapping OTH radars R_A located at the origin, and R_B, located at coordinates (u, v). We further assume that R_A gives biased measurements of range, while R_B gives biased measurements of target azimuth. Denote the vector of radar measurements by

$$\Psi_k = (r_{Ak}, \theta_{Bk})^T$$

where r_{Ak} and θ_{Bk} denote the range and azimuth measurements from radar R_A and R_B, respectively, and k denotes the time index.

The generalized measurement equations from the two sensors is, as mentioned above, $\Delta P = F(Z, \beta)$, based on the measurements Ψ_k and the set of biases $\beta = (\Delta r_A, \Delta\theta_B)^T$. For this application, these measurements are

$$g_A(x(k), z_1(k), \Delta r_A)$$

$$= r_A(k) - \sqrt{{x_1}^2(k) + {x_2}^2(k)} - \Delta r_A$$

and

$$g_B(x(k), z_2(k), \Delta\theta_B)$$

$$= \theta_B(k) - tan^{-1}(\frac{x_2(k) - v}{x_1(k) - u}) - \Delta\theta_B$$

Here $z_1 k$ are range measurements from R_A at time k, $z_2(k)$ are azimuth measurements from sensor R_B at time k, Δr_A and $\Delta\theta_B$ are the bias parameters to be estimated and $x_1(k)$ and $x_2(k)$ are target state vectors at time k. These equations relate the set β of bias parameters to be estimated from the set of measurements Ψ_k and the vector of observations z. However this relationship is nonlinear.

To apply the theory of generalized least squares, we will need to represent the observations as a linear function of the parameters to be estimated, namely β. This can be accomplished by defining a function f as follows:

$$f(\Psi_k, \beta)$$

$$= [g_A(x(k), z_1(k), \Delta r_A), g_B(x(k), z_2(k), \Delta\theta_B)]$$

Further, let ${\Psi_k}'$ and β' denote the actual measurement sets and an initial estimate of β, respectively. Now Taylor's Theorem can be used to in the usual way to approximate the function f at the true values of Ψ_k and β in terms of the measurements ${\Psi_k}'$ and the initial estimate β'.

3 Conclusions

This paper has presented a comprehensive and generalized method for estimating the bias parameters arising in OTH radar surveillance. Before system implementation, it is planned to utilize the analysis approach on a wide variety of target scenarios. Recently, researchers (cf. [3,4]) have developed software packages for the bias estimation procedures. The former approach [3] is interesting, because it does not impose any distributional constraints on the system measurements, although an assumption of Gaussianity on the measurements allows for optimal (i.e., maximum likelihood) statistical estimates (cf [4]).

4 Acknowledgements

The authors would like to thank the Relocatable OTHR Program Office, US Naval Space Command, for their support of this work. They would also like to thank Mr. Robert Musil, ROTHR Project Leader at Space and Naval Warfare Systems Center, San Diego, for his support and encouragement.

References

[1] William C. Torrez and William J. Yssel. Over-the-horizon radar surveillance sensor fusion for enhanced coordinate registration. *Proceedings of the 1999 Information, Decision, and Control*, 227–230, Feb 1999, Adelaide, Australia.

[2] Mati Wax. Position location from sensors with position uncertainty. *IEEE Trans. Aero. Elect. Sys.* AES-19: 658-662, Sep 1983.

[3] Egil Sviestins. On-line bias estimation for multisensor tracking. *Proceedings of the 1999 Information, Decision, and Control*, 221–226, Feb 1999, Adelaide, Australia.

[4] Daniel W. McMichael and Nickens N. Okello. Maximum likelihood registration of dissimilar sensors. *Proc. 1st Australian Data Fusion Symp.* 31–34, Nov 1996, Adelaide, Australia.

[5] Martin P. Dana. Registration: A prerequisite for muliple sensor tracking. In *Multitarget-Multisensor Tracking: Advanced Applications.* Ed. by Y. Bar-Shalom. Artech House, 1990.

Field Evaluations of Dual-Band Fusion for Color Night Vision

M. Aguilar, D. A. Fay, D.B. Ireland, J.P. Racamato, W. D. Ross, and A.M. Waxman
M.I.T. Lincoln Laboratory
Sensor Exploitation Group
244 Wood St.
Lexington, MA 02420

Abstract - As part of an advanced night vision program sponsored by DARPA, a method for real-time color night vision based on the fusion of visible and infrared sensors has been developed and demonstrated. The work, based on principles of color vision in humans and primates, achieves an effective strategy for combining the complementary information present in the two sensors. Our sensor platform consists of a 640x480 low-light CCD camera developed at MIT Lincoln Laboratory and a 320x240 uncooled microbolometer thermal infrared camera from Lockheed Martin Infrared. Image capture, data processing, and display are implemented in real-time (30 fps) on commercial hardware. Recent results from field tests at Lincoln Laboratory and in collaboration with U.S. Army Special Forces at Fort Campbell will be presented. During the tests, we evaluated the performance of the system for ground surveillance and as a driving aid. Here, we report on the results using both a wide-field of view (42 deg.) and a narrow field of view (7 deg.) platforms.

Keywords: sensor fusion, image fusion, infrared, night vision.

1. Introduction

In night vision applications, both low-light visible and thermal infrared are effective but not sufficiently capable when used independently. These two sensor modalities image different physical properties of a scene, and the complementarity of the information they provide is well known. The work presented here addresses the real-time fusion of visible and IR (MWIR or LWIR) imagery into a single color composite either for presentation, or for further processing, with the aim of efficiently exploiting the complementarity of the multi-sensor information.

Dual-sensor fusion has been achieved using an architecture based on principles of processing in primate retinal circuits. The architecture achieves contrast enhancement, adaptive dynamic range compression, and single-opponent color contrast processing, from which a color image is derived. Here, we report on progress in this approach to image fusion and on field tests conducted in collaboration with the Special Forces group at Ft. Campbell, KY.

Prior to our introduction of opponent-color fusion strategies [1, 2, 3], other methods for image fusion were rooted in pixel-level choice or blending of modalities, aimed at maximizing contrast and implemented on multiscale image representations [4, 5, 6, 7, 8]. The results are grayscale-fused images, which don't support target detection as accurately as our color fused images do in human factors tests [9, 10]. While there have been other approaches to obtaining color fused results [11, 12], they often produce a degradation in performance and/or image quality [9, 10]. Thus, in assessing the utility of fused imagery for select tasks such as target detection and localization, we found that one must be careful not to lose sight of the importance of image quality (i.e. resolution), for it certainly plays a role in object recognition tasks.

We have made important progress in the design of dual modality fusion architectures and of the associated hardware platforms for real-time processing. We introduce the theoretical background for our architectures in the next section. In addition, we have assembled a dual-sensor platform for color fusion based on a higher resolution, night-capable CCD camera developed at Lincoln Laboratory and a LWIR camera. Given the new computational needs for preprocessing, image combination, and color generation at video rates (currently 30 fps at 640x480 resolution) and the desire for future expandability, we adopted a new multi-node C80 architecture that we describe in a latter section. In particular, we report on the development and field-testing of a real-time platform based on COTS hardware.

2. Biologically-based sensor fusion

Our computational approach to image fusion derives its basis from biological models of color vision. In particular, in the retinal circuitry which has three types of retinal cones (*i.e.,* detectors) each has sensitivities to short, medium, and long wavelengths of the visible spectrum. The resulting images coded by each cone type are contrast enhanced *within band* by spatial opponent processing creating both ON and OFF *center-surround* channels [14]. These signals are also color-contrast enhanced via center-surround interactions *between bands* [15]. A significant insight that one obtains from these neurological findings is that nonlinear center-surround receptive fields come in

many varieties, are used to process imagery within and between bands, are the substrate for opponent processes in vision, and in general play an enormous role in the hierarchical design of biological image processors. In particular, they provide examples of working multi-band fusion mechanisms.

Our fusion strategy helps to complement the information provided by brightness contrast in the grayscale domain by utilizing color contrast to enhance information content in the displayed image. The combination of both forms of contrast has been shown to greatly enhance human perception [16, 17]. The neural architecture utilized in our real-time platform to fuse visible/LWIR imagery is constructed from center-surround opponent processing fields, specifically, shunting neural networks [18], as illustrated in Figure 3.

Following noise-cleaning of the visible and IR imagery and distortion correction to ensure image registration, a first stage of center-surround interactions within-band leads to contrast enhancement and dynamic range compression. Then, in a second stage of center surround processing across bands, we form two *grayscale fused* single-opponent color-contrast images with the enhanced Visible (*+Vis*) feeding the excitatory centers and the enhanced IR (ON-IR, *+IR*, and OFF-IR, *-IR*, respectively) feeding the inhibitory surrounds. We label these two single-opponent images *+Vis–IR* and *+Vis+IR*. In this context, our opponent-color contrast images can be interpreted as *coordinate rotations* in the color space of *Visible vs. IR*, along with *local adaptive scaling* of the new color axes, which leads to a decorrelation of the information in the two bands.

To achieve a useful color presentation of these opponent images (each being an 8-bit grayscale image), we assign the following color channels (8-bits each) to our digital imagery: (1) *+Vis* to *Green*, (2) *+Vis–IR* to *Blue*, and (3) *+Vis+IR* to *Red*. These channels are consistent with our natural associations of *warm red* and *cool blue*. The result is an image that uses brightness contrast to present information from the visible bands while utilizing color contrast to represent the thermal vs. visible information in the scene.

Finally, as shown in the architecture of Figure 3, these three channels can also be interpreted as *R, G, B* inputs to a color remapping stage in which, following conversion to *H, S, V* (hue, sat., val.) color space, hues can be remapped to alternative "more natural" hues. The result is a high quality *fused color* presentation of visible/IR imagery.

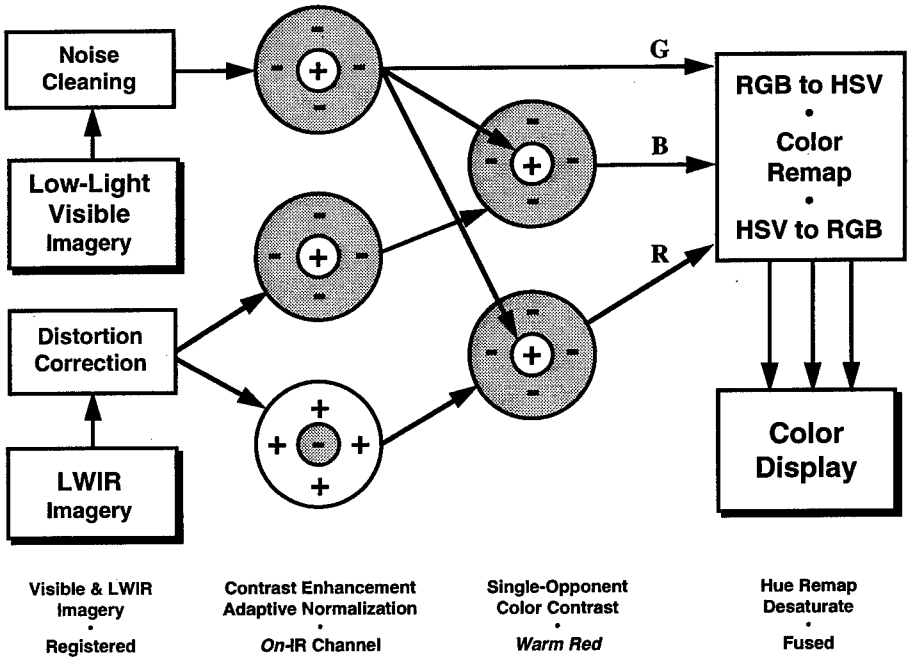


Figure 1. Single-opponent visible/LWIR image fusion architecture built from adaptive center-surround receptive fields. This architecture is well suited to sensors of non-equal resolution, with the higher resolution visible imagery providing input to the centers of the color contrast fields.

3. Sensor platform

Two different pods were assembled to provide for both a wide and a narrow field-of-view capabilities. Figure 2 illustrates the dual-sensor visible/LWIR imaging pods constructed at Lincoln Laboratory for the DARPA Integrated Imaging Sensors program. The imaging pods consists of a Lincoln Lab low-light CCD imager of 640x480 pixel resolution, able to provide useful 12-bit imagery at 30 frames/sec (or slower) below starlight illumination levels [13], an uncooled microbolometer LWIR thermal imager of 320x240 resolution and 15-bit dynamic range from Lockheed Martin Infrared, and a dichroic beam splitter that transmits the visible-NIR band but reflects the LWIR band.

The lenses utilized on both pods, in conjunction with the beam splitter, provide a nearly registered 42° and 7° field of view respectively. Deviations from registration (magnification and distortion) are compensated for in the real-time fusion processor.

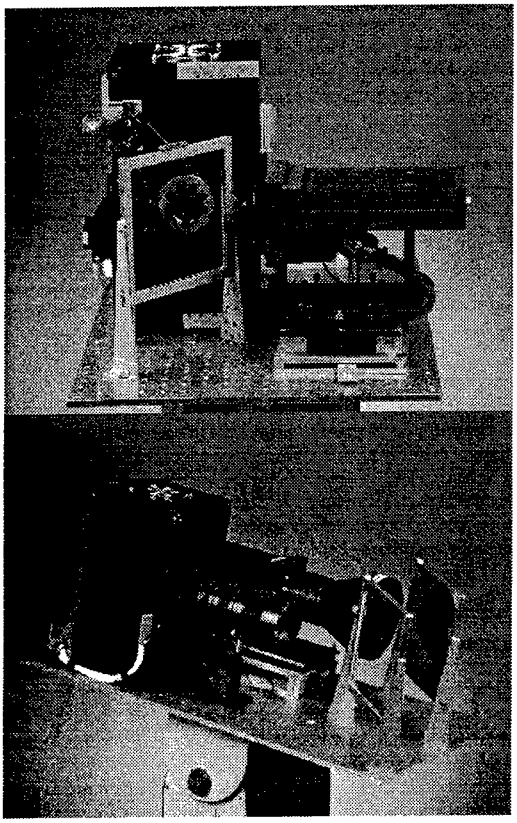


Figure 2. Dual-sensor fusion pods with Lincoln 640x480 pixel low-light CCD, a Lockheed Martin IR 320x240 pixel uncooled LWIR camera, and a dichroic beam splitter. Top: Wide field-of-view optics (42°). Bottom: Narrow field-of-view (7°).

4. Real-time COTS hardware

We developed a real-time visible/IR color fusion processor to support the wide dynamic range digital imagery provided by both cameras. We utilize a set of four Matrox Genesis C80 boards, providing for dual-digital video input and six C80 processing nodes, in an industrial PC rack-mount chassis, with a Pentium host processor card (see Figure 3).

The total number of operations per second is around 1.5 billion (640x480 pixels with 150 operations per pixel at 30 fps). Due to these requirements, we selected the C80 DSP as the core processing unit. This processor consists of 4 parallel integer processing units and a fifth floating-point processing unit. The Matrox board was selected because they offer a modular architecture revolving around two useful types of "Genesis" boards: (1) a main board, consisting of one C80 processor, 8Mb SDRAM, an analog/digital grab daughter board, and a video/display section; and (2) a co-processor board (see bottom of Figure 3), with two processing nodes, each with a C80 processor, 8Mb of SDRAM, and independent communication and control hardware. The main boards allow simultaneous capturing of the imagery from the two cameras. Two C80 nodes are allocated for preprocessing the imagery from each of

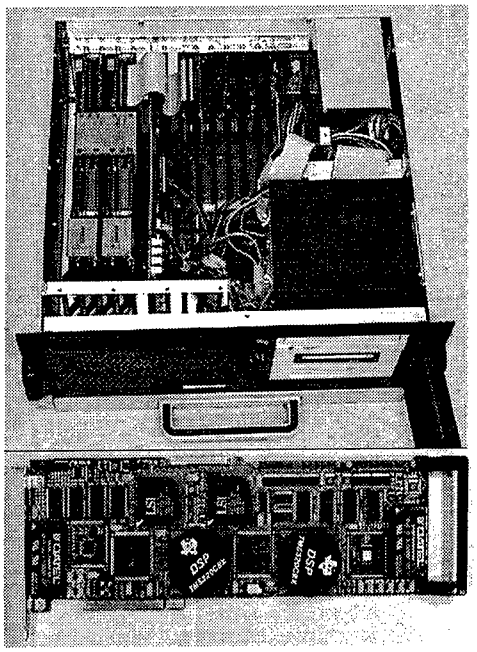


Figure 3. Real-time computing platform. Top: Computer chassis with four Matrox Genesis boards interconnected via a VM-channel bus and two proprietary grab-port buses. Bottom: Close-up of one of the dual C80 boards (co-processor board).

the sensors. Preprocessing consists of image warping for registration, noise cleaning, contrast enhancement, and dynamic range compression. The remaining two nodes are then used to fuse the preprocessed imagery and to drive the color display.

5. Field demonstrations

A first set of demonstrations took place in an open field and while driving without headlights in Bedford, MA. There, the real-time fusion system was evaluated under all phases of the moon during the fall of 1998. Figure 4 presents an example image taken under full-moon conditions. Figure 4a being the preprocessed visible image, 4b, the pre-processed IR image, and 4c, the color fused image. All were imaged with the wide field of view pod with the subject standing at 100 m. from the sensors. The results confirmed the preservation of both image quality and information content as obtained from both sensor bands.

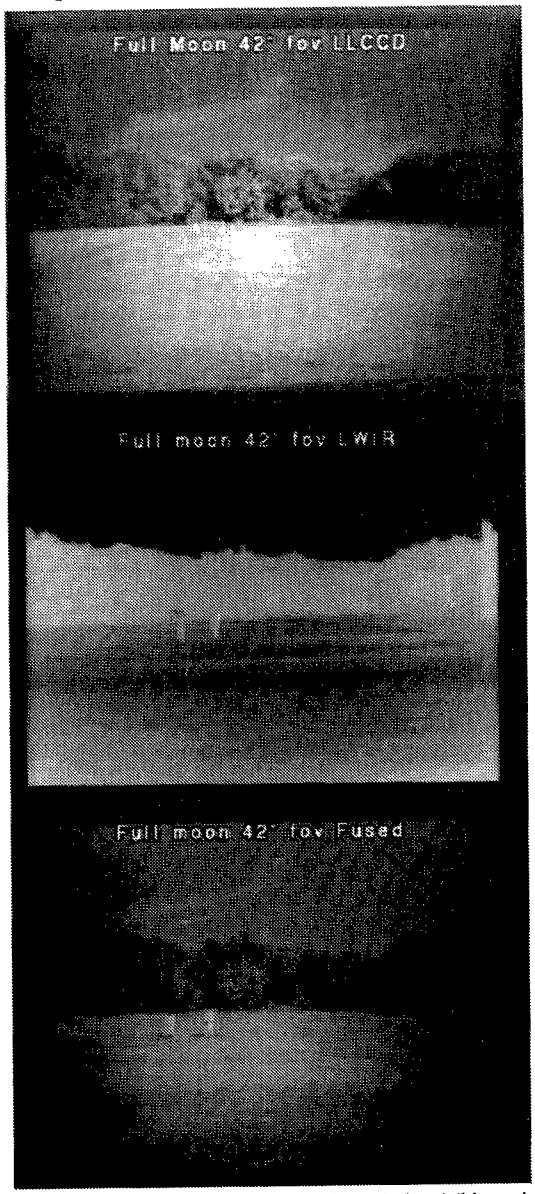


Figure 4. Color night vision by fusion of low-light visible and uncooled thermal IR imagery. (a-c) Lincoln Lab imagery (dusk conditions) using image intensified CCD and LWIR sensor pod.

The second set of field demonstrations with our real-time fusion platform was targeted at evaluating the system under more realistic conditions for military night-time operations. These tests took place during October 1998 at Ft. Campbell, KY in collaboration

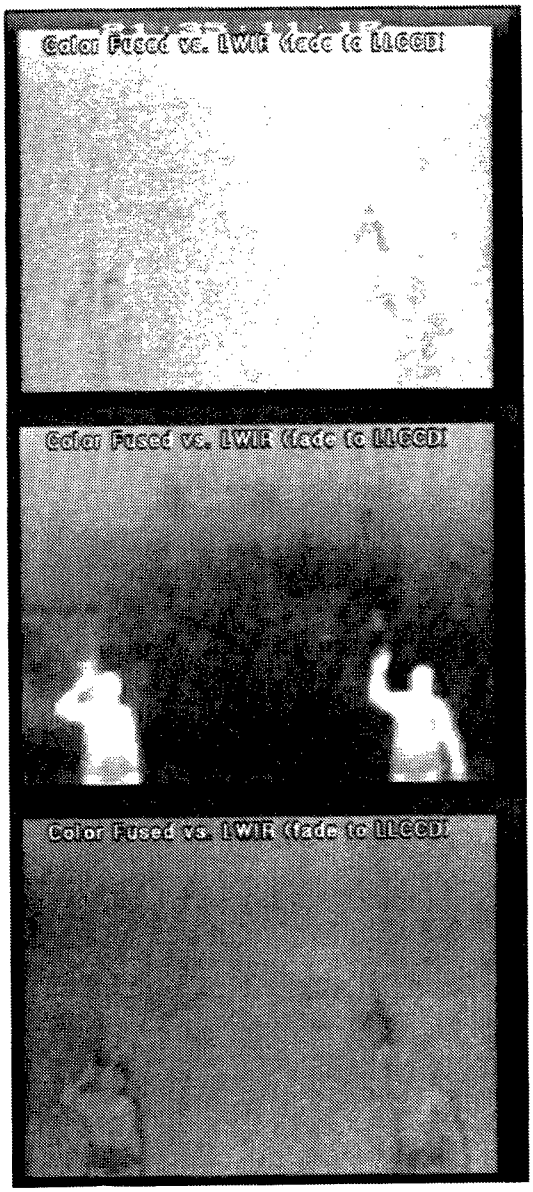


Figure 5. Color night vision by fusion of low-light visible and uncooled thermal IR imagery. (a-c) Lincoln Lab imagery (dusk conditions) using image intensified CCD and LWIR sensor pod.

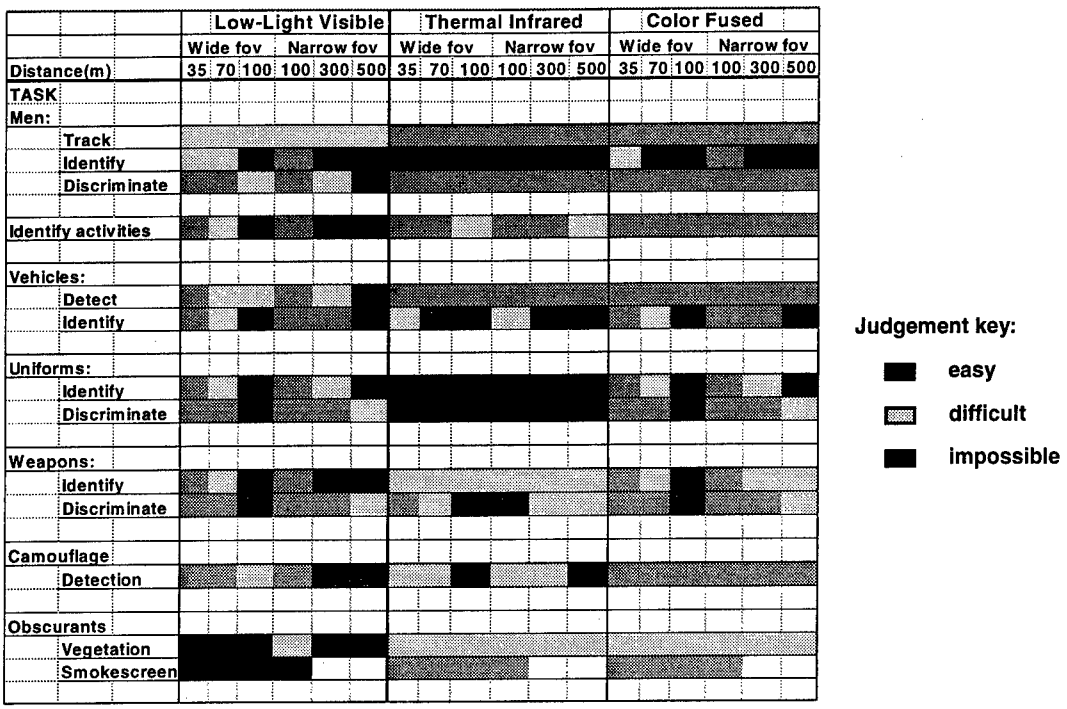


Figure 6. Qualitative assessment matrix. Observers judged their ability to perform the different tasks while using each of the modalities (visible, thermal ir, and color fused).

with US Army Special Forces. The goal of the tests was to evaluate the performance of the fusion imagery as compared to the individual modalities alone (visible and IR), as well as comparing it to direct view Omni IV intensifier tubes.

Several tasks were evaluated which included tracking people and objects, recognizing uniforms and weapons, detecting camouflage, seeing through smoke and vegetation, and recognizing vehicles of various types. All tests were performed after 8:00 pm under starlight conditions. The light level measured with a calibrated photometer was between 1.5-2.1 mLux. Tests were evaluated in both ground and water operations. Both fields-of-view pods were utilized at 35, 70, and 100m. for the wide fov and 100, 300, and 500 m. for the narrow fov. Special Forces personnel wore uniforms of various types, carried a variety of weapons, and performed exercises in the open, among vegetation, and in the water.

Figures 5a-c illustrate enhanced visible, LWIR and color fused imagery results with the narrow field-of-view pod. The fused color image shown in Figure 5c is obtained using the architecture shown in Figure 3. Notice how this fused result combines the complementary information provided by the source imagery. In this example, with two subjects at 100 m., both holding automatic weapons of various makes. The man on the left is wearing a ski mask which can be seen in the visible image but not in the IR image. Similarly, the weapons standout in the visible band which is also being preserved in the fused result. On the other hand, the IR band is leading to a more evident pop-out of the human targets which is preserved in the color fused in the form of color contrast.

All images were captured at their original wide dynamic range with a 640x480 resolution for the visible and 320x240 for the infrared. Here, we can see that the information and resolution from the Low-Light CCD is preserved in the form of the brightness contrast in the fused image. On the other hand, the IR vs. Visible imagery serves to "paint" the fused image in the blue-red gamut to code the thermal contrast information.

During these exercises, qualitative analysis were conducted by night vision experts from the US Army and various research laboratories. Here, observers were provided with an evaluation matrix as shown in Figure 6. Utilizing this table, they recorded their judgements in their ability to perform the various tasks while utilizing the different modalities (i.e. visible-

only, IR-only, or visible-IR fused). Some of these results are summarized in Figure 6.

In summary, the color fused result was found to provide high image quality, support enhanced depth perception down the field, and produce target pop-out capabilities. Also, not evaluated in Figure 6, were the Omni IV intensifier tubes due to the fact that they could not be imaged and recorded for analysis. However, users in the field reported being unable to perform the majority of the tasks involved in these tests.

6. Summary

We have shown that an effective strategy for the fusion of imagery derived from two complementary sensors is to emulate the early stages of opponent-color processing in the human visual system. Single-opponent color architectures are sufficient for fusing two sensors, such as a CCD camera and a thermal IR imager. A real-time fusion processor has been developed from commercial DSP boards for fusing the Lincoln Lab 640x480 low-light CCD with an uncooled 320x240 LWIR camera to provide color night vision. Field tests with night vision experts have corroborated psychophysical tests in the laboratory that support the use of color fused imagery in place of the original, separate bands.

7. Acknowledgments

This work was sponsored by the Defense Advanced Research Projects Agency, under Air Force Contract F19628-95-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and not necessarily endorsed by the U.S. Air Force.

8. References

1. A.M. Waxman, A.N. Gove, M.C. Seibert, D.A. Fay, J.E. Carrick, J.P. Racamato, E.D. Savoye, B.E. Burke, R.K. Reich, W.H. McGonagle, and D.M. Craig, "Progress on Color Night Vision: Visible/IR Fusion, Perception & Search, and Low-Light CCD Imaging", *Proc. SPIE Conf. on Enhanced and Synthetic Vision 1996,* **SPIE-2736**, 96-107, 1996.

2. A.M. Waxman, D.A. Fay, A.N. Gove, M.C. Seibert, J.P. Racamato, J.E. Carrick and E.D. Savoye, "Color night vision: Fusion of intensified visible and thermal IR imagery", *Synthetic Vision for Vehicle Guidance and Control,* **SPIE-2463**, 58-68, 1995.

3. A.M. Waxman, D.A. Fay, A.N. Gove, M.C. Seibert, and J.P. Racamato, "Method and apparatus for generating a synthetic image by the fusion of signals representative of different views of the same scene". *U.S. Patent 5,555,324,* issued 9/10/96 (filed 11/1/94); rights assigned to the MIT, 1996.

4. P. J. Burt and R. J. Kolczynski, "Enhanced image capture through fusion", *Fourth Internat'l Conf. on Computer Vision,* 173-182. Los Alamitos: IEEE Computer Society Press, 1993.

5. D. Ryan and R. Tinkler, "Night pilotage assessment of image fusion", *Helmet- and Head-Mounted Displays and Symbology Design Requirements II,* **SPIE-2465**, 50-67, 1995.

6. A. Toet, "Hierarchical image fusion", *Machine Vision and Applications,* **3**, 1-11, 1990.

7. A. Toet, "Multiscale contrast enhancement with applications to image fusion". *Optical Eng.,* **31**, 1026-1031, 1992.

8. A. Toet, L. J. van Ruyven and J. M. Valeton, "Merging thermal and visual images by a contrast pyramid", *Optical Eng.,* **28**, 789-792, 1989.

9. P. M. Steele and P. Perconti, "Part task investigation of multispectral image fusion using gray scale and synthetic color night vision sensor imagery for helicopter pilotage", *Targets and Backgrounds: Characterization and Representation III,* **SPIE-3062**, 88-100, 1997.

10. A. Toet, J. K. IJspeert, A. M. Waxman, and M. Aguilar, "Fusion of visible and thermal imagery improves situational awareness", *Proc. SPIE Conf. on Enhanced and Synthetic Vision,* **SPIE-3088**, 177-188, 1997.

11. A. Toet and J. Walraven, "New false color mapping for image fusion", *Optical Engineering,* **35**, 650-658, 1996.

12. D.A. Scribner, P. Warren, J. Schuler, M.P. Satyshur, and M.P. Kruer, "Infrared Color Vision: An approach to sensor fusion", *Optics and Photonics News,* 8, 27-32, 1998.

13. R. K. Reich, B. E. Burke, W. M. McGonagle, D. M. Craig, A. M. Waxman, E. D. Savoye, and B. B. Kosicki, "Low-light-level 640x480 CCD camera for night vision application", *Proceedings of the Meeting of the IRIS Specialty Group on Passive Sensors,* 1998.

14. P. Schiller, "The ON and OFF channels of the visual system", *Trends in Neuroscience,* **TINS-15**, 86-92, 1992.

15. P. Schiller and N. K. Logothetis, "The color-opponent and broad-band channels of the primate visual system", *Trends in Neuroscience,* **TINS-13**, 392-398, 1990.

16. F. S. Frome, S. L. Buck, and R. M. Boynton, "Visibility of borders: separate and combined effects of color differences, luminance contrast, and luminance level", J. Opt. Soc. Am., **71**, 145-150, 1981.

17. W. Fink, "Image coloration as an interpretation aid", *Proc. of SPIE/OSA Conf. on Image Process.,* **74**, 209-215, 1976.

18. S. A. Ellias and S. Grossberg,, "Pattern formation, contrast control, and oscillations in the short memory of shunting on-center off-surround networks", *Biological Cybernetics,* **20**, 69-98, 1975.

On the Maximum Number of Sensor Decision Bits Needed for Optimum Distributed Signal Detection

Jun Hu and Rick S. Blum *
EECS Department
Lehigh University
Bethlehem, PA 18015

Abstract ***Distributed multisensor detection problems with quantized observation data are investigated for cases of nonbinary hypotheses. The observations available at each sensor are quantized to produce a multiple-bit sensor decision which is sent to a fusion center. At the fusion center, the quantized data are combined to form a final decision using a predetermined fusion rule. Firstly, it is demonstrated that there is a maximum number of bits which should be used to communicate the sensor decision from a given sensor to the fusion center. This maximum is based on the number of bits used to communicate the decisions from all the other sensors to the fusion center. If more than this maximum number of bits is used, the performance of the optimum scheme will not be improved. Then in some special cases of great interest, the bound on the number of bits that should be used can be made significantly smaller. Finally, the optimum way to distributed a fixed number of bits across the sensor decisions is described for two-sensor cases. Illustrative numerical examples are presented at the end of this paper.***

Keywords: distributed signal detection, decentralized detection, quantization for detection, M-ary hypothesis, multiple bit sensor decisions.

*This paper is based on work supported by the Office of Naval Research under Grant No. N00014-97-1-0774 and by the National Science Foundation under Grant No. MIP-9703730.

1 Introduction

Signal detection algorithms which process quantized data taken from multiple sensors continue to attract attention [1, 2, 3, 4, 5]. Such algorithms have been classified as distributed signal detection algorithms. The majority of distributed signal detection research has focused on cases with statistically independent observations, binary hypothesis testing problems and binary sensor decisions [6, 7]. Studies on nonbinary hypothesis testing problems have been lacking. An early paper on this topic [8] provided equations describing the necessary conditions for the optimum sensor processing. A more complete discussion which includes a thorough treatment of the necessary conditions for the case of independent observations is given in [9]. A nice discussion of the complexity of cases with dependent observations is also given in [9]. Neither [8] nor [9] give any numerical examples. A numerical procedure for finding the optimum processing scheme was provided in [10] for cases with dependent observations and nonbinary hypothesis and a few numerical examples are provided. However, studies of the properties of optimum schemes have been lacking.

In this paper, we demonstrate that no more than a certain number of bits should be used to communicate a sensor decision from a particular sensor to the fusion center. Using more than that number of bits at a given sensor is unnecessary and will not generally lead to increases in performance. The number of bits which should be used is limited by the number of bits used to communicate all the other sensor

decisions to the fusion center. The paper is organized as follows. In Section 2 we present the model for the observations and the distributed decision making. In Section 3, we prove that a finite number of bits can be used for the sensor decision at a given sensor without sacrificing any performance. In Section 4, we strengthen our results for some special cases of great interest. In Section 5, we consider how to allocate bits between two sensors in the case where the total number of bits used for sensor decisions is fixed. Section 6 provides some illustrative numerical examples. Finally, our conclusions appear in Section 7.

2 Problem Formulation

Consider a multiple hypothesis testing problem using L sensors, where $H_0, H_1, \ldots, H_{K-1}$ are K hypotheses, and $P(H_i), i = 0, \ldots, K-1$ are the prior probabilities for each hypothesis. Assume an m_k dimensional vector of observations

$$y_k = [y_{k,1}, y_{k,2}, \ldots, y_{k,m_k}], \quad y_{k,l} \in \mathbf{R}$$

is observed at the kth sensor, $k = 1, \ldots, L$. Define $y = [y_1, y_2, \ldots, y_L]$, and let $p(y \mid H_i), i = 0, \ldots, K-1$ denote the known joint conditional probability density functions (pdfs) under each hypothesis. Note that we do not assume $y_1, y_2, \ldots, y_L$ are independent when conditioned on the hypothesis. Let $P(D_j \mid H_i)$ represents the probability that a final decision for hypothesis H_j is made given hypothesis H_i is true.

Here we consider the criterion of minimum probability of error.

$$P_e = 1 - P_c \tag{1}$$

where

$$P_c = \sum_{i=0}^{K-1} P(H_i) P(D_i \mid H_i)$$

Our goal is to design a system minimizing P_e.

A typical centralized detection system requires each sensor to transmit its observations to a fusion center, then the fusion center produces a decision U_0 as

$$U_0 = d(y) = d(y_1, y_2, \ldots, y_L) \tag{2}$$

where d is the decision rule and $U_0 = j$ corresponds to deciding that hypothesis H_j is true. A well known decision rule for centralized detection systems is to compare the joint likelihood ratio to a threshold. Due to a variety of reasons, it may be difficult to realize such a centralized system in practice. In a distributed detection system, the observation data at the kth sensor are first quantized to a discrete variable U_k taking on only N_k possible values. For simplicity[1], it is common to choose N_k to be a power of K, or $N_k = K^{n_k}$. Thus U_k can be represented as a n_k-digit, K-ary number produced by an n_k dimensional vector of decisions

$$U_k = [U_{k,0}, U_{k,1}, \ldots, U_{k,n_k-1}],$$

$$U_{k,l} \in \{0, 1, \ldots, K-1\}$$

made at the kth sensor, using a group of decision rules

$$\begin{aligned} d_k &= [d_{k,0}, d_{k,1}, \ldots, d_{k,n_k-1}] \\ U_{k,l} &= d_{k,l}(y_k) \quad k = 1, \ldots, L, \\ &\qquad l = 0, \ldots, n_k - 1 \end{aligned} \tag{3}$$

Please note that each $d_{k,l}$ denotes a scalar function while d with only single subscript denotes a vector of functions. To avoid confusion between N_k and n_k, N_k is called the number of quantization levels in the sequel. Define the combination of all but the last sensor decision as $U = [U_1, U_2, \ldots, U_{L-1}]$. All sensor decisions are sent to the fusion center to determine a final decision U_0 with a chosen fusion rule F

$$\begin{aligned} U_0 &= F(U; U_L) = F(U_1, U_2, \ldots, U_L) \\ &= F(d_{1,0}(y_1), \ldots, d_{1,n_1-1}(y_1), \ldots \\ &\quad , d_{L,0}(y_L), \ldots, d_{L,n_L-1}(y_L)) \end{aligned} \tag{4}$$

where U_0 and all $U_{k,l}$ can take on any of the values $0, \ldots, K-1$.

[1] Also see Remark 1 following Theorem 1.

3 Limits on Number of Bits Used in Sensor Decisions

Here we will show that the number of bits which should be used to represent the sensor decision at one given sensor is limited by the number of bits used at all the other sensors. Furthermore, due to the non-uniqueness of describing a distributed detection system by defining a fusion rule and a set of sensor decision rules, there are many combinations of fusion rules and sensor decision rules which are optimum. To see this just consider a $K = 2$ case. Complementing all the sensor decision rule outputs and the fusion rule inputs will change nothing. We also present one specific fusion rule that can always be used to achieve optimum global performance for some special cases.

Theorem 1 *If the combination of the first $L-1$ sensor decisions, U, consists of n components, where $n = \sum_{k=1}^{L-1} n_k$, then a scheme which obtains optimum global performance can be found in the case where the Lth sensor makes $n_L = K^n$ decisions. Thus using an Lth sensor with $n_L > K^n$ will not improve the optimum global performance.*

Outline of the proof. U is a vector of K-valued integers. We can also see it as a number in the K-ary number system. Then U takes on the values of $0, \ldots, K^n - 1$ and we can view $F(U; U_L)$ as a function of two variables, U and U_L. The theorem will be proved if we show that whenever $n_L > K^n$

$$\exists i, j = 0, \ldots, K^{n_L} - 1, \quad i \neq j$$
$$\forall U = 0, \ldots, K^n - 1 \quad F(U; i) \equiv F(U; j) \quad (5)$$

Thus it does not make sense to distinguish the decision of $U_L = i$ and $U_L = j$ since either decision leads to exactly the same final decision. We can choose a new decision rule at the Lth sensor which uses just one value to represent these two values i, j without changing performance. If U_L is fixed, $F(U; U_L)$ degenerates to a function of one variable which maps each possible value of U into a possible value of U_0. Alternately we can say each value of U_L will correspond to a function from U to U_0. Since U takes on K^n different values and U_0 takes on K different values. Obviously there are only K^{K^n} different mapping patterns between the domain and the range. Hence there are totally K^{K^n} different functions from U to U_0. Whenever $n_L > K^n$, the number of possible values of U_L is greater than the number of different functions. Thus at least two values of U_L will correspond to the same function. This statement is equivalent to (5). □

Remark 1 *Suppose the number of quantization levels at the kth sensor, N_k, can not be represented as a power of K, then a slightly more general result can be obtained. Using the same argument we can prove an optimum scheme can be found in the case of $N_L = K^N$, where $N = \prod_{k=1}^{L-1} N_i$.*

Theorem 2 *In the case of $n_L = K^n$, we can obtain optimum performance by employing the specific fusion rule*

$$U_0 = F(U; U_L) = U_{L,U} \quad (6)$$

where $U_L = [U_{L,0}, U_{L,1}, \ldots, U_{L,K^n-1}]$, $U = 0, \ldots, K^n - 1$.

Outline of the proof. We have shown that a scheme with $n_L = K^n$ can be used to achieve optimum performance. If we could also show that the fusion rule in (6) can be used with some groups of sensor decision rules to implement any scheme in this case, Theorem 2 is proved. Consider an arbitrary scheme for the case of $n_L = K^n$. It consists of a fusion rule $\hat{F}$ and a group of decision rules $\hat{d}_1, \ldots, \hat{d}_L$. For a given K-ary integer B taking on any of the values $0, \ldots, K^{K^n} - 1$, let $[b_0, \ldots, b_{K^n-1}]$ denote its digits. Now define a group of sets

$$\Omega_B = \{y_L : \hat{F}(0; \hat{d}_L(y_L)) = b_0, \ldots, \hat{F}(K^n - 1; \hat{d}_L(y_L)) = b_{K^n-1}\}$$

After determining the set Ω_B corresponding to each possible value of B, we can use these sets to define a new decision rule for the Lth sensor.

$$d_{L,l}(y_L) = b_l \quad if \; y_L \in \Omega_B, \quad l = 0, \ldots, K^n - 1$$

The fusion rule in (6), with the sensor decision rules $\hat{d}_1, \ldots, \hat{d}_{L-1}$ and d_L can insure the overall scheme produces the same output for any y_L. Thus the fusion rule in (6) will always allow us to obtain optimum performance. Further, generally we can not decrease n_L any more without affecting the global performance. □

4 A Stronger Result

In Theorem 1, the upper bound of n_L grows rapidly as n increases. However in some more specific cases, we can have a stronger result. As an example, consider a hypothesis testing problem with statistically independent Gaussian data with different mean vectors under each hypothesis.

$$\begin{aligned} m_k = 1, y_k \in \mathbf{R}, \quad k = 1, \ldots, L \\ \mathbf{H_i} : y = [y_1, y_2, \ldots, y_L] \sim N(M_i, C), \\ i = 0, \ldots, K-1 \end{aligned} \tag{7}$$

where $C = diag[\sigma_1^2, \sigma_2^2, \ldots, \sigma_L^2]$, σ_j^2 are the variances of observation data y_j and $M_i = [E\{y_1 \mid H_i\}, E\{y_2 \mid H_i\}, \ldots, E\{y_L \mid H_i\}]^T, i = 0, \ldots, K-1$ are the mean vectors under each hypothesis.

Definition 1 *A single-interval is a set of real numbers, $\{x \mid a < x < b\}$. Such a single-interval could be a finite length interval, the entire real line (a=-∞ and b=+∞), a semi-infinite interval (either a=-∞ or b=+∞ while the other is finite), or the empty set (a=b).*

Theorem 3 *For the Gaussian shift in mean problem in (7), a scheme which obtains optimum global performance can be found in the case of $n_L = n + 1$.*

Outline of the proof. It is sufficient to prove that all optimum schemes used in this case can be implemented by a new scheme with $n_L \leq n + 1$. Let us consider an arbitrary optimum scheme. For statistically independent observations, It is equivalent to a set of comparisons of a likelihood ratio to a corresponding threshold. Each comparison yields a semi-infinite interval. So the intersection of these comparisons is a single-interval. For fixed U, $F(U; d_L(y_L))$ is a multiple-step function that can be determined by those thresholds between neighboring single-intervals, the number of which is less than K. For each values of U, we need less than K thresholds, $t_{1,U}, \ldots, t_{K-1,U}$, and U has K^n possible values, so less than $K \cdot K^n = K^{n+1}$ thresholds can completely determine $F(U; d_L(y_L))$. We can put all the thresholds together and sort them by ascending order. The sorted sequence of thresholds are denoted by $t'_1 < \ldots < t'_{K^{n+1}-1}$. These thresholds will divide the entire real axis of y_L into not more than K^{n+1} intervals. For any value of U, all points in each interval $\{y_L : t'_k < y_L < t'_{k+1}\}$ will yield the same $F(U; d_L(y_L))$ If we assign a different value of U_L to each interval, the new scheme will satisfy $n_L \leq n + 1$ and implement the considered arbitrary optimum scheme. □

Remark 2 *From the proof of Theorem 3, it is clear that this stronger result is only based on the optimum sensor test being a direct threshold test of the observation. In the case we consider, this is true due to the monotone likelihood ratio property of Gaussian pdfs. Hence the result can be applied to any problem with independent observations and monotone sensor likelihood ratios. Studies have shown that a large family of pdfs have monotone likelihood ratios [11]. Cases of dependent observations may also possess the required property. Some cases with Gaussian pdfs are shown to have this property in [12]. Theorem 3 can be used in all these cases.*

5 A Case with Fixed Overall Number of Sensor Decisions

In a two-sensor system where the total number of bits is fixed, we denote a scheme by (n_1, n_2), where n_1 and n_2 are the number of bits used for decision from the first sensor and the second sensor respectively. Note that increasing both n_1 and n_2 will improve the performance, so the constraint is reasonable.

Theorem 4 *Given $n_1 + n_2$ is fixed, then regardless of the statistical characteristics of ob-*

servation data, a scheme which obtains optimum global performance and satisfies the following inequality can be found.

$$log_K n_1 \leq n_2 \leq K^{n_1}$$

Outline of the proof. Suppose we have found a scheme (n_1, n_2) which achieves optimum global performance but $n_2 > K^{n_1}$. From Theorem 1, we know the excessive decisions used at the second sensor is wasted. We can allocate some of them to the first sensor to improve the overall performance. Thus we prove $n_2 \leq K^{n_1}$. Now switching n_1 and n_2 in the argument, we can also prove that $n_1 \leq K^{n_2}$, or $log_K n_1 \leq n_2$. □

Remark 3 *Suppose the number of quantization levels N_k can not be represented as a power of K. Using the relationship between N_k and n_k, we can get an inequality for N_1 and N_2.*

$$log_K N_1 \leq N_2 \leq K^{N_1}$$

Theorem 5 *In those special cases with independent observations, known signals and noise pdfs which lead to monotone likelihood ratio, a scheme which obtains optimum global performance and satisfies the following inequality can be found.*

$$n_1 - 1 \leq n_2 \leq n_1 + 1$$

Outline of the proof. The proof of this theorem is quite simple. All we need to do is to substitute the stronger result from Theorem 3 in place of the result from Theorem 1 in the proof of Theorem 4. Then we can prove both $n_2 \leq n_1 + 1$ and $n_1 \leq n_2 + 1$. So the theorem is proved. This result implies the optimum scheme would like to divide the number of decisions evenly or nearly evenly between the two sensors. If the sum of $n_1 + n_2$ is even, then $n_1 = n_2$, otherwise $n_1 = n_2 \pm 1$. □

Remark 4 *In the above cases, if the number of quantization levels N_k can not be represented as a power of K, the inequality for N_1 and N_2 will be*

$$\frac{N_1}{K} \leq N_2 \leq KN_1$$

As opposed to the result of Theorem 5, the value of N_1 and N_2 are still to some extent undetermined. Only numerical techniques can find the exact N_1, N_2 used by the optimum scheme.

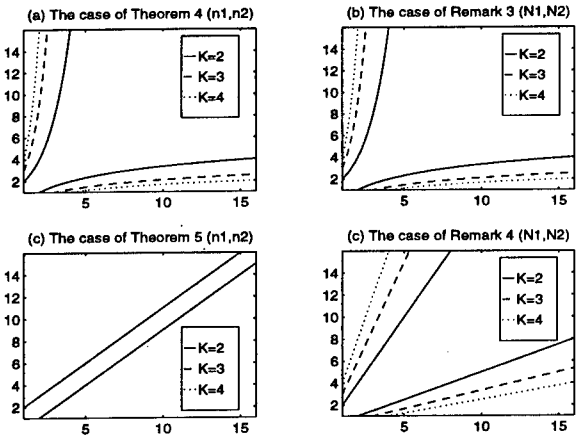


Figure 1: Regions for optimum scheme

We can demonstrate our results in the plane of (N_1, N_2) or the plane of (n_1, n_2), as shown in Figure 1. In each of the four parts of this figure we label the Theorem or Remark it illustrates. We know only schemes in the region between the two curves can obtain optimum global performance, so we never need to search in those regions outside the two curves.

6 Numerical Results

In the following numerical investigations, we consider detecting a known signal in Gaussian noise with two sensors. If both U_1 and U_2 are binary variables, there are $2^{2^{1+1}} = 16$ possible nonrandomized fusion rules: two rules are trivial $(U_0 = 1, U_0 = 0)$, four rules ignore one of the sensors $(U_0 = U_1, U_0 = \overline{U_1}, U_0 = U_2, U_0 = \overline{U_2})$, four are AND rules $(U_0 = U_1U_2, U_0 = U_1\overline{U_2}, U_0 = \overline{U_1}U_2, U_0 = \overline{U_1U_2})$, four are OR rules $(U_0 = U_1 + U_2, U_0 = U_1 + \overline{U_2}, U_0 = \overline{U_1} + U_2, U_0 = \overline{U_1} + \overline{U_2})$, and two rules are exclusive-or operations $(U_0 = U_1 \oplus U_2, U_0 = \overline{U_1} \oplus \overline{U_2})$. If either U_1 or U_2 takes on more than two values, or one bit, there will be more fusion rules. We can use the following equations to calculate the

total number of possible fusion rules.

$$M = K^{K^{n_1+n_2}} \tag{8}$$

Of course, theoretically we could compute the performance of the best scheme using each possible nonrandomized fusion rule and pick the one with minimum probability of error. But as n_1 and n_2 grow, this require significant computation. For instance, in a scheme of (1,3) or (2,2), the number of fusion rules is $M = 2^{2^4} = 256$. which is 16 times of that in a scheme of (1,1).

For a fixed fusion rule, we employ a discretized Gauss-Seidel iterative algorithm [10, 12] and attempt to find all solutions to the necessary conditions in Appendix A. With every set of initial decision rules we have tried, the algorithm always converges to the same result in a finite number of iteration steps. Due to this, we take the solution we found as the optimum solution and use this group of sensor decision rules and the fusion rule to calculate P_e.

Assume the observation data at the sensors consists of different constant signals for each hypothesis and additive Gaussian noise $n_1 \sim N(0,3)$ and $n_2 \sim N(0,2)$.

$$\mathbf{H_0}: Y_1 = -1 + n_1, \quad Y_2 = -1 + n_2$$

$$\mathbf{H_1}: Y_1 = \ 1 + n_1, \quad Y_2 = \ 1 + n_2$$

We also assume all hypotheses have equal prior probabilities and the noise samples at the two sensors are statistically independent. Therefore,

$$P(H_0) = P(H_1) = \frac{1}{2}$$

$$p(Y_1, Y_2 \mid H_0) \sim N(\begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix})$$

$$p(Y_1, Y_2 \mid H_1) \sim N(\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix})$$

After trying all possible fusion rules for the scheme of (1,1), (1,2) and (1,3), we find the best rule for each scheme. The resulting optimum decision regions and P_e for optimum centralized rules and optimum distributed rules

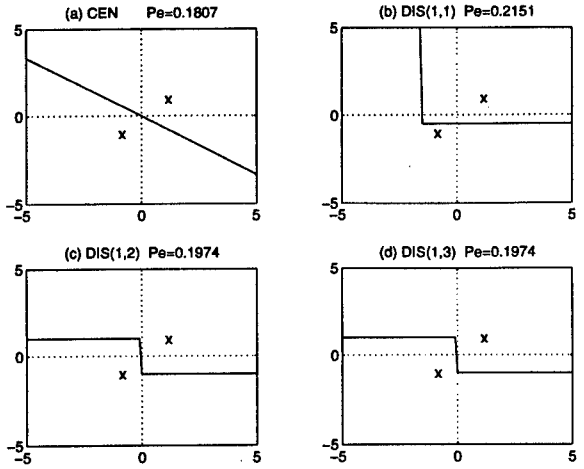


Figure 2: Optimum decision regions (I)

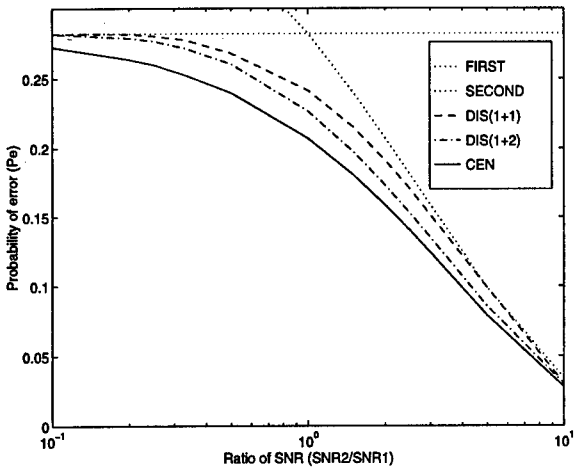


Figure 3: Optimum global performance (I)

with different number of bits used in the sensor decisions are provided in Figure 2. The notation "CEN" stands for centralized rules, and the notation "DIS" indicates distributed rules. In each part, a line divided the entire plane into two regions. Each region is assigned to a hypothesis. The little "x"s represent the signal for each hypothesis. Observing these interesting examples, we see that optimum centralized regions for this problem have boundary which is a straight line. All distributed fusion rules try to use a combination of horizontal lines and vertical lines to approximate the straight line. The more bits are used, the better approximation is achieved. But when $n_L > K^n$, in this case $n_2 > 2^1$, excessive decisions at the

second sensor can not improve the optimum global performance. This is illustrated by part (c) and (d) of Figure 2.

In Figure 3, we vary SNR at the second sensor while fixing SNR at the first sensor and compute P_e for centralized and various distributed schemes. The dotted line represents the result obtained by using only the first or the second sensor. The figure indicates that when the ratio of SNRs is less than 0.1, all distributed schemes will choose only the first sensor to make a decision. It is not surprising because the second sensor can provide little information. When the ratio of SNRs is greater than 10, all distributed schemes have similar performances to the centralized schemes. We can see that for nearly identical SNRs there is a distinguishable improvement in global performance when using two sensors instead of one. However the difference of performance between centralized scheme and distributed scheme is much smaller. This suggests that only one or two binary decisions can give adequate performance with considerable complexity decrease in cases like this one.

We continue our investigation in this case for other distributed schemes. This time instead of fixing the number of decisions at the first sensor, we fix the total number of decisions at the two sensors. Again we have tried all possible fusion rules and show the best results we have found in Figure 4. From Figure 4, we see that the scheme of (2,2) implements the best approximate boundary so that it yields the best performance in this case. Figure 5 provides the curve of probabilities of error plotted against the ratio of sensor SNRs for this case. We can see that the three distributed schemes perform quite well. They always perform much better than the scheme using only one sensor. Moreover, the scheme of (2,2) always yields the best global performance of the three distributed schemes.

7 Conclusion

We investigated distributed detection problems for cases with nonbinary hypotheses. We uncover some interesting general properties of

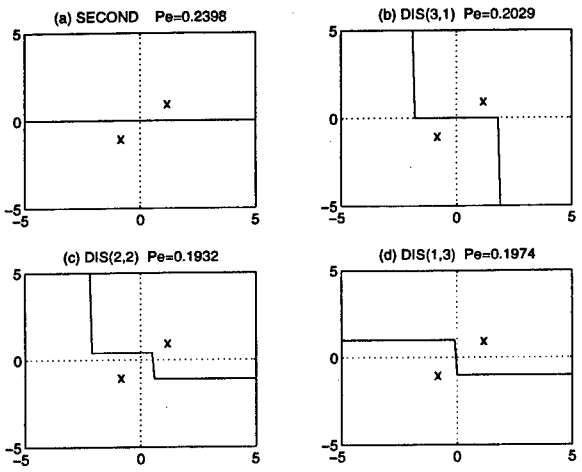


Figure 4: Optimum decision regions (II)

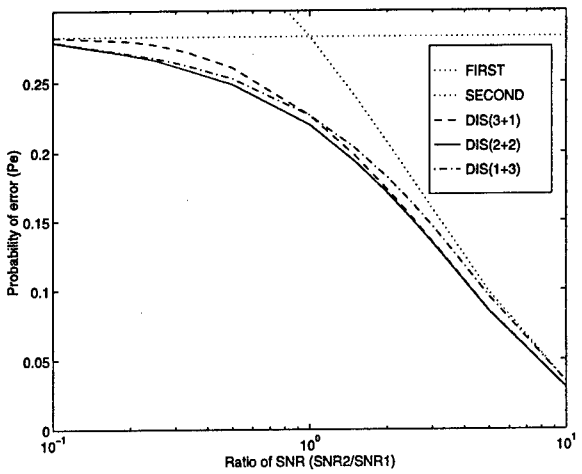


Figure 5: Optimum global performance (II)

distributed detection schemes. In particular we consider how much information must be transmitted from one sensor when the total amount of information transmitted from all the other sensors is fixed. We show there is an upper bound on the number of bits for a given sensor decision that should be used. Using more bits at the sensor will not generally lead to improvements in the performance of the optimum scheme. Further, in some special cases, for example, with independent observations and monotone likelihood ratios, we show that a stronger result can be obtained. The stronger result provides with a much smaller bound on the number of bits which should be used. These results give important guidance

in the design of distributed detection schemes. Finally we analyze the issue of how to allocate bits across the sensors when the total number of bits used for all sensor decisions is fixed.

Appendix

A Necessary Conditions for the Optimum Sensor Rules

Under a fixed fusion rule, the sensor decision rules must satisfy a group of necessary conditions to minimize P_e. Due to our configuration, $Prob(U_0 = i \mid y)$ is an indicator function

$$Prob(U = i \mid y) = \begin{cases} 1 & \text{if } d(y) = i, \\ 0 & \text{otherwise.} \end{cases}$$

Definition 2 *A class of functions $L_{j,k,l}(y_k)$ are defined as follows*

$$L_{j,k,l}(y_k) = \sum_{i=0}^{K-1} \int \cdots \int Prob(U_0 = i \mid y, d_{k,l}(y_k) = j) P(H_i) p(Y \mid H_i) \frac{dy}{dy_k} \quad (9)$$

where $dy = \prod_{k=1}^{L} dy_k$

Theorem 6 *Suppose P_e is minimized, then $\forall k = 1, \ldots, L, l = 0, \ldots, n_k - 1, d_{k,l}(y_k)$ must satisfy the following conditions*

$$\forall y_k \in \mathbf{R}^{m_k} \quad d_{k,l}(y_k) = j \quad if \quad \forall j_0 \neq j \quad \frac{L_{j,k,l}(y_k)}{L_{j_0,k,l}(y_k)} > 1 \quad (10)$$

References

[1] R. R. Tenney and N. R. Jr. Sandell, "Detection with distributed sensors," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-17, pp. 501-510, April 1981.

[2] Z. Chair and P. K. Varshney, "Optimal data fusion in multiple sensor detection systems," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-22, pp. 98-101, Jan. 1986.

[3] I. Y. Hoballah and P. K. Varshney, "Distributed Bayesian signal detection," *IEEE Transactions on Information Theory*, Vol. IT-35, No. 5, pp. 995-1000, Sept. 1989.

[4] P. K. Willett and D. Warren, "The suboptimality of randomized test in distributed and quantized detection systems," *IEEE Transactions on Information Theory*, Vol. IT-38, pp. 355-361, 1992.

[5] W. W. Irving and J. N. Tsitsiklis, "Some properties of optimal thresholds in decentralized detection," *IEEE Transactions on Automatic Control*, Vol. 39, No. 4, pp. 835-838, April 1994.

[6] R. Viswanathan and P. K. Varshney, "Distributed detection with multiple sensors - part I: fundamentals," *Proceedings of the IEEE*, pp. 54-63, Jan. 1997.

[7] R. S. Blum, S. A. Kassam, and H. V. Poor, "Distributed detection with multiple sensors: part II - advanced topics," *Proceedings of the IEEE*, pp. 64-79, Jan. 1997.

[8] F. A. Sadjadi, "Hypothesis testing in a distributed environment," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-22, No. 2, pp. 134-137, March 1986.

[9] J. N.Tsitsiklis, "Decentralized Detection," *Advances in Statistical Signal Processing - Vol. 2: Signal Detection*, H. V. Poor and J. B. Thomas, Eds. JAI Press: Greenwich, CT, 1993.

[10] Z. B. Tang, K. R. Pattipati, and D. L. Kleinman, "A distributed M-ary hypotheses testing problem with correlated observations," *IEEE Transactions on Automatic Control*, Vol. 37, No. 7, pp. 1042-1046, July 1992.

[11] E. L. Lehmann, *Testing statistical hypotheses*, New York: Wiley, 1986.

[12] P. F. Swaszek, P. Willett and R. S. Blum, "Distributed detection of dependent data - the two sensor problem," *29th Annual Conference on Information Sciences and Systems*, Princeton University, pp. 1077-1082, Princeton, N.J., March 1996.

Data Fusion in a Multi-Sensor Mine Detection System

Wilson Sing-Hei So
Computing Devices Canada
1020 - 68th Avenue N.E., Calgary, Alberta,
Canada T2E 8P2
Email: wilson.so@cdcgy.com

Ray Kacelenga
Computing Devices Canada
1020 - 68th Avenue N.E., Calgary, Alberta,
Canada T2E 8P2
Email: ray.kacelenga@cdcgy.com

Abstract

This paper describes the investigation by Computing Devices Canada (CDC) of the synergistic combination of detection results from multiple scanning sensors, using data fusion techniques, in the detection of buried anti-tank (AT) mines.

I. Introduction

Landmines are currently used in all types of warfare, from local conflicts to high level military operations. They are inexpensive to make and easily deployed. The proliferation of landmines throughout the world, resulting from them being indiscriminately used during regional conflicts, has caused disastrous consequences in resettlement and economic renewal.

Mine detection is a difficult problem and requires the use of multiple sensors to achieve satisfactory detection levels in a variety of operating conditions. Currently, two types of mine detection technology are used. The first looks for anomalies associated with the presence of landmines, e.g. infrared (IR) imager, minimum metal detector (MMD), and ground penetrating radar (GPR). The second detects the presence of explosives directly. Thermal neutron activation (TNA) and nuclear quadrupole resonance (NQR) are two such techniques which detect the bulk nitrogen content of the explosives. When the technique detects explosives in trace amount, it is called trace explosive detection of which chemical sensing is an example [1].

While each of these sensor technologies is effective in detecting landmines in certain conditions, each has an associated false alarm rate (FAR) which is often excessive and impractical for mine clearance operations. In response to an urgent Canadian Forces (CF) operational requirement, Computing Devices Canada (CDC) and the Defence Research Establishment Suffield (DRES) have co-developed a multi-sensor mine detection system which employs data fusion techniques to reduce the system-level FAR such that mine detection operations can proceed at practical rates of advance. This paper describes these techniques. The resulting mine detection prototype, shown in Figure 1, has a number of unique characteristics as listed below [2]:

- The use of multiple scanning sensors, the IR imager, the MMD, and the GPR, to increase the probability of detection (Pd);
- The use of data fusion to reduce the FAR of detection;
- The use of a confirmation sensor, the TNA point detector, to further reduce the FAR of detection;
- The use of a remotely-operated vehicular sensor platform for personnel safety;
- The use of an operator to select IR targets; and
- The use of a marking system to mark potential mine locations for the mine clearing crew.

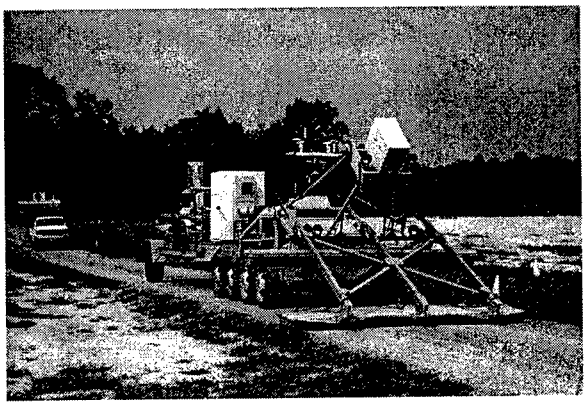


Figure 1. The Multi-sensor Mine Detection Prototype

The detection process begins with the three scanning sensors identifying potential mine targets which are reported as individual sensor alarms to the data fusion processor. Initial data fusion processing reduces the potential mine targets by grouping individual sensor alarms into equivalence classes. Members from an equivalence class are assumed to originate from the same location. The TNA point detector is next positioned over the suspected locations to confirm the presence of landmines.

Other mine detection systems have also been prototyped by a few other companies. An account of these other systems can be found in [3]. However, to the best of the authors' knowledge, the CDC system is the first in production.

The remainder of this paper is divided into five sections. Section II gives an overview of the application of data fusion to mine detection. Section III discusses in detail the spatial correspondence and scanning sensor fusion modules on which this paper is based. The results of our study are presented in Section IV, followed by a discussion of the results in Section V. Section VI summarizes the paper.

II. Data Fusion

The overall data fusion process includes the following primary components:

- Calibration;
- Navigation Sub-system;
- Spatial Registration;
- Spatial Correspondence;
- Scanning Sensor Fusion; and
- Confirmation Fusion.

2.1 Calibration

Calibration refers to the overall process used to derive reference frame transformation, optical, and auxiliary sensor calibration parameters for the system. It is accomplished through a combined process of geometric calibration and optical calibration. Each sensor, scanning or confirmation, has its own frame of reference. So too does the vehicle, the navigation sub-system and its components, and all auxiliary encoders and sensors which measure relative positions or angles of system components. Geometric calibration gives numerical parameters for translations and rotations relating the various reference frames to one another. This information is essential in order to transform positional information, originally reported relative to a sensor reference frame, to the vehicle-centered reference frame. Optical calibration of the FLIR is performed so that operator designations within the displayed imagery can be transformed to positional vectors relative to the IR reference frame. Proper geometric calibration of auxiliary sensors is used to determine the TNA sensor position relative to the vehicle, which is also essential.

2.2 Navigation Sub-system

The navigation problem is one of state estimation which filters and transforms raw navigation sensor information to derive robust and highly accurate estimates of the motion state of the system. The vehicle motion state consists of translational velocity, translational acceleration, attitude, and angular velocity. The navigation sensors include a ground speed measurement unit, a three-axis accelerometer unit, and a three-axis rate gyroscope unit.

Measurements provided by the navigation sensors are input to the navigation processor which derives the motion state through the use of Kalman filtering. The derived motion state is used in the registration of scanning sensor alarms in a common reference frame.

2.3 Spatial Registration

Spatial registration is the process of transforming positional information from any of the three scanning sensors to a common frame of reference. The accuracy of this process is highly dependant on the accuracy of the calibration process and the navigation filters. Once the sensor alarms are spatially registered in the local world reference frame, the detection information can be displayed to the operator(s) in a spatially correct manner.

2.4 Spatial Correspondence

Once all sensor alarms are spatially registered in a common reference frame, spatial correspondence algorithms are applied to partition the set of sensor alarms into classes, with each class representing those sensor alarms which could have resulted from the same local patch of ground or a single landmine. The correspondence decision for any two sensor alarms is based on their positions and the variance in this information.

2.5 Scanning Sensor Fusion

The information contained in the sensor alarms within a correspondence class is used to determine an overall position and confidence level for the suspect patch of ground. The overall confidence level is derived through a weighted summation strategy in which the weights are computed based on environmental and operational parameters. If the overall confidence level for a correspondence class is significant, a position for placement of the TNA sensor is computed, and the system automatically stops and positions the TNA point detector over the suspected mine location.

2.6 Confirmation Fusion

Measurements from the TNA sensor generate a confidence level that the local patch of ground under observation contains a sufficient amount of nitrogen to indicate the presence of a landmine. This confidence level is combined with the scanning sensor confidence level for this local patch of ground in order to generate the system confidence that this location contains a landmine. If this system confidence level is significant, a detection is declared, followed by the firing of the marking system.

III. Spatial Correspondence and Scanning Sensor Fusion

The following discussion is concerned with the spatial correspondence and scanning sensor fusion components/modules shown in Figure 2. Therefore, it is assumed that a sensor alarm has already been registered in a common reference frame. Each sensor reports an (x, y) alarm position and a corresponding detection confidence value.

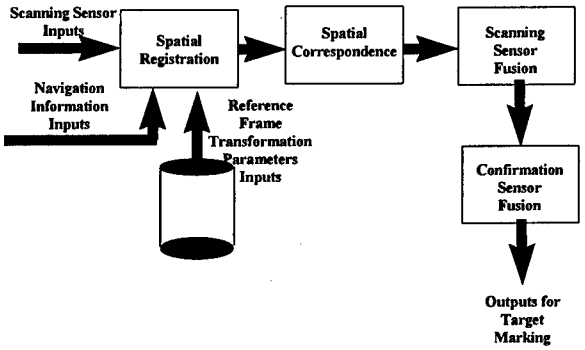


Figure 2. The Data Fusion Process

Two gating strategies, the error ellipse-based gating strategy and the chi-square gating strategy, will be presented below as candidates for the spatial correspondence module. Both gating strategies are followed by the use of a heuristic-based confidence value updating method and a Kalman-based

positional update method in the scanning sensor fusion module.

3.1 Error Ellipse-based Gating Strategy

The error ellipse-based gating strategy is based on the premise that any alarm position is surrounded by an error ellipse. The minor and major axes of the ellipse can be derived directly from the variances of the sensor localization error in the x- and y-directions. Any two alarms are said to be in correspondence if their error ellipses intersect. The errors in the x and y-directions can take on different values.

Determining the intersections of two error ellipses is equivalent to solving a fourth order polynomial. In general, intersection yields purely real roots for each intersection point. While it is necessary to check for real roots as a condition for intersection, this check by itself is not sufficient. Two special cases exist where the fourth order polynomial offers no root but the two sensor alarms are still in correspondence. Therefore, it is necessary to first check for the case of two error ellipses being identical and co-located as well as for the case of one error ellipse being contained within the other and the two are co-located. The existence of either condition is sufficient to declare correspondence without having to solve the fourth order polynomial.

3.2 Chi-square Gating Strategy

The chi-square gating strategy uses the Mahalanobis distance metric in calculating the separation between two sensor alarms [4]. The Mahalanobis distance between two sensor alarms, with position vectors $\mathbf{a}_1$ and $\mathbf{a}_2$, is defined as follows:

$$Dist_{Maha} = \sqrt{(\mathbf{a}_1 - \mathbf{a}_2)^T \mathbf{R}^{-1} (\mathbf{a}_1 - \mathbf{a}_2)}$$

where $\mathbf{a}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}, i = 1$ and 2. The residual, $\mathbf{r} = \mathbf{a}_1 - \mathbf{a}_2$, yields a vector that indicates how far apart the two alarms are. The distance measure provides a single figure that quantifies the spatial separation between the two alarms. The residual covariance matrix, $\mathbf{R}$, is defined as

$$\mathbf{R} = E\{\mathbf{r}\mathbf{r}^T\}.$$

$\mathbf{R}$ is calculated from the individual sensor localization covariance matrices, $\mathbf{P}_{IR}$, $\mathbf{P}_{MMD}$, and $\mathbf{P}_{GPR}$. $\mathbf{P}$ for each scanning sensor is formed from the individual sensor localization error in the x- and y-directions. Since the sensor localization error in the x- and y-directions are assumed to be uncorrelated for each scanning sensor, the off-diagonal elements are set to zero and $\mathbf{P} = \begin{bmatrix} \text{var}_x & 0 \\ 0 & \text{var}_y \end{bmatrix}$. $\mathbf{R}$, used in the Mahalanobis distance formula when associating two sensor alarms, is simply the sum of the two individual sensor localization covariance matrices. This simple result follows directly from a well known property of Gaussian random variables which states that all Gaussian random variables remain Gaussian under linear transformation [5].

All the alarms within a spatial correspondence class must satisfy the requirement that the Mahalanobis distance between each and every pair of alarms is less than a pre-determined threshold. This threshold (or gating distance) is chosen from the Chi-square distribution table based on the Chi-square probability and the number of degrees of freedom in the data. The positional vector of an alarm consists of its x and y components arranged in column format. Therefore, the square of the Mahalanobis distance between two sensor alarms, for the two dimensional case, can be written as:

$$Dist_{Maha}{}^2 = \begin{bmatrix} x_1 - x_2 \\ y_1 - y_2 \end{bmatrix}^T \begin{bmatrix} \frac{1}{\sigma_x^2} & 0 \\ 0 & \frac{1}{\sigma_y^2} \end{bmatrix} \begin{bmatrix} x_1 - x_2 \\ y_1 - y_2 \end{bmatrix}.$$

The covariance-weighted residual of the two sensor alarms has a Chi-square distribution with two degrees of freedom. If the residual is small, then $Dist_{Maha}{}^2$ will be small. The Chi-square probability

$$P(\chi^2, 2) = \int_{\chi^2}^{\infty} P(\bar{\chi}^2, 2) d\bar{\chi}^2$$

is used to decide the proximity of the two alarms. The Chi-square probability is the probability that the square of the Mahalanobis distance between two sensor alarms is greater than or equal to χ^2. The gating distance χ^2 is chosen according to a pre-determined level of confidence between zero and one. For example, a confidence value of 95% (corresponding to a Chi-square probability of 0.05) dictates the use of a Chi-square distance of 5.99 when there are two degrees of freedom. In other words, 95% of the fused alarms will be correctly associated.

Both gating strategies are followed by the use of a heuristic-based confidence value updating method and a Kalman-based positional update method in the scanning sensor fusion module. The Kalman-based positional update will be discussed next, followed by the discussion of two variations to the heuristic-based confidence value updating method.

3.3 Kalman-based Positional Update

The advantage of Kalman-based positional update is that it utilizes sensor variances in determining the positional weights that are attached to the sensors. For example, if the MMD positional error variance is four times smaller than that of the GPR in one direction, then the MMD is more accurate than the GPR by a factor of two in that direction. A detailed mathematical description of the Kalman position update method is presented below.

When a spatial correspondence class contains only two members, the position vectors of two sensor alarms, a_1 and a_2, can be combined by weighting them with the covariance matrices, $\mathbf{P}_1$ and $\mathbf{P}_2$, and the cross-covariance matrices, $\mathbf{P}_{12}$ and $\mathbf{P}_{21}$, of their sensor localization errors. If the positional errors of one scanning sensor are independent of the positional errors of another scanning sensor (which is assumed to be the case here), then $\mathbf{P}_{12}$ and $\mathbf{P}_{21}$ are zero. The positional update for the fused alarm a_{12} is simply:

$$a_{12} = \mathbf{P}_2(\mathbf{P}_1 + \mathbf{P}_2)^{-1} a_1 + \mathbf{P}_1(\mathbf{P}_1 + \mathbf{P}_2)^{-1} a_2.$$

The corresponding covariance of the fused alarm M_{12} is:

$$\mathbf{M}_{12} = \mathbf{P}_1(\mathbf{P}_1 + \mathbf{P}_2)^{-1} \mathbf{P}_2.$$

Since a correspondence class can contain more than two sensor alarms, the general expression for the positional update is:

$$a_{1,2,\ldots,N} = \ldots$$
$$\mathbf{P}_2 \ldots \mathbf{P}_N \left(\sum_{i=1}^{N} \mathbf{P}_1 \ldots \mathbf{P}_{i-1} \mathbf{P}_{i+1} \ldots \mathbf{P}_N \right)^{-1} a_1 + \ldots$$
$$+ \mathbf{P}_1 \ldots \mathbf{P}_{N-1} \left(\sum_{i=1}^{N} \mathbf{P}_1 \ldots \mathbf{P}_{i-1} \mathbf{P}_{i+1} \ldots \mathbf{P}_N \right)^{-1} a_N.$$

3.4 Confidence Value Updating Scheme

The confidence value updating method employs two linear ramp mapping functions to map confidence values to a value between zero and one, one for spatial correspondence classes containing single alarms and the other for spatial correspondence classes containing multiple alarms. Single alarm

classes are de-emphasized with confidence values within a prescribed range compared to multiple alarm classes. As shown in Figures 3 and 4 for the GPR, the weight applied to single alarm classes (represented by the slopes of the two linear ramp mapping functions) is reduced somewhat in order to reflect the fact that there is no other supporting evidence that there is indeed a mine at the reported location.

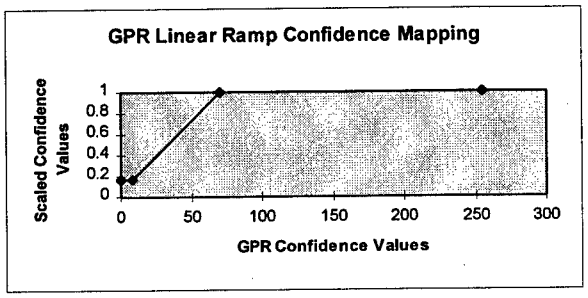


Figure 3. Linear Ramp Mapping Function for Multiple Alarm Clusters

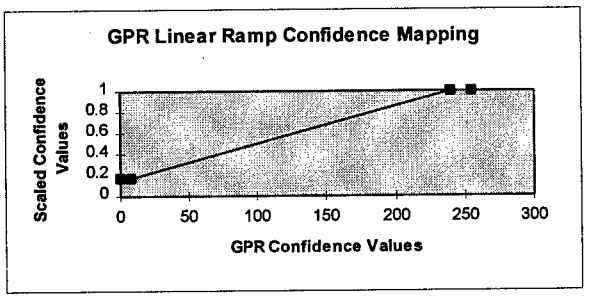


Figure 4. Linear Ramp Mapping Function for Single Alarm Clusters

The effect of applying the two sets of linear ramps to correspondence classes of multiple and single alarms on the Pd and the FAR depends on the employed gating strategy. The form of the linear ramps remains unchanged, but the thresholds and slopes will change. For example, if the gating strategy that is employed has a tendency to group alarms that should really not be included (i.e. a large gate), both the Pd and the FAR will tend to increase. In order to maintain the FAR low, single-alarm classes can be subjected to more severe thresholding than would normally be the case. The actual values for the thresholds and slopes are determined via a confidence value analysis on data collected before each mine detection mission.

IV. Results

The two gating strategies, together with the heuristic-based confidence update scheme and the positional update method, have been implemented and tested in Matlab. The data used in the analysis was collected at the Ground Standoff Mine Detection System (GSTAMIDS) Advanced Technology Demonstration (ATD) trials sponsored by the U.S. Army CECOM.

The GSTAMIDS ATD trials were conducted at the Aberdeen Proving Grounds (APG) and Socorro trial site. The APG site provided a warm, humid test environment, while the Socorro site was hot and dry. A number of AT mine targets were used for the ATD including metal mines (M15, M15I, TM46, TM62M, and TM62MI), low metal mines (M19, M19I, TM62P, and TMA4), and non-metal surrogates (EM12). These mine targets were surface laid and buried at depths of 1 to 4 inches. Approximately 40% of the mines were surface laid.

The Pd and FAR results of the two data fusion algorithms, each employing a different gating strategy, are tabulated below in Table 1 for 9 test runs at the APG trial site. The Pd and FAR results for the IR imager and the MMD are tabulated in Table 2. The GPR Pd and FAR results are not included for presentation due to the fact that its performance is proprietary to the manufacturer. The corresponding Pd and FAR for the simple "OR" operation on all the three scanning sensors are tabulated in Table 3.

Table 1 Results for Two Data Fusion Algorithms

	Data Fusion (Error Ellipse)		Data Fusion (Chi-square)	
Run	Pd (%)	FAR (/m²)	Pd (%)	FAR (/m²)
1	96.55	0.039	100.00	0.058
2	93.10	0.046	96.55	0.086
3	91.43	0.027	91.43	0.066
4	94.29	0.022	94.29	0.062
5	88.57	0.020	91.43	0.054
6	85.71	0.025	85.71	0.061
7	94.59	0.043	100.00	0.096
8	96.55	0.026	100.00	0.054
9	86.21	0.030	96.55	0.063

Table 2 Results for Individual Scanning Sensors

	MMD		IR	
Run	Pd (%)	FAR (/m²)	Pd (%)	FAR (/m²)
1	62.07	0.116	100	0.056
2	65.52	0.151	9655	0.063
3	60	0.111	77.14	0.078
4	60	0.108	85.71	0.059
5	51.43	0.096	77.14	0.066
6	54.29	0.004	62.86	0.073
7	51.35	0.176	81.08	0.077
8	58.62	0.110	100	0.014
9	58.62	0.091	79.31	0.061

Table 3 Results After an "OR" Operation

	"OR" Operation on IR, MMD, and GPR	
Run	Pd (%)	FAR (/m²)
1	100	0.298
2	100	0.347
3	100	0.296
4	97.14	0.297
5	100	0.262
6	94.29	0.272
7	100	0.378
8	100	0.314
9	96.55	0.347

The data fusion Pd and FAR results are then compared to the Pd and FAR results obtained with the IR imager and the MMD as well as the Pd and FAR results obtained with the use of a simple "OR" operation on the three scanning sensors. The comparison results are presented in Figure 5 below.

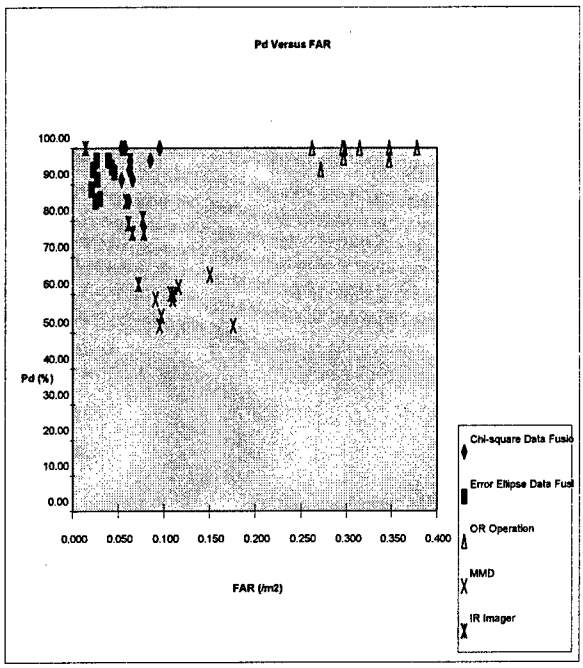


Figure 5. Pd Versus FAR Results

V. Discussion of Results

During the course of this investigation, two different gating strategies were examined under the spatial correspondence process. The first gating strategy is based on the premise that an alarm position is equally likely to be located anywhere within an error ellipse surrounding it. The second one is based on thresholding the "tail" in the distribution of the covariance-weighted residual of two sensor alarms to arrive at spatial correspondence classes. The "tail" in the distribution of the covariance-weighted residual of two sensor alarms gives an indication of the percentage of sensor alarms that are incorrectly associated. Both gating strategies are followed by the use of a heuristic-based confidence value updating method and a Kalman-based positional

update method in the scanning sensor fusion module.

The investigation results embody the following observations:

- The use of data fusion produces superior Pd and FAR performance (as indicated by the diamond and square scatter plots residing closer to the top left corner) compared to the individual scanning sensors.
- The use of data fusion produces a much better FAR performance compared to the use of multiple scanning sensors without data fusion (represented by the simple "OR" operation). The use of multiple scanning sensors increases the Pd but the resulting FAR is also extremely high, as indicated by the triangular scatter plot.
- The error ellipse-based gating technique in the data fusion algorithm is superior to the chi-square gating technique. The error ellipse-based gating achieves the low FAR by accepting a slight reduction in the Pd. The percentage reduction in FAR is far greater than the percentage reduction in Pd.
- The heuristic-based confidence value mapping scheme based on the results from a confidence value analysis provides good results. This is not surprising since a heuristic-based confidence value mapping scheme does owe its success to the availability of a good understanding of the confidence value behavior for a specific operational environment.

VI. Conclusions

It can be concluded from our study that the use of data fusion, in conjunction with multiple sensors, provides a viable solution to the mine detection problem. It is important to note that the achievable FAR at this point is prior to the confirmation by the TNA sensor. Therefore, it is expected that the overall system FAR will be lower than the values tabulated in Table 1. It is also important to note that data fusion tends to reduce the overall system Pd (as is indicated by the higher Pd achieved by the "OR" operation over that achieved by the two data fusion algorithms in Figure 4). Therefore, individual sensor detection performance is critical in simultaneously achieving high Pd and low FAR for all system operating conditions.

References

[1] J. McFee, Y. Das, A. Carruthers, S. Murray, P. Gallagher, and G. Briosi, "CRAD Countermine R&D Study - Final Report," Suffield Special Publication No. 174, March 1994.

[2] B. Cain and T. Meidinger, "The Improved Landmine Detection System," Proceeding of the EUREL International Conference on the Detection of Abandoned Landmines, October 1996.

[3] Proceedings of the SPIE Conference on Detection and Remediation Technologies for Mines and Minelike Targets II, Vol. 3079, April 1997.

[4] D. Hall, *Mathematical Techniques in Multisensor Data Fusion*, Artech House, Boston, 1992.

[5] Y. Bar-Shalom and X. Li, *Estimation and Tracking: Principles, Techniques, and Software*, Artech House, Boston, 1993.

Session TB2
Image Fusion I
Chair: Elisa Shahbazian
Lockheed-Martin, Canada

Wavelets for Image Fusion

Padmavathi Ramanathan and Satyanarayan S. Rao
Department of Electrical and Computer Engineering
Villanova University
Villanova, PA, U.S.A.

Abstract - ***Image fusion aims at the integration of complementary information from multisensor images, such that the resulting image is suitable for further processing. Multisensor images may be of different resolutions. Wavelets with their multiresolution property, have proven to be effective in the blending of the coarse features and finer resolution details of these images to produce a good fused image. The performance of two wavelet- based methods for image fusion is studied. One is the maximum-frequency rule and the other is a rule based on the standard deviation of the image coefficients. Multifocal images and panchromatic-multispectral images are used as the test images. For both the image sets, the proposed standard deviation-based rule performs better than the maximum-frequency rule. The resulting fused images have good spatial resolution and preserve the salient features of the source images.***

Key words: image fusion, wavelets, multiresolution

1 Introduction

Information from different sensors relative to the same scene can be used to obtain better knowledge of the scene than the use of a single sensor's information. Image fusion falls into the category of pixel-level sensor fusion. Multisensor image fusion finds many applications in the fields of remote sensing, medical imaging, machine vision and Department of Defence (DoD). For land-use classification, for example, the Thematic Mapper (TM) images of LANDSAT and SAR images can be fused to obtain a better picture of the area under consideration. In military applications, image fusion is generally applied for object or target recognition. Data can be provided by radar, optical, infrared and other sensors.

An important pre-requisite for image fusion is that the images to be fused must be registered. This means that the pixels in the images to be fused must precisely coincide to the same points of the image they represent. We consider registered images as inputs to the fusion process. The basic idea is to perform a wavelet packet decomposition of the source images and use the best tree decomposition, to combine the coefficients according to some fusion rule. The most often used fusion rule is the maximum-frequency rule which picks the coefficient whose absolute value is the greatest. Another method uses an energy measure to choose between the coefficients. The standard deviation of a 3x3 neighborhood centered around a pixel serves as an energy measure associated with that pixel. The fused image is obtained from the composite wavelet tree (formed by selecting the coefficients) by the reconstruction process.

The results of fusing multifocal images and panchromatic-multispectral images using the two fusion rules mentioned above are presented in this paper. Two source images are used in each case, but the methods are generally applicable to multiple source images. Multifocal images arise when the distortion in the images is due to parts of the image being

out of focus. These types of images are encountered in digital camera applications. The panchromatic-multispectral images are satellite images of a region at different resolutions. While the panchromatic image has good spatial resolution, it has very little spectral information. The multispectral image, on the other hand, has very good spectral resolution but poor spatial resolution. The composite or fused images in both the examples have good overall picture clarity and preserve the features at various resolutions.

2 Wavelets and wavelet packets

Multiresolution analysis of images provides useful information for computer vision and image processing applications. The multiresolution formulation is designed to represent signals where a single event is decomposed into finer and finer detail. In the context of image analysis, multiresolution decomposition gives a coarse approximation image and three detail images *viz.,* horizontal, vertical and diagonal detail images. Thus the features dominant at various resolutions can be studied, which is not possible if conventional Fourier analysis is used. The multiresolution methods most commonly used for image fusion are the Laplacian Pyramid transform and the Discrete Wavelet Transform. Most recently, the Discrete Wavelet Frame (an overcomplete shift-invariant type of DWT) was also used for image fusion in [1].

2.1 The Discrete Wavelet Transform

The concept of resolution defines a scaling function, and the wavelet function is derived from it. A set of scaling functions is defined in terms of integer translates of the basic scaling function [2] by

$$\phi_k(t) = \phi(t-k) \tag{1}$$

$k \in Z$ and $\phi \in L^2$. Z and R denote the sets of integers and real numbers, respectively. $L^2(R)$ denotes the vector space of square-integrable one-dimensional functions. The subspace of $L^2(R)$ spanned by these functions is defined as

$$\nu_0 = \overline{Span\{\phi_k(t)\}}_k \tag{2}$$

for all integers k from minus infinity to infinity. The over-bar denotes closure. This means that

$$f(t) = \sum_k a_k \phi_k(t) \tag{3}$$

for any $f(t) \in \nu_0$. A two-dimensional family of functions is generated from the basic scaling function by scaling and translation by

$$\phi_{j,k}(t) = 2^{j/2}\phi(2^j t - k) \tag{4}$$

whose span over k is

$$\nu_j = \overline{Span\{\phi_k(2^j t)\}}_k = \overline{Span\{\phi_{j,k}(t)\}}_k \tag{5}$$

for all integers $k \in Z$. This means that if $f(t) \in \nu_j$, then it can be expressed as

$$f(t) = \sum_k a_k \phi(2^j t + k). \tag{6}$$

For $j > 0$, the span can be larger since $\phi_{j,k}(t)$ is narrower and is translated in smaller steps. This can represent finer detail. For $j < 0$, $\phi_{j,k}(t)$ is wider and is translated in larger steps. So these wider scaling functions can represent only coarse information, and the space they span is smaller. A change of scale thus implies a change in resolution.

The important features of a signal can better be described or parameterized by defining a slightly different set of functions $\psi_{j,k}(t)$ that span the differences between the spaces spanned by the various scales of the scaling function. These functions are *wavelets*. Wavelets are basis functions of effectively limited duration and are well-known for their localization properties. The scaling functions and wavelets are generally required to be orthogonal. This is because orthogonal functions allow simpler calculation of expansion coefficients and follow Parseval's theorem that allows a partitioning of the signal energy in

the wavelet transform domain. The orthogonal complement of ν_j in ν_{j+1} is defined as W_j. This means that all members of ν_j are orthogonal to all members of W_j. This requires

$$\langle \phi_{j,k}(t), \psi_{j,l}(t) \rangle = \int \phi_{j,k}(t)\psi_{j,l}(t)dt = 0 \quad (7)$$

for all appropriate $j, k, l \in Z$. The relationship of the various subspaces starting at ν_0 is $\nu_0 \subset \nu_1 \subset \nu_2 \subset ... \subset L^2$. The wavelet spanned subspace W_0 is defined as

$$\nu_1 = \nu_0 \oplus W_0 \quad (8)$$

which extends to

$$\nu_2 = \nu_0 \oplus W_0 \oplus W_1. \quad (9)$$

This can be generalized as

$$L^2 = \nu_0 \oplus W_0 \oplus W_1 \oplus ... \quad (10)$$

when ν_0 is the initial space spanned by the scaling function $\phi(t-k)$. The wavelets reside in the space spanned by the next narrower scaling function, $W_0 \subset \nu_1$, they can be represented by a weighted sum of shifted scaling function $\phi(2t)$ as

$$\psi(t) = \sum_n h_1(n)\sqrt{2}\phi(2t-n) \quad (11)$$

where $n \in Z$ for some set of coefficients $h_1(n)$. This function gives the prototype or mother wavelet $\psi(t)$ for a class of expansion functions of the form

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k) \quad (12)$$

where 2^j is the scaling of t, $2^{-j}k$ is the translation in t, and $2^{j/2}$ maintains the L^2 norm of the wavelet at different scales. The set of functions $\phi_k(t)$ and $\psi_{j,k}(t)$ span all of $L^2(R)$. Any function $g(t) \in L^2(R)$ could be written as

$$g(t) = \sum_{k=-\infty}^{\infty} c(k)\phi_k(t) + \sum_{j=0}^{\infty}\sum_{k=-\infty}^{\infty} d(j,k)\psi_{j,k}(t) \quad (13)$$

as a series expansion in terms of the scaling function and wavelets. The first summation in the above equation gives a function that is a low resolution or coarse approximation of $g(t)$. For each increasing index j in the second summation, a higher or finer resolution function is added, which adds increasing detail.

2.2 Wavelet Packets

The wavelet packet method is a generalization of wavelet decomposition that offers a wide range for signal analysis [3]. In wavelet packet analysis, the details as well as the approximations are split to yield 2^n different ways to represent the signal where n is the decomposition level. A single decomposition using wavelet packets generates a large number of bases which offer a more complex and flexible analysis. An entropy-based criterion is used to select the most suitable decomposition of a signal or image. This implies that at each node of the decomposition tree, the information to be gained by performing each split is quantized. The leaves of every connected binary sub-tree of the wavelet packet tree correspond to an orthogonal basis of the initial space. For a finite energy signal, any wavelet packet basis will provide exact reconstruction and offer a specific way of coding the signal, using information allocation in frequency scale subbands.

3 Image fusion using Wavelet Packet Decomposition

The general method for image merger using the Wavelet Packet decomposition is as follows:

1. The wavelet packet decomposition of the source images is performed, having chosen the wavelet basis and the depth or level of decomposition.

2. The best trees for the images are found on the basis of some entropy-based criterion (in our case Shannon criterion) and the tree which has the greatest number of leaf nodes is chosen as the composite tree structure to be populated.

3. The wavelet packet coefficients are then selected from the source images according to a fusion rule to populate the tree. The rules used in this paper are:
   Maximum frequency rule: selects the coefficient with the highest absolute value.

The high values indicate salient features like edges and are thus incorporated into the fused image. The rule is applied at all the resolutions under consideration.
Standard-deviation rule: calculates an activity or energy measure associated with a pixel. A decision map is created, which indicates the source image from which the coefficient has to be selected.

4. The wavelet packet reconstruction of this synthetic or composite tree gives the required fused image.

Some variations in the above procedure may be necessary when dealing with special images like color images or Synthetic Aperture Radar (SAR) images.

3.1 Multifocus image fusion

Multifocus image fusion has been considered in [1]. These images arise in situations where only a portion of the scene is in focus while the rest is blurred. Camera position, quality and motion may generate such images which call for correction.

Two multifocus images are considered, one in which the left half (pepsi can) is focused and another in which the right half (the chart) is focused. To get a fused image, the wavelet packet decomposition scheme is used to select the coefficients based on a fusion rule. The resulting image is focused in all regions. A performance measure, ρ, is defined as the standard-deviation of the difference between the fused image and the ideal fusion result [4],

$$\rho = \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{N}[I_{pr}(i,j) - I_{fd}(i,j)]^2}{N^2}} \quad (14)$$

where I_{pr} is the ideal fusion result, created by manual cut and paste and I_{fd} is the fused image. However, this performance measure only serves as a criterion for comparing the performance of various fusion rules and is generally not applicable to many of the real multisensor fusion problems as it is not possible to obtain the ideal fusion result manually.

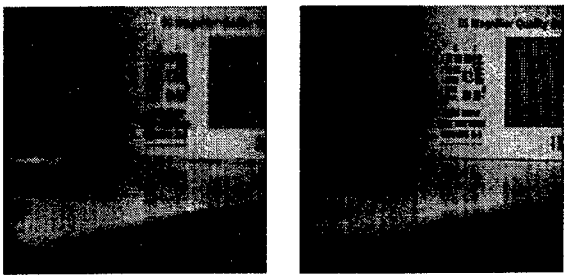
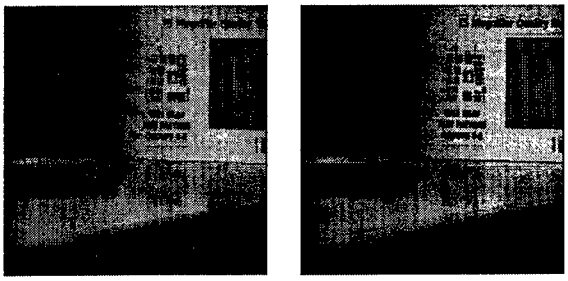


Figure 1: Source images with different focus regions



(a) Maximum frequency rule (b) Standard deviation rule

Figure 2: Fused images

The maximum-frequency rule gives a fused image with good overall focus but the letters on the chart are not quite clear. The error ρ, is 0.0402. The standard-deviation rule gives a better fused image in terms of overall focus. The associated error, ρ, is 0.0343, which is less than that of the maximum-frequency rule. The raw source images and the fused images are shown in Figure 1 and Figure 2 respectively.

3.2 Panchromatic-Multispectral Image Fusion

The IRS-1C satellite provides high spatial resolution panchromatic (5m) images and multispectral images (25m) with poor spatial resolution. The merged image should ideally have good spatial resolution and color information from the multispectral image. This gives a good picture of the scene under consideration. In [5], a Local Mean Variance Matching (LMVM) algorithm is used for the fusion

process, which yields a very good result. The result of this process is used as the ideal fused image for comparison purposes.

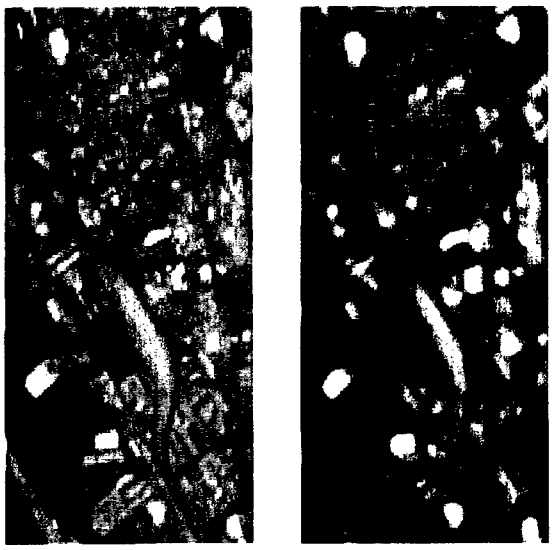
The source images require the additional process of histogram matching before the Wavelet Packet decomposition. The histogram matching of the high resolution channel (panchromatic image) to each of the three low resolution channels - R, G, B of the multispectral image is performed to adjust radiometry and improve the initial correlation between the images [5]. Then the Wavelet Packet decomposition and fusion processes are applied to each of the three channels. The detail coefficients are chosen from the high resolution image matched to the low resolution channels and the approximation coefficients are chosen from the low resolution channels according to the fusion rule. The composite color image with the required spatial details is formed from these three images.

The maximum-frequency fusion rule gives a good reconstruction, with some blurring. The errors in the R, G, B channels ρ_r, ρ_g, ρ_b, are 0.0669, 0.0481 and 0.0572 respectively for the maximum-frequency rule and 0.1063, 0.0524 and 0.0618 respectively for the standard-deviation fusion rule. The latter rule results in the details and brightness being enhanced, while the error in red increases and is visible as a distortion in the red patch of the fused image. Figure 3 and Figure 4 show the source images and fused images respectively.

4 Conclusions

In the study of the fusion of the two image sets (multifocal and panchromatic-multispectral images), it was found that the standard-deviation rule preserved the details well when compared to the maximum-frequency fusion rule. In the panchromatic-multispectral image fusion, the error in all the three channels was found to be greater for the standard-deviation fusion rule but is preferred over the maximum-frequency rule when a little distortion in the color can be tolerated. The choice

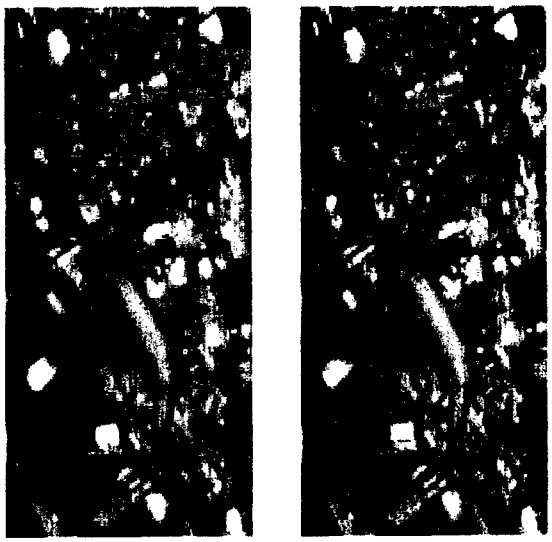
Figure 3: Source images



(a) Panchromatic image (good spatial resolution)

(b) Multispectral image (good spectral resolution)

Figure 4: Fused images



(a) Maximum frequency rule

(b) Standard deviation rule

of the fusion rule used depends on the application. In a generic framework for image fusion [1], window-based, region-based activity levels were used for fusion of multifocal images alongwith a consistency verification scheme. Similar methods can be used for panchromatic-multispectral image fusion to improve the color information in the fused image. The Daubechies family of wavelets was used in this paper for a two-level wavelet packet decomposition. Other wavelet bases could be used. The source images in this paper were considered to be registered. The effects of slight misregistration on the fusion process is another area of active research.

Acknowledgements
We thank Laboratory SURFACES, University of Liège, Belgium, for providing the satellite images for this work.

References

[1] Zhong Zhang and Rick S. Blum. Image fusion for a digital camera application. In *Proceedings of the 32nd Asilomar Conference on Signals, Systems and Computers, (Pacific grove, CA, Nov 1-4, 1998)*, pages 603–607, 1998.

[2] C. Sidney Burrus, Ramesh A. Gopinath, and Haitao Guo. *Introduction to Wavelets and Wavelet Transforms: A primer.* Prentice Hall, Upper Saddle River, New Jersey, USA, 1998.

[3] Michel Misiti, Yves Misiti, Georges Oppenheim, and Jean-Michel Poggi. *Wavelet Toolbox.* The MathWorks,Inc., Natick, Mass. USA, 1996.

[4] H. Li, B.S. Manjunath, and S.K. Mitra. Multisensor image fusion using the wavelet transform. *Graphical Models and Image Processing*, 57(3), May 1995.

[5] S. de Béthune, F. Muller, and M. Binard. Adaptive intensity matching filters: A new tool for multi-resolution data fusion. In *AGARD Conference Proceedings 595, Multi-Sensor Systems and Data Fusion for Telecommunications, Remote Sensing and Radar, (Lisbon, Portugal, Sep 29 - Oct 2, 1997)*, 1998.

[6] Zhong Zhang and Rick S. Blum. A categorization of multiscale decomposition-based image fusion schemes with a performance study for a digital camera application. *Proceedings of the IEEE*, June 1999.

[7] Mario Constantini, Alfonso Farina, and Francesco Zirilli. The fusion of different resolution sar images. *Proceedings of the IEEE*, 1997.

[8] David A. Yocky. Image merging and data fusion by means of the discrete two-dimensional wavelet transform. *Journal of the Optical Society of America A*, 12(9), September 1995.

[9] David L. Hall and James Llinas. An introduction to multisensor data fusion. In *Proceedings of the IEEE*, 1997.

[10] Laurie J. Chipman, Timothy M. Orr, and Lewis N. Graham. Wavelets and image fusion. *SPIE*, 2569, July 1995.

[11] Alexander Toet. Multiscale contrast enhancement with applications to image fusion. *Optical Engineering*, 31(5), May 1992.

[12] P.J. Burt and R.J. Kolczynski. Enhanced image capture through fusion. In *Proceedings of the 4th Intl. Conference on Computer Vision, (Berlin, Germany)*, pages 173–182, May 1993.

[13] Z. Zhong and R.S. Blum. A region-based image fusion scheme for concealed weapon detection. In *Proceedings of the 31st Annual Conference on Information Sciences and Systems*, pages 168–173, March 1997.

Remotely sensed images fusion for linear planimetric features extraction

Luc PIGEON [1,2], Bassel SOLAIMAN [1], Keith P. B. THOMSON [2], Thierry TOUTIN [3]

[1] **École Nationale Supérieure des Télécommunications de Bretagne,** Dépt. I.T.I, , B.P.832, 29285 Brest, France;

[2] **Centre de Recherche en Géomatique,** Pavillon Casault, Université Laval (Québec), G1K 7P4, Canada;

[3] **Canada Centre for Remote Sensing,** 588 Booth Street #439, Ottawa (Ontario), K1A 0Y7, Canada.

{Luc.Pigeon, Basel.Solaiman}@enst-bretagne.fr
{Luc.Pigeon, Keith.Thomson}@scg.ulaval.ca
Thierry.Toutin@ccrs.nrcan.gc.ca

ABSTRACT

This paper presents an overview of a satellite images fusion system for mapping applications. The main goal of this system is to dilate the map's feature extraction bottleneck by semi-automating this process. This study deals with the linear planimetric features (LPF) extraction for the 1:50 000 topographical map creation. These features include roads, railroads, energy transmission lines and some types of rivers. Actually, the only data source used for their extraction is aerial black and white photographs. The objective here is to fusion multi-sources and multi-types information. This information ranges from satellite images (visible and radar) to domain based models and of expert's modeled knowledge, strategies and rules. The whole system includes an operator who will give inputs and validates the results along the whole task.

Key Words: *remote-sensing, satellite images fusion, semi-automatic mapping systems, expert systems*

1. INTRODUCTION

Since childhood, everybody learns to position himself in his environment. As some seem to possess an integrated inertial positioning system in their brain, the others needs to consult maps on regular basis. Maps exist since the beginning of humanity. Their development depends on the improvement in data acquisition and processing technologies. Data acquisition traditionally reserved to land surveyors extend to aerial photographs and now to digital images. Maps edition develops chronologically from unique hand-made maps to sophisticated software digital maps. If land surveyor's data can be edited directly, it is not the case of aerial photograph data. The land surveyor always preprocesses the first as the second is given in a raw format. Thus, a new task appears in the map creation process: map's features extraction from visual data. Experts assigned to this task are known as photo-interpreters. Their task is to extract map's features from aerial black and white (B&W) photographs used in stereoscopic (3D) models. For the 1:50000 official topographic map production in Canada, their job consists on extracting from 1:60000 aerial B&W photographs the main features such as the hydrographic network, contour lines, roads, railroads, energy transmission lines, vegetation and buildings. In Canada, topographical maps are updated each three years in urban areas and each five to eight years in rural areas. However, some areas such as forestry companies clear cuts zones are mapped each year for control purposes.

Given that photo-interpreters training process takes up to ten years in some case [1], a problem occurs to get enough experts for the amount of incoming data and required maps (particularly in Canada). That's one of the reasons why official mapping is still using as a unique image data source aerial B&W photographs. They don't have the human resources to process satellite's remotely sensed data. Hence, this rich data sources are lost, in the mapping area. The second reason is the high accuracy required for mapping. If this criterion was good few years ago to discard satellite images, it is now less true and will probably be false in a close future with the apparition of high-resolution satellites.

Hence, the bottleneck in mapping is between the data acquisition and the map edition, at the photo-interpreter tasks level. The objective of the research project presented in this paper is to dilate this bottleneck by semi-automating the linear planimetric features (LPF) extraction and classification tasks . To reach this objective, photo-interpreters capacities and

the rich sources of information offered by satellite remotely sensed data should be integrated in a unique system.

LPFs include roads, railroads, energy transmission lines and rivers (only rivers geometrically looking like roads, as the main objective is road extraction). The choice of the LPFs is based on their strategic importance and their complexity. Everybody from the military to John Doe uses road maps. This feature is the most changing cartographic element. New roads are constructed, existing roads are modified relatively often if compare with the other map's features. The other LPFs are also strategically of first importance. On an other level, LPFs extraction represents a good challenge for their complexity. These features can require radiometric, geometric and topological information for their extraction. Some 3D geometric information is available only by depth perception. No algorithms today allow getting such information.

Thus, a human operator have to be include through the LPF extraction process. Moreover, the human-machine cooperation should be optimized (i.e. the most interesting as possible for the human and with the highest level of performance).

The system under development aims at the integration of multi-sources and multi-types (visible and radar) images, domain models and expert's knowledge. These last interacting with the human operator. The output of the proposed system is considered as a 3D symbolic map. Each voxel of this map will contain (X,Y,Z) terrain coordinates, class membership, accuracy information, etc. On this map, LPFs will be symbolized by their respective cartographic symbols. Hence, LPFs extraction does not only consist on identifying a road, for example, but also to classify this road as highway, principal or secondary (with respect to the details of the cartographic symbols). Figure 1 shows the diagram of the global system under development.

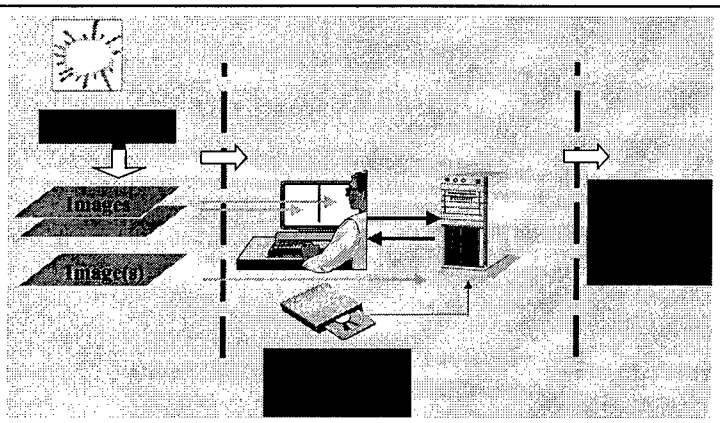


Figure 1: The image inputs, semi-automatic processing unit and the symbolic map output

II. METHODOLOGY

Photo-interpreter task is positioned between the raw image data and the extracted and classified edited map data. This task can be resumed in three steps: structuring the data, identifying and classifying the feature. Previous works on LPFs extraction are almost exclusively concentrate on structuring the data.

II.a. Primitive extraction

Raw image data consists of a two-dimensional (2D) pixel array. The primitive extraction task (to structure the data) consists on grouping the pixels into basic structures. For road detection, these structures are linear segments that are extracted following one or many criteria. These criteria can be single pixel's radiometry, radiometry variations, geometry of the structure, etc. Unfortunately, many of these criteria vary from one zone to another. Hence, many road detection algorithms are specific to a particular road's type, context (urban, rural) and image's resolution and type. Some methods detect roads in a rural context using visible images [2][3]. Few methods are enough general to detect roads in both urban and rural contexts and on many resolution images [4][5][6]. The two first are developed on 10-meters resolution SPOT Panchromatic visible images. The third is developed for radar type images. As the first is relatively fast and easy to use in order to detect a road network, the second needs much *a priori* knowledge. The third is very time-consuming and not usable actually for real-time applications. Few studies are conducted concerning others LPFs primitives extraction [7][8][9] and no algorithms seem specifically developed for this purpose.

II.b Features identification

As previously mentioned, no studies seem to have been conducted in order to identify railroads, energy transmission lines and road looking rivers. The only studies on this subject are primitives filtering methods in roads primitive extractions. For example, [6] extracts in

a first time road's looking primitive and in a second time removes primitives which seem to be artifacts. This subject is conducted in almost all studies concerning road detection.

II.c Features classification

A research field that seems to be neglected is the features classification. If a road primitive is extracted, how to classify it as a highway or a principal road? Moreover, how to assign to it a specific cartographic symbol?

The current research project tackles the challenge of linking the image data to the final symbolic map by structuring, identifying and classifying the LPFs. To reach this result, the following methodology is used:

- fusion of the existing LPFs primitive extraction techniques;
- Combining them with the problem reality, the data acquisition systems and the decision space (symbolic map) models;
- Propagates each known data by domain expert's modeled information such as rules and strategies;
- Finally, integrates all the data sources and information in a unique system including a human operator that gives primitives extraction input (starting point) and who validates the system's results.

III. MODELS

To extract mapping features, knowledge about the features in the reality have to be known. In a second time, knowledge about the data acquisition systems have to be modeled. In fact, each sensor shows a particular facet of the reality. Visible and radar images of the same zone can be completely different. Hence, it is important to know for each reality LPFs, the characteristics of their images. Finally, knowledge about the decision space, here the symbolic map, should be well known. The expected result knowledge will constraint the system's accuracy for positioning of the extracted LPF (centimeters, meters or decimeters) as of the required details (all the road network or only highways and principal roads).

III.a Problem reality model

The problem reality model is principally based on road construction norm books. The current project uses the Québec's road construction standards [10]. In this paper, each road types are presented and detailed as their geometric characteristics. Each road type is presented with its number possible tracks, their width, their minimum and maximum curvatures and slopes, etc. This information is "translated" to an object-oriented UML (Unified Modeling Language [11]) model following the specific LPF extraction from image data source. Thus, information such as road's tracks inclination in curves that are at the centimeters level, were not modeled. Photo-interpreter depth perception accuracy ranges from one to five meters on high-resolution 1:60000 aerial photographs. Figure 2 gives an overview of the reality based LPF hierarchy.

Another source of information related to the reality is the relation between each the LPFs. Topological information will be modeled here following the intersection, disjunction, inclusion, neighborhood and equality operators. For example, a forestry road cannot physically intersect a highway. A railroad cannot physically intersect a river. In counterpart, the railroad can intersect the river at another level, on a bridge for example.

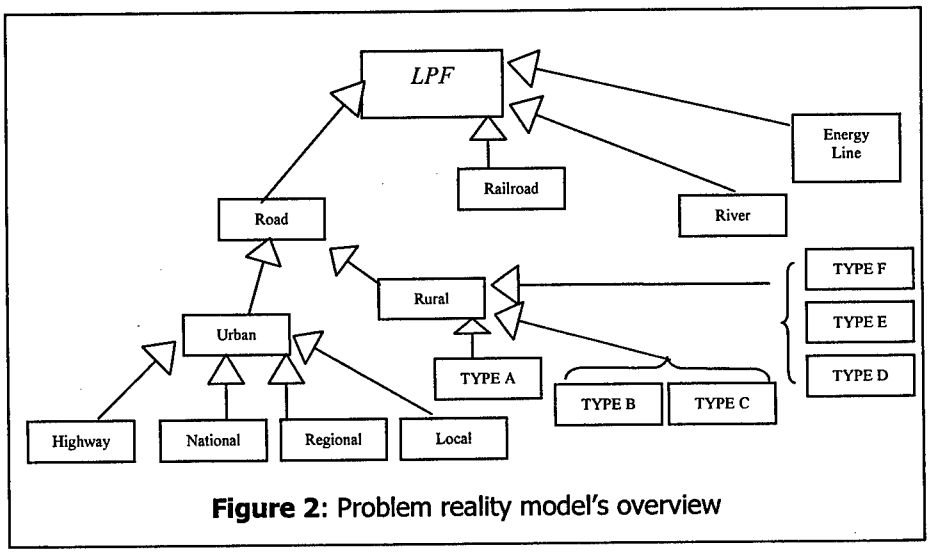


Figure 2: Problem reality model's overview

III.b Image models

Image models share a common part that is the image, the stereoscopic model and the segment (primitive) structures. For each sensor, a specific part uses the main reality structures applied to the specificity of the sensor. For instance, road radiometric characteristics will be described differently in visible and radar images. In visible, road pixels appear bright. On the other hand, in radar they appear dark. On another level, structures definition will not be as much specific as in the reality. For example, the identification of a LPF on a satellite image cannot go more specific than "road in a rural zone". The classification of this road as highway or forestry road will use knowledge contained in the reality model or expert's knowledge source. Figure 3 shows an overview of the general image model.

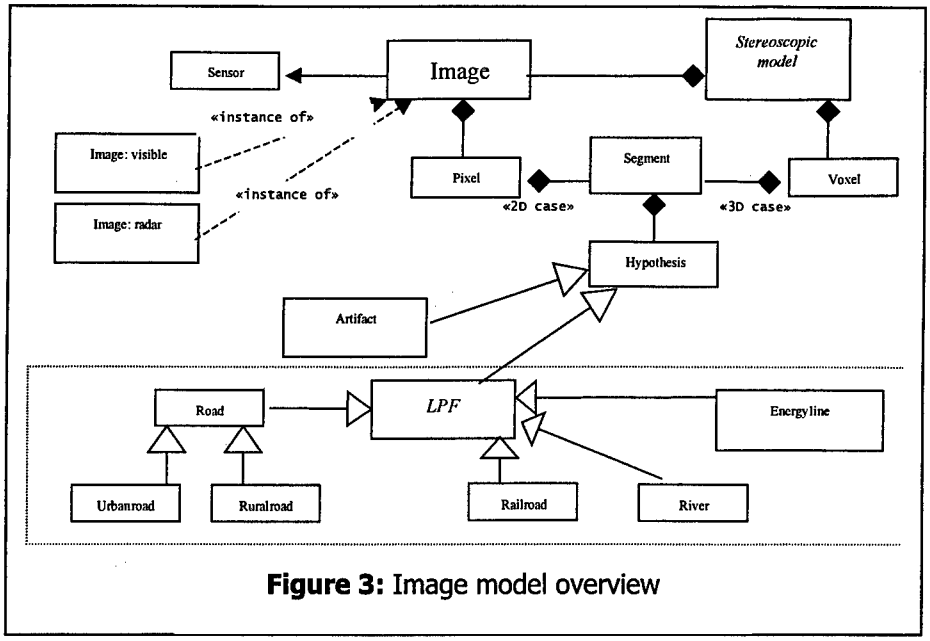


Figure 3: Image model overview

III.c Decision space model

Finally, the decision space model contains the information relative to the symbolic map. It shows the characteristics of the mapping symbols that will be used and what sets of structures they include. Knowledge relative to the map's visualization is also contained here. Figure 4 shows a symbolic map model overview.

However, instead of all the data and information contained in these models, the system is incomplete without tools for propagating the information from the image raw data to the decision space. These tools are here knowledge structures modeled in a procedural way such as rules and strategies.

IV. EXPERT'S KNOWLEDGE

The experts knowledge elicitation task is one of the most delicate task in the development of knowledge based systems. Various techniques can be used to perform this. Here, familiar and unfamiliar cases technique [12] was used. Four experimented photo-interpreters working for between 15 to 25 years at the official mapping service in Canada were interviewed. As the resulting symbolic map will be complete with the 1:50000 topographical map constraints, 1:60000 aerial photographs were first used. Within this familiar case, experts explain their methods and tricks to extract the LPFs from the stereoscopic model. They explain how they discriminate the different LPFs from the others, and

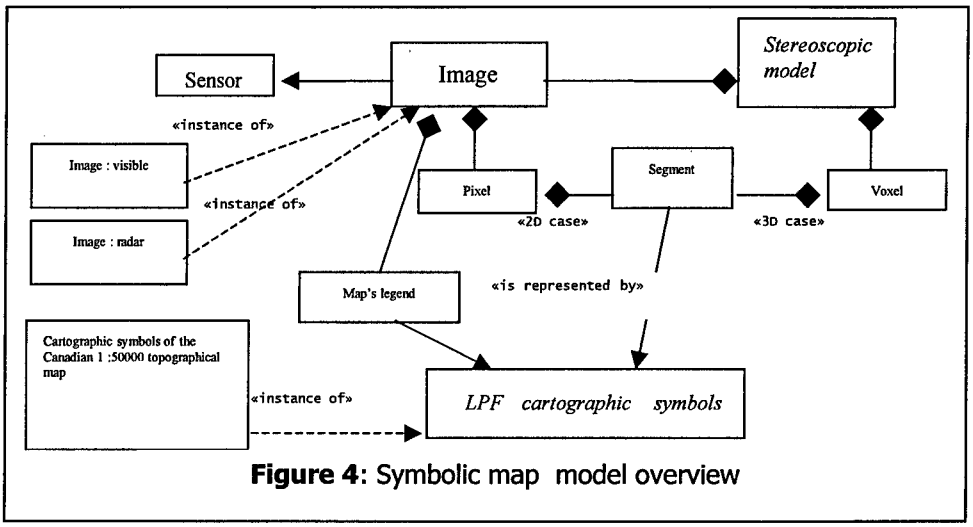


Figure 4: Symbolic map model overview

how they classified them. Moreover, they explain the entire topographical map features extraction task.

In a second time, satellite images were used. At first, IRS-1 5,5 meters-resolution visible image was shown to them. They had to explain the same extraction tasks as with the aerial images. Their reasoning was almost the same and they face no difficulties to extract the LPFs. Next, a Radarsat fine mode 8-meters resolution image was used with the same requirements. With explanations on the data acquisition techniques, they feel able to identify all the map's features again. Their knowledge was compiled and translated in two procedural knowledge structures: strategies and rules.

IV.a. Strategies

A Strategy is defined as "the art of devising or employing plans or stratagems toward a goal". In the context of the problem it can be defined as "the art of devising the 1:50000 topographical map features extraction tasks". It clearly appears that the photo-interpreters begin their features extraction task by extracting the terrain modeling information (hydrographic and contour lines). Next, they extract the human-made LPFs (roads, railroads and energy transmission lines), the vegetation and finally the buildings (all in urban areas and norm's specifics in rural areas). As been a hydrographic feature, rivers will logically be extracted before the other LPFs. Following individual expert's strategies, rivers will be extracted far much or not much before the other LPFs, but always before due to their 3D informational contents. If this information cannot be obtained in 100% cases verifiable (human are not machines), it gives at least a good hint for the identification of the processed LPFs.

At specific LPF classification level, experts use also the strategies. Roads is the LPF which present far the most different class types. These classes range from urban highways to rural local roads. The knowledge elicitation task leads to the modeling of three distinct strategies. In the two first, experts start roads extraction by the most important i.e. highways. When they begin to extract a road, they extract the whole road in one step. Next, they extract national roads and sometimes, secondary roads. At this level, a difference occurs. The experts using the first strategy continue to extract the roads in a "from the most important to the less important" until the end of the road extraction task. The experts using the second strategy continue through the same way as the previous except that they do it only in specific areas. Their goal here is to fractionate their working area in approximately equal zones before to continue through the first strategy in these zones. One of the difficulties of their task is not to forget any feature. Then, it is possible for them to be concentrated on restricted perimeters. This facilitates their job and in the same time increases their performances. The third strategy was not encountered with the experts. However, it seems that it is well used. It consists as super-impose an artificial grid to the process stereo-model. As the second strategy, the operator works in a reduced area. On a feature point of view, the two first strategies can be qualified as hierarchical, and the third as sequential.

Strategies lead to hypotheses about "what the operator is working with" based on the expert's behavior. Another part of knowledge, the rules, try to answer the same question, but based on different information sources.

IV.b. Rules

Rules are defined as "a knowledge structure that relates some known information to other information that can be concluded or inferred". They are presented in the form of "if A then B". A is called the *premise* as B the *conclusion* of the rule. Rules can have a single or multiple premises. However, their structure should be as simple as possible to avoid lost of information. For example, if a rule is composed of three premises like "if A and B and C then D", an error on only one premise can shutdown the whole rule (conjunction). In counterpart, the same rule split up in two or three separate rules decrease this lost of information. It is not always possible to do so, but it should be that as often as possible. Notice that a single premise rule is already complex in the current data fusion context. In fact, a premise such as "if road's pixel then LPF is a road" implies the fusion of pixel's information coming from different data sources, here in each process images. Figure 5 presents a single premise rule in a multiple image fusion context. Figure 6 shows a concrete example of the complexity of multi-sensors observation fusion. On a visible-type image (SPOT Panchromatic), a straight energy transmission line is clearly visible. For the same area, radar-types images (Radarsat and ERS-1) contain a curved feature. As radar is an active system, it is sensible to terrain geometry. The image's zone is in the Canadian Rockies, a very high relief area. It explains why a straight line in the reality can be represented as a curve on an image. This example shows the importance to have good

models of the problem (section II) and the complexity to process multi-types image-based rules.

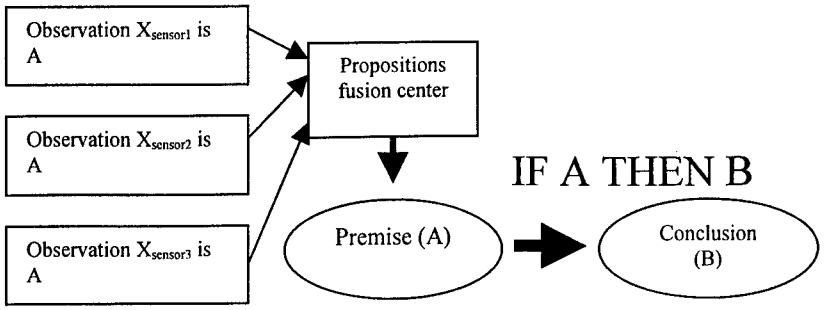


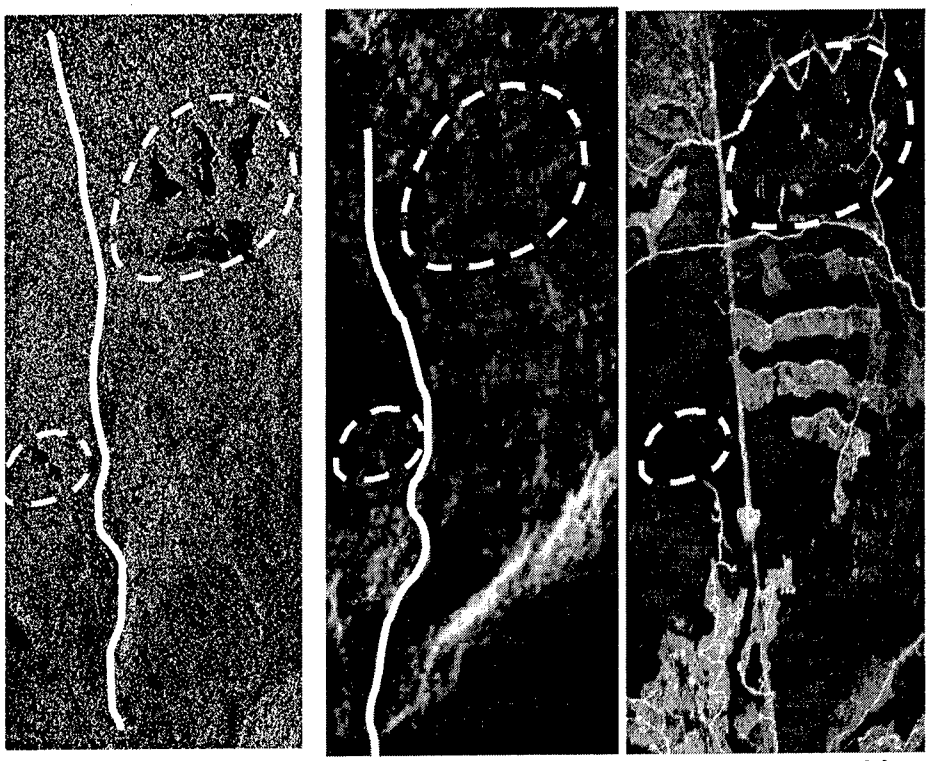
Figure 5: Single premise rule in a multiple image fusion context.

The expert's rules modeled premises fall in the next four categories:

- radiometric;
- geometric (2D and 3D);
- topologic;
- hypotheses (i.e. rules where one or more premises are based on previous hypothesis).

The two first categories relate image data to the decision space trough the reality model. For example, a bright pixel on a visible image primitive and the same pixel dark on the equivalent primitive in a radar image (image model) will lead to a road conclusion (decision space) based on the reality knowledge of this LPF (reality model). The third category relates the decision space (hypothesis) to the reality model (see the section III problem reality model). Finally, the fourth category relates the decision space information to the three previous categories and the reality model. For example, state two hypotheses that seem, after some rules testing, lead to a road conclusion. A fourth category rule can be "if the 2 tested hypotheses lead to a road conclusion (premise 1) and that the angle between these LPFs ranges between 75 and 90 degrees (premise 2 - geometrical) then add confidence in the roads hypothesis (conclusion)".

To translate experts knowledge in terms of rules, confidence factors (CF) were used. It has the advantage of being close to the expert's language. These CF range from –1 to 1, where –1 (resp. 1) means "definitely not" (resp. "definitely yes") as 0 states for "I don't know". For example, if an expert said "if I see curves on the analyzed primitive, then it is definitely not an energy transmission line". The confidence on the premise "I see curves" and on the rule "it is definitely not" is given by the CF. These CF can range either from –1 to 1 or from 0 to 1



Fine Mode Radarsat 8-meters resolution | **ERS-1 30-meters resolution** | **SPOT (Panchromatic) 10-meters resolution**

(images © Canadian Space Agency)

Figure 6: An example of multi-types image data

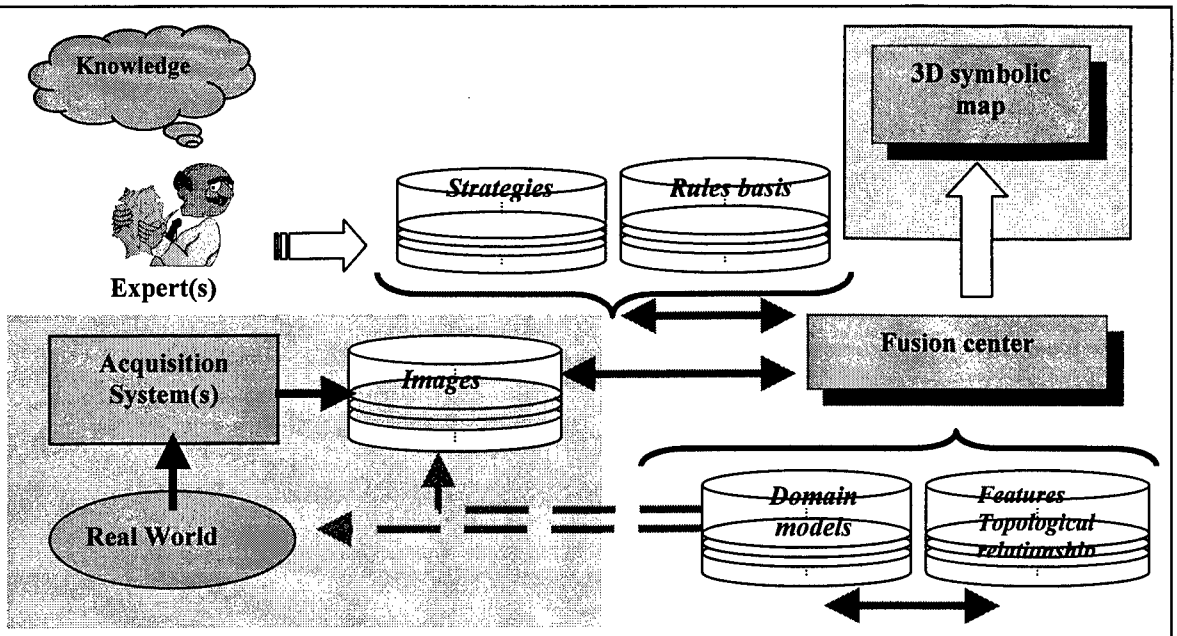


Figure 7 : The system's data sources fusion

following their use. The last form can be more convenient in for use with the Dempster-Shafer theory or the fuzzy set theory. It respects the original idea of ranging from "not to yes" with zero as minimum of knowledge. In fact, the Shannon entropy maximum of uncertainty is located at 0.5 for a binary source of knowledge. Thus, certainty theory or Dempster-Shafer can easily be used for rules combination. However, each situation requires only a specific set of the modeled rules. If all the information leads to two specific results, it is more logical to try to relieve the ambiguity on them instead of searching information for another hypothesis. The rules should thus be structured in trees where the first nodes will be chosen following specific criteria. These criteria will be based on the other part of the modeled knowledge: the strategies. The combination of these two types of knowledge will allow the linking of the analyzed primitives to the decision space. [13][14] present in more details this part of this research project.

V. MODEL AND KNOWLEDGE FUSION

Figure 6 represents diagrammatically the relations between the system's data sources. These sources, presented in the previous sections are the system inputs and output, the different domain models, the rules and strategies basis and the operator (human). All the information is processed in an information fusion center.

A multi-agents architecture system is currently under development for the current research project system's implementation.

VI. CONCLUSION AND FUTURE DIRECTIONS

The main objectives of the presented project are actually reached. The delicate task of expert's knowledge elicitation and modeling is complete. Domain models are also complete. The system under development uses far much knowledge sources than any previous works on road features extraction.

However, the next step is the concrete integration of the systems parts. On a fusion point of view the final system should handled:

- Images fusion
- Propositions fusion
- Premises fusion
- Rules fusion
- Expert's based knowledge fusion
- Hypotheses fusion
- Results fusion.

VII. REFERENCES

[1] R. Hoffman and J. Conway, "Psychological factors in remote sensing: a review of some recent research", *Geocarto International,* Vol. 4, no. 4, 1989.

[2] D. Geman and B. Jedynak, "An active testing model for tracking roads in satellite images", *IEEE Transaction on Pattern Recognition and Machine Analysis*, Vol. 18, no. 1, pp.1-14, 1996.

[3] A. Gruen and H. Li, "Road Extraction from Aerial and Satellite Images by Dynamic Programming", ISPRS *Journal of Photogrammetry and Remote Sensing*, 50(4): 11-20, 1995.

[4] B. Solaiman, R. Fiset, F. Cavayas, "Automatic road extraction using fuzzy mask concepts", *IEEE International Geoscience and Remote Sensing Symposium Proceedings,* pp. 894-896, 1998.

[5] N. Merlet and J. Zerubia, "New prospect in line detection by dynamic programming", *IEEE Transaction on Pattern Recognition and Machine Analysis*, Vol. 18, no. 4, pp. 426-431, 1996.

[6] F. Tupin, H. Maître, J. -F. Mangin, J. M. Nicolas, E. Pechersky, "Detection of linear feature detection in SAR images: application to road network detection", *IEEE Transaction on Geoscience and Remote-Sensing*, Vol. 36, no. 2, pp. 434-453, 1998.

[7] T. Toutin, "SPOT and Landsat stereo fusion for data extraction over mountainous areas", *Photogrammetric Engineering and Remote Sensing*, Vol. 64, no. 2, pp. 109-113, 1998.

[8] T. Toutin, "3D data stereoscopic extraction from mixed VIR and SAR sensors", Proceedings of the 20th Remote Sensing Symposium, Calgary AB, Canada, May 10-13, 1998.

[9] V. Prinet, Z. Zhixin, M.A. Songde, O. Monga, "Communication networks recognition from SPOT images", *IEEE International Geoscience and Remote Sensing Symposium Proceedings,* pp. 2044-2046, 1998.

[10] Québec's Government, Ministère des Transports "Normes de construction des routes", Publications du Québec, 1998.

[11] G. Booch, J. Rumbaugh, Y. Jacobson, "Unified Modeling Language User Guide", *Addison-Wesley,* 1999.

[12] J. Durkin, "Expert Systems: Design and Development", Prentice Hall, 800 p., 1994.

[13] L.Pigeon, B. Moulin, B. Solaiman, T. Toutin, K. P. B. Thomson, "Human-experts features extraction strategies for topographical map production and updating", *IEEE International Geoscience and Remote Sensing Symposium Proceedings,* 1999.

[14] L. Pigeon, B. Solaiman, K. P. B. Thomson, B. Moulin, T. Toutin, "Human-experts rules modeling for linear planimetric features extraction in a remotely sensed images data fusion context", *IEEE International Geoscience and Remote Sensing Symposium Proceedings,* 1999.

Application of Image Fusion to Wireless Image Transmission

Liane C. Ramac and Pramod K. Varshney
EECS Department, Syracuse University
121 Link Hall, Syracuse, NY 13244
Email: lcramac@mailbox.syr.edu, varshney@cat.syr.edu

Abstract - *Image fusion is proposed as a method to combat errors during transmission of images on wireless channels. For images represented in the wavelet domain, diversity is used to obtain multiple data streams corresponding to the transmitted image at the receiver. These individual image data streams are fused to form a composite image with higher perceptual quality. Diversity combining methods using image fusion exploit the characteristics of the wavelet transform. Simulation results demonstrate that the perceptual quality of the received image can be significantly improved.*

Key Words: image fusion, diversity combining, image transmission

1. Introduction

The use of multiple images in fields such as remote sensing, medical imaging and automated machine vision has increased in the past decade. As a result of this, several image fusion techniques have been developed to produce a composite image with more useful information content for automatic computer analysis tasks as well as for human perception [1,2]. This paper applies image fusion concepts to a new area, namely to image transmission systems that employ wireless channels. For image transmission over wireless channels, several methods have been proposed in the literature [3-8] that use different types of error-correction coding, ARQ, or post-processing to deal with the channel errors. The goal of this paper is to introduce a novel image transmission method based on image fusion that can produce an image of high perceptual quality at the receiver.

Before an image is transmitted over a wireless channel, it is desirable to implement a method for representing the image that is resilient to channel errors. For an error resilient representation, wavelet based decomposition will be utilized for transmitting the image in its uncompressed state. During transmission, the image will be subject to bursty channel errors. Therefore, a technique is needed at the receiver to correct or conceal any errors that may degrade the perceptual quality of an image beyond acceptable limits.

Diversity is a communication method used to improve wireless transmission that utilizes independent (or highly uncorrelated) communication signal paths to combat channel noise. The independent signal paths provide the receiver with multiple signals for appropriate diversity processing of the received signals. For image transmission, a diversity technique has been employed in conjunction with ARQ [4]. This approach involves switched antenna diversity that operates in the data domain.

Unlike data domain diversity combining methods, the diversity combining method we propose here operates in the image domain by using the properties of the original image or its wavelet transform. Our novel approach to wireless image transmission combats the effects of fading and other channel impairments by employing a diversity combining method that attempts to directly improve image quality. This diversity combining method was inspired by the image fusion work of Burt [9] where he produced one composite image from multiple source images with different information content. Burt implemented his fusion method by taking a Laplacian pyramid transform of each source image, combining the transforms based on measures in the transform coefficient neighborhoods, and performing the inverse transform to obtain the composite image. Later, Li et al [10] used this same image fusion methodology but with the wavelet transform. For image transmission over wireless channels, two or more diversity channels can be utilized to obtain multiple bit streams at the receiver, with each bit stream independently representing the image data. Then these bit streams can be fused in the image domain to improve the perceptual quality of the received image. Due to the random nature of radio propagation, we expect the errors on the individual channels to be independent or at least highly uncorrelated. This allows for a fusion method that yields excellent quality images in the presence of wireless channel errors.

The organization of the rest of the paper is as follows. Section 2 briefly describes the channel model used for simulations. Our diversity combining method based on image fusion is discussed in Section

3 along with some results and conclusions are given in Section 4.

2. Channel Model

Wireless channels are corrupted by errors that are bursty in nature. Modeling of the physical channel is a complex problem that depends upon the movement of the transmitter, receiver, and other objects in the signal path. While a number of models that characterize the physical phenomena have been proposed in the literature, here we employ an channel model to generate error sequences that attempts to represent the input-output relationships of the wireless channel.

One popular input-output error model is in terms of a finite state Markov chain. In this model, each state represents a different channel condition and the associated error behavior. These models are specified in terms of transition probabilities between the individual states and the corresponding error probability for each state. The model we use for our simulations is a two-state Gilbert-Elliott channel [11, 12].

The two-state Gilbert-Elliott channel has one good state and one bad state, represented by 0 and 1 respectively as shown in Figure 1. This channel can also be described by its burst error length and error rate parameters, which are related to the transition probabilities between states and the error probabilities of the individual states. The average error rate is the proportion of errors to the total number of transmitted bits and the average burst error rate is the time spent in the bad state. While in the good state the bits are transmitted incorrectly with probability $P_e(0)$, and while in the bad state the bits are transmitted incorrectly with probability $P_e(1)$. For this model it is assumed that $P_e(0) << P_e(1)$. The two-state channel model can be described by the binary Markov process γ_n with the following transition matrix:

$$\mathbf{P} = \begin{bmatrix} P(\gamma_n = 0|\gamma_{n-1} = 0) & P(\gamma_n = 1|\gamma_{n-1} = 0) \\ P(\gamma_n = 0|\gamma_{n-1} = 1) & P(\gamma_n = 1|\gamma_{n-1} = 1) \end{bmatrix} = \begin{bmatrix} p & 1-p \\ r & 1-r \end{bmatrix}$$

where γ_n=0 if the channel is in the good state at time n, and γ_n=1 if the channel is in the bad state at time n. The average burst length L is a geometric random variable with mean $1/r$, and the average time the channel is in the good state is also a geometric random variable with mean $1/(1-p)$. The steady state probability of the channel being in a bad state is $\pi_1 = (1-p)/(r+1-p)$. Also, the steady-state error rate is given as $\varepsilon = (P_e(0)\cdot r + P_e(1)\cdot(1-p))/(r+1-p)$ [13]. This model will be used to generate errors to corrupt the images in our simulations in order to evaluate the performance of our diversity combining method.

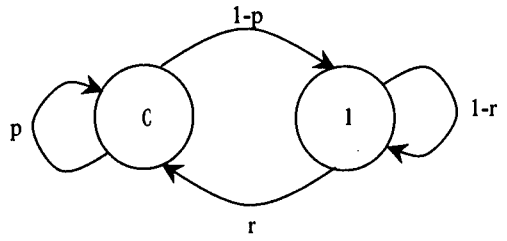


Figure 1. Two-state Gilbert-Elliott channel

3. Image Fusion for Diversity Combining

Our diversity combining method for uncompressed images involves computing the two-dimensional wavelet decomposition of the source image and quantizing the resulting wavelet coefficients. The coefficients are then transmitted as a bit stream over a wireless communications system employing diversity without any error control. Diversity is used to obtain multiple copies of the decomposed image data at the receiver. At the receiver, the individual decomposed images are fused to form a composite wavelet decomposition and then the final received image is reconstructed. This diversity combining method based on image fusion is depicted in Figure 2.

The first step in wireless image transmission is to consider how the image will be represented for transmission. The two-dimensional wavelet decomposition of an image is implemented with traditional subband filtering [14] using one-

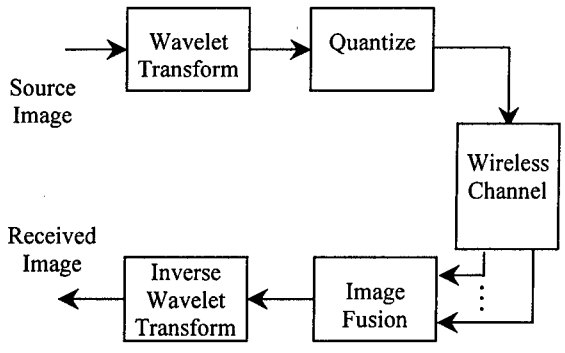


Figure 2. Image transmission method.

dimensional low-pass (H) and high-pass (G) quadrature mirror filters. First, the input image is convolved with H and G in the horizontal direction and then the output rows are down-sampled by two. Then the two resulting sub-images are further filtered along the vertical direction followed by down sampling of the columns. At the output, the source image at resolution k is decomposed into four sub-images: an image at lower resolution level k-1, a horizontally oriented detail image, a vertically oriented detail image, and a diagonally oriented detail image. The filtering can be repeated by using the low-resolution image as the source image until the desired decomposition level is reached. The image at resolution k is reconstructed from the four sub-images at resolution k-1 using reconstruction filters $\underline{H}$ and $\underline{G}$. The rows are up-sampled by two (one row of zeros is inserted between each row) and filtered in the vertical direction. Then the same procedure is followed in the horizontal direction. At the output, a reconstructed image at resolution k is obtained. Repeating the same procedure, the original level at which the decomposition was started can be reached.

In this paper, we use images transformed in the wavelet domain with uniform scalar quantization of the coefficients. The results obtained will help demonstrate the usefulness of image domain diversity combining for image transmission over wireless channels. For images without compression, the wavelet representations are obtained from the bit streams received on the individual diversity channels. In general, the low-resolution subband is more important perceptually and a large error in pixel intensity can seriously affect image quality. An error in the high frequency subband is not as important to the overall image quality. Because the characteristics of the subbands are different, the diversity-combining rule for the low-resolution subband differs from the combination rule for the high frequency subbands. After obtaining the composite decomposed image from fusing the individual transformed images, the inverse wavelet transform is performed to obtain the final image.

The idea behind diversity combination is to significantly reduce visible errors in the received image without necessarily using techniques such as ARQ or error correction coding. The diversity combining method is demonstrated here using two independent channels, channel one and channel two, but the idea can easily be extended to more channels. When the bit streams containing the decomposed images are received, a decision is made as to whether to take the data from channel one, channel two, or from a combination of both. Depending upon the channel state the two received bit streams will contain the same values for many of the coefficients.

The low frequency subband and high frequency subbands have different sensitivities to bursty channel errors. Therefore, the rules for the two types of subbands are different. For both of the different subband types there are two combination modes: selection and coefficient combining. In the selection mode, one coefficient is selected from the two decomposed images and placed in the composite. In the coefficient-combining mode, groups of coefficients from neighborhoods of both decomposed images are examined and a value is placed in the composite decomposed image based on measures from both coefficient neighborhoods. The combination method is similar to using both image averaging and spatial filtering to remove channel noise.

Since the low-resolution subband is more perceptually important to the image, more care must be taken when dealing with detected channel errors in the low-resolution subband. First, the coefficients from the two diversity bit streams are compared as they arrive at the receiver. If the received wavelet coefficient values are the same, we assume that the value is correct and select the coefficient from either channel to place in the combined transform. If the coefficient values are different, the receiver waits until an m by n neighborhood of coefficients surrounding the coefficient of interest is available from both channels. Small neighborhoods (i.e. 3 by 3) of an image are generally smooth. Therefore, the intensity values usually do not vary significantly within these neighborhoods. When the two received coefficients at location (i, j) are different, the m by n neighborhoods of coefficients around them are grouped into a set of $2mn$ values. Then the median value is chosen as the coefficient to place in the combined low-resolution sub-image at location (i, j). In general, this median-based method tends to be more robust to large channel errors than averaging the coefficients in order to obtain a combined coefficient value. Therefore, for each (i, j), the coefficient placed in the combined low resolution subband image is defined as follows (assuming m and n are odd):

$$c_{Lc}(i,j)=\begin{cases} c_{L1}(i,j) & \text{if } c_{L1}(i,j)=c_{L2}(i,j) \\ med\left[\{c_{L1}(k,l)\},\{c_{L2}(k,l)\}\right] & \text{if } c_{L1}(i,j)\neq c_{L2}(i,j) \end{cases}$$

$$\text{for} \quad (k,l)\in\left(\left\{i-\frac{m-1}{2},\ldots,i+\frac{m-1}{2}\right\},\left\{j-\frac{n-1}{2},\ldots,j+\frac{n-1}{2}\right\}\right)$$

where c_{Lc} represents the wavelet coefficients in the low-resolution subband of the combined transform,

and c_{L1} and c_{L2} are the low-resolution coefficients obtained from two diversity channels.

An error in the high frequency subbands does not affect the quality of the final reconstructed image as much as in the low frequency subbands. Also, most of the coefficients have magnitudes close to zero. Therefore, the errors in the detail subbands are processed differently when the received wavelet coefficients are not the same. Again, if the received wavelet coefficient values are the same, we assume that the value is correct and place this value in the combined transform. However, if the received coefficients are different, the coefficient with the minimum absolute value is chosen and placed in the final combined transform. The idea behind this selection method is that a coefficient that implies a strong edge where one does not exist will visually degrade the image more than a coefficient that implies no edge where one really exists. Since most of the coefficients in the high frequency subbands are near zero, there is a better chance that the coefficient with the minimum absolute value will be correct. Even if we set the coefficients to zero in the high frequency subbands, the quality of the final image will still be acceptable. The combined coefficient values for each location (*i,j*) in the high frequency subbands are given as follows:

$$c_{Hc}(i,j)=\begin{cases} c_{H1}(i,j) & \text{if } c_{H1}(i,j)=c_{H2}(i,j) \\ c_{H1}(i,j) & \text{if } |c_{H1}(i,j)|<|c_{H2}(i,j)| \\ c_{H2}(i,j) & \text{if } |c_{H2}(i,j)|<|c_{H1}(i,j)| \end{cases}$$

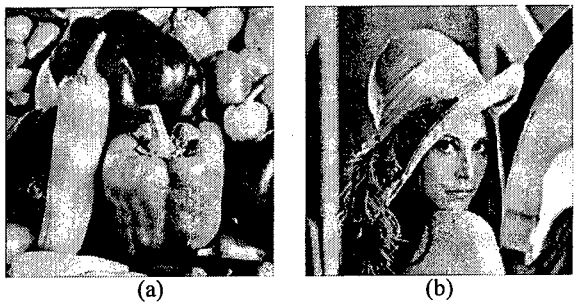
where c_{Hc} represents the wavelet coefficients in the detail subbands of the combined transform, and c_{H1} and c_{H2} are the detail subband coefficients obtained from two diversity channels.

In order to show the feasibility of using diversity combination for wireless image transmission, simulations were performed using uncompressed images. The results are compared to a system that uses error control coding for error protection. In our experiments, images were transmitted using a BCH(255, 179) code with error correction capability of 10 bits. For each simulation, two bit error patterns were generated using the two-state Markov model described in Section II. Both error patterns were applied to the image data bit streams for the diversity combination method and one of the error patterns was used for the error coding method. The parameters used for generating the bit error patterns were an average burst error length of 500 bits and various bit error rates (.0001, .0005, .001, .005, .01). The error probabilities within the individual states were set to $P_e(0) = 0.0$ and $P_e(1) = 0.5$. Performance is measured using peak signal to noise ratio (PSNR):

$$PSNR = 10\log_{10}\frac{255^2}{\frac{1}{N}\sum_i\sum_j (p(i,j)-\hat{p}(i,j))^2},$$

where $p(i,j)$ are the pixel values of the original image and $\hat{p}(i,j)$ are the pixel values of the received image.

For our simulations we tested our diversity combining method on the two images shown in Figure 3. Both are 8-bit graylevel images with 256 by 256 pixels. First, the source images were decomposed to two levels using the wavelet transform. Then the wavelet coefficients were uniformly quantized to 8 bits per pixel in order to maintain the same number of bits as in the original image. But for the bit stream with BCH coding, the total number of transmitted bits is greater than 8 bits per pixel. For uncompressed images we did not attempt to match bit rates for performance comparisons. The given PSNR results were averaged over twenty runs.



(a) (b)

Figure 3: Original test images for wireless image transmission: (a) Peppers and (b) Lenna.

Table 1 gives the PSNR results for the Peppers image using image fusion versus BCH(255, 179). In this table, we see that the PSNR results for image fusion are about 11 to 13 dB higher than error coding. Examples of the received Peppers images are shown in Figure 4 for bit error rates of 0.005 and 0.01.

Table 2 gives the PSNR results for the Lenna image using image fusion versus BCH coding where the image fusion method exceeds the error coding by about 15 to 16 dB. Examples of the received Lenna images are shown in Figure 5 for bit error rates of 0.005 and 0.01. These examples demonstrate that

image fusion can significantly improve performance compared to using BCH error correction coding.

Table 1: PSNR (dB) for Peppers

Bit error rate	Image Fusion	BCH(255, 179)
.0001	31.5507	20.2975
.0005	31.8934	19.9742
.001	32.2498	20.8729
.005	32.8253	20.3403
.01	30.2323	17.0615

Table 2: PSNR (dB) for Lenna

Bit Error Rate	Image Fusion	BCH(255, 179)
.0001	34.1783	20.7307
.0005	35.5003	20.8015
.001	35.5044	19.6481
.005	35.0813	20.3599
.01	33.0693	17.5477

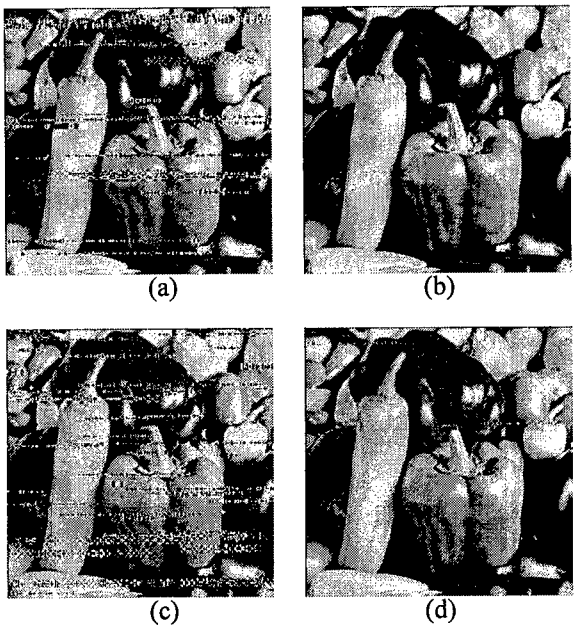


Figure 4: Received images for BER = 0.005 (a) BCH coding, (b) image fusion and BER = 0.01 (c) BCH coding and (d) image fusion.

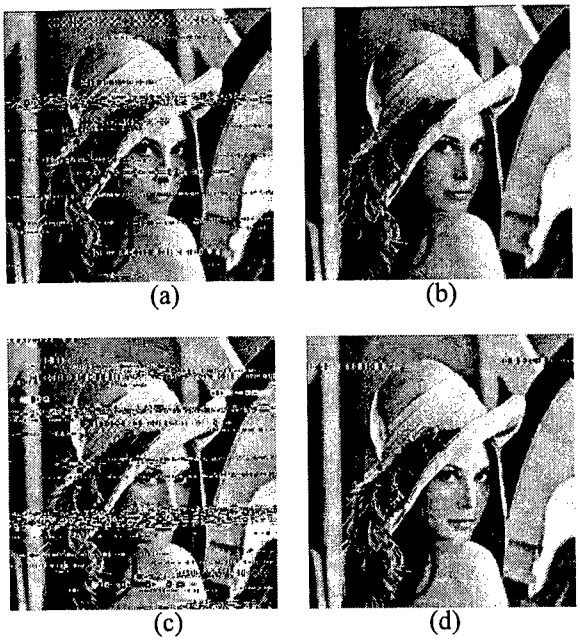


Figure 5: Received images for BER = 0.005 (a) BCH coding, (b) image fusion and BER = 0.01 (c) BCH coding and (d) image fusion.

4. Conclusions

An image domain diversity method has been presented for the transmission of images over wireless channels. For images represented in the wavelet domain, diversity is used to obtain multiple data streams of the image at the receiver where these data streams are fused to obtain a composite image. The methods proposed here use some of the properties of the wavelet transform to significantly improve the perceptual quality of the received image. Our results showed that image domain diversity could be used to improve performance for images transmitted over wireless channels. We have also implemented similar image fusion methods for compressed images to improve image quality and have obtained excellent results [15].

5. Acknowledgements

We would like to thank Mucahit Uner, Mark Alford and Dave Ferris for their help and support during this research.

This work was supported by Air Force Research Laboratory, Air Force Materiel Command, USAF, under grant number F30602-95-1-0027. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily

representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory or the U.S. Government.

References

[1] A. Toet, "Hierarchical image fusion," Machine Vision Applications, pp. 1-11, March 1990.

[2] M. Pavel, J. Larimer and A. Ahumada, "Sensor fusion for synthetic vision," in Proceedings of AIAA Conference on Computing in Aerospace, (Baltimore, MD), October 1991.

[3] P. Burlina and F. Alajaji, "An Error Resilient Scheme for Image Transmission over Noisy Channels with Memory," IEEE Trans. on Image Processing, Vol. 7, no. 4, pp. 593-600, April 1998.

[4] V. Weerackody and W. Zeng, "ARQ schemes with switched antenna diversity and their applications in JPEG image transmission," IEEE GLOBECOM, vol. 3, pp. 1915-1919, 1995.

[5] P.G. Sherwood and K. Zeger, "Error Protection for Progressive Image Transmission Over Memoryless and Fading Channels," IEEE Trans. on Communications, vol. 46, no. 12, pp. 1555-1559, December 1998.

[6] W.M. Lam and A.R. Reibman, "An Error Concealment Algorithm for Images Subject to Channel Errors," IEEE Transactions on Image Processing, vol. 4, no. 5, pp. 533-541, May 1995.

[7] Y. Koyama and S. Yoshida, "Error Control for Still Image Transmission over a Fading Channel," IEEE 45th Vehicular Technology Conference, Vol. 2, pp. 609-613, 1995.

[8] Z. Sun, J, Luo, C.W. Chen and K.J. Parker, "Analysis of a Wavelet-based Compression Scheme for Wireless Image Communication," Proc. of SPIE, Vol. 2762, pp. 454-465, 1996.

[9] P.J. Burt and R.J. Lolczynski, "Enhanced image capture through fusion," Proc. of Fourth International Conference on Computer Vision (Berlin, Germany), pp. 173-182, May 1993.

[10] H. Li, B.S. Manjunath and S.K. Mitra, "Multisensor image fusion using the wavelet transform," Graphical Models and Image processing, vol. 57, pp. 235-245, May 1995.

[11] E.N. Gilbert, "Capacity of a Burst-Noise Channel," The Bell Systems Technical Journal, pp. 1253-1265, September 1960.

[12] E.O. Elliott, "Estimates of error rates for codes on burst error channels," Bell Systems Technical Journal, vol. 42, p. 1977, Sept. 1963.

[13] J.R. Yee and E.J. Weldon, "Evaluation of the performance of error-correcting codes on a Gilbert channel," IEEE Trans. on Communications, vol. 43, pp. 2316-2323, Aug. 1995.

[14] J.W. Woods and S.D. O'Neil, "Subband coding of images," IEEE Trans. on Acoust., Speech, Signal Processing, vol. ASSP-34, Oct. 1986.

[15] L.C. Ramac, Ph.D. dissertation in progress.

Matching Segments in 3D Reconstruction using the Fuzzy Integral.

Andre BIGAND (*), Ludovic EVRARD (*), Jean Paul DUBUS (**)
* LASL, Universite du Littoral, BP649, 62228 Calais Cedex, France
** Laboratoire ID3, USTL, 59655 Villeneuve D'Ascq Cedex, France

Abstract ***This paper presents a matching algorithm based on linear features and fuzzy integral. The algorithm is primarily aimed at problems such as object description, pattern recognition problems, and related to the analysis of industrial plants (3D metrology) considered like polyhedric objects.An initial treatment provide labeled lines, each line being structured in labeled segments. We have to match these extracted segments into homogeneous surfaces. We manage uncertainties by two ways. The patterned light permits to obtain labeled segments belonging to the same surface with a good accuracy but it remains some uncertainty to match them, for they are not exactly coplanar. So we have a qualitative decision (do the segments belong or not to the same surface ?) under uncertainty in a finite setting to make. This decision is a one-shot decision so the Bayesian methods are difficult here to use and we decided to choose the Choquet integral-based utility, a generalisation of expected utility that is sum-decomposable for such acts in this numerical framework. We use three attributes of the planarity, based on fuzzy measures, to characterize the segments. These attributes are the parallelism, the overlapping zone, and the distance between the segments. This method gives good results on real images (as well as 3D scene description, an average variance of 5 per cent for the length and one per cent for the angles), and proves the interest of this tool (fuzzy integral) introduced in fusion information for image processing.***

Keywords : fuzzy logic, image fusion and machine vision, manufacturing.

1 Introduction

Information fusion is an important aspect of any decision system. Dealing with multiple input information sources is that the information coming from individual source is either incomplete or noisy that is , uncertain or imprecise. Numerous image processing systems or computer vision systems (pattern recognition, scene analysis, image processing, 3D reconstruction,...) belong to this category of decision taking problems. This paper presents a matching algorithm based on linear features and fuzzy integral. The algorithm is primarily aimed at problems such as object description, pattern recognition problems,... and related to the analysis of industrial plants (3D metrology) considered like polyhedric objects. Two different ways are known to describe scenes of three-dimensionnal objects. The three-dimensionnal scene description, using region-based 3D reconstruction techniques [Tar96] or the invariants of 3D structures approach to obtain reliable 3D primitives, [CB96], is investigated by many authors and needs two or more perspective views and the application of the projective geometry. The two dimensionnal scene description is more classical but have to deal with the importance of the 3D depth uncertainties and it is difficult to detect points belonging to the same planar surface. So we propose a new method to reconstruct planar surfaces of 3D objects with a good accuracy in 2D scene descrip-

tion. The scene is illuminated with patterned light and we use an effective decision theory tool, the fuzzy integral, to deal with the depth uncertainty and obtain a good accuracy.
Our device is composed by one CCD camera and a laser, which is able to generate 11 paralell planes through an optical head. The calibration procedure and the extraction of the luminous pattern which gets rid of optical defects, inherent to this system of vision, have been previously described in [EBD96a] and [EBD96b]. We can summarize the four steps of the initial treatment : the first step represents the application of the laser signal on a polyhedric plant, the second one the acquisition of the scene in the obscurity, the third one the obtaining of the labeled lines and the fourth one the extracted luminous pattern. So at the end of the initial treatment, we obtain labeled lines that are structured in labeled segments.
The second stage of our work is the matching of the segments which had been extracted into homogeneous surfaces. We manage uncertainties about the 3D points by two ways. The patterned light permits to obtain labeled segments belonging to the same surface with a good accuracy (this is one of the advantage of active vision) but it remains some uncertainty to match them, for they are not exactly coplanar. So we have a qualitative decision (do the segments belong or not to the same surface ?) under uncertainty in a finite setting to make. This decision is a one-shot decision so the Bayesan methods are difficult here to use and we decided to choose the Choquet integral-based utility, a generalisation of expected utility that is sum-decomposable for such acts in this numerical framework. We use three attributes of coplanarity, based on fuzzy measures, to characterize the segments. These attributes are the parallelism, the overlapping zone (recovery area) and the distance between the segments. This method gives good results (as well as 3D scene description, an average variance of 5 per cent for the length and one per cent for the angles), and proves the interest of this tool (fuzzy integral) introduced in fusion information for image processing.

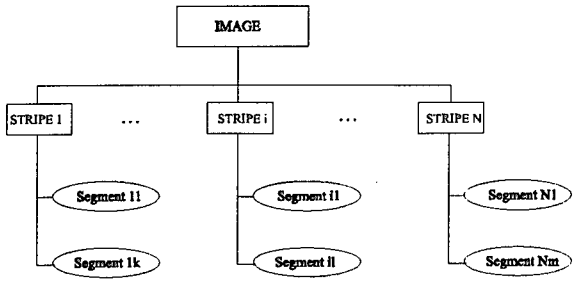


FIG. 1: Geometrical Interpretation of the Image

2 Problem Formulation by mean of synthetic evaluation

2.1 Introduction

The treatment presented in the previous chapter can be interpreted as an arborescent image (figure 1).
If a 2D segment belongs to a 3D surface, we need at least two segments and their attributes to completely characterize a 3D facet. It is possible to match the 2D segments of two adjacent stripes, as well as we can detect the edges between two 3D facets by image analysis. This method permits to deal with the raw sensorial data and avoid us to treat the 3D data (the reconstructed 3D data are imprecise). In fact, we prefer to judge the coplanarity from attributes on 2D segments than on 3D reconstructed segments considering the disparity of the 3D space. Then a decision tool, based on the Sugeno (fuzzy) measures and the Choquet integral is used to provide a *confidence measure* on the matching of two 2D segments according to the coplanarity hypothesis.

2.2 Choquet Integral

The Choquet integral is interesting to aggregate information in an uncertain environment and has been introduced in this sense by Denneberg, [D.D94] and Grabisch, [Gra95] and [Gra98]. Some applications in image processing were proposed in the last years, [H.T90].This fuzzy integral may be defined by the following

form :

$$(c)\int f d\mu = \int_0^{\infty} \mu(F_\alpha) d\alpha \qquad (1)$$

where $F_\alpha = \{x | f(x) \geq \alpha, x \epsilon X\}$, and f is a measurable function on $(X, \mathcal{P}(X))$ which associates a value f_j to each attribute x_j , this value being named *marginal valuation.*
In a numerical framework, the Choquet integral may be defined as following :

$$(c)\int f d\mu = \sum_{j=1}^{n} [f(x_j^*) - f(x_{j-1}^*)], C_\mu(\delta_1, \delta_2, ..., \delta_n) \qquad (2)$$

with :

$$\begin{cases} C_\mu(\delta_1, \delta_2, ..., \delta_n) = \mu(\bigcup_{i|\delta_i=1} x_i), \\ \forall(\delta_1, \delta_2, ..., \delta_n) \epsilon \{0,1\}^n \\ \\ and \\ \\ f(x_0^*) = 0 \end{cases}$$

In the equation 2, the "x_j^*" represent a new arrangment of the x_j for the marginal valuation $f_j = f(x_j)$ relative to each attribute x_j are in a non-decreasing order :

$$f(x_1^*) \leq f(x_2^*) \leq ... \leq f(x_n^*)$$

The coefficients C_μ are the *importance degrees* (of a belief measure μ) for the Choquet integral, defined on the set $\mathcal{P}(X)$.
For an application with a vector of attributes of length n, there exists 2^n measures C_μ. For instance, the importance degree relative to the attribute x_2 may be defined by :

$$C_\mu(0, 1, 0...0) = \mu(\{x_2\})$$

The non-additivity of the fuzzy integral comes from these coefficients C_μ. In fact, it is possible to define for instance $\mu(\{x_i, x_j\})$ with $i \neq j$, showing the interaction between the attributes x_i and x_j, that is not authorized by the arithmetic averaging by instance.

To calculate the synthetic valuation of the Choquet integral, we can define three steps :

- The first one is the choice of the attribute $X = \{x_1, x_2, ..., x_n\}$ with $(n \geq 2)$. In our application, the attributes were choosen according to the perceptual organization : parallelism, recovery area, and distance.
- The second step defines the belief measure μ, determining the importance degree on the set $\mathcal{P}(X)$ given to the different attributes and to their interaction.
- The third step considers the marginal valuations $f(x_j)$ obtained for each attribute x_j, and using the three similarity functions f_1, f_2 and f_3.

The implementation of the Choquet integral for decision taking is characterized by a process with multiple inputs and one output. In practice, an expert system (or an a-priori knowledge) may provide multiple synthetic valuation associated to some experiments. We obtain a matrix with m samples :

$$\begin{pmatrix} f_{11} & f_{12} & \cdots & f_{1n} \\ f_{21} & f_{22} & \cdots & f_{2n} \\ \vdots & \vdots & & \vdots \\ f_{m1} & f_{m2} & \cdots & f_{mn} \end{pmatrix} \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_m \end{pmatrix}$$

For this matrix, m is the number of experiments and n the number of attributes for a given application.
The synthetic valuation E_i is there defined by :

$$E_i = (c)\int f^{(i)} d\mu \quad , \quad \forall i = 1, 2, ..., m \qquad (3)$$

where the functions $f^{(i)}$ are :

$$f^{(i)}(x_j) = f_{ij} \quad , \quad j = 1, 2, ..., n$$

A belief measure μ on $(X, \mathcal{P}(X))$ does not exist to solve this system, so it is necessary to find an optimal approximation of the equation 3. The terminology *inverse problem of the synthetic valuation* is employed by some authors [WW97].
The classical technique consists of minimizing the quadratic error :

$$e = \frac{1}{2} \sum_{i=1}^{m} (E_i - \hat{E}_i)^2 \qquad (4)$$

where $\hat{E}_i$ is defined by an heuristic marginal valuation.
The Sugeno measure we employ like belief measure μ belongs to the family of the λ-regular fuzzy measures. This measure is defined on $(X, \mathcal{P}(X))$ and satisfy the following properties :

$$\begin{cases} \mu(0) = 0; \mu(X) = 1 \\ \mu(A \bigcup B) = \mu(A) + \mu(B) + \lambda.\mu(A).\mu(B) \end{cases}$$

A and B are two disjunctive sets, and $\lambda \epsilon (-1, +\infty)$.

Each Sugeno measure μ on X is characterized by n real values $a_j = \mu(\{x_j\}) \epsilon [0, 1]$. Wang et Wang [WW97] used the following form for the quadratic error :

$$e = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (E_i - \hat{E}_i)^2} \qquad (5)$$

The non-linearity of this expression does not authorize to find the relation $\frac{\partial e}{\partial a_j}$ and Wang et Wang used a neural net to calculate the Choquet integral.
We propose to use an optimisation method (Gauss-Newton algorithm) associated to a knowledge base to solve this problem.
The initial vector is $X_0 = \{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\}$, and an authorized error ε of 0.01. For instance,after 32 iterations, we obtain the optimal following values for our problem :

$$\begin{cases} a_1 = 0.7246 \\ a_2 = 0.5324 \\ a_3 = 0.1209 \end{cases}$$

The number λ is equal to 1.9232.

3 2D segments matching

3.1 Introduction

The matching of segments, well known in stereoscopic matching, is considered in this work to define planar surfaces, according to a perceptual organization. So we have to identify the different attributes according to a relation of similarity. This relation may be hierarchical, where we compute first a relation using one attribute (parallelism), then another relation using a second attribute (recovery area) and so on.
The drawback of this method is that we can eliminate "non-parallel" segments, non-parallel because of errors coming from previous treatments and noise, although they are in fact parallel. A global approach is better in that way like the rule-base introduced by Jain and Hoffmann [JH88]. We propose in the same idea to use an aggregate method based on the Choquet integral (a generalisation of weighted sum operators) and using the previous defined attributes.

3.2 Mathematical expression of the geometrical attributes

We remind that the illuminated image is represented by a set of stripes $\Re$:

$$\Re = \{R^{(1)}, R^{(2)}, ..., R^{(N)}\}$$

where N is the number of stripes, and each stripe $R^{(k)}$ is made of M_k linear segments $S_j^{(k)}$:

$$R^{(k)} = \{S_1^{(k)}, S_2^{(k)}, ..., S_{M_k}^{(k)}\} =$$

$$\{(d_1^{(k)}, f_1^{(k)}), (d_2^{(k)}, f_2^{(k)}), ..., (d_{M_k}^{(k)}, f_{M_k}^{(k)})\}$$

where $S_j^{(k)} = (d_j^{(k)}, f_j^{(k)})$ and

$$\begin{cases} d_j^{(k)} = (u_{d_j}^{(k)}, v_{d_j}^{(k)}) \\ f_j^{(k)} = (u_{f_j}^{(k)}, v_{f_j}^{(k)}) \end{cases}$$

The values $d_j^{(k)}$ and $f_j^{(k)}$ define the begining and the ending of the segment j of the stripe k, and are composed of the image coordinates u et v.

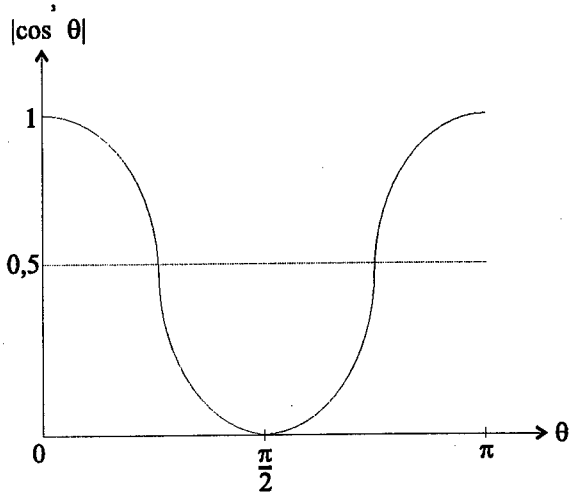


FIG. 2: Function representation f_1

To implement the Choquet integral, we need to calculate a marginal valuation for each attribute. The first function we define is f_1 which gives a measure of similarity on the orientation of the segment i of the stripe k ($S_i^{(k)}$) and the segment j of the stripe $k+1$ ($S_j^{(k+1)}$).
Parallelism between two segments is then defined modulo Π, and we can propose the following function f_1 :

$$f_1 = |\cos^{n+1}(\Theta_{ij}^{(k)})| \qquad (6)$$

with $n = 0, 1, 2, ..., \infty$.
The integer and positive coefficient n permits to make the function f_1 more selective.

The function f_1 is represented figure 2, with a constant $n = 2$. If the segments are parallel, the similarity measure tends to 1, and to 0 at the opposite.

While a segment is defined by the image coordinates of its extremities, the measure of $\Theta_{ij}^{(k)}$ is defined by the following expression :

$$\Theta_{ij}^{(k)} = \arctan\left(\frac{u_{d_j}^{(k+1)} - u_{f_j}^{(k+1)}}{v_{d_j}^{(k+1)} - v_{f_j}^{(k+1)}}\right) - \qquad (7)$$

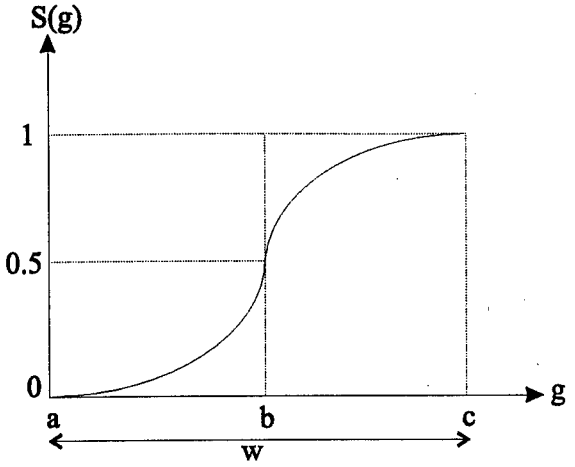


FIG. 3: S Function

$$\arctan\left(\frac{u_{d_i}^{(k)} - u_{f_i}^{(k)}}{v_{d_i}^{(k)} - v_{f_i}^{(k)}}\right) \qquad (8)$$

The second function we have to define is f_2, for the valuation of the similarity measure (given in [0,1]), relative to the recovery area. This calculus is based on the orthogonal projection of each segment on their bisectrix and define three classes of recovery, partial, complete and separation. We have to distinguish the complete recovery from the separation, that shows the belonging of the segments to the same surface. That implies a value "1" for the function f_2, and the separation implies the value "0". The uncertainty is about the partial recovery so we have choosen for the function f_2 the "S" function defined by Zadeh in fuzzy logic and represented figure 3.

$$S(g) = \begin{cases} 0 & if \quad g \leq a \\ 2(\frac{g-a}{c-a})^2 & if \quad a < g \leq b \\ 1 - 2(\frac{g-c}{c-a})^2 & if \quad b < g < c \\ 1 & if \quad g \geq c \end{cases} \qquad (9)$$

with $b = \frac{a+c}{2}$ and $w = c - a$.

The g parameter is the ratio of (length of recovery area) on (length of the segment), $a = 0$ (separation) and $c = 1$ (total recovery), so the S function can be modified :

$$S(g) = \begin{cases} 0 & if \quad g = 0 \\ 2g^2 & if \quad 0 < g \leq \frac{1}{2} \\ 1 - 2(g-1)^2 & if \quad \frac{1}{2} < g < 1 \\ 1 & if \quad g = 1 \end{cases} \tag{10}$$

The S function defines the measure of the recovery area relative to a segment. The function f_2 have to calculate this measure relatively to the right and left segment $S_i^{(k)}$ and $S_j^{(k+1)}$. As the two segments are close, the function f_2 may be written using the relation :

$$f_2 = \frac{1}{2}\left[S(\frac{L_{ij}}{L_i^{(k)}}) + S(\frac{L_{ij}}{L_j^{(k+1)}}) \right] \tag{11}$$

where L_{ij} defines the length of the recovery area between the segments $S_i^{(k)}$ and $S_j^{(k+1)}$, $L_i^{(k)}$ represents the length of the segment $S_i^{(k)}$ and $L_j^{(k+1)}$ represents the length of the segment $S_j^{(k+1)}$. These lengthes are defined by :

$$\begin{cases} L_i^{(k)} = \sqrt{(u_{d_i}^{(k)} - u_{f_i}^{(k)})^2 + (v_{d_i}^{(k)} - v_{f_i}^{(k)})^2} \\ L_j^{(k+1)} = \sqrt{(u_{d_j}^{(k+1)} - u_{f_j}^{(k+1)})^2 + (v_{d_j}^{(k+1)} - v_{f_j}^{(k+1)})^2} \end{cases}$$

The length of the recovery area depends closely to the distance separating the segments. So we have to define a third attribute :

$$f_3 = \frac{1}{1 + \frac{dist}{(\frac{512}{N})}} \tag{12}$$

The value *dist* represents the distance (in pixels) separating the centers of each segment , while the ratio $\frac{512}{N}$ define a reasonable distance with 512 being the horizontal resolution of the image and N the number of stripes .

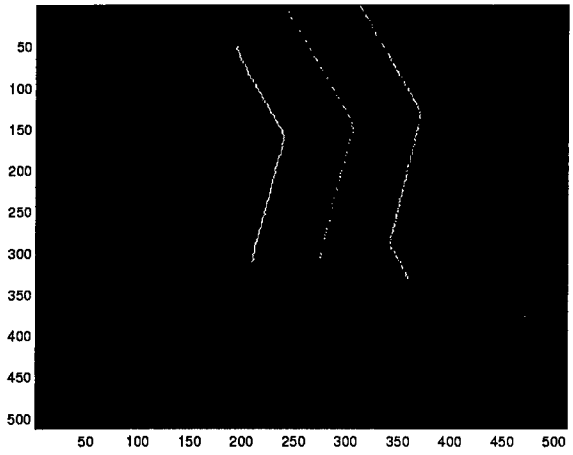


FIG. 4: Test image

When the stripes are close, $dist \leq \frac{512}{N}$ and f_3 tend to 1. When they are far, $dist \geq \frac{512}{N}$ and f_3 tend to 0. The function f_3 is $\frac{1}{2}$ if the distance separating two segments is reasonable.

3.3 Application

The Sugeno measures μ obtained have been applied to a test image (figure 4), with three stripes and we can represent this image by the following set :

$$\Re = \{R^{(1)}, R^{(2)}, R^{(3)}\} \quad with$$
$$\begin{cases} R^{(1)} = \{S_1^{(1)}, S_2^{(1)}\} \\ R^{(2)} = \{S_1^{(2)}, S_2^{(2)}, S_3^{(2)}\} \\ R^{(3)} = \{S_1^{(3)}, S_2^{(3)}, S_3^{(3)}\} \end{cases}$$

The results were the following :

- the 2D segments, which 3D homologuous segments belong to the same surface have a synthetic valuation upper to 0.7652
- the 2D segments, which 3D homologuous segments do not belong to the same surface have a synthetic valuation lower to 0.5515

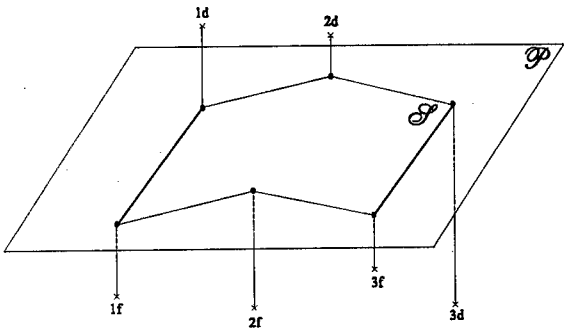


FIG. 5: Planar surface reconstruction

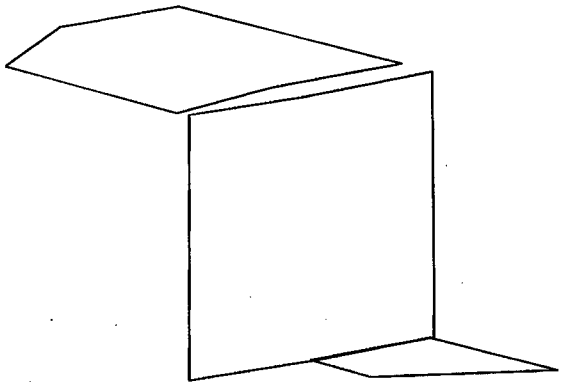


FIG. 6: Planar surface reconstruction of the test image

4 3D Surface Reconstruction

When the segments $S_i^{(1)}$ and $S_j^{(2)}$ have been matched, we deal with groups Gr of segments belonging to the same surface :

$$Gr = S_i^{(1)} \bigcup S_j^{(2)} \bigcup \ldots \bigcup S_m^{(N)}$$

For each group, we know the 2D coordinates of the different segments. Each stripe being labeled, we know too the equation of the plane corresponding to the surface created by these segments. The equation of the 3D plane is then calculated by a least-square method and illustrated figure 5.

The 3D reconstruction of the test image is illustrated figure 6.

5 Conclusion

We have mathematicaly formulate three geometrical attributes based on the Choquet integral and Sugeno measures for perceptual organization. This original method in matching segments to planar surfaces provide good results, as well as 3D planar surfaces reconstruction obtained by Tarel [Tar96] for instance, and remains simple to implement.
The greater interest of the fuzzy integral in this application is that we assign importance degrees to the interaction between attributes, non authorized in other aggregation method and permitting to deal with the depth uncertainty. For the exemple proposed, sensors are placed at 40 cm from the scene. We obtain a statistical error of 0.55 mm (for a length of 5cm), that is a relative error of 1,1% and an angle error of 1,3° .
With the Tarel 3D method we obtain 1,17 % of relative error on length and 2,16° for the angle error. So this method offers a new possibilty in 2D image processing and we are being to estimate the potentiality of fuzzy integral for complex objects (linear and curve primitives) and in dynamic vision.

Références

[CB96] H. Chabbi and M.O. Berger. Using projective geometry to recover planar surfaces in stereovision. *Pattern Recognition*, 29(4) :533–548, 1996.

[D.D94] D.DENNEBERG. *Non-Additive Measure and Integral.* Kluwer Academic Publishers, London, 1994.

[EBD96a] L. Evrard, A. Bigand, and J.P. Dubus. Fuzzy image analysis using multistripe structured light. In *Europto/SPIE Congress*, Besançon, 1996.

[EBD96b] L. Evrard, A. Bigand, and J.P. Dubus. Structural fuzzy-image analysis and structured light. In *3rd France-Japan Congress of MECATRONICS*, Besançon, 1996.

[Gra95] M. Grabisch. On equivalence classes of fuzzy connectives-the case of fuzzy integrals. *IEEE Transactions on Fuzzy Systems*, 3(1) :96–109, February 1995.

[Gra98] M. Grabisch. Fuzzy integral as a flexible and interpretable of aggregation. In Physica Verlag, editor, *Agregation and Fusion of Imperfect Information*. B. Bouchon-Meunier, 1998.

[H.T90] J.M.KELLER H.TAHANI. Information fusion in computer vision using the fuzzy integral. *IEEE Trans. on SMC, Vol.20, No3,*, 20 :733–741, 1990.

[JH88] A.K. Jain and R. Hoffmann. Evidence-based recognition of 3-d objects. *IEEE Transactions on PAMI*, 10(6) :783–802, november 1988.

[Tar96] J.-P. Tarel. Reconstruction globale et robuste de facettes 3d. Technical Report 2813, INRIA, Février 1996.

[WW97] J. Wang and Z. Wang. Using neural networks to determine sugeno measures by statistics. *Neural Networks*, 10(1) :183–195, 1997.

Sensor Fusion of a CCD Camera and an Acceleration-Gyro Sensor for the Recovery of Three-Dimensional Shape and Scale

Toshiharu Mukai
Bio-Mimetic Control Research Center
The Institute of Physical and
Chemical Research (RIKEN)
Nagoya, 463-0003 Japan

Noboru Ohnishi
Dept. of Information Engineering
Faculty of Engineering
Nagoya University
Nagoya, 464-8603 Japan

Abstract

Shape recovery methods from an image sequence have been studied by many researchers. Theoretically, these methods are perfect, but they are sensitive to noise, so that in many practical situations, we could not obtain satisfactory results. In addition, we could not obtain the scale of the recovered object because of the image-projection property. To solve these problems, we propose a shape recovery method based on the sensor fusion technique. This method uses an acceleration-gyro sensor attached on a CCD camera for compensating images.

Keywords: Recovery from images, Sensor fusion, Gyro sensor, Three-dimensional model

1 Introduction

Image changes produced by a moving camera are an important source of information on the observer's motion and structure of the environment. These changes are represented by velocities called optical flow on an image screen or point-correspondences between two or more images. The recovery of a three-dimensional structure and motion from an image sequence is one of the most important issues in computer vision. It can be used in many fields such as three-dimensional object modeling, tracking, passive navigation, and robot vision.

Recovery methods from an image sequence have been proposed by many researchers (for example, [1, 2, 3, 4]). Theoretically, these methods are perfect, but they are very sensitive to noise, so that, in many practical situations, we could not obtain satisfactory results.

Recently the factorization method developed by Tomasi and Kanade has attracted researchers' attentions [5]. This method has been proposed for orthogonal projection [5] and then extended for approximations of perspective projection [6]. It is reported that good results have been obtained in practical situations by the use of this method, when the approximation of the camera model is suitable. However when the assumed camera approximation is not suitable for the situation or the amount of camera motion through an image sequence is small, the results are not satisfactory yet.

There is another limitation comes from an image property. Under the perspective projection or orthogonal projection which is widely used as a camera model, a slowly-moving small object near to a camera produces perfectly the same image sequence as a fast-moving object far from a camera. It means that we could not recover the scale concerning the object and camera motion (velocity or displacement).

In order to solve these problems, we propose the use of an acceleration-gyro sensor attached onto a CCD camera. We selected the sensor because it does not require any environmental setting, so that the sensing system can be carried anywhere. In the following sections, we propose a method for the recovery of object shape and scale from the output of the CCD camera and acceleration-gyro sensor. Experimental results are also shown.

2 Sensor Fusion for Obtaining Good-Quality Information

One of the causes of the difficulty in shape recovery is the fact that the discrimination of small rotation and small translation, as shown in Figure 1, is diffi-

cult when the object's width along the optical axis relative to the distance between the camera and the object is small, because they invoke similar image changes.

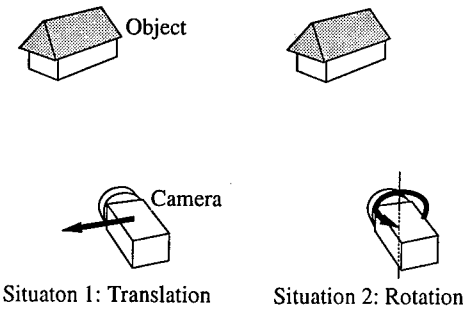


Figure 1: Small rotation and small translation have a similar effect on the image screen.

When we study animals, we find that many control their eye motion so as to obtain better visual information. For example, the vestibulo-ocular reflex makes us possible to obtain stabilized images on the retina. This reflex rotates our eyeballs so as to cancel rapid head motions, by using information on our head's rotation obtained from three semicircular canals [7]. It is also reported that, when flying, insects control the direction of their bodies to obtain visual information without rotation [8].

In our system, we do not control the video camera to remove rotation, in order to build a compact, inexpensive and free-from-mechanical-problems system. Instead, we process the image sequence by a computer to remove rotation using output from the gyro sensor obtained simultaneously with the image sequence. Conceptually, we design a virtual camera, as shown in Figure 2. This virtual sensor receives input from the video camera and gyro sensor and outputs an image sequence without rotation.

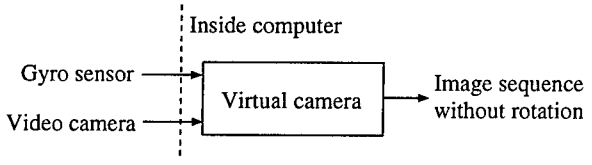


Figure 2: The virtual camera outputs an image sequence without rotation.

3 Overview of Our System

3.1 Setup

Our system consists of two sensors, a CCD camera and an acceleration-gyro sensor, and a computer for processing. The purpose of our system is the recovery of an object's three-dimensional structure including its scale from the sensor output. We assume the following situation. The rigid object is fixed in the environment and the sensor system moves around it. The object has feature points which can be tracked through an image sequence and its structure is determined by three-dimensional feature point positions.

The acceleration-gyro sensor (GU-3011 by Data Tec), mounted on the CCD camera, as shown in Figure 3, is used to compensate the CCD camera. It consists of 3 vibration gyroscopes and 3 acceleration sensors in a cube with sides 36 mm long and outputs 3-axial acceleration, 3-axial angular velocity and 3-axial rotation angle at 60 Hz. The rotation angle is obtained by integrating angular velocity, so that it drifts even though it can be corrected to some extent by using gravity as reference. In the present paper, we use the acceleration and angular velocity information.

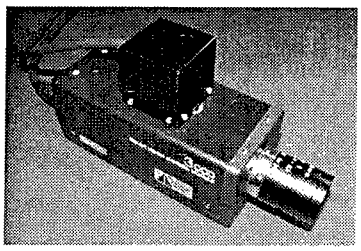


Figure 3: A photograph of the CCD camera and acceleration-gyro sensor

3.2 Four Stages in Our Method

Our shape and scale recovery method using the CCD camera and acceleration-gyro sensor consists of four stages.

In the first stage, optical flow or point-correspondences through the image sequence are obtained. This is usually achieved by tracking feature points in the image sequence. Many methods are studied for this purpose but, are not discussed here.

In the second stage, we use one of any shape-from-image-sequence methods, modified for use of

the acceleration-gyro sensor. This stage recovers the shape of the object and camera velocity and angular velocity at a point of time. It should be noted that the scale factor concerning the camera velocity and the recovered point positions cannot be obtained in this stage.

The third stage is the integration of recovered parameters at different time points, which are obtained in the previous stage. By integration of the images, three effects can be expected. When recovered points exist in more than one structure at different time points, more accurate positions can be obtained by taking their average. In addition, if there are points which exist in only some of the recovered structures, they are appended to the rest of the points. That is, occluded points in some parts of the image sequence can be recovered by integration if the points are viewed in other parts of the image sequence. Finally, transition of camera motion is obtained.

In the fourth stage, we obtain the scale concerning recovered positions and camera velocity. By integrating acceleration output from the acceleration-gyro sensor, camera velocity is obtained, theoretically. However, the acceleration output includes noise in practice, so that the velocity obtained from the acceleration-gyro sensor drifts. By using both acceleration from the acceleration-gyro sensor and velocity with the unknown scale factor obtained from the image sequence, the scale is obtained.

3.3 Three Coordinate Systems

Vector elements depend on the coordinate system to which the vector is related. For representing vector elements, we use three kinds of coordinate systems.

The first is fixed in the world and is constant over time. We call this a base world coordinate system. Recovered structures and camera motion parameters at different time points are integrated in this coordinate system. When a vector $\boldsymbol{x}$ is represented in this coordinate system, it is denoted as $\boldsymbol{x}^B$.

The second is a camera coordinate system which is attached to and moves with the CCD camera. When a vector $\boldsymbol{x}$ is represented in this coordinate system, it is denoted as $\boldsymbol{x}^C$.

The last coordinate system is also fixed in the world, but its position is changed according to the referred time. It is used to represent the sensor output obtained at the referred time. The coordinate system is positioned so as to correspond with the camera coordinate system when the sensor output is obtained. Therefore this depends on time. We termed this a temporary world coordinate system. When a vector $\boldsymbol{x}$ is represented in this coordinate system, it is denoted as $\boldsymbol{x}^W$.

An example of the application of the coordinate systems is as follows. When camera velocity is $\boldsymbol{v}^B$, $\boldsymbol{v}^C = \boldsymbol{o}$ because the camera coordinate system moves with the camera. The relation between the base world coordinates and the temporary world coordinates is $\boldsymbol{v}^W = R\boldsymbol{v}^B$ where R is the rotation from the base world coordinates to the temporary world coordinates. Vector coordinates do not depend on the position of the origin of the related coordinate system, so that they can be transformed by only rotation.

Output of the acceleration-gyro sensor is based on the temporary world coordinate system. That is, acceleration $\boldsymbol{a}^W$ and angular velocity $\boldsymbol{\omega}^W$ are obtained from the sensor.

4 The Recovery of Object Shape and Camera Motion from an Image Sequence

The recovery of object shape and camera motion from an image sequence at a point of time is studied by many researchers. We modify one of these methods in order to use the acceleration-gyro sensor output for compensating images, then use it in our system. In this section, we briefly introduce the method previously proposed by us. The details are reported in [4].

Assume that we observe a point on the object at time t and $t + \delta t$. We denote the unit vector from the camera center to the point as $\boldsymbol{q}^W$ and camera translation as $\delta\boldsymbol{u}^W$, as shown in Figure 4. Then we obtain

$$(\boldsymbol{q}^W(t+\delta t) \times \boldsymbol{q}^W(t)) \cdot \delta\boldsymbol{u}^W = 0, \qquad (1)$$

because $\boldsymbol{q}^W(t)$, $\boldsymbol{q}^W(t+\delta t)$ and $\delta\boldsymbol{u}^W$ are on the same plane.

Taking $\delta t \longrightarrow 0$ and using the following relation (this can be easily proved)

$$\dot{\boldsymbol{q}}^W = \dot{\boldsymbol{q}}^C + \boldsymbol{\omega}^W \times \boldsymbol{q}^C, \qquad (2)$$

we obtain

$$((\dot{\boldsymbol{q}}^C + \boldsymbol{\omega}^W \times \boldsymbol{q}^C) \times \boldsymbol{q}^C) \cdot \boldsymbol{v}^W = 0. \qquad (3)$$

By arranging the above equation on $n(\geq 8)$ points, we obtain

$$G\boldsymbol{\xi} = \mathbf{o}, \qquad (4)$$

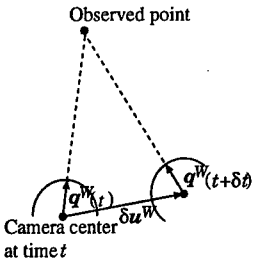


Figure 4: Relationship of camera positions before and after infinitesimal time lapse.

where

$$\boldsymbol{\xi} = [v_1^W \mid v_2^W \mid v_3^W \mid \omega_1^W v_1^W \mid \omega_2^W v_2^W \mid \omega_3^W v_3^W \mid \omega_1^W v_2^W + \omega_2^W v_1^W \mid \omega_2^W v_3^W + \omega_3^W v_2^W \mid \omega_3^W v_1^W + \omega_1^W v_3^W]^T \quad (5)$$

and G is a $n \times 9$-matrix composed of only observed values. The ith row of G is

$$[\dot{q}_{i;2}^C q_{i;3}^C - \dot{q}_{i;3}^C q_{i;2}^C \mid \dot{q}_{i;3}^C q_{i;1}^C - \dot{q}_{i;1}^C q_{i;3}^C \mid \dot{q}_{i;1}^C q_{i;2}^C - \dot{q}_{i;2}^C q_{i;1}^C \mid (q_{i;1}^C)^2 - 1 \mid (q_{i;2}^C)^2 - 1 \mid (q_{i;3}^C)^2 - 1 \mid q_{i;1}^C q_{i;2}^C \mid q_{i;2}^C q_{i;3}^C \mid q_{i;3}^C q_{i;1}^C], \quad (6)$$

where the dot on variables denotes the time derivative of the variable and $q_{i;j}^C$ is the jth element of $\boldsymbol{q}_i^C$ which is the unit vector from the camera center to the ith point.

By finding nontrivial $\boldsymbol{\xi} \neq \mathbf{o}$ from (4), camera velocity $\boldsymbol{v}^W$ up to its scale and angular velocity $\boldsymbol{\omega}^W$ are obtained. We select the unit vector $\hat{\boldsymbol{v}}_s^W$ as the recovered camera velocity and denote the recovered camera angular velocity as $\hat{\boldsymbol{\omega}}^W$. The positions of observed points are also recovered as

$$\boldsymbol{x}_i^W = -s\frac{\hat{\boldsymbol{v}}_s^W \cdot (\dot{\boldsymbol{q}}_i^C + \hat{\boldsymbol{\omega}}^W \times \boldsymbol{q}_i^C)}{\|\dot{\boldsymbol{q}}_i^C + \hat{\boldsymbol{\omega}}^W \times \boldsymbol{q}_i^C\|^2}\boldsymbol{q}_i^C, \quad (7)$$

where s is the unknown scale factor and $\boldsymbol{v}^W = s\hat{\boldsymbol{v}}_s^W$ if noise is absent.

5 Using the Acceleration-Gyro Sensor for Compensating an Image Sequence

In this section, we describe a modification of the shape and motion recovery method described in the previous section.

From the acceleration-gyro sensor, $\boldsymbol{\omega}^W$ is obtained. By substituting this into (2), we obtain $\dot{\boldsymbol{q}}^W$. This can be considered as the output of the virtual camera in Figure 2. The virtual camera output when observing m points ($m \geq 2$) yields

$$H\boldsymbol{v}^W = \boldsymbol{o}, \quad (8)$$

where H is the $m \times 3$-matrix and its ith row is

$$\dot{\boldsymbol{q}}^W \times \boldsymbol{q}_i^C. \quad (9)$$

This can be determined by observed values only. This equation is obtained from (3).

By finding nontrivial $\boldsymbol{v}^W \neq \mathbf{o}$ from this equation, we obtain the velocity up to its scale. This velocity is expected to be better than the previous one because the degree of freedom in the equation is smaller.

In practice, m is usually much larger than 3 and H is disturbed by noise, so that the matrix has rank 3. Hence this equation is ill-conditioned for obtaining $\boldsymbol{v}^W \neq \mathbf{o}$. The SVD (Singular Value Decomposition) method is suitable for solving this equation. By the SVD, H is decomposed as

$$U\Sigma V^T = [\boldsymbol{u}_1|\boldsymbol{u}_2|\boldsymbol{u}_3]\mathrm{diag}\{\sigma_1, \sigma_2, \sigma_3\}[\boldsymbol{v}_1|\boldsymbol{v}_2|\boldsymbol{v}_3]^{\mathrm{T}}, \quad (10)$$

where U and V are orthonormal matrices and $\sigma_1 \geq \sigma_2 \geq \sigma_3$. Then $\boldsymbol{v}_3$ is adopted as $\hat{\boldsymbol{v}}_s^W$. Point positions can be determined by (7).

6 Integration of Recovered Structures and Motion Parameters at Different Time Points

The integration of recovered structures at different time points is expected to improve accuracy, recover occluded points and clarify the transition of camera motion. However, the object structures and camera motion parameters at different time points are obtained with respect to different temporary world coordinate systems. The scales are also different because they cannot be determined by the method in the previous stage. Hence we cannot simply integrate recovered structures.

This problem can be solved as follows. The object shapes are the same even though coordinate systems are different. Therefore we can determine (relative) scaling, rotation and translation transformations which make transformed structures overlap each other. We take the structure at the first time

point as the base structure and find the transformation to this structure. This means that the temporary world coordinate system at the first time point is used as the base world coordinate system.

We denote the recovered position of point i with respect to the temporary world coordinate system at time k as $\boldsymbol{x}_i^k$. The superscript W is dropped in this section for concise description. The transformed point position $\boldsymbol{x}_i'^k$ from $\boldsymbol{x}_i^k$ is defined as

$$\boldsymbol{x}_i'^k = s^k R^k \boldsymbol{x}_i^k + \boldsymbol{t}^k, \tag{11}$$

where s^k, R^k and $\boldsymbol{t}^k$ are the scaling, rotation and translation from the structure at time k to the base structure.

In order to obtain $s^k, R^k, \boldsymbol{t}^k$, we minimize

$$E_1^k(s^k, R^k, \boldsymbol{t}^k) = \frac{1}{2}\sum_i \{\tilde{\boldsymbol{x}}_i - \boldsymbol{x}_i'^k\}^2, \tag{12}$$

where $\tilde{\boldsymbol{x}}_i$ is the position of point i in the base structure. In practice, the translation $\boldsymbol{t}^k$ is obtained from $\partial E_1^k/\partial \boldsymbol{t} = \mathbf{o}$ as

$$\boldsymbol{t}^k = \tilde{\boldsymbol{g}} - s^k R^k \boldsymbol{g}^k, \tag{13}$$

where $\tilde{\boldsymbol{g}}, \boldsymbol{g}^k$ are the centroids of $\tilde{\boldsymbol{x}}_i, \boldsymbol{x}_i^k$. Hence we minimize

$$E_1^k(s^k, R^k) = \frac{1}{2}\sum_i \left(\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{g}} - s^k R^k (\boldsymbol{x}_i^k - \boldsymbol{g}^k)\right)^2. \tag{14}$$

In our implementation, we used the conjugate gradient method for minimizing the function numerically.

Using $s^k, R^k, \boldsymbol{t}^k$ obtained above, we can obtain a better object structure and camera motion as follows.

Object structure: Taking the average of the structures transformed using scaling, translation and rotation, accuracy is improved. If the corresponding point does not exist in the integrated structure yet, it is appended to the integrated structure.

Camera velocity: Transforming using only scaling, $\boldsymbol{v}^W(t)$ is recovered. Transforming using scaling and rotation, $\boldsymbol{v}^B(t)$ is recovered.

Camera angular velocity: From the recovery at a point of time, $\boldsymbol{\omega}^W(t)$ is recovered. So, transforming using rotation, $\boldsymbol{\omega}^B(t)$ is recovered.

Camera position: In the recovery at a point of time, the camera center is assumed to be at the origin of the temporary world coordinate system. Using the scaling, translation and rotation information, transition of the camera center position and direction in the base world coordinate system is obtained.

7 Determining the Scale of Structure and Velocity

In this section, we denote camera velocity obtained from an image sequence as $\boldsymbol{v}_I^W(t)$, that from the acceleration-gyro sensor as $\boldsymbol{v}_G^W(t)$, and true camera velocity as $\boldsymbol{v}_T^W(t)$. Then, if noise is absent,

$$\boldsymbol{v}_T^W(t) = s\boldsymbol{v}_I^W(t), \tag{15}$$

where s is the unknown scale factor. Therefore if the relation between $\boldsymbol{v}_T^W(t)$ and $\boldsymbol{v}_G^W(t)$ is known, we can determine s form the above equation.

Theoretically, $\boldsymbol{v}_G^W(t)$ is obtained by integrating acceleration $\boldsymbol{a}^W(t)$ obtained from the acceleration-gyro sensor if the initial value is known. It is formulated as

$$\boldsymbol{v}_G^W(t) = R(t)\Big\{\int_{t_0}^{t}\{R^{-1}(\tau)\boldsymbol{a}^W(\tau) - \boldsymbol{g}^B\}d\tau + \boldsymbol{v}^B(t_0)\Big\}, \tag{16}$$

where $\boldsymbol{v}^B(t_0)$ is the initial velocity and $R(t)$ is the rotation form the base world coordinates to the temporary world coordinates. It can be obtained from the acceleration-gyro sensor output $\boldsymbol{\omega}^W(t)$ if the initial value $R(t_0)$ is known.

However, in practice, the acceleration-gyro sensor output includes noise so that $\boldsymbol{v}_G^W(t)$ drifts. Hence the following relation holds.

$$\boldsymbol{v}_T^W(t) = \boldsymbol{v}_G^W(t) + \boldsymbol{b}(t) \tag{17}$$

The $\boldsymbol{b}(t)$ represents the effect of drift and the unknown initial value. The change of $\boldsymbol{b}(t)$ comes from the drift, so we can assume that the change in short time is small.

From discretization of above equations, we obtain

$$s\boldsymbol{v}_I^{W;k} = \boldsymbol{v}_G^{W;k} + \boldsymbol{b}^k, \tag{18}$$

where $\boldsymbol{v}_I^{W;k}$ is $\boldsymbol{v}_I^W$ at time k ($k = 0, 1, \ldots, K$) and so on, and changes of $\boldsymbol{b}^k$ along k are small. We minimize the following function for obtaining s.

$$E_2(s, \boldsymbol{b}_i) = \frac{1}{2}\sum_{k=0}^{K}\|s\boldsymbol{v}_I^{W;k} - (\boldsymbol{v}_G^{W;k} + \boldsymbol{b}^k)\|^2 + \frac{1}{2}\alpha\sum_{k=0}^{K}\|2\boldsymbol{b}^k - \boldsymbol{b}^{k-1} - \boldsymbol{b}^{k+1}\|^2, \tag{19}$$

where $\boldsymbol{b}^{-1} = \boldsymbol{b}^{K+1} = \mathbf{o}$ and α is some positive value for weighting.

8 Experiments

8.1 Experimental Environment

We used a cube with a known size, shown in Figure 5, for examining the recovery errors. The cube has sides 20 cm long. The acceleration-gyro sensor output is obtained at 60 Hz via serial connection. The CCD camera has lens whose focal length is 8 mm and its output (640 x 240 pixels) is captured at 15 frames/sec synchronously with acceleration-gyro sensor output. The CCD camera's inner parameters are obtained in preliminary calibration. The CCD camera was moved by human hand, so we know only the rough trajectory of the camera motion.

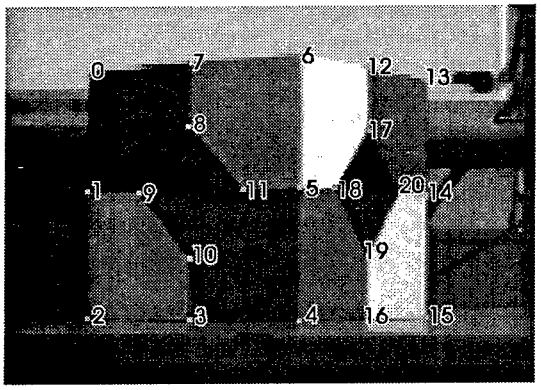


Figure 5: Photograph of the object with sides 20 cm long.

The interval between the two images for obtaining optical flow are automatically determined by a certain method, but we do not mention it for lack of space.

In order to examine the accuracy of recovery, we found the rotation, translation and scaling (if needed) from the recovered structure to the actual structure, because the recovered structures are related to a different coordinate system from that of the actual structure. We adopted the RMS (Root Mean Square) of the distances between actual and recovered points as the recovery error.

8.2 Experiment 1: Simple and Short Camera Motion

In experiment 1, the CCD camera was moved in short period almost straightly, as shown in Figure 6. The lengths mentioned in this figure are rough estimates as explained before.

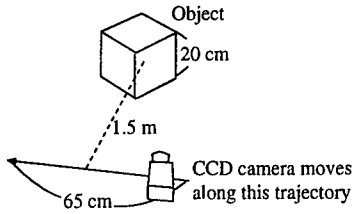


Figure 6: Camera motion in experiment 1.

We obtained 27 frames (in 1.68 sec) in the motion. When we do not use the acceleration-gyro sensor, optical flow for obtaining results must be large. In this case, only two structures were recovered and the average of the errors was 9.4 cm. When the acceleration-gyro sensor output was used, 13 structures were recovered and the average of the errors was 3.1 cm. The accuracy was much improved by using the acceleration-gyro sensor.

In Figure 7, errors of each of the recovered structures and the results of the integration of structures when the acceleration-gyro sensor output was used are plotted. The error of the integration result at index i is the result of integration from 0 to i. It is shown that the integration of recovered structures at different time points improves the accuracy.

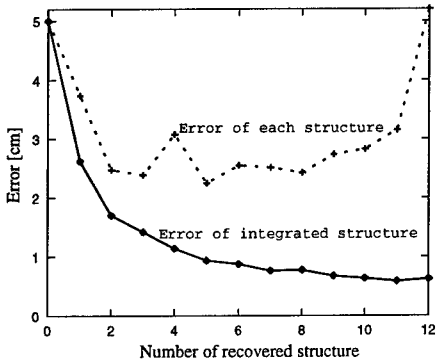


Figure 7: Errors of recovered structures using the acceleration-gyro sensor in experiment 1.

In Figure 8, the relation between $\|\boldsymbol{v}_I^W\|$ and

$\|v_G^W\|$ are plotted. The plotted points are on almost the same line through the origin, because the motion finished in short period, so that the drift of v_I^W was small. To determine s, (19) with $\alpha = 1$ was used. The results are shown in Table 1, where point numbers for specifying sides are shown in Figure 5.

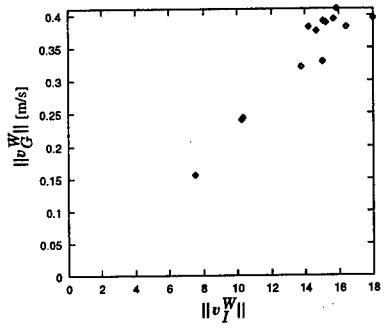


Figure 8: Velocities obtained from the image sequence and acceleration-gyro sensor output in experiment 1.

Table 1: Recovered length and actual length in experiment 1

Side	Recovered [cm]	Actual [cm]
0-1	82.7	100
1-2	91.9	100
2-3	94.2	100
3-4	80.8	100
6-12	90.0	100
12-13	82.9	100
7-8	45.2	50
1-9	58.0	50
5-18	44.5	50
12-17	49.0	50

8.3 Experiment 2: Complex Camera Motion

More complex motion in long period was adopted in experiment 2. In this experiment, the CCD camera moved around the object as shown in Figure 9. We obtained 94 frames (in 5.64 sec) in the motion.

When the acceleration-gyro sensor was not used, 6 structures were recovered as shown in Figure 10. In this case, the average of the errors before the integration of recovered structures was 3.2 cm.

In Figure 11, the results when the acceleration-gyro sensor output was used are plotted, where 20

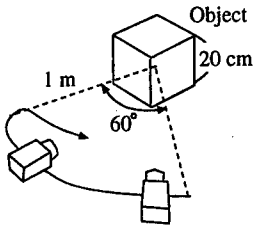


Figure 9: Camera motion in experiment 2.

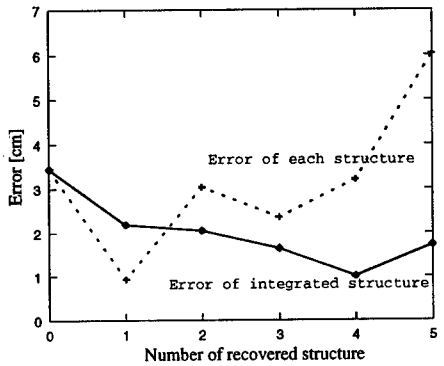


Figure 10: Errors of recovered positions without the acceleration-gyro sensor in the experiment 2.

structures were recovered. It is shown that the integration of the recovered structures at different time points improves the accuracy. In this case, the average of the errors before integration was 3.3 cm. It is a little worse than the case without the acceleration-gyro sensor, but we obtained the larger number of recovered structures, so that the integrated structure was better than the results without the acceleration-gyro sensor.

However, in this experiment where the camera motion is complex, we could not obtain reliable scale. A part of results is shown in Table 2. We need more study to improve the accuracy.

The recovered structure using the acceleration-gyro sensor output projected to new screen positions is shown in Figure 12. It is displayed using a wire frame or texture mapping. In the wire frame image, points are connected by lines in order to clearly show the structure.

9 Conclusion

We have proposed a method for shape and scale recovery using a CCD camera and an acceleration-

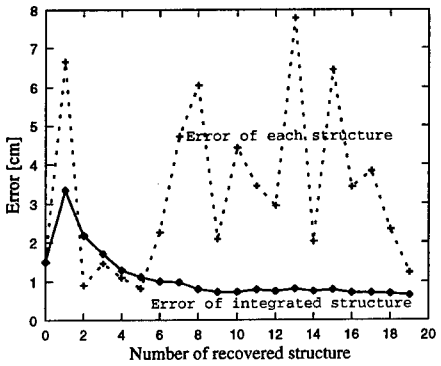


Figure 11: Errors of recovered positions using the acceleration-gyro sensor in the experiment 2.

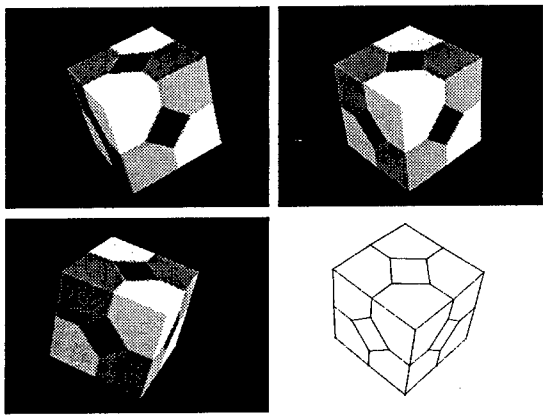


Figure 12: Recovered structure using the acceleration-gyro sensor in experiment 2.

gyro sensor. We modified the method proposed by us before, in order to use both the CCD camera and the acceleration-gyro sensor.

In the experiments, improvement of recovered structure is verified. However, recovered scales are not so reliable when camera motion is complex.

In the next step, we try to improve the accuracy of our method. In particular, improvement of scale recovery is necessary.

Table 2: Recovered length and actual length in experiment 2

Side	Recovered [cm]	Actual [cm]
0-1	60.1	100
1-2	64.6	100
5-18	31.9	50
12-17	30.4	50

References

[1] R. Y. Tsai and T. S. Huang. Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces. *IEEE Trans. Pattern Anal. Machine Intell.*, 6(1):13–27, 1984.

[2] T. S. Huang and A. N. Netravali. Motion and Structure from Feature Correspondences: A Review. *Proc. of the IEEE*, 82(2):252–268, 1994

[3] J. K. Aggarwal and C. H. Chien. 3-D Structure from 2-D Images. in *Advances in Machine Vision* (J. L. C. Sanz, Ed.), 64–121, 1989.

[4] T. Mukai and N. Ohnishi. Motion and Structure from Perspectively Projected Optical Flow by Solving Linear Simultaneous Equations. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'97)*, 740–745, 1997.

[5] C. Tomasi and T. Kanade. Shape and Motion from Image Streams under Orthography: a Factorization Method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[6] C. J. Poelman and T. Kanade. A Paraperspective Factorization Method for Shape and Motion Recovery. *IEEE Trans. Pattern Anal. Machine Intell.*, 19(3):206–218, 1997.

[7] O. Coenen and T. J. Sejnowski. A Dynamical Model of Context Dependencies for the Vestibulo-Ocular Reflex. in *Advances in Neural Information Processing Systems 8*, MIT Press, 1996.

[8] N. Franceschini, J.M. Pighon and C. Blanes. From insect vision to robot vision. *Phil. Trans. Roy. Soc. B*, 337:283–294, 1992.

Session TB3
Fusion for Target Tracking I
Chair: Mohamad Farooq
Royal Military College of Canada

Track Association and Track Fusion with Non-Deterministic Target Dynamics

Shozo Mori, William H. Barker
smori@raytheon.com, bbarker@raytheon.com
Raytheon Systems Company,
Advanced C^{3}I Systems
San José, CA

Chee-Yee Chong
chong_chee@bah.com
Booze-Allen & Hamilton, Inc.,
San Francisco, CA

Kuo-Chu Chang
kchang@gmu.edu
George Mason University,
Dept. of Systems Engineering,
Fairfax, VA

Abstract - This paper compares various track fusion algorithms and track association metrics, using a simple linear-Gaussian-Poisson model, to examine their performance under various degrees of non-deterministicitity of the target dynamics, i.e., process noises. Track fusion algorithms are compared by an analytical method while track association metrics are evaluated by Monte Carlo simulations.

Keywords: Distributed Data Fusion, Distributed Tracking, Track-to-Track Association, Track Fusion, Non-Deterministic Dynamics

1. Introduction

In many modern information gathering systems with multiple physically distributed sensors, a distributed data fusion architecture possesses several advantages over a centralized architecture. Among them is avoidance of data flow/processing bottle necks through well-designed data distribution and processing, without any significant loss of optimality achieved by a centralized architecture with infinite processing power. In the last three decades, distributed information processing algorithms have been one of the most studied areas in data processing. In the field of target tracking, according to [1], we can trace a pioneering work in distributed processing back to Singer's 1971 paper [2].

A general theory of distributed tracking was built ([4] – [7]) based on a general theory of distributed estimation described in [3] and on a general theory of multi-target tracking described in [9] (which generalizes the multi-hypothesis tracking algorithm developed by D. B. Reid [8]). Recently this general distributed tracking theory was also described in the random-set formalism ([10],[15]) using the general theory of multi-target tracking re-written in such formalism ([10] – [14]). This theory is general enough to be applicable to almost any kind of information flow pattern[1] and to a very wide class of target and sensor models. However, it was pointed out from the very beginning of its development that there is difficulty in applying this theory to non-deterministic target models.

[1] including dynamically changing information flows.

In a traditional sense of distributed tracking, track-to-track association and track fusion (distributed filtering) replaces measurement-to-track association and dynamic state estimation (i.e., filtering) in a single-site or centralized tracking. In the area of distributed filtering, there have been a large volume of papers and reports that describe various distributed filtering algorithms [16] – [25]. Also, in this traditional framework, it was recognized that effects of target model's non-deterministicity cause certain difficulties and several attempts have been made to study such effects and develop algorithms to alleviate such difficulties, as shown in [26] – [29]. In addition, efforts to expand the general framework mentioned before to cope with non-deterministic target dynamics were also made ([30]).

The objective of this paper is to explore issues related to non-deterministic target dynamics in search of effective distributed tracking algorithms. This paper characterizes and evaluates several representative algorithms brewed through almost three decades of development in distributed tracking as described above. For this purpose, we will take a traditional approach to distributed tracking, i.e., we will discuss track-to-track association (Section 3) and track (state-estimate) fusion (merge) (Section 2) separately. Furthermore, in order to make evaluation and comparison easy, we will restrict ourselves to a linear gaussian case with the simplest information network, i.e., two sensors with local processors and one central processor.

2. Track Fusion (Distributed Filtering) Algorithms

There is a wide class of distributed filtering problems, and even more ways to describe them. In this paper, however, we will consider the simplest case in which two tracks generated by two local data processing nodes are to be fused together.

2.1. Problem Statement: Let us assume that the state x_t at time t of a target is described by a linear stochastic differential equation, $dx_t = A_t x_t dt + B_t dw_t$ on an interval $[t_1, \infty)$. As

usual, the initial condition on x_{t_1} is given as a gaussian random vector independent of the unit-intensity standard Wiener process $(w_t)_{t=t_1}^{\infty}$. We assume that matrices A_t and B_t with compatible dimensions satisfy all the necessary regularity assumptions to guarantee unique existence of a solution $(x_t)_{t=t_1}^{\infty}$ to the linear stochastic equation. Assume two sensors observe the target states as $y_{ik} = H_{ik}x_{t_k} + \eta_{ik}$ at given time[2], $t_1 < t_2 < \cdots < t_n$, for each sensor i, with appropriate matrices H_{ik} and independent additive measurement noises η_{ik}, i.e., zero-mean gaussian vectors with positive definite[3] variance matrix R_{ik}.

Define a local target state estimate as any conditional mean $\hat{x}\left(t \middle| (y_{ij})_{j=1}^{k}\right)$ of target state x_t at time t given a cumulative local measurement $(y_{ij})_{j=1}^{k}$. Needless to say these estimates can be obtained by the two local (disjunctive) Kalman filters, and if necessary, by a smoother. For the sake of simplicity, let us write the local target estimate $\hat{x}\left(t_n \middle| (y_{ik})_{k=1}^{n}\right)$ after the last observation at t_n is processed as $\hat{x}_i$, and in place of x_{t_n}, we write the target state at time t_n simply as x. Then, in a wide sense, the track fusion problem is to come up with a "good" global estimate $\hat{x}$ of x from the two local tracks $(y_{1k})_{k=1}^{n}$ and $(y_{2k})_{k=1}^{n}$. In a narrow sense, the estimate $\hat{x}$ is restricted to be a function of only the two most recent state estimates, $\hat{x}_1$ and $\hat{x}_2$, calculated only from the two local tracks.

2.2. Bar-Shalom-Campo Fusion Algorithm: The Bar-Shamon-Campo algorithm as described in [27] is to calculate the global estimate $\hat{x}$ from the two local estimates, $\hat{x}_1$ and $\hat{x}_2$, as

$$\hat{x} = W_1\hat{x}_1 + W_2\hat{x}_2 \qquad (1)$$

where $W_i = (V_{jj} - V_{ji})(V_{11} + V_{22} - V_{12} - V_{21})^{-1}$ for $i \in \{1,2\}$ with $j = 3 - i$, with V_{ii} being the variance matrix of the estimate error $\hat{x}_i - x$ and V_{ij} being the covariance matrix between the estimation errors, $\hat{x}_i - x$ and $\hat{x}_j - x$.

This estimate can be viewed as a "convex combination" of[4] the two estimates, $\hat{x}_1$ and $\hat{x}_2$, since we have $W_1 + W_2 = I$. This estimate is a kind of "maximum likelihood" estimate in the following sense. Let $p(\cdot,\cdot)$ be the density function[5] of the joint distribution of the local estimation errors, $\hat{x}_1 - x$ and $\hat{x}_2 - x$, and let $\ell(x) = p(\hat{x}_1 - x, \hat{x}_2 - x)$ be the "likelihood function" for the target state x given the two local estimates, $\hat{x}_1$ and $\hat{x}_2$. As shown in [28], the estimate (1) maximizes this[6] "likelihood function."

2.3. Simple Convex Combination Fusion Algorithm: It is not clear to the authors to whom we should attribute this algorithm although it is used rather commonly probably because of its simplicity and relatively small amount of necessary data. This algorithm is also to calculate the global estimate $\hat{x}$ by the "convex combination," eqn. (1), but with a simplified weights $W_i = V_{jj}(V_{11} + V_{22})^{-1}$. In other words, we obtain this algorithm by ignoring the covariance matrices, V_{12} and V_{21}.

2.4 Maximum *A Posteriori* Probability Density Estimate: Since our model is linear-gaussian, the maximum *a posteriori* probability density estimate of the target state x given two local

[2] Simultaneous observation by the two sensors is not essential to the discussions in this paper but we assume it for the sake of simplicity.

[3] In this paper, positive definiteness is always mean to be in the strict sense.

[4] The convex combination appears in a pair of quotation marks since it is not in a usual sense because of the lack of positivity concept of the coefficient matrices W_i.

[5] Namely,
$p(\tilde{x}_1, \tilde{x}_2)d\tilde{x}_1 d\tilde{x}_2 = \text{Prob.}\{\hat{x}_1 - x \in d\tilde{x}_1, \hat{x}_2 - x \in d\tilde{x}_2\}$.

[6] The word likelihood appears within a pair of quotation marks since the "likelihood function" here is not a likelihood function in a usual sense, i.e., $p(\hat{x}_1, \hat{x}_2 | x)$.

estimates, $\hat{x}_1$ and $\hat{x}_2$, is the conditional mean of x given $\hat{x}_1$ and $\hat{x}_2$, which is also the minimal variance (linear) estimate, and can be written ([31]) as,

$$\hat{x} = \bar{x} + L_1(\hat{x}_1 - \bar{x}) + L_2(\hat{x}_2 - \bar{x}) \qquad (2)$$

where the gain matrix $L = [L_1 \quad L_2] = V_{x\hat{z}} V_{\hat{z}\hat{z}}^{-1}$ is calculated from the covariance matrix $V_{x\hat{z}}$ between the target state x and the joint local estimates $\hat{z} \overset{\text{def}}{=} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix}$ and the variance matrix $V_{\hat{z}\hat{z}} = \begin{bmatrix} \hat{V}_{11} & \hat{V}_{12} \\ \hat{V}_{21} & \hat{V}_{22} \end{bmatrix}$ of $\hat{z}$. $\bar{x}$ in (2) is the *a priori* mean of the target state x at time t_n.

2.5 Tracklet Fusion: It might be said that almost all the algorithms described in a vast volume of the distributed filtering literature [16] – [25] is either equivalent to or modification of the algorithm of obtaining the global estimate $\hat{x}$ from the local estimates $\hat{x}_1$ and $\hat{x}_2$ by

$$\hat{V}^{-1}\hat{x} = V_{11}^{-1}\hat{x}_1 + V_{22}^{-1}\hat{x}_2 - \bar{V}^{-1}\bar{x} \qquad (3)$$

with $\hat{V}^{-1} = V_{11}^{-1} + V_{22}^{-1} - \bar{V}^{-1}$. $\bar{x}$ is the *a priori* mean of the state x, and $\bar{V}$ is its variance matrix of x. Eqn. (3) can be rewritten as $\hat{V}^{-1}\hat{x} = (V_{11}^{-1}\hat{x}_1 - \bar{V}^{-1}\bar{x}) + (V_{22}^{-1}\hat{x}_2 - \bar{V}^{-1}\bar{x}) + \bar{V}^{-1}\bar{x}$, and the subtraction $V_{ii}^{-1}\hat{x}_i - \bar{V}^{-1}\bar{x}$ can be viewed as an operation to extract "new" information obtained by sensor i from the mixture with *a priori* information. For that reason, this algorithm is called *information differencing* in [38]. The so-called information-matrix form (3) can be rewritten in the variance-matrix form, which can be viewed as addition of information in terms of an equivalent measurement or pseudo measurement to the fusion agent as described in [20]. The terminology, *tracklet*, is attributed to [23] and [24], and is meant to be a "small" conditionally independent fragment of a track.

We should also note that eqn. (3) can be derived from a general fusion equation expressed as $\hat{p}(x) = C^{-1} p_1(x) p_2(x) / \bar{p}(x)$ to calculate the probability density $\hat{p}$ of the target state x conditioned by the information from both sensors, from the two local estimation results represented by the probability density functions, p_1 and p_2, and from the density $\bar{p}$ of the *a priori* target state distribution. As mentioned before, however, the derivation of this formula (and hence eqn. (3)) is based on the non-deterministicity assumption. Nonetheless, usually the non-deterministic dynamics are used for the time alignment extrapolation. It is generally understood, therefore, this algorithm works well only when either the non-deterministicity is small or the frequency of using the fusion rule (3) is high enough.

2.6. Comparison of Track Fusion Algorithms: A very simple example was chosen to compare the track fusion rules listed above, using a simple two-dimensional target tracking example. The target dynamics are defined by $A_t \equiv \begin{bmatrix} 0 & I \\ 0 & -\beta I \end{bmatrix}$ and $B_t \equiv \begin{bmatrix} 0 \\ qI \end{bmatrix}$ where I and 0 are the 2×2 identity and zero matrices, and β and q are two positive parameters. The initial condition is a zero-mean gaussian random vector with variance matrix Block Diag.$(\sigma_0^2 I, \sigma_v^2 I)$ with[7] $q = 2\beta\sigma_v^2$. We assume a supplementary (or redundant) sensor case assuming two independent but identical sensors with $H_{ik} \equiv [I \quad 0]$ and $R_{ik} \equiv \sigma_M^2 I$, and an identical sampling rate as $t_{k+1} - t_k \equiv \Delta t$ with[8] $n = 2$. The diagonal elements of the variance matrix of the stationary velocity process are set as $\sigma_v^2 = q/(2\beta) = (2\sigma_M)^2$. In order to compare the fusion rules, we vary the process noise intensity q and the standard deviation σ_0 of the initial condition[9]. Fig. 1 shows a result when the process noise intensity is varied.

In this figure, three fusion rules, (1) convex-combination, (2) MAP (maximum *a priori*

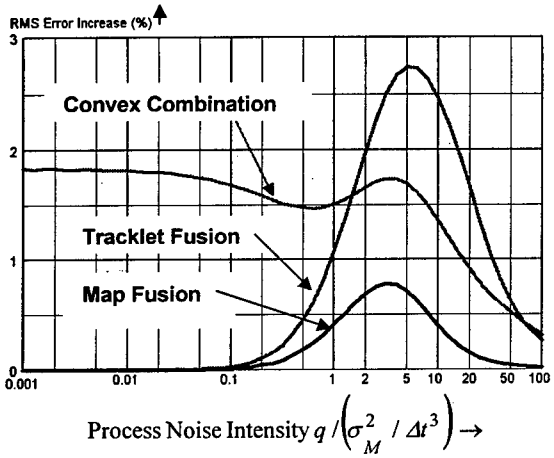
[7] In this target model, the velocity is modeled by a stationary stochastic process which is referred to as *Ornstein-Uhrenbeck model* [32]. The model is also called Singer model [33].

[8] Namely, each local sensor has only two measurements.

[9] It should be noted that the initial state contributes to the covariance V_{12} between the estimation errors of the two local estimates as well as the process noise.

probability) fusion, and (3) tracklet fusion, are compared. Because of the symmetry between the two sensors, the Bar-Shalom-Campo algorithm is identical to the simple convex combination algorithm, and they are simply referred to as the convex combination algorithm. In this example, performance measured by the total RMS errors, i.e., the square root of the trace of the state estimate error variance matrix, by any of the fusion rules, does not deviate much from the best performance, i.e., the performance achieved by the centralized processing. For this reason, performance is displayed as the percent increase in the RMS errors over the minimum RMS achieved by the centralized tracking.

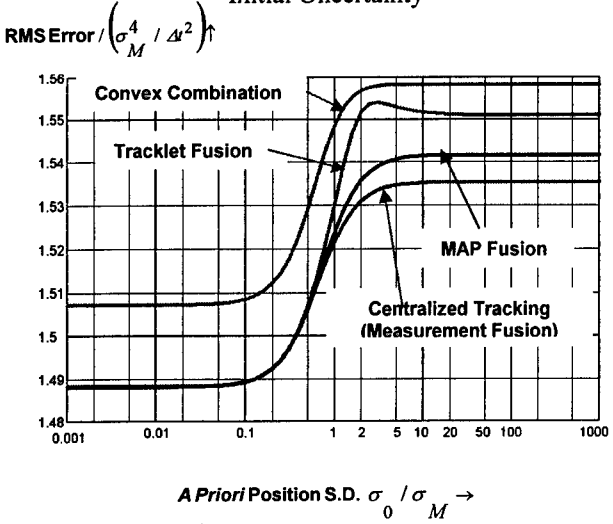
Fig. 1: Comparison of Fusion Rules with Varying Process Noise Intensity



It is interesting to see that the percentage increase becomes insignificant when the process noise is very small as well as when it is very large. The exception is the performance by the convex combination algorithm when the process noise is small. When the process noise is small, the system becomes almost deterministic, and the system uncertainty is dominated by the target's initial condition. The two local processing agents both use the *a priori* information. When the local estimates are fused together, this use of the *a priori* information might be double-counted. In the tracklet algorithm, as well as the MAP algorithm, this double counting is negated by subtraction, but not by the convex combination algorithm.

Fig. 2 shows performance of the same three fusion rules, compared with performance by a centralized tracking, when the *a priori* position variance is varied. The process noise intensity was set as $q/\left(\sigma_M^2/\Delta t^3\right)=1$. The convex combination algorithm seems to be consistently worse than other fusion schemes, which we may attribute to the "double counting" of the *a priori* information. Both tracklet fusion and MAP fusion schemes deviate from the optimal performance as the initial state uncertainty increases.

Fig. 2: Comparison of Fusion Rules with Varying Initial Uncertainty



3. Track-to-Track Association

We will restrict ourselves to the simplest form of track-to-track association problems, in which two sets of tracks are to be associated. We are interested in performance of various track association metrics. To simplify the problem, we assume that the two sets have exactly the same number of tracks and that the true association is always one-to-one. Under a certain set of assumptions, it can be shown that, given a two sets of n tracks, the track association problem can be defined as a problem of choosing a best hypothesis represented by a permutation a on the set $\{1,...,n\}$ to minimize the association cost,

$$J(a)=\sum_{i=1}^{n} C_{ia(i)} \tag{4}$$

where each $C_{ij}\in[0,\infty]$ is a given track association metric. $C_{ij}=\infty$ results from appropriate thresholding.

3.1. Problem Statement: We will model targets as a Poisson-Gaussian finite random set (point process) in the sense described in [10] – [15]. Namely, the number of targets is a Poisson random variable with a given mean, and given a number of targets, the target states are independent, and

identically distributed with a common gaussian distribution. As a common distribution, we use the linear-gaussian model described in Section 2.1. Given N targets, we assume that they are all detected and correctly correlated together by each local sensor data processing agent, without any mis-association or fragmentation, which means that each of the two sets of local tracks has a one-to-one correspondence to the set of targets. Using the linear-gaussian model described in Section 2.1., all the tracks from sensor 1 share the same variance matrix V_{11}, and so do those from sensor 2, the variance matrix V_{22}. Thus we can identify each track i from sensor 1 with the target state estimate $\hat{x}_{1i}$, and track j from sensor 2 with $\hat{x}_{2j}$. For any pair of track i from sensor 1 and track j from sensor 2, assuming that they originate from the same target, the covariance matrix between the two local target state estimates can be represented by a single matrix[10] $V_{12} = (V_{21})^T$.

3.2. Bar-Shalom Metric: This metric was derived to be used in a classical chi-square test in [26] and can be written as[11]

$$C_{ij} = \left\| \hat{x}_{1i} - \hat{x}_{2j} \right\|^2_{(V_{11}+V_{22}-V_{12}-V_{21})^{-1}} \qquad (5)$$

3.3. Mahalanobis Metric: This metric is probably the most frequently used and obtained by ignoring the covariance $V_{12} = (V_{21})^T$ in (5), i.e.,

$$C_{ij} = \left\| \hat{x}_{1i} - \hat{x}_{2j} \right\|^2_{(V_{11}+V_{22})^{-1}} \qquad (6)$$

which can be identified with the square of the Mahalanobis distance between two gaussian distributions, $(\hat{x}_{1i}, V_{11})$ and $(\hat{x}_{2j}, V_{22})$.

3.4. Chong Metric: This metric can be derived from a general form of track-to-track association likelihood $\int \left(p_{1i}(x) p_{2j}(x) / \bar{p}(x) \right) \mu(dx)$ between track i from sensor 1 having the target state distribution p_{1i} described as a density with respect to a certain measure μ on a target state space and track j from sensor 2 with p_{2j}, where $\bar{p}$ is the common *a priori* target distribution density. In our gaussian case, we have

$$C_{ij} = \left\| \hat{x} - \hat{x}_{1i} \right\|^2_{V_{11}^{-1}} + \left\| \hat{x} - \hat{x}_{2i} \right\|^2_{V_{22}^{-1}} - \left\| \hat{x} - \bar{x} \right\|^2_{\bar{V}^{-1}} \qquad (7)$$

where $\hat{V}^{-1}\hat{x} = V_{11}^{-1}\hat{x}_1 + V_{22}^{-1}\hat{x}_2 - \bar{V}^{-1}\bar{x}$ with $\hat{V}^{-1} = V_{11}^{-1} + V_{22}^{-1} - \bar{V}^{-1}$, and $(\bar{x}, \bar{V})$ is the *a priori* mean vector and the variance matrix of the target state x. The subtraction in (7) can be interpreted as an operation to eliminate the double-counted *a priori* information. It can be easily shown that, as $\bar{V} \to \infty$, Chong metric (7) converges to Mahalanobis metric (6).

3.5. Expanded State Metric: This metric can be obtained by first expanding target state space from[12] $\Re^d$ to $\Re^{dn}$ where d is the dimension of the original target state space, i.e., expanding the target state to be estimated from $x(t_n)$ to $(x(t_k))_{k=1}^n$, and then by applying the Chong metric (7) to the expanded target state estimates. This metric was suggested in [7] but fully explored in [30]. This expansion of target state space generally makes calculation directly using (7) impractical. However, in [30], it was shown that the metric can be obtained by recursively calculating the properly defined *track likelihood function* of each track to be fused, as well as the fused track, using the measurements.

3.6. Performance Comparison: Unlike the performance analysis of Section 2.6., there is no obvious way[13] of predicting the track association performance by various association metrics. Therefore, Monte Carlo analysis was conducted. In each run, a random set of targets with the average number 100 of targets was generated according to the model described in Section 2.6., with the initial position uncertainty standard deviation being ten times as big as the measurement error, i.e., $\sigma_0 = 10\sigma_M$.

---

[10] By X^T we mean the transpose of a vector or a matrix X.

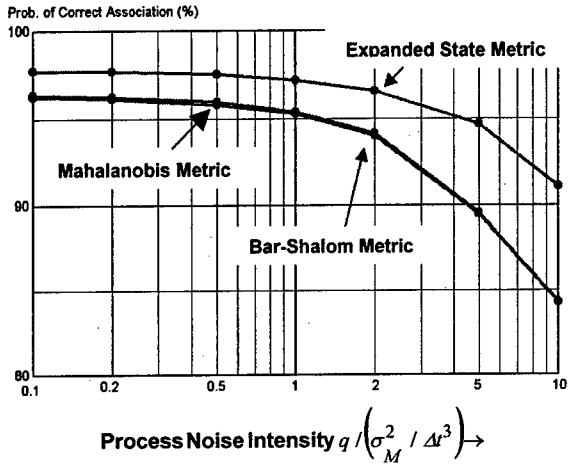
[11] $\|\cdot\|_A$ is a norm on a Euclidean space determined by a symmetric positive definite matrix A as $\|x\|_A = \sqrt{x^T A x}$.

[12] $\Re = (-\infty, \infty)$ = set of reals.

[13] In [34], a method for predicting data association performance is described but it may not be appropriate when the intensity measure of the random set (point process) is Gauss-Poisson, rather Uniform-Poisson.

Fig. 3 shows comparison of association performance by (1) Bar-Shalom metric, (2) Mahalanobis metric, and (3) expanded state metric, with various process noise intensity q. For each run, for each target, it was examined whether the tracks originating from that target are correctly associated or not. Then the probability of correct association is calculated as the number of correctly associated targets over the total number of targets. Each point in Fig. 3 was obtained by averaging 300 samples.

Fig. 3: Track Association Performance



Since we have chosen relatively large initial state variance, we expected the difference between Chong metric and Mahalanobis metric to be very small. Consequently, only the simper algorithm, i.e., Mahalanobis metric, was evaluated. Because of its completeness and amount of required computation, better performance by the expanded state metric is not surprising at all. However, we should note that there is no significant difference between Bar-Shalom metric considering the covariance between the two tracks and Mahalanobis metric that ignores such cross-correlation. Lack of difference between two algorithms may be explained as follows: Bar-Shalom metric uses the covariance matrix to adjust the weights in the distance function (5); in a sense, decreasing with positive correlation and increasing with negative correlation. However, in our example, because we assume tracks with uniform quality, those differences become the same to all the track pairs, and between two different metrics, the effects appear only as a scaling difference. This scaling effect resulted in an extremely small difference caused by different relative scaling with respect to the fixed threshold (i.e., the chi-square value of 30).

4. Conclusion

Using very simple linear-Gaussian-Poisson models, several different track fusion algorithms and track association metrics were evaluated. Because of linear-gaussian-ness, track fusion algorithms can be compared analytically. Track association performance, however, had to be measured by Monte Carlo simulations.

Generally computational and communication requirements were not considered. Considering these requirements, in practical cases, those track fusion algorithms requiring evaluation of cross-correlation between tracks form two sensors may not be very practical. In many cases, simply communicating all the measurements in a track with all the necessary statistics including observable partials, etc., would be more practical. Under a reasonable communication restriction, tracklet fusion would be more practical. Using an appropriate time intervals between two tracklets from local sites, we may treat each tracklet as equivalent measurements from which the expanded track association metric can be easily calculated.

This approach has been used for a group of tactical data fusion products called Advanced Tactical Workstation (ATW [35]) developed by Advanced C^3I Systems Unit of Raytheon Systems Company. As described in [30], in a particular variant of ATW specially tuned to undersea bearing-only tracking, the expanded state track association metric is used very effectively. Recently, it was also proved, in an ONR-sponsored program, on-board sensor fusion may be effectively architectured using this combination of tracklet fusion and expanded state association metric ([36],[37]).

REFERENCES

[1] Y. Bar-Shalom, and T. E. Fortman, *Tracking and Data Association*, Academic Press, Orland, FL 1988.

[2] R. A. Singer, and A.J. Kanyuck, "Computer Control of Multiple Site Correlation," *Automatica*, Vol. 7, pp. 455-463, July 1971.

[3] C. Y. Chong, "Hierarchical Estimation," *Proc. MIT/ONR Workshop on C^3*, 1979.

[4] C. Y. Chong, and S. Mori, "Hierarchical Multi-Target Tracking and Classification – Bayesian Approach," *Proc. 1984 American Control Conf.*, San Diego, CA, June 1984.

[5] C. Y. Chong, E. Tse, and S. Mori, "Distributed Estimation in Network," *Proc. 1983 American Control Conf.*, San Francisco, CA, June 1983.

[6] C. Y. Chong, S. Mori, and K. C. Chang, "Information Fusion in Distributed Sensor Networks," in *Proc. 1985 American Control Conf.*, pp. 830 - 835, 1985.

[7] C. Y. Chong, S. Mori, and K. C. Chang, "Distributed Multitarget Multisensor Tracking," in *Multitarget Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom, ed., Chap. 8, Norwood, MA: Artech House, 1990, pp. 247 - 295.
[8] D. B. Reid, "An Algorithm for Tracking Multiple Targets," *IEEE Trans. Autmat. Contr.*, AC-24, Dec. 1979.
[9] S. Mori, C. Y. Chong, E. Tse, and R. P. Wishner, "Tracking and Classifying Multiple Targets wihout *A Priori* Identification," *IEEE Trans. Automat. Contr.*, AC-31, May 1998.
[10] S. Mori, "Random Sets in Data Fusion – Multi-Object, State-Estimation as a Foundation of Data Fusion Theory," in *Random Sets: Theory and Applications*, ed. By J. Goutsias, R.P.S. Mahler, and H.T. Nguyen, Springer-Verlag, NY, 1997.
[11] S. Mori, "Random Sets in Data Fusion Problems," *Proceedings of 1997 National Symposium on Sensor and Data Fusion,* MIT Lincoln Laboratory, Lexington, MA, April 1997.
[12] S. Mori, "Random Sets in Data Fusion Problems," *Proceedings of 1997 National Symposium on Sensor and Data Fusion,* MIT Lincoln Laboratory, Lexington, MA, April 1997.
[13] S. Mori, "Random Sets in Data Fusion," *Proc. of SPIE Symposium on Data Processing and Tracking of Small Targets,* San Diego, CA, August 1997.
[14] S. Mori, "Multi-Target tracking Theory in Random Set Formalism," *Proc. First International Conference on Multisource-Multisensor Information Fusion*, pp. 116 – 123, Las Vegas, NV, July 1998.
[15] S. Mori, "A Theory of Information Exchanges: Random-Set Formalism," *Proc. 1998 IRIS National Symposium on Sensor and Data Fusion*, Marietta, GA, April 1998.
[16] J. L. Speyer, "Computation and Transmission Requirements for a Decentralized Linear-Quadratic-Gaussian Control Problem," *IEEE Trans. Automat. Contr.*, Vol. AC-24, pp. 266 - 269, April 1979.
[17] A. Willsky, M. Bello, D. Castanon, B. Levy, and G. Verghese, "Combining and Updating of Local Estimates Along Sets of One-Dimensional Tracks," *IEEE Trans. on Automat. Contr.,* Vol. AC-27, pp. 799-813, Aug. 1982.
[18] H. R. Hashemipour, S. Roy, and A. J. Laub, "Decentralized Structures for Parallel Kalman Filtering," *IEEE Trans. Automat. Contr.,* Vol. AC-33, No. 1, pp. 88 - 93, Jan. 1988.
[19] H. F. Durrant-Whyte, B. S. Y. Rao, and H. Hu, "Toward a Fully Decentralized Architecture for Multi-Sensor Data Fusion," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1990.
[20] B. Belkin, S. L. Anderson, and K. M. Sommar, "The Pseudo-Measurement Approach to Track-To-Track Data Fusion," in *Proc. 1993 Joint Service Data Fusion Symposium*, pp. 519 - 538, 1993.
[21] R. Lobbia and M. Kent, "Data Fusion of Decentralized Tracker Outputs," *IEEE Trans. Aero. Elect. Syst.,* vol. AES-30, vo. 3, pp. 787-799, July 1994.
[22] O. E. Drummond, "Feedback in Track Fusion without Process Noise," in *Proc. SPIE, Signal and Data Processing of Small Targets*, vol. 2561, pp. 369 - 383, 1995.
[23] O. E. Drummond, "A Hybrid Sensor Fusion Algorithm Architecture and Tracklets," in *Proc. SPIE, Signal and Data Processing of Small Targets*, vol. 3163, 1997.
[24] O. E. Drummond, "Tracklets and a Hybrid Fusion with Process Noise," in *Proc. SPIE, Signal and Data Processing of Small Targets*, vol. 3163, 1997.
[25] M. D. Miller, O. E. Drummond, and A. J. Perrella, "Tracklets and Covariance Truncation Options for Theater Missile Tracking," in *Proc. 1998 International Conf. on Multisource-Multisensor Data Fusion (FUSION'98)*, 1998.
[26] Y. Bar-Shalom, "On the Track-to-Track Correlation Problem," *IEEE Trans. Automatw. Contr.*, AC-25, pp. 802-807, Aug., 1980.
[27] Y. Bar-Shalom, and L. Campo, "The Effects of the Common Process Noise on the Two-Sensor Fused-Track Covariance," *IEEE Trans. on Aero. and Elect. Syst.,* Vol. AES-22, No. 6, pp. 803-804, 1986.
[28] K. C. Chang, R. K. Saha, and Y. Bar-Shalom, "On Optimal track-to-Track Fusion," *IEEE Trans. on Aero. and Elect. Syst.,* Vol. 33, No. 4, pp. 1271-1276, Oct., 1997.
[29] K. C. Chang, Z. Tian, and R. K. Saha, "Performance Evaluation of Track Fusion with Information Filter," *Proc. First International Conference on Multisource-Multisensor Information Fusion*, pp. 648 – 654, Las Vegas, NV, July 1998.
[30] S. Mori, K. A. Demetri, W. H. Barker, and R. N. Lineback, "A Theoretical Foundation of Data Fusion – Generic Track Association Metric -," *Technical Proceedings of Seventh Joint Service Data Fusion Symposium*, pp. 585-594, Laurel, MD, Oct. 1994.
[31] I. B. Rhodes, "A Tutorial Introduction to Estimation and Filtering," *IEEE Trans. Automat. Contr.,* AC-16, No. 6, pp. 688-706, 1971.
[32] Tiburon Systems, "Over-The-Horizon Detection, Classification and Tracking (OTH/DC&T) System Level Specification Ship Tracking Algorithm," Rev. C, TIB-01411, June, 1991.
[33] S..S. Blackman, *Multiple Target Tracking with Radar Application*, Artech House, Norwood, MA, 1986.
[34] S. Mori, K. C. Chang, and C. Y. Chong, "Performance Analysis of Optimal Data Association

with Applications to Multiple Target Tracking," in *Multitarget-Multisensor Tracking: Applications and Advances, Vol. II*, ed. by Y. Bar-Shalom, Artech House, 1992.
[35] K. A. Demetri, W. H. Barker, S. Mori, and R. N. Lineback, "Advanced Tactical Workstation," *Proceedings of 1997 National Symposium on Sensor and Data Fusion,* MIT Lincoln Laboratory, Lexington, MA, April 1997.
[36] W. H. Barker, S. Mori, E. G. Sullinger, and M. P. Boe, "Data Fusion Processing for the Multi-Spectral Sensor Surveillance System (M4S)," *Proc. 1998 IRIS National Symposium on Sensor and Data Fusion*, Marietta, GA, April 1998.
[37] W. H. Barker, S. Mori, E. G. Sullinger, and M. P. Boe, "Data Fusion of the Multi-Spectral Sensor Surveillance System (M4S)," *Proc. Third NATO/IRIS Joint Symposium*, Quebec City, Quebec, Canada, Oct., 1998.
[38] J. M. Covino, and B. E. Griffiths, "A New Estimation Architecture for Muti-Sensor Data Fusion," SPIE Vol. 1478, Sensor Systems for Guidance and Navigation, pp. 114-125, 1991.

Architectures and Algorithms for Track Association and Fusion

Chee-Yee Chong
Booz Allen & Hamilton, Inc.
San Francisco, CA 94111

Kuo-Chu Chang
George Mason University
Dept. of System Engineering
Fairfax, VA 22030

Shozo Mori

William H. Barker

Raytheon Systems Company
Advanced C^3I Systems
San Jose, CA 95126

Abstract – *Target tracking using multiple sensors can provide better performance than using a single sensor. One approach to multiple target tracking with multiple sensors is to first perform single sensor tracking and then fuse the tracks from the different sensors. Two processing architectures for track fusion are presented: sensor to sensor track fusion, and sensor to system track fusion. Technical issues related to the statistical correlation between track estimation errors are discussed. Approaches for associating the tracks and combining the track state estimates of associated tracks that account for this correlation are described and compared by both theoretical analysis and Monte Carlo simulations.*

Key Words: Sensor fusion, target tracking, distributed tracking/fusion, distributed data processing

1. Introduction

The use of multiple sensors for target tracking can potentially provide better performance than a single sensor due to better visibility, complementary information, etc. Theoretically, the best tracking performance is achieved by fusing the measurements from the sensors directly. However, due to communication or organization constraints, many real-world systems have a hierarchical structure where the fusion system has no direct access to the sensor data. Instead, the sensor data are processed locally to form sensor tracks, which are then fused to form system tracks. Track fusion is then needed to associate the sensor tracks and generate an improved target state estimate.

Track fusion has technical issues that are not present in measurement fusion or centralized tracking. In particular, the state estimates of sensor tracks cannot be treated like sensor measurements and fused with a standard centralized tracking algorithm. This is due to the fact that while sensor measurement errors are usually independent across sensors and time, the errors in target state estimates associated with the tracks, i.e., tracker outputs, are generally correlated with one another. This has significant impact on the two main functions in track fusion: association and state estimate fusion. Both the computation of the association metrics and the fusion of the track state estimates need to consider any possible dependence between the track state estimation errors. The specific fusion architecture affects the nature of the statistical correlation and the algorithms that should be used.

This paper presents technical issues associated with track fusion and compares several algorithms. We first describe the track fusion problem and possible fusion architectures. This is followed by the technical issues associated with track fusion. Algorithms for track state estimate fusion and track association are then presented and compared. Specifically, we describe several algorithms for combining track state estimates, their optimality, and their advantages and disadvantages. Theoretical analysis and Monte Carlo simulations are used to compare their performance. We also describe different ways of handling the correlation between the target state estimates in computing the association metrics.

This paper focuses on deterministic target dynamics. A companion paper [1] will address the non-deterministic problem.

2. Track Fusion and Architectures

We consider a track fusion system with the components shown in Figure 1.

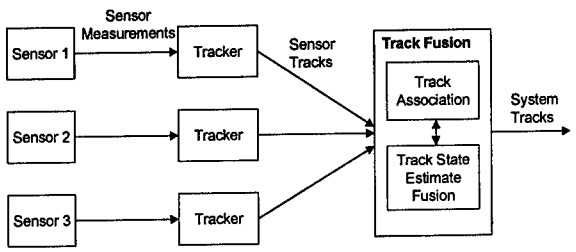


Figure 1: Track Fusion System

Single sensor tracking generates single sensor tracks and target state estimates. Periodically, the tracks from different sensors are sent to a central site to be fused.

Track fusion involves two steps: association and track state estimation fusion. In association, tracks from different sensors are associated to form system tracks, each corresponding to a single hypothesized target. Given an association, the state estimates of the system tracks are then obtained by fusing the state estimates of associated sensor tracks.

There are two possible processing architectures for track fusion depending on whether the state estimates of the system track are used.

Sensor to sensor track fusion (Figure 2). The state estimates from different sensor tracks (propagated to a common time) are associated and fused with each other to obtain the state estimate for the system track. The previous state estimate of the system track is not used in this process. Note that with this architecture, fusion will in general involve sets of tracks from more than two sensors.

This architecture does not have to deal with the problem of correlated estimation errors (if the common prior is ignored). Since it is basically a memory-less operation, the errors in association and track estimate fusion are not propagated from one time to the next. However, this approach may not be as efficient as sensor to system track fusion since past processing results are discarded.

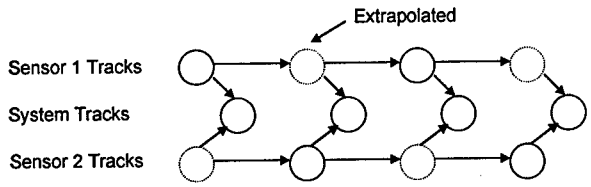


Figure 2: Sensor to Sensor Track Fusion

Sensor to system track fusion (Figure 3). Whenever a set of sensor tracks is received, the state estimates of the system tracks are extrapolated to the time of the sensor tracks, and fused with the newly received sensor tracks. The process is repeated when another set of sensor tracks is received.

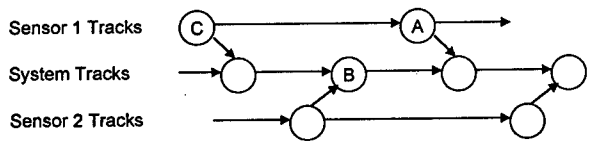


Figure 3: Sensor to System Track Fusion

Sensor to system track fusion reduces the association problem to a bi-partite assignment problem so that common assignment algorithms can be used. However, it has to deal with the problem of correlated estimation errors. In Figure 3, the sensor tracks in A and the system tracks in B have correlated errors since they all depend on C. Furthermore, any errors in system tracks due to past processing errors in association or fusion will affect future fusion performance.

3. Technical Issues

The main technical issues with track fusion are due to the fact that tracks and not measurements have to be fused. The inputs to track fusion are sensor tracks formed from local measurements and represented by position and velocity estimates and their error covariance matrices.

3.1. Correlated Estimation Errors

Fusion will be relatively straightforward if the estimation errors of the two tracks to be fused are uncorrelated. The estimates can be viewed as measurements with independent errors, fused with other estimates using a standard approach (e.g., association and Kalman filter update). The estimation errors between two tracks may be correlated for the following reasons.

A. Common prior estimates. This occurs in sensor to system track fusion as in Figure 4, which shows an information graph formulation [2], [3] of the track fusion problem. The solid squares in the graph represent measurements and the hollow squares represent fusion, either of a measurement with a track or a track with another track. The tracks are assumed to have been propagated to a common time. The placement of the tracks in the graph represents information contained in the tracks. Basically a track at a node will contain all the information in the predecessor nodes (both tracks and measurements). In the example, both the sensor track estimate $\hat{x}_j$ and the system track estimate $\hat{x}_i$ contain the sensor track estimate $\bar{x}_j$ propagated from an earlier time. Figure 4 also illustrates that two sensor tracks do not share common prior estimates (except for a common prior). In general, there is correlation from this source if there are multiple paths from a measurement to the fusion node in the information graph.

B. Correlated estimation errors due to common process noise. This occurs even when fusion is between sensor tracks not sharing common measurements. The measurements from two sensor tracks are not necessarily conditionally independent given the target state at a single time when the target dynamics is not deterministic. Thus the estimation errors from two sensor tracks may not be independent.

The correlated estimation errors have to be considered in associating the tracks and in combining the state estimates for the associated tracks. Otherwise, the target state estimates in the system tracks may degrade.

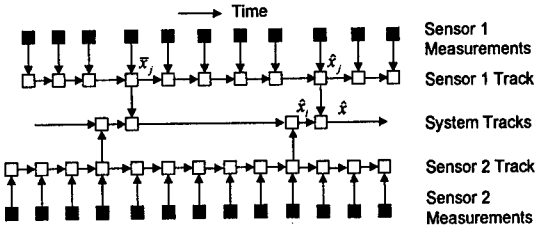


Figure 4: Dependence in Track Estimates

3.2. Imperfect Association

Track fusion has to associate those sensor tracks that originate from the same targets. If the sensor tracks were pure, i.e., each sensor track consists of measurements from a single target, or the previous associations were perfect, i.e., the sensor tracks that were associated were indeed from the same targets, then track association would need only deal with new sensor tracks. At any time in the track fusion process, sensor tracks that have been previously associated should keep their previous associations. Only new sensor tracks have to be associated to determine whether they are from new targets or from previously detected targets (associated with old sensor tracks).

In practice, sensor tracks are seldom perfect. They may be impure, i.e.,. each sensor track may consist of measurements from different targets (mis-association), or it may be fragmented, i.e., the same target may appear in multiple sensor tracks. Furthermore, previous track associations may be incorrect in that some sensor tracks may have been mis-associated with other sensor tracks. Thus it may be necessary to re-associate sensor tracks even though they have been previously associated to system tracks. If the sensor to system track fusion processing architecture is used, the computation of the association metrics has to consider the dependence between the sensor and system tracks.

4. Track State Estimate Fusion

Track fusion consists of two steps: (1) track-to-track association, or selection of a best association hypothesis, and (2) fusion of target state estimates given an association hypothesis. We will discuss track state estimates fusion first because the same techniques can be used for computing the association metrics.

Track State Estimate Fusion problem. Suppose there are two tracks, i and j with state estimates and error covariance matrices (both propagated to a common time) $\hat{x}_i$ and $\hat{x}_j$, P_i and P_j, respectively. The estimate fusion problem is to find the best fused estimate $\hat{x}$ and the error covariance matrix P. The two tracks may be two sensor tracks in a sensor to sensor track fusion architecture, or a system track and a sensor track in a sensor to system track fusion architecture.

Track state estimate fusion algorithms have been investigated extensively over the past two decades with most of the research performed under the topic of decentralized or distributed estimation. In the following sub-sections, we discuss two approaches to track fusion: "best" linear combination of track estimates and reconstruction of optimal centralized estimate.

4.1. "Best" Linear Combination of Estimates

The fused estimate is constrained to be a linear combination of the track estimates, i.e., $\hat{x} = A\hat{x}_i + B\hat{x}_j$. The matrices A and B are then chosen to optimize some criteria, e.g., weighted least squares or minimum variance. If the track estimates are not the sufficient statistics for the sensor measurements in the fused track, then the optimal linear combination may not be as optimal as an estimate that is allowed to use information other than the current estimates. Two algorithms have been developed for linearly combining the track estimates depending on whether the cross covariance between the track estimates is considered.

4.1.1 Basic Convex Combination

When the cross covariance between the two track estimates can be ignored, the fusion algorithm is given by [4]:

- State estimate:

$$\begin{aligned}\hat{x} &= P_j(P_i + P_j)^{-1}\hat{x}_i + P_i(P_i + P_j)^{-1}\hat{x}_j \\ &= P(P_i^{-1}\hat{x}_i + P_j^{-1}\hat{x}_j)\end{aligned} \quad (1)$$

- Error covariance:

$$P = P_i - P_i(P_i + P_j)^{-1}P_i = P_i(P_i + P_j)^{-1}P_j = (P_i^{-1} + P_j^{-1})^{-1} \quad (2)$$

The basic convex combination algorithm has been used extensively because of its simple implementation. It is suboptimal when the estimation errors are correlated, such as when one track is a system track and the other track is a sensor track, or when process noise is present. However, when both tracks are sensor tracks and there is no process noise, then the fusion algorithm is (almost) optimal, i.e., it produces almost the same results as when the sensor measurements are fused directly (as will be seen in Section 4.2.4.)

4.1.2 Linear Combination with Cross Covariance

When the cross covariance between the two estimates cannot be ignored, the linear combination algorithm becomes [5]:

- State estimate:

$$\hat{x} = \hat{x}_i + (P_i - P_{ij})(P_i + P_j - P_{ij} - P_{ji})^{-1}(\hat{x}_j - \hat{x}_i) \quad (3)$$

- Error covariance:

$$P = P_j + (P_i - P_{ij})(P_i + P_j - P_{ij} - P_{ji})^{-1}(P_j - P_i) \quad (4)$$

The cross-covariance's P_{ij} and P_{ji} are computed from the observation matrices and the Kalman filter gains [6].

This fusion algorithm was originally developed to account for correlation due to common process noise. However, the derivation depends only the correlation between the two estimation errors and not on the specific source of the errors, which could result from common prior estimates. The algorithm was originally thought to the optimal in the minimum mean square error (MMSE) sense but was claimed recently [7] to be only optimal in the maximum likelihood (ML) sense.

The advantage of this algorithm is its ability to handle common process noise. For example, when the estimates come from two sensor tracks, even if there is no common prior, the estimation errors may still correlated if there is common process noise. The main disadvantage of this algorithm is the amount of information needed to compute the cross-covariance. If the system is linear and time-invariant, the cross covariance can be computed from off-line information. Otherwise, the entire history of Kalman filter gains and observation matrices need to be communicated to whoever is doing the fusion. Since extended Kalman filters are usually used for sensor level tracking, the Kalman filter gains depend on the measurement data. In this case, it may be more efficient to communicate the measurements for centralized fusion than sensor tracks for track fusion.

4.2. Reconstruction of Centralized Estimate

The other basic approach to track state estimate fusion attempts to reconstruct the optimal estimate (by fusing the measurements in the tracks) from the individual track estimates and possibly some additional information [8] – [23]. Figure 5 illustrates the philosophy behind this approach for sensor to system track fusion. The sensor measurements from the individual sensors are used to form estimates for the sensor tracks. Periodically, these estimates are fused to obtain state estimates for the system tracks. As seen in the figure, the sensor track estimate $\hat{x}_j$ and system track estimate $\hat{x}_i$ to be fused share the same measurements in the sensor track estimate $\bar{x}_j$ which was transmitted earlier to fused with the system track.

The algorithms in this approach avoid double counting of information by either recognizing the common information and removing it in the fusion process or by only sending information that is uncorrelated with the system track. The latter uses the concept of so called "tracklets" [17] – [20], where a tracklet is loosely defined as a track segment computed so that its errors are not cross-correlated with the errors of other track segments.

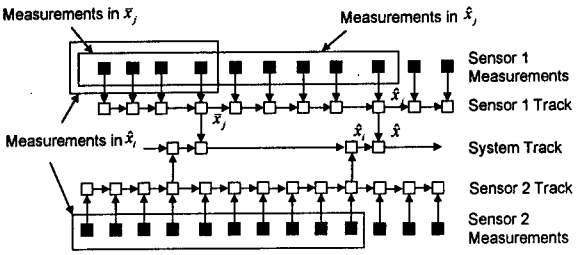


Figure 5: Reconstruction of Centralized Estimate

4.2.1 Information De-correlation

The information de-correlation approach can be derived [9], [11], [16] easily using the information filter form of the Kalman filter. The key idea is to identify the common information in the two estimates to be fused and remove it in fusion. This approach is useful when one track is the system track and the other track is the sensor track.

The state estimate fusion algorithm is given by:

- State estimate:

$$\hat{x} = P(P_i^{-1}\hat{x}_i + P_j^{-1}\hat{x}_j - \bar{P}_j^{-1}\bar{x}_j) \quad (5)$$

- Error covariance:

$$P = (P_i^{-1} + P_j^{-1} - \bar{P}_j^{-1})^{-1} \quad (6)$$

where $\bar{x}_j$ and $\bar{P}_j$ are the state estimate and error covariance (propagated to the common fusion time) of the sensor track last communicated to the system track. This is the additional information that is used for the fusion algorithm. Basically, both the sensor and system tracks contain this common information. In order to avoid double counting, it has to be removed from the results.

This fusion algorithm is based upon a general theory for distributed fusion [2], [3], [16] that can support arbitrary fusion and communication architectures, e.g., fusion with feedback. The information graph introduced earlier is used to identify the common information shared by two estimates, and the fusion algorithm then avoids the double counting. In addition to fusing state estimates and their error covariances, the general approach can also be used for fusing target classification probabilities.

The main advantage of this approach is its simple implementation. No additional communication is needed since the state estimate and error covariance of the previously transmitted sensor track can be stored and propagated to the current fusion time. This ap-

proach is optimal when there is no process noise. When the process noise is small, and/or the update rate by sensor tracks is reasonably high, the degradation in performance has been shown to be small.

4.2.2 Equivalent Measurement

This algorithm de-correlates the sensor track by finding an equivalent measurement for the "tracklet", i.e., the sensor measurements in the sensor track since the last communication with the system track [21]-[23].

The equivalent measurement is generated from the current estimate and error covariance and the previous estimate and error covariance of the sensor track (all propagated to a common time) as follows:

- Equivalent measurement

$$u_j = \bar{x}_j + \bar{P}_j(\bar{P}_j - P_j)^{-1}(\hat{x}_j - \bar{x}_j) \qquad (7)$$

- Error covariance

$$U_j = \bar{P}_j(\bar{P}_j - P_j)^{-1}P_j \qquad (8)$$

The error of the equivalent measurement is conditionally uncorrelated with the estimation error of the global track. Thus, the standard Kalman update equation can be used to combine the equivalent measurement with the current state estimate. More specifically, the update equation is:

- State estimate:

$$\hat{x} = \hat{x}_i + \hat{P}_i(\hat{P}_i + U_j)^{-1}(u_j - \hat{x}_i) \qquad (9)$$

- Error covariance:

$$P = \hat{P}_i - \hat{P}_i(\hat{P}_i + U_j)^{-1}\hat{P}_i \qquad (10)$$

Equations (9) and (10) with (7) and (8) are completely equivalent to (5) and (6) in Section 4.2.1 Equations (5) and (6) are in the information matrix form while (7) – (10) are in the covariance matrix form. Hence, numerical calculation issues aside, the performance and behavior of the two algorithms should be exactly the same. The information de-correlation algorithm has the advantage that the same approach can be used for general non-Gaussian or discrete probability distributions.

4.2.3 Restarting Local Filters

Another way of de-correlating the sensor track from the system track is to generate local track estimates using only the measurements since the last communication. This approach is sometimes called "tracklets from measurements" [18]. As seen in Figure.6, the local and system tracks are de-correlated since they do not share any common information. In terms of the information graph, there is a single unique path from each measurement to the fusion node.

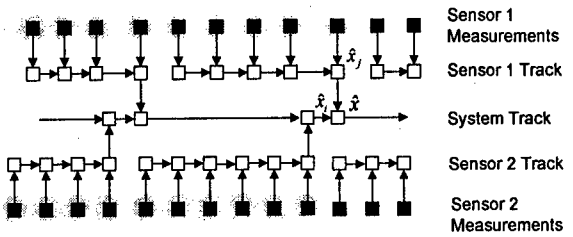


Figure 6: Restarting Local Filters

In this approach, after a sensor track has been transmitted to be fused with the system track, the local filter is re-started using the new measurements. The estimate from these measurements is then uncorrelated with the system track and can be fused easily with the system track. Since sensor tracking needs good estimates to evaluate the matrices for the extended Kalman filter and to support association, the usual tracker that processes all measurements is also used. Thus the local filter that is restarted periodically can be viewed as the "shadow tracker" [19].

The advantage of this approach is its simplicity. The disadvantage is the need to modify the existing tracking algorithm for the sensors. This approach is also equivalent to Equations (5) – (10).

4.2.4 Global Restart

The main problem in track fusion is the correlation between the sensor and system tracks. The correlation problem does not exist if the state estimate of the system track is not used in fusing the sensor track estimates. Since the sensor tracks already contain all the available measurements up to the current time, this global estimate formed will be optimal (Figure 7).

At each fusion time, the estimates of the sensor tracks are fused with each other to obtain the global estimate. The fusion algorithm is given by:

- State estimate:

$$\hat{x} = P(P_i^{-1}\hat{x}_i + P_j^{-1}\hat{x}_j - \bar{P}^{-1}\bar{x}) \qquad (11)$$

- Error Covariance:

$$P = (P_i^{-1} + P_j^{-1} - \bar{P}^{-1})^{-1} \qquad (12)$$

where $\bar{x}$ and $\bar{P}$ are the common prior state and covariance used by the sensor trackers propagated to the current time. Note that even though (11) and (12) are the same as (5) and (6), they are based on different processing architectures and use different priors.

When the prior covariance matrix $\bar{P}$ is much larger than the updated estimation error covariance matrix, or when it becomes very large due to forward propagation, then $\bar{P}^{-1} \to 0$, and these equations are equivalent to the basic convex combination equations (1) and (2) described before, i.e.,

$$\hat{x} = P(P_i^{-1}\hat{x}_i + P_j^{-1}\hat{x}_j) \quad (13)$$

$$P = (P_i^{-1} + P_j^{-1})^{-1} \quad (14)$$

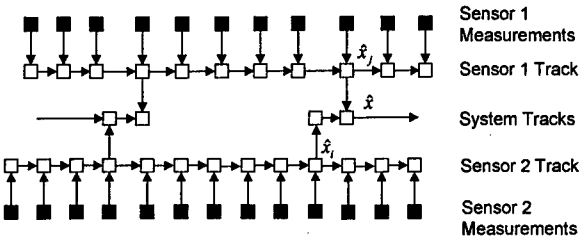


Figure 7: Global Restart

The advantage of this approach is that it does not have to perform any de-correlation since the sensor tracks to be fused do not have correlated estimation errors.

4.3. Numerical Results

We present some numerical results to compare the performance of the fusion algorithms.

4.3.1 Simulation Approach

Two sensors were located such that there is overlapping coverage of the target trajectory and the viewing geometry offers potential performance improvement from fusion. Measurements from the two sensors were then simulated.

Single sensor tracking algorithms were simulated by standard extended Kalman filters. The outputs from the two single sensor trackers were periodically fused using the algorithms to be analyzed.

The estimated target states (position and velocity) were compared with the true target states to find the estimation error. The errors from multiple Monte Carlo runs were averaged to find the root mean square (RMS) position and velocity errors. The parameter to be varied is the communication or fusion frequency.

We compared the performance of two fusion algorithms representing the two main approaches – convex combination and tracklet by means of information decorrelation. The performance of single sensor tracking and centralized tracking, was computed analytically by the Cramer Rao bounds. The two reference cases provide lower and upper bounds for what can be achieved with the sensor measurements.

The Cramer Rao bound provides a theoretical lower bound on the estimation error covariance matrix that is achievable by an unbiased nonlinear estimate [24]. For continuous time nonlinear deterministic models with discrete-time nonlinear measurements with additive Gaussian white noise, it can be shown [25] that the extended Kalman filter covariance propagation equations linearized about the true (unknown) trajectory provide the Cramer Rao bound to the estimation error covariance matrix. This is computationally far more efficient than Monte Carlo runs.

4.3.2 Simulation Results

Figures 8 to 11 show the Monte Carlo performance results of the convex combination and tracklet fusion algorithms for 10 sec. communication interval and the Cramer Rao bounds for single sensors and both sensors (centralized fusion). The local sensor observation interval is 1 sec. Note that there is definitely benefit in fusion. Both fusion algorithms essentially achieve the performance of centralized fusion (measurement fusion) as predicted by the Cramer Rao bounds. This is consistent with the theoretic analysis that shows the optimality of both algorithms. The results also show that the performance right after communication does not seem to be affected by the communication frequencies simulated. Nominal values in the figures are 1 Km. for position and 1 m/s for velocity.

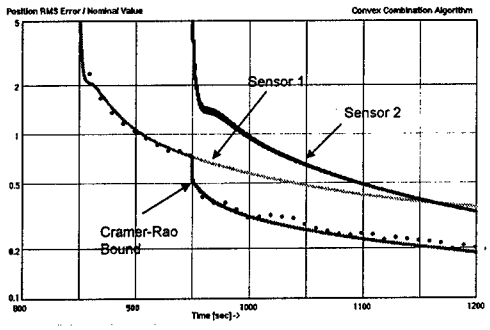


Figure 8: Position Error for Convex Combination

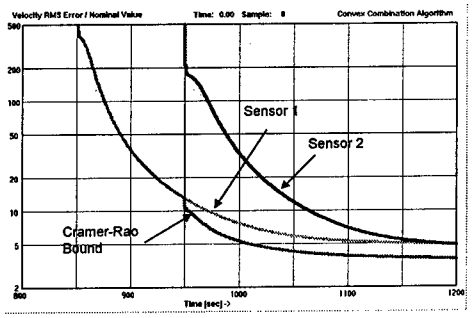


Figure 9: Velocity Error for Convex Combination

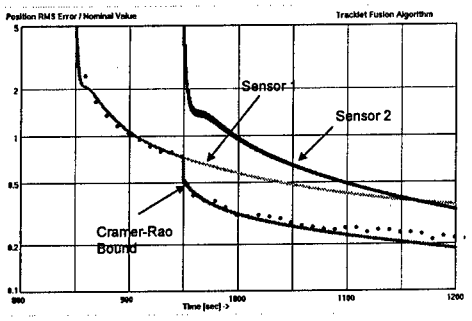


Figure 10: Position Error for Tracklet Fusion

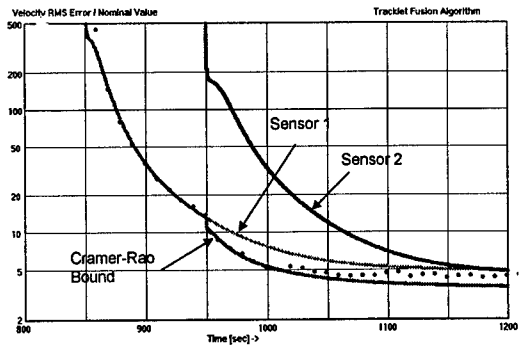


Figure 11: Velocity Error for Tracklet Fusion

5. Track Association

Before the track state estimates can be combined, the sensor tracks have to be associated either with each other (sensor track to sensor track fusion) or with the system tracks (sensor track to system track fusion). As discussed before, if the sensor tracks were perfect and the previous associations can be trusted, then only the new sensor tracks need to be associated. The state estimates of the other sensor tracks are fused with the system tracks that contain them to update the estimates. In practice, both the sensor tracks and the previous associations may have errors. Thus some form of re-association is needed. In a multiple hypothesis approach, this is handled naturally as the probabilities of the association hypotheses are re-evaluated. In a single hypothesis approach, the system tracks and sensor tracks are evaluated to determine which ones need to be re-associated along with the new sensor tracks.

Track association consists of the two key steps: computing a table of association metrics and selecting the best association hypothesis, usually by some assignment algorithm.

5.1. Track Association Metrics

The association metric measures how close one track is to another so that association decisions can be made. A traditional association metric is the squared Mahalanobis distance. Given two tracks i and j with mean estimates and covariances represented by $(\hat{x}_i, V_i)$ and $(\hat{x}_j, V_j)$, the Mahalanobis distance is defined as:

$$\chi_{ij} = \sqrt{(\hat{x}_i - \hat{x}_j)^T (V_i + V_j)^{-1} (\hat{x}_i - \hat{x}_j)} \qquad (15)$$

The association metric has to be modified when the track state estimation errors are correlated. The problem was considered by Bar-Shalom [6] when the correlation was due to common process noise and the result is similar to that in (3) and (4). Even when process noise is absent, one still has to be careful. In particular, in computing the association metric between a sensor track and a system track that contains the sensor track, one cannot ignore the correlation between these tracks. In the following we discuss how the association metrics should be modified depending on the quality of the sensor track or quality of the previous association.

5.1.1 Imperfect Association

When the previous association between a sensor track and a system track is questionable, the sensor track and the system track need to be re-associated with other system tracks and sensor tracks. Since the sensor track and system track have correlated estimation errors, the association metric has to account for the correlation in order to give correct results. In Figure 12, the system track shown was previously associated with Sensor Track 1 and Sensor Track 2 (and possibly other sensor tracks). Suppose it is necessary to reconsider the association between the system track and the sensor track due to large Mahalanobis distance between the tracks. Assume also that the purity of the sensor track 1 can be trusted. The effect of the sensor track 1 has to be removed from the system track before the association metric can be computed. Otherwise the system track will be too close to the sensor track. The error covariance and state estimate of the de-correlated system track is given by:

$$(V_{\text{dSYS}})^{-1} = V_{\text{i}}^{-1} - \overline{V}_{\text{j}}^{-1} \qquad (16)$$

$$(V_{\text{dSYS}})^{-1} \hat{x}_{\text{dSYS}} = V_i^{-1} \hat{x}_i - \overline{V}_j^{-1} \overline{x}_j \qquad (17)$$

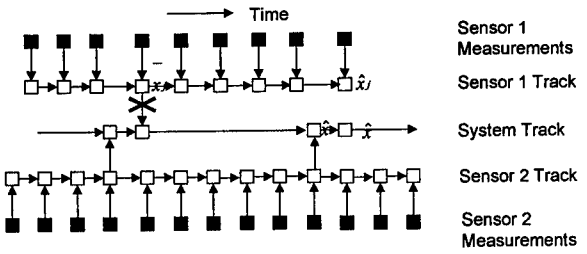


Figure 12: De-correlation of System Track

The de-correlated system track state estimate can then be used in calculating the association metric.

5.1.2 Imperfect Sensor Track

When re-association is needed due to impurity in the sensor track, the tracklet formed from the measurements since the last association is re-associated with the system track. This is the case when the previous association can be trusted, but the continuity of the sensor track is questionable. The state estimate (both mean and covariance) of the de-correlated sensor track is computed using the approach of Section 4.2.1 and given by:

$$(V_{\text{dSEN}})^{-1} = V_j^{-1} - \bar{V}_j^{-1} \quad (18)$$

$$(V_{\text{dSEN}})^{-1}\hat{x}_{\text{dSEN}} = V_j^{-1}\hat{x}_j - \bar{V}_j^{-1}\bar{x}_j \quad (19)$$

Essentially we have replaced the sensor track 1 with the tracklet since the last association with the system track (Figure 13). This de-correlated sensor track is then used to compute the association metric with the system track.

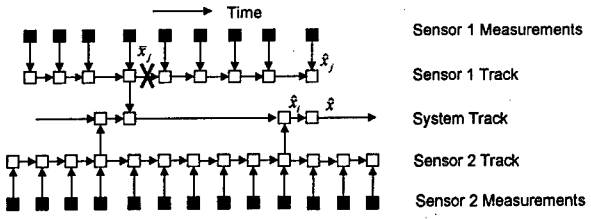


Figure 13: Sensor Track De-correlation

6. Summary

In this paper we have considered the track fusion problem and technical issues including correlated estimation errors in the tracks, imperfect sensor tracks and impure previous association. Among these, correlated estimation errors depend on the fusion processing architecture and affect the choice of track fusion algorithms. We presented different approaches for fusing track state estimates, and compared their performance through theoretical analysis and Monte Carlo simulations. We also discussed different approaches for computing the association metric.

7. References

1. S. Mori, W. H. Barker. C. Y. Chong, and K. C. Chang, "Track Association and Track Fusion with Non-Deterministic Target Dynamics," in *Proc. 2nd International Conf. On Information Fusion*, 1999.
2. C. Y. Chong, S. Mori, and K. C. Chang, "Distributed Multitarget Multisensor Tracking," in *Multitarget Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom, ed., Norwood, MA: Artech House, 1990, pp. 247 - 295.
3. C. Y. Chong, "Distributed Fusion Architectures and Algorithms," in *Proc. 1998 International Conf. on Multisource-Multisensor Data Fusion*, 1998.
4. Y. Bar-Shalom and T. E. Fortman, *Tracking and Data Association*, San Diego, CA: Academic Press, 1988.
5. Y. Bar-Shalom, and L. Campo, "The Effect of the Common Process Noise on the Two-Sensor Fused Track Covariance," *IEEE Trans. Aero. Elect Syst.*, Vol. AES-22, pp. 803 - 805, Nov. 1986.
6. Y. Bar-Shalom, "On the Track-to-Track Correlation Problem," *IEEE Trans. Automat. Contr.*, Vol. AC-26, No. 2, pp. 571 - 572, Apr. 1981.
7. K. C. Chang, R. K. Saha, and Y. Bar-Shalom, "On Optimal Track-to-Track Fusion," *IEEE Trans. Aero. Elec. Syst.*, Vol. 33, No. 4, pp. 1271 – 1276, Oct. 1997.
8. J. L. Speyer, "Computation and Transmission Requirements for a Decentralized Linear-Quadratic-Gaussian Control Problem," *IEEE Trans. Automat. Contr.*, Vol. AC-24, pp. 266 - 269, April 1979.
9. C. Y. Chong, "Hierarchical Estimation," in *Proc. MIT/ONR Workshop on C3*, Monterey, CA, 1979.
10. A. Willsky, M. Bello, D. Castanon, B. Levy, and G. Verghese, "Combining and Updating of Local Estimates Along Sets of One-Dimensional Tracks," *IEEE Trans. on Automat. Contr.*, Vol. AC-27, pp. 799-813, Aug. 1982.
11. C. Y. Chong, S. Mori, and K. C. Chang, "Information Fusion in Distributed Sensor Networks," in *Proc. 1985 American Control Conf.*, pp. 830 - 835, 1985.
12. H. R. Hashemipour, S. Roy, and A. J. Laub, "Decentralized Structures for Parallel Kalman Filtering," *IEEE Trans. Automat. Contr.*, Vol. AC-33, No. 1, pp. 88 - 93, Jan. 1988.
13. H. F. Durrant-Whyte, B. S. Y. Rao, and H. Hu, "Toward a Fully Decentralized Architecture for Multi-Sensor Data Fusion," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1990.
14. B. Belkin, S. L. Anderson, and K. M. Sommar, "The Pseudo-Measurement Approach to Track-To-Track Data Fusion," in *Proc. 1993 Joint Service Data Fusion Symposium*, pp. 519 - 538, 1993.
15. R. Lobbia and M. Kent, "Data Fusion of Decentralized Tracker Outputs," *IEEE Trans. Aero. Elect. Syst.*, vol. AES-30, vo. 3, pp. 787-799, July 1994.
16. M. E. Liggins, II, C. Y. Chong, I. Kadar, M. G. Alford, V. Vannicola, and S. Thomopoulos, "Distributed Fusion Architectures and Algorithms for Target Tracking," *Proc. IEEE*, vol. 85, no. 1, pp. 95 – 107, Jan. 1997.
17. O. E. Drummond, "Feedback in Track Fusion without Process Noise," in *Proc. SPIE, Signal and Data Processing of Small Targets*, vol. 2561, pp. 369 - 383, 1995.
18. O. E. Drummond, "A Hybrid Sensor Fusion Algorithm Architecture and Tracklets," in *Proc. SPIE, Signal and Data Processing of Small Targets*, vol. 3163, 1997.
19. O. E. Drummond, "Tracklets and a Hybrid Fusion with Process Noise," in *Proc. SPIE, Signal and Data Processing of Small Targets*, vol. 3163, 1997.
20. M. D. Miller, O. E. Drummond, and A. J. Perrella, "Tracklets and Covariance Truncation Options for Theater Missile Tracking," in *Proc. 1998 International Conf. on Multisource-Multisensor Data Fusion (FUSION'98)*, 1998.
21. G. Frankel, "Flexible Architectures for Sensor Fusion in Theater Missile Defense," IDA Paper P-2935, April 1994.
22. G. Frenkel, "Track Fusion for Intercept Support," IDA Working Paper, April, 1995.
23. G. Frenkel, "Multisensor Tracking of Ballistic Targets," in *Proc. SPIE, Signal and Data Processing of Small Targets*, Vol. 2561, 1995.
24. H. L. Van Trees, *Detection, Estimation and Modulation Theory, Part I.* New York, Wiley, 1968.
25. J. H. Taylor, "The Cramer-Rao Estimation Error Lower Bound Computation for Deterministic Nonlinear Systems," *IEEE Trans. Auto. Control.*, Vol. AC-24, No.2, pp. 343 – 344, April 1979.

A Multiple Hypothesis Tracker for a Multiple Sensor Integrated Maritime Surveillance System

Zhen Ding and Ken Hickey
Advanced System Development
Raytheon Systems Canada Limited
400 Philip Street
Waterloo, Ontario, Canada, N2J 4K6
Phone: 519-885-0110 ext. 549, 526
Fax: 519-885-1252
zhen_ding@raytheon.com
ken_hickey@raytheon.com

Abstract *This paper presents a review of a Multiple Hypothesis Tracker (MHT-2000) used for oceanic surface target tracking. A mixed co-ordinate system is selected in a multi-station radar configuration and a Converted Measurement Extended Kalman Filter (CMEKF) is implemented in the fixed co-ordinate frame. This tracker is a component of an Integrated Maritime Surveillance system which consists of multiple High Frequency (HF) Surface Wave Radars (SWRs) and Automatic Dependent Surveillance (ADS) systems. Simulated and real time multi-sensor data were used to evaluate the system and some of the results from this investigation are presented in this paper.*

Keywords: Surface Wave Radar; Automatic Dependent Surveillance System; Integrated Maritime Surveillance; Multiple Hypothesis Tracking; Multiple Sensor Data Fusion.

1. Introduction

Raytheon Systems Canada Limited (RSCL) and the Canadian Department of National Defense are collaborating on the development of a long range maritime surveillance system for the monitoring of Canada's Exclusive Economic Zone (EEZ) [1]. The system has been designed to provide continuous, all weather surveillance of aircraft, surface vessels, icebergs, and environmental conditions to aid in the protection of Canada's natural resources, and to monitor and control the coastline for smuggling, drug trafficking, etc.

The system employs two overlapping HF Surface Wave Radars (SWRs) and several Automatic Dependent Surveillance (ADS) systems [1, 2]. The radars main database and the Operation Control Center (OCC) is located in St. John's, Newfoundland. However, all the system operations are remotely controlled from RSCL, in Ontario. A Multiple Hypothesis Tracking (MHT) algorithm is used as the focal point of the tracking system since it favorably accommodates multiple targets, new targets, false alarms and missed detections.

This paper describes the fusion processor that performs four major tracking functions: MHT processing and management of clusters, fusion of the two radar reports, track tagging using the ADS systems, and the final fusion of the radar and ADS tracks.

Detections from the polar co-ordinate system are converted to a common co-ordinate frame and the corresponding Converted Measurement Extended Kalman Filter (CMEKF) is employed. The specially selected CMEKF is used to utilize the range rates from the two radars that have different measurement co-ordinate systems.

2. An Outline of the System

The Integrated Maritime Surveillance (IMS) system (as illustrated in Figure 1) is a shore based system, which detects, tracks, and identifies aircraft and ships throughout the EEZ.

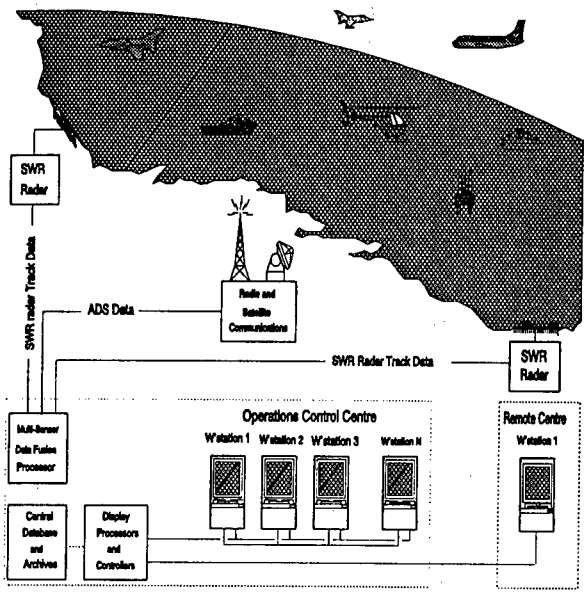


Figure 1: An IMS system.

The IMS comprises four principle components:

2.1 Long Range Surface Wave Radars

Radar coverage of coastal waters has traditionally been limited to line-of-sight and this is an inherent characteristic of radar systems operating at microwave frequencies [3]. Radar operations at the lower end of the HF band (3 MHz to 6 MHz) use the surface wave mode of propagation. In this mode, the radar signal follows the curvature of the earth such that targets hundreds of kilometers beyond the horizon may be detected. Since the SWR is a coherent radar, range rate measurements are also provided.

2.2 ADS Systems

Aircraft and/or vessels equipped with ADS systems transmit identification and position information on a regular schedule over preassigned communication channels to a shore based processing system.

2.3 Other Adjunct Sensors

Adjunct sensors are the systems that traditionally provide surveillance and include communications, mandatory-reporting procedures, and visual identification from patrol vessels and aircraft. These sensor reports are characterized by their infrequent and often tardy nature.

2.4 Multi-sensor Data Fusion

The data fusion system automatically correlates tracks from multiple SWR sites with the ADS tracks and target attributes obtained from communication channels. ADS reports are sent to the OCC database. The database is polled for new reports. After processing, all the tracks are stored in the OCC database.

3. The MHT Processing

The MHT algorithm is described in [4, 5]. It is a statistical approach, incorporating false targets, new tracks, missed detections and finite track lifetimes. The basic premise is that, through the application of Bayes' rule, the probability of any track-to-detection combination, over a given number of radar updates, is solely dependent on the probability of the combination from the previous update, and the probability of the current track-to-detection updated association. The algorithm thus does not make 'hard' assignments at each step. Instead, it keeps the N most likely possible track-to-detection associations, where N is typically 2-4, and ranks them by their probabilities (i.e. how likely a given association is) [7, 8]. Such hypotheses may be efficiently updated at each step merely by calculating the current probabilities for association. Thus, a track and detection association that looks very likely at one stage, may at a later time, be revealed to be less feasible as its updated probability decreases. The correct (or more likely) association hypotheses will then predominate.

The MHT processing assumes that each new radar report is either an extension of an existing track, a new potential target or a false alarm. The possibilities of these extensions (or

events), together with a missed detection scenario for each of the tracks, account for the propagation of the hypotheses at each update time. The combination and extension of tracks and hypotheses imply that there is an exponential growth as new hypotheses are formed at each update. Practical implementations using MHT processing requires that limits are placed on this growth. The RSCL implementation (MHP-2000) propagates several of the most likely possible track-to-detection scenarios forward, thus allowing for the likelihood of missed detection, crossed tracks and false alarms. Efficiency is maintained by clustering the data to reduce the gross complexity of the problem. Multiple hypotheses are considered only for clusters, each of which is a group of tracks and detections that are close to each other. These clusters encompass hypotheses that share common reports but these are separated from those in other clusters. In this way, clusters may be processed independently and in parallel, thus preventing unconstrained growth of the hypothesis tree. The best track-to-detection hypotheses are determined as solutions to a linear assignment problem, where the 'cost' is determined ,probabilistically, by the targets and detection distance. Figure 2 shows the basic data flow in the MHT algorithm.

The MHP-2000 performs target tracking, data/track association and fusion of the HF radar detections and the ADS report processing. Air and ship data detection streams are input to the tracker for independent processing. Once a track is set up, the target trajectories are propagated using a CMEKF which is a linearized Kalman Filter. Associations (track-to-detection) are performed in antenna-based polar co-ordinates, and propagated in a North-East Cartesian co-ordinate system, centered at the primary radar site. The second radar data is transformed to the same co-ordinate system. This allows optimal usage of the detection data, since it is the most accurate in the Doppler or range rate domain, and can thus be exploited during this association. The processing proceeds as follows.

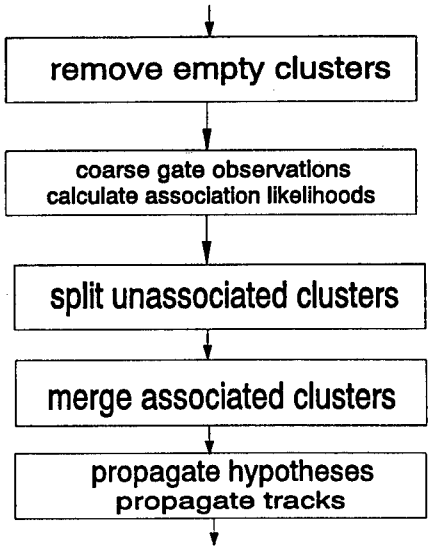


Figure 2: One cycle of MHT processing.

1. Determine the penalty matrix (i.e. the log-likelihood function) by associating all existing tracks with all new reports as well as possible missed detections. (Coarse and fine gating are first applied here to prevent obvious associations between reports and targets that are distant from each other).

2. Trim the clusters by removing deleted tracks and clusters that have become empty.

3. Split the clusters and find those tracks that have common measurements. Reform the clusters based on any new associations which may include further cluster merging and deletion.

4. Update the clusters by:

   (a) Forming the appropriate track updates by using the CMEKF to update tracks and coasts (missed detections), as well as initializing potential new tracks.

   (b) Sorting the hypotheses and determining the N-best ones (N will be set in the tracker after initial analysis and testing).

   (c) Updating the hypothesis list, based on the above processing.

4. Kalman Filter Design

In a single site MHT tracker, the Kalman filter state could be represented in track-oriented rectangular co-ordinates (TORC) [9], where the X-axis is the radial line between the target and the radar. This has the advantage that the range is approximately the x value and the y value is very small. As a result, the matrices associated with the Kalman filter process are very sparse and updates may be performed efficiently. Since the MHP-2000 fuses SWR radar reports from two or more sites, such a co-ordinate system is no longer acceptable. In the overlapped region, consecutive same target reports may come from different radars. As a result, tracking and association in such a radial co-ordinate system would be optimal for only one of the reports. The system therefore tracks in a co-ordinate system which is centred about one of the radar sites.

The tracker state matrix is in the form $[x(k), \dot{x}(k), y(k), \dot{y}(k)]$. The track association, using probability distance measures, is in track-oriented polar co-ordinates (TOPC) prior to the conversion to a fixed Cartesian co-ordinate system, i.e., a mixed co-ordinate system is used. Tracking is done in the Cartesian co-ordinate fame where measurements are converted detections and original Doppler measurements (accordingly, a mixed CMEKF is employed to use Doppler measurements). In addition, since converted measurements are used, additional processing is needed to remove the pseudo linearized measurement bias.

5. Radar/ADS Fusion

ADS reports are input to the sensor fusion processor to enhance track fusion and target updating. These reports are received, regularly, from vessels equipped with a Global Positioning System (GPS). A functional diagram is shown below.

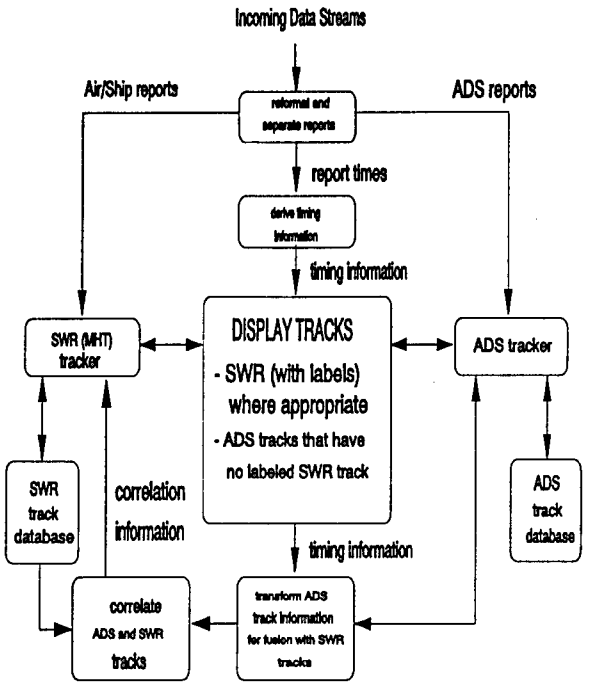


Figure 3: RADAR/ADS track fusion.

Track level fusion occurs using the ADS reports and any other system that may enter similar information into the OCC database. ADS reports are entered to their own track database in the MHP-2000 which employs a simple polynomial tracker to improve the track quality. The ADS reports are fused as follows:

1. Generation and updating of ADS tracks.

2. Time synchronization of all confirmed radar tracks and ADS tracks which involves track filter updating to an appropriate time reference.

3. Gating of radar tracks and ADS tracks, i.e., the elliptical (probabilistic) gating of surviving radar tracks and ADS tracks.

4. The ADS identifier (ID) will be passed to and preserved by the radar track that it corresponds to.

5. Covariance estimation and state fusion are performed if the latest report is current and there are at least three points in the ADS tracks [6].

6. If the new ADS report is current and there is only one or two points in the track, association is done just for the purposes of track ID assignment.

Other sensor data, such as those from the air defense network and ATC systems, can be fused into the existing system.

6. Testing Results

In this section, we present some test results using both simulated and real time data.

During the IMS trials, questions arose concerning the detection and tracking of the large Hibernia oil platform and nearby tankers [2]. Since their dimensions are in the order of the radar wavelength, mutual RCS interactions result in severe fluctuations in detected power, which make detection and tracking harder. Hibernia is a large structure 90m tall and 110m wide. In the vicinity of the platform, large tankers, of approximately 100m in length, load oil from the platform.

A simulated radar return from the Hibernia structure was used to test and design the tracking filter. This simulated data was based on a positional estimate of 328km and 84.5 degrees relative to Cape Race with a range standard deviation of 2000m, an azimuth standard deviation of 1 degree and a range rate standard deviation 0.5m/s.

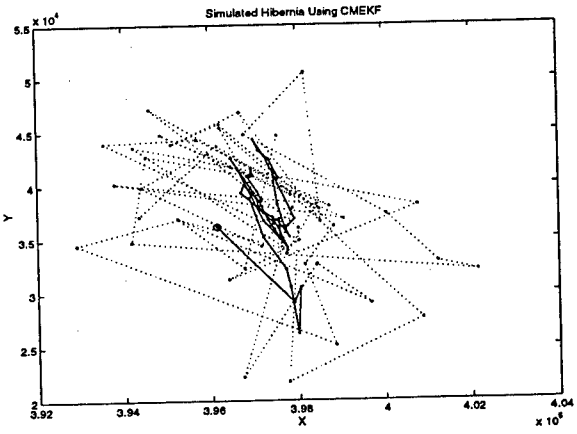


Figure 4: Hibernia tracking, where dotted lines denote measurements and solid lines are track outputs.

Figures 4 displays the test results of the MHP-2000 processor. The tracker works well in reducing the variance of the original positional estimates. Figures 5 and 6 show the corresponding converted measurement (dashed line) and the tracker estimation. The X converted measurement is observed to exhibit a lower error variance than the original measurement.

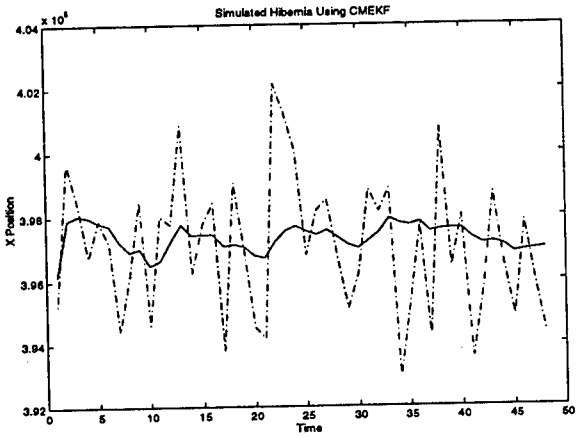


Figure 5: X measurement and estimation.

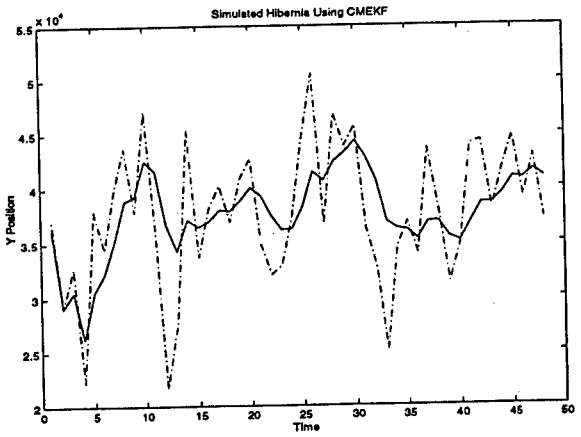


Figure 6: Y measurement and estimation.

Numerous real time tests were performed to test the tracking capability of the MHP-2000. Furthermore, the following results are noteworthy:

- **Long range detection ability.** The system has successfully detected and tracked targets to 450km. Figure 7 demonstrated a real time test case with distant ship tracks. Long range and small target detection requires low CFAR threshold which

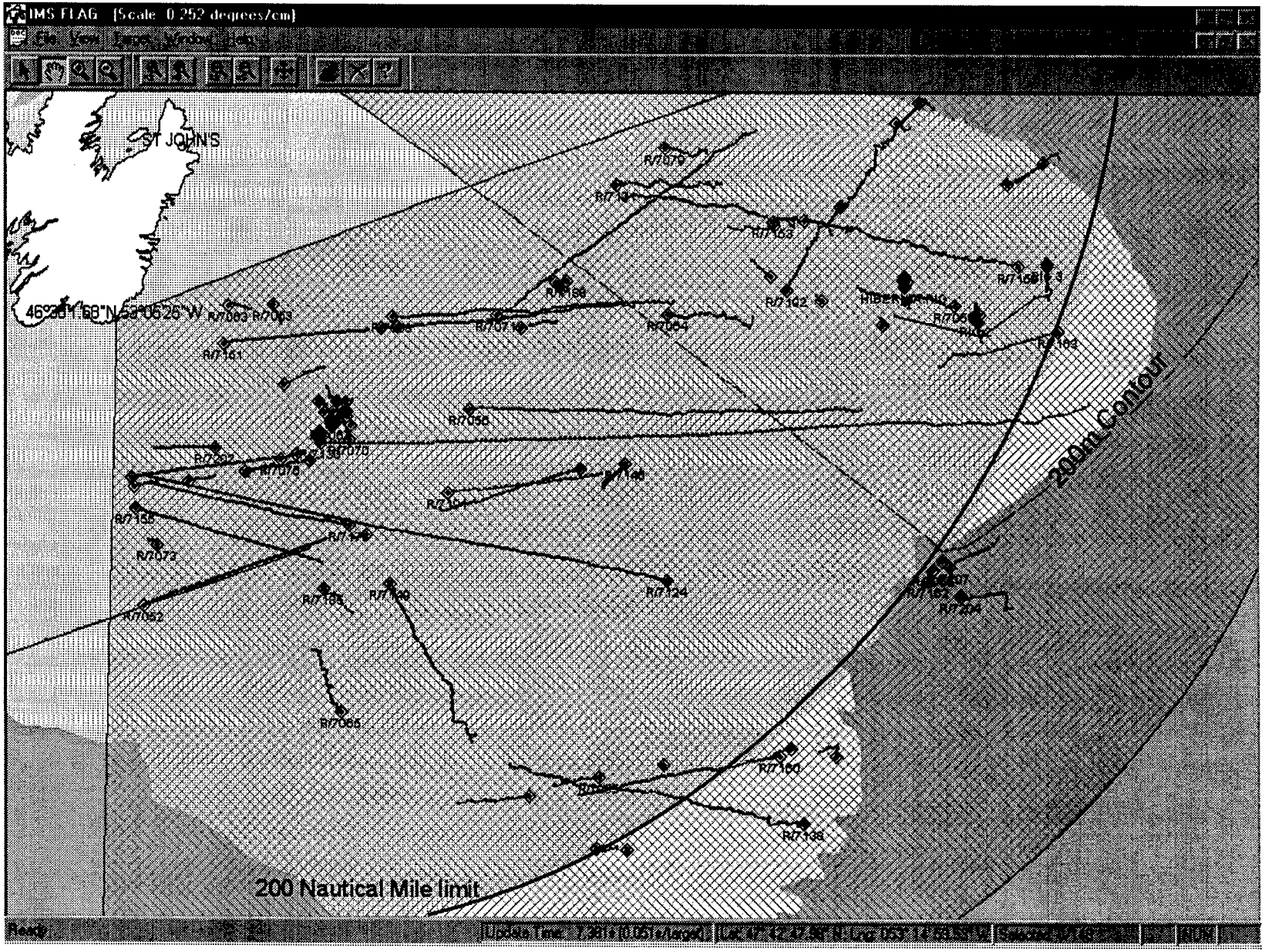


Figure 7: Ship target tracking (on-line mode)

introduces dense clutter. In this environment, the MHP-2000 tracker works well. Note that two modes (diagnostic and on-line modes) are set for the tracker. The diagnostic mode is used for off-line diagnostic processing and the on-line mode is for real time processing.

- **Robust tracking ability in a highly dense false alarm environment.** The average number of measurements is around 300, with up to 400 measurements per update. The MHP-2000 has demonstrated its target tracking capability in this environment.

- **Simutaneous tracking of air and ship targets.** Figures 8 and 9 illustrate an example of air target tracking. Since an air target moves much faster than a surface target, the dwell time must be set much lower. Therefore, a wider band Kalman filter model is required whose parameters are changed accordingly. Differing CFAR threshold scheme are also employed for each data stream. Hence, both data streams are independently processed in the MHP-2000 tracker. All tracks are sent and stored in the OCC database.

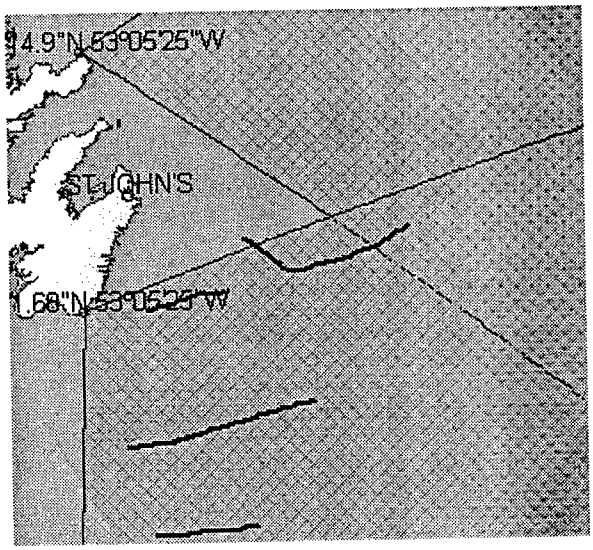


Figure 8: Air tracking results.

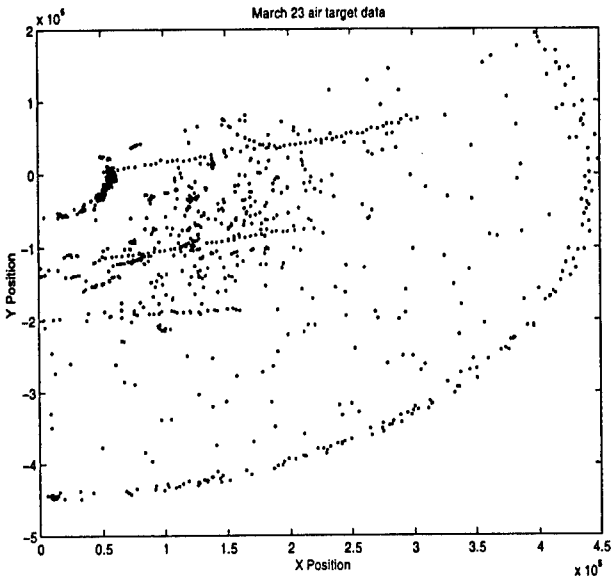


Figure 9: Air reports.

- **Fusion of multiple radars and other sensors.** Currently, two radars and ADS are used. The dual radar tracking capability of the MHP-2000 has been tested using simulated data. Figure 10 shows an eight target case where two targets are detected by both radars. Real time evaluation of the two radar system is ongoing.

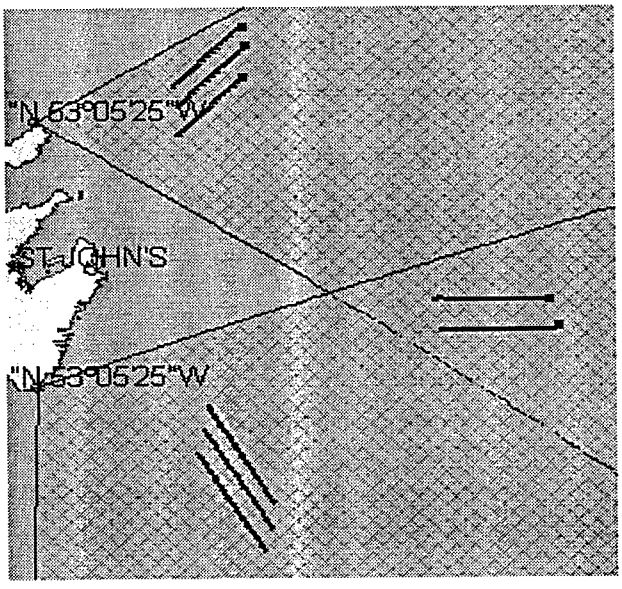


Figure 10: Eight target tracking.

- **Night detection and tracking ability** The East Coast IMS system demonstrated

its night detection and tracking ability. Figure 11 shows a night tracking test (data set was collected from 12:34 pm Feb. 18 to 5:05 am Feb. 19).

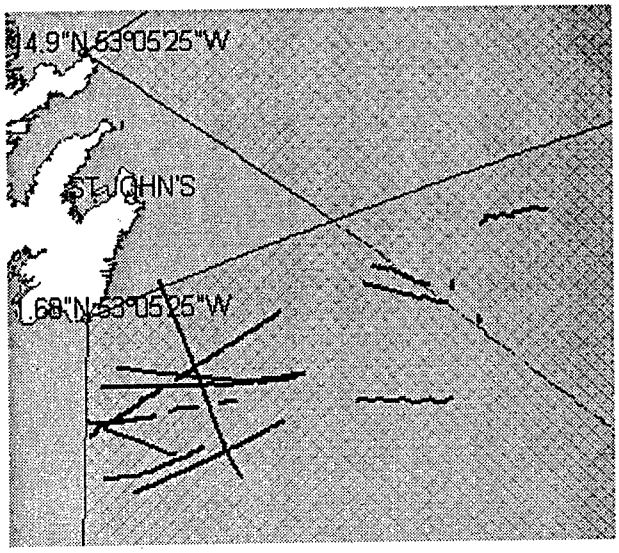


Figure 11: Target tracking in night.

7. Conclusion

This paper presents a review of a Multiple Hypothesis Tracker (MHT-2000) used for oceanic surface target tracking. A mixed co-ordinate system is selected and a Converted Measurement Extended Kalman Filter (CMEKF) is implemented. This tracker is used in an Integrated Maritime Surveillance system. Simulated and real time multi-sensor data were used to evaluate the system and some of the results are presented in this paper.

Acknowledgement

The East Coast IMS system is a collaborative project between the Canadian Government and Raytheon Systems Canada Limited.

References

[1] *Canadian East Coast Surveillance System*, Raytheon Canada Ltd., RCL7401-1, ON, Canada, 1997.

[2] L. Sevgi and A.M. Ponsford, "An HF Radar Based Integrated Maritime Surveillance System", *The 3rd International Conference on Circuits, Systems Communication and Computers,* Athens, Greece, July 4-8, 1999.

[3] L. Sevgi, "Stochastic Modelling of Target Detection and Tracking in Surface Wave HF Radars", *International Journal of Numerical Modeling: Electronic Networks, Devices and Fields*, Vol. 11, pp. 167-181, 1998.

[4] D.B. Reid, "An Algorithm for Tracking Multiple Targets", *IEEE Transactions on Automatic Control*, Vol. 24, No. 6, pp. 8434-854, 1979.

[5] S.S. Blackman, *Multiple Target Tracking with Radar Applications*, Artech House, 1986.

[6] Y. Bar-Shalom and T.E. Fortmann, *Tracking and Data Association*, Academic Press, 1988.

[7] R. Danchick and G.E. Newnam, "A Fast Method for Finding the Exact N-best Hypotheses for Multitarget Tracking", *IEEE Transactions Aerospace and Electronic System*, Vol. 29, No. 2, pp. 555-560, 1990.

[8] I.J. Cox and M.L. Miller, "On finding ranked assignments with application to multitarget tracking and motion correspondence", *IEEE Transactions Aerospace and Electronic System*, to appear.

[9] *Artificial Intelligence Applied to Target Tracking-Phase 2: Implementation*, Raytheon Canada Ltd., RCL00158-671/A, ON, Canada, 1995.

Fuzzy Clustering for Association and Fusion in Multitarget Tracking with Multisensors*

by

Carl Looney and Yaakov Varol
Computer Science Department/171
University of Nevada
Reno, NV 89557
Looney@cs.unr.edu, Varol@cs.unr.edu

Abstract - *Three problems are involved in updating tracks of multiple targets at a central processing station based on the state estimates coming in from multiple local radar stations. The first is the synchronization of the estimated states to move them forward to a single time reference and to similarly obtain the predicted states at that reference time by means of the central tracks. The second problem is the association of the states to determine which ones represent the same target. The third is that of fusing the estimates for the same target into an updated state estimate with reduced error. The latter two problems are the subjects of study in this paper. We employ our new fuzzy clustering algorithm to associate the local track states and the predicted states. States that belong to the same cluster are associated. The clustering also finds a fuzzy weighted prototype that is the typical (smoothed) state of each cluster, which is an updated fused state for the cluster. Thus both problems are solved by clustering. We show simulation results for preliminary testing.*

Keywords: *radar, tracking, association, fusion, fuzzy clustering*

1. Introduction

1.1 Multitracking with Multisensors

We consider a multisensor and multitarget tracking system that consists of multiple local radar stations that update their tracks and then transmit the data to a central tracking station. The *state* of a target at a specific time instant consists of the target position, the velocity, the acceleration, the reference time, the track number and possibly other fields. A sequence of consecutive states over time forms a *track* of the target. If the local stations process their own tracks, in which case they would transmit their newly updated track states to the central station, the system is called *decentralized* [1,12]. If they transmit their positional readings and the central station performs all of the tracking, then the system is *centralized*. A decentralized system is efficient because it reduces both the complexity at the central processor and the bandwidth of the communications.

In the *hierarchical* case that we investigate here, the local tracks are updated by the local track-while-scan [5] stations from their measurements and the new states are transmitted to the central station, which uses them to update its *central tracks*. Generally the local states are transmitted after every fixed number of local updates.

The central station solves the first problem of *synchronization* by extending all received track states, and also its own previous central track states, forward to a common reference time [7]. For simplification of our simulation we omit this step and take all readings of target positions at the same time. The next problem for the central station is the track-to-track *association* problem for making a decision as to whether or not multiple tracks states coming from different sensors represent the same target [1]. The third problem is the *fusion* of the associated track states and the central predicted state into a central track state with smaller error. These problems and the basics of radar tracking are discussed in [5].

* This work was supported by ARO Contract #39713-MA

1.2 Approaches to Association and Fusion

The current methods are mostly probabilistic. Many use *multiple hypothesis testing* (MHT) to associate the track states, which sometimes requires significant computation. One example [10], uses MHT with sequential likelihood ratio tests and signal strength. The fusion to obtain new updates is usually done by Kalman and extended Kalman filtering [1,6,7,12] that requires matrix inversion. The $\alpha\beta$ and the $\alpha\beta\gamma$ (simplified Kalman) filters are also popular and they may depend upon correlations [9] as do Kalman filters.

Certain problems can, and do, arise, however, in probabilistic approaches to multisensor tracking. Target acceleration noise [7] necessitates taking into account the *cross-correlation* between sensors when employing a Kalman (or extended Kalman) filter. Without the cross-correlation matrix in the updating, a strong bias can be introduced [1]. Also, the transformation from polar (radar measurement) to Cartesian (tracking) coordinates further degrades the tracking performance due to nonlinear effects that are nonnegligible [6].

We seek to increase both computational efficiency and accuracy while avoiding the use of unknown apriori probability distributions and the real-time computation of inverse matrices as required by Kalman filtering. It occurred to us that the training of a multiple layered perceptron neural network [2,3] over real world data could build into the tracking system the ability to deal with nonlinear effects without the need for extraneous correction methods. The training data could be gathered by actually flying aircraft trajectories and recording the *global positioning system* (GPS) data to obtain essentially true positions for the output training data. The radar measurements of a flight become the input training data.

Upon checking the various types of neural networks, which led to self-organizing (clustering) networks, we were struck with the notion that clustering can do both association and fusion simultaneously. This would circumvent the extra effort required to gather real world data on which to train different neural networks for the many different situations for the various scenarios that arise in the field.

2. A Combined Approach

2.1 Tracking

We employ our new fuzzy clustering algorithm [4], which is very fast and which both associates and fuses the states in a single process. This contrasts with the commonly used method for associating two tracks by testing the hypothesis as to whether or not they are the same track, which can be problematic for multiple targets because of the number of pairwise combinations. Also, each pair of tracks has the same underlying process noise that makes the track estimation errors dependent.

Each of the *local radar stations* RS1,...,RSn measures the target positions in polar coordinates (r,θ,ϕ) relative to the local radar station, where r is the *range*, θ is the *azimuth* and ϕ is the *elevation*. For our purposes here, we store the local tracks with respect to the central Cartesian coordinate system. These updated local target states are transmitted to the *central processing station* (CPS) for associating and fusing with its tracks.

Our simulation here uses 2 local radar stations RS1 and RS2 and three target trajectories T1, T2 and T3. The discrete trajectories are target paths with points generated at 5 second time increments. There are thus 6 target states to be transmitted to the CPS for each update time (3 targets from each of the 2 local radar stations). The CPS is not a radar station here but is only a computer center for processing tracks. The simulation avoids moving the states forward to a common reference time by reading all targets at the same 5 second increments. The updated local states are transmitted to the CPS at 5 second increments.

2.2 Association and Fusion via Clustering

The local target states at update time t_{up} that are reported to the CPS are fed into our clustering algorithm [4], along with the predicted states obtained by moving the previous central state forward in time to t_{up}. A few iterations compose the

clusters with the results that those states in the same cluster are associated to represent the same target. The fuzzy clustering algorithm also determines a fuzzy expected state that is a fused estimate for the new updated state for the central track. This new fused state is stored in the track associated with the cluster with track number of the predicted central state. The predicted central states are used as the initial centers of clusters. Thus the fusion problem is also solved simultaneously.

3. The Simulation

3.1 Generating the Data

The two local radar stations designated as RS1 and RS2 each sense the 3 target trajectories denoted by T1, T2 and T3 that are generated by computing the target positions at 5 second increments. T1 is a straight line, T2 is an ellipse at distance and T3 flies from East to West with consecutive periods of 10 seconds of rightward acceleration followed by 10 seconds of leftward acceleration (from the direction of forward motion of the target). For example, T1 starts at 400 km/hr and accelerates at 100 km/hr^2.

We store these trajectories in the files *target1.dta*, *target2.dta* and *target3.dta*. Their positions are with respect to the central Cartesian coordinate system, which contains the CPS at the origin with RS1 at (10,50,0) and RS2 at (100,5,0), where the numbers are in kilometers. Figure 1 shows the functional diagram for the hierarchical system for tracking T1, T2 and T3.

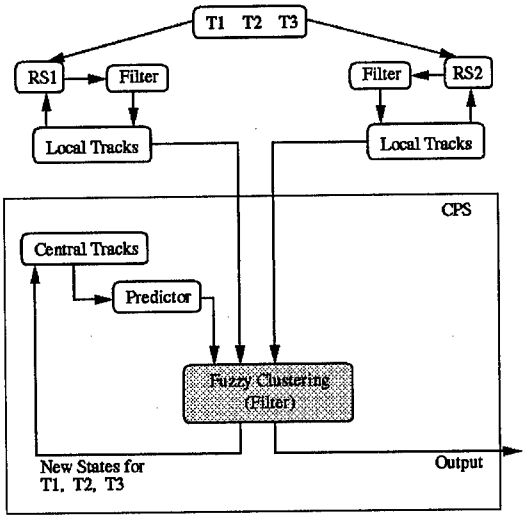


Figure 1. The hierarchical system.

Our local radar simulation function *locsim* reads the target trajectory files at 5 second intervals and translates their Cartesian positions to Cartesian coordinate systems centered at each of RS1 and RS2. It then converts the target positions to polar coordinates with respect to each of RS1 and RS2 for the modeling of the sensor readings. To the polar positions we add Gaussian white noise and then convert the noisy polar positions back to local Cartesian coordinates. We then translate these back to the central Cartesian coordinate system. The local stations update their tracks with the noisy position readings in this coordinate system by means of $\alpha\beta\gamma$ filters [5,9] (see Section 3.4 below).

The noise on each of r, θ and ϕ sensed at the two local radar stations is Gaussian white noise with zero-mean. For each of the three target trajectories, each of RS1 and RS2 reads the relative target position in polar coordinates and adds random errors from the noise distributions. For the range, we use standard deviations of $\sigma_r = 100$ meters. For σ_θ we use $1°$, and similarly for the elevation angle. Then we convert the noisy polar positions back to the central Cartesian system.

The polar coordinate readings by RS1 and RS2 have now become noisy Cartesian readings for updating the local tracks. After the local radar stations RS1 and RS2 update their tracks by means of $\alpha\beta\gamma$ filters, they store their updated track states in the files *loctrack1.dta* and *loctrack2.dta*. The first file stores the current local tracks for the 3 targets as given by RS1, while the second stores the 3 target tracks for RS2. These files contain the data to be transmitted to the CPS and are in the central Cartesian coordinate system.

The central processing simulation program *cpsim* function now reads and processes the local state files listed above as received data transmitted by RS1 and RS2. First, this program moves the previous central track state for each target forward to the reference time of the local states. The resulting *predicted* states are to be associated and fused with the received local states. At this point

there are 3 states for each target: the 2 local station states and the central predicted state.

The 6 track states from the local stations and the 3 predicted states from the CPS are all put through the clustering algorithm. The 9 track states are to be clustered into 3 clusters for the respective 3 targets. The initial cluster centers are taken to be the predicted states, which have central track numbers. A cluster associates all states that belong to it. The fuzzy weighted center for a cluster is the fused update state for the CPS tracks. After the clustering, the fused track states may be adjusted slightly in the velocity and acceleration components to be consistent with the smoothed positions. They are then written to the track file *CPStrack.dta*, which is used in the next update of the tracks.

3.2 Simulating Multisensor Multitracking

In our simulation, each state has the format $(x,y,z,v_x,v_y,v_z, a_x,a_y,a_z, t,ID,r)$, where the first three triplets are for position, velocity and acceleration, t is the reference time of the updated state (in 5 second increments), ID is the track identification number for the local state (1, 2 or 3) and r is the number of the local radar station (1 for RS1 or 2 for RS2).

The 9 states for 3 targets are used on each update at the CPS by clustering using only the position, velocity and acceleration components. The clustering process uses each central predicted state as an initial center for a cluster, but over a few iterations, a fuzzy expected value is determined for each cluster. In the usual case where each final cluster contains three current states, one from each of RS1, RS2 and CPS, those states are associated. The final fuzzy expected value for that cluster is the fused current state which is influenced mostly by the pair of states that are closest to each other. Clustering of states generalizes the concept of gates [5], where each cluster defines a gate.

3.3 Quality Parameters

We are currently working on the use of *quality parameters* q1, q2 and qC for the estimates given by the respective RS1, RS2 and CPS. Here we define it for RS1 (the others are analogous). Suppose that $\mathbf{s} = (x,y,z,v_x,v_y,v_z, a_x,a_y,a_z)$ is the latest state in the track for a target RS1 and let $\mathbf{S} = (X,Y,Z,V_X,V_Y,V_Z,A_X,A_Y,A_Z)$ be the latest MWFEV fused state given by the fuzzy clustering process. The difference $d = \|\mathbf{s} - \mathbf{S}\|$ is added onto the sum of the previous p-1 differences for RS1 to yield q1. The more recent squared differences are weighted more (and all weights sum to unity). Each station has such a parameter for each track.

It is well known [10] that tracks can be lost due to strong maneuvers, fading effects or incorrect data associations. On the other hand, false detections and incorrect associations can create false tracks. Thus in cases where not all 3 states fall into the same cluster, decision mapping must be implemented.

For example, when there are only 2 states in a cluster, with one being a local state from RS1 and the other being the central state, then the quality parameters are examined. If q2 > q1 and q2 > qC, where higher value means less quality, then we disregard the state from RS2 in updating that track. If q2 is low and appears in a separate cluster, then it is given a tentative new track ID and may be upgraded to a track if more corroborating returns are received on multiple future track time increments. At this stage we do not have all of the decision making mapped out but are gaining experience for doing this.

3.4 The Simulated Filtering

Let $\mathbf{x}_m$ be the noisy *measurements* vector at either of RS1 or RS2, where the polar coordinates have been converted to the central Cartesian system. Let $\mathbf{x}_T$, $\mathbf{v}_T$ and $\mathbf{a}_T$ be the position, velocity and acceleration from the central track for the previously filtered state. The *predicted* state vectors are determined from the central track by

$$\mathbf{x}_p = \mathbf{x}_T + (\Delta t)\mathbf{v}_T + \tfrac{1}{2}(\Delta t)^2\mathbf{a}_T \qquad (1)$$

$$\mathbf{v}_p = \mathbf{v}_T + (\Delta t)\mathbf{a}_T, \quad \mathbf{a}_p = \mathbf{a}_T \qquad (2a,b)$$

where Δt is the time increment.

The local state vectors are filtered (smoothed) via the $\alpha\beta\gamma$ filter [5] that uses noisy measurements and predicted position, velocity and acceleration to smooth the state components via

$$\mathbf{x}_s = \mathbf{x}_p + \alpha(\mathbf{x}_m - \mathbf{x}_p) \quad (3)$$

$$\mathbf{v}_s = \mathbf{v}_p + \beta\{1/(\Delta t)\}(\mathbf{x}_m - \mathbf{x}_p) \quad (4)$$

$$\mathbf{a}_s = \mathbf{a}_p + \gamma\{2/(\Delta t)^2\}(\mathbf{x}_m - \mathbf{x}_p) \quad (5)$$

The parameters α, β and γ are found from a small positive adaptable parameter ξ by [5]

$$\alpha = 1 - \xi^2 \quad (6a)$$

$$\beta = 1.5(1 - \xi^2)(1 - \xi) \quad (6b)$$

$$\gamma = 0.5\,(1 - \xi^3) \quad (6c)$$

As ξ increases toward 1 [5], the noise on the measurements is smoothed more strongly (the predicted value has greater weighting). But as ξ decreases toward 0, there is less smoothing (the noisy measurements have more influence).

It is well known that velocity derivations from positions strongly increase any noise in the position measurements and that acceleration derivations from velocity data increase the noise even more and can be unstable. For this reason, the velocity must be smoothed and the acceleration must be strongly smoothed.

If the range rate velocity were available from a *moving target indicator* (MTI), then it could be projected onto the Cartesian coordinates as a consistency check on the velocity. We do not use MTI data in this study, but we will do so in future work.

4. Fuzzy Clustering

4.1 A Fuzzy Expected Value

The *weighted fuzzy expected value* of $\{x_1,..,x_P\}$ was defined by Schneider and Craig [8] to be

$$\mu = \frac{\sum_{(p=1,P)} \exp[-\beta|x_p - \mu|]\, x_p}{\sum_{(r=1,P)} \exp[-\beta|x_r - \mu|]} \quad (7)$$

In place of the decaying exponentials we use the bell shaped Gaussian function that is a canonical *fuzzy set membership function* for the linguistic variable *CLOSE_TO_CENTER*. Vectors close to the center yield fuzzy truth values close to unity, but as they move away from the center, their truth values decrease rapidly toward zero. Starting with the mean $\mu^{(0)}$ of $\{x_1,..,x_P\}$, we employ the Picard iterations on the (r+1)st iteration via

$$\mu^{(r+1)} = \sum_{(p=1,P)} \alpha_p^{(r)} x_p \quad (8)$$

$$\alpha_p^{(r)} = \frac{\exp[-(x_p - \mu^{(r)})^2/(2\sigma^2)]}{\sum_{(r=1,P)} \exp[-(x_r - \mu^{(r)})^2/(2\sigma^2)]} \quad (9)$$

$$\sigma^2 = \sum_{(p=1,P)} \alpha_p^{(r)} (x_p - \mu)^2 \quad (10)$$

We call the value $\mu = \mu^{(\infty)}$ to which this process converges the *modified weighted fuzzy expected value* (MWFEV). An initial value for the *spread parameter* σ can be set at 1/4 of the average distance between cluster centers that we obtain with the k-means algorithm. We find prototypical MWFEV vectors via *componentwise* MWFEVs. The *weighted fuzzy variance* (WFV) is σ^2.

Figure 2 shows an example of the MWFEV versus the mean and the median for a 2-dimensional example of 5 vectors (1,2), (2,2), (1,3), (2,3) and

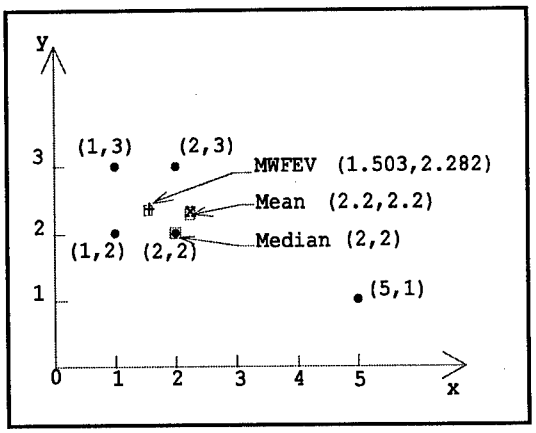


Fig. 2. An MWFEV, mean and median.

(5,1). The single outlier (5,1) influences the mean (2.2,2.2) and the median (2,2) strongly, but it affects the MWFEV vector (1.503,2.282) only slightly (a little more along the y-axis because it is closer in that dimension). The MWFEV is more densely situated in a cluster of vectors.

4.2 Our Fuzzy Clustering Algorithm

To implement the MWFEV on a set of vectors, we find the MWFEV over each component. Our new *fuzzy clustering* algorithm is

Step 1: Use predicted states as initial cluster centers and run the usual k-means algorithm (see [2], for example) to get initial clusters.

Step 2: Compute σ_k values for each cluster k.

Step 3: Compute all MWFEVs vectors $\mathbf{v}^{(k)}$.

Step 4: Assign all $\mathbf{x}^{(q)}$ to clusters of nearest centers.

Step 5: If a cluster has changed then go to Step 2.

Step 6: For each k, put $\mathbf{v}^{(k)}$ into the central tracks as fused state k.

The *Xie-Beni clustering validity* [11] is a product of compactness and separation defined by

$$\nu = \{(1/Q)\Sigma_{(k=1,K)}\,\sigma_k\}\{1/D_{min}\}^2 \qquad (11)$$

$$\sigma_k = \Sigma_{(q=1,Q)}\, w_{qk}\|\mathbf{x}^{(q)} - \mathbf{v}^{(k)}\|^2 \qquad (12)$$

D_{min} is the minimum distance between the cluster centers. Each σ_k is a fuzzy weighted mean-square error. We modify the Xie-Beni validity measure to sum over only the members of the kth cluster for each σ_k (instead of all vectors) and use this as a measure of the goodness of clustering.

5. Results and Conclusions

Figure 3 presents the 3 actual trajectories and also the 3 noisy trajectories as seen from the viewpoint of RS1 at (10,50,0) (kilometers). Figure 4 shows these from the perspective of RS2 at (100,5,0).

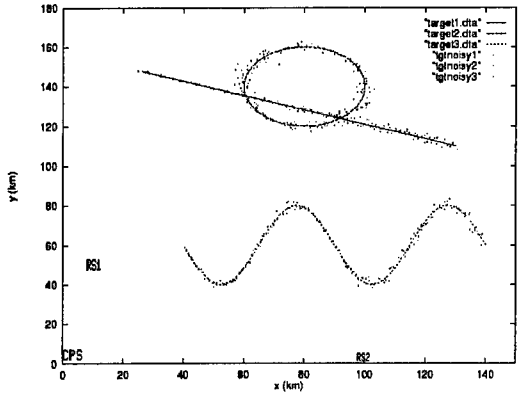


Figure 3. Trajectories seen by RS1.

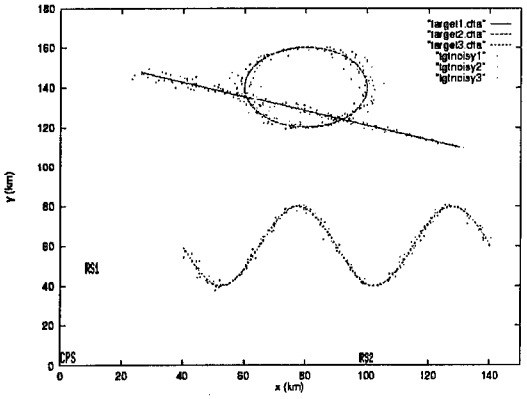


Figure 4. Trajectories seen by RS2.

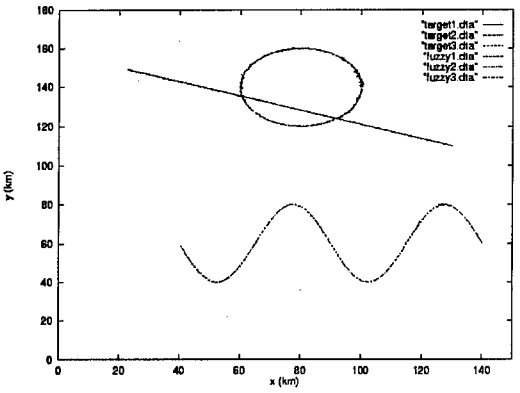


Figure 5. Fuzzy clustering results.

The standard deviations for the Gaussian white noises for these runs were

$$\sigma_r = 100 \text{ km}, \ \sigma_\theta = 1°, \ \sigma_\phi = \sigma_\theta$$

We note that the $\alpha\beta\gamma$ filters at RS1 and RS2 smoothed the noisy trajectories. It used a value of $\xi = 0.8$, which yields moderately strong smoothing. For example, from Equation (6a) we see that this value of ξ gives the smoothing value $\alpha = 0.36$ for use in Equation (3). However, at the CPS the association must be done via the fuzzy clustering with the fusing being provided as a by-product. In this case, the association was correct and the fusion reduced the noise level.

This first study was more of a test of the feasibility of the fuzzy clustering for use in tracking. There remain several detailed problems to be worked out in the association decision making. We continue to work on this approach and expect to have more to report in the future.

References

[1] K. C. Chang, R. K. Saha and Y. Bar-Shalom, "On optimal track-to-track fusion," *IEEE Trans. Aerospace and Electronic Systems*, vol. 33, no. 4, 1271-1275, 1997.

[2] C. Looney, *Pattern Recognition Using Neural Networks*, Oxford University Press, N.Y., 1997.

[3] C. Looney, "Advances in feedforward neural networks: demystifying knowledge acquiring black boxes," *IEEE Trans. Knowledge and Data Engineering*, vol. 8, no. 2, 211-226, 1997.

[4] C. Looney, "Fuzzy clustering: a new algorithm," *Proc. Thirteenth International Conf. on Systems Engineering*, Las Vegas, 1999.

[5] B. R. Mahafza, *Introduction to Radar Analysis*, CRC Press, Boca Raton, 1998.

[6] S.-T. Park and J. G. Lee, "Design of a practical tracking algorithm with radar measurements," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 34, 1337-1344, 1998.

[7] R. K. Saha and K. C. Chang, "An efficient algorithm for multisensor track fusion," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 34, no. 1, 200 - 210, 1998.

[8] M. Schneider and M. Craig, "On the use of fuzzy sets in histogram equalization," *Fuzzy Sets & Syst.*, vol. 45, 271-278, 1992.

[9] C. C. Schooler, "Optimal α-β filters for systems with modeling inaccuracies," *IEEE Trans. Aerospace and Electronic Systems*, vol. 11, no. 6, 1300-1306, 1975.

[10] G. Van Keuk, "Sequential Track Extraction," *IEEE Trans. on Aerospace and Electronics Systems*, vol. 34, 1134-1148, 1998.

[11] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, 841-847. 1991.

[12] Y. Zhang, H. Leung, M. Blanchette, T. Lo and J. Litva, "An efficient decentralized multiradar multitarget tracker for air surveillance," *IEEE Trans. Aerospace and Electronic Systems*, vol. 33, no. 4, 1357-1363, 1997.

Multitarget Tracking Using an IMM Estimator with Debiased Measurements for AEW Systems

T. Kirubarajan* and Y. Bar-Shalom*
Electrical Engineering Department
University of Connecticut
Storrs CT 06269-2157, USA
ybs@ee.uconn.edu

R. McAllister, R. Schutz and B. Engelberg
Surveillance Battle Management Systems Group
Northrop Grumman Corporation
Bethpage NY 11714, USA
mcallri@mail.northgrum.com

Abstract *In this paper we present an Interacting Multiple Model (IMM) estimator for tracking the motion of a large number of aircraft using the measurements obtained from an airborne sensor. The scenario under consideration, which is part of a study on Airborne Early Warning (AEW) weapon systems, consists of about 120 targets, whose motions evolve in a wide variety of ways, for example, benign constant velocity, constant acceleration, weaving and coordinated turns with accelerations up to 6g. The measurements consist of range, azimuth and range rate. This AEW scenario presents a challenging environment to work with due to long sampling intervals, high measurement errors, close target formations and high maneuvers. The IMM estimator is used in conjunction with an assignment algorithm for data association. It is shown that the IMM/Assignment estimator yields significantly better results, in all measures of performance, than those obtained with a single Kalman filter (with a similar assignment) on the same problem.*

Keywords: Multitarget tracking, state estimation, data association, assignment algorithm, Airborne Early Warning (AEW) weapon systems.

1 Introduction

The problem of tracking a large number of aircraft with varying motion parameters was considered in [9]. In [4] a benchmark tracking problem, where it was required to track six different aircraft with widely different maneuver characteristics using a single estimator, was considered. In these problems, the Interacting Multiple Model (IMM) estimator [2] has been shown to be very effective. Other applications of the IMM estimator can be found in [1]. In the IMM estimator, it is assumed that, at any time, the target trajectory evolves according to one of a finite number of models, which differ in their noise levels and/or structures. The system model is assumed to evolve according to a Markov chain. By probabilistically combining the estimates of the individual filters, typically Kalman or Extended Kalman, matched to these modes, an overall estimate is found [2].

In [9] it was demonstrated that the IMM estimator, in conjunction with a two-dimensional assignment, performs well enough to handle hundreds of civilian air targets. It was also shown that two-dimensional assignment (association between the list of established tracks and the list of measurements from the latest scan) is sufficient for civilian air traffic control. Further, an IMM estimator containing a (nonlinear) coordinated turn model performed better than that with only linear motion models. In this paper, we present the development of an IMM/Assignment estimator for tracking more than a hundred highly maneuvering (military) air targets. The scenario under consideration, which is part of a study on Airborne

[1]Supported by Northrop Grumman Contract C996549, ONR Grant N00014-97-1-0502 and AFOSR Grant 49620-97-1-0198.

Early Warning (AEW) weapon systems, consist of about 120 targets, whose motions evolve in a wide variety of ways, for example, benign constant velocity, constant acceleration, weaving and coordinated turns with accelerations up to 6g. The measurements consist of range, azimuth and range rate. This AEW scenario presents a challenging environment to work with due to long sampling intervals, high measurement errors, close target formations and high maneuvers.

The measurements, which are obtained from an airborne sensor, consist of range, azimuth and range rate. In order to keep the estimator Cartesian, where the target motion is better modeled, the measurements are transformed from polar to Cartesian. Due to large measurement errors and long sensor-to-target distances, the standard conversion introduces a bias, which is not negligible. In order to rectify this, the recently developed multiplicative debiasing is employed before using the converted measurements in the estimator [6].

The performance metrics used in the study are the total track life, which gives the percentage of frames during which a target is tracked by an acceptable track segment (subject to certain longevity and purity constraints) and the mean track life, which is the average life of acceptable track segments for a given target. In addition, RMS position/velocity errors are also used. These are evaluated globally as well as for individual targets. Estimation results indicate that the IMM/Assignment estimator yields significantly better results, in all measures of performance, than those obtained with a single Kalman filter (with a similar assignment) on the same problem [8].

This paper is organized as follows. In Section 2, the measurements obtained from the sensor, the converted measurements and their error statistics are discussed. The IMM estimator and the data association via assignment are discussed in Sections 3 and 4, respectively. Estimation results are presented in Section 5.

2 Scenario

The measurements are obtained from an airborne sensor with varying revisit intervals. The number of detections in scan k is denoted by $M(k)$. The m-th detection report ($1 \leq m \leq M(k)$) consists of a time stamp t_{m_k}, the measurement vector $\mathbf{z}(t_{m_k})$, and the sensor state $\mathbf{x}_p(t_{m_k})$ at time t_{m_k}. Note that the time stamps for different measurements within the same scan may be different.

Let the m-th measurement in scan k be from the n-th target and the true state of the n-th target at time t_{m_k} be defined by the 4-dimensional vector

$$\mathbf{x}^n(t_{m_k}) \triangleq \begin{bmatrix} \xi^n(t_{m_k}) \\ \dot{\xi}^n(t_{m_k}) \\ \eta^n(t_{m_k}) \\ \dot{\eta}^n(t_{m_k}) \end{bmatrix}' \tag{1}$$

where $\xi^n(t_{m_k})$ and $\eta^n(t_{m_k})$ are the distances of the target in the X and Y directions respectively from a reference point (origin). The corresponding velocities are $\dot{\xi}^n(t_{m_k})$ and $\dot{\eta}^n(t_{m_k})$, respectively. The state of the sensor platform, which is known, is defined similarly by $\mathbf{x}_p(t_{m_k})$. In the following, only the scan index k is kept while the other indices m and n have been dropped for simplicity. Also, the sensor-to-target range is defined as

$$r(\mathbf{x}(t_k)) = \sqrt{r_\xi^2(\mathbf{x}(t_k)) + r_\eta^2(\mathbf{x}(t_k))} \tag{2}$$

where $r_\xi(t_k))$ and $r_\eta(t_k))$ are the relative position components of the target at time t_k with respect to the platform in the X and Y directions, respectively.

Then, the range rate, $\dot{r}(\mathbf{x}(t_k))$, of the target is given by

$$\begin{aligned} \dot{r}(\mathbf{x}(t_k)) &= \left(\dot{\xi}(t_k) - \dot{\xi}_p(t_k)\right) \cos\theta(\mathbf{x}(t_k)) \\ &\quad + \left(\dot{\eta}(t_k) - \dot{\eta}_p(t_k)\right) \sin\theta(\mathbf{x}(t_k)) \end{aligned} \tag{3}$$

where

$$\theta(\mathbf{x}(t_k)) = \tan^{-1}\left(\frac{\eta(t_k) - \eta_p(t_k)}{\xi(t_k) - \xi_p(t_k)}\right) \tag{4}$$

The error statistics for the sensor measurements are given in terms of the range standard

deviation σ_r, range rate standard deviation $\sigma_{\dot{r}}$ and the azimuth standard deviation σ_θ, which are known. That is, the received range $r_z(t_k)$, azimuth $r_z(t_k)$ and range rate $\dot{r}_z(t_k)$ measurements are given by

$$\begin{aligned} r_z(t_k) &= r(\mathbf{x}(t_k)) + \mathcal{N}(v_r, 0, \sigma_r^2) \quad (5) \\ \theta_z(t_k) &= \theta(\mathbf{x}(t_k)) + \mathcal{N}(v_\theta, 0, \sigma_\theta^2) \quad (6) \\ \dot{r}_z(t_k) &= \dot{r}(\mathbf{x}(t_k)) + \mathcal{N}(v_{\dot{r}}, 0, \sigma_{\dot{r}}^2) \quad (7) \end{aligned}$$

where $\mathcal{N}(v, \bar{v}, \sigma^2)$ denotes the measurement noise v with mean $\bar{v}$ and standard deviation σ.

Note that the measurements are in polar coordinates whereas a target's motion is better modeled in Cartesian [3] for the estimator. This necessitates the transformation of the received range-azimuth measurements into X-Y position measurements. It has been shown that this nonlinear transformation introduces a bias and that a debiasing technique is required to compensate [3, 5, 6]. This is especially true in the presence of large measurement errors and long sensor-to-target distances, as in the present problem. In order to rectify this, the recently developed multiplicative debiasing is employed before using the converted measurements in the estimator [6]. This exact multiplicative debiasing technique is preferred over the previously proposed additive debiasing technique [5], due to the former's superior consistency and robustness.

The unbiased converted measurement vector $\mathbf{z}(t_k)$ is given by [6]

$$\mathbf{z}(t_k) = \begin{bmatrix} \lambda_\theta^{-1} r_z(t_k) \cos\theta_z(t_k) \\ \lambda_\theta^{-1} r_z(t_k) \sin\theta_z(t_k) \\ \dot{r}_z(\mathbf{x}(t_k)) \end{bmatrix} \quad (8)$$

where

$$\lambda_\theta = E\{\cos v_\theta\} = e^{-\sigma_\theta^2/2} \quad (9)$$

With these, the position variances in the respective directions and the covariance are given by [6]

$$\begin{aligned} \sigma_\xi^2(t_k) &= \left(\lambda_\theta^{-1} - 2\right) r_z^2(t_k) \cos^2\theta_z(t_k) + \\ &\quad \frac{1}{2}\left(r_z^2(t_k) + \sigma_r^2\right)\left(1 + \lambda_\theta' \cos 2\theta_z(t_k)\right) \quad (10) \\ \sigma_{\xi\eta}^2(t_k) &= \frac{1}{2}\left(\lambda_\theta^{-1} - 2\right) r_z^2(t_k) \sin 2\theta_z(t_k) + \\ &\quad \frac{1}{2}\left(r_z^2(t_k) + \sigma_r^2\right) \lambda_\theta' \sin 2\theta_z(t_k) \quad (11) \\ \sigma_\eta^2(t_k) &= \left(\lambda_\theta^{-1} - 2\right) r_z^2(t_k) \sin^2\theta_z(t_k) + \\ &\quad \frac{1}{2}\left(r_z^2(t_k) + \sigma_r^2\right)\left(1 - \lambda_\theta' \sin 2\theta_z(t_k)\right) \quad (12) \end{aligned}$$

where

$$\lambda_\theta' = E\{\cos 2v_\theta\} = e^{-2\sigma_\theta^2} \quad (13)$$

and, thus, the measurement covariance matrix $R(t_k)$ is given by

$$R(t_k) = \begin{bmatrix} \sigma_\xi^2(t_k) & \sigma_{\xi\eta}^2(t_k) & 0 \\ \sigma_{\xi\eta}^2(t_k) & \sigma_\eta^2(t_k) & 0 \\ 0 & 0 & \sigma_{\dot{r}}^2 \end{bmatrix} \quad (14)$$

3 Estimator

In order to handle the various maneuvering characteristics of different targets, which range from a benign (almost) constant velocity motion to high maneuvers of 6g, an IMM estimator consisting of a number of EKF filter modules is used. Typical models used in the IMM estimator include a (nearly) constant velocity model, a (nearly) constant acceleration model and a coordinated turn model [2, 9].

Using a direct discrete time kinematic model [2] and assuming linear motion in the X-Y coordinate frame, the evolution of the true target state $\mathbf{x}(t_k)$, which was defined in (1), can be written as

$$\mathbf{x}(t_k) = F(\delta_k)\mathbf{x}(t_{k-1}) + \Gamma(\delta_k) v(t_{k-1}) \quad (15)$$

where t_k is the time of the k-th scan, $\delta_k = t_k - t_{k-1}$ and v_{t_k} is the white Gaussian process noise sequence with covariance $Q(\delta_k)$.

Because of the nonlinearity in the state-to-measurement relationship, the target originated measurement can be written as

$$\mathbf{z}(t_k) = h\left(\mathbf{x}(t_k)\right) + w(t_k) \quad (16)$$

where

$$h(\mathbf{x}(t_k)) = \begin{bmatrix} \xi(t_k) \\ \eta(t_k) \\ \dot{r}(\mathbf{x}(t_k)) \end{bmatrix} \quad (17)$$

and $\dot{r}(\cdot)$ is defined in (3). The white Gaussian measurement noise sequence $w(t_k)$ is independent of $v(t_k)$ and its covariance $R(t_k)$ is given in (14).

The above nonlinearity means that the standard Kalman filter cannot be used for state estimation either as a stand-alone filter or as an IMM estimator module — i.e., an extended Kalman filter (EKF), which uses first or second order series expansion to linearize the measurement equation, is required [2]. For a first order EKF, the state can be estimated as follows[1]:

The predicted state $\hat{\mathbf{x}}_e(t_k^-)$ at time t_k is

$$\hat{\mathbf{x}}_e(t_k^-) = F(\delta_k)\hat{\mathbf{x}}_e(t_{k-1}) \quad (18)$$

and the associated predicted state covariance $P_e(t_k^-)$ is

$$\begin{aligned} P_e(t_k^-) &= F(\delta_k)P_e(t_{k-1})F(\delta_k)' + \\ &\quad \Gamma(\delta_k)Q(\delta_k)\Gamma(\delta_k)' \end{aligned} \quad (19)$$

where $\hat{\mathbf{x}}_e(t_{k-1})$ is the state estimate from time (t_{k-1}) and $P_e(t_{k-1})$ is the associated covariance evaluated below using (28).

The predicted measurement $\hat{\mathbf{z}}_e(t_k^-)$ is given by

$$\hat{\mathbf{z}}_e(t_k^-) = h\left(\hat{\mathbf{x}}_e(t_k^-)\right) \quad (20)$$

and the associated innovation covariance is

$$S_e(t_k) = H(t_k)P_e(t_k^-)H(t_k)' + R(t_k) \quad (21)$$

where $H(t_k)$ is the Jacobian of $h(\cdot)$ evaluated as

$$\begin{aligned} H(t_k) &= \left[\nabla_{\mathbf{x}} h(\mathbf{x})'\right]'\Big|_{\mathbf{x}=\hat{\mathbf{x}}_e(t_k^-)} \\ &= \begin{bmatrix} 1 & 0 & \sin\theta(\mathbf{x})l(\mathbf{x}) \\ 0 & 0 & \cos\theta(\mathbf{x})) \\ 0 & 1 & -\cos\theta(\mathbf{x})l(\mathbf{x}) \\ 0 & 0 & \sin\theta(\mathbf{x}) \end{bmatrix}'\Bigg|_{\mathbf{x}=\hat{\mathbf{x}}_e(t_k^-)} \end{aligned} \quad (22)$$

[1] A more comprehensive treatment of the Kalman filter, EKF and the IMM estimator can be found in, e.g., [2]. The equations are provided here for completeness and to introduce the notations for later use.

with

$$\begin{aligned} l(\mathbf{x}) &= \frac{\left(\dot{\xi}(t_k) - \dot{\xi}_p(t_k)\right)\sin\theta(\mathbf{x})}{r(\mathbf{x})} \\ &\quad - \frac{(\dot{\eta}(t_k) - \dot{\eta}_p(t_k))\cos\theta(\mathbf{x})}{r(\mathbf{x})} \end{aligned} \quad (23)$$

and, $\theta(\cdot)$ and $r(\cdot)$ are defined in (4) and (2), respectively.

As in the standard Kalman filter, the state estimate is updated using

$$\hat{\mathbf{x}}_e(t_k) = \hat{\mathbf{x}}_e(t_k^-) + W_e(t_k)\ \nu_e(t_k) \quad (24)$$

where $W_e(t_k)$ is the filter gain given by

$$\begin{aligned} W_e(t_k) &= P_e(t_k^-)H(t_k)' \cdot \\ &\quad \left[H(t_k)P_e(t_k^-)H(t_k)' + R(t_k)\right]^{-1} \quad (25) \\ &= P_e(t_k^-)H(t_k)'S_e(t_k)^{-1} \quad (26) \end{aligned}$$

and

$$\nu_e(t_k) = \mathbf{z}(t_k) - \hat{\mathbf{z}}_e(t_k^-) \quad (27)$$

is the measurement residual or the innovation. The covariance matrix associated with the state estimate is given by

$$P_e(t_k) = P_e(t_k^-) - W_e(t_k)\ S_e(t_k)\ W_e(t_k)' \quad (28)$$

In order to model the non-maneuvering intervals, one can use a (second order) piecewise constant white noise acceleration model (WNA, with two position and two velocity components) with

$$F(\delta_k) = \begin{bmatrix} 1 & \delta_k & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \delta_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (29)$$

$$\Gamma(\delta_k) = \begin{bmatrix} \delta_k^2/2 & 0 \\ \delta_k & 0 \\ 0 & \delta_k^2/2 \\ 0 & \delta_k \end{bmatrix} \quad (30)$$

with low process noise. For on-going maneuvers, the same model with high process noise can be used [3].

It has been shown that the so-called coordinated turn model, where the turn rate $\omega(t_k)$

of the target is an additional state component, yields better results in estimating the motion of highly maneuvering targets during maneuvers [9]. In this case, the state $\mathbf{x}(t_k)$ is given by

$$\mathbf{x}(t_k) = \begin{bmatrix} \xi(t_k) \\ \dot{\xi}(t_k) \\ \eta(t_k) \\ \dot{\eta}(t_k) \\ \omega(t_k) \end{bmatrix}' \tag{31}$$

$$= \begin{bmatrix} 1 & \frac{\sin\Omega(t_{k-1})\delta_k}{\Omega(t_{k-1})} & 0 & -\frac{1-\cos\Omega(t_{k-1})\delta_k}{\Omega(t_{k-1})} & 0 \\ 0 & \cos\Omega(t_{k-1})\delta_k & 0 & -\sin\Omega(t_{k-1})\delta_k & 0 \\ 0 & \frac{1-\cos\Omega(t_{k-1})\delta_k}{\Omega(t_{k-1})} & 1 & \frac{\sin\Omega(t_{k-1})\delta_k}{\Omega(t_{k-1})} & 0 \\ 0 & \sin\Omega(t_{k-1})\delta_k & 0 & \cos\Omega(t_{k-1})\delta_k & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{x}(t_{k-1}) + \begin{bmatrix} \frac{1}{2}\delta_k^2 & 0 & 0 \\ \delta_k & 0 & 0 \\ 0 & \frac{1}{2}\delta_k^2 & 0 \\ 0 & \delta_k & 0 \\ 0 & 0 & 1 \end{bmatrix} v(t_{k-1}) \tag{32}$$

$$= F_{CT}(\delta_k)\mathbf{x}(t_{k-1}) + \Gamma_{CT}(\delta_k)v(t_{k-1}) \tag{33}$$

Note that for the coordinated turn model, $F(\delta_k)$ in (19) is replaced with the Jacobian of $F_{CT}(\delta_k)$ [2]. The initial turn rate is assumed to be zero for the coordinated turn filter module [2].

4 Data Association

In multitarget tracking with non-unity target detection probability P_D and spurious measurements (non-zero false alarm probability P_{FA}), it is necessary to decide which one of the received measurements should be used to update a particular track — one needs a mechanism for measurement-to-track data association [3, 9].

Two dimensional assignment, where the association is performed between the latest list of measurements in frame (scan) k and the track list from $k-1$, is one of the data association algorithms which has been used successfully in large scale tracking problems [9]. The basic idea behind 2-D assignment is that the measurements from $\mathcal{M}(k)$ are matched (deemed to have come from) the tracks in $\mathcal{T}(k-1)$ by formulating the matching as a constrained global optimization problem. The optimization is carried out to minimize the global "cost" of associating (or not associating) the measurements to tracks.

To present the 2-D assignment, define a binary assignment variable $a(k,m,n)$such that

$$a(k,m,n) = \begin{cases} 1 & \mathbf{z}(t_{m_k}) \text{ is assigned} \\ & \text{to track } \mathcal{T}^n(k-1) \\ 0 & \text{otherwise} \end{cases} \tag{34}$$

A set of complete assignments, which consists of the associations of all the measurements in $\mathcal{M}(k)$ and the tracks in $\mathcal{T}(k-1)$, is denoted by $\mathbf{a}(k)$, i.e.,

$$\mathbf{a}(k) = \{a(k,m,n); m = 0,1,\ldots,M(k) \tag{35}$$

$$n = 0,1,\ldots,N(k-1)\} \tag{36}$$

where $M(k)$ and $N(k-1)$ are the cardinalities of the measurement and track sets, respectively. The indices $m = 0$ and $n = 0$ correspond to the non-existent (or "dummy") measurement and track, which are used with a special meaning in the assignment problem — the assignment $a(k,0,n)$ denotes the event that track $\mathcal{T}^n(k)$ is not associated with any of the measurements in $\mathcal{M}(k)$. In this case, the track $\mathcal{T}^n(k)$ is said to have been associated with the non-existent dummy measurement. Similarly, $a(k,m,0)$ corresponds to the event that measurement m is not associated with any of the existing tracks in $\mathcal{T}(k-1)$ — the measurement is associated with the dummy track. The "dummy" notation is used to formulate the assignment problem in a uniform manner, where the non-association possibilities are also considered, making it computer-solvable.

The objective of the assignment is to find the optimal assignment $\mathbf{a}^*(k)$ which minimizes the global cost of association

$$C(k|\mathbf{a}(k)) = \sum_{m=0}^{M(k)} \sum_{n=1}^{N(k-1)} a(k,m,n)c(k,m,n) \tag{37}$$

where $c(k,m,n)$ is the cost of the assignment $a(k,m,n)$.

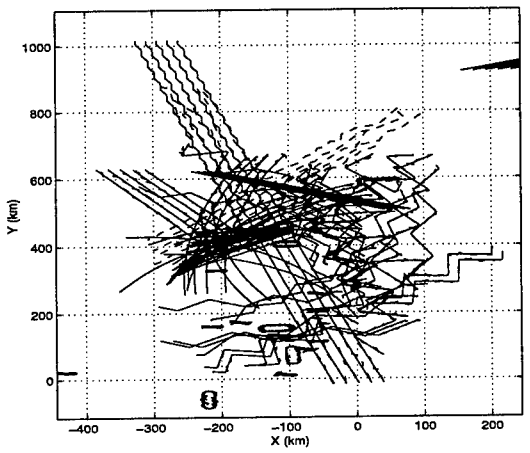


Figure 1: AEW Scenario (— Ground truth, - - - Measurements)

The costs $c(k,m,n)$ are derived from the dimensionless global likelihood ratio of the measurements conditioned on a particular assignment [9]. The best assignments are obtained using the modified Auction Algorithm.

5 Results

Target trajectories and the corresponding position measurements are shown in Figure 1. There are 120 targets in the surveillance region and, due to measurement corruption (misalignment), only 115 of those are considered for tracking. It can be seen that the targets undergo a wide variety of maneuver modes, including benign constant velocity, constant acceleration, weaving and coordinated turns. The measurements error standard deviations are $\sigma_{\rm r} = 0.0323{\rm nm} = 60{\rm m}$, $\sigma_\theta = 0.8^\circ$ and $\sigma_{\dot r} = 8.8{\rm kts} = 4.5{\rm m/s}$. The target detection and false alarm probabilities are given by $P_D = 0.8$ and $P_{FA} = 10^{-6}/{\rm cell}$, respectively.

The IMM estimator consisted of three filter modules, namely:

1. Constant velocity model (M^1) – WNA low process noise, which corresponds to the non-maneuvering intervals of the target trajectory.

2. Maneuver model (M^2) – WNA with high process noise, which corresponds to ongoing maneuvers.

3. Coordinated turn (CT) model (M^{CT}), which corresponds to maneuvering turn intervals.

For the above modules, the mode transition probabilities $p_{ii}(k)$ at time t_k are calculated (modeled) as follows. First

$$p_{ii} = \max\{l_i, 1 - \frac{\delta_k}{\tau_i}\} \tag{38}$$

where $l_i = 0.25$ is the lower limit for the $i^{\rm th}$ model transition probability δ_k is the revisit interval [2, 4]. The other elements of the transition matrix are calculated using

$$\begin{aligned} p_{12} &= 0.6(1-p_{11}) \quad p_{13} = 0.4(1-p_{11}) \\ p_{21} &= 0.9(1-p_{22}) \quad p_{23} = 0.1(1-p_{22}) \\ p_{31} &= 0.9(1-p_{33}) \quad p_{32} = 0.1(1-p_{33}) \end{aligned}$$

In [9] the so-called directional process noise model, where the target motion is assumed to have a higher lateral uncertainty (acceleration) than axial, was shown to be more appropriate for air targets than the standard EKF model with equal uncertainties in X-Y directions. To track the AEW targets, the directional process noise models with axial and lateral acceleration standard deviations σ_{a_r} and σ_{l_r} for the r-th model, respectively, are used for the M^1, M^2, and $M^{\rm CT}$ (linear motion portion only). These values are given by

$$\begin{aligned} \sigma_{a_1} &= 0.2{\rm m/s}^2, \sigma_{l_1} = 0.2{\rm m/s}^2 \\ \sigma_{a_2} &= 5{\rm m/s}^2, \sigma_{l_2} = 20{\rm m/s}^2 \\ \sigma_{a_3} &= 1{\rm m/s}^2, \quad \sigma_{l_3} = 5{\rm m/s}^2 \end{aligned} \tag{39}$$

Also, the coordinated turn model assumed a turn rate process noise standard deviation of $\sigma_\omega = 0.3^\circ/{\rm s}^2$.

Previously, estimation results obtained using a single Combined Kalman Filter (CKF) in conjunction with the JVC algorithm[2]

[2]The JVC algorithm is equivalent to the Auction. As shown in [7] the Auction is faster than JVC for problems with sparsity above 80% (when only 20% of the assignments are feasible). Assignment problems in tracking generally have sparsities exceeding 90%.

for data association were presented in [8]. Here we present the results obtained with the IMM/Assignment estimator to illustrate the advantages of multiple model estimation. These advantages result from the provision in the IMM estimator for the target motion model to "softly switch" from one model to another, which makes the IMM into an *adaptive bandwidth* filter. This adaptive bandwidth capability is the key requirement for a maneuvering target tracking filter for good performance.

The main performance metric is track continuity or track purity, which quantifies better estimation and association. Track continuity is measured in terms of Mean Track Life (MTL) and Total Track Life (TTL). These two statistics are based on track segments that are, at least, six scans long (longevity), have been updated within the last seven scans (continuity), have been updated by a particular target at least 45% of the time (purity) and do not have consecutive updates by more than two other targets (purity). Total track life is then the total number of times (scans) a target has been tracked by some track segment divided by the total number of scans the target has been in the scenario. Mean track life is the total track life divided by the number of segments that the target has been tracked by [8].

The IMM estimation results are compared with the baseline results presented in [8] in Figures 2 and 3. It can be seen that the IMM estimator yields *uniformly* better results than the results obtained by using a Kalman Filter. This indicates better measurement-to-track association and estimation (fewer broken tracks: from 68% to 12% for the difficult targets).

The RMS position and velocity errors obtained with the IMM/Assignment and CKF/JVC estimators are shown in Figures 4 and 5 respectively. It can be seen that the IMM estimator improves the position estimation errors by a factor of 1.5–2 over the Kalman filter — using its adaptive bandwidth capability, the IMM estimator is able handle targets with different maneuver parameters and yields better estimation results. In velocity also, the IMM estimator results in reduced estimation errors.

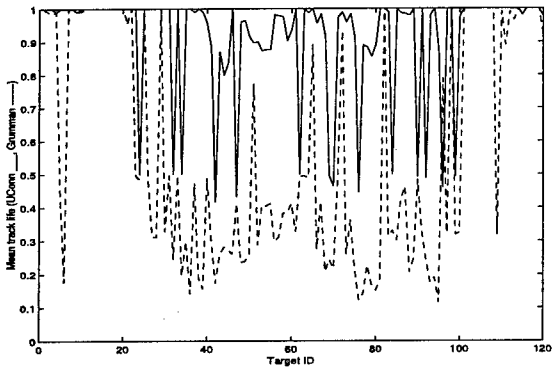


Figure 2: Mean track life
(— IMM/Assignment, - - - CKF/JVC).

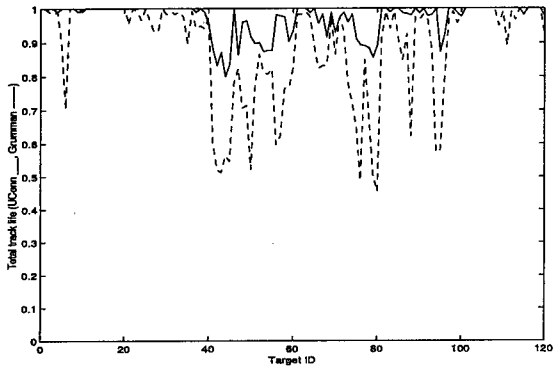


Figure 3: Total track life
(— IMM/Assignment, - - - CKF/JVC).

6 Conclusions

In this paper, we presented the development and implementation of a tracker based on the IMM/Assignment estimator for tracking the motion of a large number of air targets with different motion characteristics. The measurements, which included range, azimuth and range rate, were obtained from an airborne sensor. Because of the desire to keep the estimator in Cartesian coordinates, the polar measurements were converted into Cartesian and the resulting conversion bias (due to large measurement errors and long sensor-to-target distances) was compensated for using a multiplicative debiasing technique. Performance metrics in terms of track continuity/purity and position/velocity estimation errors indi-

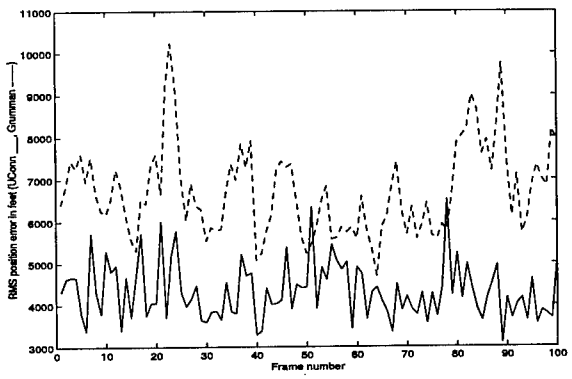


Figure 4: RMS position errors
(— IMM/Assignment, - - - CKF/JVC).

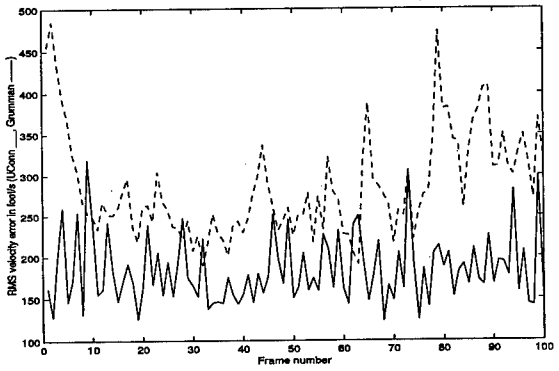


Figure 5: RMS velocity errors
(— IMM/Assignment, - - - CKF/JVC).

cate that the IMM/Assignment estimator performs significantly better than the previously published CKF/JVC estimator. This improvement results from the adaptive bandwidth capability of the IMM estimator.

References

[1] Y. Bar-Shalom (Editor), *Multitarget-Multisensor Tracking: Advanced Applications,* Volumes I and II, Artech House, Dedham, MA, 1990 and 1992. Reprinted by YBS Publishing, Storrs, CT, 1999.

[2] Y. Bar-Shalom and X. R. Li, *Estimation and Tracking: Principles, Techniques and Software.* Dedham, MA: Artech House, 1993. Reprinted by YBS Publishing, Storrs, CT, 1999.

[3] Y. Bar-Shalom and X. R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques.* Storrs, CT: YBS Publishing, 1995.

[4] T. Kirubarajan, Y. Bar-Shalom, W. D. Blair and G. A. Watson, "IMMPDA solution to benchmark for radar resource allocation and tracking in the presence of ECM", *IEEE Trans. Aerospace and Electronic Systems*, Vol. 34, No. 3, pp. 1015-1022, July, 1998.

[5] D. Lerro and Y. Bar-Shalom, "Tracking with debiased consistent converted measurements versus EKF", *IEEE Trans. Aerospace and Electronic Systems*, Vol. 29, No. 4, pp. 1115-1134, July, 1993.

[6] M. Longbin, S. Xiaoquan, Z. Yizu, S. Z. Kang and Y. Bar-Shalom, "Unbiased converted measurements for tracking", *IEEE Trans. Aerospace and Electronic Systems*, Vol. 34, No. 3, pp. 1016-1022, July, 1998.

[7] R. L. Popp, K. R. Pattipati and Y. Bar-Shalom, "Dynamically adaptable m-best 2D assignment algorithm and multi-level parallelization for multitarget tracking", To appear in *IEEE Trans. Aerospace and Electronic Systems*, 1999.

[8] R. Schutz, R. McAllister and B. Engelberg, "Combined Kalman filter and JVC algorithms for AEW target tracking applications" *Proc. SPIE Conf. Signal and Data Processing of Small Targets*, Vol. 3163, San Diego, CA, July, 1997.

[9] H. Wang, T. Kirubarajan and Y. Bar-Shalom, "Precision large scale air traffic surveillance using IMM/Assignment estimators", *IEEE Trans. Aerospace and Electronic Systems*, Vol. 35, No. 1, pp. 255-266, January, 1999.

Session TB4
Classification I
Chair: Nageswara S. V. Rao
Oak Ridge National Laboratory, TN, USA

Numerical and Implementation Studies of Conditional and Relational Event Algebra, Illustrating Use and Comparison with Other Approaches to Modeling of Information

M. J. George
Houston Associates
10675 Treena St., Suite 103,
San Diego, CA 92131

I.R. Goodman
SSC-SD,
Code D44215 Topside, Bldg. A33
San Diego, CA 92152

Abstract – *Conditional event algebra (CEA) – and more generally, relational event algebra (REA) -- is a means for establishing a space of events – called "conditional events" – or "relational events"—whose probabilities yield corresponding ratios – or functions -- of probabilities (the former being conditional probabilities). This paper establishes an outline of procedures for both analyzing and implementing the numerical aspects of CEA and REA. Computations of probabilities of conjunctions of conditional events are considered via assignment of appropriate atomic probabilities. Tests of similarity hypotheses, applying CEA or REA, are detailed and certain of these tests are shown to be related to tail percentages of F- distributions. Alternative forms for conditional events from PSCEA, the product space form of CEA, are also considered, including truncated, probability dependent, and distributionally-derived. Furthermore, the metrics used in similarity hypotheses tests are determined for linear-weighted probability models via REA and are compared with the standard euclidean metric. For a relatively simple geometric setting and the choice of a natural loss function, the Minkowski or pointwise averaging of patterns is seen to produce significantly higher losses than an algebraic averaging procedure using REA.*

Key Words: conditional event algebra, relational event algebra, metrics, measures of similarity, combination of information

1. Introduction

This work is concerned with establishing a basis for investigating both the numerical and implementation aspects of *conditional event algebra* (CEA) *and relational event algebra* (REA) and is a summary and modification of an earlier effort [1].

Briefly stated, CEA -- and more generally, REA-- are new procedures which expand significantly the scope of applicability of traditional probability theory. (See [2- 4] for general background.) CEA, as considered here, and denoted, from now on as PSCEA (Product Space Conditional Event Algebra) in order to distinguish it from other approaches to CEA [3], results from the extending of a given probability space of ordinary or unconditional events to a larger countable infinite product probability space containing the given one in an isomorphic isometric imbedding sense. When such a construction is made, certain of the events in the larger space, can then be shown to be identifiable in a natural way as the algebraic counterparts of conditional probabilities. In turn, this allows for the development of a comprehensive and rigorous logic and calculus of boolean operations and relations among such conditional events. Even though the conditional events themselves in general are countable infinite disjoint disjunctions of simpler events, the CEA calculus of operations and relations here typically consists of *finite closed-form results.* One of the main applications of PSCEA has been to the modeling, comparing, and combining of inference rules (see again the above cited references).

Just as CEA (and PSCEA) was developed to answer the need for extending probabilistic techniques to address issues involving conditional probabilities, more generally, REA has been developed to permit sound analysis of probability-based models in the form of given functions of probabilities (other than just arithmetic divisions, which correspond to conditional probabilities and CEA). Some applications of REA have included the comparisons and combining of probability models in the form of linear and nonlinear combinations – representing, e.g., the modeling of expert opinion. In addition, it has been demonstrated that models initially in fuzzy logic form – such as those representing natural language descriptions – can be put into equivalent probability functional form, using the one-point random set coverage representation of fuzzy logic [4]. Thus, this class of models also can be addressed via REA. (See, again [2, 3].)

All of the above, however, up to this point, has not considered as a prime focus the associated problems of actual implementation of these new techniques, as well as specific numerical comparisons with parallel results obtained without their use. This paper considers these issues only to some degree, because of limited space and the requisite background needed here to establish, even in a minimal sense, the pertinent concepts involving general computations,

hypotheses testing, and estimation of event-based procedures.

For the implementation aspect, it is pointed out first that computations required for the probability evaluation of the conjunction of conditional events -- which plays a critical element in implementation -- can be approached via assignment of probabilities to appropriately chosen atoms. Second, it is seen that the actual calculations involved in computing one of the metrics and implementing basic testing of hypotheses between single events representing the models in question– utilizing REA (and Second Order Probabilities (SOP)) – can be related directly to tail percentage tables of F-distributions. Third, finite approximations to conditional events themselves (such events being in actuality an infinite disjunction of events), "empirical" conditional events or those possibly dependent upon particular choices of probability measures (unlike PSCEA), are considered as viable alternatives to full conditional events, in an attempt to reduce complexity of calculations. In a related direction, a distribution-based approach to conjunction of conditional events in PSCEA is also demonstrated. Fourth, probability metrics are evaluated for establishing measures of similarity between models in the form of weighted combinations of probabilities. Finally, it is shown for a relatively simple geometric setting and the choice of a natural loss function, that the well-known Minkowski or pointwise averaging of patterns produces significantly higher losses than a newly-proposed algebraic averaging procedure emanating from REA.

2. Summary of Event-Based Techniques for General Computations, Hypotheses Testing, and Estimation

2.1 Summary of PSCEA

As stated above, this paper is only concerned with the form of CEA known as PSCEA. For a brief history of CEA and background on various non-boolean structured CEA's, see [2, 3].

One basic application of PSCEA– as utilized in this paper -- is a rigorous basis for uniting, for the first time, classical deductive logic, commonsense reasoning, and probability logic [5]. Another use of PSCEA is to compare in a universal quantitative manner similarity and differences of inference rules, the validity of which is interpreted via naturally associated conditional probabilities. This also makes use of the tool SOP (Second Order Probabilities) -- to be explained later -- and the fact that all probability spaces can be made into (pseudo-) metric spaces using relatively simple unconditional and conditional probabilities involving the boolean symmetric sum operator – as considered, e.g., in [2]. Part III. A brief outline of this issue will be presented following the introduction of REA below.

Summarizing here only the essential properties, for any a, b, c, d,... in B, for given probability space (Ω,B,P) consider the countable infinite product probability space derived from it, (Ω_o,B_o,P_o), where

$$\Omega_o = \Omega\times\Omega\times\Omega\ldots , \qquad (2.1)$$

and *conditional events* (a|b), (c|d),... in B_o. Here, e.g., (a|b) is given directly and recursively, respectively, as

$$(a|b) = (ab \mid b) = \bigvee_{j=0}^{+\infty} (b')^j \times ab \times \Omega_o \qquad (2.2)$$

$$= (ab \times \Omega_o) \vee (b'\times(a|b)), \qquad (2.3)$$

with compatible evaluation, provided $P(b) > 0$,

$$P_o((a|b)) = \sum_{j=0}^{+\infty} (P(b'))^j P(ab) = P(ab)/P(b) = P(a|b). \qquad (2.4)$$

For purpose of convenience, all boolean operators and relations extending the usual ones for (Ω,B,P) to (Ω_o,B_o,P_o) are indicated by the same symbols when unambiguous. The natural (isomorphic, probability-preserving) imbedding of unconditionals as conditionals holds between any a in B and $(a|\Omega)$ in B_o

$$a \leftrightarrow (a|\Omega) = a\times\Omega_o , \quad P(a) = P_o((a|\Omega)). \qquad (2.5)$$

The imbedding in eq.(2.5) however is not an identity, and thus Lewis' triviality theorem is avoided (see [2], Sections 11.5 and 12.2.2.) Identification of conditional events with the null or universal events in B_o, as well as relating to each other over B_o, includes for any a, b, c, d in B:

$$(a|b) = \varnothing_o = (\varnothing|b) \text{ iff } ab=\varnothing ;$$
$$(a|b) = \Omega_o = (b|b) = (\Omega|\Omega) \text{ iff } a \geq b > \varnothing,$$

and for nontrivial (a|b), (c|d) in B_o, i.e., $\varnothing < a < b$ and $\varnothing < c < d$,

$$(a|b) \leq (c|d) \text{ iff } a \leq c \text{ and } c'd \leq a'b$$
$$\text{iff } P(a|b) \leq P(c|d), \text{ all P};$$
$$(a|b) = (c|d) \text{ iff } a = b \text{ and } c = d$$
$$\text{iff } P(a|b) = P(c|d), \text{ all P}. \qquad (2.6)$$

The following extensions of (.)', &, $\vee$ over B hold over B_o:

$$(a|b)' = (a'|b) ; \; P_o((a|b)') = 1-P(a|b) = P(a'|b),$$
$$P_o((a|b) \,\&(c|d)) = P_o(\alpha) / P(b\vee d),$$
$$P_o(\alpha) = P(abcd) + P(abd')P(c|d) + P(cdb')P(a|b),$$
$$P_o((a|b)\vee(c|d)) = P(a|b) + P(c|d) - P_o((a|b)\&(c|d)), \qquad (2.7)$$

provided P(b), P(d) >0. In fact, all laws of probability are respected relative to any conditional events in B_o and the actions of P_o. The conjunctive probability in eq.(2.7) for two arguments can be extended recursively to any finite number of arguments exceeding two: Using obvious multivariate notation, for arbitrary $\varnothing < a_j < b_j$ in B, j in J, where now $(a|b)_J$ indicates $(a_j|b_j)_{j \text{ in } J}$, $\&(b_K)$ indicates $\&(b_j)$, etc., first: define for any sets $\varnothing \subseteq K \subseteq J$:

$$\alpha((a,b)_{J,K}) = \&(b')_{J-K} \,\&\, (\&(a_K)) \text{ in } B,$$

$$P_o(\alpha_o((a,b)_J)) = \sum_{\varnothing \neq K \subseteq J} \left(P(\alpha((a,b)_{J,K})) \cdot P_o(\&(a|b)_{J-K})\right). \quad (2.8)$$

Then, the desired recusive relation using eq.(2.8) holds, where

$$P_o(\&(a|b)_J) = P_o(\alpha_o((a,b)_J)) \,/\, P(\vee(b_J)). \quad (2.9)$$

2.2 Summary of REA

REA, as stated above, extends PSCEA. Consider the role of PSCEA, where, given probability space and resulting conditional probabilities, i.e., arithmetic divisions of probabilities of events, such as

$$P(a|b) = P(ab)/P(b), \quad P(c|d) = P(cd)/P(d), \ldots \quad (2.10)$$

one determines an extension of (Ω,B,P) to probability space (Ω_o,B_o,P_o), via eq.(2.5) and conditional events, say (a|b), (c|d)... in B_o satisfying eq.(2.4) connecting numerical divisions of probabilities to algebraic "division". *In fact, note that the expression in eq.(2.2) is the complete algebraic analogue of the standard power series expansion of the formal division of events* $a / b = a/ (1-b')$. In turn, where desired, one then can compute probabilities of prescribed logical operators acting upon such conditional events. One such type of situation giving rise to this issue, as mentioned above, involves the computation of probability metrics determining the degree of similarity of models. This is pertinent for models arising in conditional form as inference rules – and hence with PSCEA applicable -- as well as the more general situation where REA is applicable. Again, use of SOP is required. More details on this will be presented below.

More generally, given (Ω,B,P) and functions of probabilities of events, such as for any a_j in B,

$$f(P(a_1),\ldots,P(a_n)), \quad g(P(a_1),\ldots,P(a_n)), \quad (2.11)$$

with the range of f and g in the unit interval [0,1], one determines an extension of (Ω,B,P) to some probability space (Ω_1,B_1,P_1), via eq.(2.5) and conditional events, say (a|b), (c|d)... in B_o satisfying appropriate analogues of eqs.(2.4), (2.5). Here, there exist *relational events*, say $f_1(a_1,\ldots,a_n)$, $g_1(a_1,\ldots,a_n)$ in B_1 connecting numerical functions f, g of probabilities with algebraic counterparts, for all well-defined P and all a_j in B, up to some restriction,

$$f(P(a_1),\ldots,P(a_n))=P_1(f_1(a_1,\ldots,a_n)),g(P(a_1),\ldots,P(a_n))=P_1(g_1(a_1,\ldots,a_n)),$$
$$\ldots . \quad (2.12)$$

Then, analogous to the role CEA plays, one may determine probabilities of prescribed logical operators acting upon such relational events. One class of example of such functions is weighted averages – as in the case of experts combining evidence via forced weights of probabilities of events which are *not* necessarily mutually disjoint, so that the total probability expansion theorem cannot be used. Another class of examples pertains to fuzzy logic, where at first the given functions are in the form of single argument increasing or decreasing functions representing truth modifiers (such as "very", "not so much", etc.). Then, utilizing the conversion of certain classes of fuzzy logic expressions to probabilities of corresponding events via use of random set representations, one obtains a situation where REA is useful. (See [2], Part III, [4].)

Before proceeding further, note that many given numerical-valued functions – as is obviously true in the weighted linear case – are in linear, or more generally, power series form, with prescribed fixed coefficients in unit interval [0,1]. In order to represent such functions via REA, one must first represent such coefficients or constants w separately. One simple approach is simply to represent any w in [0,1] as event [0,w] (interval) in $B[0,1]$, part of probability space $([0,1], B[0,1], vol_1)$, which is made independent of the "variable" events, vol_1 being lebesgue measure. (See [2], Part III] for more details.) In any case, denote $\theta(w)$ as the *constant event* in B_1, corresponding to w, for probability space (Ω_1,B_1,P_1) extending (Ω,B,P). Note, in the spirit of eq.(2.12),

$$P_1(\theta(w)) = w, \text{ all } P. \quad (2.13)$$

But, when formally $P_1 = P$ only the trivial case can hold in eq.(2.13): w = 0 or 1, corresponding to $\theta(0) = \varnothing$, $\theta(1) = \Omega$. Returning to the forced weighted average application, suppose f, g in eq.(2.11) become:

$$f(x_1,\ldots,x_n) = w_{10} + w_{11}x_1 + \ldots + w_{1n}x_n;$$
$$g(x_1,\ldots,x_n) = w_{20} + w_{21}x_1 + \ldots + w_{2n}x_n; \text{ all } x_j \text{ in } [0,1],$$
$$\text{fixed wts: } 0 \le w_{ij} \le 1, \; w_{i0}+w_{i1}+\ldots+w_{in} = 1, \; i=1,2. \quad (2.14)$$

Utilizing relative atomic forms for the x_j in terms of products of combinations of x_k and $x_l'=1-x_l$ and simple combinatorics, produces the replacement of eq.(2.14)

$$f(x_1,\ldots,x_n) = \sum_{\text{(over all } k_j \text{ in } \{0,1\},\, i=1,\ldots,n)} x_1^{(k_1)} \cdots x_n^{(k_n)} \cdot W_1(k_1,\ldots,k_n),$$

$$g(x_1,\ldots,x_n) = \sum_{\text{(over all } k_j \text{ in } \{0,1\},\, i=1,\ldots,n)} x_1^{(k_1)} \cdots x_n^{(k_n)} \cdot W_2(k_1,\ldots,k_n); \quad (2.15)$$

for all x_j in [0,1], where

$$x_j^{(1)} = x_j, \; x_j^{(0)} = x_j', \tag{2.16}$$

$$W_i(k_1,\ldots,k_n) = w_{i0} + \sum_{\text{(over all } 1\le j\le n, \text{ with } kj = 1)} w_{ij} . \tag{2.17}$$

Then, taking into account eqs.(2.13)-(2.17), the natural algebraic counterparts of eq.(2.14) is

$$f_1(a_1,\ldots,a_n) = \bigvee_{\text{(over all } k_i \text{ in } \{0,1\},\, i=1,\ldots,n)} a_1^{(k_1)}\cdots a_n^{(k_n)} \times \theta(W_1(k_1,\ldots,k_n)) \text{ in } B_1,$$

$$g_1(a_1,\ldots,a_n) = \bigvee_{\text{(over all } k_i \text{ in } \{0,1\},\, i=1,\ldots,n)} a_1^{(k_1)}\cdots a_n^{(k_n)} \times \theta(W_2(k_1,\ldots,k_n)) \text{ in } B_1, \tag{2.18}$$

where a_j in B is arbitrary, from which eq.(2.12) holds here (due to disjointness, etc.). For a concise table of relational events for a wide variety of functions of probabilities, see [7].

2.3 Uncertainty of Knowledge of Probabilities of Logical Operators

First, recall the important tightest bounds on the probability of the conjunction and disjunction of events in terms of the individual probability evaluations, for two arguments. These are denoted the (slightly extended) Fréchet-Hailperin (F-H) Bounds [6], and are used extensively in previous work [1, 2, 4]. In summary, for any probability space (Ω,B,P) any a, b in B, and real value t, $0 \le t \le 1$

$$\begin{aligned} &\max(P(a)+P(b)-1,0) \le P(a\&b) \le \min(P(a),P(b)) \\ &\le tP(a) + (1-t)P(b) \\ &\le \max(P(a),P(b)) \le P(a\vee b) \le \min(P(a)+P(b),1). \end{aligned} \tag{2.19}$$

The top lower bound is achieved iff the bottom upper bound is achieved iff $P(a\vee b) = 1$ or $P(ab) = 0$. The top upper bound is achieved iff the bottom lower bound is achieved iff $P(ab') = 1$ or $P(a'b) = 1$ iff $P(a\le b$ or $b\le a) = 1$, slightly abusing notation. The range of possible values, for example, for conjunction, say rng(a,b;P), a natural measure of the uncertainty of prior knowledge of P(a&b), is:

$$\begin{aligned} \text{rng}(a,b;P) &= \min(P(a),P(b)) - \max(P(a)+P(b)-1,0) \\ &= \min(P(a), P(a'), P(b), P(b')) \le \tfrac{1}{2}, \end{aligned} \tag{2.20}$$

which for various cases can be significantly large. Thus, if reasonable assumptions lead to the knowledge of P(a&b), a great reduction in the uncertainty of prior knowledge can be achieved. (See below.)

2.4 Probability Metrics and Model Similarity

Background on this brief review may be found in [2], Part III. Three basic metrics $d_{j,P}: B^2 \to [0,1]$ (or pseudometrics obeying at least the triangle inequality and reflexivity) – among a number of others – based upon probability for a given probability space (Ω,B,P) are defined for any a, b in B as

$$\begin{aligned} &d_{0,P}(a,b) = |P(a)-P(b)| = |P(ab')-P(a'b)|, \\ &d_{1,P}(a,b) = P(a\Delta b) = P(ab')+P(a'b) = P(a)+P(b)-2P(ab), \\ &d_{2,P}(a,b) = P(a\Delta b \mid a\vee b) \\ &\quad = [P(ab')+P(a'b)]/[P(ab')+P(a'b)+P(ab)] \\ &\quad = [P(a)+P(b)-2P(ab)] / [P(a)+P(b)-P(ab)]. \end{aligned} \tag{2.21}$$

It readily follows that

$$0 \le d_{0,P}(a,b) \le d_{1,P}(a,b) \le d_{2,P}(a,b) \le 1, \tag{2.22}$$

with strict inequality holding in general in eq.(2.22). Note that, given knowledge of P(a), P(b) separately, in order to determine $d_{j,P}(a,b)$ for $j = 1, 2$, *one must also know the conjunctive probability* P(ab). Furthermore, when P(ab) is not known, noting that such $d_{j,P}(a,b)$ are decreasing functions of P(ab), for fixed P(a), P(b), then the F-H bounds also provide the tightest bounds on the knowledge of $d_{j,P}(a,b)$ for $j = 1, 2$, as follows:

$$\begin{aligned} &d_{0,P}(a,b) \le d_{1,P}(a,b) \le \min(2-P(a)-P(b),\, P(a)+P(b)), \\ &d_{0,P}(a,b)/\max(P(a),P(b)) = 1-[\min(P(a),P(b))/\max(P(a),P(b))] \\ &\quad \le d_{2,P}(a,b) \le \min(2-P(a)-P(b),1), \end{aligned} \tag{2.23}$$

The lower bound in the top equation is achieved iff the lower bound in the bottom equation is achieved iff $P(a\le b$ or $b\le a)$; the upper bound in the top equation is achieved iff the upper bound in the bottom equation is achieved iff $P(a\vee b) = 1$ or $P(ab) = 0$. Note the limited use of $d_{0,P}$ compared with $d_{1,P}$ or $d_{2,P}$, because of the possibilities of quite distinct events a, b existing with low values of $d_{0,P}$ (as readily seen by inspection).

2.5 Use of SOP and Probability Metrics to Test Hypotheses of Model Similarity

Background on the summary of results presented in this section may be found in [2], Part III. The SOP technique -- with a basic application to deduction -- is also discussed in [5]. While the probability metrics considered in Section 2.4 are natural measures of similarity of events in that – except for $d_{0,P}$ – they are themselves either probabilities or conditional probabilities of events, one can go further and determine how significant such distances themselves are with respect to *variation of events and probabilities*. A natural way to capture this is to consider, in a minimal sense, all relevant atoms and corresponding probability evaluations of them generated by the events a, b, and to assign a prior distribution to such possible variations, summarized as the random vector $X_* = (X_1,X_2,X_3,X_4)$, where X_1 is identified with P(ab) as a random variable (rv), and similarly, slightly abusing notation,

$$(X_1,X_2,X_3,X_4) = (P(ab),P(ab'),P(a'b),P(a'b')) \text{ (as rv's).} \tag{2.24}$$

In turn, this makes the $d_{j,P}(a,b)$, each as a function of certain of the components of X_* (see eq.(2.21)), become a rv over [0,1], denoted correspondingly as D_j. For convenience, define the *null hypothesis* H_o as the situation where events a and b have no specific level of similarity due to collapsing to $\varnothing$ of any of the relative atoms, while H_1 is its logical negation. Under a *uniform joint distribution assumption* for X_* over its natural domain (taking into account the usual constraints on P), or equivalently, for $X = (X_1,X_2,X_3)$, the first three components of X_* (omitting $X_4=P(a'b')=1-X_1-X_2-X_3$, etc.), over the usual 3-simplex $S_3=\{(x_1,x_2,x_3)$: x_j in [0,1], $x_1+x_2+x_3 \leq 1\}$, one obtains the following closed-form expressions for the cumulative distribution functions F_j of D_j, under H_o:

$$F_o(t) = 1-(1-t)^3,\ F_1(t) = t^2\cdot(3-2t),\ F_2(t) = t^2;\ t \text{ in } [0,1]. \quad (2.25)$$

In addition, extensions of the results in eq.(2.25) to the case of X assumed to have a joint *dirichlet distribution* Dir($\underline{\lambda}$), where $\underline{\lambda}=(\lambda_1,\lambda_2,\lambda_3;\lambda_4)$, can be carried out in relatively simple form for D_1 and D_2. In this case,

$$E(X_j|H_0) = \lambda_j / (\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4) \quad (2.26)$$

is a useful constraint to account for possible prior bias. (See, e.g., [8] for review, motivation, and applications of such a distribution to an updating problem, noting that when $\lambda_1=\lambda_2=\lambda_3=\lambda_4=1$, it reduces to the joint uniform one. The case of D_o is also obtainable, but is rather complicated and is omitted here. Using obvious notation, where $F_{j,\underline{1}}(t) = F_j(t)$, apropos to eq.(2.25),

$$F_{1,\underline{\lambda}}(t) = B_t(\lambda_2+\lambda_3,\lambda_1+\lambda_4)/ B(\lambda_2+\lambda_3,\lambda_1+\lambda_4),$$
$$F_{2,\underline{\lambda}}(t) = B_t(\lambda_2+\lambda_3,\lambda_1) / B(\lambda_2+\lambda_3,\lambda_1);\ t \text{ in } [0,1], \quad (2.27)$$

where $B_t(\lambda,\mu)$ is the *incomplete beta function* and $B(\lambda,\mu)$ is the (complete, t=1) *beta function.* (See, e.g., [9], Sections 6.6 and 26.5.)

In any case, the test of hypotheses is carried out in the usual way for given "observed" $d_{j,P}(a,b)$, using for simplicity F_j: For significance level s, $0 < s << 1$,

Accept H_0 (and Reject H_1) iff $d_{j,P}(a,b) > C_{s,j}$;
Reject H_0 (and Accept H_1) iff $d_{j,P}(a,b) \leq C_{s,j}$; (2.28)

where $C_{s,j}$ is determined from

$s=P(\text{Reject } H_0 \mid H_0 \text{ true})= P(D_j(a,b) \leq C_{s,j}|H_0) = F_j(C_{s,j})$,

i.e.,

$$C_{s,j} = F_j^{-1}(s). \quad (2.29)$$

Alternatively, one can simply compute the "observed significance level"

$$s_j = F_j(d_{j,P}(a,b)) \quad (2.30)$$

and determine if it is too high, etc.

One can then combine REA or PSCEA together with the test of hypotheses outlined in this section above for determining similarity of models that are described as functions of probabilities as given in general in eq.(2.11): First obtain (via REA or CEA such in the case of comparing inference rules) extended probability space (Ω_1,B_1,P_1), and relational events $f_1(a_1,\ldots,a_n)$, $g_1(a_1,\ldots,a_n)$ in B_1 satisfying eq.(2.12). Then, choosing d_{j,P_1} for j = 2 or 3, replace a by $f_1(a_1,\ldots,a_n)$, b by $g_1(a_1,\ldots,a_n)$, and (Ω,B,P) by (Ω_1,B_1,P_1) relative to all results connected with eqs.(2.25)-(2.30). Note the key use here of REA again in evaluating $P_1(f_1(a_1,\ldots,a_n)$ & $g_1(a_1,\ldots,a_n))$ in order to determine $d_{j,P_1}(f_1(a_1,\ldots,a_n),g_1(a_1,\ldots,a_n))$, etc.

2.6 Algebraic Combination of Models

It is natural following testing of hypotheses of similarity for models as indicated in Section 2.5 to combine those models for which an affirmation is indicated by the test. One fundamental loss function L that has been proposed for such combining of models is a natural generalization of weighted square loss, as given, e.g., in [10], Section 3. More specifically, for any events α, β in B_1, relative to extension probability space (Ω_1,B_1,P_1) of (Ω,B,P), slightly abusing notation concerning the further extension of B_1 and P_1, for any third event γ in B_1 -- or the extended product version of B_1, P_1 -- and for any real number w in [0,1], define the loss function

$$L(\alpha,\beta;\gamma;w) = [(\alpha\Delta\gamma)\times(\alpha\Delta\gamma)\times\theta(w)]\vee[(\beta\Delta\gamma)\times(\beta\Delta\gamma)\times(\theta(w))'] \quad (2.31)$$

As before, Δ is the boolean symmetric difference operator and $\theta(w)$ is the constant event corresponding to w. Define a partial order over the L (event) values in eq.(2.31) through the relation

$$L(\alpha,\beta;\gamma_1;w) \prec L(\alpha,\beta;\gamma_2;w)$$
$$\text{iff } P_1(L(\alpha,\beta;\gamma_1;w)) \leq P_1(L(\alpha,\beta;\gamma_2;w)), \text{ all } P, \quad (2.32)$$

for any given α, β, γ_1, γ_2 in B_1. Also, define the w-*weighted event average of* α *and* β as

$$av(\alpha,\beta;w) = (\alpha\times\theta(w)) \vee (\beta\times(\theta(w))') \text{ in } B_1. \quad (2.33)$$

Again, slightly abusing notation), it is readily shown that for all w, w_i in [0,1], with inequalities holding in general in the strict sense,

$$L(\alpha,\beta;av(\alpha,\beta;w);w) = [(\alpha\&\beta')^2\vee(\alpha'\&\beta)^2]\times\theta(w)\times(\theta(w))',$$
$$L(\alpha,\beta;av(\alpha,\beta;w);w) \prec L(\alpha,\beta;\alpha\&\beta;w),\ L(\alpha,\beta;\alpha\vee\beta;w),$$
$$L(\alpha,\beta;av(\alpha,\beta;w_1);w_1) \prec L(\alpha,\beta;av(\alpha,\beta;w_1);w_2),$$
$$\alpha\&\beta \leq av(\alpha,\beta;w) \leq \alpha\vee\beta. \quad (2.34)$$

Moreover, the actual difference in the second equation of (2.34) can also be determined. Finally, one other candidate for the "average of two events", provided both α and $\beta \subseteq R^m$ (m-dimensional euclidean space) is the *Minkowski average*, given as

$$\text{mink}(\alpha,\beta;w) = w\cdot\alpha + (1-w)\cdot\beta \text{ in } R^m, \quad (2.35)$$

using pointwise scalar multiplication (·) by w and 1-w and pointwise addition (+). When α and β are convex; it follows directly, conv(.) indicating convex hull of,

$$\alpha\&\beta = w\cdot(\alpha\&\beta) + (1-w)\cdot(\alpha\&\beta) \subseteq \text{mink}(\alpha,\beta;w) \subseteq w\cdot(\alpha\cup\beta) + (1-w)\cdot(\alpha\cup\beta) \subseteq \text{conv}(\alpha\cup\beta), \quad (2.36)$$

It is thus of interest to compare $\text{av}(\alpha,\beta;w)$ with $\text{mink}(\alpha,\beta;w)$ in R^m.

3. Computational Aspects

This section, utilizes the results in [1], as well as new modifications and additions, in conjunction with the background provided in Section 2.

3.1 Calculations involving conjunction of multiple conditional event arguments for PSCEA

Consider now the number of computations required for the probability evaluation of the conjunction of conditional events – which plays a critical element in implementation. Clearly, eqs.(2.8), (2.9) show at least an exponential growth in computational complexity as the number of conditional event arguments grows. In addition, the relation of the exact conjunctive probability values to the much-easier-computed F-H bounds is also relevant. (Here, one replaces, of course, the unconditional events in eqs.(2.19) by corresponding conditional ones in boolean algebra B_o. More specifically, a somewhat-detailed numerical study has been carried out for the case of two and three arguments (see [1], Sections 2.1, 2.4 describing the procedure). Essentially, for the case of two conditional event arguments (see eq.(2.7)), one first forms the relative atoms from nontrivial conditional events (a|b), (c|d), with 8 of 16 conjunctive combinations of affirmations and negations of a, b, c, d collapsing to ∅ due to the nontriviality assumption. Then, one attempts to compute – via probability assignments determined over the atoms – a reasonable span of the possible (and typical) probability values of $P_o((a|b)\&(c|d))$, as a, b, c, d vary, relative to the F-H bounds. This means if each atomic probability is a multiple of 1/n (such as n=40), then there are $G_{n,2} = \binom{8+n}{n}$ (using standard combinatorial notation) possible probability assignments (viewed as partitions). Thus, for n=40, $G_{n,2}$ is approximately $(3.8)\cdot 10^8$. Similarly, considering the three nontrivial conditional event argument case, 37 atoms formed from, say, a, b, c, d, e, f, collapsing to ∅, the number of possible probability assignments is $G_{n,3} = \binom{26+n}{n}$. A reasonable value for n here is n=150, resulting in $G_{n,3}$ being approximately $(8.6)\cdot 10^{30}$. Therefore, the Monte Carlo method of sampling is the only feasible approach to implementing such numerical analysis. This leads to results in Section 3.2.

3.2 Calculations Involved in Hypotheses Testing via Probability Metrics

First, it is of some interest to note that the normalized incomplete beta function – the form of $F_{1,\underline{\lambda}}$ under the dirichlet assumption given in eq.(2.27) – can be identified with the tail or 1- cdf form Q of a Snedecor or *F* distribution (corresponding to the scaled ratio of two independent chi-square random variables with different degrees of freedom). More specifically, referring to [9], Section 26.6, the following holds (using Abramowitz & Stegun's notation)

$$F_{1,\underline{\lambda}}(t) = Q((\lambda_2+\lambda_3)(1-t)/[(\lambda_1+\lambda_4)t] \mid 2(\lambda_2+\lambda_3), 2((\lambda_1+\lambda_4))). \quad (3.1)$$

Eq.(3.1) is useful for hypotheses testing – as outlined in Section 2.5 – because of the well-tabulated Q (see e.g., again [9], pp. 986-989). Alternatively, the calculation of $F_{1,\underline{\lambda}}(t)$ could be carried out in the form of a computer program generating a Maclaurin series for the incomplete beta function. But this series converges very slowly. However, the continued fraction representation converges rapidly (see, e.g., [11], p. 227). For the uniform case, although $F_1(t)$ is a simple cubic, one can calculate the threshold level (which requires inversion of F_1) by Newton's method and compare the result with the tabulated value in [9], as cited above.

Values for $d_{0,P}$, $d_{1,P}$, $d_{2,P}$ have been calculated for two nontrivial conditional events (a|b), (c|d), for various values of P generated by the Monte Carlo method mentioned in Section 3.1. (For these numerical results, see [1], Appendix A, pp. 44-93.) It was found that typically such metric values often fell almost midway between the upper and lower F-H bounds.

3.3 Truncated, Empirical and Other Approaches to Conditional Events

Apropos to eq.(2.2), a natural question to pose is what will the result be to PSCEA if the full infinite disjunctive series of mutually disjoint events in B_o is truncated at various levels. It has been demonstrated [3] that closed form results always hold for all finite boolean logic operators and relations acting on the full conditional events; but these become computationally

intense for larger arguments. Thus, one can ask if the computations can be reduced in some sense via truncation techniques applied to the conditional events themselves. A related question is concerned with the construction of conditional events in finite form – or "empirical " conditional events that may well depend on the probability measure of choice – unlike the conditional events of PSCEA. In fact, the key idea here is extending the fixed point property of the recursion relation in eq.(2.3) to a more general setting, where at the beginning of the recursion, (a|b) is replaced by *any* event $(a|b)_{1,P}$ -- which may well depend upon P -- whose probability P_1 over the space in which $(a|b)_{1,P}$ exists is

$$P_1((a|b)_{1,P}) = P(a|b). \qquad (3.2)$$

Such examples of P-dependent conditional events $(a|b)_{1,P}$ can be generated geometrically (see [1], Section 3.2 for details) which also produce matching with all desired levels of PSEA operational counterparts, including the special independence conditions as in the characterization theorem for PSCEA (see [3], Theorem 16, pp. 298-299). In fact, recall that the characterization of PSCEA is only through the structure of the resulting probability evaluations of the boolean operators and relations -- *not* actually depending on the structure of the conditional events themselves. As a simple example of empirical conditional events, let $\Omega = \{\omega_1,\ldots,\omega_{16}\}$ with each atom ω_j distinct and with probability measure P over Ω being uniform, i.e., $P(\omega_j) = 1/16$. Let $a = (a|\Omega)_{1,P} = \{\omega_4, \omega_7\}$, $b = (b|\Omega)_{1,P} = \{\omega_1,\omega_2,\omega_3,\omega_4\}$, $c = (c|\Omega_{1,P}) = \{\omega_2,\omega_4, \omega_{16}\}$, $(a|b)_{1,P} = \{\omega_4,\omega_8,\omega_{12},\omega_{16}\}$. Then, all logical relations, operations, and their P-evaluations here coincides with that of PSCEA, for a,b,c, (a|b), where (a|b) is formally replaced by $(a|b)_{1,P}$.

In yet another direction concerned with extending or modifying the definition of a conditional event, it can be easily shown that the conjunctive probability evaluation in eq.(2.7) is expressed alternatively as

$$\begin{aligned} P_o(a|b)\&(c|d)) = {} & [wt_1(b,d;P)\cdot comb_1(P_{bd}(.),P_{bd}(..)) \\ & + wt_2(b,d;P)\cdot comb_2(P_{bd'}(.), P_{b'd}(..) \\ & + wt_3(b,d;P)\cdot comb_3(P_{bd}(.), P_{b'd}(..) \\ & + wt_4(b,d;P)\cdot comb_4(P_{bd'}, P_{bd}(..))](a\times c), \end{aligned} \qquad (3.3)$$

where P_{bd}, $P_{bd'}$, $P_{b'd}$ indicate the usual conditional probability measures P(.|bd), P(.|bd'), P(.|b'd), respectively, all well-defined over B, provided P(bd) > 0, P(bd') > 0, P(b'd) > 0. Also,

$$\begin{aligned} & wt_1 = P(bd \mid b\vee d), \\ & wt_2 = P(bd' \mid b\vee d)\cdot P(b'|d) + P(b'd \mid b\vee d)\cdot P(d'|b), \\ & wt_3 = P(b'd \mid b\vee d)\cdot P(d|b),\ wt_4 = P(bd' \mid b\vee d)\cdot P(b|d). \end{aligned} \qquad (3.4)$$

Furthermore, $comb_2 = comb_3 = comb_4 =$ product. Thus, $prod(P_{(1)},P_{(2)})(a\times c) = P_{(1)}(a)\cdot P_{(2)}(c)$ yields the standard *product jointing*, or probability measure $prod(P_{(1)}, P_{(2)})$ over $sigma(B\times B)$. In addition, for any two probability measures $P_{(1)}$, $P_{(2)}$ over B, $comb_1(P_{(j)},P_{(j)}):B\times B\to[0,1]$ is defined as $comb_1(P_{(j)},P_{(j)})(a\times c) = P_j(ac)$, which, when extended in the usual way, yields a legitimate probability measure over $sigma(B\times B)$. This is designated here as the *identification jointing* of P_j . More generally, the evaluation of P_o over any finite conjunction of conditional events in B_o for PSCEA is the same as a weighted mixture of probability measures over the n-fold cartesian product of B relative to the cartesian product of the consequent events. Here the weights depend only upon probabilities involving the atoms of the antecedent events and each component probability in the mixture arises as some combination of repeated identification and/or product jointing. Much more is expected in using this approach to generalize PSCEA.

3.4 Computations Involving REA Models of Linear Combinations of Probabilities

Referring to eqs. (2.12)-(2.18), for the case n = 2 and using eq.(2.21), it follows that

$$\begin{aligned} & f(P(a_1),P(a_2)) = w_{10} + w_{11}P(a_1) + w_{12}P(a_2), \\ & g(P(a_1),P(a_2)) = w_{20} + w_{21}P(a_1) + w_{22}P(a_2), \\ & f_1(a_1,a_2) = a_1a_2 \vee a_1a_2'\times\theta(w_{10}+w_{11}) \vee a_1'a_2\times\theta(w_{10}+w_{12}), \\ & g_1(a_1,a_2) = a_1a_2 \vee a_1a_2'\times\theta(w_{20}+w_{21}) \vee a_1'a_2\times\theta(w_{20}+w_{22}), \\ & P_1(f_1(a_1,a_2)) = f(P(a_1),P(a_2)),\ P_1(g_1(a_1,a_2)) = g(P(a_1),P(a_2)). \end{aligned} \qquad (3.5)$$

Assuming from now on that $w_{10} = w_{20} = 0$,

$$\begin{aligned} d_{0,P_1}(f_1(a_1,a_2), g_1(a_1,a_2)) &= |\, f(P(a_1),P(a_2)) - g(P(a_1),P(a_2))| \\ &= |P(a_1)-P(a_2)|\cdot|w_{11}-w_{21}| \\ &= |P(a_1a_2')-P(a_1'a_2)|\cdot|w_{11}-w_{21}|; \\ d_{1,P_1}(f_1(a_1,a_2), g_1(a_1,a_2)) &= P_1(f_1(a_1,a_2)\Delta g_1(a_1,a_2)) \\ &= P(a_1\Delta a_2)\cdot|w_{11}-w_{21}|; \\ d_{2,P_1}(f_1(a_1,a_2), g_1(a_1,a_2)) &= P_1(f_1(a_1,a_2)\Delta g_1(a_1,a_2)\,|\, \\ & \qquad f_1(a_1,a_2)\vee g_1(a_1,a_2)) \\ &= [P(a_1\Delta a_2)\cdot|w_{11}-w_{21}|\,] / \ [P(a_1a_2) + \\ & [P(a_1a_2')\cdot\max(w_{11},w_{21}) + P(a_1'a_2)\cdot(1-\min(w_{11},w_{21}))]. \end{aligned} \qquad (3.6)$$

It is also natural to consider as a basic measure of distance between the two models $f(P(a_1),P(a_2))$ and $g(P(a_1),P(a_2))$ the ordinary euclidean distance applied to vectors $\begin{bmatrix} w_{11}\cdot P(a_1) \\ w_{12}\cdot P(a_2) \end{bmatrix}$ and $\begin{bmatrix} w_{21}\cdot P(a_1) \\ w_{22}\cdot P(a_2) \end{bmatrix}$, i.e.,

$$\begin{aligned} & d_{w,P}(f(P(a_1),P(a_2)),\ g(P(a_1),P(a_2))) \\ & = \sqrt{(w_{11}P(a_1) - w_{21}P(a_1))^2 + (w_{12}P(a_2) - w_{22}P(a_2))^2} \end{aligned}$$

$$= \sqrt{(P(a_1))^2 + (P(a_2))^2} \cdot |w_{11}-w_{21}| . \quad (3.7)$$

Comparing eqs.(3.6) and (3.7) (noting eq.(2.22)), we then have the total ordering, with strict inequality holding in general,

$$d_{0,P_1}(f(P(a_1),P(a_2)), g(P(a_1),P(a_2))) = |P(a_1)-P(a_2)|\cdot|w_{11}-w_{21}|$$
$$\leq d_{1,P_1}(f_1(a_1,a_2), g_1(a_1,a_2)) = P(a_1\Delta a_2)\cdot|w_{11}-w_{21}|$$
$$\leq d_{2,P_1}(f_1(a_1,a_2), g_1(a_1,a_2)); \quad (3.8)$$
$$d_{0,P_1}(f(P(a_1),P(a_2)), g(P(a_1),P(a_2)))$$
$$\leq d_{w,P}(f(P(a_1),P(a_2)), g(P(a_1),P(a_2)))$$
$$= \sqrt{(P(a_1))^2 + (P(a_2))^2} \cdot |w_{11}-w_{21}|. \quad (3.9)$$

But, depending on $P(a_1)$, $P(a_2)$, which can be made precise,

$d_{w,P}(f(P(a_1),P(a_2)), g(P(a_1),P(a_2))) \leq$ or $\geq$
$d_{1,P_1}(f_1(a_1,a_2), g_1(a_1,a_2))$, $d_{2,P_1}(f_1(a_1,a_2), g_1(a_1,a_2))$. (3.10)

Reference [1], Sections 4.4 - 4.7 indicates the analogues of the above results for n=3 arguments and a method for carrying out explicit computations to obtain specific numerical comparisons.

3.5 Comparison of Minkowski and Algebraic Averaging Procedures for Combining Information

Here, for probability space (Ω,B,P) , let Ω = [-0.5, +0.5]×[0,1], P be uniform (normalized lebesgue) measure over Ω and B be the borel field over Ω. Let $\alpha, \beta \subseteq \Omega$ be two rectangles of non-equal length, in general, of the same height Δy, and with bases colinear on the x-axis where the origin of the plane is midway between them. The width of α is Δx_1,where the left edge of α is at $x_1 < 0$ and the right edge of α is at $x_1 + \Delta x_1 < 0$; the width of β is Δx_2 and the left edge of β is at $x_2 > 0$ and the right edge of β is at $x_2+\Delta x_2 > 0$, where all $\Delta x_i > 0$. This immediately implies that mink(α,β;w) is also a rectangle with base on the x-axis of height Δy and with left side being at $w\cdot x_1+(1-w)\cdot x_2$ and right side being at $w\cdot(x_1+\Delta x_1) +$ $(1-w)\cdot(x_2+\Delta x_2)$. For av($\alpha,\beta$;w), see eq.(2.33) (as a disjoint disjunction of events). In turn, this yields first the algebraic loss function evaluations, followed by the probability evaluations:

$P(L(\alpha,\beta;av(\alpha,\beta;w);w)) = ((\Delta x_1)^2+(\Delta x_2)^2)\cdot(\Delta y)^2\cdot w\cdot(1-w)$;
$P(L(\alpha,\beta;mink(\alpha,\beta;w);w))$
$= (w\cdot[\min(\Delta x_1, (1-w)\cdot(x_2-x_1)) + \Delta x_2]^2$
$+(1-w)\cdot[\min(\Delta x_2, w.(x_2+\Delta x_2 - x_1-\Delta x_1)) + \Delta x_1]^2)\cdot(\Delta y)^2$. (3.9)

Again using a Monte Carlo sampling technique, extensive numerical comparisons have been carried out in [1], Appendix B between av(α,β) and mink(α,β). The result is that the latter almost always produces a significantly larger value of P∘L compared to the former in eq.(3.9).

4. Acknowledgments

Both authors wish to express their appreciation for both direct and indirect support for this work, as part of the In-House Laboratory Independent Research Program at SSC-SD, under the aegis of the Office of Naval Research, IR Project ZU58.

5. References

[1] M.J. George & I.R. Goodman. Numerical Studies of Conditional and Relational Event Algebras. Preliminary Report (no. N66001 98-F-AC04). Houston Associates, San Diego, CA, Dec. 23, 1998.

[2] I.R. Goodman, R.P. Mahler & H.T. Nguyen. Mathematics of Data Fusion. Kluwer Academic Press, Dordrecht, Netherlands, 1997.

[3] I.R. Goodman & H.T. Nguyen. Mathematical foundations of conditionals and their probabilistic assignments. International Journal of Uncertainty, Fuzziness & Knowledge-Based Systems 3(3): 247-339, Sept., 1995.

[4] I.R. Goodman & G.F. Kramer. Extension of relational and conditional event algebra to random sets with applications to data fusion. In monograph Random Sets: Theory & Applications (J. Goutsias, R.P. Mahler & H.T. Nguyen, eds.). Springer-Verlag Publishers, New York, 1997, pp. 209-242.

[5] I.R. Goodman. A decision-aid for nodes in Command & Control Systems based on cognitive probability logic. To appear in Proceedings 1999 Command & Control Research & Technology Symposium. US Naval War College, Newport, RI, June 29-July 1, 1999.

[6] T. Hailperin. Best possible inequalities for the probability of a logical function of events. American Mathematical Monthly 72: 343-359, 1965.

[7] I.R. Goodman & H.T. Nguyen. Application of conditional and relational event algebra to the defining of fuzzy logic concepts. To appear in Proceedings SPIE 13th Annual International Symposium on Aerosense, Conference on Signal Processing Sensor Fusion and Target Recognition VIII. SPIE vol. 3720 (held at Orlando, FL, April 5-9, 1999).

[8] I.R. Goodman & H.T. Nguyen. Probability updating using second order probabilities and conditional event algebra. To appear in *Information Sciences*.

[9] M. Abramowitz & I.A. Stegun (eds.). Handbook of Mathematical Functions. National Bureau of Standards, No. 55, Wash., D.C., 1968.

[10] I.R. Goodman & H.T. Nguyen. Adams' high probability and combination of information in the context of product probability conditional event algebra. Proceedings Fusion'98 (held July 6-9, 1998, Las Vegas, NV), vol. 1, pp. 1-8.

[11] W.H. Press, S.A, Teukowsky, W.T.Vetterling & B.P. Flannery. Numerical Recipes in C: The Art of Scientific Computing, 2nd Ed.. Cambridge University Press, Cambridge, U.K., 1992.

Committees of Gaussian Kernel Based Models

T.J. Dodd and C.J. Harris
Image, Speech and Intelligent Systems Research Group
Department of Electronics and Computer Science
University of Southampton, Southampton, U.K.

Abstract *Gaussian kernel models offer a powerful nonparametric approach to regression where a priori information is available about the solution in terms of smoothness functionals. Where this a priori information is uncertain or the problem is nonstationary the choice of smoothness functional may prove unsatisfactory. One solution is to seek models which adapt to the data. Alternatively a committee of models can be used where each model represents different prior knowledge about the solution. By forming an appropriate combination of these models the output will then, on average, perform better than the average of the individual models. It will be shown that for Gaussian kernel models the optimal combination strategy is to form linear combinations of the outputs of the models. The linear weightings are found by considering the performance of the models in terms of Gaussian error bars and assuming conditional independence of the model outputs. The approach is demonstrated on an illustrative example.*

Keywords: kernel methods, combining models, Gaussian processes, regression

1 Introduction

Data fusion usually refers to a combination of information derived from multiple sources (sensors) to arrive at a consistent estimate of some desired output [1]. Alternatively we may wish to combine information from several algorithms. Often, when constructing models for a particular task, for example regression or classification, several candidate models are assessed and the model with the best performance is chosen [2]. These different models may embody different a priori information about the solution, use different features as inputs, or simply be trained from different random initialisations of the parameters.

We describe a class of parsimonious nonparametric models for data fusion which are kernel based. This class of methods includes Gaussian processes [3], support vector machines [4], regularisation networks [5] and splines [6]. These models are motivated from rigorous statistical theory including reproducing kernel Hilbert spaces, the solution of ill-posed problems and regularisation theory. Kernel methods have a natural Bayesian interpretation and provide in a straightforward manner a measure of the uncertainty on the model. These models have been applied to classification and regression problems with great success.

In the presence of nonstationarities and/or input dependent noise we must look to models which can adapt to their current operating conditions. One solution to this is to seek models which must adapt their parameters and structure recursively to sequential data. Alternatively we can train a set of models each optimised for different prior knowledge about the solution. The outputs of these models can then be combined to form an improved prediction which will be optimal for the current operating conditions.

In this paper we present an approach to the optimal linear combination of models based on estimates of the variances on the predictions of the models. Linear combinations of neural networks for regression and classification are not new [2, 7]. It is also well known that they lead to improved generalisation errors over the (weighted) average of the individual models [8, 9]. Variance based linear combinations have been investigated by Tresp and Taniguchi where they follow an intuitive approach which simply weights the models by the inverse of the variance of the prediction [10, 11]. In this paper we present a theoretical framework for variance based linear combinations based on conditional independence of the model predictions. The final algorithm is similar to the naive approach of Tresp and Taniguchi [10, 11].

The rest of the paper is organised as follows. The next section describes the Bayesian motivation for using kernel based models in regression problems. The form of the solution is then presented

as motivated by regularisation theory and a simple explanation of the idea of smoothness functionals given in terms of Fourier transforms. Bayesian techniques are then applied to the problem of obtaining confidence intervals for the predictions from the models. The problem of combining kernel models is then described and a simple linear combination rule presented. The combination approach is motivated from the Gaussian posteriors over the outputs of the models and the conditional independence of these outputs. Finally, the approach is demonstrated on a simple illustrative example.

2 Bayes in Function Spaces and Regularisation

Given a data set $\mathcal{D} = \{\mathbf{x}_i, z_i\}, i = 1, \ldots, N$ consisting of d dimensional inputs $\mathbf{x}_i$ and univariate observations z_i the problem of regression is to infer the relationship between the $\mathbf{x}_i$'s and z_i's for the set $\mathcal{D}$. We denote our estimate of this relationship by $y(\mathbf{x})$ which will be found by minimising some risk function. We can write Bayes' rule for the regression problem as

$$p(y|\mathcal{D}) = \frac{p(\mathcal{D}|y)p(y)}{p(\mathcal{D})}$$

which, ignoring the normalising constant $p(\mathcal{D})$, is equivalent to

$$p(y|\mathcal{D}) \propto p(\mathcal{D}|y)p(y). \tag{1}$$

The likelihood $p(\mathcal{D}|y)$ is the probability that the observed data were generated by a particular function and $p(y)$ is a prior over functions. This prior reflects our beliefs as to what class of functions the final model should belong. The posterior $p(y|\mathcal{D})$ is the probability density over the possible functions given the observed data. We now have a single level of Bayesian inference which consists of finding the most probable functional approximator amongst a class of approximators. This class is specified via the prior and is usually chosen to incorporate a specific degree of assumed smoothness.

We choose, for reasons which will become apparent later, the prior to have a form

$$p(y) \propto \exp\{-\lambda\psi[y]\} \tag{2}$$

where λ is a positive constant and $\psi[y]$ is a smoothness functional. The exact nature of this smoothness functional will be discussed later but, for now, it will be assumed to embody some a priori preference for smooth functions. The form of the prior $p(y)$ then gives high probability to those functions for which the smoothness functional $\psi[y]$ is small.

All that remains is to find the likelihood $p(\mathcal{D}|y)$. If we make the assumption that the observations are corrupted by additive Gaussian noise, i.e.

$$z_i = f(\mathbf{x}_i) + \varepsilon_i$$

where $\varepsilon_i \sim N(0, \sigma_\varepsilon^2)\ \forall\ i$ then it is trivial to show that

$$p(\mathcal{D}|y) \propto \exp\left\{-\frac{1}{2\sigma_\varepsilon^2}\sum_{i=1}^{N}(z_i - y(\mathbf{x}_i))^2\right\}. \tag{3}$$

Substituting for the prior and likelihood into Bayes' rule, Eq. 1, the posterior can be written as

$$p(y|\mathcal{D}) \propto \exp\{-H[y]\}$$

where

$$H[y] = \frac{1}{2\sigma_\varepsilon^2}\sum_{i=1}^{N}(z_i - y(\mathbf{x}_i))^2 + \lambda\psi[y]. \tag{4}$$

The most probable function, in a maximum a posteriori sense, is then the one that minimises the functional $H[y]$.

The form, Eq. 4, is a penalised least squares solution for the desired function where the penalty term $\lambda\psi[y]$ forces the solution to have certain desired characteristics. We will now see how we can solve for y.

3 Kernel Based Models for Regression

In Eq. 4 are present two terms, the second is a penalty term which will be discussed later. The first term measures the closeness of the estimate to the data, in this case a simple least squares error term. More generally this term is called an empirical risk function which we will denote by $Q_{emp}[y]$. The maximum likelihood approach to regression is to find the function $y(\mathbf{x})$ which minimises the empirical risk. However, such a problem is ill-posed.

A problem is defined to be well-posed in the sense of Hadamard if it has a solution which satisfies the conditions that it [4]

- exists;
- is unique; and
- is stable.

An ill-posed problem is then simply one which is not well-posed. The problem of regression from finite noisy data is an example of an ill-posed problem in the sense that the solution may not be unique and will often be unstable. However we will now see that the Bayesian formalism above actually leads to a well-posed problem.

In order to make the approximation problem well posed we must impose some form of restriction on the possible solutions. If we are to make realistic restrictions we must rely on a priori knowledge about the problem. The simplest form of a priori knowledge is to assume that the underlying function is "smooth" in the sense that if two inputs are close then the corresponding outputs should be close. We define a smoothness functional $\psi[y]$ on the outputs of our learning machine which takes large values for non-smooth functions and small values for smooth functions. We will discuss the exact nature of this smoothness functional shortly, but for now we assume it to be convex and continuous.

The idea of regularisation theory is then to solve an ill-posed problem from a variational principle, which contains both the data and prior smoothness information [12]. There are then three basic possible settings for the optimisation problem [5]. We can minimise $Q_{emp}[y]$ subject to the constraint that $\psi[y] \leq \Lambda$. This is what should be done when following the principle of empirical risk minimisation and can be interpreted as incorporating a form of capacity (or complexity) control. We are minimising the empirical risk while keeping the model complexity fixed by enforcing an upper bound on the measure of complexity.

The second possible setting is to minimise the regularisation term $\psi[y]$ with an upper bound on the empirical risk, i.e. $Q_{emp}[y] \leq \Lambda'$. However, the situation we consider here, and which results in a simple solution is to minimise the regularised risk functional

$$Q_{reg}[y] = Q_{emp}[y] + \lambda\psi[y]$$

where λ is a positive number called the regularisation (or smoothing) parameter. The regularisation parameter controls the trade-off between the closeness to the data and the smoothness of the solution. For the remainder we will assume that the empirical risk is given by the simple sum of squared errors such that the regularised risk functional is now

$$Q_{reg}[y] = \sum_{i=1}^{N}(z_i - y(\mathbf{x}_i))^2 + \lambda\psi[y] \qquad (5)$$

which now bears a close resemblance to our Bayesian solution in function space, Eq. 4. In fact, the equations are equivalent and can be made equal by including the term $1/2\sigma_\varepsilon^2$ in Eq. 4 within the regularisation parameter. We therefore see how the Bayesian approach in function space leads to the same result as that motivated by statistical learning and approximation theory via regularisation theory. In particular the choice of a Gaussian prior of the form given by Eq. 2 can now be justified and actually corresponds to a smoothness functional.

All that remains is to choose the smoothness functional and solve for the learning machine. Natural measures of the smoothness of a function are provided by differential operators which in the case of univariate inputs lead to the class of Tikhonov regularisers of the form

$$\psi[y] = \frac{1}{2}\sum_{r=0}^{R}\int_a^b h_r(x)\left(\frac{d^r y}{dx^r}\right)^2 dx$$

where $h_r \geq 0$ for $r = 0, 1, \ldots, R-1$ and $h_R > 0$. In higher dimensions this class of regularisers can be generalised through Laplacian type differential operators.

More generally, these differential operators form part of a class of smoothness functionals for which the solutions of the minimisation of the regularised risk functional, Eq. 5, have the same form. This class of smoothness functionals have the general form

$$\psi[y] = \int_{\mathbb{R}^d} \frac{|\tilde{y}(\mathbf{s})|^2}{\tilde{G}(\mathbf{s})} d\mathbf{s} \qquad (6)$$

where $\tilde{}$ denotes the Fourier transform and $\tilde{G}$ is a positive function that falls off to zero as $\|\mathbf{x}\| \to \infty$. We can see immediately why this class of functionals measures the smoothness of the function $y(\mathbf{x})$. The effect of the function $1/\tilde{G}(\mathbf{s})$, which corresponds to a high-pass filter is to extract the high frequency content of $y(\mathbf{x})$. The power of this high frequency content is then measured via the L_2 norm of the result. The smoothness functional then has the desired characteristic of taking high values for non-smooth functions (which have a large high frequency content) and low values for smooth functions (which have a small high frequency content).

There are in fact two (related) classes of functions $\tilde{G}$ which we are interested in. Either $\tilde{G}$ is the Fourier transform of a positive definite (p.d.) function or of a conditionally positive definite (c.p.d.) function where we denote this function by G. In the case where G is positive definite then the associated smoothness functional $\psi[y]$ is a norm. Similarly, for G conditionally positive definite then $\psi[y]$

is a semi-norm with a finite dimensional null space $\mathcal{N}$.

The solution of the regularised risk functional then has the form [12]

$$y(\mathbf{x}) = \sum_{i=1}^{N} c_i G(\mathbf{x} - \mathbf{x}_i) + \sum_{\alpha=1}^{k} d_\alpha \gamma_\alpha(\mathbf{x}) \quad (7)$$

where $\{\gamma_\alpha\}_{\alpha=1}^{k}$ is a basis in the k dimensional null space $\mathcal{N}$ and the coefficients d_α and c_i satisfy the following linear system:

$$\begin{aligned} (\mathbf{G} + \lambda \mathbf{I})\mathbf{c} + \mathbf{\Gamma}^T \mathbf{d} &= \mathbf{z} \\ \mathbf{\Gamma c} &= \mathbf{0} \end{aligned}$$

where $\mathbf{z}$ is the vector of observations, $\mathbf{c}$ and $\mathbf{d}$ are the vectors of the coefficients d_α and c_i, $\mathbf{G}$ is a matrix of kernel activations, i.e.

$$\mathbf{G}_{ij} = G(\mathbf{x}_i - \mathbf{x}_j) \quad (8)$$

and $\mathbf{\Gamma}$ is a matrix of the values of the null space basis

$$\mathbf{\Gamma}_{\alpha i} = \gamma_\alpha(\mathbf{x}_i).$$

This learning machine is referred to as a regularisation network. The general solution includes various other models including radial basis function networks [8], splines [6], Gaussian processes [3], and certain classes of support vector machines [4]. The nature of these models is dictated by the choice of prior (smoothness functional) and whether this is interpreted in a Bayesian framework. Gaussian processes have been developed from a strong Bayesian motivation, splines as the solution of approximation in certain reproducing kernel Hilbert spaces, and support vector machines from the idea of structural risk minimisation. We now see how, for the class of regularisation networks with positive definite priors, we can assign confidence intervals on the predictions.

4 Confidence Intervals

We now return to the probabilistic (Bayesian) interpretation of the kernel models. We previously saw how positive definite smoothness functionals are equivalent to Gaussian priors over the functions. Similarly the output of a kernel model, Eq. 7, can be interpreted as a Gaussian posterior probability density. From this we are able to define Gaussian confidence intervals on our estimates.

Based on the training data $\{\mathbf{x}_i, z_i\}, i = 1, \ldots, N$ we can define a covariance matrix $\mathbf{G}_N$ for this data based on the inputs with ijth element given by

$$\mathbf{G}_N^{ij} = G(\mathbf{x}_i, \mathbf{x}_j)$$

where $G(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel function and $\mathbf{G}_N$ is simply the matrix of kernel activations, Eq. 8. Similarly we can define a vector of covariances, $\mathbf{g}_{N+1}$ between the training data points and the point at which we wish to make a prediction, i.e. $y(\mathbf{x}_{N+1})$. This vector then has ith entry

$$\mathbf{g}_{N+1}^i = G(\mathbf{x}_i, \mathbf{x}_{N+1}).$$

Finally the scalar variance of the new data point is defined as $g = G(\mathbf{x}_{N+1}, \mathbf{x}_{N+1}) + \lambda$. Using standard results from multivariate Gaussian conditional distributions the mean and variance of the prediction for the new data point are then given by [13]

$$\begin{aligned} y(\mathbf{x}_{N+1}) &= \mathbf{g}_{N+1}^T \mathbf{G}_N^{-1} \mathbf{z} \quad (9) \\ \sigma_y^2 &= g - \mathbf{g}_{N+1}^T \mathbf{G}_N^{-1} \mathbf{g}_{N+1} \quad (10) \end{aligned}$$

where $\mathbf{z}$ is the vector of observations. The variance term can now be interpreted as error bars on the prediction which are usually shown on plots as one or two standard deviations from the mean prediction (we shall always show them as one standard deviation).

However, we must be careful in our interpretation of the error bars and subsequently their role in combining kernel models. Consider Figure 1 where a function is estimated using an optimal and suboptimal kernel model. A suboptimal model is defined as one for which one or more of the hyperparameters converges to a value such that the estimated error bars are inconsistent with the actual performance of the model This is different from a poor model which does not predict the true function very well but for which the error bars reflect this by being correspondingly large.

According to MacKay [14] the error bars, as described above, are found assuming the model is correct. For a suboptimal model the true interpolant can lie significantly outside the error models. This is demonstrated in Figure 1 where the error bars on the suboptimal model bear little relation to the actual poor performance of this model. Therefore, in applying any technique which uses the error bars as estimates of model performance we must take care to ensure that the models are optimal in the sense that the error bars are consistent with the actual model performance.

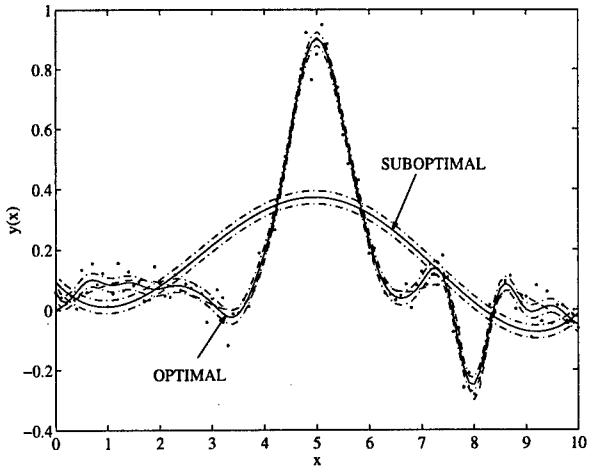


Figure 1: Optimal and suboptimal predictions using a Gaussian process model showing the mean predictions [–], 1σ error bars [- - -] and observations [$\cdots$]. Whilst the suboptimal model cannot adequately predict the function the error bars indicate that this prediction is as good as the optimal prediction.

5 Combining Kernel Models

We propose, in this section, an algorithm for combining kernel models which forms linear combinations of the corresponding outputs of the models. However, the approach is also directly applicable to other models for which the posterior probability densities are Gaussian, for example generalised linear regressors and certain classes of neural networks [15].

5.1 Theoretical Framework

We first look more generally at combining different models from a statistical perspective. We assume that we have M models and that the posterior output probability from model i is given by $p_i(y|\mathbf{x}_i)$. There are principally two scenarios for combination of the models [16]. In the first the input to each model is unique as might be the case in combining classifiers where the distinct input patterns are of different measurements/features. The combination strategy must then take account of the fact that the posterior probabilities are no longer estimates of the same functional value. In the second approach each model takes the same inputs but the models themselves are distinct. This may arise due to differing model structures, a priori assumptions on the data, or initialisations of the parameters.

We are interested, in this paper, in the second scenario and in particular the case where different a priori assumptions are made about the data. However, the basic theoretical framework and algorithm are more widely applicable to the other situations. The posterior output probabilities can now be denoted by $p_i(y|\mathbf{x}, \mathcal{H}_i)$ where we have assumed a common input and the output is now conditioned on the model, or hypothesis, $\mathcal{H}_i$. The combined output posterior can be expressed using Bayes' theorem as

$$p(y|\mathbf{x}, \mathcal{H}_1, \ldots, \mathcal{H}_M) = \frac{p(\mathbf{x}|y, \mathcal{H}_1, \ldots, \mathcal{H}_M)p(y|, \mathcal{H}_1, \ldots, \mathcal{H}_M)}{p(\mathbf{x}|, \mathcal{H}_1, \ldots, \mathcal{H}_M)}.$$

In most cases it will not be practical to compute the full joint densities and we must therefore make an assumption that the posteriors of the models are conditionally independent, i.e.

$$p(y|\mathbf{x}, \mathcal{H}_1, \ldots, \mathcal{H}_M) = \prod_{i=1}^{M} p_i(y|\mathbf{x}, \mathcal{H}_i). \qquad (11)$$

Is this reasonable? The denominator in Bayes' theorem is simply a normalising constant which is independent of y and can therefore be ignored. The conditional independence of the prior, $p(y|, \mathcal{H}_1, \ldots, \mathcal{H}_M)$, must be reasonable as we are deliberately assigning different (independent) priors to each model. These different priors are in terms of the expected smoothness of the solution and/or the initialisations of the parameters. The conditional independence of the likelihood, $p(\mathbf{x}|y, \mathcal{H}_1, \ldots, \mathcal{H}_M)$ will be realistic in most situations as we would expect the independence of the models, even for a common output, to infer that this output is the result of different inputs to the models. Based on the assumption of conditional independence then the output of the committee of models must be calculated using Eq. 11 which for certain classes of models leads to a simple solution.

5.2 The Algorithm

A key feature of the kernel models described above is that, taking a Bayesian perspective, it is possible to assign confidence intervals to the predictions. These confidence intervals are of a known form and are in fact Gaussian. We assume then that, for M models, the outputs of the models are Gaussian with mean μ_i and variance σ_i^2, $i = 1, \ldots, M$. Under the assumption of conditional independence the

output of the committee will be given by

$$p(y) = \prod_{i=1}^{M} p_i(y)$$

where $p_i(y)$ are the probability densities of the outputs of the individual models. As the $p_i(y)$ are Gaussian then the output of the committee, $p(y)$, will also be Gaussian with mean

$$\mu_y = \frac{\sum_{i=1}^{M} \frac{\mu_i}{\sigma_i^2}}{\sum_{i=1}^{M} \frac{1}{\sigma_i^2}} \tag{12}$$

and variance given by the equation

$$\frac{1}{\sigma_y^2} = \sum_{i=1}^{M} \frac{1}{\sigma_i^2}. \tag{13}$$

The combination rule is a simple weighted linear summation of the outputs of the individual models. The mean of the committee can be written as

$$\mu_y = \sum_{i=1}^{M} \alpha_i \mu_i \tag{14}$$

where

$$\alpha_i = \frac{1/\sigma_i^2}{\sum_{i=1}^{M} \frac{1}{\sigma_i^2}}. \tag{15}$$

We are now in a position to analyse the effect of the committee in reducing the error on the approximation. The following analysis follows that due to Bishop [8]. Denoting the true regression function by $f(\mathbf{x})$ then the mean output of model i can be written as equal to the value of the regression function plus some error term, i.e.

$$\mu_i(\mathbf{x}) = f(\mathbf{x}) + \eta_i(\mathbf{x})$$

where the error term, $\eta_i(\mathbf{x})$ should not be confused with the observation error ε_i.

The average sum of squares error for model i is then given by

$$e_i = E[(\mu_i(\mathbf{x}) - f(\mathbf{x}))^2] = E[\eta_i^2]$$

where the expectation $E[\cdot]$ is taken over the input space $\mathcal{X}$ weighted by the unconditional density of $\mathbf{x}$, i.e.

$$E[\eta_i^2] = \int \eta_i^2(\mathbf{x}) p(\mathbf{x}) d(\mathbf{x}).$$

The average error for the M individual models is then

$$e_{av} = \frac{1}{M} \sum_{i=1}^{M} e_i = \frac{1}{M} \sum_{i=1}^{M} E[\eta_i^2]. \tag{16}$$

Using the combined output from the committee, Eq. 14, then we can define the error on this prediction as

$$\begin{aligned} e_{com} &= E\left[\left(\sum_{i=1}^{M} \alpha_i \mu_i(\mathbf{x}) - f(\mathbf{x})\right)^2\right] \\ &= E\left[\left(\sum_{i=1}^{M} \{\alpha_i \mu_i(\mathbf{x}) - \alpha_i f(\mathbf{x})\}\right)^2\right] \end{aligned}$$

where we have assumed $\sum_{i=1}^{M} \alpha_i = 1$ (which can always be ensured if necessary by normalising the weighting coefficients). The committee error is then given by

$$e_{com} = E\left[\left(\sum_{i=1}^{M} \alpha_i \eta_i\right)^2\right] = \sum_{i=1}^{M} \alpha_i^2 E[\eta_i^2]$$

where it has been assumed that the η_i have zero-mean and are uncorrelated. In order to analyse this further we make the additional assumption that the weighting coefficients are all equal to $1/M$ such that we arrive at the important result that

$$e_{com} = \frac{1}{M^2} \sum_{i=1}^{M} E[\eta_i^2] = \frac{1}{M} e_{av}. \tag{17}$$

In other words the sum-of-squares error of the committee is a factor of M lower than the average of the sum-of-squares errors of the individual models. In practise the reduction in error will not be a factor of M as the errors η_i will be uncorrelated. However, this will be offset to an extent as the α_i weight better models greater than for the case where $\alpha_i = 1/M \ \forall \ i$.

6 Example

We now present the application of the above ideas to an illustrative example. The data were generated from the function:

$$\begin{aligned} z(t) &= 0.1\sin(t) + \exp\left\{-\frac{1}{0.7}(t-5)^2\right\} + \\ &\quad 0.4\exp\left\{-\frac{1}{0.1}(t-8)^2\right\} + \varepsilon(t) \end{aligned}$$

where the noise process $\varepsilon(t)$ has zero mean and variance 0.0025. The observations were generated over the interval $[0, 10]$ but with missing data in the regions $(1, 2)$ and $(7, 8)$. A committee of six Gaussian process models was trained using the noisy observations. Each model was initialised with a different

value for the regularisation parameter reflecting a priori different beliefs in the smoothness of the final solution.

The results for the six models are shown in Figure 2 and reflect this difference in expected smoothness most particularly in the region of missing data $(1, 2)$. The final model, whilst poor, cannot be considered suboptimal in the sense described earlier as the error bars correctly reflect the lack of confidence in the predictions.

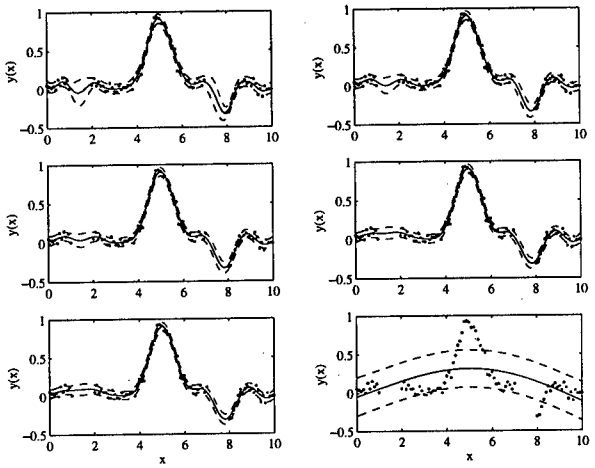


Figure 2: Predictions for the individual models in the committee. Note the increase in error bars for regions of missing data and the different smoothness characteristics of the solutions. The prediction [–], data [· · ·] and error bars [- - -] are shown.

The final output of the committee is shown in Figure 3. The performance of this committee is also reflected by the mean-squared errors (MSE) over the data set. The MSE of the committee prediction (0.0034) was better than all the individual models except for one which had an MSE of 0.0027 and was over a factor of 3 times better than the mean MSE over the model of 0.0110.

7 Conclusions

We have described a class of models for regression problems based on a Bayesian formalism in function spaces and a variational principle for solving ill-posed problems using prior information. The resulting predictor encompasses various classes of models including splines, Gaussian processes, support vector machines and regularisation networks. The solution is based on a kernel function which can

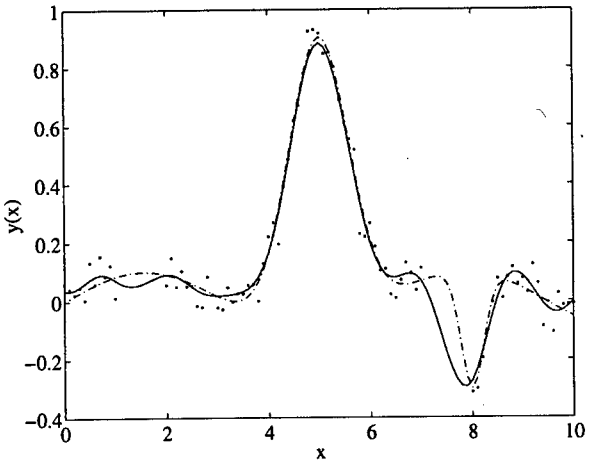


Figure 3: The predicted output [–] versus true output [- - -] for the illustrative example. The prediction generally shows close correspondence with the true output even in the region $(1, 2)$ of missing data. However, in the second region of missing data, $(7, 8)$, the prediction diverges from the true signal. The noisy observations [· · ·] are also shown.

be interpreted as a Gaussian prior from a Bayesian perspective. For positive definite kernels (for which the null space is empty) we can define Gaussian confidence intervals on our predictions. These form the basis of a simple linear weighted combination rule for a committee of kernel models. This rule is motivated by a probabilistic framework and shown to perform better than the average of the individual models. The approach was demonstrated using a committee of Gaussian processes applied to a simple regression problem.

8 Acknowledgements

The authors would like to thank Matra British Aerospace Dynamics and EPSRC for their partial support of this work.

References

[1] C.J. Harris, A. Bailey, and T.J. Dodd. Multisensor data fusion in defence and aerospace. *The Aeronautical Journal*, 102(1015):229–244, May 1998.

[2] Sherif Hashem and Bruce Schmeiser. Improving model accuracy using optimal linear com-

binations of trained neural networks. *IEEE Transactions on Neural Networks*, 6(3):792–794, May 1995.

[3] David J.C. MacKay. Gaussian processes: A replacement for supervised neural networks? Lecture notes for a tutorial at NIPS 1997. Available at http://wol.ra.phy.cam.ac.uk/mackay/, 1997.

[4] Vladimir N. Vapnik. *Statistical Learning Theory.* Adaptive and Learning Systems for Signal Processing, Communications and Control. John Wiley & Sons, 1998.

[5] Alexander Johannes Smola. *Learning with Kernels.* PhD thesis, Informatik der Technischen Universitat Berlin, 1998.

[6] Grace Wahba. *Spline Models for Observational Data*, volume 50 of *Series in Applied Mathematics.* SIAM, Philadelphia, 1990.

[7] Michael LeBlanc and Robert Tibshirani. Combining estimates in regression and classification. *Journal of the American Statistical Association*, 91(436):1641–1650, December 1996.

[8] Christoper M. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford, 1995.

[9] Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. In G. Tesauro, D.S. Touretzky, and T.K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 231–238, Cambrigde MA, 1995. MIT Press.

[10] Michiaki Taniguchi and Volker Tresp. Averaging regularized estimators. *Neural Computation*, 9:1163–1178, 1997.

[11] Volker Tresp and Michiaki Taniguchi. Combining estimators using non-constant weighting functions. In G. Tesauro, D.S. Touretzky, and T.K. Leen, editors, *Advances in Neural Information Processing Systems 7*, Cambridge MA, 1995. MIT Press.

[12] Frederico Girosi, Michael Jones, and Tomaso Poggio. Priors, stabilizers and basis functions: from regularization to radial, tensor and additive splines. Technical Report C.B.C.L. Paper No. 75, Artificial Intelligence Laboratory, MIT, 1993.

[13] T.W. Anderson. *An Introduction to Multivariate Statistical Analysis.* John Wiley & Sons, second edition, 1984.

[14] David J.C. MacKay. *Bayesian Methods for Adaptive Models.* PhD thesis, California Institute of Technology, Pasadena, California, 1992.

[15] T.J. Dodd and C.J. Harris. A new multiple model framework for recursive Bayesian modelling of time series by neural networks. 1999. Also to be presented at Fusion '99.

[16] Josef Kittler, Mohamad Hatef, Robert P.W. Duin, and Jiri Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, March 1998.

Combining Models to Improve Classifier Accuracy and Robustness

Dean W. Abbott
Abbott Consulting
P.O. Box 22536
San Diego, CA 92192-2536 USA
Email: dean@abbott-consulting.com

Abstract

Recent years have shown an explosion in research related to the combination of predictions from individual classification or estimation models, and results have been very promising. By combining predictions, more robust and accurate models are almost guaranteed to be generated without the need for the high-degree of fine tuning required for single-model solutions. Typically, however, the models for the combination process are drawn from the same model family, though this need not be the case.

This paper summarizes the current direction of research in combining models, and then demonstrates a process for combining models from diverse algorithm families. Results for two datasets are shown and compared with the most popular methods for combining models within algorithm families.

Key Words: Data mining, model combining, classification, boosting

1. Introduction

Many terms have been used to describe the concept of model combining in recent years. Elder and Pregibon [1] used the term *Blending* to describe "the ancient statistical adage that 'in many counselors there is safety'". Elder later called this technique, particularly applied to combining models from different classifier algorithm families, *Bundling* [2]. The same concept has been described as *Ensemble of Classifiers* by Dietterich [3], *Committee of Experts* by Steinberg [4], and *Perturb and Combine* (P&C) by Breiman [5]. The concept is actually quite simple: train several models from the same dataset, or from samples of the same dataset, and combine the output predictions, typically by voting for classification problems and averaging output values for estimation problems. The improvements in model accuracy have been so significant, Friedman *el al* [6] stated about one form of model combining (boosting) "is one of the most important recent developments in classification methodology."

There is a growing base of support in the literature for model combining providing improved model performance. Wolpert [7] used regression to combine neural network models (*Stacking*). Breiman [8] introduced *Bagging* which combines outputs from decision tree models generated from bootstrap samples (with replacement) of a training data set. Models are combined by simple voting. Fruend and Shapire [9] introduced *Boosting*, an iterative process of weighting more heavily cases classified incorrectly by decision tree models, and then combining *all* the models generated during the process. *ARCing* by Breiman [5] is a form of boosting that, like boosting weighs incorrectly classified cases more heavily, but instead of the Fruend and Shapire formula for weighting, weighted random samples are drawn from the training data. These are just a few of the most

popular algorithms currently described in the literature, and many more methods have been developed by researchers as well.

While most of the combining algorithms described above were used to improve decision tree models, combining can be used more broadly. Trees often show benefits from combining because the performance of individual trees are typically worse than other data mining methods such as neural networks and polynomial networks, and because they tend to be structurally unstable. In other words, small perturbations in training data set for decision trees can result in very different model structures and splits. Nevertheless, results for any data mining algorithm that can produce significant model variations can be improved through model combining, including neural networks and polynomial networks. Regression, on the other hand, is not easily improved through combining models because it produces very stable and robust models. It is difficult through sampling of training data or model input selection to change the behavior of regression models significantly enough to provide the diversity needed for combining to improve single models.

While the reasons combining models works so well are not rigorously understood, there is ample evidence that improvements over single models are typical. Breiman [5] demonstrates bagging and arcing improving single CART models on 11 machine learning datasets in every case. Additionally, he documents that arcing, using no special data preprocessing or classifier manipulation (just read the data and create the model), often achieves the performance of hand-crafted classifiers that were tailored specifically for the data.

However, it seems that producing relatively uncorrelated output predictions in the models to be combined is necessary to reduce error rates. If output predictions are highly correlated, little reduction in error is possible as the "committee of experts" have no diversity to draw from, and therefore no means to overcome erroneous predictions. Decision trees are very unstable in this regard as small perturbations in the training data set can produce large differences in the structure (and predictions) of a model. Neural networks are sensitive to data used to train the models and to the many training parameters and random number seeds that need to be specified by the analyst. Indeed, many researchers merely train neural network models changing nothing but the random seed for weight initialization to find models that have not converged prematurely in local minima. Polynomial networks have considerable structural instability, as different datasets can produce significantly different models, though many of the differences in models produce correlated results; there are many ways to achieve nearly the same solution.

A strong case can be made for combining models across algorithm families as a means of providing *uncorrelated* output estimates because the difference in basis functions used to build the model. For example, decision trees produce staircase decision boundaries via rules effecting one variable at a time. Neural networks produce smooth decision boundaries from linear basis functions and a squashing function, and polynomial networks use cubic polynomials to produce an even smoother decision boundary. Abbott [10] showed considerable differences in classifier performance class by class—information that is clear to once classifier is obscure to another. Since it is difficult to gauge

a priori which algorithm(s) will produce the lowest error for each domain (on unseen data), combining models across algorithm families mitigates that risk by including contributions from all the families.

2. Method for Combining Models

Model combining done here expands on the bundling research done by Elder [2]. Models from six algorithm families were trained for each dataset. To determine which model to use for each algorithm family, dozens to hundreds of models were trained and only the single best was retained; only the best model from each algorithm family was represented.

2.1. Algorithms and Combining Method

Once the six models were obtained, they were combined in every unique combination possible, including all two, three-, four-, five-, and six-way combinations. Each of the combinations was achieved by a simple voting mechanism, with each algorithm model having one vote. To break ties, however, a slight weighting factor was used, with the models having the best performance during training given slightly larger weight (Table 2.1). For example, if an example in the evaluation dataset had one vote from a first-ranked model, and anther from a second-ranked model, the first-ranked model would win the vote 1.28 to 1.22. The numbers themselves are arbitrary, and only need to provide a means to break ties.

Table 2.1: Model Combination Voting Weights to Break Ties

Model Rank on Training Data	Weight
First	1.28
Second	1.22
Third	1.16
Fourth	1.10
Fifth	1.05
Sixth	1.00

The six algorithms used were neural networks, decision trees, k-nearest neighbor, Gaussian mixture models, radial basis functions, and nearest cluster models. Five of the six models for each dataset were created using the PRW by Unica Technologies [11], and the sixth model (C5 decision trees) was created using Clementine by SPSS [12] . Full descriptions of the algorithms can be found in Kennedy, Lee, et al [13].

2.2. Datasets

The two datasets used are the glass data from the UCI machine learning data repository [14] and the satellite data used in the Statlog project [15]. Characteristics of the datasets are shown in Table 2.2:

Table 2.2: Dataset Characteristics

	Number Examples			Number Inputs/Outputs	
Dataset	Train	Test	Eval	Vars	Classes
Glass	150	0	64	9	6
Satellite	3105	1330	2000	36	6

Training data refers to the cases that were used to find model weights and parameters. Testing data was used to check the training results on independent data, and was used ultimately to select which model would be selected from those trained. Training and testing data split randomly, with 70% of the data used for training, 30% for testing. No testing data was used for the glass dataset because so few examples were available; models were trained and pruned to reduce the risk of overfitting the data.

A third, separate dataset, the evaluation dataset, was used to report all results shown in this paper. The evaluation data was not used during

the model selection process, only to score the individual and combined models, so that bias would not be introduced. The glass data was split in such as way as to retain the relative class representation in both the training and evaluation datasets.

A breakdown of the number of cases per class is shown in following two tables, 2.3 and 2.4:

Table 2.3: Glass Data Class Breakdown

	Number Examples	
Class	Train	Eval
1	49	21
2	54	22
3	12	5
5	9	4
6	6	3
7	20	9

Note that there are no examples for class 4.

Table 2.4: Satellite Data Class Breakdown

	Number Examples		
Class	Train	Test	Eval
1	752	320	461
2	323	156	224
3	649	312	397
4	283	132	212
5	343	127	237
7	755	283	470

Note that there are no examples for class 6.

3. Results

Results are compiled single models for each of the six algorithms, and all possible model combinations.

Table 3.1: Number of Model Combinations

Number Models	Number Combos
1	6
2	15
3	20
4	15
5	6
6	1

The emphasis here is on minimizing classifier error without going through the process of fine-tuning the classifiers with domain knowledge to improve performance—a necessary step for real-world applications.

3.1. Glass Dataset Results

The single best models for each algorithm family is shown in Figure 3.1 below. Results are presented in terms of classification errors, so smaller numbers (shorter bars) are better. For each model, a search for the best model parameters was performed first, increasing the likelihood that the best model for each algorithm was found.

Nearest neighbor had perfect training results (by definition), and the best remaining algorithms were, in order, neural networks, decision trees, Gaussian mixture, nearest cluster, and radial basis functions, and ranged from 28.1% error to 37.5% error. Interestingly, nearest cluster and Gaussian mixture models, both using PDF measures, had the best on evaluation data.

Model combinations produced the following results shown in Figure 3.2. Not all datapoints can be seen as model combinations sometimes produce identical error scores. Two interesting trends can be seen in the figure. First, the trend is for the percent classification error to decrease as the number of models combined increases, though the very best (lowest classification error) case occurs with 3 or 4 models. The lower error rate (23.4%) occurs for the combinations in Table 3.1 below.

Amazingly, radial basis functions occur in all four of the best combination, even though it was clearly the single worst classifier. Each of the other classifiers was represented exactly twice except the Gaussian mixture which occurred once. Radial basis functions also appeared in two of the four *worst* combinations of more than 3 classifiers as well (Table 3.2), so it appears

this algorithm is a wild card, and one cannot tell from the training result alone whether or not it will combine well. The worst 2-way models always include neural networks or k-nearest neighbor, and in these case, the models were not improved compared to the single model results (34.4% for k-nearest neighbor, 31.3 for neural networks).

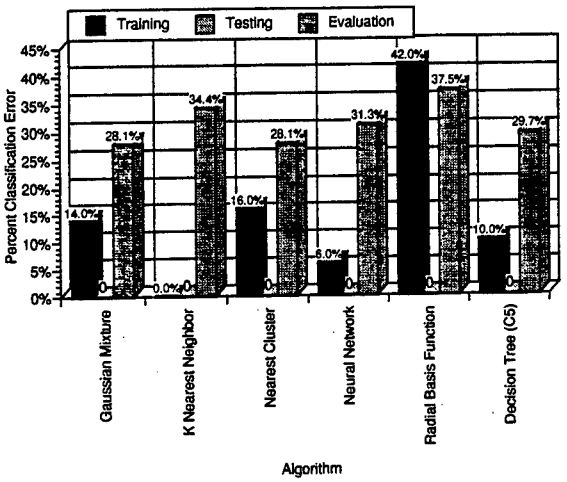


Figure 3.1: Single Model Results on Glass Data

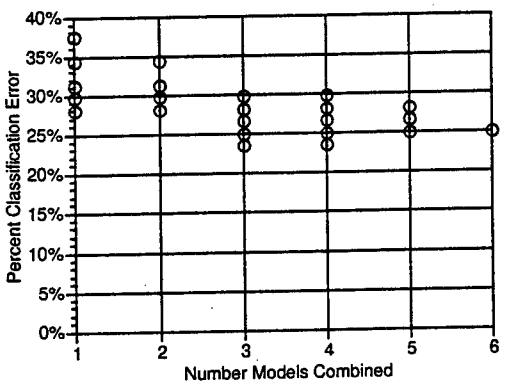


Figure 3.2: Combine Model Results on Glass Data

Table 3.1 Combinations Yielding Lowest Error Rate

k-Nearest Neighbor		Neural Networks	Radial Basis Functions
Decision Trees,	Nearest Cluster		Radial Basis Functions
Decision Trees	Gaussian Mixture		Radial Basis Functions
k-Nearest Neighbor	Nearest Cluster	Neural Networks	Radial Basis Functions

Table 3.2 Greater than 3-way Combinations Yielding Highest Error Rate (29.7%)

Decision Trees	k-Nearest Neighbor	Nearest Cluster		
Decision Trees,			Radial Basis Fn.	Neural Networks
	k-Nearest Neighbor	Nearest Cluster	Radial Basis Fn.	
	k-Nearest Neighbor	k-Nearest Neighbor		Neural Networks

When the combination model results are represented only by the summary statistics minimum, maximum, and average error, the trends become clearer, as seen in Figure 3.3.

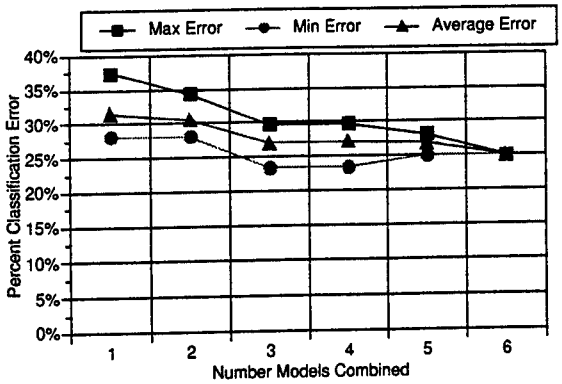


Figure 3.3: Combine Model Trend on Glass Data

The average model error never gets worse as more models are added to the combinations. Additionally, the spread between the best and the worst shrinks as the number of models combined increases: both bias and variance are reduced: the error was reduced by 4.7%, a 16.7% error reduction compared to the best Gaussian mixture model which had 28.1% error. However, the reduction found here is not as good as the reduction found by Brieman [5] using *boosting* (Figure 3.4), which brought the error down to 21.6%.

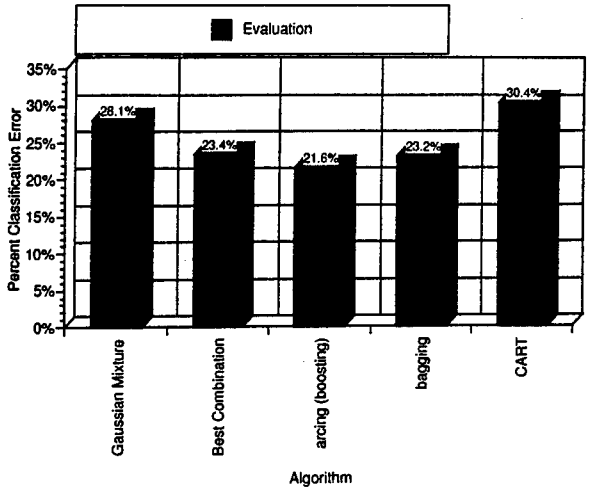


Figure 3.4: Comparison of Model Combination with Breiman Arcing.

3.2. Satellite Dataset Results

Results for the satellite data are similar to the glass data. First see in Figure 3.5 the train, test, and evaluation results for the single models. Results are more uniform than for the glass data, but radial basis functions and decision trees are the worst performers on evaluation data. The best are nearest neighbor, neural networks, and Gaussian mixture models.

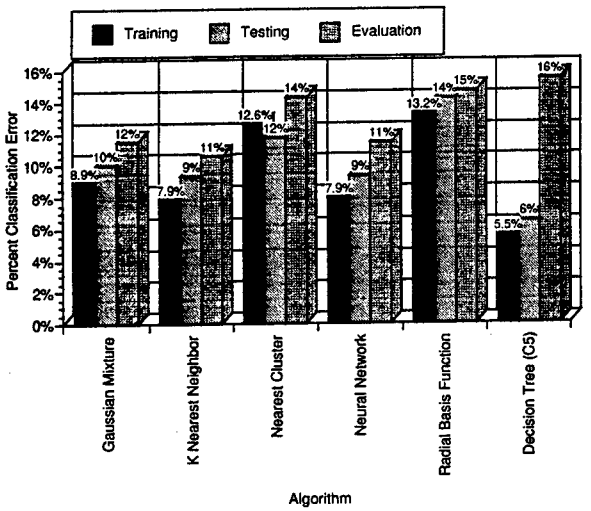


Figure 3.5: Single Model Results on Satellite Data

The trends are shown in Figure 3.6. Once again the errors and the spread between maximum and minimum errors are both reduced as the number of combined models increases, though once again the very best models occur for the 3-way combination (k-Nearest Neighbor, Neural Network, Radial Basis Function, and the same three with Decision Trees). Once again, radial basis functions are involved in the best combination, and again are also involved in the worst combination models.

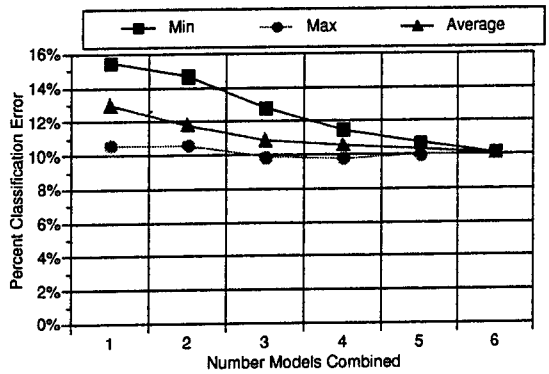


Figure 3.6: Combine Model Trend on Satellite Data

Comparing the model combination results to Breiman's results using Arcing (boosting) shows once again the boosting algorithm performing better, though the combination betters bagging by a small amount here.

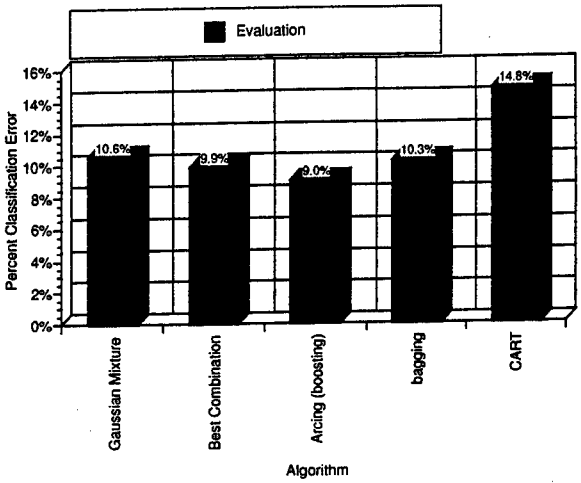


Figure 3.7: Comparison of Model Combination with Breiman Arcing.

4. Conclusions and Discussion

Clearly, combining models improves model accuracy and reduces model variance, and the more models combined (up to the number investigated in this paper), the better the result. However, determining which individual models combine best from training results only is difficult—there is no clear trend. Simply selecting the best individual models does not necessarily lead to a better combined result.

While combining models across algorithm families reduces error compared to the best single models, it does not perform as well as boosting. The advantage of boosting over simple model combining is that boosting acts directly to reduce error cases, whereas combining works indirectly. The model combining voting methods are not tuned to take into account the confidence that a classification decision is made correctly, nor do they concentrate more heavily on the difficult cases. More research is necessary to confirm these suggested explanations.

References

[1] Elder, J.F., and Pregibon, D. 1995. A Statistical Perspective on Knowledge Discovery in Databases. *Advances in Knowledge Discovery and Data Mining.* U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Editors. AAAI/MIT Press.

[2] Elder, J. F. IV, D.W. Abbott, Fusing Diverse Algorithms,. *29th Symposium on the Interface,* Houston, TX, May 14-17, 1997.

[3] Dietterich, T. 1997. Machine-Learning Research: Four Current Directions. *AI Magazine.* 18(4): 97-136.

[4] D. Steinberg, *CART Users Manual,* Salford Systems. 1997.

[5] Breiman, L. 1996. "Arcing Classifiers", *Technical Report,* July 1996.

[6] J. Friedman, T. Hastie, and R. Tibsharani, "Additive Logistic Regression: A Statistical View of Boosting", Technical Report, Stanford University, 1998.

[7] D.H. Wolpert, Stacked generalization, *Neural Networks* **5**: 241-259, 1992.

[8] L. Breiman, "Bagging predictors", *Machine Learning* **24**: 123-140, 1996.

[9] Y. Freund, and R.E. Shapire, "Experiments with a new boosting algorithm", In *Proceedings of the Thirteenth International Conference on Machine Learning*, July 1996.

[10] Abbott, D.W., "Comparison of Data Analysis and Classification Algorithms for Automatic Target Recognition", *Proc. of the 1994 IEEE International Conference on Systems, Man, and Cybernetics*, San Antonio, TX, October, 1994.

[11] Unica Technologies, Inc., http://www.unica-usa.com

[12] SPSS, Inc., http://www.spss.com/software/clementine

[13] R.L. Kennedy, Y. Lee, BV Roy, C.D. Reed, and R.P. Lippman, *Solving Data Mining Problem Through Pattern Recognition*, Prentice Hall, PTR, Upper Saddle River, NJ, 1997.

[14] http://www.ics.uci.edu/~mlearn/MLRepository.html

[15] D. Michie, D. Spiegelhalter, and C. Taylor, *Machine Learning, Neural and Statistical Classification*, Ellis Horwood, London, 1994.

On Optimal Projective Fusers For Function Estimators

Nageswara S. V. Rao
Computer Science and Mathematics Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831-6355

Abstract *We propose a fuser that projects different function estimators in different regions of the input space based on the lower envelope of the error curves of the individual estimators. This fuser is shown to be optimal among projective fusers and also to perform at least as well as the best individual estimator. By incorporating an optimal linear fuser as another estimator, this fuser performs at least as well as the optimal linear combination. We illustrate the fuser by combining neural networks trained using different parameters for the network and/or for learning algorithms.*

Keywords: Sensor fusion, fusion rule estimation, empirical estimation

1 Introduction

Recently, combinations of estimators have been shown to be very effective in a number of disciplines such as forecasting, reliability, and pattern recognition (see [6] for an overview). In specific methods such as neural networks, it has been shown that better performance can be achieved by suitably combining the networks rather than choosing the best [1, 9, 4]. The reasons for the success of the combination methods are often problem-specific:

(a) errors from different estimators may cancel one another, and

(b) certain estimators, although have a high overall error, might achieve lower error in certain regions of input space.

Roughly speaking, the linear combinations of estimators exploit the scenario (a) [3, 2, 5]. In this paper, we present a projective method that exploits the scenario (b). In general, estimators are designed to achieve a low overall error but not necessarily low local errors in all regions of the input space. Our method is aimed at exploiting the local behavior of the various individual estimators.

We consider the problem of estimating a function $f : \Re^d \mapsto [0,1]$ based on a sample $(X_1, f(X_1)), (X_2, f(X_2)), \ldots, (X_n, f(X_n))$, where $X_1, X_2, \ldots, X_n$ are randomly generated according to the distribution P_X. An estimator $\hat{f} : \Re^d \mapsto [0,1]$ has a square error of $(f(X) - \hat{f}(X))^2$ at a given $X \in \Re^d$. The quality of the estimator $\hat{f}$ for f is given by the *expected error* defined as

$$I(\hat{f}) = \int_{X \in \Re^d} (f(X) - \hat{f}(X))^2 dP_X.$$

Consider that we are given N function estimators $\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_N$. We are required to compute a fuser that combines these estimators such that the fuser guarantees the performance of the best estimator as a minimum.

The linear combinations are one of the widely employed fusers for function estimators [3, 2], and for neural network estimators in particular [1, 9, 4]. The projective fusers proposed here are qualitatively different from the linear fusers and provide complementary performances. Informally speaking, a linear fuser chooses a fixed constant for each estimator for the entire range of x which could make it ineffective in certain localities. The projective fusers, on the other hand, exploit the local errors of the estimators. Furthermore, if a linear

fuser is very effective, then it can be incorporated as constituent estimator for a projective fuser which performs at least as well as the best of its constituent estimators.

Projective fusers are described in Section 2. Comparison with linear fusers, and methods to combine linear and projective fusers are provided in Sections 3 and 4, respectively. In Section 5, we describe simulation results dealing with fusing a set of neural networks which are trained with different parameters.

2 Projective Fusers

A *projective fuser*, f_P, corresponding to a *partition* $P = \{\pi_1, \pi_2, \ldots, \pi_k\}$, $k \leq N$, of input space $\Re^d$ ($\pi_i \subseteq \Re^d$, $\bigcup_{i=1}^{k} \pi_i = \Re^d$, and $\pi_i \cap \pi_j = \phi$ for $i \neq j$), assigns to each block π_i to an estimator $\hat{f}_j$ such that

$$f_P(x, \hat{f}_1, \ldots, \hat{f}_N) = \hat{f}_j(x)$$

for all $x \in \pi_i$. For simplicity, we denote $f_P(x, \hat{f}_1, \ldots, \hat{f}_N)$ by $f_P(x)$. An *optimal projective fuser*, denoted by f_{P^*}, minimizes $I(.)$ over all projective fusers corresponding to all partitions of $\Re^d$ and assignments of blocks to estimators $\hat{f}_1, \hat{f}_2, \ldots \hat{f}_N$.

We define the *error curve* of the estimator $\hat{f}$ for f as $\mathcal{E}(x, \hat{f}) = (f(x) - \hat{f}(x))^2$. The *error-curve projective fuser* is defined by

$$f_{EC}(x, \hat{f}_1, \ldots, \hat{f}_N) = \hat{f}_{i_{EC}(x)}(x)$$

where

$$i_{EC}(x) = \arg \min_{i=1,2,\ldots,N} \mathcal{E}(x, \hat{f}_i).$$

In other words, $f_{EC}(x, \hat{f}_1, \ldots, \hat{f}_N)$ simply outputs the output of the estimator which has the lowest error at x. Thus, we have $\mathcal{E}(x, f_{EC}) = \min_{i=1}^{N} \mathcal{E}(x, \hat{f}_i)$, or equivalently the error curve of f_{EC} is the lower envelope with respect to x of the set of error curves $\{\mathcal{E}(x, \hat{f}_1), \ldots, \mathcal{E}(x, \hat{f}_N)\}$.

Example 1: We first consider a simple example where $f(x) = 1_{[1/4,3/4]}(x)$ for $x \in [0,1]$,

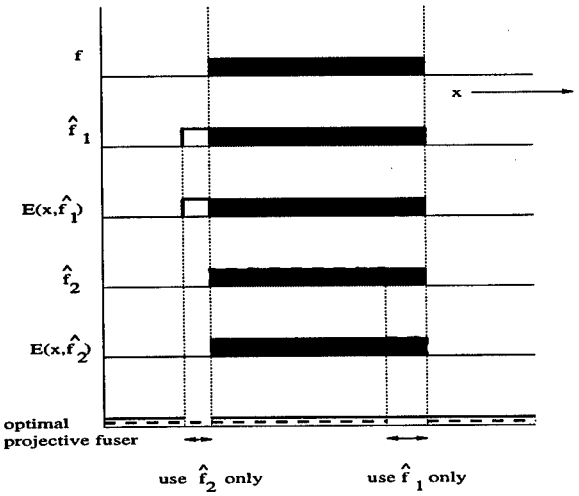


Figure 1: *Illustration of error-curve projective fuser.*

as in Figure 1, where $1_A(x)$ is the *indicator function* which has a value 1 if and only if $x \in A$ and has value 0 otherwise. For $\hat{f}_1(x) = 1_{[1/4-\epsilon_1,3/4]}(x)$ and $\hat{f}_2(x) = 1_{[1/4,3/4-\epsilon_2]}(x)$ for some $0 < \epsilon_1, \epsilon_2 < 1/4$. The error curves are given by $\mathcal{E}(x, \hat{f}_1) = 1_{[1/4-\epsilon_1,1/4]}(x)$ and $\mathcal{E}(x, \hat{f}_1) = 1_{[3/4-\epsilon_2,3/4]}(x)$, which correspond to disjoint intervals. The lower envelope of the two error curves is the zero function hence $I(f_{CE}) = 0$. The profile of f_{CE} is shown at the bottom of Figure 1, wherein $\hat{f}_1$ and $\hat{f}_2$ are projected in the intervals $[3/4 - \epsilon_2, 3/4]$ and $[1/4 - \epsilon_1, 1/4]$, respectively, and in other regions either estimator can be projected. □

For any projective fuser $f_P(x, \hat{f}_1, \ldots, \hat{f}_N)$, let $i_P(x)$ denote the index of the estimator such that $f_P(x) = \hat{f}_{i_P(x)}(x)$. Then, for $x \in \Re^d$, we have

$$\begin{aligned}(f(x) - f_P(x))^2 &= (f(x) - \hat{f}_{i_P(x)}(x))^2 \\ &\geq (f(x) - \hat{f}_{i_{EC}(x)}(x))^2 \\ &= (f(x) - f_{EC}(x))^2.\end{aligned}$$

Thus, we have $\mathcal{E}(x, \hat{f}_{i_P(x)}) \geq \mathcal{E}(x, f_{i_{EC}(x)})$ for all $x \in \Re^d$. By taking the expectations on both sides, we have $I(f_{EC}) \leq I(f_P)$, and hence f_{EC} is an optimal projective fuser.

We close this section by showing that f_{EC} in not optimal in a larger class of fusers that can

project a function of the output (as opposed to just the output) of an estimator.

Example 2: Consider $f(x) = 1_{[1/4,3/4]}(x)$, $\hat{f}_1(x) = 1_{[1/4-\epsilon_1,3/4-\epsilon_1]}(x)$, and $\hat{f}_2(x) = 1_{[1/4,3/4-\epsilon_2]}(x)$ for some $0 < \epsilon_1, \epsilon_2 < 1/8$, and $\epsilon_1 < \epsilon_2$. Thus, we have $\mathcal{E}(x, \hat{f}_1) = 1_{[1/4-\epsilon_1,1/4]}(x)$ and $\mathcal{E}(x, \hat{f}_1) = 1_{[3/4-\epsilon_2,3/4]}(x)$, whose lower envelope is not the zero function. Thus, we have $\mathcal{E}(x, f_{CE}) = 1_{[3/4-\epsilon_2,3/4-\epsilon_1]}(x)$ and

$$I(f_{CE}) = \int_{[3/4-\epsilon_2,3/4-\epsilon_1]} dP_X > 0,$$

for uniform P_X. By changing the assignment of f_{CE} to $1-\hat{f}_1$ for $x \in [3/4-\epsilon_2, 3/4-\epsilon_1]$, one can easily achieve zero error. □

3 Linear Fusers

We now compare the performance of projective fusers with linear fusers. A linear fuser is defined as

$$f_L(x, \hat{f}_1, \ldots, \hat{f}_N) = \sum_{i=1}^{N} \alpha_i \hat{f}_i(x)$$

for some $(\alpha_1, \ldots, \alpha_N) \in \Re^d$. An *optimal linear combination fuser*, denoted by f_{L^*}, minimizes $I(.)$ over all linear combinations. Roughly speaking, the performance of f_{P^*} is better than f_{L^*} if the individual estimators perform better in certain localized regions of $\Re^d$. On the other hand, if the estimators are equally distributed around f in a global sense, f_L performs better as illustrated follows.

Example 3: In the Example 1, for $f_L = \alpha_1\hat{f}_1 + \alpha_2\hat{f}_2$, we have

$$\begin{aligned} I(f_L) = \; & \alpha_1^2 \int_{[1/4-\epsilon_1,1/4)} dP_X \\ & +(1-\alpha_1-\alpha_2)^2 \int_{[1/4,3/4-\epsilon_2)} dP_X \\ & +(1-\alpha_1)^2 \int_{[3/4-\epsilon_2,3/4]} dP_X \end{aligned}$$

which is non-zero no matter what the coefficient are. The error curves of $\hat{f}_1$ and $\hat{f}_2$ take non-zero values in the intervals $[1/4-\epsilon_1, 1/4]$ and $[3/4-\epsilon_2, 3/4]$, respectively. Since these intervals are disjoint, there is no possibility of error of one estimator being cancelled by a scaler multiplier of the other. The disjointness of $[1/4-\epsilon_1, 1/4]$ and $[3/4-\epsilon_2, 3/4]$ yields $\mathcal{E}(x, f_{EC}) = 0$, and hence $I(f_{P^*}) = 0$. □

The conclusions of this example are true in general that if the error curves of the estimators take non-zero values on disjoint intervals, then any linear fuser will have a non-zero error. On the other hand, the disjointness of the error curves is sufficient to yield zero error for the optimal projective fuser.

We now present an example where a linear fuser outperforms f_{P^*}.

Example 4: Consider $f = 1$ for $x \in [0,1]$, $\hat{f}_1(x) = \epsilon x + 1 - \epsilon$, and. $\hat{f}_1(x) = -\epsilon x + 1 + \epsilon$, for $0 < \epsilon < 1$. The optimal linear fuser is given by $f_L(x) = 1/2(\hat{f}_1(x) + \hat{f}_2(x)) = 1$ for $x \in [0,1]$. At every $x \in [0,1]$, we have

$$\mathcal{E}(x, \hat{f}_1) = \mathcal{E}(x, \hat{f}_2) = \epsilon^2(1-x)^2 = \mathcal{E}(x, f_{p^*}).$$

Thus, $I(f_{P^*}) = \epsilon^2 \int_{[0,1]} (1-x)^2 dP_X > 0$ for a non-discrete P_X, whereas $I(f_L) = 0$. □

In Example 4, the error curves of the estimators are "symmetrically" distributed around the function so that error of one is cancelled by a scalar multiple of the other.

In summary, the performance of the optimal linear and projective fusers are complementary as illustrated in this section.

4 Composite Fusers

We now discuss the *isolation property* of the fuser class that ensures that the fuser is at least as good as the best estimator. A fuser class $\mathcal{G} = \{g(x, f_1, \ldots, f_k)\}$ has the *isolation property* with respect to f_i if it contains the function $g_i(x, f_1, \ldots, f_k) = f_i(x)$ for all $x \in \Re^d$ and all $i = 1, 2, \ldots, k$. The isolation property was first proposed in [8, 7] for concept and sensor

data size		projective	other better		performance (times)		average
training	test	as good	linear	best	linear	best network	error
Without noise							
10	10	8	1	1	1.009269	10.489711	0.075042
25	25	8	2	0	1.039855	13.426878	0.021926
50	50	10	0	0	1.304039	31.157175	0.013454
75	75	10	0	0	1.530556	89.050201	0.004725
100	100	10	0	0	1.788104	87.905518	0.003764
With noise							
10	10	8	2	0	0.982823	9.205843	0.041874
25	25	8	2	0	1.045973	14.115362	0.026983
50	50	10	0	0	1.293410	19.121033	0.010399
75	75	9	1	0	1.275850	33.192585	0.008435
100	100	10	0	0	1.227069	37.937778	0.007115

Table 1: Computational results for f_1.

fusion problems. If $\mathcal{G}$ is the set of linear combinations, i. e. $g(x, f_1, \ldots, f_k) = \alpha_1 f_1(x) + \ldots + \alpha_k f_k(x)$, for $\alpha_i \in \Re$, this property is trivially satisfied for each of f_i, $i = 1, 2, \ldots, k$. Similarly, the class of projective fusers satisfies the projection property with respect to $\hat{f}_i(x)$ for $i = 1, 2, \ldots, k$, wherein g_i corresponds to entire $\Re^d$ forming one block assigned to the single estimator $\hat{f}_i$.

Consider that $\mathcal{G} = \{g(x, \hat{f}_1, \ldots, \hat{f}_N)\}$ satisfies the isolation property, then we have for all $i = 1, 2, \ldots, k$,

$$\inf_{g \in \mathcal{G}} \int (f(X) - g(X))^2 dP_X \leq \int (f(X) - \hat{f}_i(X))^2 dP_X$$

which implies $\inf_{g \in \mathcal{G}} I(g) \leq \min_{i=1}^{N} I(\hat{f}_i)$. Hence, by including the optimal linear combination as $\hat{f}_{N+1}$, we can guarantee that

$$I(f_{P^*}(x, \hat{f}_1, \ldots, \hat{f}_N, f_{L^*})) \leq I(f_{L^*})$$

by the isolation property of projective fusers. Since linear combinations also satisfy the isolation property, we have

$$I(f_{L^*}) \leq \min_{i=1}^{N} I(\hat{f}_i).$$

The roles of f_{L^*} and f_{P^*} can be switched such that

$$f_{L^*} = \alpha_1^* \hat{f}_1 + \ldots + \alpha_N^* \hat{f}_N + \alpha_{N+1}^* f_{P^*}$$

for suitable $\alpha_i^* \in \Re$, for $i = 1, \ldots, N$. Then by the isolation properties of linear and projective fusers, we have

$$I(f_{L^*}) \leq I(f_{P^*}(., \hat{f}_1, \ldots, \hat{f}_N)) \leq \min_{i=1}^{N} I(\hat{f}_i).$$

5 Simulation Example

We implemented six neural network estimators for the target function [4]

$$f_2(x) = 0.02(12 + 3x - 3.5x^2 + 7.2x^3)(1 + \cos 4\pi x)(1 + 0.08 \sin 3\pi x).$$

For each network, the number of hidden nodes is randomly chosen. Each network is trained with backpropagation algorithm with a different learning rate which is again randomly chosen. Then we compute empirical versions of optimal linear and projective fusers. Entire computation is repeated with several training and test sample sizes. We illustrate a typical run in Figure 2 based on 50 training and 50 testing examples. The neural network 3 has the lowest mean test error. The network 6 does

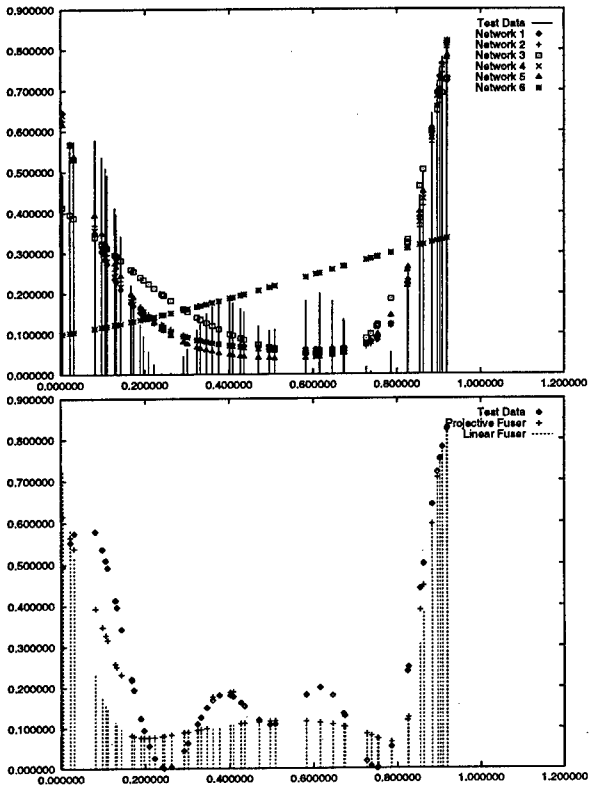


Figure 2: *Performance of neural networks trained with different parameter values.*

not have the lowest mean test error, but for the region around $x = 0.4$ it has the lowest error.

The linear fuser is computed for the neural networks which is then combined with the six networks to compute the projective fuser proposed here. Note that the linear fuser does not perform well in the region $x = 0.4$ but the network 6 does, which is utilized by the projective fuser. The linear fuser reduces the test error of the best network by a factor of 31.15 times, and the projective fuser reduces that of linear fuser by another factor of 1.3 times.

The fusers computation is performed under different sample sizes and with additional noise. For each sample size, the computation is performed 10 times. The performance of the fuser is compared with the linear fuser and the best neural network. In most cases, the projective fuser performed better than linear fuser and significantly better than the best neural network as indicated in Table 1.

6 Conclusions

We presented a class of fusers that project individual estimators at various points in the input space. We identified optimal fuser in this class, and compared it with the linear fusers, which are the most applied and analyzed fusers. The projective fusers provide a complementary performance compared to linear fusers. By suitably combining projective fusers with linear fusers, composite fusers can be obtained which are at least as efficient as the best of the projective and linear fusers.

Future research directions include the cases that involve randomness in the function estimators, and finite sample implementation and performance analysis of projective fusers.

Acknowledgements

This research is sponsored by the Engineering Research Program of the Office of Basic Energy Sciences, U.S. Department of Energy, under Contract No. DE-AC05-96OR22464 with Lockheed Martin Energy Research Corp., and the Office of Naval Research under order N00014-96-F-0415.

References

[1] E. Alpaydin. Multiple networks for function estimation. In *Proceedings of International Conference on Neural Networks*, pages 9–14, 1993.

[2] L. Brieman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[3] L. Brieman. Stacked regressions. *Machine Learning*, 24(1):49–64, 1996.

[4] S. Hashem. Increasing model accuracy using optimal linear combinations of trained neural networks. *IEEE Transactions on Neural Networks*, 6(3):792–794, 1995.

[5] S. Hashem. Optimal linear combinations of neural networks. *Neural Networks*, 10(4):599–614, 1997.

[6] R. N. Madan and N. S. V. Rao. Guest editorial on information/decision fusion with engineering applications. *Journal of Franklin Institute*, 336B(2), 1999. 199-204.

[7] N. S. V. Rao. Fusion methods for multiple sensor systems with unknown error densities. *Journal of Franklin Institute*, 331B(5):509–530, 1994.

[8] N. S. V. Rao, E. M. Oblow, C. W. Glover, and G. E. Liepins. N-learners problem: Fusion of concepts. *IEEE Transactions on Systems, Man and Cybernetics*, 24(2):319–327, 1994.

[9] G. Rogova. Combining the results of several neural network classifiers. *Neural Networks*, 7(5):777–781, 1994.

A New Multiple Model Framework for Recursive Bayesian Modelling of Time Series by Neural Networks

T.J. Dodd and C.J. Harris
Image, Speech and Intelligent Systems Research Group
Department of Electronics and Computer Science
University of Southampton, Southampton, U.K.

Abstract *Bayesian parameter estimation for neural network type models with batch learning is now well established. This allows the incorporation of a priori knowledge about the solution and naturally results in regularisation. Recent results have shown that this framework can be extended to sequential data for time series problems. However, this approach assumes that the correct model structure is known a priori and that the hyperparameters of the model can be estimated accurately. These conditions may not be met in practise which suggests that an alternative approach may be required. Multiple model approaches are investigated as a solution where each model is allowed to adapt its parameters sequentially. A committee of models is trained each with a different structure and/or initialisations of the hyperparameters. A simple weighted combination rule is found for the committee of models based on Gaussian assumptions. This approach is found to demonstrate good performance for nonlinear, nonstationary time series problems.*

Keywords: multiple models, Kalman filter, neural networks, Bayesian modelling, time series

1 Introduction

Time series modelling plays an important role in many data fusion problems, including target tracking, condition monitoring, robot navigation and guidance, and collision avoidance. The nature of these environments is such that we require models which are nonlinear, nonstationary and can adapt on-line to new data. In this paper we present an approach to this nonlinear, non-stationary problem using a multiple model framework in which we fuse together the outputs from a bank of models.

We have recently developed a Bayesian solution to the recursive estimation of certain classes of neural networks [1] for time series modelling. Under Gaussian approximations for the noise and parameters we can train general linear models (GLiM) using the Kalman filter. The algorithm recursively learns the network parameters from sequential data and incorporates on-line regularisation.

A significant drawback of the above framework is the a priori requirement of the structure and hyperparameters of the model. We have investigated an adaptive solution to the estimation of the hyperparameters based on a maximum evidence framework. We found that this approach leads to consistently biased estimates for the hyperparameters. There is also no immediately obvious solution to the adaptive estimation of the network structure, for example the number, position and size of the basis functions and input variable selection.

We propose in this paper to use a multiple model framework to overcome the limitations described above. The framework consists of a bank of models and a combination rule. Each model represents a different realisation of the uncertain hyperparameters and/or model structures. The outputs of the models are combined linearly based on the estimated posterior probabilities of these outputs.

We demonstrate the above approach on the estimation of nonlinear non-stationary time se-

ries. The first problem is an illustrative demonstration problem where the committee consists of models with different structures. The second problem is motivated by an analytical model of slender delta wings [2]. The committee is based on a set of identical models where the parameters and hyperparameters are randomly initialised. In both examples the performance of the committee is better than the average of the individual models and in certain instances better than the best individual model.

2 Bayesian Parameter Estimation

We now consider a class of models which are linear in the parameters. For these general linear models (GLiMs) the output is simply a weighted linear combination of a fixed set of m basis functions. This class of models includes many standard functional approximators including B-splines, polynomials, Fourier series and certain classes of feedforward neural networks.

We assume the data arrive sequentially and that at any time instant $k+1$ the complete set of observations $\mathcal{Z}^{k+1} = \{z_1, z_2, \ldots, z_{k+1}\}$ are available. In deriving the recursive parameter estimation we will subsequently see that, in fact, only the current observation z_{k+1} is required. The physical model of the system consists of two equations, the first, for the observations describes the data and is given by

$$z_{k+1} = \boldsymbol{\phi}^T(\mathbf{x}_{k+1})\mathbf{w}_{k+1} + \varepsilon_{k+1} \qquad (1)$$

where $\mathbf{w} \in \mathbb{R}^m$ is a vector of unknown parameters, $\boldsymbol{\phi}(\cdot)$ is a vector of fixed basis functions and the subscript $k+1$ indicates that the value of the quantity is taken as at time $k+1$. The noise term ε_{k+1} is assumed to be zero-mean with time-varying variance $\sigma^2_{\varepsilon,k+1}$. The evolution of the parameters is described by the second equation

$$\mathbf{w}_{k+1} = \mathbf{F}_k\mathbf{w}_k + \boldsymbol{\Gamma}_k\boldsymbol{\xi}_k \qquad (2)$$

where $\mathbf{F}_k \in \mathbb{R}^{m\times m}$ and $\boldsymbol{\Gamma}_k \in \mathbb{R}^{p\times m}$ are assumed known and possibly time varying, and $\boldsymbol{\xi}_k \in \mathbb{R}^p$ is a sequence of zero-mean Gaussian noise with covariance

$$E[\boldsymbol{\xi}_k\boldsymbol{\xi}_k^T] = \mathbf{Q}_k.$$

Eq. 2 is a general form for the parameter updates in which each parameter is updated as a linear combination of the current parameters plus some random component to account for unknown effects. In the simplest, and perhaps most useful, case where

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \boldsymbol{\xi}_k \qquad (3)$$

and $\boldsymbol{\xi}_k \in \mathbb{R}^m$ this update law has a simple interpretation. We are assuming that the new (updated) parameters are equal to the old ones plus some random component. The degree to which the parameters are allowed to vary is controlled by the covariance of the noise term $\boldsymbol{\xi}_k$. For the trivial case where $\mathbf{Q}_k = \lambda_k\mathbf{I}_m$ then λ_k acts as a learning rate with large parameter updates being associated with large λ_k.

The recursive form of Bayes' rule for the parameters is [3]

$$p(\mathbf{w}_{k+1}|\mathcal{Z}^{k+1}) = \frac{p(z_{k+1}|\mathbf{w}_{k+1})p(\mathbf{w}_{k+1}|\mathcal{Z}^k)}{p(z_{k+1}|\mathcal{Z}^k)}. \qquad (4)$$

The posterior density over the parameters given the observed data $p(\mathbf{w}_{k+1}|\mathcal{Z}^{k+1})$ is a function of:

- the likelihood $p(z_{k+1}|\mathbf{w}_{k+1})$ which reflects the prediction the model makes about the new observation z_{k+1} for the particular values of the parameters;
- the updated prior $p(\mathbf{w}_{k+1}|\mathcal{Z}^k)$ which is equal to the predicted parameter values given only the observations upto and including the current time step k; and
- the normalising constant (evidence) $p(z_{k+1}|\mathcal{Z}^k)$.

We make the additional assumptions that the noise terms are mutually independent and independent of the parameters, i.e.

$p(\varepsilon_{k+1}, \boldsymbol{\xi}_k|\mathbf{w}_k) = p(\varepsilon_{k+1})p(\boldsymbol{\xi}_k)$ and that the probability density $p(\mathbf{w}_{k+1}|\mathcal{Z}^{k+1})$ is Gaussian with mean $\hat{\mathbf{w}}_{k+1}$ and covariance $\mathbf{P}_{k+1}$.

Substituting for these densities in Bayes' rule, Eq. 4 and after some simplification we then find for the posterior mean value of the parameters

$$\hat{\mathbf{w}}_{k+1} = \mathbf{F}_k\hat{\mathbf{w}}_k + \mathbf{G}_{k+1}(z_{k+1} - \boldsymbol{\phi}^T(\mathbf{x}_k)\mathbf{F}_k\hat{\mathbf{w}}_k)$$

where

$$\mathbf{G}_{k+1} = \mathbf{M}_{k+1}\boldsymbol{\phi}(\mathbf{x}_k)(\boldsymbol{\phi}^T(\mathbf{x}_k)\mathbf{M}_{k+1}\boldsymbol{\phi}(\mathbf{x}_k) + \sigma^2_{\varepsilon,k+1})^{-1}$$

and the posterior covariance matrix for the parameters

$$\mathbf{P}_{k+1} = \left(\mathbf{M}^{-1}_{k+1} + \frac{1}{\sigma^2_{\varepsilon,k+1}}\boldsymbol{\phi}(\mathbf{x}_k)\boldsymbol{\phi}^T(\mathbf{x}_k)\right)^{-1}. \quad (5)$$

The estimated parameters are a function only of the parameters and observed data at the current time step. This defines a Markov process or sequence where all the available information upto the current time step is summarised by the data at the current time step. So we see that, under the Gaussian assumptions, the Bayesian formulation of the parameter estimation problem results in a simple recursive relation for the evolution of the parameters.

At a particular time step $k+1$ our prediction for a new input $\mathbf{x}_{k+1}$ will then have mean

$$y(\mathbf{x}_{k+1}, \hat{\mathbf{w}}_{k+1}) = \boldsymbol{\phi}^T(\mathbf{x}_{k+1})\hat{\mathbf{w}}_{k+1} \quad (6)$$

and the variance about this mean is given by

$$\sigma^2_y(\mathbf{x}_{k+1}) = \boldsymbol{\phi}^T(\mathbf{x}_{k+1})\mathbf{P}_{k+1}\boldsymbol{\phi}(\mathbf{x}_{k+1}). \quad (7)$$

To arrive at the final predictive variance we assume that the noise variance $\sigma^2_{\varepsilon,k+1}$ is uncorrelated with the output variance $\sigma^2_y(\mathbf{x}_{k+1})$ and simply sum the two terms, i.e. the final predictive variance is equal to $\sigma^2_{\varepsilon,k+1} + \sigma^2_y(\mathbf{x}_{k+1})$.

3 Discussion

The Bayesian approach described in the previous section provides a natural framework for parameter estimation where data arrive sequentially and we desire to incorporate some form of prior knowledge. Such an approach allows us to specify a prior over the parameters based on a priori knowledge. This prior is then modified as more data becomes available. This is intuitively what we desire as for little data we need the parameters to behave in a reasonable manner as given by our prior. However as more data becomes available the impact of the prior should be lessened to the point where, for an infinite data set, the prior has no influence and the parameter values are inferred totally from the data.

An implicit assumption with the above approach is that the model structure is known a priori. For a general linear model this means specifying the number, type and positions of the basis functions. By imposing such structure we are actually incorporating a form of prior knowledge as the structure will, to an extent, determine the classes of functions that the model can approximate. We must also determine the number of inputs which, we will see, for time series means the number of time delayed versions of the observations or derivatives thereof. For certain problems the model structure may be learnt off-line. However, if the time series is nonstationary or we are uncertain as to the correct model structure then it would seem appropriate to use multiple model structures covering a variety of a priori expected situations.

In order to implement the learning strategy described above we must estimate the covariances $\mathbf{Q}_k$ and $\sigma^2_{\varepsilon,k+1}$ where estimating $\mathbf{Q}_k$ is usually reduced to estimating the learning parameter λ_k. Estimating these so called hyperparameters in a stable, unbiased fashion where there may be nonstationarities in the data presents significant difficulties [1]. A possible solution to overcoming the problems of estimating the hyperparameters is to train a committee of models each with different ini-

tialisations of the hyperparameters and then combining the final outputs. It would then be hoped that any effects of poor initialisations would be averaged out.

4 Time Series Prediction

We are interested in the representation of time series as simple functions of time

$$y(t) = f(t), \tag{8}$$

as nonlinear autoregressive models

$$y(t) = g(y(t-1), \ldots, y(t-d+1)) \tag{9}$$

or in nonlinear differential form

$$y(t) = h(\dot{y}(t), \ddot{y}(t), \ldots, y^{(d)}(t)) \tag{10}$$

where $y^{(d)}(t) \equiv \frac{d^d y}{dt^d}$. The theoretical motivation for using the latter two models is well established [4, 5, 6, 7]. In particular, under certain assumptions, a time series can always be represented in the forms, Eq. 9 and 10, provided d is great enough. In the case of noise-free observations then $d \geq 2n+1$ is sufficient where n is the natural order of the state of the underlying dynamical system generating the time series.

The particular choice of model depends very much on the goal of this modelling. If we are simply interested in making estimates about the time series, for example to find missing data, estimating the derivatives of the function or making short term predictions, then Eq. 8 is appropriate. However, for system identification purposes, where we are interested in finding a model of the long term behaviour of the system, then the models, Eq. 9 and 10 are more appropriate. In this paper we will look at an example of short term prediction and a system identification problem using the nonlinear differential form.

5 Combining Models

The key feature of the model described above is that, via the assumption of a Gaussian density over the parameters and taking a Bayesian perspective, it is possible to assign confidence intervals to the predictions. These confidence intervals are of a known form and are in fact Gaussian given by Eq. 7. We assume then that we have a committee of such networks, and at any time step each model makes a prediction of the output with Gaussian density. Then how can we combine these outputs in a consistent manner? This basic issue has been addressed by Manyika and Durrant-Whyte [8] where they look at the generic problem of combining probabilistic information for multiple sensors. Our situation is slightly different as the information source is the same for each model. The theoretical motivation for the combination strategy described below is described in a companion paper [9]. Here we simply provide a general discussion of the underlying ideas of combining probabilistic information.

Each model is making predictions based on a common set of observations, $\mathcal{Z}^{k+1}$, and the output of each model can be considered as a local posterior $p_i(y|\mathcal{Z}^{k+1})$ with mean $y_i(\mathbf{x}_{k+1}, \hat{\mathbf{w}}_{k+1})$ and variance $\sigma^2_{y_i}(\mathbf{x}_{k+1})$ where $i = 1, \ldots, M$ and M is the total number of models in the committee. In deciding how to combine the predictions we must consider the nature of the prior information for each model and the conditional independence of the inputs. Now, it is immediately obvious that the inputs are the same for each model and therefore they cannot be conditionally independent. However, the prior information in each model is likely to be independent as we are deliberately setting the structure and/or hyperparameters of each model to be different.

The combination strategy should embody certain characteristics: an ability to reinforce opinion, a reduction in uncertainty over any single model in the committee, the combination rule should be simple and computationally inexpensive, consistency with the nature of the outputs of the models. A strategy which embodies all these characteristics is to simply form the product of the probability density functions of the predictions.

We know that, for M models, the outputs of the models are Gaussian with mean y_i and

variance $\sigma^2_{y_i}$, $i = 1, \ldots, M$. Assuming, then, that these outputs are independent then the output of the committee will be given by

$$p(y) = \prod_{i=1}^{M} p_i(y)$$

where $p_i(y)$ are the probability densities of the outputs of the individual models. As the $p_i(y)$ are Gaussian then the output of the committee, $p(y)$, will also be Gaussian with mean [9]

$$\mu_y = \frac{\sum_{i=1}^{M} \frac{y_i}{\sigma^2_{y_i}}}{\sum_{i=1}^{M} \frac{1}{\sigma^2_{y_i}}} \tag{11}$$

and variance given by the equation

$$\frac{1}{\sigma^2_y} = \sum_{i=1}^{M} \frac{1}{\sigma^2_{y_i}}. \tag{12}$$

6 Results

In this section we consider the application of the multiple recursive model approach to two time series problems.

6.1 Simulated Time Series

This example is a simulated time series which is highly nonstationary. The data were generated from the function:

$$z(t) = 0.1\sin(t) + \exp\left\{-\frac{1}{0.7}(t-5)^2\right\} + 0.4\exp\left\{-\frac{1}{0.1}(t-8)^2\right\} + \varepsilon(t)$$

where the noise process $\varepsilon(t)$ has zero mean and variance 0.0025. A committee of four networks was training using the noisy observations. The details of each network are summarised in Table 1 where the basis functions were equally spaced over the interval $[0, 10]$. In each case the parameters and hyperparameters were randomly initialised using Gaussian probability density functions. The parameters were updated as described previously and the hyperparameters were estimated using an evidence framework [10, 11].

Model	No. Basis Fns.	Variance
1	41	0.25
2	21	1.00
3	14	2.00
4	11	4.00

Table 1:

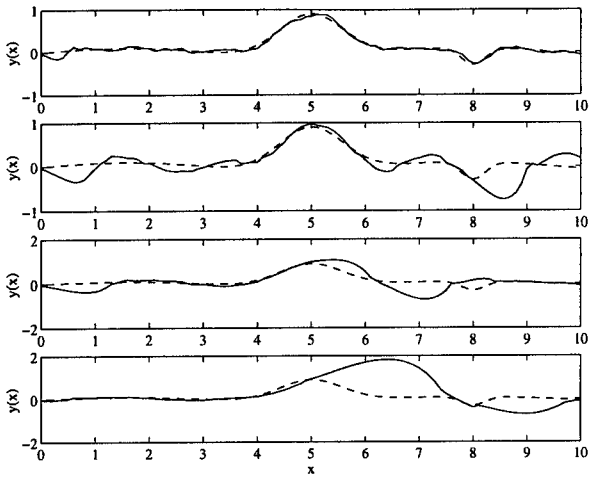


Figure 1: Comparison between predicted [–] and actual [- - -] outputs for the simulated signal. The different predictions correspond to the different network structures described in the main text.

The outputs of the four models and associated predicted variances are shown in Figures 1 and 2. We see that as the number of basis functions decreases and associated widths of the basis functions increases the predicted outputs tend to be smoother and less able to model the fine detail. The is reflected in the predicted variances for the model outputs, Figure 2. Where the output of the function is relatively smooth the models with larger basis functions perform best. However, in the regions of greatest curvature the variances increase dramatically reflecting a lack of confidence in the predictions. The committee therefore includes models which are suited to different regions of the function.

The combined output of the committee is

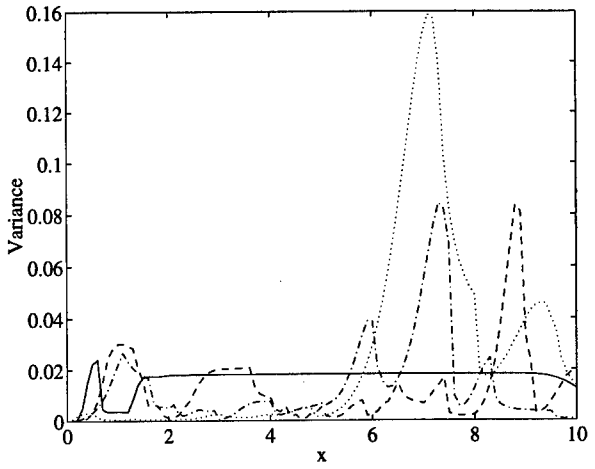


Figure 2: The predicted variances over the data set for each model. Model 1 [–], model 2 [- - -], model 3 [–·] and model 4 [···].

shown in Figure 3 which tracks the true output reasonably accurately. The important point to note is that from a set of models, three of which perform relatively poorly the average output performs comparatively as well as the best model.

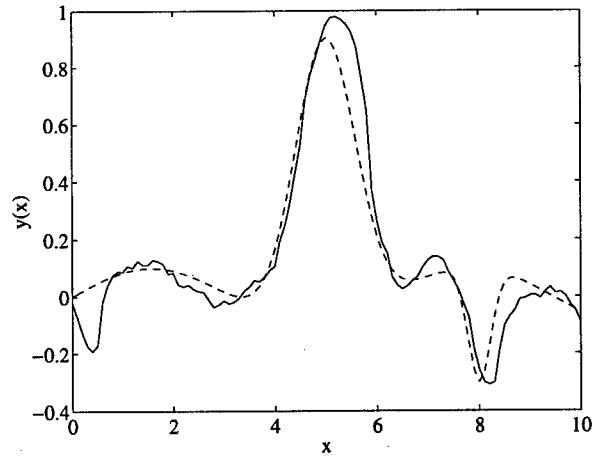


Figure 3: The predicted output [–] versus the true output [- - -] for the simulated signal. Whilst the combined fit is not as good as that for the first model it shows a significant improvement over the other models.

6.2 Wing Rock in Slender Delta Wings

The second problem is based on an analytical study of wing rock in slender delta wings [2]. The system is described by a set of continuous time nonlinear differential equations:

$$\begin{aligned} \dot{x}_1 &= \dot{\theta} = x_2 \\ \dot{x}_2 &= \ddot{\theta} \\ &= c_1 a_1 x_1 + (c_1 a_2 - c_2) x_2 + c_1 a_3 x_1^3 \\ &\quad + c_1 a_4 x_1^2 x_2 + c_1 a_5 x_1 x_2^2 \end{aligned} \tag{13}$$

where $c_1 = 0.354$ and $c_2 = 0.001$ and the a_i vary with the angle of attack α of the wing. We simulated the model using a 4th order Runge-Kutta method in order to generate the inputs $x_1 = \theta$ and $x_2 = \dot{\theta}$. The simulated data for the output $\ddot{\theta}$ were then generated by adding Gaussian noise of variance 0.01 to the simulated output from Eq. 13. The wing was simulated for a stable case with an angle of attack of 15° for which $a_1 = -0.01026, a_2 = -0.02117, a_3 = -0.14181, a_4 = 0.99735, a_5 = -0.83478$.

A committee of eight radial basis function networks was training recursively using the noise corrupted outputs. The structure of each network was the same with the centres of the Gaussian basis functions equally spaced over the input domain and a total of 49 basis functions used. A weight decay prior was used in each network and the parameters and hyperparameters were given different random initialisations. Whilst for individual networks the initialisations of the (hyper)parameters results in markedly different performance it was hoped that by effectively averaging over these effects the committee would, on average, perform satisfactorily.

The prediction performance of the individual models and the committee, in terms of mean squared error (MSE), is shown in Figure 4. The MSE is shown for the three best individual networks along with that of the committee and the average for the individual models. We can see that the performance of the committee outperforms, by some margin, the average of the models and is generally at least

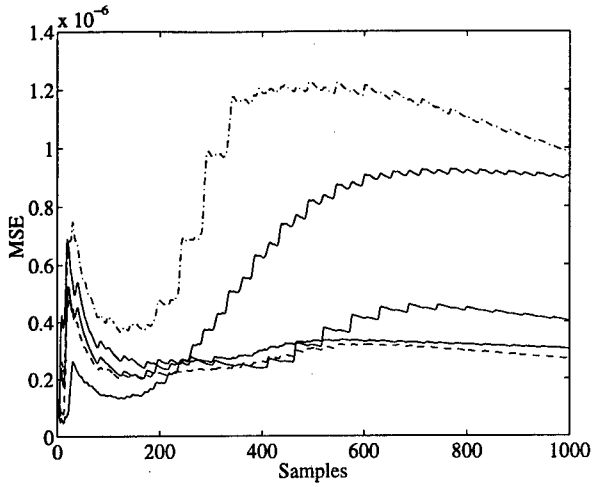


Figure 4: Mean-squared error across the data for the wing rock data set. Individual models [–], average of models [–·] and committee of models [- - -].

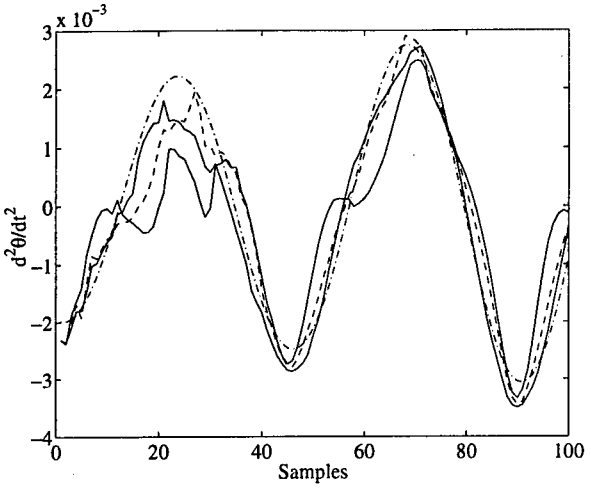


Figure 5: Prediction over the first 100 data points for the wing rock data set showing relative performance between individual models and output from the committee of models. Individual models [–], committee [- - -] and true output [–·].

as good as the best individual models. Of particular importance is the lack of a steep increase in the MSE evident for most of the models. This increase is probably due to the training algorithm being insufficient to deal with the nonstationarities in the data over this period.

The average mean squared error for the individual models over the simulation period varied between 3.1383×10^{-7} and 2.5176×10^{-6} whilst that of the committee was only 2.7493×10^{-7}. At any particular time instant, as expected, at least one of the individual models usually performed better than the committee. However, this was not always the case and for a small number of time instances the committee actually performed better than the best individual model.

The averaging effect of the committee can also be seen in Figure 5 which shows the predictions over the initial 100 samples. The prediction from the committee shows good correspondence with the true output. However, the predictions from the two individual models shows a marked difference in performance. It is this random nature, whereby models can show good performance in certain regions and poor performance in others, that the committee tends to alleviate.

7 Conclusions

A recursive Bayesian approach to parameter estimation applicable to certain classes of neural networks has been described. Via Gaussian assumptions a simple linear combination rule for committees of such models was presented. Time series examples have demonstrated that this approach can be applied successfully to problems where the time series is nonlinear and nonstationary. The resulting predictions are more robust to parameter and network initialisations than for individual models.

8 Acknowledgements

The authors would like to thank Matra British Aerospace Dynamics and EPSRC for their partial support of this work.

References

[1] T.J. Dodd and C.J. Harris. Recursive Bayesian modelling of time series by neural networks. 1999. Accepted for presentation at ICANN 99, Edinburgh.

[2] A.H. Nayfeh, J.M. Elzebda, and D.T. Mook. Analytical study of the subsonic wing-rock phenomenon for slender delta wings. *Journal of Aircraft*, 26(9):805–809, September 1989.

[3] Y.C. Ho and R.C.K. Lee. A Bayesian approach to problems in stochastic estimation and control. *IEEE Transactions on Automatic Control*, AC-9:333–339, October 1964.

[4] Dirk Aeyels. Generic observability of differentiable systems. *SIAM J. Control and Optimization*, 19(5):595–603, 1981.

[5] I.J. Leontaritis and S.A. Billings. Input-output parametric models for non-linear systems part I: Deterministic non-linear systems. *International Journal of Control*, 41(2):303–328, 1985.

[6] I.J. Leontaritis and S.A. Billings. Input-output parametric models for non-linear systems part II: Stochastic non-linear systems. *International Journal of Control*, 41(2):329–344, 1985.

[7] Floris Takens. Detecting strange attractors in turbulence. In D.A. Rand and L.S. Young, editors, *Dynamical Systems and Turbulence*, pages 366–381. Proceedings of a Symposium Held at the University of Warwick, 1979/80, Springer-Verlag, 1981.

[8] James Manyika and Hugh Durrant-Whyte. *Data Fusion and Sensor Management: a decentralized information-theoretic approach.* Ellis Horwood, 1994.

[9] T.J. Dodd and C.J. Harris. Committees of Gaussian kernel based models. 1999. Also to be presented at Fusion '99.

[10] A.H. Jazwinski. Adaptive filtering. *Automatica*, 5:475–485, July 1969.

[11] J.F.G. de Freitas, M. Niranjan, and A.H. Gee. Hierarchical Bayesian-Kalman models for regularisation and ARD in sequential learning. Technical Report CUED/F-INFENG/TR 307, Cambridge University Engineering Department, 1998.

Session TC1
International Collaboration in Fusion Research and Development
Chair: James Llinas
State University of New York at Buffalo, NY, USA

Data Fusion: The Benefits of Collaboration and Barriers to the Process

Mark Bedworth[1], Jane O'Brien[1] and James Llinas[2]

[1]Jemity, PO Box 113, Malvern,
Worcestershire, WR14 3YJ, UK
Email: {mark.bedworth, jane.obrien}@datafusion.clara.net

[2]State University of New York (SUNY) at Buffalo,
NY, USA
Email: llinas@acsu.buffalo.edu

Abstract – *We describe some aspects of the data fusion community infrastructure. We set them in the context of the technology transfer cycle and sub-divide this cycle by identifying the main players at each stage. We argue that the disparate nature of data fusion (both the technologies that it encompasses and the domains in which it can be applied) makes a co-operative approach, not just desirable, but necessary. We highlight the lack of substantial international collaboration as a key barrier to the establishment of an effective data fusion community. Such collaboration is fraught with questions, which are listed and then elucidated upon. The paper is intended to catalyse discussion on this subject rather than to provide answers to all the questions. For this purpose we take a somewhat provocative stand on those elements of data fusion which have been found lacking.*

Keywords: Collaboration, community, data fusion, society.

1. Introduction

The global data fusion community has seen a recent acceleration in its development. There are now thousands of data fusion researchers and systems engineers worldwide. There is a fledgling society and several fusion related conferences. Now seems a good time to take stock of where the field has evolved to and to make some strategic decisions regarding its future development. This paper illuminates some of the current issues and identifies the difficulties that we will have to face. It is intended as a catalyst for discussion rather than a prescription for success.

2. The Fusion Cottage Industry

There are many researchers and users of data fusion technology throughout the world. Despite this, however, many of them are working in isolation. This is a lamentable situation which Llinas [1] likens to a cottage industry. Researchers may be unaware that they are working on a recognised technology with a growing community. This may lead to:

- implementation of inappropriate solutions;
- re-invention of existing techniques;
- duplication of effort;
- under-utilisation of their results.

Of particular relevance to the data fusion community is the issue of resourcing. Despite falling defence budgets the allocation of funding to data fusion projects is approximately stable. Furthermore, as the exploitation of data fusion in the commercial world matures, the industrial applications funding of data fusion is likely to increase. As a result of this and other factors there is a worldwide scarcity of high-calibre data fusion researchers. It is a pity that this finite resource is

currently being deployed so inefficiently from a global perspective.

Many of these problems could be overcome if a proper data fusion community were to be established. There are a number of existing initiatives addressing this community issue.

3. The Fusion Community

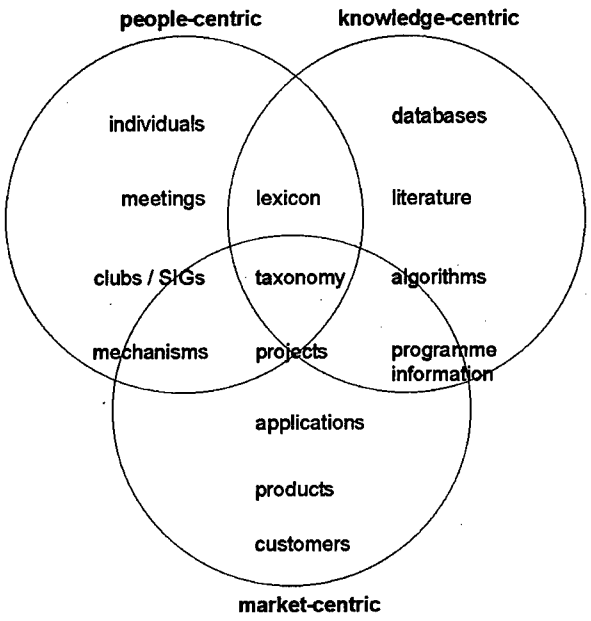


Figure 1: The community issues in the context of different viewpoints.

A fusion community should assume several different viewpoints:

- the people;
- the knowledge they develop;
- the market they work in;
- the communication processes they use.

There are several pressing issues associated with each of these viewpoints as shown in Figure 1.

3.1 People Centric Issues

There is a global shortage of scientists and engineers who wish to pursue a data fusion career and have the appropriate academic qualifications. The pervasive nature of data fusion (and therefore its broad technical background) has partly been responsible for its under-representation in educational establishments. Current data fusion experts generally have a mathematical, engineering or computer science background and have migrated into data fusion from a related field such as pattern recognition or control theory. Their knowledge has often been acquired on-the-job rather than as part of a formal training programme.

There are now a small number of short courses available for providing introductions to data fusion techniques and applications. There is no agreed syllabus for such courses, nor is there a central source of information on them. A list of approved courses offering a standardised core syllabus should be a community priority.

The situation in academia is even worse. There are currently no postgraduate courses devoted specifically to data fusion anywhere. This shocking situation has prompted the present authors to initiate plans for a masters-level data fusion course on both sides of the Atlantic.

We should also realise that training and education is no longer the sole responsibility of universities. Nor is it entirely appropriate for companies to produce specialists through on-the-job training. We propose that a co-operative approach in which industry and academia work as a partnership is more suitable for such a field as diverse as data fusion.

3.2 Knowledge Centric Issues

Data fusion knowledge may be embodied in many forms. In some cases the evolved communal activity in establishing this knowledge has led to significant successes (the well understood principles of decision fusion, for example). However, in many instances whole areas of the field have been largely ignored:

Algorithms and Tools

There is no widely used, openly accessible, library of data fusion techniques and software modules. There is still a significant amount of nugatory

effort on re-implementation (for instance, nearly every researcher has their own code to implement a Kalman filter). Many fields now have accepted implementations of standard algorithms, see [2] for example.

Models

There is an (over?) abundance of data fusion models. Each of these addresses a slightly different aspect of the system design problem. It would be highly desirable to establish a standard that provided the flexibility to match most situations [3].

Architectures

There is no agreed recommendation of which architecture to use for any particular data-application-model combination. Many researchers develop their own, which results in solutions that cannot easily be integrated into a complete system.

Frameworks

Proponents of the main inferencing frameworks (probabilistic, possibilistic and evidential) have historically taken a somewhat entrenched attitude. There are few systematic comparisons of frameworks on realistic scenarios and a definitive and quantitative data fusion perspective is long overdue.

Datasets

Very few properly ground-truthed, multi-sensor datasets are available for open dissemination and re-use. A fusion equivalent to the machine learning repository held at the University of California in Irvine [4] would greatly enhance the ability to compare methods on common data.

Metrics

Fusion is essentially a system-level activity. For it to be taken seriously as a scientific endeavour it must allow measurements between prediction and reality **at this system level**. The definition of such measures of effectiveness is woefully inadequate and their use currently confined to well-constrained applications.

The communal knowledge may be capitalised upon by:

- archiving – of all aspects of knowledge in the form of easily-accessible on-line tutorials, papers, bibliographies and (pseudo) code segments;
- dissemination – of information: there are now three open, international data fusion conferences each year (SPIE [5], FUSION [6] and EuroFusion [7]). Thankfully there is useful co-ordination between the organisers of these events:

3.3 Market Centric Issues

Researchers should be mindful that the majority of their work is funded by market need (whether initially identified by the researcher or the customer) [8]. For nearly two decades the application of data fusion technologies lay almost solely within the defence domain including:

- surveillance and reconnaissance;
- air defence;
- intelligence analysis;
- non co-operative target recognition.

During the last few years, however, the benefits of fusion have found more widespread use. There is now a substantial worldwide interest in the use of data fusion in:

- aerospace industries;
- medical applications;
- machine condition monitoring;
- process monitoring;
- remote sensing;
- industrial robotics.

Despite the increase in the number of application domains for data fusion technology, there are still many relevant areas where data fusion is still not used. It behoves data fusion practitioners to champion the technology in these new domains.

3.4 Communication Centric Issues

The transfer of technology from the domain of intellectual concepts to tangible, marketable products may be regarded as a cyclic communication process, as shown in Figure 2. The energy required for maintaining this cycle stems from:

- an intellectual capability supplied by a continuing education program;
- stable funding derived from a sustainable market need.

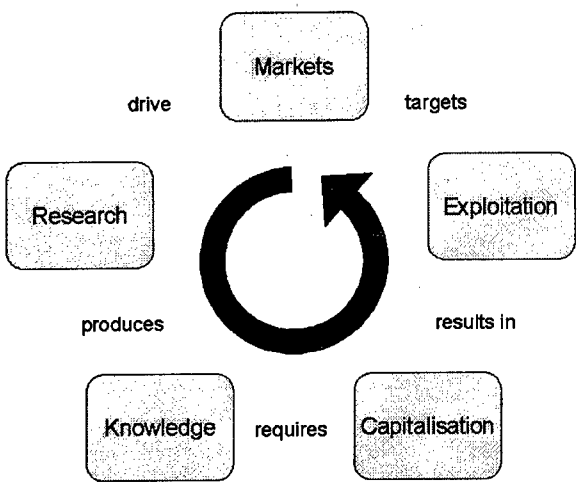


Figure 2. The technology transfer cycle

It is possible to identify several stereotypical rôles:

- the businessman – who is motivated by the current revenue-producing data fusion product and the reassurance that there will be something new to market;
- the scientific researcher – who is driven by the creation and extension of knowledge but who is funded (possibly indirectly) by the market requirements;
- the collator and archivist – who adds value by collecting, collating and storing the accumulated knowledge. They can add more value by facilitating its appropriate dissemination;
- the system engineer – who capitalises on the communal data fusion knowledge to produce solutions to realistic tasks;
- the business development manager – who is able envision a market niche for a technical data fusion solution and to exploit such a solution in the marketplace.

Some people (those who will make the biggest fusion community contribution) are involved at several stages. Others concentrate solely on one aspect and remain ignorant of the bigger picture.

4. Fusion Community Requirements

The development of a community has several requirements. These include the establishment of a set of standards, a knowledge repository and interaction and collaboration amongst the groups involved. Of these, collaboration is perhaps the hardest to achieve (international collaboration may be particularly difficult).

4.1 International Standards

To assist in the globalisation of data fusion an international standard for data fusion models, architectures and frameworks should be established. A lexicon of accepted definitions should be provided so that different groups can communicate their ideas effectively. A methodology for testing data fusion algorithms, and a standard set of problems would place data fusion system engineering on a firmer foundation.

Some national efforts have been made to establish data fusion standards including models (*e.g* US and UK), lexicons (*e.g.* US and Australia) and guidelines (*e.g.* UK and US). These need to be made truly international.

4.2 A Knowledge Repository

An openly accessible and maintained repository of the collective data fusion knowledge should incorporate:

- a directory of experts and groups giving their main areas of interest;
- links to other information sources (such as conferences);

- a bibliography (preferably with some annotation);
- case studies describing the lessons learnt from applications of data fusion;
- standards (as described above).

4.3 Collaboration

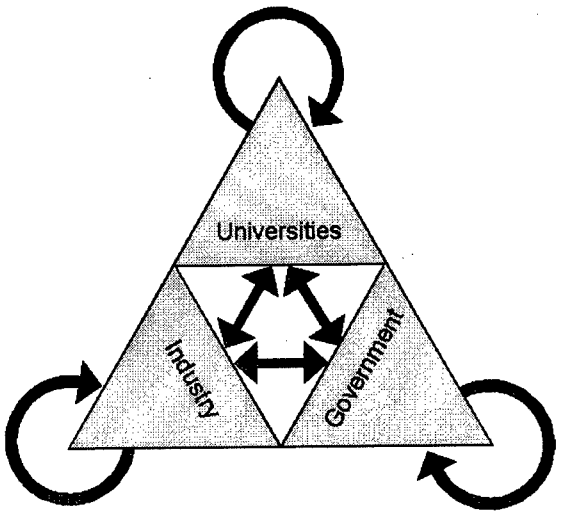


Figure 3: The required communication between different data fusion work cultures

Collaboration can take place in any part of the technology transfer cycle. The cycle can be augmented with details of the drivers, agents and recipients at each stage. These each belong to one of three main work cultures:

1. industry – characterised by short timescales and driven by revenue production for stakeholders. They engineer a provided technical solution into a working product and can therefore be thought of as engineering providers.
2. government – characterised by large (and lengthy) procurement projects and driven by politics. They generally produce solutions rather than products for industry and can be thought of as science and technology providers;
3. universities – characterised by long-term research and driven by intellectual achievement. Universities also educate and train the personnel who will later produce the science and technology and, hence, they are the education providers.

The driving force in each of these areas is described below [9].

Engineering drivers of collaboration are predominantly in industry. The requirement is for collaborators who can identify and understand real problems and provide workable solutions. As such, they often collaborate with government research laboratories and universities, rather than with other industries.

The **science and technology drivers** are often research laboratories. Some of these are industry-based, but the majority are at universities and in government organisations. They require links with mainstream academia to provide them with suitably trained staff. Their main collaborative efforts are with industry as a user of their output and a source of funded applications.

The **educational drivers** are mainly in universities. Their collaborative efforts are directed towards industry and research laboratories. Industry provides them with both a research focus and a user of their results. Research laboratories provide extra manpower on industrial projects and an additional source of project work.

5. Collaboration Benefits and Barriers

Collaboration has the mutual benefit of increased efficiency via the gearing that is obtained by the sharing of objectives and the risk reduction of using different approaches to similar problems. Collaborations of any sort, however, may encounter some difficulties, including:

- the use of different context and definitions;
- the lack of regular communication or direction;
- the parochial attitudes of potential collaborators.

Some forms are intrinsically more problematic than others owing to work culture or geographical factors.

5.1 Cultural Factors

There may be significant differences in ethos, beliefs and values between the different work cultures identified in Figure 3. If parties from different work cultures co-operate, then particular barriers to successful collaboration are introduced.

Co-operation between organisations within the same work culture should represent the easiest form of collaboration. The individuals are often from similar backgrounds and share similar constraints and desires. However, they also compete for the same resources (market, human or financial for example). As competitors they will enter into collaborations only when the benefits of exploitation are clearly and fairly laid out. Issues to be addressed include intellectual property rights, royalties and market exclusivity agreements.

Collaboration between work cultures removes this problem to some extent since the exploitation routes can often be apportioned in an obvious manner (for example intellectual property owned by the university and market exploitation rights for the industry). Inter-cultural collaborations, however, also bring additional difficulties of disparate values and different constraints. These include the differing time-scales, separate contractual requirements, potentially different fiscal cycles, the disparate views of risk management and the fundamental differences in what outcomes are regarded as worthwhile.

5.2 Geographical Factors

International collaboration which takes place within the same work culture but in different countries brings its own set of problems. These include different fiscal cycles, currency fluctuations, legal systems and national constraints (such as security). Even different time zones can cause a problem. Communication between project members is also made more difficult by distance and the cost of face-to-face meetings adds substantially to the overheads of the collaboration.

6. Forms of Collaboration

Collaboration can take many forms spanning informal information exchange on a mutually interesting topic, short-term scientist exchange and the establishment of a virtual laboratory. With the use of modern communications technology (Email, internet and video conferencing for example) such collaborative working should not, in principal, be difficult to achieve.

One difficulty that constantly arises is in finding suitable collaborators. One needs to determine what their interests are, how they operate and how their capabilities match ones own. An alternative to the traditional, serendipitous, approach is to establish a directory of data fusion research groups. On its own, however, this is not enough since such contacts only establish the mutual desire for co-operation. For successful long-term collaboration to occur a mechanism is also needed.

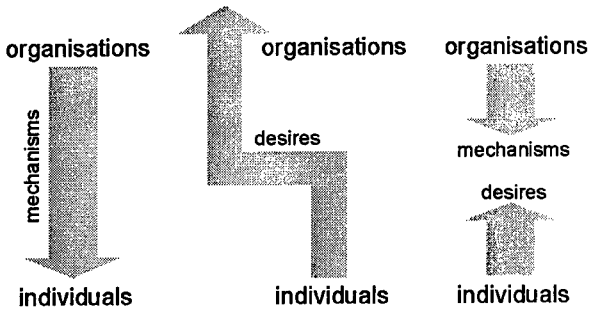


Figure 4: The three motivators for collaboration: top-down, bottom-up and outside-in.

The match between desires and mechanisms leads to three basic types of collaboration. In the first a large organisation creates a mechanism and imposes actions on groups of individuals. This group collaborates because of the top-down drive. Secondly, a group may come together because they share a common desire but be unable to find an appropriate mechanism. If the need is strong enough they will find a way around this problem – the collaboration is driven from the bottom-up. The ideal case involves a good match between desires and mechanisms and can be thought of as outside-in collaboration.

6.1 Examples of Top-Down Collaboration

A top-down collaboration is one that is envisaged by a government or large organisation. A mechanism is generally provided ahead of the formation of the co-operating group. Top-down collaborations usually cease when the funding policy changes. In some cases they then transform into bottom-up collaborations. A small selection of top-down collaborations includes:

Technology Foresight – a UK Government initiative started in the mid 1990's which set up panels to critically review the state-of-the-nation in a few key technologies, one of which was data fusion [10]. The Data Fusion Working Group identified cross-cultural collaboration as a primary issue and recommended the creation of a mechanism for defence and aerospace partnerships to facilitate co-operation between UK government, industry and academia.

Faraday INTErSECT– the INTElligent SEnsors for Control Technologies partnership [11] includes research and exploitation in data fusion for the multi-sensor engine as on eof its three main themes. A substantial amount of funding is forthcoming from government and industrial sponsors in collaboration with UK universities. The purpose is to create opportunities for technology transfer with science-push and market-pull explicitly identified.

DFSG - in 1997 the UK Ministry of Defence provided baseline funding to establish the Defence Evaluation and Research Agency Data Fusion Strategy Group. The aim of this group was to facilitate co-ordination of all the projects within DERA which had an element of data fusion in them. This was, and still is, an awareness and information exchange project. It is not aimed at developing and applying data fusion technology [12,13].

6.2 Bottom-Up Collaboration

A bottom-up collaboration is driven by a group of individuals who perceive a need and co-operate without substantial support of large organisations or governments. Such collaborations are often very successful but are also fragile since they are generally not robust to the movements of individuals. Examples of bottom-up collaborations include:

JDL DFG – the Joint Directors of Laboratories Data Fusion Group has produced insightful analyses of data fusion and provided the most widely used fusion models and fusion taxonomies. This group continues due to the commitment and dedication of its members.

ISIF – the International Society of Information Fusion is in its formative stages. It will be some time before ISIF is self-sustaining and in the meantime it continues to develop due solely to the hard work of a few key individuals whose time and effort is not funded.

Information Fusion Journal – the need for a fusion journal has been widely acknowledged for some time. The forthcoming Elsevier publication was conceived and brought to fruition largely by the single-handed (unpaid) efforts of its editor.

Clubs and Special Interest Groups – there is now an electronic club relating to information fusion and a special interest group dedicated to sensor fusion management [14,15]. These valuable forums are kept alive by their founders and the members that regularly contribute to them.

6.3 Outside-in Collaboration

A outside-in collaboration forms when there is a desire on the part of individuals **and** the simultaneous existence of a mechanism to achieve the activity. Outside-in collaborations are often the most successful examples of collaboration, both in terms of output and longevity. There are currently a few developing examples of outside-in collaborations in data fusion:

DARP – The UK Defence and Aerospace Research Partnerships are a result of the Technology Foresight initiative described above. The Government is providing baseline funding to facilitate this activity. In the UK there is a current DARP on data fusion that includes a government laboratory, several major UK industries and a number of British universities.

FUSIAC – The American FUSion Information Analysis Center is currently being brought into existence. It will provide some of the archiving and dissemination activities discussed earlier and will have US government funding. It is unclear how it will work alongside ISIF and whether it can successfully operate internationally.

7. Food for Thought

We have presented a structured list of issues and problems relating to the establishment of a global data fusion community with ongoing international collaborations. We believe the following to be of the highest priority:

- Easing of (inter)national co-operation:
  - What desires are shared – should there be a directory resource?
  - What mechanisms are appropriate (NATO, TTCP, MOU, bi-lateral, multi-lateral) and who are the points of contact?
  - How do we create more outside-in collaborations?

- Archiving and dissemination of communal knowledge:
  - What should the resource contain?
  - Where should it be held?
  - Who should maintain the resource?

- Data fusion in education:
  - What level of education is appropriate (undergraduate, masters or doctorate)?
  - What should be included in an agreed core syllabus?
  - Should virtual courses be offered which are taught at several universities?
  - How should the coupling with government and industry be handled?

The present authors would encourage discussions on these issues and would welcome specific suggestions for developing the data fusion community infrastructure.

References

[1] J. Llinas, "Center of Excellence in Multisource Information Fusion", Proceedings of the Symposium on Multisensor Fusion, Georgia Institute of Technology (1998).

[2] W. Press, S. Teukolsky, W. Vetterling and B. Flannery, "Numerical Recipes", Cambridge University Press (1988).

[3] M. Bedworth and J. O'Brien, "The Omnibus Model: A New Model of Data Fusion?", Proceedings of FUSION99, Sunnyvale (1999).

[4]ftp://ftp.ics.uci.edu/pub/machine-learning-databases

[5] http://members.tripod.com/~belur/sfaaa.html

[6] http://www.inforfusion.org/fusion99/

[7] http://www.ee.surrey.ac.uk/EuroFusion99/EuroFusion99.htm

[8] R. Mahler, "Transitioning Data Fusion R&D to Industry: Some Issues", Proceedings of the Symposium on Multisensor Fusion, Georgia Institute of Technology (1998).

[9] M. Bedworth, "UK Data Fusion System Development", Proceedings of SMI Conference on Military Data Fusion, London (1999).

[10] M. Markin, C. Harris, M. Bernhardt, J. Austin, M. Bedworth, P. Greenway, R. Johnston, A. Little and D. Lowe, "Technology Foresight on Data Fusion and Data Processing", Publication of The Royal Aeronautical Society, (1997).

[11] http://ftp.npl.co.uk/intersect/index.html

[12] J. O'Brien, "MoD / DERA Data Fusion Strategy Group: Initial Report", DERA/CIS/CIS5/CR97439 (1998).

[13] J. O'Brien, "MoD / DERA Data Fusion Strategy Group: Interim Report", DERA/S&P/SPI/CR990005 (1999).

[14]http://clubs.yahoo.com/clubs/informationfusion/resourceandsensorcontrol

[15] http://clubs.yahoo.com/clubs/

Session TC2
Diagnostic Information Fusion
Chair: Kai Goebel
GE Information Technology Lab, NY, USA

Chu Spaces - A New Approach to Diagnostic Information Fusion

Hung T. Nguyen
Department of Mathematical Sciences
New Mexico State University
Las Cruces, NM 88003-8001, U.S.A.
hunguyen@nmsu.edu

Vladik Kreinovich
Department of Computer Science
University of Texas at El Paso
El Paso, TX 79968, USA
vladik@cs.utep.edu

Berlin Wu
Department of Mathematical Sciences
National Chengchi University
Taipei, Taiwan
berlin@math.nccu.edu.tw

Abstract *This paper is rather theoretical. Its aim is to describe a general algebraic framework, known as Chu spaces, in which different type of information can be transformed into the same form, so that fusion procedures can be investigated in a single general framework.*

Keywords: Chu spaces, data fusion, fuzziness, probability

1 A Motivating Example

Data fusion means that we combine ("fuse") several pieces of information (measurement results, expert estimates) about one or several objects. To describe our new approach to formalizing data fusion, we will start with a physically meaningful (and mathematically simple) example.

In order to find the location of distant radio sources, we measure the signals from these sources received on different radiotelescopes, and then fuse the measurement results. The larger the telescope, the more accurate the measurements. Therefore, to achieve maximum accuracy, antennas forming a radiotelescope are placed as far away from each other as possible: ideally, on different continents. The resulting Very Long Baseline Interferometry method (VLBI, for short) works as follows: whenever a pair of antennas is oriented towards a radio source (e.g., a quasar), we record the signals $s_1(t)$ and $s_2(t)$ on these two antennas, and compare the records. From trigonometry, one can easily deduce that the difference between the lengths of the paths from the source to the two antennas is equal to $\tau = \vec{B} \cdot \vec{s}$, where $\vec{B}$ is a *baseline* (i.e., a vector from the first to the second antenna), and $\vec{s}$ is a unit vector in the direction of the radio source. This difference in paths leads to the corresponding difference $\Delta t = \tau/c$ between the times when the same signal reaches the two antennas (here, c is the speed of light, with which the radio signal travels). Thus, the signal $s_1(t)$ recorded by one of the antennas is *delayed* by Δt from the signal recorded by the second one. Hence, by comparing the signals $s_1(t)$ and $s_2(t)$, we can determine the delay Δt and therefore, the value $\tau = c \cdot \Delta t = \vec{B} \cdot \vec{s}$.

Our goal is to determine the source location (i.e., the vector $\vec{s}$). If we knew the baselines *exactly*, then we would get a system of linear equations for finding $\vec{s}$. In real life, we only know the *approximate* values of $\vec{B}$, and the exact values of the baselines must be determined

by the same measurements. In other words, for different baselines $\vec{B}_i$ and for different sources $\vec{s}_j$, we measure the values $\tau_{ij} = \vec{B}^{(i)} \cdot \vec{s}^{(j)}$; we would like to extract, from the exact measurement results, the exact values of the source locations $\vec{s}^{(j)}$.

The corresponding problem has two aspects:

- First, a *theoretical* (*fundamental*) aspect: If we make a sufficient number of measurements, *can we*, in principle, uniquely reconstruct all the locations? If we cannot reconstruct all the locations uniquely, then what exactly information about the source locations can be determined?
- Second, a *practical* (*computational*) aspect: *how* can we actually extract the locations $\vec{s}^{(j)}$ (or whatever information we can) from the measurement results τ_{ij}?

2 Chu Spaces and Automorphisms

2.1 General Description of Data Fusion: Chu Spaces

We have described the data fusion problem on one specific example. In general:

- We have a set of objects of interest which we will denote by X; in the above example, each object of interest $x \in X$ is characterized by a unit vector $\vec{s}$, so we can say that X is the set of all possible unit vectors.
- We also have a set of measuring instruments (or estimators) which will be denoted by A; in the above example, measuring instruments are pairs of antennas; each pair is characterized by its baseline vector $\vec{B}$, so we can say that A is the set of all (3-D) vectors.
- We assume that the construction of measuring instruments is known, and therefore, if we know the exact parameters of the object $x \in X$ and the exact parameters of the measuring instrument $a \in A$, then we can uniquely predict the measurement result; this measurement result will be denoted by $r(x, a)$, and the set of all possible measurement results will be denoted by K. In mathematical terms, we have a *map* r from $X \times A$ to K. In the above example, K is the set $\mathbb{R}$ of all real numbers, and $r(\vec{B}, \vec{s}) = \vec{B} \cdot \vec{s}$.

In mathematical terms, a general data fusion situation can be thus described as a *triple* (X, r, A), where X and A are arbitrary sets, and r is a map $r : X \times A \to K$ into the set K. Such triples are called *K-Chu spaces*, or simply *Chu spaces* [1] (when the choice of K is clear). Chu spaces have been successfully used to describe parallelism [5], information flow in distributed systems [2], etc.

2.2 General Formulation of a Fundamental Problem of Data Fusion: Chu Automorphisms

In the above terms, the fundamental problem of data fusion can be reformulated as follows: in the ideal situation, when we know the results of all the measurements, can we uniquely reconstruct all the objects? In other words, if we know the values $r(x, a)$ for all $x \in X$ and all $a \in A$, will we be able to reconstruct all x, or it is possible to mis-interpret every object x as a different object $f(x)$, so that under a certain associated mis-interpretation $a \to h(a)$ of the measuring instruments, the results are still the same:

$$r(x, a) = r(f(x), h(a)) \qquad (1)$$

In other words, the unique reconstruction is possible if and only if there are no non-trivial pairs (f, h) with a property (1), and if there are such pairs, then we can only reconstruct x uniquely modulo transformations $x \to f(x)$.

For mathematical reasons, it is sometimes convenient to consider the *inverse* transformation $g(a) = h^{-1}(a)$. In terms of the inverse transformation, the condition (1) takes the form

$$r(x, g(a)) = r(f(x), a). \qquad (2)$$

A pair of functions which satisfies this property is called an *automorphism* of a Chu space (X, r, A). Thus, the data fusion problem has a unique solution if and only if the corresponding Chu space does not have any non-trivial automorphisms, and if its has, then we only have uniqueness modulo these automorphisms.

For example, for VLBI radioastrometry, there is no uniqueness, because we can apply a *rotation* $f(x)$ and a similar rotation $h(a)$, and the resulting scalar (dot) product will not change. One can prove, however, that this is the only possible non-uniqueness, i.e., that the only pair of transformations (f, h) which satisfies the property (1) is a pair of identical rotations. Thus, from VLBI measurements, we can reconstruct the locations of all radiosources modulo rotation: e.g., we can reconstruct the *arcs* between the sources.

From the physical viewpoint, the fact that we cannot uniquely reconstruct the coordinates of all the sources makes perfect sense: the axes of the coordinate system are determined only by convention, so this non-uniqueness simply means that we can select an arbitrary Cartesian coordinate system.

2.3 From Theoretical Analysis to Practical Data Fusion

We have just shown that Chu spaces allow us to answer a *theoretical* question about data fusion. Let us now show that we can also get a *practical* data fusion algorithm out of this analysis.

In our example, both sets X and A are represented as *manifolds*, i.e., each element $x \in X$ can be characterized by several numerical characteristics ("coordinates") $x_1, \ldots, x_n$, and each element $a \in A$ can be characterized by several numerical characteristics $a_1, \ldots, a_m$ (in this example, $n = 2$ and $m = 3$). In general, when X and A are manifolds, a uniqueness theoretical result leads to a *practical* algorithm. Namely, we know:

- the measurement results $r_{ij} = r(x^{(i)}, a^{(j)})$,
- the approximate values $\widetilde{x}^{(i)}$ of the parameters $x^{(i)}$ which characterize the objects, and
- the approximate values $\widetilde{a}^{(j)}$ of the parameters $a^{(j)}$ which characterize the measuring instruments.

To find the exact values $x^{(i)}$ and $a^{(j)}$ of these parameters, it is sufficient to find the differences $\Delta x^{(i)} = x^{(i)} - \widetilde{x}^{(i)}$ and $\Delta a^{(j)} = a^{(j)} - \widetilde{a}^{(j)}$. In terms of these unknown differences, we have $x^{(i)} = \widetilde{x}^{(i)} + \Delta x^{(i)}$ and $a^{(j)} = \widetilde{a}^{(j)} + \Delta a^{(j)}$, and the above expression for r_{ij} takes the form

$$r_{ij} = r(\widetilde{x}^{(i)} + \Delta x^{(i)}, \widetilde{a}^{(j)} + \Delta a^{(j)}). \quad (3)$$

The approximate values are usually reasonably good, so these differences are small, and we can therefore expand the right hand side of the equation (3) into Taylor series and ignore quadratic and higher order terms in this expansion. As a result, we get the following system of linear equations for determining the unknown differences:

$$\sum_{\alpha=1}^{n} A_{ij\alpha} \cdot \Delta x_\alpha^{(i)} + \sum_{\beta=1}^{m} B_{ij\beta} \cdot \Delta a_\beta^{(j)} = \Delta r_{ij}, \quad (4)$$

where:

$$A_{ij\alpha} = \left.\frac{\partial r(x^{(i)}, a^{(j)})}{\partial x_\alpha^{(i)}}\right|_{x^{(i)}=\widetilde{x}^{(i)}, a^{(j)}=\widetilde{a}^{(j)}},$$

$$B_{ij\beta} = \left.\frac{\partial r(x^{(i)}, a^{(j)})}{\partial a_\beta^{(j)}}\right|_{x^{(i)}=\widetilde{x}^{(i)}, a^{(j)}=\widetilde{a}^{(j)}},$$

$$\Delta r_{ij} = r_{ij} - r(\widetilde{x}^{(i)}, \widetilde{a}^{(j)}).$$

Solving a system of linear equations is easy.

For a detailed description of our example – and for a more realistic description of VLBI astrometry which takes into consideration the inaccuracy of the clocks – see, e.g., [3, 4].

3 Other Examples of Data Fusion

In the previous section, we showed that Chu spaces can be used to formalize a general class of data fusion problems. Data fusion is a very

general concept which includes situations more general than the ones described above. In this section, we enumerate such situations; in the following section, we will argue that (at least some of) these more general situations can also be naturally described in terms of Chu spaces.

3.1 Classical Statistics

In the above example, we assumed that the measurement result is uniquely determined if we know the object x and the measuring instrument a. In real life, there are a lot of random factors (noise), as a result of which, repeated measurements of the same object leads, as a rule, to slightly different results. So, instead of the *exact* value of $r(x,a)$, we have a *probability distribution* on the set of all measurement results. The measurement results may vary, but the *probability distribution* is uniquely determined by the measurement situation (i.e., by the pair of an object and of a measuring instrument).

Let X denote the set of all measurement results x, Θ be the set of all possible measurement situations θ, and let

$$\mathcal{F} = \{f(x,\theta) \,|\, x \in X, \theta \in \Theta\}$$

denote the class of all corresponding probability density functions $f(x,\theta)$. As a result of repeated measurements, we observe a random sample $x_1,\ldots,x_n$ from X. Based on this sample, we want to estimate either the value θ (i.e., the probability distribution itself), or some characteristic $\varphi(\theta)$ of this distribution (e.g., the standard deviation). Each of the measurement results x_i provides some estimate for $\varphi(\theta)$; to get a better estimate, we must "fuse" these estimates into a single estimate depending on all the measurement results $x_1,\ldots,x_n$. Usually, we seek some "good" estimator $T(x_1,\ldots,x_n)$, in fact, the *best* one, e.g., in the sense that it will maximize (or minimize) some performance characteristic (e.g., the expected squared deviation of our estimate from the true value of $\varphi(\theta)$).

The same is true in general: *we look for fusion operator which optimizes a given performance characteristic.*

3.2 Coalitional Games

Coalitional games, i.e., situations where several participants have different interests but are willing to cooperate, are non-measurement examples of data fusion.

Let us denote the set of players (participants) by Ω. In a coalitional game, every subset $A \subseteq \Omega$ can form a *coalition*, i.e., act together as a group against all the others. For each possible coalition (i.e., for each subset $A \subseteq \Omega$), we thus get a zero-sum (antagonistic) game, and we can use known techniques to determine the *payoff* $G(A)$ of this game. Thus, a coalitional game can be described as a *set-function* $G : 2^\Omega \to \mathbb{R}$. This function is *monotone* in the sense that increasing the coalition increases its payoff (if $A \subset B$, then $G(A) \le G(B)$). The main objective of coalitional game theory is to avoid the time-consuming coalition forming and dissolving process, and to come up with a solution which is fair to all the participants. In other words, we must "fuse" (combine) the payoffs $G(A)$ corresponding to different coalitions into a single solution.

As a desired performance characteristic, we can take, e.g., *fairness* (in situations describing distribution of goods), *productivity* (in situations describing the production of goods), etc.

In mathematics, the most well-known example of a function $2^\Omega \to R$ is *measure* – an additive function μ from the set 2^Ω of subsets of Ω to the set of real numbers $\mathbb{R}$. The most natural operation which maps a measure μ to a number is a (Lebesgue) *integral* $\int f\,d\mu$. Payoff functions are not necessarily additive, so, to describe the corresponding fusion, we can, e.g., use *Choquet integrals* – a generalization of Lebesgue integrals to monotone (not necessarily additive) set-functions. This indeed leads to reasonable solutions.

3.3 Expert Systems

A typical problem for which an expert system is useful is to predict, based on the known

symptoms, whether or not an individual with these symptoms has a certain disease. To solve this diagnostic problem, we solicit the knowledge of an expert. An expert usually formulates his or her knowledge in terms of different *rules*; these rules form what is called a *rule base*. For a given patient, different rules lead to different degree of confidence that the patient has (or does not have) the disease in question. The main goal of the expert system is to combine ("fuse") these (sometimes conflicting) degrees of confidence into a single result.

3.4 Probabilistic Inference

Similar to the previous example, we consider the problem of diagnosing a certain type of disease. Let $X = (X_t, t \in T)$ be the set of all variable which describe a patient: i.e., the variables which characterize the degree of the disease, the directly measurable variables (like body temperature, blood pressure, etc.), which are used in describing the symptoms, and the variables which are not directly measurable but which are used in the expert's arguments about the disease. Usually, the set T of these variables has some *neighborhood* structure in the sense that some pairs of variables t, t' are closely related ("close", t' belongs to the neighborhood N_t of t) while other pars are not directly related ("not close"). For example, we may be able to place all these variables on a plane so that "close" variables are the ones for which the distance is smaller than a certain threshold. The notion of a neighborhood structure is naturally formalized by a condition $P(X_t \,|\, X_s, s \neq t) = P(X_t \,|\, X_s, s \in N_t\}$ which describes a *Markov random field.*

From the experience, we can collect the conditional probabilities $P(X_t = x \,|\, X_s = y)$ which describe our degree of confidence in a rule "if $X_t = x$ then $X_s = y$". The main objective of data fusion is to combine these probabilities into a single symptom-determined probability of the given disease.

3.5 Randomness and Fuzziness

In the above fusion problems, all pieces of information had *the same* type of uncertainty. Here is a situation where *different* types of uncertainty can coexist in data.

In his pioneering work on random elements in metric spaces, Fréchet pointed out that besides standard random objects (such as points, vectors, functions), nature, science, and technology offer other random elements which, he claimed, "cannot be described mathematically". For example, for a randomly chosen group of people, we may be interested in their "morality" or "spirit"; for a randomly chosen town, its "beauty" of "shape" may be of interest, etc. Nowadays, these "fuzzy" concepts are described mathematically as *fuzzy sets.* Thus, examples of Fréchet are *random fuzzy sets.*

The existence of the two types uncertainty – randomness and fuzziness – requires new fusion procedures.

4 Chu Spaces and Morphisms As A Description of General Data Fusion Problems

4.1 Chu Morphisms

As we have already argued, each measurement procedure, each type of uncertainty, can be characterized by a Chu space. In some real-life situations, we must combine *different* types of uncertainty (e.g., random and fuzzy), so, we must consider relations between *different* Chu spaces.

It's possible to combine, e.g., probabilistic and fuzzy approaches: a fuzzy set can be described as a random set and thus, combined with probabilities. However, these combinations are complicated and hardly practical.

In general, for each type of uncertainty, we have a list of objects X and a list of properties A. Ideally, we would like to know exactly which object has which property; due to uncertainty, however, we only have the "degree" (probability, degree of certainty, etc.) $r(x,a)$ to which an object x has the property a. So, a general

piece of uncertain knowledge can be described as a K-Chu space (X, r, A), where K is the set of all possible degrees (usually, $K = [0, 1]$).

Often, to check whether an object has a certain property, we design a similar object (e.g., a scaled version), find its properties, and then make conclusions about the properties of the original system. In other words, we have a transformation $f : X \to Y$ which maps each object into a new one, and a transformation $g : B \to A$ which transforms the properties of the new object into properties of the old one in such a way that if the object $f(x)$ has a property b, then the original object x has the corresponding property $g(b)$, i.e., that

$$s(f(x), b) = r(x, g(b)). \tag{5}$$

A pair (f, g) is called a *morphism* between the Chu spaces (X, r, A) and (Y, s, B).

4.2 Categories of Chu Spaces

For every K-Chu space, a pair of identical maps is an (auto)morphism. If (f, g) is a morphism between K-Chu spaces (X, r, A) and (Y, s, B), and (Z, t, C) is another K-Chu space with (u, v) being a morphism from (Y, s, B) to it, then there is a morphism from (X, r, A) to (Z, t, C) given by

$$(f, g) * (u, v) = (u \circ f, g \circ v). \tag{6}$$

In the terminology of Category Theory, this means that K-Chu spaces and morphisms form a *category* in which a morphism composition is defined by the formula (6). This category will be denoted by $\mathcal{CHU}(K)$.

4.3 Fuzzy Sets as Chu Spaces

In fuzzy set theory, for a given set of objects X, properties are described as *fuzzy subsets*, i.e., $A = [0, 1]^X = \{a : X \to [0, 1]\}$, and the degree $r_X(x, a)$ to which an object x satisfies the property a is described as $r_X(x, a) = a(x)$.

Let us denote the $[0, 1]$-Chu category of the corresponding Chu spaces $F(X) = (X, r_X, [0, 1]^X)$ by $\mathcal{FUZZ}$. The morphisms of this category are easy to describe: if $f : X \to Y$ is a function from X to Y, then the pair $F(f) = (f, \varphi_f)$, where $\varphi_f : [0, 1]^Y \to [0, 1]^X$ is defined by a formula $(\varphi_f(b))(x) = b(f(x))$, is a morphism $F(f) : F(X) \to F(Y)$. By choosing an arbitrary function $f : X \to Y$, we can conclude that there exists a morphism between every two objects of the category $\mathcal{FUZZ}$.

It is easy to check that F preserves composition, i.e., $F(h \circ f) = F(h) * F(f)$, and therefore, that F is a *covariant functor* from the category $\mathcal{SET}$ of sets and functions to $\mathcal{FUZZ}$.

4.4 Chu Category of Conditional Probabilities

In a probabilistic approach to diagnosis, the basic pieces of information (which are combined in data fusion) consist of conditional probabilities $P(a|b)$ for different events a and b. So here, X and A are both sets of events, and $r(x, a) = P(x|a)$. Let us describe the corresponding Chu space in precise terms.

A *probability (measure) space* is usually defined as a triple $\Omega = (\Omega, P, \mathcal{A})$, where $\mathcal{A}$ is a σ-field over a set Ω, and $P : \mathcal{A} \to [0, 1]$ is a probability measure on $\mathcal{A}$. For each probability space Ω, we define the corresponding Chu space as a triple $P(\Omega) = (\mathcal{A}, r_P, \mathcal{A})$, where $r_P(a, b) = P(a|b) (= P(a \cap b)/P(b))$ if $P(b) > 0$, and $r_P(a, b) = 0$ if $P(b) = 0$ (i.e, if the above formula for conditional probability cannot be directly applied).

How can we describe morphisms between these Chu spaces? Let $\Omega = (\Omega, P, \mathcal{A})$ and $\Sigma = (\Sigma, Q, \mathcal{B})$ be probability spaces. A mapping $\varphi : \Sigma \to \Phi$ is called *measurability preserving* if it is one-to-one, $\varphi(\Omega) = \Sigma$, and both φ and φ^{-1} are measurable transformations. A measurability preserving transformation is called *measure preserving* if $P(\varphi^{-1}(b)) = Q(b)$ for every $b \in \mathcal{B}$, and *isomorphic* if both φ and φ^{-1} are measure preserving. We say that a pair (φ, ψ) of measurability preserving maps is *mutually measure preserving* if $P(a \cap \varphi^{-1}(b)) = Q(\psi^{-1}(a) \cap b)$ for all $a \in \mathcal{A}$ and $b \in \mathcal{B}$. One can prove that a composition of mutually measure preserving maps is measure preserving:

Proposition 1. *Let $\Omega = (\Omega, P, \mathcal{A})$, $\Sigma = (\Sigma, Q, \mathcal{B})$, and $\Gamma = (\Gamma, R, \mathcal{C})$ be probability spaces, and let $\varphi : \Omega \to \Sigma$, $\psi : \Sigma \to \Omega$, $\theta : \Sigma \to \Gamma$, and $\lambda : \Gamma \to \Sigma$ be measurability preserving maps. If (φ, ψ) and (θ, λ) are mutually measure preserving, then $(\theta\varphi, \psi\lambda)$ is also mutually measure preserving.*

One can prove that if a pair is mutually measure preserving, then the corresponding mapping are also measure preserving; thus, they preserve conditional probabilities and define a Chu morphism:

Proposition 2. *If (φ, ψ) is mutually measure preserving, then both φ and ψ are measure preserving.*

Proposition 3. *Let $\Omega = (\Omega, P, \mathcal{A})$, $\Sigma = (\Sigma, Q, \mathcal{B})$ be probability spaces, and let $\varphi : \Omega \to \Sigma$ and $\psi : \Sigma \to \Omega$ be measurability preserving maps. Then, the pair (φ, ψ) is mutually measure preserving if and only if the mapping $(\psi^{-1}, \varphi^{-1})$ is a Chu morphism $P(\Omega) \to P(\Sigma)$.*

An example of mutually measure preserving transformation is given by the following proposition:

Proposition 4. *If both φ and φ^{-1} are measure preserving, then the pair (φ, φ^{-1}) is mutually measure preserving.*

5 Cross Product Of Chu Spaces As A Data Fusion Operation

5.1 Motivating Example

In traditional probability theory, conditional probability $P(a|b)$ is defined for events a and b from the same σ-field of events. However, from the practical viewpoint, we start with *two* different sets of properties and, correspondingly, two different σ-fields: a σ-field $\mathcal{A}$ of events related to disease and a σ-field of events $\mathcal{B}$ related to symptoms; the only reasons why we have to combine these events is because otherwise, we will not be able to use the probability formalism.

How can we describe this "combination"? To even *describe* the conditional probability $P(a|b)$ of a given disease under given symptoms, we must represent the symptoms and diseases within the same probability space. We can achieve it in two ways:

- We can describe the symptoms in the disease space. For that, we need a transformation $g : \mathcal{B} \to \mathcal{A}$ which reformulates each disease-related property b into diseases-related terms: e.g., "sneezing" would translate into "cold or allergy". In this case, the desired conditional probability of a disease a under the symptoms b can be formalized as $P(a|g(b))$.
- We can also describe the diseases in terms of symptoms. For that, we need a transformation $f : \mathcal{A} \to \mathcal{B}$ which reformulates each symptom-related property a into symptom-related form. In this case, the desired conditional probability of a disease a under the symptoms b can be formalized as $P(f(a)|b)$.

The resulting conditional probability should not depend on how exactly we define it, and therefore, the corresponding two expressions must coincide:

$$P(a|g(b)) = P(f(a)|b). \tag{7}$$

5.2 Reformulation in Terms of Chu Spaces

Let us re-describe the above construction in terms of Chu spaces. If we take into consideration that for probability Chu spaces, $P(a|b) = r(a, b)$, then the formula (7) turns into the formula (2), which defines a Chu morphism.

Thus, in terms of Chu spaces, we have the following situation:

- Originally, we had two Chu spaces $P(\Omega = (\mathcal{A}, r_A, \mathcal{A})$ and $P(\Sigma) = (\mathcal{B}, r_B, \mathcal{B}))$, and a Chu morphism $(f, g) : P(\Omega) \to P(\Sigma)$.
- Based on this information, we design a *new* Chu space $(\mathcal{A}, r_{\text{new}}, \mathcal{B})$ for which $r_{new}(a, b) = r_A(a, g(b)) = r_B(f(a), b)$.

This constructed can be repeated for an arbitrary morphism between two Chu spaces:

- We start with two Chu spaces $\mathcal{X} = (X, r, A)$ and $\mathcal{Y} = (Y, s, B)$ and a Chu morphism $F = (f, g) : (X, r, A) \to (Y, s, B)$.
- Based on this information, we design a new Chu space (X, t, B), with $t(a, b) = r(a, g(b)) = s(f(a), b)$.

This new Chu space is called a *cross-product* of two original Chu spaces with respect to the morphism (f, g) and denoted by $\mathcal{X} \otimes_F \mathcal{Y}$.

5.3 One More Possible Application of Chu Cross Product to Data Fusion: Fuzzy Logic

In traditional fuzzy approach, fuzzy logic operations ("and", "or") are used to combine fuzzy data. This combination lacks the ability to describe relationship between the fused data. The notion of a Chu cross-product gives us a general way of describing such a relationship. So, we get the following new method of fusing two pieces of fuzzy data:

- first, we find the Chu morphism which best describes the relationship between these two pieces of data, and
- then, we combine these pieces relative to this morphism (by using a cross-product construction).

6 Conclusion

In general, different parts of information are expressed in different forms, such as probabilistic information, fuzzy information, etc. To combine ("fuse") this information, we must describe all types of uncertainty in terms of a single general formalism. In this paper, we have described a new general scheme for data fusion based on the notion of Chu spaces, and presented the corresponding results.

Acknowledgments

This work was supported in part by NASA under cooperative agreement NCC5-209, by NSF grants No. DUE-9750858 and CDA-9522207, by United Space Alliance, grant No. NAS 9-20000 (PWO C0C67713A6), by Future Aerospace Science and Technology Program (FAST) Center for Structural Integrity of Aerospace Systems, effort sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number F49620-95-1-0518, and by National Security Agency under Grant No. MDA904-98-1-0564.

References

[1] M. Barr, **-Autonomous Categories*, Springer Lecture Notes in Mathematics, Vol. 752, Springer-Verlag, Berlin, 1979.

[2] J. Barwise and J. Seligman, *Information flow: The logic of distributed systems*, Cambridge University Press, Cambridge, N.Y., 1997.

[3] A. F. Dravskikh, A. M. Finkelstein, and V. Kreinovich. "Astrometric and geodetic applications of VLBI "arc method"," *Modern Astrometry, Proceedings of the IAU Colloquium No. 48, Vienna, 1978*, pp. 143–153.

[4] A. F. Dravskikh, O. M. Kosheleva, V. Ya. Kreinovich, and A. M. Finkelstein, "The method of arcs and differential astrometry", *Soviet Astronomy Letters*, 5(3):160–162, 1979.

[5] V. R. Pratt, "Chu Spaces and their Interpretation as Concurrent Objects", In: *Computer Science Today: Recent Trends and Developments*, Springer Lecture Notes in Computer Science, 1995, Vol. 1000, pp. 392–405.

Diagnostic Information Fusion for Manufacturing Processes

Kai Goebel & Vivek Badami
GE Corporate R&D
Niskayuna, NY 12309
goebelk@crd.ge.com; badami@crd.ge.com

Amitha Perera
Rensselear Polytechnic Institute
Troy, NY 12380
perera@cs.rpi.edu

Abstract - This paper addresses diagnostic information fusion for situations where several diagnostic tools are used to estimate a single system state. These estimates will always disagree to some extent and it is the task of the fusion module to provide an estimate which is more reliable than the best of the diagnostic tools. To that end, a fusion process was developed which performs a weighted average of individual tools using confidence values assigned dynamically to the individual diagnostic tools. These confidence values are derived from validation curves which are designed using individual a priori tool information and which are centered about the previous system estimate. In a further step, the fusion output is smoothed leading to additional performance improvement. In experiments, data were gathered from a high speed milling machine and fed through several developed diagnostic tools.

Key words: fusion, information fusion, diagnosis, soft computing, fuzzy fusion.

Introduction and Background

The need of manufacturers to produce inexpensive quality products has resulted in increasing demand for unattended and/or automated manufacturing systems. One problem in automating machining is how to deal with common malfunctions and disturbances such as tool wear, chatter, and tool breakage. Tool wear is a highly non-linear process which is hard to monitor and estimate. To avoid costly damage due to tool wear or breakage, manufacturers use conservative operating procedures to prevent these malfunctions [1]. However, these result in less efficient and more costly production. A number of diagnostic techniques attempt to deal with theses problems, including neural networks [2], clustering algorithms Burke [3], Kohonen's Feature Map [4], fuzzy logic [5], and influence diagrams [6]. To achieve further performance improvement, hybrid systems were proposed to overcome shortcomings of individual systems, such as fuzzy-neural systems [7]. Hybrid use of above mentioned techniques and other soft computing principles for diagnostics and prognostics are given in Bonissone and Goebel [8]. In a similar spirit, fusion techniques combine different methods to overcome shortcomings of individual tools. This paper proposes one fusion method based on fuzzy validation gates.

Diagnostic Fusion via Validation Gates

The method developed is a two-level system consisting of a number of diagnostic classification systems on the first level and a managerial fusion unit on the second. The data are fed into each of the first level units, and their output is combined in the second level to produce a single, better solution (Fig. 1).

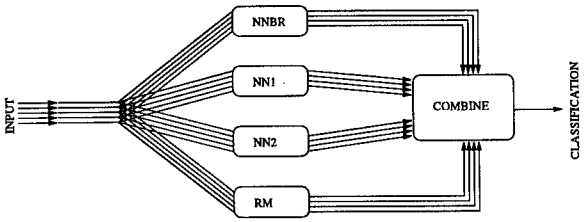


Fig. 1: The system architecture

To address some of the problems outlined above, we propose the fusion of diagnostic estimates via fuzzy validation curves called Fuzzy Diagnostic Validation and Fusion (FUDVAF). This technique is related to the FUSVAF (Fuzzy Sensor Validation and Fusion) algorithm developed for sensor fusion [10, 11, 12]. The fusion algorithm uses confidence values obtained for each diagnostic output from validation curves and performs a weighted average fusion. With increasing distance from the predicted value, readings are discounted through a non-linear validation function. The predicted value in the FUDVAF algorithm is obtained through application

of an exponential weighted moving average time series predictor

The confidence value which is assigned to a particular diagnostic output depends on the specific tool characteristics, the predicted value, and the physical limitations of the diagnostic value. The assignment takes place in a validation region which assigns a maximum value to readings which coincide with the predicted value. The curve is dependent on the sensor behavior. Generally, this is a non-symmetric curve which is wider around the maximum value if the diagnostic tool is more reliable and narrower if it is less reliable.

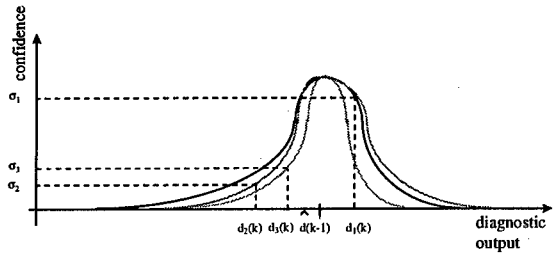
A choice for validation curves $\sigma(z)$ could be a bell curve of the form

$$\sigma(d_t) = 1 - e^{-\left(\frac{(d_t - \hat{d})^m}{a_t}\right)^2}$$

where

m is a scaling parameter
a_i is the tool accuracy
d_i is the diagnosis of tool i
$\hat{d}$ is the estimated diagnosis

A validation gate is shown in Fig. 2.



d_t diagnostic output
σ_i confidence values
$\hat{d}(k-1)$ fused value

Fig. 2: Validation gate for the assignment of confidence values

The fusion is performed through a weighted average of confidence values and diagnostic output. The sum of the confidence values times the measurements rewards measurements closest to the old fused value the most, depending on the validation curve which expresses a trust in the operation of each diagnostic tool through the design of its shape. Measurements further away are discounted. The operative equation in the FUDVAF algorithm is

$$d_f = \frac{\sum_{t=1}^{n} d_t \sigma(d_t)}{\sum_{t=1}^{n} \sigma(d_t)}$$

where

d_f: fused value
d_t: diagnostic output of tool t
σ: confidence values

Note that if all diagnostic outputs lie on one side of the predicted value, the fused value will also be pulled to the same side. This ensures that evidence from the diagnostic tools is closely followed yet discounted the further it gets away from the predicted value.

We used a time series filter to further improve the result of the system using the standard EWMA predictor of the form

$$\hat{d}(k) = \hat{d}(k-1)\alpha + \frac{\sum_t \sigma_t d_t}{\sum_t \sigma_t}(1-\alpha)$$

where

α is the smoothing parameter; α=0.1

Experimental Setup

A milling machine under various operating conditions was selected as the manufacturing environment. In particular, tool wear was investigated in a regular cut as well as entry cut and exit cut. Data sampled by three different types of sensors (acoustic emission sensor, vibration sensor, motor current sensor) were used to determine the state of wear of the tool. As the wear measure, flank wear VB (the distance from the cutting edge to the end of the abrasive wear on the flank face of the tool) was chosen. The flank wear was observed during the experiments by taking the insert out of the tool and physically measuring the wear. The setup of the experiment is as depicted in Fig. 3. The basic setup encompasses the spindle and the table of the Matsuura machining center MC-510V. An acoustic emission sensor and a vibration sensor are each mounted to the table and the spindle of the machining center. The signals from all sensors are amplified and filtered, then fed through two RMS before they enter the computer for data acquisition. The signal from a spindle motor current sensor is fed into the computer without further processing. Data are categorized into four classes and are approximated by fuzzy membership functions (no

wear, little wear, medium wear, high wear) shown in Fig. 6.

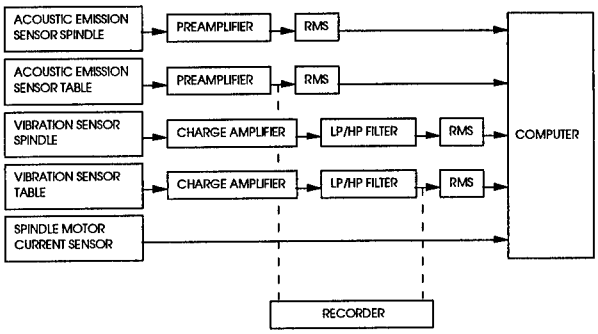


Fig. 3: Experimental setup

Input data transformations

Before being used, the following transformations were applied to the data:

1.) The data was smoothed by averaging using a window of 50 points, and then the sample size was reduced by sampling the resulting data set at 50 point intervals. 2.) Each input and the output data was normalized to lie between 0 and 1. 3.) Since the output variable was sampled at much larger intervals than the input variables, and since it represents tool wear, the output data was further smoothed by fitting a 3rd order polynomial.

Fig. 4 and Fig. 5 show the normalized and smoothed input and output data. The output data was categorized into four classes using fuzzy membership functions.

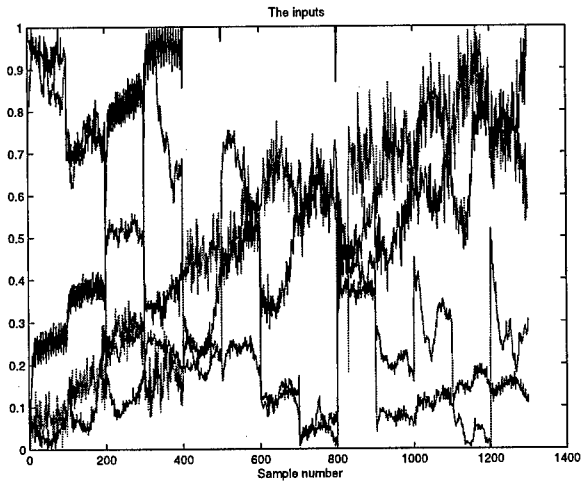


Fig. 4: Input data

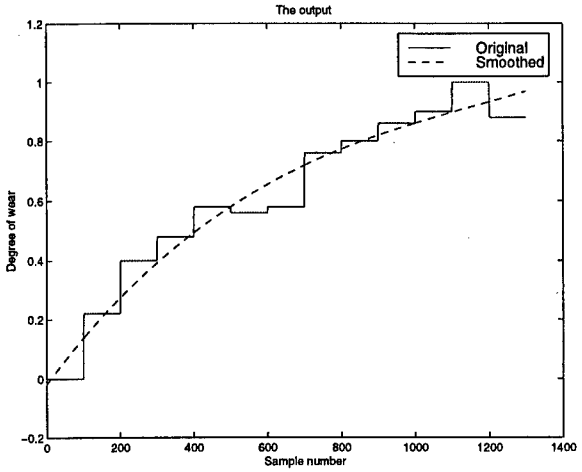


Fig. 5: Ouput data

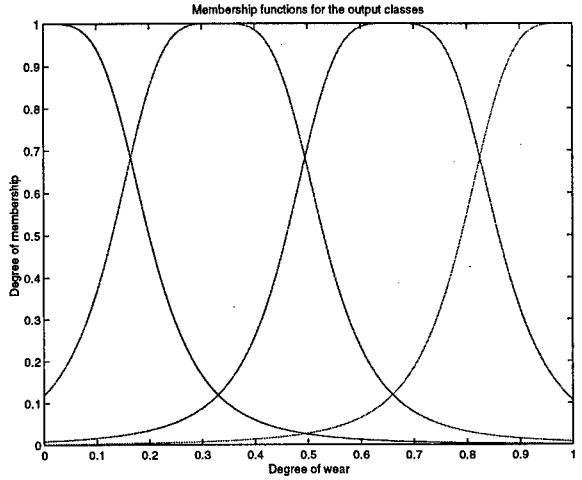


Fig. 6: Membership functions

Diagnostic tools employed

Nearest neighbor classifier (NNBR): The first subsystem uses a nearest neighbor scheme for classifying the data. The case base consists of a set of sensor readings and the associated unclassified wear value. Given an input, the k nearest data points are determined and the associated wear values are averaged. This average is then used to compute the membership degree for each of the four classes.

Neural network (NN1): The second subsystem is a neural network that was trained on binary classes. That is, the target values were 0 and 1 vectors determined by the maximum membership value over the four classes.

Neural network (NN2): The third subsystem is also a neural network, but this was trained to learn the membership values themselves, as opposed to the classes.

Fuzzy inference system (RM): The fourth subsystem is a fuzzy inference system implemented using a relation matrix.

Architecture

As shown in Fig. 1, the input to the first level of the system are the measured features. The output consists of four values indicating the degree of membership for each of the four output classes. We chose this approach over an approach where the first level subsystem generates a crisp class (from 1 to 4) because this approach gives more flexibility and information to the second level system. This is in response to the need recognized after development of the neural-fuzzy diagnostic tool [9] which attempted to segment the data into five crisp classes. In the approach chosen here, the membership approach allows a softer classification. The second level system then combines the results of the first level systems and classifies the input into a single class. We will be focusing in this paper on the fusion task and evaluate performance based on the fused membership values.

One basic problem in averaging techniques or majority voting techniques is the danger of ending up with a system which performed worth than the best individual tool because the poor estimates drag down the better estimate. One potential solution is to weigh the tools according to their performance which must be known beforehand. The FUDVAF tries to perform this task. Stand alone tests were performed to establish the accuracy of the individual diagnostic tools which are shown in Table 1.

Table 1: Classification rates

System Rate	(%)
Nearest neighbor (NNBR)	96.8
Neural network 1 (NN1)	80.6
Neural network 2 (NN2)	86.7
Relational matrix (RM)	81.0

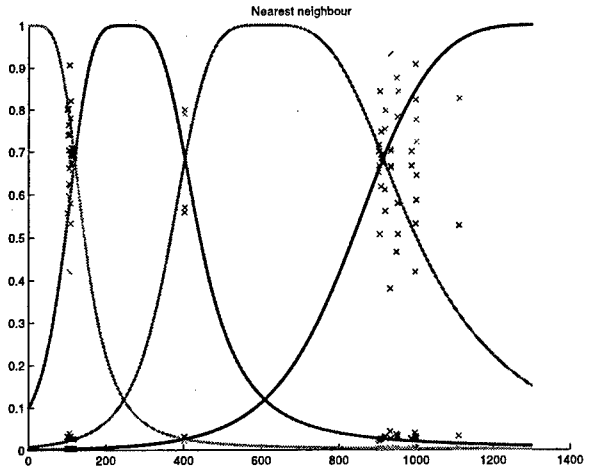


Fig. 7: Output NNBR

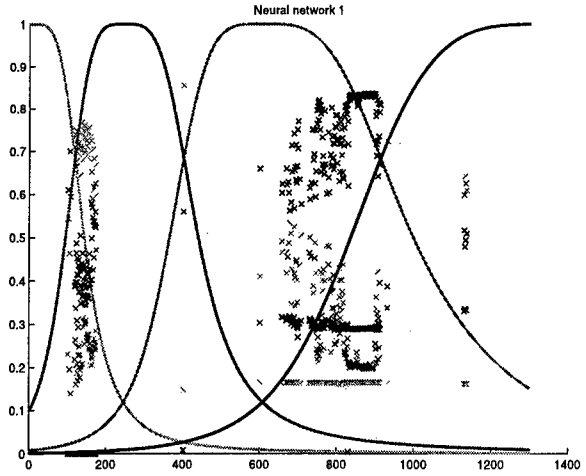


Fig. 8: Output NN1

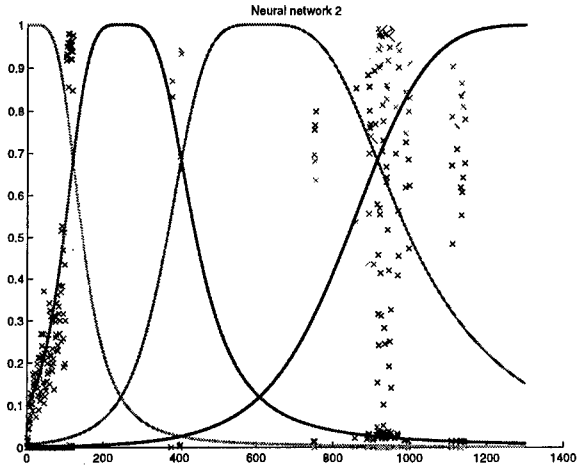


Fig. 9: Output NN2

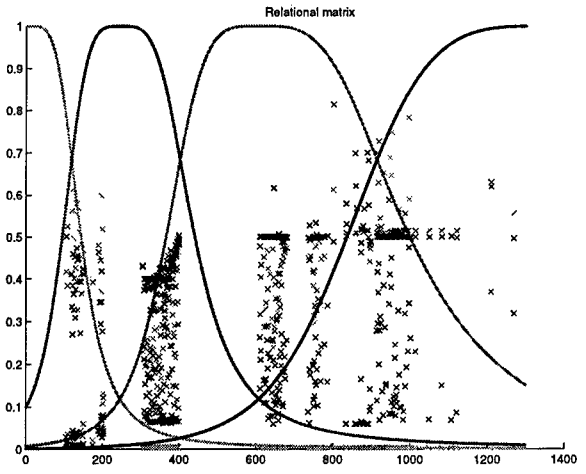


Fig. 10: Output RBS

Fig. 7 to Fig. 10 depict the performance of the individual systems. The solid lines show the true membership values for the data. The crosses indicate the membership values generated by the system

when the system disagrees on the classification. Thus, the crosses are an indication of the area(s) in which the system has difficulty in deciding on a class. Fig. 7 shows that NNBR has classification errors only near the cross-over points of the membership functions. These are areas where classification errors are expected, because a small change in the membership value results in an incorrect classification. Even at these points, however, NNBR membership values are very close to the true membership values. The success of this system is due to the continuous nature of the wear measure, and the averaging technique used in the nearest neighbor classification. The other systems are not as successful, and the membership values output do not approximate the true values to the degree that NNBR does.

The high success rate of one tool means that if it were used as part of the majority fusion, the performance degrades somewhat. This is to be expected, because the votes of the poor performers will sometimes out vote the correct one. The problem is greater with increasing number of classification regions, as there will be cases when each system will generate a different class, and the majority voting system will then pick one randomly.

Results

The fusion using assignment of confidence values provides a means to integrate a priori information about individual tool performance. This is accomplished by designing the validation curves of a better tool wider than the curves of the tools with worse performance. The fused performance improves the already very good performance of the NNBR tool from 96.8% to 99.1% correct classification with α=0.1 and m=0.1. Fig. 11 shows the result of the fused system where the membership functions no wear, little wear, medium wear, and high wear were estimated.

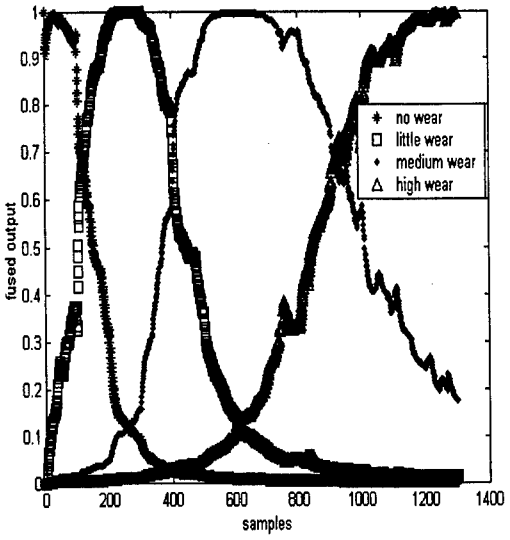


Fig. 11: Fused system output

Summary and Final Remarks

The use of the FUDVAF algorithm provides a means to improve performance of individual diagnostic tools. In experiments with data from a milling machine, we show how the FUDVAF can be used for extant systems. Much of performance improvement appears to be due to the smoothing and an increase of performance might also be expected when the smoothing is applied to the best tool alone.

Future research should address how to improve fine tuning of the validation curve parameters, depending on operating conditions and sensor history. This can be accomplished through machine learning techniques similar to the approach used for the FUSVAF algorithm [11]. Also helpful might be knowledge about locally changing diagnostic tool performance. Such local characteristics could be utilized in designing the validation curves in a dynamic manner by changing the width accordingly. Generally, it is desirable to maintain maximum independence of the diagnostic tools in the sense that tools which exhibit poor performance in certain operating conditions are matched with tools exhibiting better conditions there. This may, of course, not always be possible due to the limitations of observable conditions and shortcomings of the diagnostic tools because often times (and in this approach here), all sensor values are made available to all diagnostic tools.

Acknowledgements
The authors thank Prof. A. Agogino for her support and advice; Prof. D. Dornfeld and Prof. Tomizuka (all of UC Berkeley) for making available equipment; and Andrzej Sokolowski and Piotr Dworak for helping with the experiments.

References

[1] Rangwala, S.S., and D.Dornfeld "Learning and Optimization of Machining Operations Using Computing Abilities of Neural Networks", *IEEE Transactions of Systems, Man & Cybernetics*, Vol. 19, No. 2, March/April 1989.

[2] Rangwala, S.S. "Machining Process Characterization and Intelligent Tool Condition Monitoring Using Acoustic Emission Signal Emission Signal Analysis", PhD thesis, UC Berkeley, 1988.

[3] Burke, L. I. "Automated Identification of Tool Wear States in Machining Processes: An Application of Self Organizing Neural Networks," PhD thesis, UC Berkeley, 1989.

[4] Leem, C.S. "Input Feature Scaling Algorithm for Competitive Learning Based Cognitive Modeling with Two Applications", PhD thesis, UC Berkeley, 1992.

[5] Fei, J., and I.S. Jawahir "A Fuzzy Classification Technique for Predictive Assessment of Chip Breakability in Intelligent Machining Systems", *Proc. IEEE Joint Conference on Fuzzy Logic and Neural Networks*, San Francisco, 1993.

[6] Agogino, A. M., and K. Ramamurthi "Real Time Influence Diagrams for Monitoring and Controlling Mechanical Systems," *Influence Diagrams, Belief Nets and Decision Analysis*, Ed.: Oliver, R. M., and J.O. Smith. J.R. Wiley & Sons, Ltd., 1990.

[7] Goebel, K., and P.K. Wright "Monitoring and Diagnosing Manufacturing Processes Using a Hybrid Architecture with Neural Networks and Fuzzy Logic", EUFIT, Proceedings, Vol. 2, Aachen, Germany, 1993.

[8] Bonissone, P., and Goebel, K., "Soft Comuting Principles for Diagnostics", working notes of the AAAI Spring symposium, Palo Alto, 1999.

[9] Goebel, K., Wood, W., Agogino, A., and Jain, P., "Comparing a Neural-Fuzzy Scheme and a Probabilistic Neural Network for Monitoring of Manufacturing Processes", Working Notes of the AAAI Spring Symposium, 1994.

[10] Agogino, A., Alag, S., and Goebel, K., Proceedings of the 1995 Annual Meeting of ITS America A Framework for Intelligent Sensor Validation, Sensor Fusion, and Supervisory Control of Automated Vehicles in IVHS, 1995.

[11] Goebel, K., Management of Uncertainty for Sensor Validation, Sensor Fusion, and Diagnosis Using Soft Computing Techniques Ph.D. Thesis, University of California at Berkeley, 1996.

[12] Goebel, K., and Agogino, A.M., Proceedings of the 29th ISATA An Architecture for Fuzzy Sensor Validation and Fusion for Vehicle Following in Automated Highways, 1996.

ACTIVE FUSION FOR DIAGNOSIS GUIDED BY MUTUAL INFORMATION MEASURES[1]

John M. Agosta (johnmark@kic.com) and Jonathan J. Weiss (jjweiss@kic.com)
Knowledge Industries
334 Arbor Drive
South San Francisco, CA 94080

Abstract

In diagnostic probability models where typically there are dependencies among input variables, best selection of inputs depends on previous inputs. *Active fusion* is an iterative process of selecting the next set of inputs to acquire based on their potential to distinguish among the possible diagnoses.

While decision-theoretic *value of information* is the ideal measure for test value, we use a *mutual information* approximation that uses less demanding computations and knowledge models. The algorithms presented here are fast enough to use interactively on a personal computer or in a web-based application.

Active fusion is especially valuable in the initial stages of diagnosis, when the choice typically includes several non-specific observations. The method is being applied to diagnosis of faults in vehicle subsystems such as aircraft, locomotives and automotive vehicles.

Key Words: Mutual Information, Value of Information, Bayes Networks, Active Fusion, Diagnosis

1 Introduction

1.1 Active Fusion

Diagnosis is the process of reasoning based on information to sharpen the state of belief about possible *faults* (abnormal situations requiring adjustment, repair, or replacement). Typically there isn't sufficient information to produce absolute certainty, so the state of belief is expressed in terms of fault probabilities, and the diagnostic value of information is defined in terms of the potential to sharpen those probabilities from an initial state of high entropy (several possible disorders, none with conclusive support). [3,4,6]

Most diagnostic processes involve a large number of potential inputs: symptom reports, measurements, or tests to be performed. Some of these require human intervention, while others entail costs, take time, use bandwidth, or require materials, equipment, skilled labor or facilities. The common thread is that it is impractical or uneconomical to seek all inputs for every problem. Therefore, there must be a tradeoff between "value" and "cost". (For purposes of this paper, we limit the meaning of "value" to diagnostic value, although in practice some diagnostic procedures may also have curative or preventive value. We use the term "cost" to include any limited resource; typically one type will dominate, and even if there are multiple cost factors, they can be additively combined into a single cost-equivalent when weighted appropriately.)

Because the diagnostic value for most of these inputs depends heavily on the current state of belief about fault probabilities, the diagnostic process is typically sequential. [11] Some inputs are collected, and based on their results the state of belief is revised. If enough doubt remains, another set of inputs can be sought. The process of selecting a set of inputs dynamically based on the current information is called *active fusion*.

1.2 The Bayes Network representation

The Bayesian approach to diagnosis starts with a causal model. Typically elicited from experts (engineers, diagnosticians, service technicians, etc.), information is collected about the probabilities of various observations (symptoms, test results, etc.) given the variables that can cause these observations (i.e., the faults that are present) under some preconditions (e.g., which model of a machine is being evaluated). Generally, the relation between faults and observations is many-to-many: one fault will have many observable effects, and one effect can have many possible causes. There may also be causal relations among faults, and among observations. The consequences of the latter will be considered in this paper. This complexity is completely represented the joint probability distribution of faults and observations. Bayes Networks [6,13] provide a consistent, concise and computationally manageable representation for such causal diagnostic models.

A Bayes network is a representation consisting of nodes connected by arrows. Each node represents an uncertain

[1] Work reported in this paper was partially funded by DARPA contract # DAAH01-95-C-R176. The authors gratefully thank AIS, Inc. for the opportunity and automotive expertise to construct cited application models.

proposition or variable, either an observation, a fault, or an intermediate unobservable condition. The information known about one node depends upon the information in its predecessor nodes that represent its causes. In other words, if one node variable is the cause of another, then knowing the one would give us more information about the state of the other. We say that the one *probabilistically conditions* the other. This is expressed in the contents of the node by a probability distribution of the node variable conditioned on its predecessor variables.

Formally the network is equivalent to a factoring of the joint probability distribution of all its variables. This network structure is both a concise visual representation and a formal specification by which the diagnosis is made. The structure of the model as represented by the network is intuitive to the experts and decision makers so that it is easy to construct the network, and easy to explain the results of a diagnosis when observations are made. Just as important, a Bayes network is a precise formal specification of the problem that ensures consistency in probabilities with which it is formulated, in a form that it can be solved by exact, fixed time methods. In practice a diagnostic network may have hundreds of nodes, representing the combination of possible state distinctions that is exponential in the number of nodes. It is a concise representation of an inconceivably large fault tree.

There are several known exact solution methods for updating the probability distribution of fault variables as observation variables take on values. [12,15,16] As a by-product of the solution, one obtains value of information measures and their approximations such as mutual information. Exact solution methods are NP-hard in the number of nodes. This presents a practical limitation in model size when models are dense with conditioning arrows, however, as our example shows, this limitation does not prevent the practical application of Bayes networks to large-scale models.

The computational methods for Bayes networks have the ability to invert the direction of reasoning in the causal model. The model is constructed by reasoning causally from faults to observations, whereas the diagnostic process reasons from observations to faults. Just as Bayes rule inverts the conditioning between two variables, a Bayes network can be solved for the conditioning of any disjoint subset of variables on another. While such computations would be extremely tedious if done manually, Bayes Network software such as Knowledge Industries' *DX Solution Series* can quickly and efficiently compute posterior probabilities of all faults in the model given any combination of possible observations as inputs. These posterior probabilities summarize the state of belief given all the observation information entered.

1.3 The problem of test selection

The hardest part of diagnosis is not reasoning about faults from the information given; it is trying to select the next set of inputs (observations) in a way that will arrive at the correct diagnosis efficiently. This may mean selecting an inexpensive test over a more precise but more expensive test. It may mean avoiding tests whose outcome is known with near certainty (unless the exceptional outcome would have overwhelming diagnostic value). It may mean choosing one of two highly redundant tests rather than selecting both. It may mean performing an inexpensive preliminary test whose results will tell whether it is worthwhile to perform another more expensive test.

The key to each step is determining what would be the best single input, the best pair of inputs, the best set of 3 inputs, etc. and then deciding which of these "best n input" sets to choose given the constraints and tradeoffs that apply.

The problem of test selection is similar to the problem of feature subset selection when constructing a classifier. The difference is that feature subset selection typically is done once during construction and prior to the use of the classifier, whereas test selection is dynamic, and done during the diagnostic process. Here is an example of the feature subset selection problem. This example comes from Ripley [14]:

> To illustrate the difficulty, consider a battery of diagnostic tests $T_1 \ldots T_m$ for a fairly rare disease, which perhaps around 5% of all patients tested actually have. Suppose test T_1 correctly picks up 99% of the real cases and has a very low false positive rate. However, there is a rare special form of the disease that T_1 cannot detect, but T_2 can, yet T_2 is inaccurate on the normal disease form. If we test the diagnostic tests one at a time, we will never even think of including T_2, yet T_1 and T_2 together may give a nearly perfect classifier by declaring a patient diseased if T_1 is positive or T_1 is negative and T_2 is positive. This illustrates that considering features one at a time may not be sufficient. (p.327)

2 Decision theory formulation

2.1 Definition of value of information (VOI)

Value of information analysis is a central part of diagnosis, since it determines which test or observation to pursue. [2,8,9,10] This section explains value of information and its approximation as mutual information.

The general question is to compare the improvement in the consequences of the subsequent decision based on the information generated or received. The degree of improvement becomes a measure of the quality of the information.

The value of information measure is a by-product of an expected value optimization problem; it requires no additional information. Put in the context of diagnosis, the problem becomes the maximization of a value $v(d,f)$ that is a function of the repair actions taken d and the fault state f of the device. The fault state is uncertain, described by its probability distribution $p\{F_k\}$. The optimization problem before acquiring test data is

$$v^* = \max_d E_f[v(d,f)], \qquad (1)$$

where the expectation over f is written as $E_f[\ ]$.

The Bayes network diagnostic model specifies the probability distribution over faults and observations, by which information from test observation is introduced into the model. The tests or observations, Q_j, can be known directly and reveal partial information about the faults. By solving the model given the test values, we can obtain $p\{F_k|Q_j\}$, the distribution over fault states which are revealed by the test values in addition to $p\{Q_j\}$, the marginals on the tests. These are the distributions necessary to calculate VOI.

When posed as an expected-value decision problem, the formulation of VOI consists of a sequence of at least two decisions, together with the observation, fault and outcome variables. In the following equation, two distributions summarize the entire diagnostic model, $p\{Q_j\}$ representing the set of observation variables and the $p\{F_k|Q_j\}$ representing the set of all fault variables. Looking at the decisions as time-ordered, the first decision t is called the *test*; the decision d that follows it is the *repair*. The repair affects the outcomes and hence the value. The value is also a function of the fault state. The test affects solely the information available when the decision to repair is made.

$$v^*(t) = E_q[\max_d E_f[v(d,f) \mid q,t] \mid t] \qquad (2)$$

Equation 2 gives the value as a function of the test variable, or the *value with information.* For VOI we need the difference between this and the value without information:

$$VOI(t) = v^*(t) - v^* \qquad (3)$$

Test selection is done by selecting t that maximizes $VOI(t)$. In greedy test selection, the computation of VOI over the remaining set of tests is repeated after obtaining each test result. As long as there remains a test whose outcome would change the decision made, *VOI* will be positive. Positivity of *VOI* offers a valid stopping rule for testing. This rule extends naturally when there is a fixed cost, hence a net value, for each test. In cases where test resources or time for diagnosis are constrained, [7] the optimization problem can be extended to a constrained optimization problem by optimization under a cost constraint.

VOI can be burdensome to model since it requires a value function for combinations of repair and fault outcomes in addition to the complete Bayes network model. A full value model would range over the probable actions to be taken based on a diagnosis, and the both the costs and benefit consequences of those actions.

2.2 Entropy as a surrogate for VOI

In cases where a value function is not available VOI can be approximated qualitatively by entropy-based methods. [1,17] Entropy-based methods do not require a value model because in effect they assume that the "regret values" in the value functions are the same for any wrong diagnosis (false alarm or missed fault). Entropy is a function of a probability distribution that corresponds roughly to the non-specificity of the distribution. To assign an entropy-based value to information we will derive a measure that ranks observations by their ability to decrease the entropy of the fault probability distribution. The effect of this decrease in entropy will be to drive the probabilities of faults toward extreme values.

To measure the impact that knowing an observation variable will have on the probability distribution of a fault, we need an entropy measure of the correspondence of between the fault and observation random variables. The joint entropy of two random variables can be partitioned into a part that "overlaps," and the part that does not. The overlap is the pair's *mutual information.* For independent random variables it is zero. For identical random variables, it is equal to either random variable's entropy. These properties make mutual information appropriate to rank observations by their ability to confirm or refute a fault. The derivation of mutual information parallels that of *VOI.* The entropy of F is given by:

$$H(F) = -E_f[\ln P\{F_k\}] \qquad (4)$$

The entropy of F conditioned on an additional observation Q, called the *conditional entropy* [1] is given by:

$$H(F|Q) = -E_{f,q}\left[\ln P\{F_k|Q_j\}\right]. \quad (5)$$

The difference between the prior entropy of the fault $H(F)$ and the entropy conditional on observation Q, gives the *mutual information* for F and Q:

$$I(F;Q) = H(F) - H(F|Q) \quad (6)$$

Equation (6) mimics Equation (3). They are both the difference of two terms, one an expectation over F, the other an expectation over F and Q. They differ in that mutual information uses the logarithm of a probability in place of a value. If we assign a "regret value" of $-ln(p\{F|Q\})$ to a missed fault and $-ln(1-p\{F|Q\})$ to a false alarm, this formula corresponds to an expected value of information. It effectively penalizes any incorrect belief in proportion to the logarithm of the probability assigned to that false belief. In mutual information, the log function provides the convexity that the maximization operator provides in *VOI.*

Unlike *VOI, I(F;Q)* is almost always positive and does not offer a natural stopping rule. To put a limit on testing either the urgency of the fault, or a constraint on test costs can be included in the model.

2.3 VOI, mutual information and test relevance

There are two independence criteria that are necessary for any test value $value(Q_j)$, both of which are satisfied by both VOI and mutual information. The first is that the value is non-negative, and if the observation is independent of the fault, then the measure is zero. This is necessary to eliminate tests that are irrelevant.

Assumption 1: $p\{Q \mid F\} = p\{Q\}$ implies that $value(Q_j) = 0$.

Furthermore, we want the dependence among value measures to mimic the conditional dependence among observations. This is a necessary condition to be able to identify conditionally independent tests that will not present problems with greedy selection, and so do not have to be considered in non-myopic algorithms.

Assumption 2: If $p\{Q \mid Faults\} = p\{Q \mid other\ observations, Faults\}$ then $value(Q_j) = value(Q_j|other\ observations)$. In other words, the conditional independence relations among tests are respected by the test value measures.

It is also desirable that the test measure not depend strongly on the current beliefs about the faults, but rather on the qualities of the tests and the test dependencies. This is desirable because the test values and thus the test rankings will be stable as the fault probabilities change due to previous observations. This condition is not necessary, and both VOI and mutual information have a weak dependence on fault posteriors. A complete axiomatic specification of the desirable properties of test relevance remains an open question. The question of axiomatizing relevance has been addressed in the machine learning literature. See [18].

If we assume that all observations are conditionally independent, then the optimal test ordering is just the greedy selection of tests in decreasing order by their value/cost ratios as initially computed. If all tests are assigned equal costs, this reduces to a ranking by value. In practice, conditional independence of tests is an unrealistically strong assumption, since it ignores dependencies that are captured in the Bayes network diagnostic model. The next section of this paper shows what can be done to account for dependencies among observations contained in the Bayes network.

3 Greedy active fusion and its extension

3.1 Example of greedy failure

The dependencies that confuse greedy test selection occur for both VOI measures and its entropy-based approximations. They are a property solely of the probabilistic dependency structure of the model and occur irrespective of any cost constraints on tests. In the first example the failure occurs when two tests are meaningless by themselves, but form a powerful test when used in combination. The observations are valuable only as a pair, thus in greedy test sequences the tests will be passed over, leading to test orderings that are not optimal.

Figure 1 is an example of a diagnostic network with two dependent observation variables. In this example the conditional distribution of fever is uniform

P{ Fever | Disorder }	Fever = absent	Disorder = present
Disorder = absent	**0.5**	**0.5**
Disorder = present	**0.5**	**0.5**

and the conditional distribution of headache has this structure:

P{ Headache | Disorder, Fever }	Headache = absent	Headache = present
Disorder = absent, Fever = absent	**0.3**	**0.7**
Disorder = absent, Fever = present	**0.8**	**0.2**

Disorder = present, Fever = absent	**0.7**	**0.3**
Disorder = present, Fever = present	**0.2**	**0.8**

The result is that the test values of *headache* or *fever* individually are zero. If *headache = true*, the probability of disorder conditioned on this observation's value does not change. Similarly for *headache = false*, or for either value of *fever*. If it is the case that *redness* or *irritation* are weaker effects of the disorder than the combination of fever and headache, then ordering the test sequence test value should place the combination of fever and headache first.

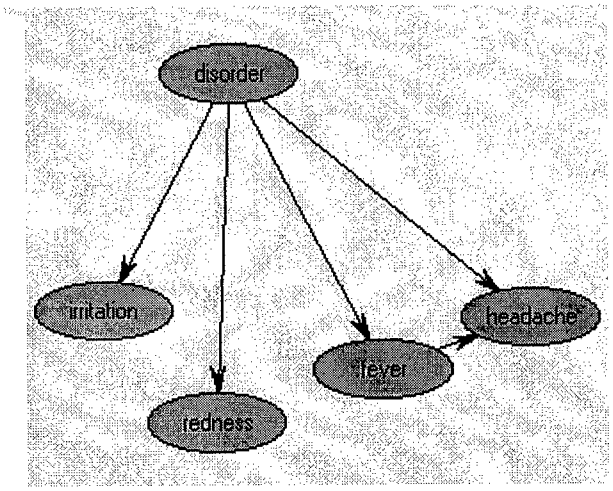


Figure 1: A diagnostic network with one fault node, "disorder," and four observation nodes. The observation node "irritation" and "redness" are conditionally independent given the fault node. The nodes "fever" and "headache" are not, because of the conditioning of "headache" by "fever."

Dependencies among observation variables can also occur for tests that nullify one another. Perhaps the disassembly necessary for one test precludes the measurements called for by another. In that case either test may have nominal value, but taking one test reduces the value of the second. Instead of looking for tests to combine, such conditions will disqualify one of the two tests, so that one will be used in lieu of the other.

Typical of multiple-fault models, the dependency between observations will be indirect, by a path through another fault. In Figure 2 *irritation* and *redness* have a second-order conditional dependence, itself conditional on the posterior of *disorder2*. This is due to the conditional probability tables for *irritation* and *redness*, which have the same form as the table for headache. The priors on *disorder* and *disorder2* are uniform, and the conditional probability for context is any such that the test value with respect to *disorder2* is positive.

Irritation and *redness* exhibit a second order conditional independence from the point of view of diagnosing *disorder*. The conditional dependence shows up only when the posterior on *disorder2* is perturbed, in this case by observing *context*. Thus the value of both *irritation* and *redness* is zero, regardless of whether the other is observed, unless the variable *context* is observed. Characteristic of this model fragment is that the test value of all three variables with respect to *disorder* is zero initially.

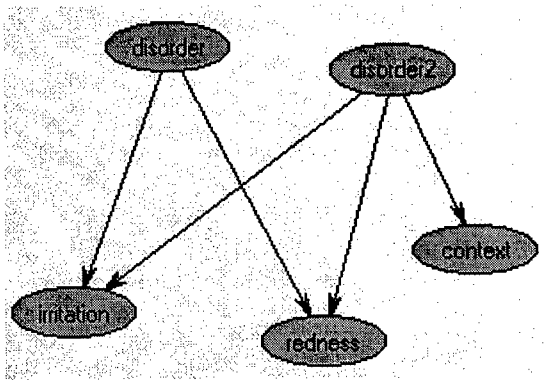


Figure 2: A multiple fault network, with fault nodes "disorder" and "disorder2." The disorder nodes create dependencies among the two observations "irritation" and "redness."

3.2 Finding globally optimal sequences of tests

The basic greedy algorithm for sequencing tests based on diagnostic value computes the value for all potential tests (or more generally, fusion inputs). It then selects the test with the highest value as the recommended next test. In an interactive setting, we would perform this test, then re-evaluate the remaining tests because their values may have changed based on the findings from the first test.

However, in building up a set of tests to recommend, we do not have the information about the first test's outcome at the time we select the second test; all we know is that the first test will be done. So "greedy" in this context is a little more complicated than it is in the one-test-at-a-time context. Here, the value for the second test must be computed for each possible result of the first test; these are then combined additively, weighted by the marginal probabilities of the outcomes on the first test. Similarly, once the second test has been selected, the remaining candidates must be re-evaluated based on all combinations of results from the first two tests.

It should be obvious that when there is a large number of possible tests, the number of combinations expands quite rapidly. Some simplifications can be achieved by noting that under certain conditions that can be ascertained from

the Bayes Network, the value for several tests will be independent of the newly added test, so no recalculation is needed. Furthermore, it may be possible to compute a bound on the potential impact of the newly added test on each remaining test, and recompute only the ones that show large potential effects.

If this algorithm is continued until the entire set of remaining tests is exhausted, the results can be plotted on a cumulative-value chart like the one shown in Figure 3. The values are shown as percentages, where zero corresponds to the state of information before any of the remaining tests have been done, and 100 percent corresponds to the value of performing all of the tests.

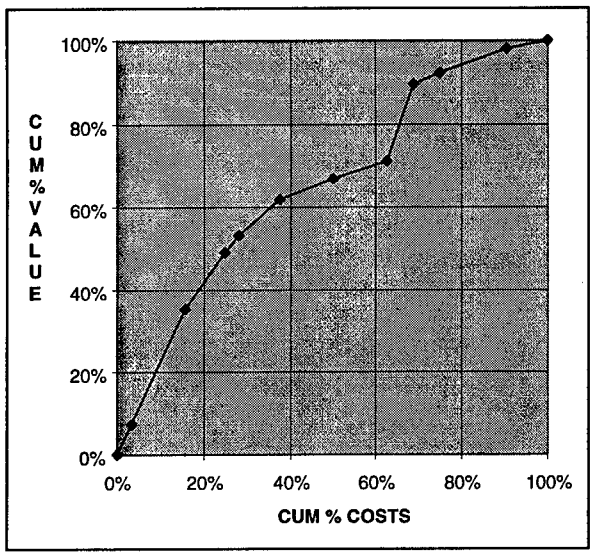


Figure 3: Initial results of greedy algorithm

Most of the time, the slope of each successive line segment will be less than or equal to its predecessor and the overall curve will be convex upward. In the curve shown, there is one instance where the curve displays a concavity. Any such concavity in the context of our greedy algorithm must indicate that the incremental value of the newly added item has changed because of a dependency on the immediately prior item (in the context of the other prior items).

If we call the two tests in question A and B, then the fact that A was chosen first indicates that, given whatever tests lie to the left on the chart, the value-slope for B was lower than that for A. The concavity indicates that the value-slope for B given A is greater than the value-slope for A. This algorithm assumes that while values of the various tests in a set may change depending on the sequencing of the tests, the total value of all tests in the set does not depend on sequencing. This is because we are simply adding tests to a "to-do" list, not observing their results.

The dependency suggests a way to repair this concavity by linking A and B together into a single entity "AB" whose value is *value(A) + value(B|A),* and cost is *cost(A) + cost(B|A).* We now reorder the tests using AB in place of A and B. (Only a limited number of tests need to be reordered; details of the reordering algorithm are beyond the scope of this paper.)

This combination step that is added to the greedy algorithm must be performed for all sequences of tests that have not been taken. This imposes a computational burden. In the next section we examine how the structure of the Bayes network can be exploited to ease this burden.

3.3 Simplifying the search for dependencies from the Bayes network structure.

The search for tests that must be combined in order to compute accurate test values can be simplified by the dependency relations that are evident on the Bayes network. From assumption 2 it is clear that observation variables whose only parent is a single fault will not change their rank with respect to other single parent siblings of the same fault. Thus once we have combined *fever* and *headache* in Figure 1, this principle tells us that no further test combinations need to be considered. Unfortunately this simplification applies only in single fault models. Heckerman *et al.* [5] made the equivalent observation that non-myopic computations needed to be considered only among sets of observation nodes that remain connected once the fault node is removed from the network.

In the more general case, the indirect dependencies between observations due to common faults must be considered. It is true in a strict sense that observations whose dependencies are only due to common faults are conditionally independent give the set of fault variables. This fact has little practical value since it requires combining all common faults when calculating the test values. The problem in the multiple fault model is that observations can be dependent on observations connected by paths through alternating fault and observation nodes. There are two principles that simplify this. The first is a consequence of d-separation [13], which says that paths through observation nodes that have not been observed can be ignored. Initially this is helpful, but as the set of observations starts to fill up the network, the dependencies proliferate. The second principle is the Markov property of the network. The consequence of this is that the effect of an observation path that passes through an observation node that has already been observed can have no greater effect that the already observed observation node had. The formalization of these principles with respect to test

value measures and their ability to limit the complexity of non-myopic search is beyond the scope of this paper.

4 An application to automobile diagnostics

The concept of active fusion applies to a comprehensive "senses only" automobile diagnostic model that we are building. The model covers 188 observations that can be made by a typical consumer without recourse to tools or instrumentation. Based on the observations that a user makes, the model ranks 180 subsystem faults that correspond to repair recommendations. It is implemented as a multiple fault model in a single Bayes network with 360 nodes. As a multiple fault model, it can identify simultaneously more than one fault from sets of observations that arise from the co-occurrence of faults. This is one of the largest diagnostic Bayes networks that has been constructed.

In addition to the fault rankings, the model ranks observations by their ability to discriminate best among the current candidate faults. For observation ranking the model currently uses an entropy-style measure of test value. The observation ranking features efficiently localize the systems in which faults are likely, based on the non-specific responses (e.g. noise, odor and drivability concerns) of a user who is not an expert auto mechanic. The ranking for next test candidates is central to making the model usable for the target audience. Even if the data were available at no cost, the "cost of confusion" to the user makes it necessary to guide the user's choice of tests based on active fusion concepts that we have presented. The advantages of active fusion for costless observations are similar to the reasons that feature subset selection is valuable in classification problems.

Our plans are to extend the automobile diagnostic model to incorporate test measurements made by a trained mechanic. This includes the digital code that current model cars provide through interfaces such as the new OBD II standard. In a model intended for mechanics, variables that were treated as unobservable can become directly observable. For example, low refrigerant levels that can only be inferred in the "consumer" model are measurable with the proper equipment. Metaphorically the "fringe" of observations in the model "rolls up" and the previous faults serve as observations, while "higher level" faults are added above. The profusion of new observation and fault variables will multiply the size of the model several times, but the higher diagnostic value of the added observations should permit convergence on a clear diagnosis with far fewer observations. We are pursuing the active fusion problems raised in this paper to address the needs of such a model.

5 Discussion and Future Directions

5.1 Previous work on non-myopic VOI

The limitations of myopic VOI were addressed in a paper by Heckerman *et al.* [5]. They formulated a true VOI problem where the decision variable has two alternatives so that the switch point between the alternatives is indicated by a probability threshold. The novel contribution of the paper was to approximate the series of observations by applying the central limit theorem to the sum of log-likelihoods of the observation variables to approximate the distribution of which side of the switch point that combined effect of the tests would fall. This gave an estimate of the probability that the set of observations would result in a change of the decision. This approximation assumes that the set of dependent observations is large. It has not been tested in practice.

5.2 Evaluation of the approach: Benefits of active fusion at different stages of diagnosis

The problem of optimal active fusion still raises many questions. The points that we have made in this paper are:

- Optimal active fusion can be defined by reference to VOI in a probabilistic model of diagnosis – this serves as a gold standard for any solution that can be proposed. Deviations from optimality can be found by looking for non-convexities in the test sequence Pareto curve.
- Non-myopic methods are necessary when observation variables are not conditionally independent given the fault variables of interest.
- The complexity of an exact solution to the problem depends strongly on the dense-ness of dependencies among observation variables. These dependencies may be mediated by other fault variables in the model.

5.3 Anticipated applications and future research

There is a recent growth in interest in the related problem of feature subset selection in the classification literature [14, 18] that is applicable to the active fusion problem. They address the question of what are the appropriate measures of test value. Optimal methods for active fusion depend upon this question. Additionally, decision theory addresses how outcome values come into play in comparison to the purely statistical approaches in the classification literature. We expect that the development of optimal active fusion methods will be driven strongly by results in these two areas.

6 References

[1] Cover., T.M. and Joy A. Thomas. *Elements of Information Theory* (New York, NY: Wiley, 1991).

[2] Ezawa, K. "Value of Evidence on Influence Diagrams" In Ramon de Mantaras and David Poole, Eds, *Proceedings of the Tenth Conference on Uncertainty and Artificial Intelligence*, (San Mateo, CA: Morgan Kaufmann, 1994) , pp212-220.

[3] Heckerman, D.; J. S. Breese and Koos Rommelse. "Troubleshooting under Uncertainty" (Redmond, WA: Microsoft Technical Report MSR-TR-94-07, January 1994).

[4] Heckerman, D.; J. S. Breese and Koos Rommelse. "Decision Theoretic Troubleshooting" *Communications of the ACM,* Vol. 38, No. 3, (March 1995) pp. 49-56.

[5] Heckerman, D.; E. J. Horvitz and B. Middleton. "An Approximate Non-Myopic Computation for Value of Information," In D'Ambrosio, Smets and Bonissone Eds, *Proceedings of the Seventh Conference on Uncertainty and Artificial Intelligence*, (San Mateo, CA: Morgan Kaufmann, 1991), pp135-141.

[6] Heckerman, D. *Probabilistic Similarity Networks* (Cambridge, MA: MIT Press, ACM doctoral dissertation series, 1990).

[7] Horvitz, E. J. *Computation and Action under Bounded Resources* (Stanford, CA: Stanford University Ph. D. Thesis, 1990).

[8] Howard, R. A. (1962) "The Used Car Buyer" In *Readings in Decision Analysis,* (Menlo Park, CA: Stanford Research Institute, 1977), pp. 491-520.

[9] Howard, R. A. "Information Value Theory" *IEEE Transactions on Systems Science and Cybernetics,* Vol, SSC-2, No. 1, August 1966, pp.22-26. In *Readings in Decision Analysis,* (Menlo Park, CA: Stanford Research Institute, 1977).

[10] Howard, R. A. "Information Value Theory" *IEEE Transactions on Systems Science and Cybernetics,* Vol, SSC-2, No. 1, August 1966, pp.22-26. In *Readings in Decision Analysis,* (Menlo Park, CA: Stanford Research Institute, 1977).

[11] Kalagnanam, J. and M. Henrion. "A Comparison of Decision Analysis and Expert Rules for Sequential Diagnosis," In R. D. Shachter, T. S. Levitt, L. N. Kanal and J. F. Lemmer, Eds. *Uncertainty and Artificial Intelligence, 4* (New York, NY: North Holland, 1990), pp. 253-270.

[12] Lauritzen, S. and D. Spiegelhalter. "Local computations with probabilities on graphical structures and their applications to expert systems," *J. Royal Statistical Society B*, 50, pp.157-224, 1988.

[13] Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference,* (San Mateo, CA: Morgan-Kaufman, 1988).

[14] Ripley, B. *Pattern Recognition and Neural Networks,* (Cambridge University Press, 1996).

[15] Shachter, R. "Evaluating Influence Diagrams" *Operations Research* vol. 33 No. 6 pp. 871-882 (1986).

[16] Shenoy, P.P. "A Valuation-Based Language for Expert Systems," International Journal of Approximate Reasoning, Vol. 3, No. 5 pp. 383-411 (1989).

[17] Tribus, M. *Rational Descriptions, Decisions and Designs* (New York: Pergamon Press, 1969).

[18] Wang, H.; D. Bell and F, Murtagh "Axiomatic Approach to Feature Subset Selection Based on Relevance," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 21, No. 3, March 1999, pp. 271-276.

Correlation of Heterogeneous Data With Fuzzy Logic

Chris Tseng and Arkady Epshteyn
Stottler Henke Associates Inc.
1660 S. Amphlett Blvd., Suite 350, San Mateo, CA 94402 USA
Email: tseng@shai.com, epshteyn@shai.com

Abstract:

A fuzzy logic based approach is used to infer the correlation of data in linguistic and numeric formats. Case based reasoning is used in our design to categorize our linguistic database. This technique is applied to an aircraft guidance problem to help the aircraft land more safely on the aircraft carrier. By correlating the numerical motion trajectories with the previous grading of related aircraft approaches in a linguistic database, an average of the latest ten approaches is presented to facilitate decision making. Fuzzy logic proves to be effective in delivering the data mining result in this problem environment characterized by heterogeneous information, uncertainties, and incomplete data.

1 Introduction

With increasingly widespread use of computers in recent years, the number of formats and types of data being stored has also increased dramatically. Correlating this stored data with the need of specific problems and producing a decision or recovering related information is not a trivial issue. Data mining of data of different nature, e.g., linguistic vs. numeric, is even more challenging. Fuzzy logic [1] has been used extensively in relating linguistic domain knowledge to numeric computation. Linguistic rules that summarize the domain knowledge are interpolated with numeric fuzzy membership functions for inference purposes. Fuzzy logic approach has proved to be a low cost and robust way of producing a quick around solution for many engineering problems [2]. Areas of fuzzy applications include control[3], query[4], data mining[5], and pattern recognition[6].

It is common to encounter uncertainties and noises in data mining problems. It is also typical that not all the information is available to provide a solution needed. The fuzzy approach is effective in dealing with both of these challenges. Case based reasoning [7] is most effective in retrieving similar cases to solve problems at hand. Aided with fuzzy logic reasoning, a case based reasoning design is expected to be more competitive in solving problems with uncertainties. Our innovative design is applied to construct a guidance aid for guiding fighter planes to land on aircraft carriers.

We will start by defining the problem in section 2. Basics of fuzzy logic are provided in section 3. The issue of data mining heterogeneous data is introduced in section 4. A real problem of creating a decision aid through data mining heterogeneous data for guiding aircraft to land on carriers is used to illustrate the design principle. Finally, future research issues are summarized in the Conclusion section.

2 Problem Statement

Consider the problem in which input information is in a different format than the data stored in the system and relevant data and/or inference are to be drawn based on some specified criterion.

Without loss of generality, we assume the input information is in numeric and the stored data is in linguistic form. Figure 2.1 illustrates this kind of problem. An inference engine is to be designed so that it can retrieve multiple pieces of data from the linguistic database and match the description of the numeric data presented as input. There are several challenging issues that need to be dealt with in this problem:

1) A mapping that maps the numeric features of the input data to the corresponding linguistic variable in the database.
2) More than one dataset, with different matching scores, may be produced from the data mining process.

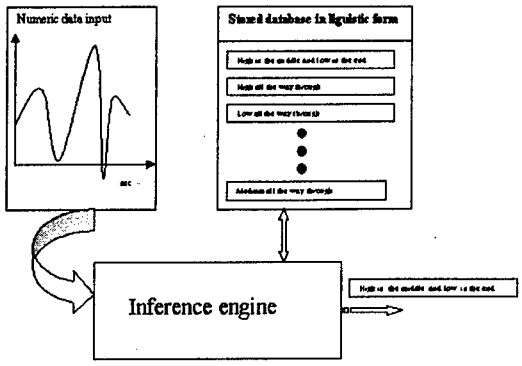


Figure 2.1 Heterogeneous Data Mining Problem

3) As more input data becomes available, the inference system needs to adaptively update the data mining result based on the latest overall input data.
4) The input data may be noisy.

Filtering will be applied in the preprocessing so that the effect caused by 4) is minimal. We shall assume the input data is prefiltered but noise may still be present. An expert may be consulted to construct the mapping needed for 1). However, different domain experts may have different subjective definitions of proper linguistic-to-numeric correlations. For example, the notion of high may be interpreted as 10 by one individual and 12 by another. Even though different interpretations of the same linguistic variables may not differ by too much, it is safe to assume that the chosen mapping can only function approximately in general. A methodology like fuzzy logic is needed to deal with this need. Furthermore, since the mapping between the linguistic and the numeric domain is only approximate, it makes sense for the system to generate more than one output with matching scores for reference purposes. This requirement can also be handled if fuzzy logic approach is adopted.

To be able to adjust the data mining result on the fly, previous inference outcome and the current inference result should both be considered in determining the matching data to be retrieved. It makes sense to weight the recent information more heavily in the overall inference process.

A real-life example of data mining a stored flight database in order to guide Navy aircrafts to land safely on carriers will be used to illustrate our approach in section 4.

3 Fuzzy Logic Inference

3.1 Fuzzy Membership Functions

The technology employed to fuse heterogeneous numeric and linguistic data is fuzzy logic. The concept of fuzzy sets was introduced by Zadeh in 1965. Since then, fuzzy logic has advanced in a wide variety of disciplines such as control theory, topology, linguistics, optimization, and category theory. Unlike a crisp set, a fuzzy set allows partial membership. Fuzzy logic is a generalization of the traditional TRUE/FALSE bilevel logic, one that allows for non-sharp transition, representing a region of partial truth, between absolute true and absolute false. For example, although the assertion that an individual is male is either true or false (and is therefore crisp), the assertion that an individual is fat is not so clearcut. Figure 3.1 demonstrates how the fuzzy sets may be used to capture this concept. A person with a body fat percentage of 16.5 has membership values of 0.12 and 0.43 in the "lean" and "moderately overweight" fuzzy sets, respectively.

3.2 Fuzzy Inference

The basic architecture of a fuzzy logic data analysis system is illustrated in Figure 3.2. The numerical input data is codified through the fuzzifier into the equivalent linguistic parameters (such as lean, moderately overweight, and obese), with associated membership function values. The inference engine uses the knowledge in a particular representation to derive some expert conclusion or offer expert advice. It includes the system's general problem-solving knowledge. Various rules in the knowledge base and decision-making logic are invoked and recover the decision actions with different degrees of emphasis depending on their respective membership values.

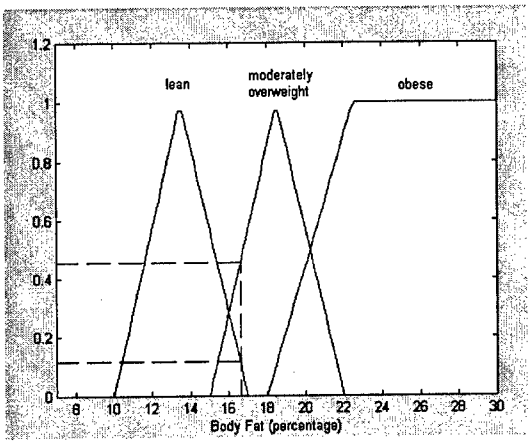


Figure 3.1 Fuzzy Membership Functions

A typical fuzzy rule might be: If you feel hot, the temperature is high. The final stage in the fuzzy logic data processor aggregates all the inferred fuzzy data and produces an appropriate conclusion or classification of the system's input. If the system's output needs to be in non-fuzzy numerical format, it is the responsibility of the defuzzification module to convert fuzzy data to numerical from.

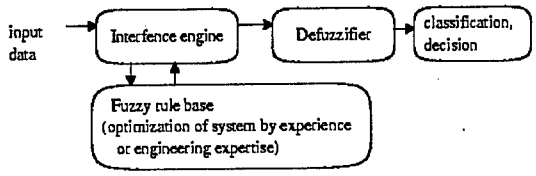


Figure 3.2 General architecture of fuzzy logic data analysis system

4 Data Mining Heterogeneous Databases

4.1 PADAL

The data mining approach discussed in this paper will be illustrated in solving the problem of construction of a Piloted Approach Decision Aid Logic system (PADAL), designed to provide guidance and advice in the domain of planes landing aboard aircraft carriers. This function is presently handled by landing signal officers (LSOs), navy personnel that offers corrections and feedback to the pilots attempting to land aboard the carriers. After the completion of each landing pass, the responsible LSO records a linguistic comment describing the pilot's trajectory and rates the pilot's performance. This information is subsequently stored in APARTS (Automated Performance and Readiness Training System) database. The goal of PADAL is to process the numerical radar data which provides information about the current landing trajectory and retrieve linguistic descriptions of similar landing passes executed by the same pilot in the past. These retrieved comments are merged to provide the landing signal officer with a concise summary of the pilot's past flight pattern. This procedure summarizes the pilot's performance in a succinct linguistic form and enables the user of PADAL to predict the pilot's future actions by consulting the summary of similar past behavior.

4.2 Linguistic APARTS Database.

The existing system stores trajectory descriptions in 2 formats: landing signal officers' (LSO) linguistic comments describing the previously executed landing approaches and numerical radar data which provides information about the current landing trajectory.

This section focuses on the linguistic representation of aircrafts' trajectories and the decoding technique used to analyze it. Each landing approach is subdivided into 5 stages based on the aircraft's distance from ship's deck: One Nautical Mile(1NM), At Start(X), In the Middle(IM), In Close(IC), and At Ramp(AR). These stages describe how far away the landing aircraft is from the deck. Signal officers' comments are recorded in a special shorthand code which describes various aspects of the pilot's approach for each landing stage. These shorthand comments can be subdivided into several major types: comments referring to glideslope (approach angle) of the landing aircraft, its lineup (horizontal distance from the center of the deck), rate of descent, power, pilot's attitude, and miscellaneous comments. An example of a lineup comment would be LUL (lineup left), LUR (lineup right), or CB (coming back), as well as several others. Plane's glideslope might be described by H (high) or LO (low) comments. TMRD (too much rate of descent) or NERD (not enough rate of descent) are two of the comments used to describe the

aircraft's rate of descent. Numerous other comments are used to reflect different properties of airplane's landing trajectory. The comments may be modified by the following 2 sets of symbols: ( ) (= a little) and _ _ (= very) which denote the degree of comment's applicability. For example, (F) means a little fast, while _TMP_ may be deciphered as way too much power.

The following sample comments illustrate the use of LSO's shorthand code:

H(LUL)X High and a little lined up left at the start.
HFIM High and fast in the middle
NEPLOIC Not nearly enough power, very low in close

The stage comments are combined to create a linguistic description of the entire landing trajectory:
(HX) NEP.CDIM LOBIC-AR A little high at the start. Not enough power on come down at the middle. Low and flat from in close to at the ramp.

The system contains a database of such linguistic comments which describe different landing approaches performed by various pilots. In order for this data to be useful, information contained in the comments must be extracted. The first step towards extraction and utilization of this information is parsing. Parsing reveals the structure of the comments represented by parse trees constructed from appropriate grammar rules. A chart parser based on the LSO-specific domain grammar is used to process the comments.
The following several rules are representative of the domain grammar:

COMMENT → DESCRIPTOR*
(a comment may consist of a number of consecutive descriptors)
DESCRIPTOR → LINEUP
(lineup is one type of a descriptor)
DESCRIPTOR → GLIDESLOPE (glideslope is another type of a descriptor)
LINEUP → LUL
(LUL (lineup left) is a shorthand code which contains lineup information)
LINEUP → LUR
(LUR (lineup right) is a shorthand code which contains lineup information).

The parser functions by reading a comment from left to right, trying to match it against all the applicable rules in the grammar, and keeps a constantly updated chart of all the active rules describing the currently processed portion of the comment. For example, LU... could refer to the beginning of the lineup left (LUL) comment, lineup right (LUR) comment, or simply describe the fact that the pilot is trying to lineup (LU). The next symbol in the string might disambiguate this expression. When the parser is done with the comment, it constructs a parse tree which summarizes the comment's structure.
The following example illustrates the result of application of this procedure to a sample comment:

Parse tree which represents HLULIM-IC high, lineup left in the middle and in close:

Comment
- Glideslope
  - H (high)
- Lineup
  - LUL (lineup left)
- Stage
  - Stage
    - IM (in the middle)
  - -
  - Stage
    - IC (in close)

Once a parse tree is constructed, the relevant information which can be used to infer the plane's trajectory is extracted by the program. For example, the above comment contains glideslope information (high) and lineup information (lineup left) which allows the application to determine the plane's position in the specified distance range (in the middle and in close). This intermediate analysis will be used subsequently in heterogeneous data fusion.

4.3 Numeric Motion Profile.

When a plane is attempting to land on a ship's deck, the landing signal officer's comment

describing the pilot's performance is not yet available to the system. However, the ship's radar constantly monitors the pilot's progress and relays the numerical aircraft position data to the system. This motion profile provides the

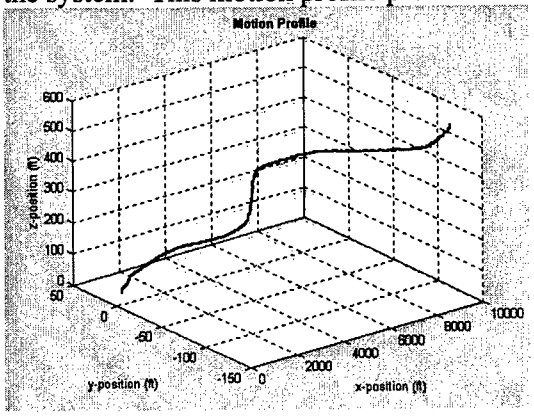


Figure 4.1 Aircraft Motion Profile

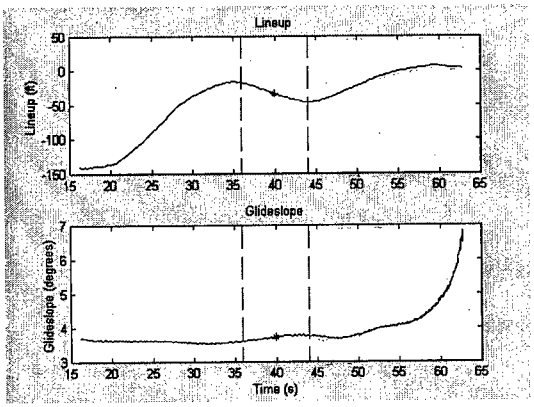


Figure 4.2 Lineup and Glideslope vs. Time

basis for analysis of the current landing trajectory and allows for its comparison with the previously executed landings. Figure 4.1 shows a sample numeric motion profile. Figure 4.2 illustrates decomposition of thus profile into corresponding lineup and glideslope trajectories.

4.4 Fuzzy Logic in PADAL Domain.

Fuzzy logic is employed in PADAL to perform numeric-to-linguistic conversion in order to ensure homogeneous data format necessary for information fusion. Fuzzy lineup and glideslope functions are represented in Figure 4.3. The lineup category consists of 7 fuzzy sets, ranging from significant left lineup (_LUL_) to significant right lineup (_LUR_). The glideslope category is subdivided into 7 analogous fuzzy sets which construct a "very high" (_H_) to "very low" (_LO_) classification of the aircraft's glideslope. These fuzzy sets map directly onto the comments used by LSOs to describe the aircraft's position.

Similar fuzzy definitions are constructed for various other parameters that define the landing trajectory. These fuzzy concepts enable the system to classify any point in the landing trajectory by associating fuzzy membership values with it. For example, a marked point in Figure 4.2 has the glideslope

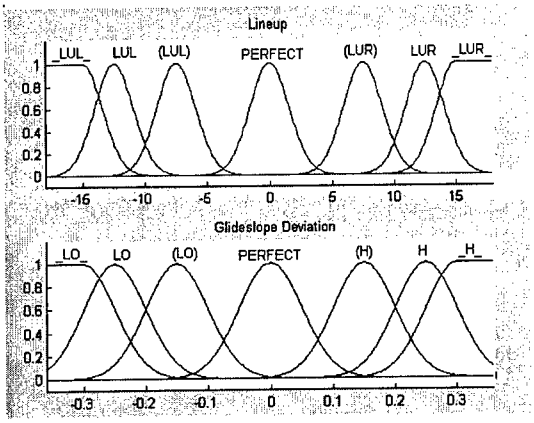


Figure 4.3 Lineup & Glideslope Fuzzy Membership Functions

deviation from the nominal glideslope (3.5°) of 3.7314°-3.5° = 0.2314°, which corresponds to the following glideslope classification:

Glideslope:

$\mu_{\_LO\_} = 0.00$ $\mu_{\_H\_} = 0.39$

$\mu_{LO} = 0.00$ $\mu_{H} = 0.93$

$\mu_{(LO)} = 0.00$ $\mu_{(H)} = 0.27$

$\mu_{PERFECT} = 0.00$

This means that an aircraft in that position is very likely to be classified as high by a landing signal officer, somewhat likely to be classified as very high or a little high, and extremely unlikely to be classified as low.

Some concepts may be represented by a union or intersection of fuzzy sets. For example, WU (wrapped up) is an LSO comment used to describe significant deviation of the aircraft from nominal lineup or glideslope in the

beginning of its landing pass. In cases like this, the fuzzy membership is calculated with a fuzzy OR operation which is defined by a maximum operation:
$\mu(WU) = MAX(\mu(\text{lineup WU}), \mu(\text{glideslope WU}))$

Once classification of a trajectory point is achieved, it is easy to classify a region by calculating the average of all the membership values of all the points in that region, i.e.

$$\mu_F(\text{region}) = \frac{\sum_{i=1}^{N} \mu_F(\text{point}_i)}{N} \qquad (4.1)$$

where
μ_F - membership value for fuzzy set F
$\text{point}_1 \ldots \text{point}_N$ – points which comprise the region in question
N - number of points in the region in question.

Figure 4.4 shows all the fuzzy set membership values for the region between 2 dotted lines in Figure 4.2 (corresponding to "In the Middle" aircraft landing stage).

At this point, the system is ready to retrieve previously stored linguistic cases most closely resembling the current landing trajectory. This is done by computing a similarity measure (SM) of the current numeric trajectory with respect to each

	[SLIGHT]	[NORMAL]	[HIGH]
LINEUP:			
DL	0.000	0.000	0.000
DR	0.000	0.000	0.000
CHLU	0.000	0.000	0.000
CB	0.189	0.189	0.189
LU	0.000	0.000	0.000
LUL	0.000	0.004	1.000
LUR	0.000	0.000	0.000
OS	.	.	.
NESA	.	.	.
WU	.	.	.
GLIDESLOPE:			
LO	0.000	0.000	0.000
H	0.598	0.061	0.008
B	0.000	0.000	0.000
RATE OF DESCENT:			
UP	0.000	0.000	0.000
DOWN	0.000	0.000	0.000
TMRD	0.099	0.033	0.001
NERD	0.336	0.187	0.209
SRD	0.000	0.000	0.000
CD	0.011	0.446	0.244
S	0.000	0.000	0.000
CO	.	.	.

Figure 4.4 Fuzzy Membership Values For the Selected Region of Figure 4.2

stored linguistic comment based on the following formula:

Let C be a comment which consists of several descriptors $D_1 \ldots D_N$:
Then

$$SM_{stage} = \sum_{i=1}^{N} \mu_{Di}(\text{region}) \qquad (4.2)$$

Each membership value μ_F represents the extent to which the current trajectory in the current stage can be classified as F. SM for each stage with respect to some comment C is computed by evaluating a sum of the membership values which determine how closely the numeric motion profile approximates each component of C.

Example: computation of $SM_{InTheMiddle}$ for comment (H)LULIM based on the data in Figure 4.4.
$SM_{InTheMiddle} = \mu_{(H)}(\text{InTheMiddle})$
$+ \mu_{LUL}(\text{InTheMiddle}) = 0.004 + 0.598 = 0.602$

Similarity Measure is computed online every time the approaching aircraft passes the next landing stage. Each time a similarity measure is recomputed, exponential forgetting is used to assign higher weight to the later stages:

$$SMtotal = \alpha SM_{stage} + (1 - \alpha) SM_{previous\ stage} \qquad (4.3)$$

This total similarity measure determines how similar the current motion profile is to a specified LSO comment. It is computed separately for every linguistic comment stored in APARTS database, and 10 most similar comments are retrieved for consequent processing, as illustrated in Figure 4.5.

SRD.(OSX) HDLIM HLULIC HCD.LUAR (0.413)
TMP.OSX HIM PNU.HIC SLO.CDAR (0.412)
SRD.OSX HSLOIM-IC PNU.CDAR (0.412)
(OSX) (H.CBIM) (NEP.CDIC) (LOAR) (0.315)
HOOT-X (HIM) (HCDIC-AR) (0.195) LOOSX /.CBIM HBIC-AR (0.120)
(LOOSX) OCNEP.CBIM HIC HCDAR (0.120)
(LO)OSX (TMP.CBIM) (HIC) (HCDAR) (0.120)
(LO)OSX /.CBIM HDLIC (\.LUAR) (0.120)
(/.OSX) TMP.CBIM HIC (HCDAR) (0.120)

Average: (LOOSX) CB(LURIM) HIC

Figure 4.5 Retrieval of Similar Comments. Each comment is followed by SM_{total} in parentheses.

4.5 Data Fusion

Retrieval of past similar cases is an inherently useful operation since it exposes the trends manifested by the aircraft's numeric motion profile. Moreover, it is reasonable to assume that previous landing approaches similar to the current one will also exhibit similar behavior in the future (especially if the search is restricted to previous landings performed by the same pilot). Thus, the PADAL system enables landing signal officers to predict the approaching pilot's future flight pattern based on past experience. However, PADAL is a time-critical system which needs to present information to the user in a concise and easily understandable fashion. In order to accomplish this, the system breaks up the comments into distinct stages and categories (lineup, glideslope, rate of descent, etc.) as described in the APARTS section of this paper. Once this operation is completed, the information contained in 10 distinct comments is merged within each category and each stage. This fusion operation takes place in 3 steps:

1) Transformation from linguistic to numeric domain,
2) Averaging, and
3) Transformation back to linguistic domain.

In order to accomplish the linguistic-to-numeric transformation ψ^N, a characteristic value is associated with each fuzzy set which summarizes its numeric content[5]. For a fuzzy set represented by a gaussian function, this number may be defined as the mean of that function. Conversion of a linguistic concept LC represented by a fuzzy set with characteristic value λ is defined as $\psi^N(LC) = \lambda$.

After the conversion of all the linguistic comments which belong to the same category in the same stage to numerical domain is completed, an averaging operation is applied to merge the data into a single numeric value. That number is converted back into the linguistic domain with a numeric-to-linguistic transformation function ψ^L.

Linguistic concept LC that corresponds to a numeric value v is defined as a concept represented by the fuzzy set F: $\psi^L(v)=F \mid \mu_F(v) \geq \mu_{F'}(v)\ \forall\ F' \neq F$ in the category that F belongs to.

The following example elucidates this procedure:
Assume that 3 comments contain the following lineup information in the "In The Middle" landing stage:
LUL, (LUL), (LUL), LUR
Then,
$\psi^N((LUL)) = -15$
$\psi^N(\_LUL\_) = -7.5$
$\psi^N(LUR) = 12.5$
(from Figure 4.3)

$$Average = \frac{-15-7.5-7.5+12.5}{4} = -4.375$$

$\mu_{\_LUL\_}(-4.375)=0.00$ $\mu_{\_LUR\_}(-4.375)=0.00$
$\mu_{LUL}(-4.375) = 0.00$ $\mu_{LUR}(-4.375) = 0.00$
$\mu_{(LUL)}(-4.375)=0.14$ $\mu_{(LUR)}(-4.375)=0.00$
$\mu_{PERFECT}(-4.375) = 0.02$
(from Figure 4.3)

$\text{Max}(0, 0, 0.14, 0.02, 0, 0, 0)=0.14=\mu_{(LUL)}$
$\therefore \psi^N(Average) = (LUL)$

Hence, fusion of _LUL_, (LUL), (LUL), and LUR produces (LUL). This process is used to compute the average comment that appears at the bottom of Figure 4.5.

5 Conclusion

A fuzzy logic approach aided with case-based reasoning is designed to solve a real

life problem involving data mining of heterogeneous data. The concept of computing with words through fuzzy membership functions proves to be effective in dealing with problems in which stored data and input appear in both linguistic and numeric formats. Although the aircraft landing guidance problem may not be the most challenging one, it represents a larger class of problems in which data mining and fusion of heterogeneous data is required to retrieve the needed information.

There are several important issues in PADAL domain that need to be researched further. By adaptively tuning the fuzzy inference engine or modifying the cases stored in a database, one should be able to achieve better performance and increase the robustness of the result. The use of genetic algorithms or neural nets to tune the parameters of our design may be a promising approach.

References

[1] Zadeh, L. A., "Fuzzy Sets," Informat. Control, vol. 8, pp. 338-353, 1965.

[2] Ross, T. J., "Fuzzy Logic with Engineering Applications," McGraw-Hill, 1995

[3] Lee, C. C., "Fuzzy Logic In Control Systems: Fuzzy Controllers – Part I and II," IEEE Trans. On Sys. Man and Cyber., vol. 2092, pp 404-43, 1990.

[4] Kacprzk, J. and Zardrozny S., "A Fuzzy Querying Interface For A WWW Environment," Proceedings of IFSA'97, Seventh International Fuzzy Systems Association World Congress, Czech Rep., pp285-290, 1997.

[5] Delgado, M., Herrera, F., Herrera-Viedma, E., Martinez, L., "Combining Numerical and Linguistic Information in Group Decision Making," Computing with Words, Kluwer, to appear, 1999.

[6] Bezdek, J, C., "Pattern Recognition with Fuzzy Objective Function Algorithms," Plenum Press, 1981.

[7] Kolodner, J., "Case-Based Reasoning," Morgan Kaufmann, 1993.

A Framework for Hypertext Based Diagnostic Information Fusion.

Charles-Claude Paupe
paupe@hds.utc.fr

Pierre Morizet-Mahoudeaux
pmorizet@hds.utc.fr

UMR CNRS 6599 Heudiasyc
University of Compiegne
B.P. 20529
60205 Compiegne-Cedex
France

Abstract

We have shown in preceding papers the feasibility of developing a framework for building hypertext based diagnosis systems. It was based on a novel definition and implementation of hypertext systems, which are more appropriate to account for the structural properties, which exist in any document describing a given engineering system. This framework also used a model ontology building method for representing our knowledge of a given system, its functioning process, as well as the occurrence of certain faults and the description of the corresponding diagnosis processes.

In this paper we present the development of this approach in the case of the vibration diagnosis of rotating machines. We have developed an ontology of rotating machines by using the model ontology building method embedded in our hypertext system. It permits to browse across the ontological knowledge base for performing the diagnosis process.

Given a certain problem, the user can chose among different methods, which one is the most appropriate. This choice relies on arguments, which are provided by the ontology. If several methods are used concurrently, the ontology provides a guidance for deciding, which tool to believe.

Key Words: diagnosis, information fusion, hypertext, ontology

1. Introduction

S. Abu Hakima has presented in [1] a thorough review of the state of the art in artificial intelligence (AI) techniques useful in diagnosis. Eight categories are identified, and a section of the report is dedicated to the analysis, discussion and prospective of each of the five that are the most relevant to diagnosis. They are: fault-based techniques, model-based techniques, case-based reasoning techniques, machine learning for knowledge acquisition, and integrated diagnostic techniques. The three remaining techniques are knowledge-based management, user interface and overviews relevant to diagnosis. Fault-based reasoning (FBR) techniques refer to what might be described as the experiential approach to represent human heuristic knowledge about maintenance and repair of a device or a process. Model-based reasoning (MBR) techniques use quantitative or qualitative models of the correct and expected behavior of a device to detect and to explain the discrepancy between the observation of the device and its behavior predicted by the model. Case-based reasoning (CBR) techniques refer to the ability of representing, managing, and updating our memory of previous studied cases of a device failure. Machine learning for knowledge acquisition uses either classification techniques on data examples and counter examples to build a domain theory, or conceptual models of the domain theory to build analogies. Integrated diagnostic techniques propose elaborated answers to the established fact that no single strategy is suitable for diagnosis. In each of the five sections of [1] the strength and weakness of each of the approaches are discussed, with an emphasis on how future works will resolve some of their shortcomings. The integrated diagnostic approach is presented as superior to each of the four preceding ones, in the sense that it takes advantage of each of their strength. For example, "model-based reasoning is used with fault-based reasoning to integrate in a single system the experiential knowledge of diagnosing a device with its expected behavior. This reduces computational complexity of finding a diagnosis using MBR. Model-based reasoning is also integrated with data interpretation to reduce the computational search. Explanation-based learning is used to refine the reasoning chains in rule-based FBR. Rule induction is also used to generate FBR systems. Fault-based reasoning is integrated with CBR as a means of acquiring new knowledge and reducing the search for a diagnosis. Similarly, MBR is integrated with CBR to accelerate diagnosis" [1, pp. 78]. In addition to integrating AI techniques, efficient results can be obtained by integrating AI techniques with more algorithmic techniques such as real-time (RT) approaches [2] [3].

Although very attractive one major weakness of the integrated approach is that the role played by each of the integrated techniques and their relationships are more or less fixed by the application domain. Even if the employed AI techniques use non-predictable search-based problem-solving approaches, selections are made from these

alternative problem-solving techniques given a priori knowledge of the most appropriate ones.

It is easy to imagine however that the same diagnostic system may be used differently according to the occurrence of a given failure with respect to the general state of the device. The same diagnostic system may present information, results and explanations differently according to the needs of the user. The same user may want to have different points of views of the same resolution process. The user may be interested in having some freedom for building relations and cooperation between different diagnostic techniques.

Hypertext systems have been thought as systems, which are able to represent and support the association of different information sources. Their purpose is to provide the user with the enriched information resulting from the association of the information sources. This objective is obtained by the use of navigation tools. Navigation tools use nodes and links. Each node is an information source. Links are built from one part of a node to another node (or a component of another node).

If the information sources were diagnostic techniques such as presented above, and if the links were the representation of how they interact to resolve a given diagnostic problem, the resulting hypertext system may be seen as a good candidate for diagnostic information fusion. However, although necessary for proposing interesting solutions, the common hypertext approach is not sufficient for providing the user with functions such as comparison of diagnoses, tools expansion, confidence improvement or inconsistency explanation, and so forth. This calls for proposing a hypertext scheme, which supports if not all, at least some of these functions.

In previous work, we have proposed a framework for developing task oriented navigation tools for hypertext systems. It is based on the notion of *contextual navigation*. Its objective is to account for the fact that the path followed by the hypertext reader is not only defined by the relations between nodes, but also by the context in which links and other possibilities of navigation appear. This led us to define the notion of a digital document, which contains its own potential navigation structures. These potential structures become *perceptible* through *projections*, which correspond to interpretation and instantiation operations [5]. We have also shown that by separating nodes content from hypertext documents structures, it becomes possible to implement tools, that we call the *instrumentation of the reading*, which support the dynamic creation of documents while reading [4].

We have also been working on building a diagnosis typology and diagnosis ontology in the application domain of fault diagnosis of rotating machines. The approach is based on the assumption that, when limited to certain professional domain, the behavior of the user corresponds to types, which are *normalized by their practice*. We used a generic tool, supported by a programming language, for building domain ontology, which is currently under development in our laboratory [6].

In this paper we present the third step of this work, which consists in integrating our hypertext scheme and our fault diagnosis of rotating machines ontology in a single tool. Its objective is to provide the user with efficient hypertext tools for:

- building and browsing a domain ontology, including process description such as the occurrence of a fault or a diagnostic method,
- building and browsing digital documents, which describe machines, parts, functioning, faults, diagnostic methods,
- building while reading (browsing) synthetic documents, which propose alternatives of diagnoses for given faults.

As sated above, developing such a system calls for an appropriate hypertext scheme. Section 2 presents a brief summary of our previous work on this subject. It describes a formal approach and an implementation scheme for representing the internal generic structure of a digital document and building different projections of information relevant to certain request of the reader. Section 3 presents our general approach for building domain dependent diagnosis ontology. After introducing our formal language for ontology representation, we present its hypertext based implementation. Then an application to the ontology of rotating machines is presented. And an example of vibration diagnosis is described in section 4. Section 5 is the discussion and the presentation of future work.

2. Hypertext for Structured Digital Documents Management

The definition and the implementation of a hypertext scheme, which makes it possible to provide users with efficient tools for digital document management has been thoroughly presented in [4,[5]. Nevertheless the presentation of the hypertext diagnosis concepts requires some familiarity with our novel definition of a hypertext system, structured documents and synthetic documents built while reading. The aim of this section is to present the material needed in the further sections.

Generally, hypertext is defined as a network of information nodes, connected by links that allow passing automatically from one to the other. This builds up a graph structure that defines the hypertext. It provides the users with the ability to create, manipulate, or examine a network of information-containing nodes interconnected by relational links. This representation allows using a graph to represent a whole hypertext system. One

important shortcoming of this approach is that nodes have *a fixed display*. Although not explicitly stated, this assumes that the node content implies the node display. This comes from the principle that the nodes format includes the way in which the node is displayed. Consequently, most models do not take into account the content of the node itself, stating that the way in which it is displayed is not dependant on the hypertext system. These systems are only characterized by the graph of nodes. Nodes are considered as functional units, structurally and semantically complete, as if they were still on a static medium. Another characteristic of the current models is that the links do not belong to the description of the nodes. A link is "anchored" in the node, but this anchoring is considered to be dependent on the structure and the format of the node. The model of the interaction of the reader is the description of a path in the graph of nodes. The problem to be solved is then to find the relevant path, i.e. the relevant actions of the reader for reading the hypertext, during a single session. Instead of models of the reader's path in the graph, we can build models of the reader's reading action. They are based on structuring the instantaneous state of the hypertext system during the reading action.

The action of reading can be limited neither to the study of paths in graph of nodes, nor to the study of states of the hypertext while reading. Reading includes also the context in which navigation functions appear in a node. The information, which appear in a hypertext document, influence the information supported by the link. Thus the meaning of a link depends both on the structure and on the content of a node. Hypertext nodes *embed* their own way of navigation for browsing documents. Therefore, they are highly dependent on the content of the node. By putting this navigation tools into context, hypertext allows the reader to navigate while reading. It results in a *contextual navigation*. This implies that the path taken by the hypertext reader is not only defined by the relation of the nodes, but also by the context in which links and other possibilities of navigation appear.

Thus we come up with a definition of digital documents, which is closer to its use. A digital document is composed of a limited and linear set of values. Each document follows a format, which allows its interpretation into a more complex structure. For example, a bitmap graphic file contains the size of the picture and the values of the pixels. By knowing the width of the picture and the number of bit per pixel, any program can convert it into a matrix of values, which can be transformed further into a screen picture. The format specifies the structure of the file, and the use of the data.

We cannot access the document itself, as a set of values. We only access it through what we call *projection*. It is the means by which a format is made perceptible. A projection can be exhaustive, partial, or *synthetic*. The table of contents from a structured text is a projection of the document, even if it omits a great part of the file, since it is still a view of the same file. Another synthetic projection can be an index of a document; any diagram based on statistics, a graph of a resolution procedure, an intensity diagram of a graphic file.

As we have seen above, hypertext differ from other kinds of interaction with digital document by the localization of the navigation tools in the document. Thus, we can define hypertext documents from this characteristic. The navigation tools of the hypertext are produced by the projection of the node content.. With this approach hypertext is considered from the point of view of the nodes. This leads to consider that the main characteristic of hypertext is that the navigation structures are localized in the nodes.

The most general way for implementing a hypertext system is to use a structural markup scheme, for example SGML [7]. It makes it possible to build representations of documents. Rather than relying on explicitly marked links, navigation is driven by the structures of the documents. We have used this frame to develop a prototype of electronic patient record management with a hypertext system [4].

The medical record belongs to a class of hyperdocuments that we call *dossier*. The generics of the uses of the dossiers are one of their major characteristics. They can be used for very different tasks, and so they can be considered as working tools. In a professional context, it is possible to work out working tasks relying on the different uses of a dossier. This classification leads to a low number of typical consultations of the dossier. The type of the consultation determines a reading strategy. These types correspond to different reading strategies, driving to different kinds of readings. In the same way as reading situations are standardized by their professional context, reading types are also standardized as corresponding to well defined activities. We can note that these reading types rely on the same structures, which are used in very different ways. An important characteristic of these structures is the generics of their use.

We have established that some readings are standardized enough to be useful to many potential readers of the record. Then, the result of this reading may be collected in a new synthetic document. We call such a document a *structuring document*, because it proposes a reading of the record by selecting parts and organizing them into a new structure. A structuring document offers a direct access to selected contents. Therefore, structuring documents can become reading tools, with the potentiality of dramatically increasing the reader' productivity.

The implementation of these synthesis tools has been made using the Standard Generalized Markup Language (SGML). Documents are described by a

SGML Document Type Definition (DTD), using medical content tags, *e.g.* <medical-history>, and each tag used in the DTDs has a type defined by a generic tag of the architectural form, *e.g.* <section>, <section-title>, etc.
When new documents are added to the documentary database, synthesis tools are activated to generate the corresponding structuring documents. Pieces of documents are copied and organized into new structured documents. These new documents are added to the documentary database with the same status as the previous ones.
The generation of new documents, especially synthesis documents, implies the duplication of parts of the content of the generic documents. Actually, the content is not duplicated straightforward, but links are built in the generated document toward the generic document. It is however necessary to have links between duplicated and original contents in both direct and reverse direction to have a satisfying document *genesis*.

This system has been built based on an empirical ontology of the physician's practice. In the following section, we present a more formal approach for building ontology. It is based on a formal language and its hypertext implementation that we have applied to the domain of rotating machines.

3. Rotating Machines Ontology

To create a hypertext based diagnostic information fusion system, we need a thorough description and presentation of our knowledge of the domain. This knowledge is not a descriptive, but a synthetic and structural one. We need to know how each diagnosis concept is articulated to the others. This knowledge is conveyed by what is called an *ontology*. Indeed, an ontology is threefold — it contains the descriptions of the various sorts of objects of the studied domain, of their properties and of their links with other objects in the domain [9]. Thus, ontological analysis is mainly concerned with the way knowledge is structured, and not only with knowledge alone. By describing the sorts of objects in the studied domain, the analyst creates terms, which have universal value as concepts. Hence, a language is created, which represents the knowledge involved in the domain. The words used in this language are distinct from the words of the natural language. Indeed, in a natural language, the definition of words is supposed to be based on a common understanding between users, but in practice it is well-know that users frequently misuse words. Moreover, natural language definitions are too loose and often do not make clear the difference between two close items. On the contrary, an ontological language is newly created, even though it may use words belonging to the natural language. It is based on a newly established convention, and therefore, it can reduce ambiguity. For example, when this language needs define two close items, two different terms will be provided.
We can therefore say that ontologies do not depend on the kind of task which is to be performed in the domain. Ontology defines knowledge in a given domain, by capturing its intrinsic conceptual structure [9].
According Chandersekaran et al. [9] ontology has two dimensions:

- *Domain factual knowledge* provides knowledge about the objective realities in the domain of interest (objects, relations, events, states, ...)
- *Problem-solving knowledge* provides knowledge about how to achieve various goals. A piece of this knowledge might be in the form of a problem-solving method specifying in a domain independent manner how to accomplish a class of goals.

However Valente et al. [10] consider that in the case of a KBPS (Knowledge Based Problem Solving), ontologies are not used to describe the domain, but to support applications. In this case, ontologies do not develop the whole knowledge involved in the field, but only that which is necessary to a good understanding of the specific application. We will develop our problem solving ontology according to this line.
There are several knowledge representation languages for describing domain through an ontology. Genesereth and Fikes describe KIF (Knowledge Interchange Format), an enabling technology that facilitates expressing domain factual knowledge using a formalism based on augmented predicate calculus [11]. Neches et al. describe a knowledge-sharing initiative [12], while Gruber has proposed a language called *Ontolingua* to help construct portable ontologies [13]. The CommonKADS project has taken a similar approach to modeling domain knowledge [14]. These languages mainly describe knowledge of a domain. They permit to share knowledge, but we want more than knowledge sharing. We want to be able to describe and use some problem solving and resolution methods.
Our approach is based on the formal language Def-* [15]. It belongs to the type of languages developed to formalize and/or operationalize conceptual models, according to methods similar to CommonKADS. Def-* is dedicated to the formalization of operational models. It is a high level programming language and is more declarative than the rules production languages used in early expert systems. Besides, Def-* is based on "epistemological premises", which define the items of knowledge the language is able to represent, thus forming a "representation ontology".

One of the central characteristics of Def-* is that it makes it possible to formalize reflexive tasks. These tasks can be assimilated to problem-solving activities, just like diagnosis tasks.
So, we have developed an ontology of rotating machine by using the model ontology building method embedded in our system.

We define three types of concepts in our ontology of rotating machines. They are the generic concept, which concerns objects, the relation concept, which concerns relations between concepts, and the task concept, which concerns diagnostic processes.

Generic Concept

In Def-*, a generic concept is both a series of objects and an entity. A definition introduced by Def-Concept encapsulates the representation of the concept's intension together with the representation of its properties. The conceptual definition situates the concept in the taxonomy by using the properties, which make it different from all other concepts. A definition of the concept is also written in natural language, for a better understanding by the user. The natural language definition is twofold: first, a conceptual definition translating the properties defined by Def-*, and second a "dictionary" type definition with a reference. It may be completed by an example and a multimedia illustration (picture, film, sound) of the concept. The concepts are therefore defined as the specialization of other concepts. The result is a tree-shaped taxonomy (Figure 1).

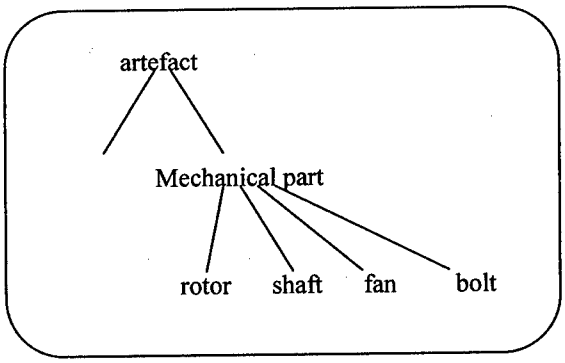


Figure 1: the tree-shaped taxonomy

Relations Concept

After defining concepts, we define their relations. These relations are especially important for the definition of the properties of each concept. Relations are defined by the "Def-relation" primitive in the same ways as the Def-concept primitive. Moreover, this construction makes it possible to define both the relational concept and all the couples, which are concerned by the concept. Figure 2 presents a simple relation concept of "is composed by".

```
Def-relation #is composed by

        is-a [#object-relation]
```

Figure 2:example of Def-relation syntax

Task Concept

The tasks concern the representation of control knowledge, which is actually the representation of diagnosis. The "Def-Task" primitive makes it possible to represent both a goal and the resolution method necessary to reach this goal. For example, in Figure 3, we define the global machine application corresponding to norm NF E90-300.

```
Def-task #global-state-of-the-machine

        Data = vibration speed
        Control =and

        If vibration speed = good
                Then #no problem
                Else If vibration speed =
admissible
```

Figure 3: a example of Def-task syntax

We have implemented Def-* in the Standard Generalized Markup Language (SGML) (Figure 4).

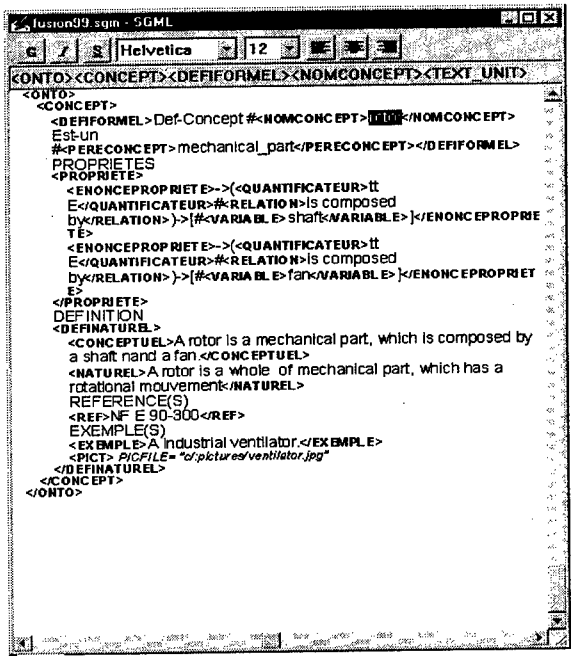


Figure 4: a example of SGML implementation of a concept written in the Def-* formalism

This makes it possible to embed the ontological knowledge base in the hypertext system, while retaining the representative power of the language. Furthermore it permits to browse across the ontological knowledge base for performing the diagnosis process. Each primitive (Def-concept, Def-relation,...) is represented by a DTD. The ontology is represented by a another DTD, which includes the DTD of the different primitive. An example is given Figure 5.

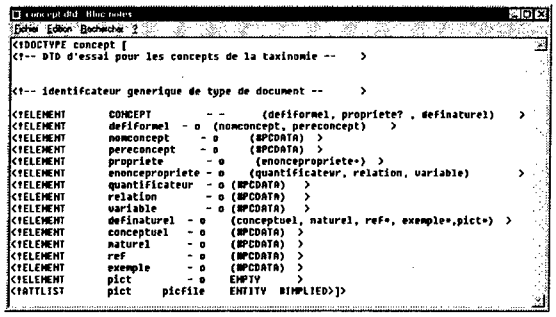


Figure 5: DTD of the Def-concept primitive

4. Example of a Vibration Diagnosis of a Rotating Machine

The validation of our approach has been tested with a simple rotating machine example. We consider a rotating machine comprised of a shaft, a fan (thus forming the rotor), and two bearings, which allow the shaft to rotate. Although it is simple, this rotating machine makes it possible to find dysfunctions (i.e. vibration speeds, which are not admissible according to the Norm NF E90-300). In our example, we mainly deal with the global state of the machine as it is defined in Norm NF E90-300. Once the state is defined, if necessary, we look for faults and their respective causes. The term "fault" means "a cause of inadmissible vibration speeds", such as unbalance or misalignment.

Our hypertext tool does not provide any definitive solution to the problem, but it purports to help the user for diagnosing. It gives information to make the right choice, and suggests pertinent elements, which will allow the user to achieve a diagnosis task, while leaning on the knowledge of the domain as it is provided by the system.

We illustrate the following description with a example of rotating machines.

The user informs the system about the type of machine to analyze, *for example a simple rotor with 1.5kW engine.* The system can thus know which group the machine belongs to according to Norm NF E90-300, (*group I*). It can also suggest various measurement locations where the user can put transducers, *axial and radial bearing measures.* These locations will contribute useful information for the diagnosis.

The user collects the measures at the defined locations and transfers them to the system.

The system then treats the data (vibratory signals) with a MatLab® software, and thus defines the global state of the machine, *the vibration speed is equals 5mm/s, so the state is not admissible.* If the state corresponds to a dysfunction of the rotating machine, the system informs the user and suggests various protocols to find the cause of faults (unbalance, misalignment). Each protocol is characterized by the measurement locations and the processes performed on the associated signals. The system thus guides the user toward a protocol using signals, which already exist, instead of guiding the user towards a protocol, which would make it necessary to acquire new signals.

The user chooses the diagnosis protocol according to the system's advice, *unbalance detection protocol with radial bearing measure.* Once the protocol has been chosen, the system gives the information, which is necessary to interpret the measurement results and to locate the fault (unbalance or misalignment), *there is a harmonic of the rotating frequency, while appears in the spectrum, so there is an unbalance.* When the fault has been located, the user can look for its cause (ball-bearing wear, broken blade or vane section, etc.). The system can then suggest to look for the cause of the faults thanks to observable signs, *an eccentric accumulation of process dirt on blade.* The table on

Figure 6 represents the correspondence between causes of unbalance and observable signs.

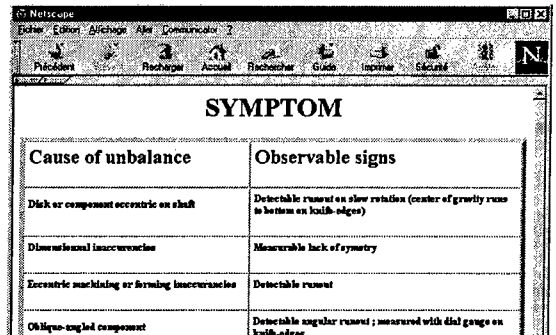


Figure 6: visualization of new document

This document, as any document orienting the user, comes from the manipulation of a SGML document and the dynamic creation of a new SGML document which is then translated into the HTML in order to be visualized.

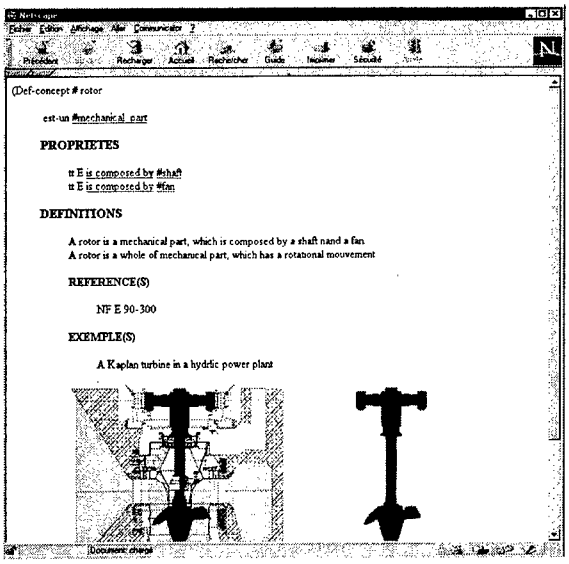


Figure 7: an example of HTML visualization

5. Conclusion and Future Work

We have presented a hypertext based system, which makes it possible to associate different information sources for performing a diagnostic task.

The system relies mainly on context and task oriented navigation tools for user guidance. Its main contribution is to facilitate the access to relevant information for choosing the most appropriate diagnostic method.

Further work will be to add information sources and reasoning tools for improving the comparison between concurrent diagnostic methods. Based on the ontology of the diagnosis problem solving, their main objective would be to facilitate the browsing, by the user of the ontology, ad thus facilitating the access to the most relevant information for choosing among different methods.

6. References

[1] Abu-Hakina, S. Artificial Intelligence Techniques in Diagnosis: a review of Approaches, Applications, and Issues, Technical Report, NRC 37142, National Research Council Canada. 1994.

[2] Musliner, D.J.; Hendler, J.A.; Agrawala, A.K.;and Simon, H. The Challenges of Real-Time AI, *Computer* 28(1): 58-66. 1995.

[3] Morizet-Mahoudeaux, P. On-Board and Real-Time Expert Control, *IEEE expert, Intelligent Systems and their Applications* 11(4): 71-81. 1996.

[4] Morizet-Mahoudeaux P., Terray P., Brunie V., Kassel G., Toward Hypertext System Based Diagnosis, Ninth International Workshop on Pirnciples of Diagnosis, Cape Cod, MA, USA, 23-27May, 1998.

[5] Brunie, V.; Morizet-Mahoudeaux, P.; and Bachimont, B. Separating Textual Content from Structures for Reading Hypertext Structured Medical Records, in Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia, Pittsburgh, PA, Forthcoming. 1998.

[6] Kassel, G.;and Traore, M. Modelling with a Strongly Intensional Language Brings Numerous Advantages, In Proceedings of the Seventh Workshop on Knowledge Engineering, Milton Keynes, UK. 1997

[7] SGML. ISO Information processing, Text and office systems, Standard Generalized Markup Language. ISO 8879. 1986.

[8] Nygren, E.; and Henriksson, P. Reading the Medical Records I: Analysis of Physicians' Way of Reading Medical Record, *Computer Methods and Programs in Biomedicine* 39: 1-12. 1992

[9] Chandersekaran B.; Josephson J. R.; and Benjamins R. What Are Ontologies, and Why Do We Need Them? *IEEE Intelligent Systems & their applications* 14(1) : 20-26. 1999.

[10] Valente A.; Russ T.; MacGregor R.; and Swartout W. 1999. Building and (Re)Using an Ontology of Air Campaign Planning. , ?, *IEEE Intelligent Systems & their applications* 14(1): 27-36.

[11] Genesereth M.; and Fikes R. Knowledge interchange Format, version 0.3, *Knowledge System Lab., Stanford Univ.* 1992.

[12] Neches R. et al. Enabling Technology for Knowledge Sharing. *AI Magazine* 12(3): 36-56. 1991.

[13] Gruber T. A translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, Vol.5: 199-220. 1993.

[14] Schreiber G. et al. CommonKADS: A comprehensive Methodology for KBS Development. *IEEE Expert* 9(6): 28-37.1994.

[15] Kassel G. Def-* manuel de référence. *Laboratoire de Recherche en Informatique Amiens (LARIA).* 1999.

Application of information fusion on flaw detection of concrete structure

Yang Xiang
College of marine engineering,
Wuhan transportation university
Wuhan 430063, China
Email:p611011@public.wh.hb.cn

Xizhi Shi
State key Lab of vibration, shock & noise,
Shanghai Jiao tong university
Shanghai 200030, China
Email:xzshi@mail.sjtu.edu.cn

Abstract

Nondestructive flaw detection of concrete structures is a difficult work, especially for detection of small or shallow flaws. In our research, both ultrasonic test and impact-echo test are used for detection of simulated flaws of different sizes because only one detecting method can not give out a believable conclusion sometimes. After signals are collected, wavelet analyses are used for feature extraction from these two kinds of signals. Then a feed-forward multi-layer neural network is used to implement local soft classification. After that, Shafer-Dempster reasoning is used for decision-level identity fusion and the hard decision of flaw detection is made.

Key words: concrete, flaw detection, information fusion, soft decision, evidence theory

I. Introduction

Nondestructive test of heterogeneous materials is an important and difficult work. Concrete is a typical heterogeneous material. Its flaw detection is very difficult, especially for detection of small or shallow flaws in concrete slabs. Ultrasonic test and Impact-echo test are usually used for flaw detection in concrete slabs. Sometimes, only one detecting method can not give out a believable conclusion[1~8]. So in our research, both ultrasonic test and impact-echo test are used for detection of simulated flaws of different sizes such as delamination and void. In this paper, after signals are collected, wavelet analyses are used for feature extraction. Thus, original signal can be expressed as a feature Vector. Because of the inherent complexity of heterogeneous materials, randomness of testing environment and influence of noise, in such cases it is not appropriate, even for an optimally designed classifier, to make hard decisions. So the inevitable uncertainty in classification has been accounted for in the form of soft decision vectors and membership values have been expressed in terms of smooth functions. Here the classifier is a typically non-linear map from the feature space to the points in the fuzzy cube. The classifier has 3 non-binary outputs, one for each class, non-defect, delamination, void, where each output takes values in[0,1]. The output of the classifier i.e., the fuzzy membership value is assigned as the basic probability mass to different classes. Thus two basic probability masses from ultrasonic test and Impact-echo test for a same target are provided. Then Shafer-Dempster reasoning can be used to get a unified belief function. In the paper, four rules are provided as the criteria of hard decision. If four rules are satisfied, a hard decision of flaw detection is given out.

II. Ultrasonic testing and impact-echo testing system

The experiments are carried out on three

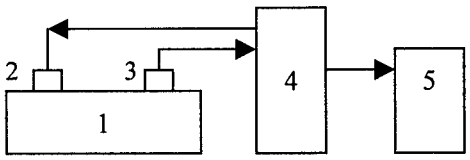
concrete slabs. The aim of our research is to identify various flaws by ultrasonic testing, impact-echo testing and their fusion.

2.1 Ultrasonic testing system

For ultrasonic test, there are two methods available to detect flaws in concrete slabs, which are through-measure method (spacing transducers oppositely on each surface of the slab) and flat-measure method (arranging transducers on a same surface of the slab). Because there are many concrete structures such as cement concrete pavement, airport's runway and tunnel spray that aren't possible for through-measure, thus flat-measure method is used in our experiments.

Table 1 Specification of specimens

Specimen	Sizes of slabs (m^3)	Types of Flaws	Sizes of flaws (mm)
1	1×1×0.2	Non-defect	
2	1×1×0.2	delamination	200×200×1、50×50×1
3	1×1×0.2	void	Φ50、Φ30

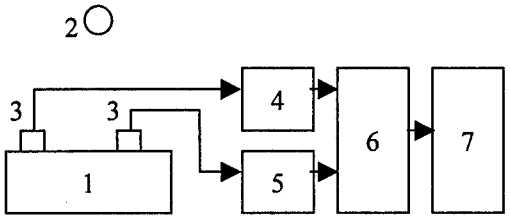


1. specimen
2, 3. transducer
4. ultrasonoscope
5. computer

Fig.1 Diagram of ultrasonic testing system

As shown in Fig.1, the system consists of two transducers, a ultrasonoscope and a computer. The frequency of the transducers used for detecting concrete should not be too high. The central frequency is at about 130 KHz. Furthermore, they must be wide-band in order to acquire high resolution in time domain, it is say that the transducers should have smooth amplitude response and linear phase response in a wide frequency range, which offers convenience for transmitting narrow pulse signal. In the experiments, NM-2B ultrasonoscope is used for transmitting ultrasonic signal to the specimen and receiving echo signal, its sampling frequency is selected as 2.5MHz. The digital signals are conveyed from the ultrasonoscope to a personal computer, in order to analyze them by applying various methods.

2.2 Impact-echo testing system



1. specimen
2. bearing ball
3. transducer
4, 5. charge amplifier
6. oscilloscope

Fig.2 Diagram of impact-echo testing system

As shown in Fig.2, the system consists of two transducers, two charge amplifiers a digital oscilloscope and a computer. The piezoelectric accelerator transducers B&K 8309 (the frequency band is 0~50KHz), the charge amplifiers YE5858, the digital oscilloscope HP infinium 54815A are chosen. Hardened bearing balls are used to produce exiting force. Echo signals are collected by the transducers. After amplified, they are transferred to

the oscilloscope and are changed as digital signals. Then they are conveyed to a computer for further analyses.

III. Signal processing

In order to reduce the influence of exiting force on amplitudes of signals, the normalization and centralization of signals are carried out before feature extraction. The arriving point of R wave is considered as the first point. The length of signals used is 800 μ s. After elimination of noise, wavelet transform is used for feature extraction. As we know, wavelet transform is a very promising technique for time-frequency analysis. By decomposing signals into elementary building block that are well localized both in time and frequency, the WT can characterize the local regularity of signals. This feature can be used to distinguish different flaws. Here, a dyadic wavelet transform is used and the local maximums of the WT modulus at the third scale are used for feature extraction. The dyadic WT of a digital signal is calculated with Mallat algorithm. The wavelet we used is a quadratic spline wavelet with compact support and one vanishing moment. It is a first derivative of a smooth function. The feature extraction process can efficiently provide class separation with a small number of features. After this process, each ultrasonic signal or each impact-echo signal can be expressed as a feature vector. In fact, non-destructive flaw detection of concrete structure is a kind of pattern recognition. The concept of pattern recognition may be expressed in terms of the partition of feature space (or a mapping from feature space to decision space). Suppose that N features are to be measured from each input pattern. Each set of N features can be considered as a vector X, called a feature (measurement) vector, or a point in the N-dimensional feature space Ω.

The problem of classification is to assign each possible vector or point in the feature space to a proper pattern class. This can be interpreted as a partition of the feature space into mutually exclusive regions and each region will correspond to a particular pattern class. For such a problem of pattern recognition, each possible pattern class J can be expressed as vector U_J:

$$U_J \subset \Omega \tag{1}$$

$$U = (U_1, U_2, \cdots, U_J, \cdots, U_L) \quad (J = 1,2,\cdots,L) \tag{2}$$

$$U_J = [u_{j1} u_{j2} \cdots u_{ji} \cdots u_{jn}]' \tag{3}$$

u_{ji} represents the i th feature measurement of U_J:

$$u_{ji} \quad (i = 1,2,\cdots,n)$$

In the application, let X be designated as the extracted feature vector from signal:

$$X = [x_1 x_2 \cdots x_i \cdots x_n]' \tag{4}$$

What we needed to do is to assign it to one possible pattern U_J belongs to U, or describe the possible degree of the vector X belongs to one pattern class U_J, i.e. a membership between X and U_J.

In our research, we have collected adequate signals of known patterns (from simulated flaws), therefore, neural networks can be used to produce this fuzzy membership. Hard decisions of detection are achieved by the integration of local soft decision according to the rules in section 4.

This membership is actual a kind of soft decision[9]. The outputs of the neural network are the vectors of soft decisions which are based on the ambiguity and fuzziness of decisions. As we

mentioned above, it is difficult for only one method to make believable hard decisions because of the inherent complexity of heterogeneous materials, randomness of testing environment, the influence of testing noise and low repeatability of tests. Thus, here we use soft vectors to represent uncertainty of decisions. Then, Shafer-Dempster reasoning is used to reduce this uncertainty and getting a final believable hard decision. In addition, in non-destructive test of heterogeneous material, clusters of feature points, corresponding to different classes, overlap each other. These overlaps represent uncertainties that may come from local random variations among samples or may be due to the inherently fuzziness. For such cases it is not appropriate, even for an optimally designed classifier, to make correct hard decisions. The inevitable uncertainties in classification should be accounted for in the form of soft decision vectors and membership values should be expressed in terms of smooth functions to create fuzzy decision boundaries between clusters.

Consider a collection of N examples $\{S_i\}_{i=1}^{N}$ from L different, but known, classes. In our application, Γ_s correspond to training schemes based on soft decisions:

$$\Gamma_s = \left\{(v_i, l_i): i = 1, \cdots, N \quad and \quad l_i \subset [0,1]^L\right\} \tag{5}$$

The classifier is typically a non-linear map $F(\cdot)$ from the feature space $\{V_s, s = (i,j)\}$ to the points in the "fuzzy" cube $[0,1]^L$. Thus

$$F : V_s \in \Re^n \rightarrow D_s \in [0,1]^L \tag{6}$$

Where D_s is a real valued decision vector whose i th element shows the vote (or in fuzzy terms the fit value) associated with class i. The classifier has L non-binary outputs, one for each class, where each output takes values in $[0,1]$.

The recent success of neural networks and fuzzy systems in dealing with uncertainty, ambiguity and randomness through distributed soft decisions makes them good candidates for implementation of these ideas. Before making local soft decision, the K—L transform of features are carried out. After that, the dimension of features is compressed to 18. A feed-forward multi-layer neural network is used to implement local soft classification in this paper. The neural network classifier consists of eighteen input, twenty hidden, and three output units. The input units are linear, whereas the hidden and output units have sigmoid non-linearity. A conjugate gradient method is used for fast convergence of the supervised learning algorithm. The training set consists of 200, randomly selected samples to provide enough information for the learning algorithm to create soft decision boundaries. Using simple feature sets and classifiers, very good performance has been obtained. The method is also robust to noise and easy for fast implementation. Of course, good feature extraction is also important for its performance.

IV. Construction of basic probability mass and rules of decision

In Shafer-Dempster reasoning, there isn't a general formula for calculating basic probability mass. A proper basic probability mass must be constructed according to applications. Based on the fuzzy membership mentioned in section 3, we constructed a basic probability mass formula that is ready for nondestructive application.

The definition of the basic probability mass in our research is:

$$m_i(j)=\frac{\mu_i(j)}{\sum_j \mu_i(j)+n_s(1-r_i)(1-\alpha_i\beta_i\omega_i)} \quad (7)$$

$$m_i(\Theta)=\frac{n_s(1-r_i)(1-\alpha_i\beta_i\omega_i)}{\sum_j \mu_i(j)+n_s(1-r_i)(1-\alpha_i\beta_i\omega_i)} \quad (8)$$

Where n_s is the number of transducers and $\mu_i(j)$ refers to the membership value that a signal collected from transducer i belongs to pattern class j. ω_i is the environmental coefficient of transducer i, which is determined by experts according to testing environment (temperature, humidity, field interference), its value within the range $[0,1]$. α_i is the maximum membership value of a signal collected from transducer i belonging to one possible pattern class ($\alpha_i=\max_j\{\mu_i(j)\}$) and β_i refers to the distribution coefficient of membership values ($\beta_i=\frac{\alpha_i}{\sum_j \mu_i(j)}$). r_i is the reliability coefficient of transducer i ($r_i=\frac{\omega_i\alpha_i\beta_i}{\sum_i \alpha_i\beta_i\omega_i}$).

$m_i(j)$ is the basic probability mass that a signal collected from transducer i designates to pattern class j and $m_i(\Theta)$ refers to the basic probability mass that a signal collected from transducer i designates to the frame of discernment Θ, i.e. the uncertainty probability of the signal collected from transducer i。

In the calculation of the reliability coefficient r of a transducer, not only the maximum membership value (α) and the distribution of membership values (β), but also the environmental coefficient (ω) are considered. In addition, $(\alpha\beta\omega)$ provides a parameter to represent the reliability between a signal and its membership value belonging to a pattern class. Second, when calculating the uncertainty probability mass of a signal, unreliable factor $(1-r_i)$ and unreliable factor between a signal and its membership value belonging to a pattern class $(1-\alpha_i\beta_i\omega_i)$ are considered。

Just as the construction of the basic probability mass, there isn't a universal method that can be used for making hard decisions after integration. Different methods are selected for different applications. Here, the rule-based method is used. According to the meaning of the basic probability mass, several rules are assigned as the criteria for making hard decisions. The main rules are four as follow:

Rule 1: The pattern class designated to detecting object must have maximum basic probability mass;

Rule 2: The difference between the basic probability mass of a designated pattern class and the basic probability mass of another pattern class must over a predetermined threshold;

Rule 3: The uncertainty probability mass must below a predetermined threshold;

Rule 4: The basic probability mass of the designated pattern class must larger than the uncertainty probability mass.

V. Flaw identification by decision-level

identity fusion

In the application of information fusion for flaw detection, three pattern classes, non-defect, delamination and void are selected. Here, same rules are used to making hard decision both for single detecting method and the identity fusion method in order to compare their results of classification. In our research, according to the steps mentioned in section 3 and section 4, the basic probability masses of flaw detection by ultrasonic test and by impact-echo test are got respectively first. Then, their local decisions can be made respectively. After that, hard decisions are performed by integrating soft local decision vectors to reduce their "ambiguities". The results of classification are shown in table 2 and table 3.

The evidence intervals $[spt(A), pls(A)]$ in table 2 and in table 3 are calculated from the basic probability masses. $spt(A)$ is equal to the minimal commitment to pattern class A, expressed the probability of proposition" A is true", called the lower bound of support for proposition A. $pls(A)$ is equal to the support plus any potential commitment, called the plausibility or the upper bound. Whereas the difference $(pls(A) - spt(A))$ between $spt(A)$ and $pls(A)$ expressed the unknown degree of the proposition. These bounds show what proportion of evidence is truly in support of a proposition, and what proportion results merely due to ignorance or the need to normalize to unity sum. This is important, for instance, if it is desired to know exactly what proportion of evidence directly implicates a particular pattern class.

From the tables, we learnt that the results of classification by integration of two detecting methods are obviously better than the results of classification by single detecting method. In the

Table 2 Results of flaw detection by decision-level identity fusion for big flaws

types of pattern class	detecting method	Probability interval $[spt(A), pls(A)]$			$m(\Theta)$	results of classification
		non-defect	delamination	void	Θ	
non-defect	Ultrasonic testing	0.618,0.734	0.146,0.262	0.120,0.236	0.116	non-defect
	Impact-echo testing	0.461,0.580	0.204,0.323	0.216,0.335	0.119	non-defect
	ID fusion	0.733,0.758	0.126,0.151	0.116,0.141	0.025	non-defect
delamination (200×200×1)	Ultrasonic testing	0.130,0.250	0.428,0.548	0.322,0.442	0.120	delamination
	Impact-echo testing	0.126,0.306	0.388,0.568	0.306,0.486	0.180	delamination
	ID fusion	0.098,0.137	0.518,0.557	0.345,0.384	0.039	delamination
void (φ50)	Ultrasonic testing	0.202,0.373	0.280,0.451	0.347,0.518	0.171	unknown
	Impact-echo testing	0.186,0.344	0.330,0.488	0.326,0.484	0.158	unknown
	ID fusion	0.186,0.236	0.354,0.404	0.410,0.460	0.050	void

Table 3 Results of flaw detection by decision-level identity fusion for small flaws

types of pattern class	detecting method	Probability interval $[spt(A), pls(A)]$			$m(\Theta)$	results of classification
		non-defect	delamination	void	Θ	
non-defect	Ultrasonic testing	0.542,0.670	0.172,0.300	0.158,0.286	0.128	non-defect
	Impact-echo testing	0.480,0.612	0.200,0.332	0.188,0.320	0.132	non-defect
	ID fusion	0.693,0.722	0.146,0.175	0.132,0.161	0.030	non-defect
delamination (50×50×1)	Ultrasonic testing	0.240,0.335	0.358,0.453	0.307,0.402	0.095	delamination
	Impact-echo testing	0.264,0.438	0.290,0.464	0.272,0.466	0.174	unknown
	ID fusion	0.259,0.292	0.385,0.418	0.323,0.356	0.033	delamination
void (φ30)	Ultrasonic testing	0.222,0.386	0.300,0.464	0.314,0.478	0.164	unknown
	Impact-echo testing	0.244,0.450	0.268,0.474	0.282,0.488	0.206	unknown
	ID fusion	0.250,0.310	0.333,0.393	0.357,0.417	0.060	void

application, when we use only one detecting method for classification, the detection of some flaws is failed. However, the results of classification by integration are all correct. Therefore, we can conclude that information fusion enhanced the detecting rate. From table 2 and table 3, we know that the values of $m(\Theta)$ have reduced obviously after integration. This indicates that the uncertainty of the system is reduced by the information fusion. At the meantime, the basic probability masses after integration have better separatability than the basic probability masses before integration, i.e. the ability of flaw identification enhanced. When we use same rules for classification, information fusion will greatly enhance the performance of the system. In other words, the basic probability mass constructed in this paper is correctly reflected the specificity of the transducers and are suitable for application on non-destructive test. Furthermore, if we can make full use of the specificity of transducers or extract better features with suitable signal processing method, then the detecting rate will be even higher by using information fusion.

VI. Conclusions

The application of decision-level identity fusion on flaw detection of concrete structures is studied in this paper. In order to apply Shafer-Dempster reasoning to perform decision-level identity fusion, the idea of local soft decision is investigated to give out the fuzzy membership by a feed-forward neural network. Based on the fuzzy membership, a kind of basic probability mass is constructed. For better flaw detection, Shafer-Dempster reasoning is used to integrate local decisions. The integrated results of classification are better than the results of classification by either of the two detecting methods. Following are the main work of the research:

(1) Soft local decision vectors that describe the relationship between a testing signal and a pattern class are provided;

(2) Feature vectors are classified locally by using a neural network to allow fuzzy membership assignment;

(3) A kind of basic probability mass is constructed for application on NDT;

(4) Rules for hard decisions are designed;

(5) Different types of flaws are used for detecting. Decision-level identity fusion is utilized to identify these flaws.

This work was supported by NSFC-69772001

References

[1] Yasunori Sakata and Masayasu Ohtsu, Crack Evaluation in Concrete Members Based on Ultrasonic Spectroscopy, ACI Materials Journal, V.92, No.6, November—December 1995, pp686—699.

[2] Tian Huisheng, Cao Xiuguo, Li Yanbing, Research on application of spectrum analysis to ultrasonic testing of concrete quality, Journal of Xi An jiaotong university, V.31, No.12,1997, pp14-19

[3] Lin Weizheng, Yuan Yiyong, Ultrasonic detection of cement concrete pavement thickness,14th World Conference on Nondestructive Testing (14th WCNDT) , New delhi, India, 8-13, 1996, pp153-156

[4] C.H.Chen, Tzu-Hung cheng, Time-frequency analysis in Ultrasonic Nondestructive testing, 14th World Conference on Nondestructive Testing (14th WCNDT) , New delhi, India, 8-13, 1996, pp161-168

[5] Yan Xuaheng, Lin Weizheng, Wavelet analysis of ultrasonic testing signal for detecting thickness of concrete, Journal of Non-destructive test, V.19, No.8,1997, pp211-213

[6] Lin Y., Sansalone M., Detecting flaws in concrete beams and columns using the impact-echo method, Materials Journal of the American Concrete Institute, July-August 1992, pp394- 405.

[7] Jiunn—Ming Lin and Mary Sansalone, Impact—Echo Studies of Interfacial Bond Quality in Concrete: Part I —Effects of unbounded Fraction of Area, ACI Materials Journal, V.93, No.3, May—June 1996, pp223—232.

[8] Jiunn—Ming Lin, Mary Sansalone, and Randall Poston, Impact—Echo Studies of Interfacial Bond Quality in Concrete: Part II —Effects of Bond Tensile Strength, ACI Materials Journal, V. 93, No.4 July—August 1996, pp318—326.

[9] Kamran Etemad, David Doermann, Rama chellappa, Multi-scale segmentation of unstructured document pages using soft decision integration, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 19, No.1, 1997,pp92-96

Integrating Different Conceptualizations for Heterogeneous Knowledge

K. Christoph Ranze
Center for Computing Technologies
Bremen University, P.O.B. 33 04 40
28334 Bremen, Germany
kcr@tzi.de

Abstract *In this paper we argue that the model-based development of knowledge-based systems built to work in partially uncertain domains benefits from the fusion of different conceptualizations for certain and uncertain parts of the required knowledge. We present conceptualizations that have proven to be useful, namely the KADS model of expertise and a causal model of uncertainty that reflects well known approaches to uncertain reasoning like Bayesian belief nets. We propose an extension of existing specification languages that aims at an integration of these conceptualizations in a common knowledge model. We present parts of the analysis and specification of a rock classification problem as an example demonstrating the demand for the combination of different conceptualizations.*

Keywords: knowledge engineering, conceptualization, uncertainty, rock classification

1 Introduction

In real-world applications adjectives like 'probable', 'possible' or 'incomplete' are attached to domain knowledge and data. We summarize these phenomena of non-categorical knowledge as 'uncertainty'. Having recognized that uncertainty plays an important role in the development of knowledge-based systems we have to find ways to deal with different kinds of uncertainty when building knowledge models. Investigating different model-based knowledge engineering approaches we found no sophisticated formalism for the explicit representation of uncertainty in any of their semi-formal or formal knowledge models.

The problem is not that there are no ways to deal with uncertain knowledge. There is a huge amount of elaborated (numerical) calculi for the representation and processing of uncertain knowledge in application systems.

So what is the real problem that prevents notions of uncertainty to be integrated in existing knowledge engineering approaches? In our opinion the problem is that existing approaches for handling uncertainty follow a conceptualization used to describe a knowledge domain that is completely different from the one used in common knowledge engineering approaches.

Our previous work in the field of knowledge engineering aimed at bridging this gap and resulted in a model of uncertain expertise (ModE-U) [1, 2]. ModE-U is the core of a methodology for the analysis, conceptualization and formalization of certain and uncertain knowledge. It has been developed as an extension of existing knowledge engineering approaches to cover those parts of human expertise that cannot be adequately formulated using ordinary first-order logic.

2 Problem Statement

As an example for the need of such a combined approach we use a slightly modified version of the Sisyphus III problem [3]. The Sisyphus experiments of the knowledge acquisition community are an attempt of comparing and eva-

Table 1: Mineral content relation QAPF

	Mineral content in %			
Rock class	quartz	alkali	plagi.	feld.
granite	20..60	35..90	10..65	0
syenite	0..5	65..90	10..35	0
diorite	0..5	0..10	70..90	0

luating different approaches used for the construction of knowledge based systems (KBS). The task of Sisyphus III is to build knowledge models to be used as a specification of a KBS solving the problem of classifying igneous rocks. Figure 1 shows a snapshot of the classification task from the Sisyphus III domain.

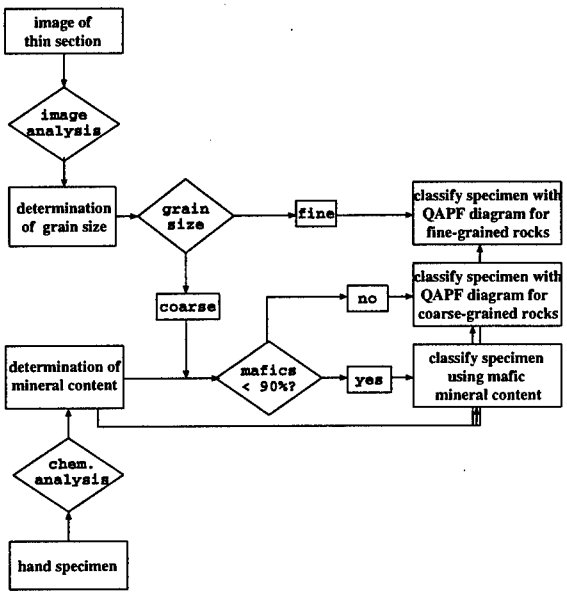


Figure 1: Classifying igneous rocks

In our example the rock class of a hand specimen can be determined through predefined classification schemes. These schemes are provided by so-called Streckeisen diagrams [4] that allow for a classification of a certain subgroup of igneous rocks using knowledge about their mineral contents. Table 1 shows a set of mineral content relations (which have been extracted from the diagrams) serving as associations for classifying a hand specimen.

The selection of an appropriate scheme depends on the specimens grain size which can be estimated from a digital image of a thin section of the specimen using common image analysis techniques. Figure 2 depicts the connection between visual properties and the grain size.

Determining the knowledge in use, we found that the subtasks of the problem-solving process are attached to different kinds of certain and uncertain knowledge.

As extracted from Streckeisen diagrams there is no uncertainty attached to the mineral content relations beyond their interval-based nature. Therefore these relations would be conceptualized as certain knowledge with almost no dissent.

On the other hand, the knowledge used to determine the grain size has a highly heuristic nature and is based on the (more or less unknown) causal relations between the grain size of the specimen and the image features of its thin section. To compute the unknown grain size from the given image features the other way round we have to perform diagnostic reasoning using these heuristics.

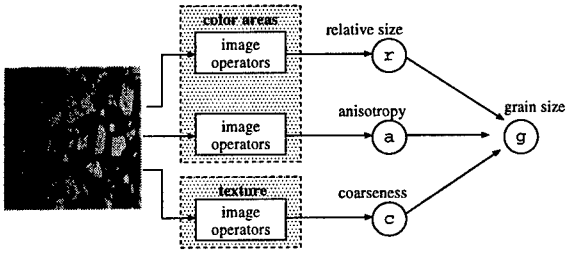


Figure 2: Determining grain size through image analysis

In our work we have conceptualized the different types of certain and uncertain knowledge within the different parts of our common knowledge model.

Specifying a KBS for the given problem we have to combine the classification task which is based on the mineral content relation with the diagnosis of the grain size based on the heuristics mentioned above. In that sense our work presents a good example for the fusion of certain and uncertain diagnostic/classification knowledge.

2.1 The Spirit of Heterogeneity

We argue that if it is necessary to deal with uncertainty in complex domains one has to bridge the gap between the different conceptualizations.

On one hand we need a rather simple conceptalization for models of uncertainty to enable uncertain reasoning. In our example we are utilizing uncertain reasoning techniques for estimating the grain size with respect to the image features. However, from our point of view the use of raw conceptualizations provided by these techniques should be restricted to a minimum.

On the other hand we are not willing to give up the more elaborate conceptualization of models of expertise that has been proven to be useful for analysis, model building, and reuse. Regarding our example we try to cover as much knowledge as possible within the structures of a common model of expertise.

3 Different Conceptualizations

The basic idea of our approach lies in the extension of a common model of expertise wich incorporates explicit notions for uncertain knowledge items. Therefore we use the well known KADS approach [5] and its specification languages *CML2* [6] and $(ML)^2$ [7]. We proposed extensions of these languages ($CML2^{unc}$ and FLUE) which allow for the integration of uncertainty in the different informal, semiformal, and formal levels of model-based knowledge engineering. These extensions cover static aspects of the domain knowledge as well as dynamic ones of the problem-solving process.

In the following section we give an overview of the basic concepts of our modeling approach. Further details are given in [1, 2].

3.1 Conceptualization of Expertise

Conceptualizations of expertise knowledge are typically subdivided into three kinds of knowledge: *domain, inference, and task knowledge* as defined in the KADS model of expertise [5]. Subsequently we are describing this conceptualization on an abstract level.

Those parts of the real world relevant to the given task are described through their properties within the domain model. The formal specification of this knowledge is realized by a set of ontological primitives enabling the user to define complex structures: *concepts*, *instances* of these concepts, *attributes* of and *relations* between concepts.

Based on the modeled elements there are *inference actions* performing single steps of the problem-solving process. Inference actions operate on elements from the domain layer. These elements are described through *input-*, *output-* and *static roles* which are placeholders determining the role the element plays in the problem-solving process and the type of domain objects that can play this role.

The task layer contains knowledge about how the elementary inference steps can be combined and executed to achieve a certain goal. This knowledge is organized in *tasks* which can be decomposed into *subtasks* including control knowledge about their execution in order to achieve the *goal* of the main task. Primitive tasks without any subtasks have a one-to-one correspondence to *knowledge sources* within the inference layer.

Together these three layers form a model of expertise that claims to capture all aspects of expert reasoning relevant to the development of knowledge-based systems. A common model is achieved by connecting the different layers in the sense that the roles of inferences are filled with domain knowledge. Tasks are executed by applying inferences which produce a result corresponding to the task's goal.

Within our approach we use these concepts to specify the structural and certain knowledge on the semi-formal level (CML2) as well as on the formal level (ML^2). On one hand the integration of uncertainty is realized through a direct extension of the different concepts of CML2. On the other hand we propose standalone models of uncertainty on the formal level

which are realized through self-contained theories following an extended ML^2 notion.

3.2 Conceptualizing Uncertainty

One of the main ideas of our approach is the integration of different types of uncertain knowledge into model-based specification languages for building KBS. Due to the different levels of specification two related conceptualization of uncertain knowledge can be indentified in our approach.

The syntactical integration is based on the differentiation of several valuation[1] categories:

- numerical valuations (like Bayes probabilities)
- interval based valuations (like Dempster/ Shafer)
- user-defined terms (like fuzzy sets)

Based on these categories we are now able to define a basic ontology for the valuation of uncertainty on the semi-formal level. Together with the basic ontology of CML2 this ontology forms the underlying basis for our language *CML2^{unc}* which is able to cover different phenomena of uncertainty within one single specification.

The following example shows the *CML2^{unc}* specification of the local image feature *anisotropy* (anisotropy per region) and the globalized parameter *globalized-anisotropy* (normalized anisotropy). The uncertainty stemming from the application of a specific image operator is represented informally in terms of fuzzy membership functions.

```
concept anisotropy;
  description: "Relation between horizontal
    and vertical extension of an image region";
  sub-type-of: local-feature;
  properties:
    value: number-range(0,1);
  membership-functions:
    function-value-oval:
      if value > 0.7 then value
        (globalized-anisotropy) = oval[0]
      else if 0.3 <= value <= 0.7 then
        oval[-2.5*(value-0.3)+1]
      else if value < 0.3 then oval[1]
      end if;
    function-value-round:
      if value < 0.3 then value
        (globalized-anisotropy) = round[0]
      else if 0.3 <= value <= 0.7 then
        value (globalized-anisotropy)
        = round[(2.5*x)-0.75]
      else if value > 0.7 then value
        (globalized-anisotropy) = round[1]
      end if;
end concept anisotropy;
```

The required connection between the local image feature and the grain size of the specimen (see fig. 2) is realized through a normalization. Again, we make use of semi-formal specifications for the fuzzy membership functions.

```
concept globalized-anisotropy;
  description: "Globalized relation between
    horizontal and vertical extension of all
    regions of an image";
  sub-type-of: global-feature;
  properties:
    value: {oval, round} :: fuzzy-val;
  membership-functions:
    function-grainsize-coarse: ...
    function-grainsize-fine: ...
end concept globalized-anisotropy;
```

In the same sense we extended the dynamic parts of CML2. In previous work we proposed uncertain knowledge roles as well as special inferences working on uncertain domain knowledge items.

By using these enriched specification techniques we are able to cover all relevant aspects of the certain and uncertain domain knowledge required for the problem-solving process in early stages of a KBS development.

3.2.1 Uncertainty on the Formal Level

As proposed in [1] the conceptualization of uncertainty on the formal level consists of

[1] A valuation attaches a degree of certainty or truth to configurations of variables/statements.

three basic concepts, which are derived from Shenoy's valuation based systems (VBS) [8] and Pearl's model of causality [9]:

(1) A set of hypotheses is a variable, whose values denote different hypotheses concerning the same assertion. The hypotheses are assumed to be conflicting in the sense that only one of the hypotheses can be true at a time. Variables are denoted by small letters. If v is a variable then W_v represents the set of all possible values for v.

(2) A valuation function [8] attaches a degree of certainty taken from a set of truth values denoted as Ψ to configurations of hypotheses. Valuation functions are denoted by capitals corresponding to a valuated variable:

$$V : W_v \rightarrow \Psi \qquad (1)$$

A set of hypotheses and a valuation function over this set form a basic modeling element for uncertain domain knowledge which is denoted as *phenomenon of uncertainty*. A phenomenon of uncertainty UP is a pair consisting of a set W_v of hypotheses and a valuation function V on this set.

$$UP = (W_v, V) \qquad (2)$$

(3) Causal relations [9] are special valuation functions defined on different phenomena of uncertainty mapping one or more phenomena of uncertainty and a special value set indicating the strength of the causal influence on a target phenomenon. Such a causal relation determines the valuation function of the target phenomenon using the valuations of the source phenomena and the strength of the causal relation. Let $\mathcal{UP}$ be the set of all phenomena of uncertainty, then a causal relation is a function C defined as follows:

$$C : \mathcal{UP}^n \times \Psi \rightarrow \mathcal{UP} \qquad (3)$$

This conceptualization can be used to describe different calculi for handling uncertainty in a graph-based setting [10] and therefore provides a useful approach for the specification of uncertain knowledge.

4 Heterogeneous Specification on the Formal Level

For the formal model of uncertainty described above to be used for the specification of uncertainty in a problem-solving process, the reasoning process has to be described formally. This can be done by using an extension of an existing specification language.

4.1 Integrating Different Conceptualizations

We assume that uncertainty is present in the model of expertise (see chapter 3.1) in the sense that terminological knowledge is available, but some assertions cannot be determined due to uncertainty. These gaps in the model can be bridged in a three step approach using the formal model of uncertainty.

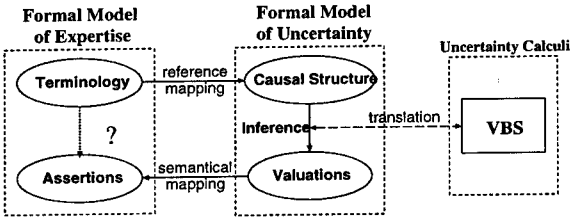


Figure 3: Integrating the models

Determination of the language to be used in the formal model of uncertainty is the first step. For this purpose, the different phenomena of uncertainty are explicitly connected to terminological constructs in the model of expertise through reference mappings. The referred concepts provide a language to describe the meaning of the phenomena.

Uncertain inference is the second step in which inference is carried out within the model

of expertise deriving valuations of previously unknown phenomena. Almost all existing calculi for processing uncertainty for which a description in terms of a VBS is available can be used for this purpose, because inference in the model is equivalent with inference in VBS.

In previous evaluations we tested triangular norms as a more general approch for an implementation of the corresponding inference algorithm [2].

Determination of assertional knowledge is the last step that uses the results of uncertain inferences to state axioms about knowledge to be used in the problem-solving process. This step depends on a semantical mapping determining the meaning of elements in the model of uncertainty in relation to the model of expertise.

4.2 Integrating the Dynamics

Following the integration concept mentioned above figure 4 gives an insight into the connection of the different inference actions used to formalize the overall problem-solving process.

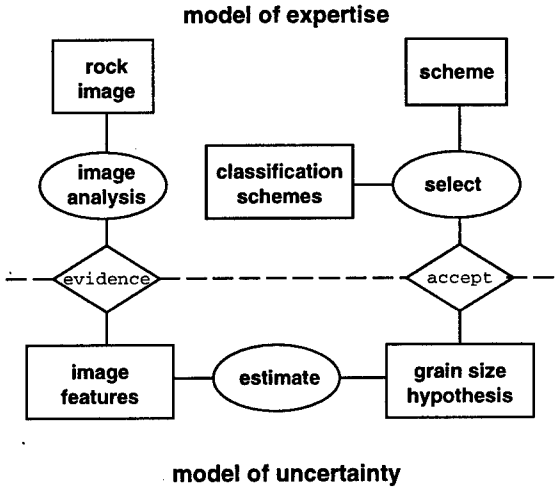


Figure 4: Integrated inference scheme

The transition from the image analysis to the estimation of the grain size is achived by an evidence that determines a valuation function over the image features. The result of the uncertain reasoning process (in our example a valuation of the hypotheses coarse/fine) is handed over to the certain model for determining the appropriate classification diagram by using an acceptance criterion [11].

5 Specification Language FLUE

The interaction described above has been used to develop a specification language for uncertain expertise, FLUE (Formal Language for Uncertain Expertise). To integrate uncertainty aspects smoothly into the existing parts of the language $(ML)^2$ [7], a textual description of our formal model of uncertainty has been developed which is based on the semantical mapping between this model and $(ML)^2$ theories. These theories consist of a signature describing the language used and a set of axioms. The overall structure of a specification is build up via import-relations.

The language FLUE adopts this scheme. Each phenomenon of uncertainty is specified in its own theory. The signature of this theory is given by concepts from the model of expertise the phenomenon refers to. Instead of logical sentences, the set of axioms defines the valuation function over the hypothesis space of the phenomenon and the structure equation determining the phenomenon. The connection to the relevant context, which is a parameter of the structure equation is established through import-relations as used in $(ML)^2$.

5.1 Formal Model of Grain Size Determination

The specifications given below describe a formal model of uncertainty for the problem of grain size determination from visual features using FLUE. It identifies relevant features, captures them in a causal structure and relates them with knowledge from the domain model.

To establish a formal model of uncertainty for the determination for the grain size for a rock specimen, the visual features on which

this process is based have to be specified. They are represented as sets containing qualitative descriptions of the possible results of the image analysis.

The first feature represented is anisotropy. The specification refers to the attribute of the same name that belongs to an image. The corresponding concept description is 'snapshot'. The set of hypotheses connected to anisotropy contains the elements *oval* and *round*. The valuations for these hypotheses are generated by a normalization over the anisotropy of all regions contained in the analyzed image. The use of a normalization operator is denoted in the axioms that describe the valuation function.

uncertain-domain-module anisotropy
  import normalize
  type simple
  signature
    hypotheses round, oval
    object $sample - picture : snapshot$
    link $has - global - anisotropy : image \rightarrow [0,1]$
  axioms
    $cert(anisotropy = round) = evidence(normalize, anisotropy, round)$
    $cert(anisotropy = oval) = evidence(normalize, anisotropy, oval)$
end-uncertain-domain-module

The next module defines a primitive inference action that can be used to calculate valuations for the different hypotheses concerning the grain-size of the hand specimen.

uncertain-pia estimate grain-size
  input anisotropy, relative-size, coarseness
  result grain-size
  assume max
    hypotheses coarse-grained, fine-grained
  signature
    pia-predicate determine-grain-size
    object $specimen : rock$
    link $grain - size : rock \rightarrow \{coarse - grained, fine - grained\}$
  axioms
end-uncertain-pia

The result of this calculation has to be interpreted and integrated into the certain model for further reasoning. Special assumption-inferences are used for this purpose, which apply an acceptance criterion to the calculated valuations. In our case we simply use a maximum operator which is accepting the hypotheses with the highest valuation to reflect the true state of the world.

6 Conclusions

The work done so far focussed on the development of semi-formal and formal models and specification languages for uncertain knowledge items and their relations to existing models of expertise. The results allow for a complete analysis and conceptualization of heterogeneous problem-solving knowledge regarding both certain and uncertain knowledge types. From our point of view the advantage of our approach lies in the separation and - simultaneous - integration of different knowledge types. To draw a conclusion from the example we are able to specify the processing of uncertain input from image data utilizing simple knowledge structures while being able to represent existing classification schemes in a highly structured model of expertise.

Furthermore, the introduced methods are offering the ability to apply the elaborated principles of established knowledge engineering approaches (reuse, knowledge level modelling, knowledge typing) to different kinds of uncertain knowledge.

The approach gives implications to further research. One of the most challenging one is the operationalization of uncertain expertise, which has been investigated very superficial, so far.

Acknowledgement

This paper is the result of several discussions concerning the evaluation of model-based knowledge engineering within our group. Support for this work and helpful comments were kindly provided by Heiner Stuckenschmidt and Jan Stanger.

References

[1] K. C. Ranze and H. Stuckenschmidt. Modelling uncertainty in expertise. In Jose Cuena, editor, *IT & KNOWS Information Technologies and Knowledge Systems, Proceedings of the XV. IFIP World Computer Congress*, Serial Publication of the Austrian Computer Society, pages 105–118, Vienna/Budapest, September 1998.

[2] K. C. Ranze and H. Stuckenschmidt. Bridging gaps in models of expertise. In J"urgen Dix and Steffen H"olldobler, editors, *Inference Mechanisms in Knowledge-Based Systems: Theory and Applications*, volume 19.98 of *Fachberichte Informatik*, pages 33–59. Universit"at Koblenz-Landau, September 1998.

[3] U. Gappa and F. Puppe. A study of knowledge acquisition - experiences from the Sisyphus III experiment for rock classification. In *Proceedings of the 11th Workshop on Knowlegde Acquisition for Knowledge-Based Systems*, Banff, Alberta Canada, 1998.

[4] A. Streckeisen. Classification and nomenclature of volcanic rocks, lamprophyres, carbonatites and melilitic rocks: Recommendations and suggestions of the IUGS subcommission on the systematics of igneous rocks. *Geology*, 7:331–335, 1979.

[5] Guus Schreiber, Bob Wielinga, and Joost Breuker, editors. *KADS: A Principled Approach to Knowledge-Based Systems Development*, volume 11 of *Knowledge-Based Systems*. Academic Press, 1993.

[6] A. Th. Schreiber, B. Wielinga, H. Akkermanns, W. van de Velde, and A. Anjewierden. CML: The CommonKADS conceptual modelling language. In L. Steels et al., editor, *A Future of Knowledge Acquisition, Proceedings of the 8th European Knowledge Acquisition Workshop (EKAW '94), Hoegaarden*, Lecture Notes in Artificial Intelligence 867. Springer-Verlag, Berlin, 1994.

[7] F. van Harmelen and J. R. Balder. $(ML)^2$: a formal language for KADS models of expertise. *Knowledge Acquisition Journal*, 4(1), 1992. Special issue: 'The KADS approach to knowledge engineering', reprinted in KADS: A Principled Approach to Knowledge-Based System Development, 1993, Schreiber, A.Th. et al. (eds.).

[8] P. P. Shenoy. Valuation-based systems: A framework for managing uncertainty in expert systems. In L. A. Zadeh and J. Kacprzyk, editors, *Fuzzy Logic for the Management of Uncertainty*. Wiley and Sons, 1989.

[9] J. Pearl. Structural and probabilistic causality. In D. R. Shanks, K. J. Holyoak, and D. L. Medin, editors, *The Psychology of Learning and Motivation*, volume 34: Causal Learning, pages 393–435. Academic Press, San Diego, CA, 1996.

[10] A. Saffiotti and E. Umkehrer. PULCINELLA - a general tool for propagating uncertainty in valuation networks. In *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence*, pages 323 – 331, Los Angeles, CA, 1991.

[11] H. E. Kyburg. Probabilistic acceptance. In Dan Geiger and Prakash Shenoy, editors, *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, San Francisco, 1997. Morgan Kaufmann Publishers.

Session TC3
Fusion of Fuzzy Information
Chair: Daniel McMichael
University of South Australia, Australia

Predictive neural networks and fuzzy data fusion for on-line and real-time vehicle detection.

E. JOUSEAU
INT Dept EPH
9, rue Charles FOURIER
91011 Evry FRANCE

B. DORIZZI
INT Dept EPH
9, rue Charles FOURIER
91011 Evry FRANCE

Abstract – *This article describes an on-line and real-time vehicle detection system. This system detects vehicles passing over magnetic sensors. It works independently of their initial position and of strong magnetic disturbance possibly induced by the load carried on the vehicles. This system is based on the co-operation between a reflective agent, using a reliability measure of its output, and a detection agent (on which this article mainly focus) based on two predictive neural networks and model selection techniques. The fusion of the data delivered by each agent is obtained through fuzzy logic rules. The system is also strengthened to resist substantial magnetic disturbances (even non-periodic ones); it uses the three components of the magnetic field, and is rotational invariant. Furthermore, its modular design opens up many possibilities of evolution.*

Keywords : Fuzzy logic, predictive neural networks, data fusion, real-time detection, on-line detection.

1 - Introduction

In this article we partly describe a system based on the architecture previously introduced by F. Smieja in 1996 and modified by ourselves in 1998 (cf. [Sm96] and [Jo98]). We will mainly focus on three points. The first one is the introduction of predictive neural networks in the detection process. The second point is the adaptation of the fusion techniques to enable the system to manage these heterogeneous data. The last point is the comparison between the results obtained by the first system (presented in [Jo98]) and by the new one.

2 – Global architecture of the system

The architecture of the full vehicle detection system is pictured in figure 1. The main principles of this system are that the division in the input space is made at a symbolic level by the sub-tasks separation and the fusion is made with fuzzy rules.

At each sampling instant, the sensors give 3 measurements (one for each dimension of the magnetic field), and the information go through the entire system so that the detection decision can be estimated.

The results of the preprocessing operations are all independent of the terrestrial magnetic field. It is obvious that the geographical position of the measurement has some influence on the magnetic properties of the vehicle and consequently on the disturbance it generates (because of the induced part of it). The different parameters calculated during the preprocessing and used by the rest of the system are described in [Jo98], they are geometrical parameters such as norm, radius of curvature, angular displacement, etc. Furthermore, all of them are rotational

invariant. As shown on figure 1, they are both inputs of the different detection modules and context parameters for fusion and detection.

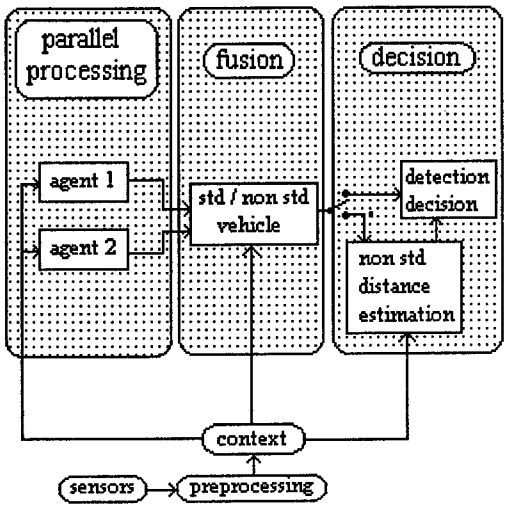


Figure 1 : general architecture of the system

The two detection agents are different and each one is dedicated to a particular subtask.

The first one is dedicated to the middle third of standard vehicles detection. It is composed of two predictive neural networks (Multi Layer Perceptrons) and a function of error estimation. It is on this module that we will focus our presentation (it is the major change of the system presented previously in [Jo98]). The main reason to change the first agent into a predictive neural network based one, is that, considering the results of the first system and the global approach we had, it seemed to us that the major inconvenient of our system was its lack of global temporal view. Compared to a classical classification neural network; a predictive one allowed us to take into account much more global temporal phenomena of the problem such as magnetic vector trajectory shape etc...

The second agent is detailed in [Jo98], it is dedicated to the non-standard vehicles approach detection (i.e. strong magnetic disturbances carriers). It is composed of a neural network (also a M.L.P.) trained with a specific detection function and an other M.L.P. which aim is to estimate the confidence that can be put on the output of the first one.

The fusion (also detailed in [Jo98]), and the detection decision are made with fuzzy logic rules based on the outputs of the two detection modules and some contextual parameters coming from the preprocessing. These parameters are also the inputs of the different neural networks of the two detection modules.

3 – First magnetic detection module : standard vehicles

For the standard vehicles detection, we used predictive neural networks. These networks modelize the magnetic field disturbances generated by a vehicle coming nearby the sensors. For this reason, we limited this approach to the standard vehicle detection problem. Actually, the non-standard vehicles have *a priori* not known characteristics as far as the close field magnetic disturbance is concerned. Consequently, a modeling approach for these vehicles seemed hopeless.

We did chose a solution with two neural networks. The first one is trained with samples of vehicles passing above the sensors, and the second one with samples of vehicles passing nearby. Both of them have been trained only with standard vehicles.

The discrimination principle between those two kind of passages is as follows : a passage to be classified is presented to each network, the class given to this passage will be the one of the network which has produced the weakest error. We obviously defined an *ad hoc* error criterion. The definition of this criterion (based on moving windows and cumulated errors) is also an original part of our work.

The parameters of such networks and

those needed for the subsequent competition fusion are difficult to estimate. We proceeded in three steps. During the first step, we designed the predictive neural networks for standard vehicles detection for off-line detection. This way, we reduced the problem to the detection of the standard vehicles middle third, knowing their complete signatures. This very hard restriction on the problem constraints allowed us to verify some of our hypotheses and to set some parameters of the system. During the second step, we optimized the other parameters of the predictive neural networks considering the on-line detection problem. We have finally integrated the predictive neural network in the global detection system with suited fuzzy logic rules.

The predictive neural networks are used more to characterize the temporal shape of the different signals received by the sensors (to discriminate between the different kinds of passages) than to produce an excellent estimation of these signals. Nevertheless, we trained each neural network for prediction and their discrimination power obviously depends on the quality of their predictions.

The predictive neural networks that we used are Multi Layer Perceptrons. Their particularities is that the desired output they are trained with, is a future value of one of their inputs. The input we have chosen for prediction is the norm of the observed magnetic disturbance : this parameter seemed to be the most informative of all.

The learning base has been separated in two parts: an "over" part (with vehicles passing over the sensors) and an "aside" part (with the others). The "over" and "aside" passages have been presented respectively only to the "over" and "aside" network.

In our application, we must have "real-time" and "on-line" detection. Two major problems occur with on-line detection. First we have to find a way to discriminate in an efficient enough way the "aside" passages from the "over" passages so that this discrimination could happen before the middle third of the vehicle. Secondly, we have to detect the middle third of the vehicle in the same time.

Consequently, achieving this discrimination leads to define and then optimize a great deal of parameters. While the off-line discrimination is an easier problem, we prefered to test and tune some of our parameters on this problem. That showed us some limitations and possibilities of our approach. For these reasons, we designed an error analysis algorithm for the two predictive neural networks based on moving windows and weightings ; first we tested it on the off-line problem and then adapted it on the on-line processing.

4 – Competition principle : the moving windows

The error analysis for each neural network on a passage (in the off-line problem) is pictured in figure 2 and can be summarized like this :

During the first step, we stock the observed norm during the whole passage and the corresponding outputs of each network (i.e. the predicted values of this norm estimated by each network) on the same passage.

During the second step, we calculate the error of each network according to a certain number of delays and advances (these delays and advances are chosen close to the value that was set for the networks learning).

During the third and last step the class ("over" or "aside") is decided to be the one of the network that made the less error in one of those cases.

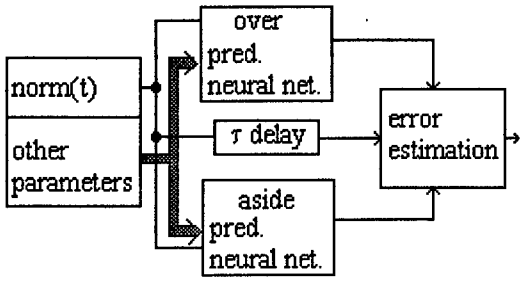


Figure 2 : error estimation architecture.

In fact, the predictions of the networks are often very good (according to the shape of the predicted curve), but a certain offset (in time) often subsist. Because of this behavior, a single value of the network error is not representative enough of the network prediction quality. So, if we call τ_0 the theoretical temporal delay between the observed norm $n_{obs}(t)$ and the predicted norm ($\tilde{n}_{over}(t)$ or $\tilde{n}_{aside}(t)$ depending on which predictive neural network is concerned), the relation we hope after learning is , in case of an aside passage :

$$n_{obs}(t + \tau_0) = \tilde{n}_{aside}(t) + \varepsilon_{as}(t)$$

and

$$n_{obs}(t + \tau_0) = \tilde{n}_{over}(t) + \varepsilon_{ov}(t)$$

with :

$$\varepsilon_{as} < \varepsilon_{ov}$$

But, due to this phenomenon of delay offset, we not only observe the difference between ε_{as} and ε_{ov}, but between $\varepsilon_{as}(\tau)$ and $\varepsilon_{ov}(\tau)$, where τ is a variable close to τ_0.

The value of τ_0 was also a difficult parameter to choose. Its choice is fundamental for the system. There are two limitations for its value :

The upper limit is due to the delay that is consequently introduce for the final detection decision. Effectively, to calculate ε_{as} and ε_{ov}, we need the value of $n_{obs}(t + \tau_0)$. Furthermore, the very beginning of each vehicle signature is the same and so, non-informative. If we assume t_b the duration of the non-informative part of each signature, the total delay to wait to have significant values of the errors is $t_b + \tau_0$.

The lower limit for τ_0 has two origins. Firstly, a very low value is impossible because of the sampling frequency used in the system and the need for a real-time system. Secondly, a low value means a short prediction horizon. If the prediction horizon is too short, the best prediction is quite always the linear prediction whatever the kind of passage. So the discrimination should become impossible (the two networks will make quite the same error).

5 – Weighting of the errors

After a brief analysis of the outputs of the predictive neural networks, it seemed that the overestimation errors should not be treated in the same way than the underestimation errors. There is a physical explanation to that observation :

The signatures of the "over" passages are, by nature, more "agitated" than the ones of the "aside" passages. In the three-dimension space, they present more direction changes and their norms are quite bigger.

Due to this matter of fact, the errors of the "over" network when facing an "aside" passage tend to be generally overestimation errors. In the opposite, the errors of the "aside" network, when facing an "over" passage, tend to be generally underestimation errors.

To take benefit from this behavior, we decided to take more into account the overestimation errors of the "over" network and the underestimation errors of the "aside" network. This is not a classical method for error processing in prediction, it comes from our need to better characterize the shape of the prediction compared to the shape of the observation rather than to seek a perfect prediction.

Formally, to take differently the different errors into account, we used different weights to calculate the errors depending on their sign.

6 – On-line processing

The major problem of the on-line processing is that the total signature is not known when the detection should happen. So, assume that t_c is the considered sampling instant, $\vec{B}(t)$ is the magnetic field vector, $n_{obs}(t)$ its norm, $\tilde{n}_{aside}(t)$ is the estimation of this norm made by the "aside" network and $\tilde{n}_{over}(t)$ is the estimation of $n_{obs}(t + \tau_0)$ made by the "over" network, we calculate the two error functions of respectively the "over" and "aside" network $E_{over}(\tau, t_c)$ and $E_{aside}(\tau, t_c)$ like this :

$$E_{over}(\tau,t_c)= \frac{1}{\min(t_c,t_f(\tau))-t_0(\tau)} \sum_{t=t_0(\tau)}^{t=\min(t_c,t_f(\tau))} c_h \left| n_{obs}(t-\tau)-\tilde{n}_{over}(t) \right|$$

$$E_{aside}(\tau,t_c)= \frac{1}{\min(t_c,t_f(\tau))-t_0(\tau)} \sum_{t=t_0(\tau)}^{t=\min(t_c,t_f(\tau))} c_b \left| n_{obs}(t-\tau)-\tilde{n}_{aside}(t) \right|$$

with

$$c_h = \alpha \quad if \quad n_{obs}(t-\tau) < \tilde{n}_{over}(t), \quad 1 \quad if \ not$$

and

$$c_b = \beta \quad if \quad n_{obs}(t-\tau) > \tilde{n}_{aside}(t), \quad 1 \quad if \ not$$

where $\alpha>1$ and $\beta>1$.

For each sample instant t_c, we chose the value of τ τ_v and τ_d which respectively give the smallest errors $E_{over}(\tau_v,t_c)$ and $E_{aside}(\tau_d,t_c)$. So, we obtain, for each network, a cumulated error, function of time, that we denote $E_{over}(t_c)$ and $E_{aside}(t_c)$.

While we have to take into account the temporal aspect of the signals, and not only the value at one instant, we define two functions $Mx(t_c)$ and $Mn(t_c)$. These functions depend on the cumulative sums of the difference between the two errors :

$$Mx(t_c) = \max_{t\in[t_0,t_c]} \left(\sum_{t=t_0}^{t_c} E_{aside}(t) - E_{over}(t) \right)$$

$$Mn(t_c) = \min_{t\in[t_0,t_c]} \left(\sum_{t=t_0}^{t_c} E_{aside}(t) - E_{over}(t) \right)$$

The detection can now simply be achieved with a couple of thresholds (Sx,Sn) with the first order logic rule :

"The middle third of a standard vehicle is detected to pass over the sensors when Mx(t)>Sx AND Mn(t)<Sn".

In fact, these functions have not been used with first order logic rules but even like this, it is remarkable that the detection almost always occurs during the middle third of the vehicle. The explanation of this behavior is certainly the delay τ_0, that prevent the system to detect anything two much early. Another influent point for this behavior is that the most informative part of the signature is roughly in the middle of the vehicle and the variance of the different vehicle speed is not very high.

7 – Fuzzy fusion

As the system still requires modularity and because we already made a fuzzy fusion module, the two functions Mx and Mn have not been used through first order logic rules but through fuzzy logic rules. The two functions are interpreted as possibilities value and one of the detection rule is : when Mx is high enough and Mn is law enough, then the passage of a standard vehicle midle third over the sensors is very possible.

Furthermore, we take into account the outputs of the other agent, which is specialized in non-standard vehicle and

owns a self confidence estimation.

Working with fuzzy logic rules also allowed us to work with symbolic contextual parameters in the same time. These parameters are a sort of contextual expert verification to prevent the system to do some very easily (with a little human expertise) avoidable mistakes.

8 – Results

The results are summarized in the following tables. They present the comparison of our new system (predictive neural networks) and the previous one (classical neural networks). These results are very satisfying in both cases because the average correct detection rate is over 80% for the standard vehicles. Moreover, as we noticed in the precedent section, the detection quite always occurs in the middle third of each vehicle.

These results have been obtained on a database containing approximately 500 vehicle signatures. We made a distinction for standard vehicles between "aside near" (when the vehicle passes closer than 50 cm to the sensors) and "aside far" passages for physical and industrial reasons. For "aside" passages, the good result is : no detection. The bad detection for the non-standard vehicles are detection that occurred not even under the vehicle.

The results of our new system are comparable to the first one. Contrary to the first system which is a little bit more efficient for "over" passages, the new one is better for "aside near" passages.

Table 1 : results for standard vehicles and over passages

	Detection	No detection
Predictive neural networks	89.24 %	10.76 %
Classical neural networks	95.62 %	4.38 %

Table 2 : results for standard vehicles and aside near passages

	Detection	No detection
Predictive neural networks	24.44 %	75.56 %
Classical neural networks	40.62 %	59.38 %

Table 3 : results for standard vehicles and aside far passages

	Detection	No detection
Predictive neural networks	1.55 %	98.45 %
Classical neural networks	2.33 %	97.67 %

Table 4 : results for non-standard vehicles and over passages

	Detection	No det.	Bad det.
Predictive neural networks	69.2 %	15.4 %	15.4 %
Classical neural networks	69.2 %	15.4 %	15.4 %

Table 5 : results for non-standard vehicles and aside passages

	Detection	No detection
Predictive neural	25 %	75 %

networks		
Classical neural networks	31.3 %	68.7 %

Concerning the quality of errors, the two systems share 75% of their errors. These 75% of common errors are made on either very near "aside" passages of big (magnetically speaking) vehicles or "over" passages of very light vehicles. This observation is in perfect agreement with the physic of the phenomena involved.

Concerning the possible evolutions of our system, we can try to optimize the architectures of the two predictive neural networks differently. For simplicity reasons, the current tests have been maid with networks of the same size. However, the modelization achieved by each network has certainly not the same complexity, so should certainly be the size of each one. Moreover, a more specific adaptation of the parameters of our fusion module is also to be done.

Furthermore, if we are able to use both systems at the same time, we can notice that 25 % of non common errors are a good potential for fusion of the two systems.

8 – References

[Sm96] F. Smieja: "The Pandemonium System of Reflective Agents", IEEE Trans. On Neural Networks, vol. n°1, pp 97-106, Jan. 1996.

[Jo98] E. Jouseau, B. Dorizzi: "Neural networks and fuzzy data fusion for on-line and real-time vehicle detection", 1st ICMMIF , Las-Vegas Ne, vol. n°2, pp 695-701, Jul. 1998.

Fusing Expert Knowledge and Information from Data with NEFCLASS

Detlef Nauck and Rudolf Kruse
University of Magdeburg, Faculty of Computer Science (FIN-IWS)
Universitaetsplatz 2, D-39106 Magdeburg, Germany
Tel: +49.391.67.12700, Fax: +49.391.67.12018
E-Mail: Detlef.Nauck@cs.uni-magdeburg.de, WWW: http://fuzzy.cs.uni-magdeburg.de/~nauck

Abstract *Information fusion refers to the acquisition, processing, and merging of information originating from multiple sources to provide a better insight and understanding of the phenomena under consideration. There are several levels of information fusion. Fusion may take place at the level of data acquisition, data pre-processing, data or knowledge representation, or at the model or decision making level. Some aspects of information fusion can be implemented by using NEFCLASS a neuro-fuzzy approach to learn fuzzy classifiers from data. Fuzzy rules provided by experts can be fused with rules obtained by a learning process. In this paper we present the information fusion capabilities of NEFCLASS-J – a JAVA-based implementation of the NEFCLASS approach.*

Keywords: classification, information fusion, knowledge revision, neuro-fuzzy system, rule induction

1 Introduction

Fuzzy systems can provide simple, inexpensive and interpretable solutions for data analysis problems [19, 20]. They can be created from expert knowledge in the form of fuzzy if-then rules or they can be created from data by learning. However, it is often important to find a combination of both ways. The idea is to use information provided by several different information sources. In this paper we consider human experts who formulate their knowledge in form of rules, and databases of sample data. We show how to fuse these different types of knowledge by using neuro-fuzzy methods and present experimental results that demonstrate the usefulness of our approach. In this paper we present a neuro-fuzzy approach that is able to fuse fuzzy rule sets, induce a rule base from data and revise a rule set in the light of training data.

Neuro-fuzzy systems are an important approach in learning in fuzzy systems. They use learning algorithms derived from neural network theory and apply them to fuzzy systems. Because interpretability is often considered to be a key feature of fuzzy systems, neuro-fuzzy approaches restrain their learning algorithms such that the semantics of the trained fuzzy system is not affected.

NEFCLASS is such a neuro-fuzzy approach that was developed for classification [7, 9]. It can induce fuzzy rules and fuzzy sets from data to create a fuzzy classifier. NEFCLASS can automatically determine the number of rules that are needed to cover a certain training data set. After training the membership functions, automatic pruning strategies try to reduce the number of rules and variables to make the classifier more interpretable.

We have recently completed a new version of the NEFCLASS model and implemented it in JAVA[15]. The tool contains several new learning techniques and can also handle missing values and symbolic data. In addition to just learning fuzzy rules from data, NEFCLASS can also fuse expert knowledge

with its current rule base at any time of the learning process, or later during application. NEFCLASS determines for each fuzzy rule in its rule base a performance value. This value is used to select rules for deletion if contradictions in the rule base must be solved, or if the number of rules is bounded. To revise a rule set in the light of training data the user can also influence, whether expert knowledge can be replaced by knowledge created during the learning process, or if rules entered by experts must remain, even when they have a lower performance value.

In this paper we first outline the NEFCLASS model and its rule induction algorithm. Then we discuss techniques to fuse expert knowledge given in the form of fuzzy rules with the fuzzy rules created from data. In Section 5 we show with the help of a small example how to revise prior knowledge and fuse it with information obtained from data.

2 NEFCLASS

NEFCLASS is a neuro-fuzzy approach to learning fuzzy classifiers from data using rules like

$$R_r: \quad \text{if } x_1 \text{ is } \mu_r^{(1)} \text{ and } \ldots \text{ and } x_n \text{ is } \mu_r^{(n)} \text{ then class } c_r,$$

where $\mu_r^{(i)} : X_i \to [0,1]$ is a fuzzy set that represents a linguistic value of a feature $x_i \in X_i$. We assume that each pattern $\mathbf{p} = (p_1, \ldots, p_n)$ belongs to one and only one class $c_j, j \in (1, \ldots, m)$, but it may not be possible to exactly determine that class.

NEFCLASS can create a classifier from a set of training data $\tilde{\mathcal{L}}$ that contains pairs $(\mathbf{p}, \mathbf{t})$, where $\mathbf{p} \in X_1 \times \ldots \times X_n$ is an input pattern and $\mathbf{t} \in [0,1]^m$ is a target pattern describing the classification of $\mathbf{p}$. If we know the class c_j of $\mathbf{p}$ exactly then $\mathbf{t} \in \{0,1\}^m$ holds with $t_j = 1$ and $t_k = 0$ for all $k \neq j$. If we do not know the class of $\mathbf{p}$ exactly, $\mathbf{t} \in [0,1]^m$ represents a fuzzy classification of $\mathbf{p}$.

The learning algorithm of NEFCLASS has two stages: structure learning and parameter learning. Rule (structure) learning is done by a variation of the approach by Wang and Mendel [17]. Each (metric) input feature is partitioned by a given number of initial fuzzy sets. This way the input space is partitioned and we can simply create antecedents for prospective rules by checking which areas of the input space contain data. This can be done by processing the training data once. An evaluation procedure then creates a rule base by assigning appropriate consequents (class labels) to the discovered antecedents and selects only a certain number of rules with good performance [7, 9, 14].

In parameter learning the fuzzy sets are tuned by a simple backpropagation-like procedure. The algorithm does not use gradient-descent, because the degree of fulfilment of a fuzzy rule is determined by the minimum and non-continuous membership function may be used. Instead a simple heuristics is used that results in shifting the fuzzy sets and in enlarging or reducing their support.

The main idea of NEFCLASS is to create readable fuzzy classifiers, by ensuring that fuzzy sets cannot be modified arbitrarily during learning. Constraints can be applied in order to make sure that the fuzzy set still match their linguistic labels after learning. In addition pruning strategies are used to reduce the number of rules and variables [8].

The most recent implementation of the NEFCLASS model is called NEFCLASS-J [13] and provides the following features:

- automatically determination of the number of fuzzy rules,
- training of triangular, trapezoidal, bell-shaped and discrete fuzzy sets,
- processing data with missing values,
- processing data that contains both numeric and symbolic attributes,
- automatic pruning strategies to reduce the rule base size,
- fusion of expert knowledge and knowledge obtained from data.

3 Learning in NEFCLASS

In this section we shortly discuss some features of the learning algorithm that computes a fuzzy rule base from a set of training data that may contain numeric and symbolic features and missing values. Due to lack of space we cannot print the complete algorithm, please refer to [6, 15, 12].

Rule learning starts by creating initial antecedents that contain only metric attributes using the Wang/Mendel procedure [17]. This means antecedents are created by selecting hyperboxes from a structured data space (structure-oriented approach [11], see Figure 1). If we encounter a missing value, any fuzzy set can be included in the antecedent for the corresponding variable. Therefore we create all combinations of fuzzy sets that are possible for the current training pattern.

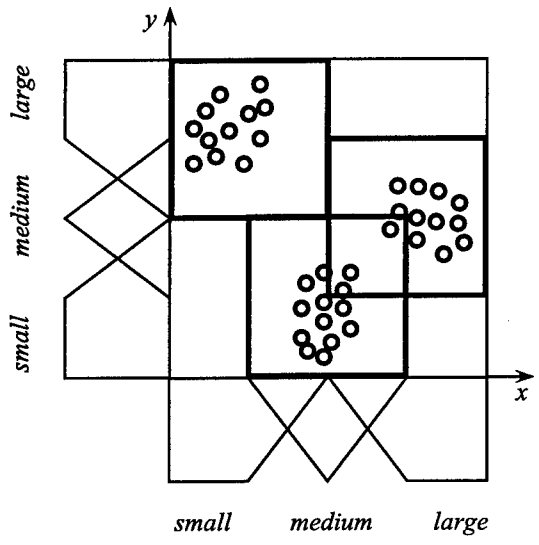


Figure 1: Learning fuzzy rules by selecting hyperboxes from a grid

If a feature value is missing, we do not make any assumptions about its value but assume that any value may be possible. We do not want to restrict the application of a fuzzy rule to a pattern with missing features. This means a missing value will not influence the computation of the degree of fulfilment of a rule. This can be done by assigning 1.0 as the degree of membership to the missing feature [1], i.e. a missing value has a degree of membership of 1.0 with any fuzzy set. A pattern where all features are missing would then fulfil any rule of the fuzzy rule base with a degree of 1.0, i.e. any class would be possible for such a pattern. After the training data is processed once, we have found all antecedents that are supported by the numeric features of the data.

Let us assume we have found k possible antecedents. If there are also symbolic attributes, we continue the rule generation process as follows. We create from each antecedent m rules, one for each class, and complete the initial antecedents by constructing fuzzy sets for the symbolic attributes. This is done by determining the relative frequencies of the attribute values [5]. This means we now have an initial rule base that contains a set of $m \cdot k$ rules. This rule set will usually be inconsistent, as it can contain contradictory rules. After resolving inconsistencies, by selecting from multiple rules with identical antecedents but different consequents the rule with better performance, a final rule base can be created.

If there are no symbolic attributes, we compute for each antecedent that is found in the data a consequent to generate complete rules. We select each consequent such that each complete rule causes a minimal number of errors. This is done by computing for each rule the following performance measure:

$$P_R = \frac{1}{s} \sum_{(\mathbf{p},\mathbf{t}) \in \tilde{\mathcal{L}}} (-1)^c R(\mathbf{p}), \text{ with}$$

$$c = \begin{cases} 0 & \text{if class}(\mathbf{p}) = \text{con}(R) \\ 1 & \text{otherwise} \end{cases}$$

where $\text{con}(R)$ denotes the class label in the consequent of a rule, $\text{class}(\mathbf{p})$ denotes the class of pattern $\mathbf{p}$, and $R(\mathbf{p})$ is the degree of fulfilment of rule R.

The performance measure $P \in [-1, 1]$ indicates the unambiguousness of a rule. For $P = 1$ a rule is general and classifies all training patterns correctly. For $P = -1$ a rule classifies all training patterns incorrectly. For $P = 0$ either misclassifications and correct classifications of

a rule level each other out, or the rule covers no patterns at all. Only rules with $P > 0$ are considered to be useful.

To reduce the size of the rule base, we apply one of the rule evaluation algorithms of NEFCLASS and select a final (smaller) rule base [10, 15]. The size of the rule base is either given by the user, or so many rules are included such that each training pattern is covered by at least one rule.

After rule creation the fuzzy sets are trained to improve the performance of the classifier. Training algorithms for membership functions of numeric and symbolic variables are given in [6, 15]. The fuzzy set learning algorithm are based on the idea of backpropagation. An output error is determined and used to locally compute updates for each fuzzy set parameter. The computations are based on simple heuristics that aim at increasing or decreasing degrees of membership depending on the current error. Figure 2 illustrates this process.

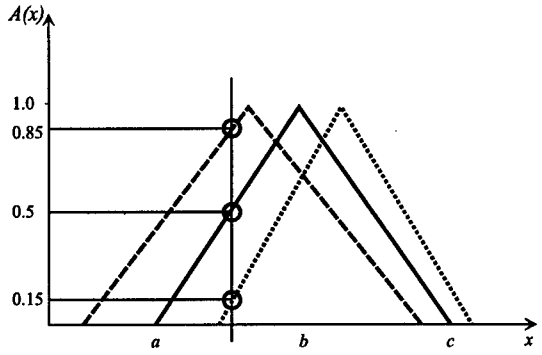


Figure 2: Training membership functions

4 Information Fusion in NEFCLASS

Information fusion refers to the acquisition, processing, and merging of information originating from multiple sources to provide a better insight and understanding of the phenomena under consideration. There are several levels of information fusion. Fusion may take place at the level of data acquisition, data preprocessing, data or knowledge representation, or at the model or decision making level. On lower levels of where raw data is involved, the term (sensor) *data fusion* is preferred. Some aspects of information fusion can be implemented by using NEFCLASS. For a conceptual and comparative study of fusion strategies in various calculi of uncertainty see [2].

If a fuzzy classifier is created based on a supervised learning problem $\tilde{\mathcal{L}}$, then the most common way is to provide a data set, where each pattern is labelled – ideally with its correct class. That means we assume that each pattern belongs to one class only. Sometimes it is not possible to determine this class correctly due to a lack of information. Instead of a crisp classification it would also be possible to label each pattern with a vector of membership degrees. This requires that a vague classification is obtained in some way for the training patterns, e.g. by partially contradicting expert opinions.

Training patterns with fuzzy classifications are one way to implement information fusion with NEFCLASS. If we assume that a group of n experts provide partially contradicting classifications for a set of training data we can fuse the expert opinions into fuzzy sets that describe the classification for each training pattern. According to the context model, we can view the experts as different observation contexts [3, 4].

If the reliability of the experts is known or preferences for selecting an expert should be described, we can assign different values p_k, $\sum_k^n p_k = 1$ to the experts. If there are no preferences, we can assign $p_k = \frac{1}{n}$ to each expert. To create the training set $\tilde{\mathcal{L}}$ for each pattern $\mathbf{x}$ we create a vector $\mathbf{t}$ with

$$t_j = \sum_{k=1}^{n} c_j^{(k)} p_k$$

where $c_j^{(k)}$ is the degree of membership of the pattern $\mathbf{x}$ for class C_j according to the kth expert. The training data then reflects fusion of expert opinions on data set level. Due to the capabilites of its learning algorithms

NEFCLASS can handle such training data in the process of creating a fuzzy classifier.

Like in every fuzzy classifier the output of NEFCLASS is also a vector of membership degrees – like the target vectors of the training data. Such an output offers more information than a crisp classification alone, therefore the interpretation is left to the user and is not done hidden inside the NEFCLASS system. If there are, for example, only low degrees of memberships for all classes or if several classes are activated to a large degree, the user might want to reject the classification and to investigate the case described by the input pattern further by other means. If decision making requires to finally assign a pattern to one class only, then the class with the highest degree of membership can be selected.

Another aspect of information fusion that can be realized with NEFCLASS is to integrate expert knowledge in form of fuzzy rules and information obtained from data. If prior knowledge about the classification problem is available, then the rule base of the fuzzy classifier can be initialized with suitable fuzzy rules before rule learning is invoked to complete the rule base. If the algorithm creates a rule from data that contradicts with an expert rule then three options are available:

- always prefer expert rule,
- always prefer the learned rule,
- select the rule with the larger performance value.

Usually the third option will be used, i.e. the performance of all rules over the training data will be determined and in case of contradiction the better rule prevails. This reflects fusion of expert opinions and observations.

Because NEFCLASS is able to resolve conflicts between rules based on rule performance, it is also able to fuse expert opinions on fuzzy rule level. If rule bases from different experts are available, they can be entered as prior knowledge. They will be fused into one rule base and contradictions are resolved automatically by deleting from each pair of contradicting rules the rule with lower performance.

After all contradictions between expert rules and rules learned from data were resolved, usually not all rules can be included into the rule base, because its size is limited by some criterion. In this case we must decide whether

- to include expert rules in any case, or
- to include rules by descending performances values.

The decision on that option depends on the trust we have in the experts knowledge and in the training data. A mixed approach can be used, e.g. include the best expert rules and then use the best learned rules to complete the rule base.

A similar decision must be made, when the rule base is pruned after training, i.e. is it acceptable to remove an expert rule during pruning, or must such rules remain in the rule base.

5 Fusing Expert Rules and Information from Data

To illustrate our considerations from the previous section, we use the "Wisconsin Breast Cancer" data set (WBC data) [18]. The WBC data has 9 attributes $x_1, \ldots, x_9$ with $x_i \in \{1, \ldots, 10\}$. There are 699 cases, where 16 cases have missing values. Each case is either assigned to the class *benign* (458 cases) or *malign* (241 cases). We randomly split the 699 cases in a training set and a validation set of equal size.

We used two fuzzy sets *small* and *large* to partition the domain of each variable. The membership functions are half trapezoids given by three parameters $a, b, c \in \mathbb{R}$:

$$\mu_{\text{small}}(x) = \begin{cases} 1 & \text{if } a \leq x \leq b \\ \frac{x-b}{c-b} & \text{if } b \leq x \leq c \\ 0 & \text{otherwise} \end{cases}$$

$$\mu_{\text{large}}(x) = \begin{cases} \frac{b-x}{b-a} & \text{if } a \leq x \leq b \\ 1 & \text{if } b \leq x \leq c \\ 0 & \text{otherwise} \end{cases}$$

To initialize the membership functions for all variables we selected $a = 1, b = 4, c = 7$ for *small* and $a = 4, b = 7, c = 10$ for *large*.

To demonstrate the fusion of expert rules and rules created from data we did three experiments. In the first experiment we included the rule

$$R_0 : (s, s, s, s, s, s, s, s, s) \rightarrow b \qquad (1)$$

into the rule base. This notation is short for
if x_1 is *small* and ... and x_9 is *small*
then class is *benign*.

This rule classifies a lot of benign cases correctly and obtains a performance value of 0.55. To fuse this rule with the information contained in the training data, we started the learning process of NEFCLASS-J such that so many rules were created that all patterns are covered. The tool created 56 rules. In this rule base R_0 (1) was still included, because it has a high performance value. All other rules found in the data have rather low performance values around 0.01. Then we started the training process for the membership functions and the automatic pruning process. The final rule base contains 5 rules (a "$-$" denotes that this variable is not used, the performance of a rule is given in brackets):

$$\begin{aligned} R_0 &: (-,-,s,s,-,-,s,s,-,-) \rightarrow b \ (0.55) \\ R_1 &: (-,-,l,l,-,-,l,s,-,-) \rightarrow m \ (0.11) \\ R_2 &: (-,-,l,l,-,-,s,l,-,-) \rightarrow m \ (0.04) \\ R_3 &: (-,-,l,l,-,-,l,l,-,-) \rightarrow m \ (0.20) \\ R_4 &: (-,-,s,s,-,-,l,l,-,-) \rightarrow m \ (0.06) \end{aligned}$$

(s = small, l = large, b = benign, m = malign)

This rule base covers all rules and causes 31 misclassifications on all 699 patterns. We can see, that a pruned version of rule R_0 is still in the rule base.

For the second experiment we allowed the tool to include only 4 rules into the rule base. We again used rule R_0 (1) as prior expert knowledge and processed the training data in the same way. After the rule base was completed by rules discovered in the data, rule R_0 was still in the rule base with a performance value of 0.55. Then we started to train the membership functions an pruned the rule base. This time the resulting rule base contained only 2 rules after pruning:

$$\begin{aligned} R_0 &: (-,-,s,-,-,-,s,-,-,-) \rightarrow b \ (0.58) \\ R_1 &: (-,-,l,-,-,-,l,-,-,-) \rightarrow m \ (0.25) \end{aligned}$$

Again a pruned version of rule R_0 remains in the rule base. This smaller rule base does not cover 8 of the 699 patterns. The rule base causes 45 misclassifications altogether (including the 8 not covered patterns).

For the last experiment we used the following inappropriate rule as expert knowledge:

$$R_0' : (s, s, s, s, s, s, s, s, s) \rightarrow m.$$

We allowed NEFCLASS-J to create 4 rules altogether and started the training process. Because rule R_0' is not supported by the training data, it is deleted and replaced by R_0 (1) during rule creation. Continuing the training process provides the same result as the second experiment.

These experiments show that NEFCLASS is able to fuse expert rules and rules created from data. If the expert rules are supported by the data, the remain in the rule base. During training of the membership functions the rules are revised via modifications of the fuzzy sets in order to improve the performance of the rule base. This can be seen as fusing information obtained from the data with the information provided by the initial membership functions. The pruning process of NEFCLASS optimizes the rule further by deleting variables from the antecedents.

If an expert rule is inappropriate and is not supported by the training data, it is either deleted from the rule base or it is modified. If the antecedent is not supported by the training data, the rule will be deleted during selection of the final rule base or during pruning. If the antecedent does cover a certain number of cases, but its consequent is inappropriate, it will be replaced by a better consequent.

Another example of information fusion by neuro-fuzzy methods in the context of stock prediction can be found in [16].

6 Conclusions

This paper discussed how the neuro-fuzzy classification approach NEFCLASS can be used to implement some aspects of information fusion. NEFCLASS is able to fuse expert rules with rules obtained from data during a learning process. This is currently done by deleting rules that are not supported by training data and by modifying rules that are supported by data. Modification of supported rules is done by training the membership functions and by pruning.

To improve the information fusion capabilities of NEFCLASS, our future work will focus on maintaining several rule bases at the same time and combining their results based on how well they are supported by training data. By this it will be possible to fuse the results of several rule bases depending on their performance without fusing the rule sets itself. Only if the user wants to obtain a single rule base, the different rule bases will be fused by the techniques described in this paper.

The tool NEFCLASS-J that was used to demonstrate the approaches discussed in this paper can be obtained from our WWW server at http://fuzzy.cs.uni-magdeburg.de.

References

[1] M. Berthold and K.-P. Huber. Tolerating missing values in a fuzzy environment. In M. Mares, R. Mesiar, V. Novak, J. Ramik, and A. Stupnanova, editors, *Proc. Seventh International Fuzzy Systems Association World Congress IFSA'97*, volume I, pages 359–362, Prague, 1997. Academia.

[2] J. Gebhardt and R. Kruse. Parallel combination of information sources. In D. Gabbay and P. Smets, editors, *Belief Change*, volume 3 of *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, pages 329–375. Kluwer Academic Publishers, Dordrecht, NL, 1998.

[3] J. Gebhardt, R. Kruse, and D. Nauck. Information compression in the context model. In *Proc. Workshop of the North American Fuzzy Information Processing Society (NAFIPS92)*, pages 296–303, Puerto Vallarta, Dec. 1992.

[4] R. Kruse, J. Gebhardt, and F. Klawonn. *Foundations of Fuzzy Systems*. Wiley, Chichester, 1994.

[5] D. Nauck. Using symbolic data in neuro-fuzzy classification. In *Proc. 18th International Conf. of the North American Fuzzy Information Processing Society (NAFIPS99)*, New York, NY, 1999.

[6] D. Nauck, F. Klawonn, and R. Kruse. *Foundations of Neuro-Fuzzy Systems*. Wiley, Chichester, 1997.

[7] D. Nauck and R. Kruse. A neuro-fuzzy method to learn fuzzy classification rules from data. *Fuzzy Sets and Systems*, 89:277–288, 1997.

[8] D. Nauck and R. Kruse. New learning strategies for NEFCLASS. In M. Mares, R. Mesiar, V. Novak, J. Ramik, and A. Stupnanova, editors, *Proc. Seventh International Fuzzy Systems Association World Congress IFSA'97*, volume IV, pages 50–55, Prague, 1997.

[9] D. Nauck and R. Kruse. NEFCLASS-X – a soft computing tool to build readable fuzzy classifiers. *BT Technology Journal*, 16(3):180–190, 1998.

[10] D. Nauck and R. Kruse. Neuro-fuzzy methods in fuzzy rule generation. In D. Dudois and H. Prade, editors, *Approximate Reasoning and Fuzzy Information Systems*, Handbook of Fuzzy Sets, chapter 5. Kluwer, Norwell, MA, 1998. To appear.

[11] D. Nauck and R. Kruse. Neuro-fuzzy systems. In E. Ruspini, P. Bonissone, and W. Pedrycz, editors, *Handbook of Fuzzy Computation*, chapter D.2. Institute of Physics Publishing Ltd., Philadelphia, PA, 1998.

[12] D. Nauck and R. Kruse. Fuzzy classification rules using categorical and metric variables. In *Proc. 6th Int. Workshop on Fuzzy-Neuro Systems 1999 (FNS'99)*, pages 133–144, Leipzig, 1999. Leipziger Universitätsverlag.

[13] D. Nauck and R. Kruse. Neuro–fuzzy systems for function approximation. *Fuzzy Sets and Systems*, 101:261–271, 1999.

[14] D. Nauck, U. Nauck, and R. Kruse. Generating classification rules with the neuro-fuzzy system NEFCLASS. In *Proc. Biennial Conference of the North American Fuzzy Information Processing Society NAFIPS'96*, Berkeley, CA, June 1996.

[15] D. Nauck, U. Nauck, and R. Kruse. NEFCLASS for JAVA – new learning algorithms. In *Proc. 18th International Conf. of the North*

American Fuzzy Information Processing Society (NAFIPS99), New York, NY, 1999.

[16] S. Siekmann, J. Gebhardt, and R. Kruse. Information fusion in the context of stock index prediction. In *Proc. European Conference on Symbolic and Quantitative Approaches to Uncertainty (ECSQARU'99)*, London, 1999. To appear.

[17] L.-X. Wang and J. M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Trans. Syst., Man, Cybern.*, 22(6):1414–1427, 1992.

[18] W. Wolberg and O. Mangasarian. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proc. National Academy of Sciences*, 87:9193–9196, Dec. 1990.

[19] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[20] L. A. Zadeh. Soft computing and fuzzy logic. *IEEE Software*, 11(6):48–56, 1994.

Maritime Avoidance Navigation, Totally Integrated System (MANTIS)

T Tran*, C J Harris, P A Wilson†

Email: tt97r@ecs.soton.ac.uk
Image, Speech & Intelligent Systems Group
Department of Electronics and Computer Science
Southampton University SO17 1BJ

Abstract

A collision avoidance system is proposed to improve the efficiency and safety of marine transport, namely Maritime Avoidance Navigation, Totally Integrated System (MANTIS). The principle behind its operation is to remove the difficulties and uncertainties involved in marine navigation through a system structure which makes marine transport deterministic - reminiscent of Air Traffic Control. The key features of MANTIS involve; localisation of vessel states and its environment (LVSE), Automatic Collision Avoidance Advisory Service (ACAAS), an Integrated Display System (IDS), Path Planning and Scheduling Service (PPSS), and Automated Ship Guidance and Control (ASGC).

Keywords: marine navigation, fusion, adaptive, modelling, control, fuzzy, expert.

1 Introduction

Ship collisions have occurred from when the first ships were set afloat. The problem has escalated due to increases in traffic, speed and size of present day vessels. Unlike road traffic, there are generally no boundaries constraining what path a ship may take moving between any two points. As a result there are situations where navigation schedules of two or more ships overlap - giving potential for collision. It is important to understand the process and demands required of the ship operator during navigation to establish the problem areas [5]. These areas need to be targeted and improved upon for safety and efficiency of ship operation.

Information collection

Navigators must collect information that is required for navigation from sensory and data sources. The number of independent sources of information means it is difficult for operators to sustain continuous monitoring. This leads to slow response times and mistakes. *There is a need to integrate all information which is delivered independently.*

Information analysis

Most information is presented to the navigator in its raw form. Due to limitations in humans analysing ability it is impossible to analysis and digest all the available data. Consequently, navigators are more concerned with their immediate situation (i.e. the most dangerous ship) and pay insufficient attention to the global surroundings or future potential predicament. *There is a need to deliver the effective information in an easily understood way for rapid situation assessment to ease decision-making.*

Decision making

Predictive analysis of the situation is very important, and is traditionally based on visual observation which can often be difficult to extrapolate (e.g. in fog). Test results show that sea mariners response for any given situation are subject to a num-

*Supported by grants from RACAL Research and EPSRC
†Department of Ship Science, Southampton University.

ber of physical and psychological factors. Their inconsistency causes uncoordinated actions between mariners because neither can be certain of the other's intent. *There is a need to automate or aid the decision making process deterministically and to display to the mariner the most appropriate collision avoidance action.*

Execution of Collision Avoidance Action

The collision avoidance action is a very complex one and causes a high work load for the navigator. He has to decide on the timing and operational qualitities of the actuators and consider external environmental forces and the maneuverability of own ship. Throughout he has to pay attention to the behaviour of other ships while deciding the timing to release the actuators. *There is a need to automate or aid the collision avoidance action by controlling or advising the movement of actuators.*

1.1 The MANTIS solution

The underlining cause of the majority of marine collisions can be put down to human error, and it has been shown that human error is directly related to work load [1]. Thus by minimising the human work load the room for error is reduced. Unfortunately, for economic reasons there is a continual reduction in the number of human operators, many of which are poorly trained [2]. To counter this adverse effect, the only viable solution is to increase the level of automation in all areas of ship operation. From the above analysis, a system to improve marine safety can be identified and needs to consist of the following parts:

- *Localisation of Vessel States and its Environment (LVSE).* Provide accurate and robust navigational information (position, velocity) of all ships, and information on sea depth, current and wind states. Confidence intervals also needs to be given for each data value.

- *Path Planning and Scheduling Service (PPS).* Safe and efficient navigational routes are generated by considering other ship paths and environmental conditions before the journey starts; thus minimising journey time, and more importantly, the event of close encounter situations.

- *Automatic Collision Avoidance Advisory Service (ACAAS).* For unforeseen or dynamic events, potential risk situations are resolved using a knowledge base system which comply with Collision Regulations (COLREGs) [3]. The algorithm needs to be capable of dealing with complex multi-ship encounter situations in an intuitive and predictable manner.

- *Integrated Display System (IDS).* These provide decision support and visualisation of collision avoidance advise. By superimposing danger zones [4]and/or encounter on scheduled course line [5] on-top of an Electronic Chart Display Information System (ECDIS).

- *Automated Ship Guidance and Control (ASGC).* Given the general collision avoidance advice from ACAAS, this subsystem calculates the precise trajectory via way-points for the ship to navigate, within the constraints of ship dynamics and environmental conditions. Automation and control of ship rudder and engine revolution can be made allowing the ship to smoothly interpolate between way-points.

At present some of these areas are only partially satisfied via Vessel Traffic Services (VTS) and electronic navigational aids. The contribution of VTS to navigational safety is in its ability to coordinate traffic flow to minimise traffic density in specific areas [6]. Navigation aids such as Automatic Radar Plotting Aid (ARPA) and ECDIS at their present state allow efficient navigational support with regard to speed and accuracy of calculation and gives an effective graphical display of own-ship immediate disposition in relation to target vessels, obstacles and land [4].

MANTIS is reminiscent of an Air Traffic Control system. The structure and deterministic approach to navigation provided by MANTIS minimises the uncertainties which causes uncoordinated vessel actions. Even potential risk situations which are unforeseen are made deterministic via collision avoidance advice. And if need be, automatic control of ships can be made.

1.2 MANTIS architecture

The system architecture is fundamental part of MANTIS. A range of distributed sensors are used to provide a rich data pool. The data is integrated in

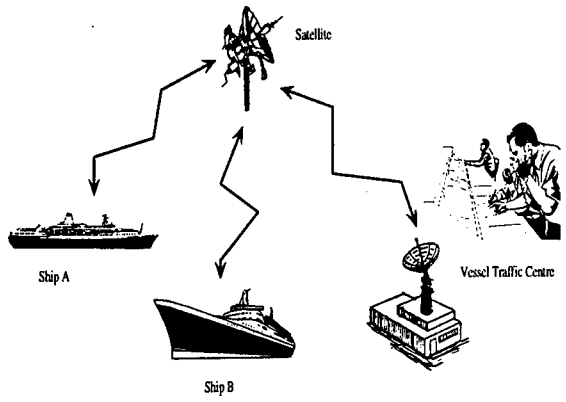


Figure 1: Diagram showing the communication between the Vessel Traffic Centre and all operational ships.

two stages - locally on-board the vessels giving local consensus features and global fusion at the Vessel Traffic Centre (VTC) giving global consensus features. Adaptive neurofuzzy sensor and ship models are used for estimation and prediction [7]. The combination of these methods ensures that accurate and robust consensus information can be provided for picture compilation and collision avoidance computation.

PPSS, ACAAS and the way-point guidance aspect of AarbitrarySGC are functions of the VTC. IDS and ship control are handled by the on-board ship computer. Communication is made via satellite using ships-to-shore data exchange topology, figure 1. Data transmitted to the VTC consist of locally fused navigational data sent by each vessel or external marine sensor. The VTC transmits global consensus information to all ships and any way-point modifications to individual vessels.

With reference to figure 2, consider any arbitrary ship j. On-board sensors on the ship gather data about the ship and its environment $[\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_s]^T$. Sensor models transform these measurements into a set of common features $[\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_s]^T$ and compensates for noise components, this is combined with estimates from the ship model using the extended Kalman filter to form local consensus features $\hat{\mathbf{x}}_j$. The common feature set consist of ship states, wind, sea and current states.

Input into the VTC consist of local consensus features from all vessels and external marine sensors $[\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \cdots, \hat{\mathbf{x}}_n]^T$. At the VTC, chart data is used to compliment depth and land features integration. The output from the global fusion process forms the global consensus feature set $\hat{\mathbf{x}}$. This information is fed back to all vessels to update their local feature states. The VTC also uses this information to assess whether any collision risk exists between the ships and if necessary the collision avoidance action or decision $\mathbf{d}$ is generated prompting an alteration of vessel course via a subset of modified way-points $\Delta\mathbf{P} = [x_i, y_i, U_i]_{i=n_t}^{n_t+m}$, where x_i, y_i are the way-point absolute position and U_i is the traveling speed advised moving from the previous way-point to way-point i, n_t is the initial way-point of the avoidance manoeuvre and m are the number of way-points necessary to execute the avoidance manoeuvre. The guidance and control subsystem determines the course and velocity change required for the vessel to reach a designated way-point $\mathbf{p} = [x, y, U]$, the outputs $\mathbf{u} = [\delta_c, n_c]^T$ are rudder angle and shaft revolution commands to the ship actuators.

Prior to the voyage, or when a complete route reassessment is needed, given the vessel's present position, the final destination point and journey time, $\mathbf{J} = [x_0, y_0, x_d, y_d, t]$ the navigator formulates his navigation plan as a set of way-points for the whole journey, $\mathbf{P} = [x_i, y_i, U_i]_{i=0}^{N}$. To aid this process, the path planning and scheduling service which contains update information on the traffic situation and sea states can help advise navigators on this task. Data of all vessel routes (way-points) are actively stored in the *Global way-point database.*

2 Local fusion

2.1 Adaptive modelling

To minimise errors and improve estimation of the ship states, on-line adaptive neurofuzzy models are used. These networks have been shown to be capable of modelling any system within an arbitrary accuracy [7]. There generalisation ability to previously unseen situations and their robustness to disturbances makes them ideal for this application.

The ship controller, ship model and sensor models are implemented us Co-Active Neurofuzzy Inference System (CANFIS) networks which can be trained both off-line and in real-time, figure 3. CANFIS is an extension of the single output system ANFIS (Adaptive Neurofuzzy Inference System) to multi-

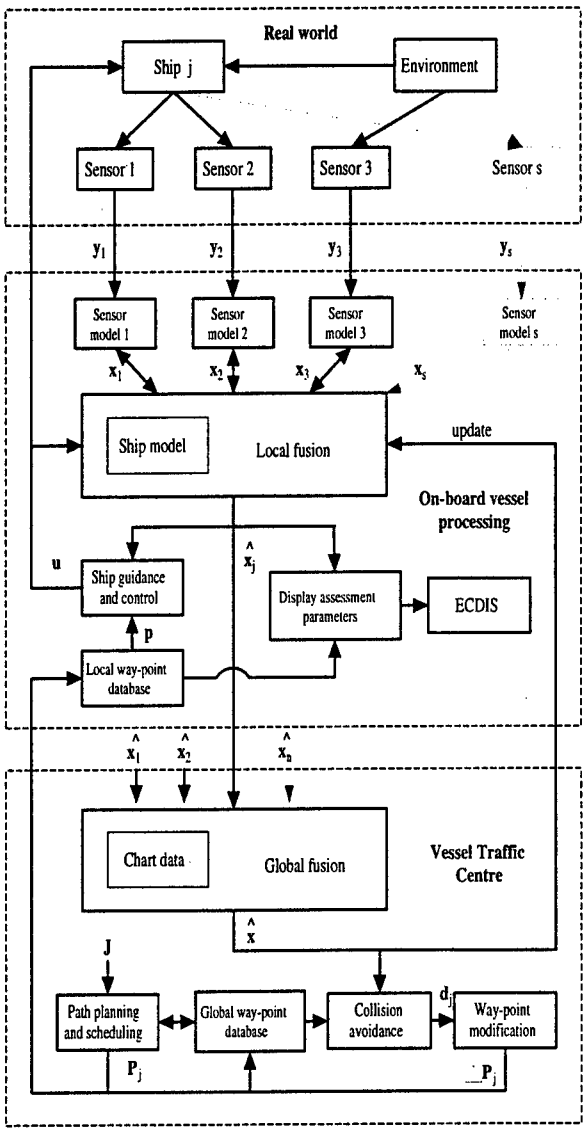


Figure 2: MANTIS architecture

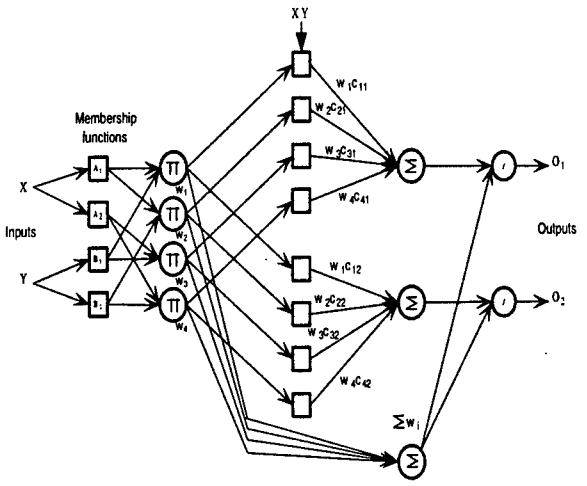


Figure 3: A two input, two output CANFIS network structure. The outputs share the same antecedents of fuzzy rules which allow correlations to be made between the outputs. In addition, the number of adjustable parameters are drastically less than the case if multiple ANFIS networks were employed for the same modelling problem.

ple outputs. These networks uses first order Sugeno consequent output functions.

For a general multi-input, multi-output system

$$\mathbf{o}(k) = \mathbf{f}(\mathbf{i}(k); \Theta) \tag{1}$$

where $\mathbf{o}$ is the network outputs, $\mathbf{i}$ is its inputs and Θ are the network parameter set. To train the network, input $\mathbf{i}$ and desired output $\mathbf{d}$ data pairs, $D_f = [\mathbf{i}(k); \mathbf{d}(k)]$ are required. The hybrid learning rule which combines least squares estimation and error back propagation is used to update the linear and nonlinear network parameters, respectively. This is achieved through minimisation of an error function, typically

$$E(k) = ||\mathbf{d}(k) - \mathbf{f}(k)||^2 \tag{2}$$

Sensor model

Sensor modelling is needed for estimation of the ship states via the extended Kalman filter where both its output and Jacobian is required. The superiority of CANFIS over ANFIS is fully exploited in this application where a single network can be used to model all the sensors on-board the ship. From equ. 1 the sensor model can be written as

$$\mathbf{y}(k) = \mathbf{h}(\mathbf{x}(k); \Theta) \quad (3)$$

Where the inputs $\mathbf{x}$ into the network are the ship states and the outputs $\mathbf{y}$ are the sensor measurements. In the simplest case this can be viewed as a coordinate transformation from ship features to sensor coordinates, e.g. in the case of a radar system, from Cartesian to polar coordinates. However, should the sensor characteristics change during its operation (e.g. due to temperature effect, atmospheric effects), the network will adapt on-line to compensate for these changes and can therefore remove bias effects.

Ship model

The ship model is required for state estimation via the extended Kalman filter and its Jacobian is also needed to update the controller parameters. The inputs into the model, $\mathbf{i} = [\mathbf{x}, \mathbf{u}]^T$, consist of the ship states $\mathbf{x}$ and actuator control inputs $\mathbf{u}$, and the output $\mathbf{y}$ are the updated ship states. The model is thus represented by

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k); \Theta) \quad (4)$$

The ships states, $\mathbf{x} = [u, v, r, x, y, \delta, n]^T$ consist of velocities in body fixed coordinates, its position in Cartesian coordinates, rudder angle and engine revolution, respectively. The inputs, $\mathbf{u} = [\delta_c, n_c, d]^T$, commanded rudder angle and engine revolution, and sea depth.

The ship characteristics can change depending on its load, and changes in the sea state, thus the benefits of on-line adaptive networks are again of great asset in this application.

Ship controller

The ship controller is trained using specialised learning which is a direct method of minimising the system error by back propagating error signals through the ship model. Inverse learning can also be used, which has the advantage that the Jacobian of the ship dynamics are not required, however minimisation of system error is not guaranteed. The controller network is as follows

$$\mathbf{u}(k) = \mathbf{F}(\mathbf{x}_d(k), \mathbf{x}(k); \theta) \quad (5)$$

Substituting equ.4

$$\mathbf{u}(k) = \mathbf{F}[\mathbf{x}_d(k), \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k); \Theta); \theta] \quad (6)$$

Given the desired states $\mathbf{x}_d(k)$ and the current ship states $\mathbf{x}(k)$ the task of the controller is to determine the control action $\mathbf{u}$ that would minimise a given criteria. The criteria or error measure stated below also penalises the amount of control action used,

$$\mathbf{E}(\theta) = \mathbf{e}^T\mathbf{Q}\mathbf{e} + \mathbf{u}^T\mathbf{R}\mathbf{u} \quad (7)$$

where $\mathbf{e} = (\mathbf{x}_d - \mathbf{f})$, and $\mathbf{f}$ is the ship model output, given above. Two diagonal matrices $\mathbf{Q}$ and $\mathbf{R}$ are used to weigh the system states and control action. The back-propagation method is used to update the controller parameters θ to minimise the error measure and to calculate the Jacobian of the ship model.

2.2 Disturbances

Disturbances effecting ship motion come from; wind, wave and sea current [10]. All are dependent on the local wind conditions. Wind and wave disturbances result in external forces acting on the ship. For slowly varying forces the ship actuators can compensate for these first order effects. The sea current can be treated as an additive term on the velocity of the ship. It remains to be seen whether on-line adaptive networks can compensate for these effects or whether additional input terms such as wind speed U_w and direction β_w are needed as inputs into the network.

2.3 Extended Kalman filter

The extended Kalman filter is used for on-line state estimation of the ship's non-linear dynamics. The ship and sensor model outputs are combined with sensor measurements to predict future states. From equ. 4 and 3 the system and sensor models are

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) + \mathbf{w}(k) \quad (8) \\ \mathbf{y}(k) &= \mathbf{h}(\mathbf{x}(k)) + \mathbf{v}(k) \end{aligned}$$

where $\mathbf{w}(k) \sim N(0, \mathbf{Q}(k))$ is the system noise and ship modelling error, and $\mathbf{v}(k) \sim N(0, \mathbf{R}(k))$ is sensor noise and sensor modelling error. The noise covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ can be obtained from their respective network error residues $\mathbf{E}$.

For robustness a prefilter should be used to remove surplus measurements from sensor readings and to detect sensor failure.

3 Vessel Traffic Centre

3.1 Global fusion

The fusion of local features $\hat{x}_j$ from ships and external sensors to form global consensus features is achieved using the standard Kalman filter. The process combines similar features j optimally taking into account their error covariance $\mathbf{Q}_j$. Furthermore, $\mathbf{Q}_j$ is adjusted to take into account delays between feature extraction and the final fusion process. A simple linear system model is used for propagation of the states.

Imaging sensors giving data on land and fixed objects are integrated with electronic chart data. Sea states such as current and wind velocities are measured, estimated and predicted for use by the Path Planning and Scheduling Service.

3.2 Automatic collision avoidance advisory service

In situations where there is potential for collision the VTC notifies the navigator and advises him of the avoidance procedure. The advise can be derived either from a human operator and/or expert system. The expert knowledge-base is constructed from collision avoidance regulations (COLREGs). The following highlights the importance of COLREG for collision avoidance [8]:

- There is worldwide acceptance and understanding of its general procedures for avoiding collision.
- The Regulations are acknowledged (in its formulation) to contain a distillation of historical navigational experience. With continual improvements and specific guidance to reflect current state of development, thus the Regulations can be assumed to reflect the present optimum practice in the inexact art of marine navigation.
- The Regulations can be easily interpreted as a series of production rules (IF-THEN statements).

A necessary requirement of a collision avoidance system is its predictability. Devising an avoidance route which optimises a mathematical function may produce time and spatially efficient paths but these paths may be non-intuitive and thus hard to foresee by other ships in the vicinity - causing uncoordinated ship manoeuvres. Here a number of heuristical stages are used making up the expert. The transparency (interpretability) of the knowledge-base allows the avoidance advise given by the expert to be validated.

- *Target ship classification* [5]. Each target ship is classified with respect to own ship as being either; clear - no threat whatever alteration of course own ship makes, restricting - prevents own ship from performing specific manoeuvres, threat - collision potential if both ships maintain their current speed and course.
- *Restriction on own ship movement.* For restricting ships determine the constraint they impose on own ship movements. For example the restricting ship may prevent own ship from turning to starboard or port, and/or, changes in own ship speed may cause problems astern or ahead.
- *Encounter type.* For restricting and threatening ships classify the encounter type relative to own ship. e.g. own ship overtaking, target crossing starboard to port, head-on, etc.
- *Risk stage* [9]. For threatening ships determine their current level of risk against own ship, i.e. developing, manoeuvring, critical. These categories determines which actions are permitted in accordance with COLREGs.
- *Collision avoidance advice.* Given the risk stage, encounter type and constraint imposed on own ship movements, the expert determines the most appropriate action to proceed, i.e. starboard, port alterations and/or speed alterations. The final avoidance advice is purposefully simple.

3.3 Way-point modification

Given the expert collision avoidance advice, the next task is to generate a subset of way-points in the general direction permitted. Constraints on

ship manoeuvrability and environmental conditions are considered. Furthermore, rule 8 of COLREG states, *any alteration of course and/or speed be large enough to be readily apparent to another vessel... (and) a succession of small alterations of course or speed should be avoided.*

3.4 Path planning and scheduling service

A major part of the VTC is to supply the navigator with information such as traffic density and weather conditions allowing them to best plan their journey. If the journey is planned correctly then potential hazardous situations are avoided and journey time and fuel will be minimised. The advisory service may also suggest a route if required, or on reflection, object to the navigator's planned route for safety reasons.

3.5 Guidance law

Given the set of way-points $[x_d(k), y_d(k)]_{k=1}^{N}$, Line of Sight (LOS) guidance can be used to direct the ship in the desired direction of travel [10]:

$$\psi_d = \tan^{-1}\left(\frac{y_d(k) - y(t)}{x_d(k) - x(t)}\right) \tag{9}$$

Once the ship lies within a *circle of acceptance* with radius ρ_0 around the way-point $[x_d(k), y_d(k)]$ the next way-point can be selected $[x_d(k+1), y_d(k+1)]$.

4 Integrated display system

An appropriate display of the current and predicted future situation is essential to help the navigator in the decision making process. Information should be delivered to the human operators with the aim of improving navigation safety, i.e. the display is easy to understand and interpret and is expressed in a manner consistent with the method used to navigate the ship. ECDIS have been shown to be an effect tool for understanding the ship current predicament. Evaluation and visualisation of future predicaments are possible using situation assessment displays, and by overlaying these displays on top of ECDIS gives an integrated display system.

Two types of situation displays for integration in ECDIS are considered here; danger zones [4] and encounter situation on the scheduled course line [5].

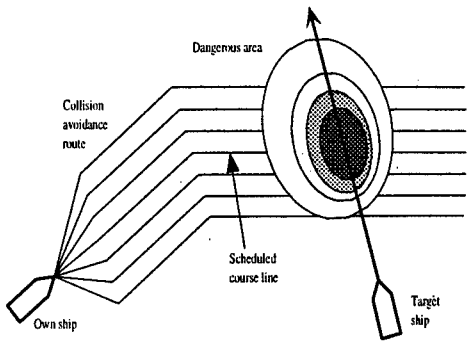


Figure 4: Danger zone situation assessment display

The modified course as the result of the collision avoidance advise can be visualised and validated by either one of these display types which helps to reassure the navigator of the advice given by the system. To reduce clutter of the display a definable number of target ships purposing the greatest threat can be set.

4.1 Danger zones

Basically the task of collision avoidance is to keep a defined zone around own ship free. Traditionally a circle around own ship is used with radius equivalent to the permissible closest point of approach C_A. The safety circle moving along with own ship gives no useful information for collision avoidance advise in ECDIS. A more initiative approach is to define boundaries in the marine environment where own ship should not encroach, known as 'danger zones'

4.2 Encounter situation on scheduled course line

The scheduled course line of own and target ship are drawn on the display. The target ship position at the distance of closest point of approach (DCPA) of the encounter situation is shown and the ship symbols are red when own ship crosses the bow of target ship, yellow when own ship crosses the stern of target ship, and white during passing or overtaking encounters.

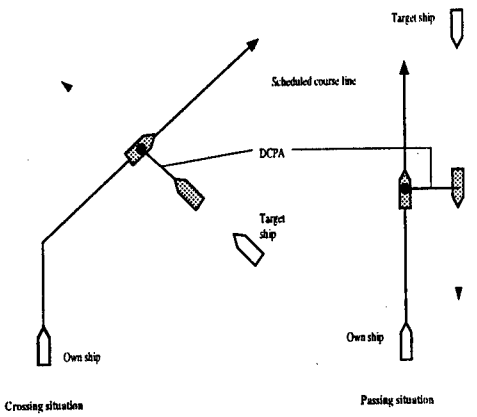


Figure 5: Encounter situation on scheduled course line assessment display.

5 Summary

The problems with the present situation in marine navigation have been discussed giving provocation for this research. In this paper a system has been proposed to improve the efficiency and safety of marine transport by alleviating these identified problem areas. An overview of the architecture and components of MANTIS have been given.

References

[1] Shinya Nakamura & Kuniji Kose. *The strategic maneuvering to avoid collision.* Yusen Marine Science Inc. Hiroshima University, Japan. Maritime Collision and Prevention, Vol 2. Edited by J.S. Zhao, P.A. Wilson, Z. Hu & F. Wang. September 1996. p172-181.

[2] P.Barber. *The need for improved curriculum development in marine simulation training. MARSIM '96. Marine Simulation and Ship Manoeuvrability.* Edited by M.S. Chislett. p77-85.

[3] A.N. Crockcroft and J.N.F Lameijer. *A guide to the Collision Avoidance Rules.* BH Newnes. Fifth Edition, 1996.

[4] J. Froese & S. Mathes. *Computer-assisted collision avoidance using ARPA and ECDIS.* Maritime Collision and Prevention, Vol 2. Edited by J.S. Zhao, P.A. Wilson, Z. Hu & F. Wang. September 1996. p200-213.

[5] Y. Ishioka, K. Kose, H. Kobayashi, C. Yang, H. Yamda & S. Nakamura. *A study on a support of decision-making for collision avoidance in INS. MARSIM '96. Marine Simulation and Ship Manoeuvrability.* Edited by M.S. Chislett. p49-58.

[6] T. Degree. The management of marine traffic by VTS and VTMIS. *The International Conference of Preventing Collision at Sea,* pages 1-6, 1996.

[7] J.-S.R. Jang, C.-T. Sun, E.Mizutani. *Neuro-fuzzy and Soft Computing.* Prentice Hall, 1997.

[8] G.P. Smeaton and F.P. Coenen. *Developing an intelligent marine navigation system.* Computing & Control Engineering Journal, March 1990. p95-103.

[9] H. Hilgert & M. Baldauf. *A common risk model for the assessment of encounter situations on board ships.* Maritime Collision and Prevention, Vol 2. Edited by J.S. Zhao, P.A. Wilson, Z. Hu & F. Wang. September 1996. p136-148.

[10] T.I. Fossen. *Guidance and Control of Ocean Vechicles.* Wiley, 1994.

A FUZZY LOGIC ALGORITHM FOR OPTIMAL ALLOCATION OF DISTRIBUTED RESOURCES

Dr. James F. Smith III
Naval Research Laboratory, Code 5741
Washington, D.C., 20375-5000
Telephone: 202.767.5358
Fax: 202.404.7690
jfsmith@drsews.nrl.navy.mil

Robert D. Rhyne II
Naval Research Laboratory, Code 5741
Washington, D.C., 20375-5000
Telephone: 202.767.5938
Fax: 202.404.7690
rrhyne@drsews.nrl.navy.mil

ABSTRACT- *A fuzzy logic based resource manager (RM) that will allocate resources distributed across many platforms is under development. The platforms will consist of ships, aircraft, etc. The resources will be various sensors: ESM, RADAR, IFF, and communications. The RM will allow codification of military expertise in a simple mathematical formalism known as the fuzzy decision tree. The fuzzy decision tree will form what is known as a fuzzy linguistic description, i.e., a formal fuzzy if-then rule based representation of the system. Since the decision tree is fuzzy the uncertainty inherent in the root concepts propagates throughout the tree. The functional form of the fuzzy membership functions for the root concepts will be selected heuristically and will generally carry one or more free parameters. The free parameters in the root concepts will be determined by optimization both initially and later at non-critical times. A genetic algorithm will be used for optimization.*

Keywords: fuzzy logic, genetic algorithms, expert systems, multisensor data fusion, distributed AI algorithms

1. Introduction

Modern naval battleforces generally include many different platforms each with its own sensors, radar, ESM, and communications. The sharing of information measured by local sensors via communication links across the battlegroup should allow for optimal or near optimal decisions. The survival of the battlegroup or members of the group depends on the automatic real-time allocation of various resources.

A fuzzy logic algorithm has been developed that automatically allocates electronic attack (EA) resources in real-time. The particular approach to fuzzy logic that will be used is the fuzzy decision tree, a generalization of the standard artificial intelligence technique of decision trees [1].

The controller must be able to make decisions based on rules provided by experts. The fuzzy logic approach allows the direct codification of expertise forming a fuzzy linguistic description [2], i.e., a formal representation of the system in terms of fuzzy if-then rules. This will prove to be a flexible structure that can be extended or otherwise altered as doctrine sets, i.e., the expert rule sets change.

The fuzzy linguistic description will build composite concepts from simple logical building blocks known as root concepts through various logical connectives: "not", "and", "or", etc. Optimization will be conducted to determine the form of the membership functions for the fuzzy root concepts.

Section 2 gives a brief introduction to the ideas of fuzzy set theory, fuzzy logic, decision trees, root and composite concepts. Section 2 uses these concepts to develop the kinematic-ID subtree, which is an important component of the decision tree. Section 3 describes the optimization of the resource manager's performance. Section 4 provides an example of the algorithm's allocation of EA resources distributed over three platforms against an airborne targeting radar with uncertain ID. Section 5 discusses association algorithms and points out the usefulness of a particular fuzzy logic based association algorithm. Section 6 discusses future developments. Finally, section 7 provides conclusions.

2. A Brief Introduction to Fuzzy Sets, Logic, and Decision Trees

The resource manager (RM) must be able to deal with linguistically imprecise information provided by an expert. Also, the RM must control a number of assets and be flexible enough to rapidly adapt to change. The above requirements suggest an approach based on fuzzy logic. Fuzzy logic is a mathematical formalism that attempts to imitate the way humans make decisions. Through the concept of the grade of membership, fuzzy set theory and fuzzy logic allow a simple mathematical expression of uncertainty. The RM will require a mathematical representation of domain expertise. The decision tree of classical artificial intelligence provides a graphical representation of expertise that is easily adapted by adding or pruning limbs. Finally, the fuzzy decision tree, a fuzzy logic extension of this concept, allows easy incorporation of uncertainty as well as a graphical codification of expertise.

This section will develop the basic concepts of fuzzy sets, fuzzy logic, and fuzzy decision trees. Examples from a primitive military doctrine set will be provided.

2.1 Fuzzy Set Theory

This subsection provides a basic introduction to the ideas of fuzzy set theory. Fuzzy set theory allows an object to have partial membership in more than one set. It does this through the introduction of a function known as the membership function, which maps from the complete set of objects X into a set known as membership space. More formally, the definition of a fuzzy set [3] is

If X is a collection of objects denoted generically by x then a fuzzy set A in X is a set of ordered pairs:

$$A = \{(x, \mu_A(x)) | x \in X\}$$

$\mu_A(x)$ is called the membership function or grade of membership (also degree of compatibility or degree of truth) of x in A which maps X to the membership space M.

The logical connectives "and", "or", and "not" are defined as

$$or: A \cup B \rightarrow \mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)]$$

$$and: A \cap B \rightarrow \mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]$$

$$not \quad B: \overline{B} \rightarrow \mu_{\overline{B}}(x) = 1 - \mu_B(x)$$

2.2 Fuzzy Decision Trees and Root Concepts

In this section methods of constructing classical and fuzzy decision trees are discussed. The fuzzy decision tree will provide a graphically intuitive way of propagating information from basic to complex concepts.

A classical decision tree is a standard artificial intelligence technique for making decisions. It's graphical nature allows an easy intuitive representation of information. The method of constructing decision trees, both classical and fuzzy, is best illustrated through an example. Consider the following simple military doctrine set, i.e., a set of rules provided by an expert:

R1: IF target is *Attacking* or *Bearing-in* or *Maneuvering*, THEN the target is *Important*
R2: IF target is *Close* and not *Friend*, THEN the target is *Attacking*.

These rules can be represented in a tree form which is given in Figure 1.

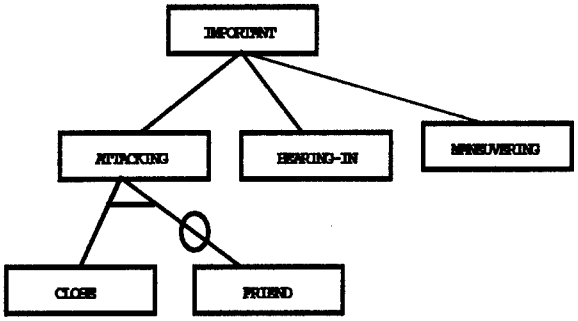


Figure 1 : Decision Tree for rules R1 and R2

In Figure 1 the root concepts are "close", "friend", "bearing-in", and "maneuvering". The composite concepts are "attacking" and "important". The root and composite concepts are placed in their own boxes. The boxes are connected with lines. Vertices marked with a horizontal line are read as "and", unmarked vertices as "or", and lines marked by a circle indicate negation.

The conversion from a classical decision tree to a fuzzy decision tree is carried out by

-assigning each classical root concept, those boxes at the bottom-most level of the decision tree, membership functions and then

-converting all classical "or", "and", and "not" operations to the analogous fuzzy operations.

So for track i, the following grades of membership associated with the corresponding root concepts must be defined:

$$\mu_{friend}, \mu_{close}, \mu_{bearing-in}, \text{and } \mu_{maneuvering}$$

Pursuing the second component of the above description, i.e., the conversion of classical "and", "or", and "not" into the related fuzzy set-theoretic quantities, gives the following grades of membership for the composite concepts "attacking" and "important":

$$\mu_{attacking}(i) = \min[\mu_{close}(i), 1-\mu_{friend}(i)]$$

$$\mu_{\text{important}}(i) =$$

$$\max[\mu_{attacking}(i), \mu_{bearing-in}(i), \mu_{maneuvering}(i)]$$

$$\mu_{\text{important}}(i) =$$

$$\max[\min[\mu_{close}(i), 1-\mu_{friend}(i)],$$

$$\mu_{bearing-in}(i), \mu_{maneuvering}(i)]$$

The resulting grades of membership for composite concepts are used for establishing priorities for resource allocation.

Figure 1 is referred to as the kinematic-ID subtree. It is a subtree of a larger fuzzy decision tree used by an isolated ship for allocation of its own EA resources. Each ship in the battlegroup has an isolated ship tree that allocates its EA resources. These isolated ship trees, when linked together by information from line of sight communication form a larger tree, known as the multi-platform tree. It is this tree together with information sent over communications links, that determines allocation of EA resources over the entire battlegroup. The full isolated ship tree, communication models, and the multi-platform tree will not be discussed in detail here due to space limitations. A more detailed account of these concepts will be published in the near future [4].

2.3 Root Concept Membership Functions

The next step required for implementation of the fuzzy linguistic description is defining membership functions for the root concepts. There is not an a priori best membership function so a reasonable mathematical form is selected. This subjective membership function will be given in terms of one or more parameters that must be determined. The parameters may be set initially by an expert or they may be the result of the application of an optimization algorithm. The possible use of a stochastic optimization algorithm to determine the unknown parameters in root concept membership functions is discussed in section 3.

As a first example of a membership function definition consider the root concept "close." The concept "close" refers to how close the target/emitter on track i is to the ship, or more generally platform of interest. The universe of discourse will be the set of all possible tracks. Each track i has membership in the fuzzy set "close" based on its range R (nmi) and range rate dR/dt (ft/sec). An appropriate membership function might be

$$\mu_{close}(i) = \frac{1}{1-\alpha\,|R_i - R_{\min}\,| / \max(-\dot{R}_i, \dot{R}_{\min})}$$

The parameters to be determined by optimization are

$$\alpha, \quad R_{\min}, \quad \text{and} \quad \dot{R}_{\min}.$$

3. Optimization

There are many different types of optimization algorithms found in the literature. Many of these algorithms are known as greedy algorithms because they will find as a solution the first extremum encountered in a parameter space. Examples of this kind of algorithm are found in reference [5].

An algorithm that has the capability to explore parameter space before settling on a solution, intuitively would seem to have greater probability of selecting an optimal or near-optimal solution than a greedy algorithm. Examples of algorithms of this kind are stochastic optimization algorithms, which include simulated annealing [5] and genetic algorithms [6].

A genetic algorithm (GA) is an optimization method that manipulates a string of numbers in a manner similar to how chromosomes are changed in biological evolution. An initial population made up of strings of numbers is chosen at random or is specified by the user. Each string of numbers is called a "chromosome" or an "individual," where each number slot is referred to as a "gene." A set of chromosomes forms a population where each chromosome represents a given number of traits that are the actual parameters being varied to optimize the "fitness function". The fitness function is a performance index that we seek to maximize.

The operation of the genetic algorithm proceeds in steps. Beginning with the initial population, "selection" is used to choose which chromosomes should survive to form a "mating pool." Chromosomes are chosen based on how "fit" they are (as computed by the fitness function) relative to the other members of the population. More fit individuals retain more copies of themselves in the mating pool so that they will have greater representation in the next generation. Next, two operations are taken on the mating pool. First, "crossover" (which represents mating, the exchange of genetic material) occurs between parents.

In crossover, a random spot is picked in the chromosome, and the genes after this spot are switched with the corresponding genes of the other parent. Following this, "mutation" occurs. Mutation represents the change of values of randomly selected genes in a chromosome. After the crossover and mutation operations occur, the resulting strings form the next generation and the process is repeated. A termination criterion is used to specify when the genetic algorithm should end (e.g., the maximum number of generations or until the maximum fitness exhibits little or no change over a certain number of generations).

The following characteristics are also considered advantages of the genetic algorithm:

- the genetic algorithm works on a population of points, not a single point,
- they work directly with strings of characters representing the entire parameter set, not the individual parameters,
- the search is guided by probabilistic rules, not deterministic rules. The inherent randomness in this procedure allows the genetic algorithm to escape local maxima,
- genetic algorithms, like simulated annealing represent a form of optimization that does not require derivatives. The genetic algorithm only requires information about how fit a given solution is, i.e., the effect of the solution on the fitness function.

The construction of good fitness functions for this application requires insight in four areas, with the rules being derived from geometry, physics, engineering, and military doctrine. Several classes of fitness functions are being explored. The fitness functions tend to be highly nonlinear and non-differentiable at many points. For classical optimization algorithms, the non-differentiability might have posed a problem, but it offers no difficulty for a genetic algorithm.

The fitness functions currently being explored are expressible mathematically as a linear combination of products of Heaviside step-functions [7]. The step function arises from the rule-based origin of the fitness functions. The arguments of the fitness functions are given by the difference of the membership function and a parameter characteristic of expertise. The linear combinations of products of the step functions are typically averaged over an ensemble of kinematic scenarios, where each element of the ensemble differs from the others in terms of initial conditions. For example, the ensemble used to optimize the membership function for the root concept "close" consists of elements with different initial values for range, and its first two derivatives with respect to time. From these initial values, the range and range rate are calculated as a function of time allowing the membership function for "close" to be optimized over many physical scenarios. This is referred to as a geometric-kinematic ensemble. Despite the complicated non-linear form that the fitness function takes because of the rules used in its construction, genetic algorithm based optimization has proven to be effective.

The method described above for constructing fitness functions is only a first step. The fitness functions constructed in this manner, are most applicable to isolated platforms. The ultimate goal is to construct a resource manager/scheduler that is optimal in its performance when dealing with multiple dissimilar platforms. By pursuing the isolated platform model first, the region of parameter space that must be explored for the multi-platform problem is reduced. It would be expected, on intuitive grounds, that parameters for the multi-platform problems should lie within some neighborhood, of those solutions for the isolated platform model. The motivation for this assumption is that at any given time, each platform may be called upon to defend itself. Once the isolated platform parameters are selected for each root concept membership function, neighborhoods around these parameters can be defined, and a parameter space for the multi-platform problem formed by constructing a product space from the coordinate spaces defined by each isolated platform neighborhood. Therefore, the potentially large parameter space that must be explored for the multi-platform problem is constrained through the use of a priori information, significantly reducing the run-time of the genetic algorithm. This procedure has proven effective in producing very high quality multi-platform performance. The performance of the model and the potential risk of restricting parameter space in this way will be examined in a future paper [4].

4. An Example of Multi-platform Response

In this section a specific example of the fuzzy RM's ability to optimally allocate electronic attack resources is examined. Input requirements and output characteristics are considered, and illustrated through the actual output of the current implementation of the RM.

4.1 Input Scenario

The fuzzy RM requires as input, the position and number of ally platforms, e.g., ships, planes, etc., also emitter range, bearing, heading, elevation, and an emitter ID with an uncertainty associated with the ID. The effect of the data is to stimulate the various kinematic concepts like "close" resulting in different "actions" by the algorithm. The emitter ID is used to determine the technique or techniques (for ID's with uncertainty) that the ally platform or platforms can execute against the emitter.

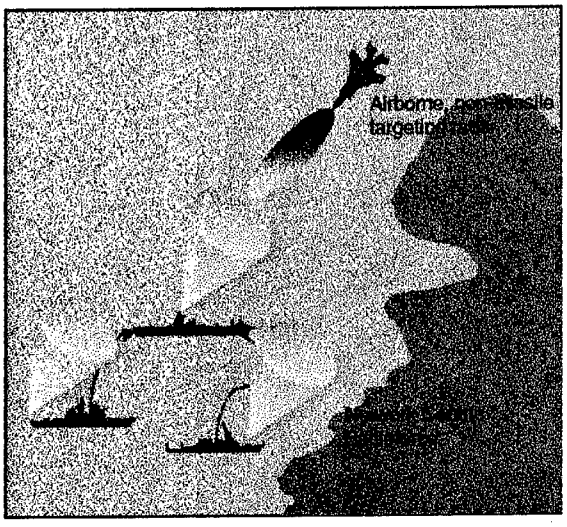


Figure 2: The fuzzy RM allocates EA resources distributed over three ships against a targeting radar with uncertain ID.

In Figure 2, there is a battleforce of three ships and also an incoming aircraft with targeting radar. However, in this scenario, the type of the threat emitter is not well-known. With the threat's classification not being well-known, and because the uncertainties indicate a foe of some type, all three ships conduct joint EA against the threat emitter.

The ship acting as command ship sends communication over the network to other adjacent ships asking for joint EA and chooses the electronic counter measures (ECM) technique most likely to be effective against this type of threat. The adjacent ships choose two other ECM techniques based on the emitter's ID and its uncertainty.

It should be noted, each ship has the same software aboard, and can act as a command ship. This significantly reduces the likelihood of the battlegroup being rendered ineffective by the loss of a single platform.

4.2 Output of the Fuzzy RM

In Figure 3, the algorithm's output for the scenario in Figure 2 is displayed. A polar plot with origin at the centroid of battlegroup is used to display the positions of the three ships (diamonds), the incoming emitter (triangle marked with designation "foe type"), and friendly aircraft (triangles marked with the designation "friend type"). Communications and electronic attack techniques used by each ship are listed to the side. The arrows running from the ships to the foe-type emitter indicate electronic attack.

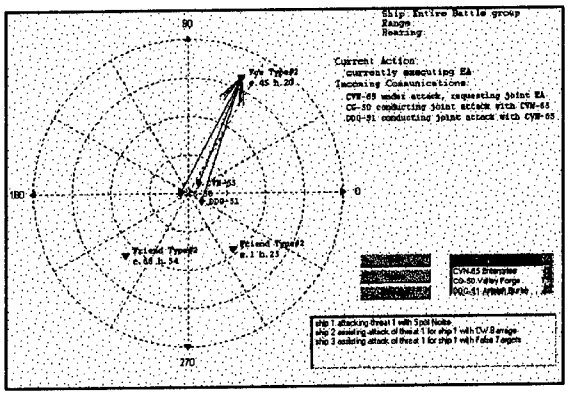


Figure 3: The algorithm's output showing how it allocates EA resources distributed over three platforms.

The algorithm, during its real-time run, displays an image of this type every second. As indicated in the box in the right-hand corner of Figure 3, the algorithm chooses the appropriate techniques for all three attacking ships. As consistent with military doctrine, all three ships are conducting joint EA. Finally, it should be noted there are two friendly aircraft in the scenario. The algorithm will not attack an emitter based on kinematic properties if the emitter has been clearly identified as a friend.

The algorithm has been determined to be effective by comparing its output to the judgement of human experts. Statistical evaluation of the algorithm's effectiveness will be published in the near future [4].

5. Dealing with Imperfect Association

It is assumed that the data provided as input to the RM has already been associated, i.e., the appropriate ESM and radar data have already been perfectly assigned to the same emitter. Association of the ESM and radar data is valuable since radar provides range and bearing information for use in the root concept "close" and ESM can provide ID, bearing, RF and PRI of the emitter. Unfortunately, the association of ESM and radar is generally not perfect given the sparse, intermittent and noisy nature of data.

The abilities of two different association algorithms to associate data as a function of the measured ESM points will be compared. These algorithms are the fuzzy association algorithm described in reference [8-12] and a Bayesian philosophy algorithm described in reference [13] and referred to here as the TW-algorithm.

The two association algorithms are compared using the same simulated ESM and radar data. The emitter has a bearing of 0 degrees. This is absolute truth for this simulation. Radar has determined there are objects traveling with bearings of 0, 1, and –1 degrees. For simulation purposes zero mean Gaussian noise with 1 degree standard deviation is added to simulate noise in the ESM measurement process. This is a difficult association problem since there are radar measurements not only at 0 degrees, but also radar measurements within one standard deviation of truth.

Since the radar measurements contain truth it is expected that a good association algorithm will associate the zero degree radar track with the ESM data. A probability of association between each radar track and the ESM data is calculated as in references [8-13]. Both algorithms give rise to five hypothesis classes describing whether or not the ESM data is associated with a radar track. It is desirable that when radar contains "truth," i.e., in this case the zero degree track, the track corresponding to truth, be firmly correlated with the ESM data. In this way the probability of making an inappropriate assignment of range is minimized. The notion of firm correlation is defined in detail in the references [8-13]. The other hypothesis classes will not be displayed, as they are not interesting for the example that follows and only serve to obscure the results.

Both the fuzzy association and TW-algorithms can be used to associate noisy ESM and noisy radar measurements [8-12]. The radar measurements for radar track j at time t_i will have zero mean Gaussian noise added to them. The variance of the noise will be denoted as σ_{ij}^2 for the j^{th} radar track at the i^{th} time.

Figure 4 presents results for three radar tracks with the following bearings: $\mu = 0^o,\ 1^o,\ -1^o$ with σ_{ij}=0.1° for all times t_i and radar tracks j. The radar noise standard deviation is consistent with levels found in modern radar systems. Since the radar results contain truth, i.e., a target moving with constant bearing of 0^o a good association algorithm will establish that there is a firm correlation between the ESM data and the 0^o bearing track. Figure 4 plots the probability the association algorithms establish a firm association between ESM data and the radar measurements. The fuzzy association algorithm results are given by the curve marked with *o*'s and the TW results are indicated by the curve marked with *+'s*. The vertical axis indicates probability of firm correlation and the horizontal axis the number of data points necessary to establish that level of probability.

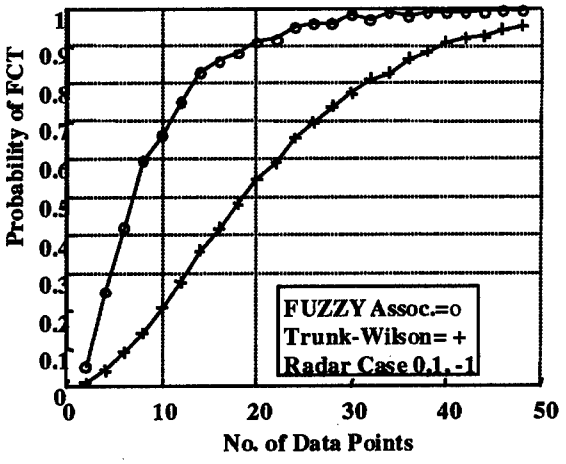


Figure 4: Fuzzy and Bayesian association

The fuzzy association algorithm results are always superior to the TW-algorithm. At ten data points the fuzzy algorithm has established a 65% probability of firm correlation, between the ESM data and the 0° radar track. The TW-algorithm requires about 24 points to establish the same level of probability of FCT. The fuzzy algorithm establishes an 80% probability of FCT by the 12th data point,

whereas the TW-algorithm requires about 30 points to reach the same level of success. The fuzzy algorithm reaches 90% probability of FCT at 20 data points and the TW-algorithm at about the 38th point. Therefore, the fuzzy algorithm establishes high probabilities of firm correlation with between 1/3 to 1/2 the data required by the TW-algorithm. In this sense the fuzzy algorithm is 2 to 3 times faster than the TW-algorithm. Also, this is a difficult example for any association algorithm since there are two additional radar measurements within one noise standard deviation. The results are only slightly inferior to the case where radar is simulated as noiseless as found in reference [11]. The ability of the fuzzy algorithm to make high quality decisions with much less data than the TW-algorithm is significant since real data is frequently sparse and intermittent.

The above examples are for the case where there is 100% detection of ESM and radar data. In reference [11] it is shown with a detection rate as low as 70% of the ESM points, the fuzzy association algorithm experiences little deterioration, whereas the TW-algorithm's performance is greatly degraded.

The example in Figure 4 is for the case of a single emitter. In reference [11] it is shown that the fuzzy association algorithm gives a similar level of performance if there are one, four or 10 emitters, even when ESM detection rates drop down to 70%. In particular, for 10 emitters closely spaced in the RF-PRI plane the fuzzy association algorithm displays results like those found in Figure 4, but the TW-algorithm deteriorates more than 40% by the 48th data point.

The use of the fuzzy association algorithm will allow association decisions to be made with 1/6 to ½ the data required by the Bayesian association algorithm. Faster association of ESM and radar tracks means better assignment of range and ID's to potential threats. As a final observation, the use of both a fuzzy RM and a fuzzy association algorithm would allow linguistic data to be shared between the two algorithms. This should increase the effectiveness of both algorithms. The easy sharing of linguistic rules and other linguistic data is not an option, if a non-fuzzy association algorithm like the TW algorithm were to be used for association.

6. Future Developments

There are several activities that will be conducted in the near future, which include: expansion of the rule set, research related to improved optimization, expansion of the technique library, the invention of new multi-platform electronic attack techniques which make good use of the resources distributed over multiple platforms, and validation of the multi-platform resource manager.

7. Conclusions

A fuzzy logic based algorithm for optimal allocation and scheduling of electronic attack resources distributed over many platforms is under development. The kinematic-ID subtree that forms the core of the isolated ship model has been discussed and used to illustrate the mathematical concepts involved. Root concept membership function construction has been discussed. Optimal performance for the algorithm is obtained by selecting values of the free parameters in the root concept membership function using a genetic algorithm. The use of a genetic algorithm requires the construction of a fitness function. The fitness functions constructed for this task are based on insights obtained from geometry, physics, engineering, and military doctrine. The fitness functions are in general non-differentiable and highly non-linear, neither property providing an obstacle for a genetic algorithm. Finally, fuzzy logic based multi-sensor association should prove very effective both in its ability to form high quality conclusions faster than a standard Bayesian algorithm and because it allows linguistic data to be shared easily between the resource manager and the multi-sensor association algorithm.

8. Acknowledgments

This work was sponsored by the Office of Naval Research. The author gratefully acknowledges useful discussions with Mr. Ed Khoury and Dr. Joseph Lawrence III.

9. References

[1] J.M. Molina Lopez, F.J. Jimenez Rodriguez, and J.R. Casar Corredera, "Symbolic Processing for Coordinated Task Management in Multiradar Surveillance Networks," *Fusion98, Proceedings of the International Conference on Multisource-Multisensor Information Fusion*, R. Hamid, and Dongping Zhu, Vol . II, pp. 725-732, CSREA Press, Las Vegas, Nevada, USA, July 6-9,1998.

[2]L.H. Tsoukalas and R.E. Uhrig, *Fuzzy and Neural Approaches in Engineering*, (John Wiley and Sons, New York, 1997).

[3]H. J. Zimmerman, *Fuzzy Set Theory and its Applications*, page 11,(Kluwer Academic Publishers Group), Boston, 1991.
[4] J.F. Smith, III and R. Rhyne, II, "A General Fuzzy Logic Resource Manager and Communication Issues for Distributed Resources," to be published, 2000.
[5] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in Fortran: The Art of Scientific Computing*, (Cambridge University Press, 1992).
[6] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning* (Addison-Wesley, Reading, MA, 1989).
[7] A.I. Saichev and W.A. Woyczynski, *Distributions in the Physical and Engineering Sciences*, (Birkhauser, 1997).
[8] J. F. Smith III, "A Fuzzy Logic Multisensor Association Algorithm: Theory and Simulation," Naval Research Laboratory Technical Formal NRL/FR/5740--97-9866, Naval Research Laboratory, Washington D.C., 20375-5000, 9/97
[9] J. F. Smith III, "A Fuzzy Logic Multisensor Association Algorithm," *Signal Processing, Sensor Fusion, and Target Recognition VI*, I. Kadar, Vol. 3068, pp. 76-87, SPIE Proceedings, Orlando, FL, April 21-24, 1997
[10] J. F. Smith III, "A Fuzzy Logic Multisensor Association Algorithm Dealing with Multiple Targets Intermittent Data and Noise" , *Signal Processing, Sensor Fusion, and Target Recognition VII*, I. Kadar, Vol. 3374, pp. 2-13, SPIE Proceedings, Orlando, FL, April 13-17, 1998.
[11] J.F. Smith III, "A Fuzzy Logic Multisensor Association Algorithm: Applied to Noisy, Intermittent and Sparse Data", Fusion98, Proceedings of the International Conference on Multisource-Multisensor Information Fusion, , R. Hamid, and Dongping Zhu, Vol . II, pp. 681-687, CSREA Press, Las Vegas, Nevada, USA, July 6-9,1998.
[12] J. F. Smith III, "A Fuzzy Logic Multisensor Association Algorithm: Multiple Emitters, Computational Complexity, and Noisy Data", Naval Research Laboratory Technical Formal Report, Naval Research Laboratory, Washington D.C., 20375-5000, to be published 1999.
[13] G.V. Trunk and J.D. Wilson, "Association of DF Bearing Measurements with Radar Track," IEEE Trans on AES, Vol. AES-23., No. 4, July 1987

Session WA1
Image Fusion II
Chair: Per Svensson
Defence Research Establishment, Sweden

Matching Segments in Stereoscopic 3D Reconstruction.

Andre BIGAND (*), Thierry BOUWMANS (*), Jean Paul DUBUS (**)
* LASL, Universite du Littoral, BP649, 62228 Calais Cedex, France
** Laboratoire ID3, USTL, 59655 Villeneuve D'Ascq Cedex, France

Abstract ***The importance of 3D data acquisition is widely recognized in robotics field. One approach is to measure the distance on the basis of triangulation principle from the disparity of two images. This stereo method has, however, a difficult problem that is to find correspondence of features between two images. This correspondence problem can be solved geometrically by adding one more camera (trinocular vision) or dealing with the depth uncertainty using an adapted aggregation operator. This paper presents the application of the fuzzy logic for the correspondence between features, for static and dynamic 3D structure in an industrial environment. The aim of the work is to propose a fuzzy 3D sensor for metrology and multimedia applications with manufactured objects. For recognition of 3D shape and measurement of 3D position it is important that a vision system can measure the 3D data of dense points in the scene. At first we remind the different stages of a multi-sensory system for 3D reconstruction, then we remind the problem of features matching. These applications need to obtain a good precision of the 3D representation, so all the algorithms are treated in this way (camera calibration, ...), and the use of fuzzy logic methods is generalised to obtain a good robustness of the algorithms. Results are presented, using standard cameras and comparating two fuzzy aggregation operators : OWA and fuzzy integral. .***

Keywords : fuzzy logic, image fusion and machine vision, manufacturing.

1 Introduction

Information fusion is an important aspect of any decision system. Dealing with multiple input information sources is that the information coming from individual source is either incomplete or noisy that is , uncertain or imprecise. Numerous image processing systems or computer vision systems (pattern recognition, scene analysis, image processing, 3D reconstruction,...) belong to this category of decision taking problems. This paper aims with the stereo matching problem, that is obtaining a correspondence between (linear) features in right and left images, and treated like a decision problem , related with industrial images (polyhedric scene analysis). At the end of the low-level image treatment, we obtain linear primitives (segments) known with some imprecision on their geometric characterization, so existing matching methods (like dynamic programming,...) have to deal with uncertainty. Dynamic programming techniques have been used to handle this search efficiently. Nevertheless, due to the nature of the problem and its uncertainty, we have verified an improvement by using a fuzzy decision tool by the mean of a hierarchical tree testing the attributes we defined previously. We first order the segments in the two images, then we test the similarity of these ordered segments by two attributes along the epipolar line (length and orientation), each attribute being scoring by a fuzzy measure. We keep the best matched segments. We have compared fuzzy connectives (classical average operators, fuzzy integral) with a classical matching algorithm and we prove that in particular fuzzy integral is better in matching noisy

segments by reducing uncertainty on this operation. Indeed the decision (at each step of the algorithm) is a one shot-decision and the Choquet integral-based utility, a generalisation of expected utility that is sum-decomposable for such acts in numerical framework, is well adapted to improve existing strategies by considering of approximate data (that gives uncertain decision). Fuzzy average operators method and classical method give about the same results in that case.

2 Description of the method

The different stages met in multivision are the following ones : calibration, localisation, segmentation, fuzzy extraction segment, establishing correspondence, 3D reconstruction. The calibration method used is global. This method consists in determining a characteristic matrix of the camera called matrix of calibration. This method was developped in ([3]). The segmentation (using a fuzzy clustering method) used is devellopped in Bezdek([1]), Bouwmans([3]). We just remember here the principles used and the advantages of the developped method. At first, from the initial image, we apply a fuzzy geometrical area extraction; This method permits to obtain the edges of fuzzy images, like fuzzy Hough methods, ([6]); But we know that these types of methods are sensitive to lightning variations, so we associate the geometrical area extraction with a fuzzy clustering method, up to increase the accuracy and the reliability of the segmentation. Once made the segmentation of the image (our method doesn't need edge following), the different images are represented by multiple segments, primitives that we are able to match. Even if these primitives (linear segments) are well located, it remains some imprecision on the length and on other geometrical attributes. So we have to deal with this imprecision that makes the stereo correspondence problem uncertain .

3 Feature matching method

Many pattern recognition problems can be simplified as line pattern matching task. Once segmentation made (and movement detection if necessary), we have to match the 2 (or 3 or n) images. This stereo correspondence problem can be defined in terms of finding pairs (or more) of true matches between features (here edge segments) that satisfy some competing constraints : ordering, similarity, smoothness, uniqueness, ([7]). Due to the uncertainty of traditionnal segment matching, we decided to use fuzzy data fusion method, presented by I.Bloch ([2]), for 3D reconstruction. A comparison of fuzzy operators, depending on their behaviour, and the dependence of these operators on conflict and on source reliability was presented in ([2]); ([8], ([4]) operate with these fuzzy operators to solve matching problems. Grabisch ([5]) shows that the Choquet integral, used with Sugeno measures, is sum-decomposable and is a generalisation of OWA operators in numerical frameworks. We present in this paper the results of fusion of 2 (or 3) images using T-norms and T-conorms, for images taken in a factory under varying lightning, the choice of fuzzy operators being important to reduce traditionnal conflicts of matching, and then using fuzzy integral . We use for T-norms "sqrt(xy)" and for T-conorms "(x+y)/2". The choice of the operators are depending on the importance of the choosen attributes, for matching images with accuracy and robustness. These operators permit to compare a segment of left image with segments of right image (or inverse).

3.1 Attributes

Our stereo matching algorithm uses edge segments as primitives (features) and the problem is seen as a multistage (hierarchical) decision process. From 2 (or 3) images to match, left and right, a number of features are to be extracted. The segments are first ordering in each image using the epipolar constraint. Then the similarity is verified by

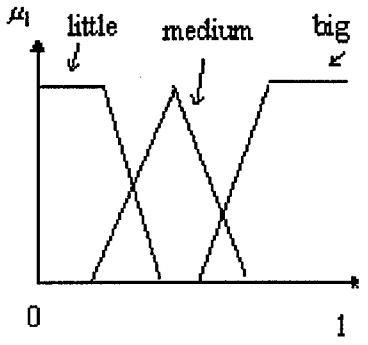


FIG. 1: Confidence measure on the attribute "length"

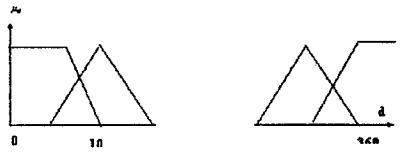


FIG. 2: Confidence measure on the attribute "orientation"

three attributes for each edge segment (length, direction, and the proximity), to optimize matching procedure. For each attribute, we associate a fuzzy measure on which we apply fuzzy operators. For the length, we use three clusters : little, medium, big (figure 1). For the direction, we use 36 clusters (from 0 to 360 degrees), (figure 2).

3.2 Fuzzy rule base

For each attribute, we calculate a criteria for the matching of edge segment in the both (or 3) images : criteria(i)= 1- S(mj-mk), with i the number of the attribute, mj the measure of the first attribute in the first image, mk the measure of the first attribute in the second image,... Each criteria is integrated in a fuzzy rule base : If criteria(1) is A and criteria(2) is B and... then the choosen edge segment : length is A and direction is B and...

We have choosen as sense for the implication "then" the Kleene-Dienes implication, because all the criteria bring redundancy on the information and we need a good coherence on the fuzzy rule base to keep the good candidates for matching. So the fuzzy rule base furnishes (from the primitives of each image) edge segments with their 3D coordinates for the 3D reconstruction that follows the matching operation, using a hierarchical model : At first, we test length and orientation of the segments of the two (or three images), to obtain a set of candidates for segment matching, and then we test the proximity, up to obtain only one pair of segments to match. If the uniqueness is not respected we pursue the treatment in the other sense (for instance we first test the right image with the left one and then the left one with the right one). And then we compared this method with a Choquet integral method associated with Sugeno measures obtained with an a-priori information. In the other hand, the Choquet Integral method is a global method (All the attributes are tested at the same time) compared with a fuzzy rule base.

4 Results

The images are taken in a noisy environment (real images). The algorithm of segmentation has found linear segments from a calibration feature that is difficult to reconstruct because all the segments are paralell. We present in the next pictures (figure 3, figure 4, figure 5), the 3D reconstruction of a classical frame from its right and left images, using only two attributes for these segments (length and direction). So we show the strength of this type of fuzzy rule base, in comparison with classical methods using these two attributes. The accuracy of the method is increasing by using a third camera and the other attibutes of each primitive in the image.

To compare OWA operators and fuzzy integral we show in the next figures, (figure 6) for OWA, (figure 7) for fuzzy integral, the measures associated to ten segments in each image

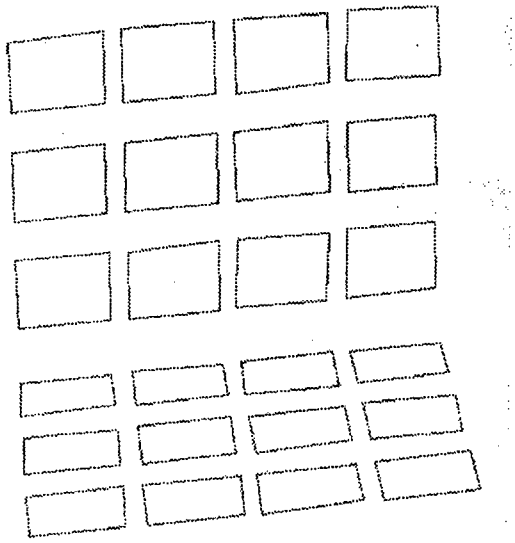


FIG. 3: Right image

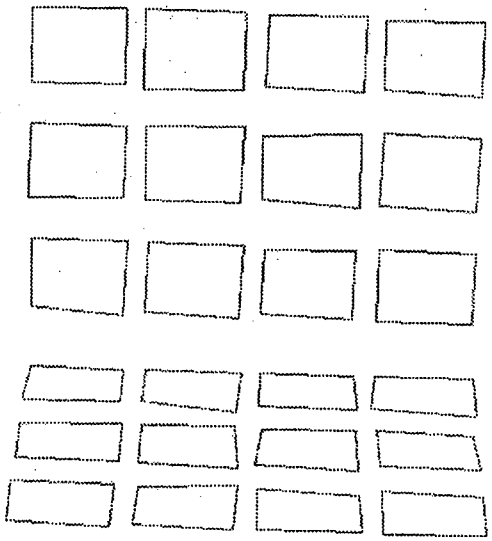


FIG. 4: Left image

at the beginning of the decision process, and figure (8) and figure(9) the same measures at the end of the decision process. We can explain these figures like the confidence degree function we associate to the imprecision we have along the decision process. We can notice that (figure 7) the measures are non-additive (the sum is equal to 1.2). At the end of the process there is only one pair of segments matched. We can also notice that there is left more information at the end of the process using the fuzzy integral. We have verified this fact in generating " noise " by adding one false segment in only one image. At the end of the first step (right image to left image), 90 per cent of the segments of the right image are well appaired by the OWA operators and 100 per cent by fuzzy integral.

5 Conclusion

We have experimented with camera calibration, stereoscopic vision and reconstruction with standard hardware (cameras and image digitizer) on an industrial piece and fuzzy software in order to build a "3D fuzzy studio". Results show that this method for matching homologous objects can be proposed. A good choice of fuzzy operators permit to reduce the matching conflicts of traditionnal matching method. We have shown that OWA operators are effective as well as fuzzy integral to match linear features but the fuzzy integral is more effective when there is a great uncertainty for matching (see figure (9)). The interesting point here is the fact that we have got good precision relevant for industrial measure in productics, so it can be applied to 3D inspection. CPU's time is too reasonable (it is often a problem for fuzzy method). We are now optimizing the different procedures we used, up to propose an optimum tool for industry inspection.

6 References

Références

[1] J. C. Bezdek, C. Coray, R.Gunderson, and J. Watson, *"Detection and characterization of cluster substructure", Appl. Math, vol 40, pp. 339-372, (1981).*

[2] I. Bloch, *"Information combination operators for data fusion : a comparative review*

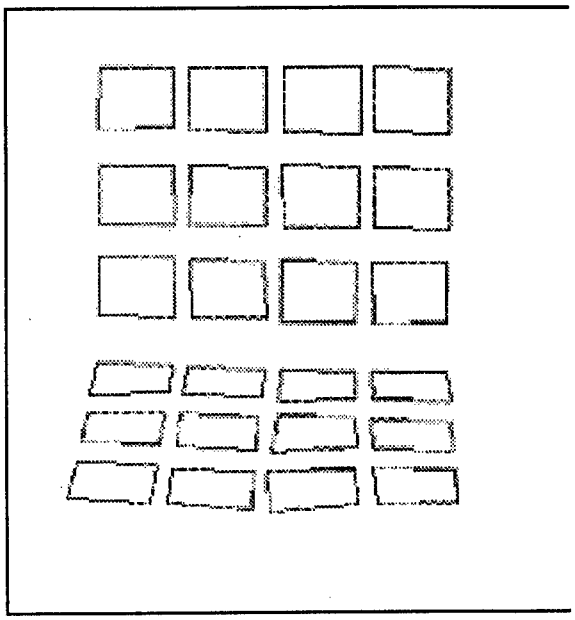


FIG. 5: 3D reconstruction of the frame

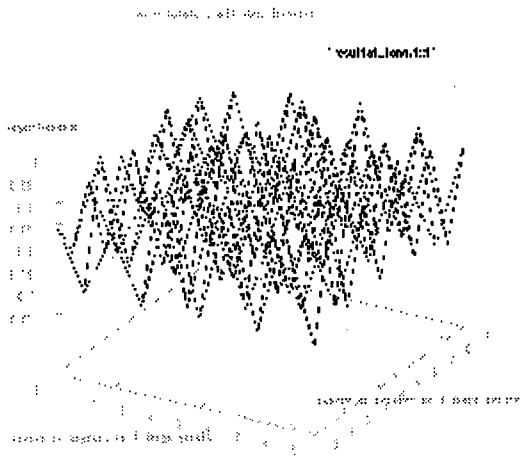


FIG. 6: Confidence function for OWA operator

with classification", ENST, communication interne, (1994).

[3] T.Bouwmans, A.Bigand, J.P.Dubus *" Application of fuzzy logic in multivision : interest and result", EUFIT'96, Aachen, Germany.*

[4] F.H.Cheng, W.H.Hsu, C.A.Chen *" fuzzy approach to solve the recognition problem of handwritten Chinese characters " Pattern Recognition, 22(2) :133-141, 1989.*

[5] M.Grabisch *" Fuzzy integral as a flexible and interpretation tool of aggregation " in " Aggregation and fusion of imperfect information ", Studies in fuzzyness and soft computing editing by B.Bouchon-Meunier, Physica-Verlag.*

[6] J.H.Han *"Fuzzy Hough Transform", Pattern Recognition Letters , vol.15, (1994).*

[7] Marr.D, Poggio T. *" A computational theory of human stereovision " , Proc. Royal Society of London, Vol.B207, pp.301-328.*

[8] H.Ogawa *" A fuzzy relaxation technique for partial shape matching " Pattern Recognition Letters, 15(4) :349-355, 1993.*

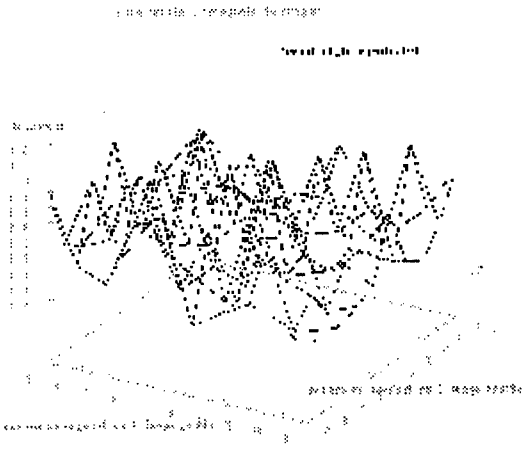


FIG. 7: Confidence function for Choquet Integral

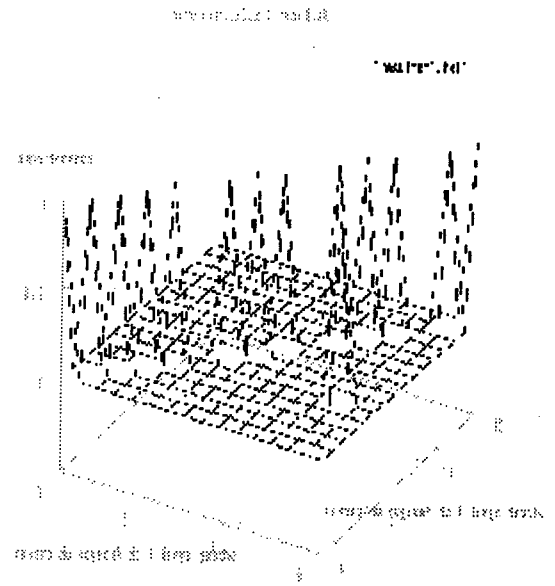


FIG. 8: Results of confidence function, OWA operator

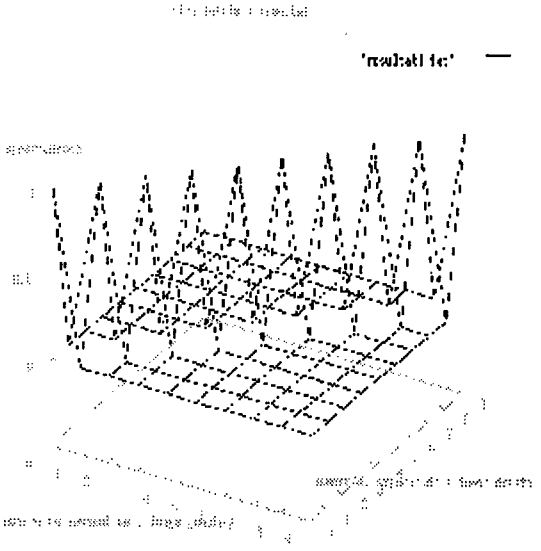


FIG. 9: Results of confidence function, Choquet Integral

Estimating the Distance between two robot arms by fusing the views of uncalibrated cameras*

Christian Scheering, Jianwei Zhang and Alois Knoll
Faculty of Technology,
University of Bielefeld, 33501 Bielefeld, Germany

Abstract *We propose an approach for estimating the distance of two moving robot arms based on the fusion of vision data. The first component of the method is the projection of high-dimensional input data into a subspace generated by a Principal Component Analysis (PCA). We show that complex sensor data can be efficiently compressed if robot movements are constrained to a local scenario. The second component is an adaptive B-spline neuro-fuzzy controller whose input space is the subspace and whose outputs are the estimated robot distances. The B-spline model is trained for smooth and correct interpolation. In the online application phase, through the cascaded two components, a sensor pattern can be mapped into the distance space. Our experimental setup consists of a two-arm robot system with an "overhead" and a camera looking at the scene from the side. Implementations with different motions show that the method works even if no robust geometric features can be extracted from the sensor readings.*

Keywords: uncalibrated vision, sensor fusion, learning neuro-fuzzy model, two arm distance estimation, collision avoidance

1 Introduction

The estimation of distances between a robot and its environment provides the basis for detecting potential collisions between two arms. We present a self-learning minimal distance estimation scheme for a two-arm robot using two uncalibrated cameras for achieving this goal. Common approaches of collision detection employ simplified geometric models of arms and the (reconstructed) environment. In [1] deformable protection zones are used to detect the collision between the two robot arms. In [2] the geometry of a dual-arm robot is approximated by a set of spheres. A 2D geometric model is utilised in [3] by constraining the area in which a collision might occur. [4] presented an obstacle count independent method which is based upon voxel-map and spherical representation.

These approaches rely on an a-priori modelling or reconstruction of the robots workspace. A different approach is to use *sensors* to detect collisions between the arms. In [5] a single arm is instrumented with infrared proximity sensors, [6] describes a reactive approach to sensor-based collision avoidance and [7] presents a method for real-time collision avoidance for a whole-sensitive arm whose whole bodies are covered with a sensitive skin sensor to detect objects in its direct vicinity.

Others [8, 9, 10, 11] extract geometrical meaning from images. The desired task is then performed by exploiting the obtained geometrical representation. A problem to be solved is how to model and extract meaningful information from multiple images such as characteristic points or objects. This makes some kind of target identification or prior marking of points necessary.

*The work described in this paper is funded by the Deutsche Forschungsgemeinschaft in the project SFB 360/D4.

In our approach we use a *trained* hand-eye transformation to estimate distances. Learning of vision-based positioning based on visual appearance information was introduced in [12]. A parametric eigenspace representation is used for describing the different objects as well as object locations. The positioning problem is thus transformed into finding the minimum distance between a point and a manifold in the eigenspace.

In this paper, we propose a multiple-view fusion-scheme by combining an adaptive fuzzy controller with principal component analysis as a dimension-reduction technique in order to estimate the minimal distance between two robot manipulators. Images obtained by two uncalibrated cameras are represented as single vectors of very high dimension and are projected into a low-dimensional subspace without any further geometric feature extraction or three-dimensional reconstruction. In the off-line phase these projected images form the input to the fuzzy-controller which in turn *learns* the relationship between images and arm-positions.

In the following section, we first introduce the employed model. In the experimental part we present and compare different results from a distance estimation during a complex rotational motion.

2 Sensing-Estimation Model

Our goal is to develop a supervised training-scheme, which enables an adaptive system to learn the relationship between two arbitrary camera-views and the two-arm distance. To achieve this we build a neuro-fuzzy controller with B-spline basis functions whose output control vertices are determined by training. It is well-known that general neural or fuzzy systems with a large number of input variables suffer from the problem of the "curse of dimensionality". If no additional image processing is performed, then for grey-level images such as ours with a size of 192×144 pixels a control system with more than 27,000 input variables would have to be modelled. Therefore it is essential to *reduce* the number of inputs but with as little information loss as possible. A well-known technique for dealing with multivariate problems in statistics is *principal component analysis* (PCA). Until now, it has been mainly applied to data compression and pattern recognition [13]. Our findings indicate, however, that this technique is also suitable for reducing the dimension of the input space of a general control problem. Depending on how "local" the measuring data are and therefore how similar the observed sensor patterns look like, a small number of eigenvectors can provide a good "summary" of all input variables. It is possible that three or four eigenvectors supply the most information indices of the original input space. An efficient dimension reduction can be achieved by projecting the original input space into the eigenspace. This step is illustrated in the left part of Fig. 1($n \ll m$).

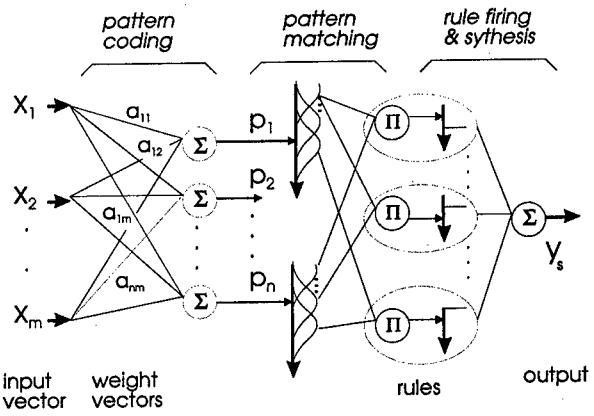


Figure 1: The task-based mapping can be interpreted as a neuro-fuzzy model. The input vector consists of pixels of a brightness image. Pattern coding is through PCA and projection.

Partitioning of eigenvectors can be done by covering eigenvectors with linguistic terms as shown in the right part of Fig. 1. In the following implementations, fuzzy controllers constructed according to the B-spline model are used [14]. This model provides an ideal implementation of CMAC as proposed by Albus [15]. We define linguistic terms for input variables with B-spline basis functions and for output

variables with singletons. Such a method requires fewer parameters than other set functions such as trapezoid, Gaussian function, etc. The output computation becomes very simple and the interpolation process is transparent. We also achieved a good approximation capability and rapid convergence of B-spline fuzzy controllers [14]. In the online application, the input data are first projected into the eigenspace and then mapped to output based on the fuzzy control model, Fig. 1.

2.1 Dimension Reduction

Let us assume k energy-normalised sample input vectors $\vec{x}^1, \ldots, \vec{x}^k$ with $\vec{x}^i = (x_1^i, \ldots, x_m^i)^T$ originating from a pattern-generating process. The PCA can be applied to them as follows [12]:

First the (approximate) mean value $\vec{\mu}$ and the covariance matrix $\mathbf{Q}$ of these vectors are computed according to

$$\vec{\mu} = \frac{1}{k}\sum_{i=1}^{k}\vec{x}^i, \mathbf{Q} = \frac{1}{k}\sum_{i=1}^{k}(\vec{x}^i - \vec{\mu})(\vec{x}^i - \vec{\mu})^T$$

The eigenvectors and eigenvalues can then be computed by solving $\lambda_j\vec{a}_j = \mathbf{Q}\vec{a}_j$, where λ_j are the m eigenvalues and $\vec{a}_j$ are the m-dimensional eigenvectors of $\mathbf{Q}$. Since $\mathbf{Q}$ is positive definite all eigenvalues are also positive. Extracting the most significant structural information from the set of input vectors $\vec{x}^i$ is equal to isolating the first n ($n \ll m$) eigenvectors $\vec{a}_i$ with the largest corresponding eigenvalues λ_j. If we now define a transformation matrix $\mathbf{A} = [\vec{a}_1 \ldots \vec{a}_n]^T$ we can reduce the dimension of the normalised $\vec{x}^i$ by

$$\vec{p}^i = \mathbf{A} \cdot (\vec{x}^i - \vec{\mu}); \quad dim(\vec{x}^i) = n \qquad (1)$$

The dimension n should be determined depending on the discrimination accuracy needed for further processing steps vs. the computational complexity that can be afforded.

Because of the high-input dimension using two camera-views we decided to calculate the salient eigenvectors using an iterative perceptron approach [16]. To calculate the first eigenvector the weights $\vec{w}$ of the single layer perceptron are randomly initialised ($\vec{w} \neq \vec{0}$) and a constant $\gamma, 0 < \gamma < 1$ is chosen. The update of $\vec{w}$ is calculated with a randomly chosen sample vector $\vec{x}^i$ as in [17]:

$$\Delta\vec{w} = \gamma \cdot \vec{w}\vec{x}^i \cdot (\vec{x}^i - \vec{w}\vec{x}^i \cdot \vec{w}) \qquad (2)$$

This step is repeated several times with decreasing γ. In order to calculate subsequent eigenvectors with this scheme, the projection of each sample $\vec{x}^i$ onto the found eigenvector is subtracted from the according sample.

3 Image Fusion

To obtain training images for the controller we move both robots to several different known positions, calculate and record the distances $\vec{d}$ as well as one image from each camera. Our idea of image fusion is then straightforward: we simply *concatenate* the images of the uncalibrated cameras and perform an overall PCA. With the concatenated and normalised images as $\vec{x}^i$ and the corresponding $\vec{d}$ a B-spline fuzzy controller is trained. We use third order splines as membership-functions and between 2 and 6 knot points for each linguistic variable. The distribution of these points is equidistant and constant throughout the whole learning process. The coefficients of the B-splines are initially zero. They are modified by the rapid gradient descent method during the training [14]. In the case of supervised learning, each learning datum corresponds to a supporting point in the control space. If a sensor pattern is taken online and its eigenvalues are calculated, the computation of the controller outputs may then be regarded as the "blending" of all the firing rules. The following steps are necessary:

1. Acquire new images.
2. Pre-process the data: concatenate images, normalise and subtract mean.
3. Project the input into the sub-eigenspace $(p_1, \cdots, p_n)$.

4. Compute the output by feeding the projection vector (principal components) into the trained fuzzy controller.

4 Experimental Setup

We used two 6 DOF Puma 260 controlled by RCCL [18]. Both are mounted up-side down as shown in Fig. 3 and their approach directions point towards each other. By defining an additional *virtual* third robot in between them it is very easy two move them both in a coherent fashion. The equations for the kinematic chain of both arms as depicted in Fig. 2 are:

$$\begin{aligned} T_{base}^{left} \cdot T_{6}^{left} \cdot T_{tool}^{left} &= T_{6}^{virt} \cdot T_{left}^{virt} \quad (3) \\ T_{base}^{right} \cdot T_{6}^{right} \cdot T_{tool}^{right} &= T_{6}^{virt} \cdot T_{right}^{virt} \end{aligned}$$

If the virtual robot is now moved (e.g. with a rotation around its roll-axis) than RCCL solves eq. (3) for T_6^{left} and T_6^{right} automatically. The robots are observed by two cameras one viewing from above and one from the side (s. Fig.4). During learning grey-level images with a size of 192×144 from each camera are obtained. The projection of the concatenated images into the eigenspace together with the appropriate distance-value forms the input data-set for the supervised learning-scheme described above. Each of the controller constructed for the experiments uses between four and six input dimensions (projection onto the eigenspace), one output (minimal distance), between three and six linguistic terms and B-splines with a degree of three. The appropriate combination of input dimensions and linguistic terms for each experiment were determined by exhaustive search over all possible controllers within the above intervals.

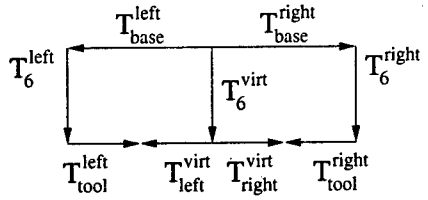


Figure 2: Kinematic chain of both arms.

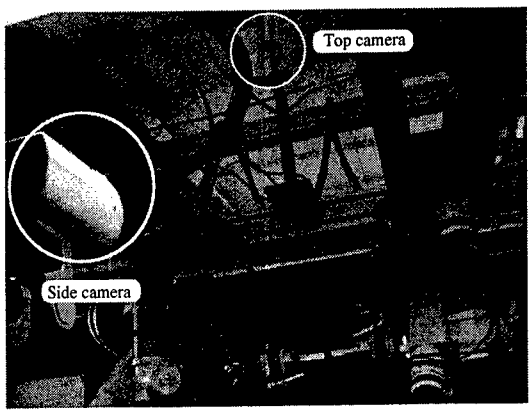
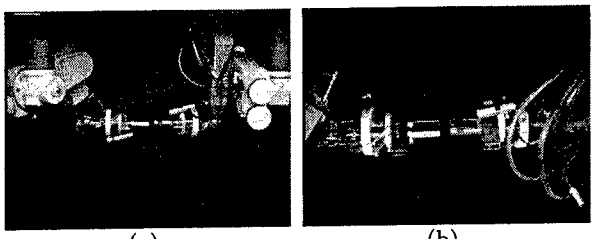


Figure 3: Global experimental setup view. The camera distance was approx. 1m.



(a) (b)

Figure 4: Side (a) and top (b) view of the experimental setup.

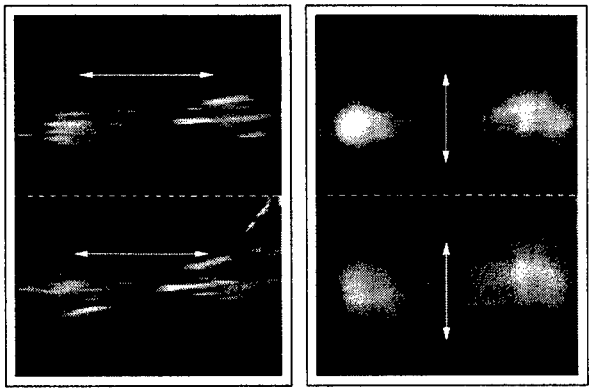
5 Experimental Results

We performed four different types of experiments to show the performance of our proposed method. The goal is to learn the distance-relationship between the two arms.

5.1 Experiment I

The first experiment was to move both robots horizontally only but with different distances between their tool tips. Since both motions were caused by moving the virtual robot, the arms maintained their relative position and therefore the minimal distance was simply between the tool tips. The virtual robot was moved along its x axis from position $\boldsymbol{p}_{start} = (-20, 300, 150)^1$ to position $\boldsymbol{p}_{end} = (10, 300, 150)$ within 20 equally spaced iterations. At each new position the distance between the robots was changed from $d_{min} =$ 10mm to $d_{max} =$ 90mm again in 20 equally

[1] All positions are in mm.



(a) Horizontal movement (Exp. I). Top view, side view.

(b) Vertical movement (Exp. II).

Figure 5: Concatenated mean images of top and side camera for exp. I and II. The arrows indicate the motion direction.

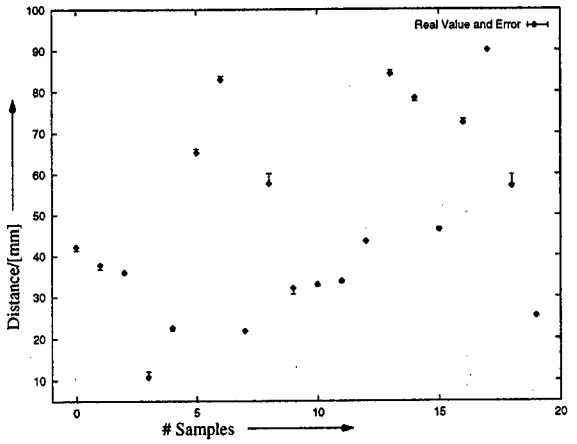


Figure 6: Error plot of exp. I. The dots indicate the real distance while the bars represent the estimation-error. The characteristic values are: the mean absolute error $\bar{e} = 0.85$mm, the standard deviation of the absolute error $\sigma = 0.67$mm and the maximum absolute error $\hat{e} = 2.70$mm.

spaced iterations, resulting in 400 images for each camera. Fig. 5(a) shows the resulting mean image for this motion. To test the controller after learning we generated 20 positions (pseudo) randomly on the line between $\boldsymbol{p}_{start}$ and $\boldsymbol{p}_{end}$. Additionally we set the distance to values between d_{min} and d_{max} randomly. As can be seen in the corresponding error plot (s. Fig. 6) the controller output is quite reasonable. Most of the samples are estimated near their true value yielding a mean error of $\bar{e} = 0.67$mm. The largest absolute error in this experiment is $\hat{e} = 2.70$mm measured at a tip-distance of $d = 57.18$mm.

5.2 Experiment II

The second experiment is orthogonal to the first one – the virtual robot is moved vertically along its z axis from position $\boldsymbol{p}_{start} = (0, 300, 110)$ to position $\boldsymbol{p}_{end} = (0, 300, 190)$. Again, at each position 20 images with different distance between 10 and 90mm are taken. Similar to the first experiment 20 random positions are calculated to evaluate the performance of the controller. Fig. 5(b)) shows the resulting mean image of this motion. In comparison to the former experiment the resulting error plot depicted in Fig. 7 shows larger deviations from the true values. The reason for this is that now a larger distance between $\boldsymbol{p}_{start}$ and $\boldsymbol{p}_{end}$ is possible (80mm compared to 30mm in the first experiment). On the other hand the interpolation is still good enough to check whether the robots are close to each other or far away.

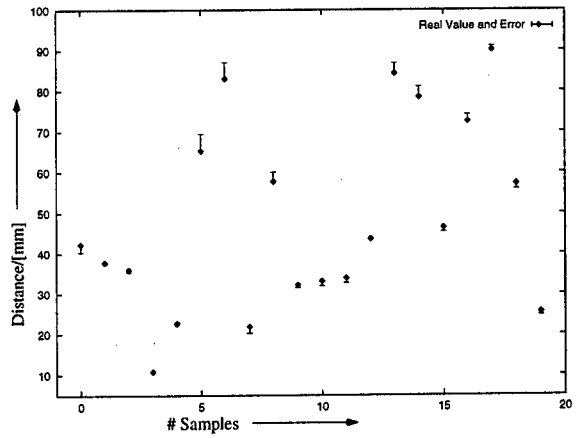
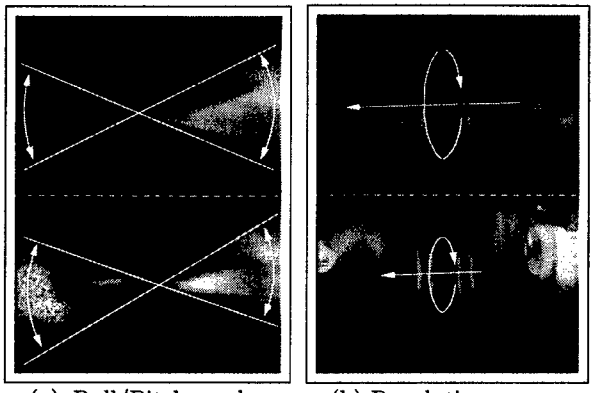


Figure 7: Error plot of exp. II, $\bar{e} = 1.47$mm, $\sigma = 1.13$mm, $\hat{e} = 4.13$mm.

5.3 Experiment III

In the third experiment we increased the degrees of freedom of the motion by rotating the virtual robot around its roll- and pitch-



(a) Roll/Pitch angles (Exp. III).

(b) Revolution (Exp. IV).

Figure 8: Mean images of exp. III and IV.

axis but maintaining its position. The approach directions of both arms still face each other (s. Fig. 10) and the distance calculation for training and evaluation purposes is still very easy. Again 400 training images were recorded. For 20 roll/pitch angle-pairs between $\varphi_{roll} = \varphi_{pitch} = \pm 10\text{deg}$, 20 different distances were measured (s. Fig. 8(a) for the corresponding mean image). To test the resulting controller, again 20 random-positions in between the training-interval were calculated. The re-

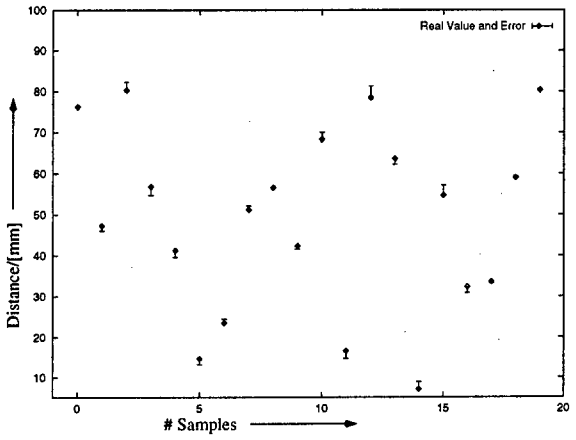


Figure 9: Error plot of exp. III, $\bar{e} = 1.29\text{mm}$, $\sigma = 0.77\text{mm}$, $\hat{e} = 2.80\text{mm}$.

sulting errors shown in Fig. 9 are better than those of the second experiment. The reason is that although the same distance interval as in the former experiment was used the position

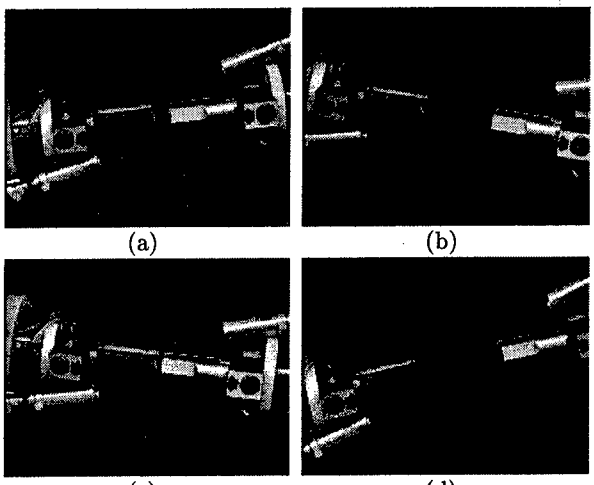


Figure 10: Although moving against each other the approach vectors of both arms maintain their relative orientation during the third experiment.

of the virtual robot remained constant. Therefore the input images are more similar to each other allowing a smoother interpolation by the controller.

5.4 Experiment IV

Common to the first three experiments is that there were no images in which an overlapping and therefore partial occlusion of one arm occurred. That means that in turn it would be possible to use a single view only in order to estimate the distance during these rather simple relative motions.

Therefore we increased the complexity of the examined situation in our last experiment further in order to benefit from the two independent views. Both arms performed a circular motion around each other. This results in a periodical overlap of parts behind and including the wrist as shown exemplary in Fig. 12(b) for the top-view. In this case a monocular distance estimation might be possible (e. g. using a model based approach) but rather difficult. But as will be seen in the following our image-fusion approach is capable to handle this problem quite well.

The training data-set was obtained by four circle-runs each sampled by 100 images. The

circle-diameters were 100mm, 120mm, 140mm and finally 160mm. Due to the fact that both arms revolved each other the distance between them is equivalent to the appropriate diameter of the circular-motion (s. Fig. 8(b) for the mean image corresponding to this motion). The test data-set consists of 30 random-

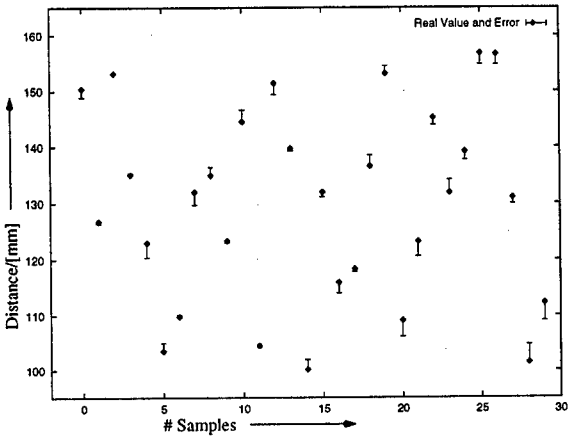


Figure 11: Error plot of exp. IV, $\bar{e} = 1.53$mm, $\sigma = 0.92$mm, $\hat{e} = 3.29$mm.

positions of both arms on a circle with a random-radius between 50mm and 80mm. Although the characteristic error values of this experiment are within the range of the (less complex) previous ones, the error plot (s. Fig. 11) shows larger deviations for a slightly larger number. Especially in cases where the arms overlap each other the error is larger (within the range of 1.00 and 3.29mm). Fig. 12 shows such an example. From the top camera-view both wrists are merged while from the side camera both are still separated yielding a still reasonable absolute estimation error of 1.35mm.

6 Conclusions

We have shown that high-dimensional problems such as estimating robot distance using uncalibrated cameras can be solved with a neuro-fuzzy model. The B-spline model serves as an efficient interpolator which can be interpreted as fuzzy control rules. The advantages

(a) Side view. (b) Top view.

Figure 12: Overlap during circular motion around each other. In this case the real distance is about 145.22mm and is estimated as being 143.87mm ($e = 1.35$mm).

of this approach are:

- By projecting the high-dimensional input space onto a reduced eigenspace, the most significant information for control is maintained. A limited number of transformed inputs can be partitioned with the B-spline model and a sufficient precision can be obtained for determining the distance.
- The statistical indices used in the approach provide a suitable solution to describe the information in images with a high degree of uncertainty.
- A vector in the eigenspace is directly mapped onto the controller output based on the B-spline model. This makes real-time computation possible.
- Training motion can be programmed so that representative images can be generated automatically.

If the observed scenarios are not "local" enough, i.e. the images possess less similarity it could happen that distance precision cannot be satisfied with too few (e.g. with four) eigenvectors. For these cases, we are investigating methods to classify the image sequence into more local scenarios by using some simple criteria. Additionally, the self-adaptation capability of the controller in the case of slightly changed camera positions will be quantitatively investigated. For testing the multisensor fusion, it is also an interesting work to incorporate more than two cameras to study the performance/efficiency ratio.

References

[1] M. Uchiyama, L. Cellier, P. Dauchez, and R. Zapata. Collision avoidance for a two-arm robot by reflex actions: Simulations and experimentations. *Journal of Intelligent Robotic Systems*, 14:219–238, 1995.

[2] R. G. Beaumont and R. M. Crowder. Real-time collision avoidance in two-armed robotic systems. *Computer-Aided Eng. J.*, pages 233–240, December 1991.

[3] Xiaoqing Cheng. On-line collision-free path planning for service and assembly tasks by a two-arm robot. In *Proc. Int. Conf. on Robotics and Automation*, volume 2, pages 1523–, 1995.

[4] M. Greenspan and N. Burtnyk. Obstacle count independent real-time collision avoidance. In *Proc. Int. Conf. on Robotics and Automation*, volume 2, pages 1073–1080, 1996.

[5] H. Seraji, R. Steele, and R. Ivlev. Sensor-based collision avoidance: Theory and experiments. *Journal of Robotic Systems*, 13(9):571–, 1996.

[6] J. D. Taylor and C. L. Boddy. Whole-arm reactive collision avoidance control of kinematically redundant manipulators. In *Proc. Int. Conf. on Robotics and Automation*, volume 3, pages 382–387, 1993.

[7] E. Cheung and V. Lumelsky. Real-time collision avoidance in teleoperated whole-sensitive robot arm manipulators. *IEEE Transactions on Systems, Man and Cybernetics*, 23(1):194–203, 1993.

[8] F. Chaumette, B. Espiau, and P. Rives. A new approach to visual servoing in robotics. *Lecture Notes in Computer Sciences*, pages 106–, 1993.

[9] K. Hosoda and M. Asada. Versatile visual servoing without knowledge of true jacobian. In *Proc. Int. Conf. on Intelligent Robots and Systems*, pages 186–193, 1994.

[10] M. Jägersand, R. Nelson, and O. Fuentes. Experimental evaluation of uncalibrated visual servoing for precision manipulation. In *Proc. Int. Conf. on Robotics and Automation*, 1997.

[11] C. Scheering and B. Kersting. Uncalibrated hand-eye coordination with a redundant camera system. In *Proc. Int. Conf. on Robotics and Automation*, 1998.

[12] S. K. Nayar, H. Murase, and S. A. Nene. Learning, positioning, and tracking visual appearance. In *Proc. Int. Conf. on Robotics and Automation*, pages 3237–3244, 1994.

[13] E. Oja. *Subspace methods of pattern recognition.* Research Studies Press, Hertfordshire, 1983.

[14] J. Zhang and A. Knoll. Constructing fuzzy controllers with B-spline models - principles and applications. *Int. Journal of Intelligent Systems*, 13(2/3):257–285, 1998.

[15] J. S. Albus. A new approach to manipulator control: The Cerebellar Model Articulation Controller (CMAC). *Transactions of ASME, Journal of Dynamic Systems Measurement and Control*, 97:220–227, 1975.

[16] T. Sanger. An optimality principle for unsupervised learning. In Touretzky, editor, *Advances in neural information processing systems 1.* Morgan Kaufmann, 1989.

[17] E. Oja. A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15:267–, 1982.

[18] J. Lloyd. *RCCL User's Guide.* Computer Vision and Robotics Laboratory, McGill University, 1988.

Session WA2
Fusion Architecture and Management I
Chair: Alan Steinberg
Environment Research Institute of Michigan, USA

Pitfalls in Data Fusion (and How to Avoid Them)

David L. Hall and Amulya K. Garga
Applied Research Laboratory
The Pennsylvania State University
P. O. Box 30
State College, PA 16804-0030
(814) 863-4155, (814) 863-5841
dlh28@psu.edu, garga@psu.edu

Abstract

Data fusion is a process that seeks to improve the ability to estimate the position, velocity, and identity/characteristics of entities by combining information from multiple sensors and sources. A rich legacy in data fusion technology exists, ranging from the Joint Directors of Laboratories (JDL) data fusion process model to taxonomies of algorithms and engineering guidelines for architecture selection and algorithm selection. To date, numerous data fusion systems have been developed, especially for department of defense (DoD) applications. Despite this history and legacy there remains a number of common misconceptions in data fusion, which can lead to pitfalls in system development. This paper provides a brief review of the state of practice in data fusion. Recommendations are provided on how to avoid these pitfalls and research needed to advance the state of data fusion system development.

1. Introduction

Data fusion is a process that seeks to improve the ability to estimate the position, velocity, and identity or characteristics of entities by combining information and data from multiple sensors and sources (Waltz and Llinas[1], Hall[2], and Hall and Llinas[3]). Applications of data fusion range from situation and threat assessment systems to *smart* weapons, automatic target recognition systems, identification-friend-foe-neutral (IFFN) systems, and intelligence applications. For these applications, multiple techniques are required for the fusion process. Techniques for fusion are drawn from disciplines such as signal and image processing (for characterization and processing of single sensor data), statistical estimation and pattern recognition, and decision-level processing methods from the domain of artificial intelligence. The selection of a processing architecture and specific algorithms is a systems engineering problem, dependent upon a number of factors such as the specific application, types of sensors, computing resources available, communication bandwidth available, and many other factors.

In recent years there has been a rapid evolution of data fusion technology including:

1) Development of a process model by the (JDL) data fusion working group[4];
2) Creation of a taxonomy and hierarchy of processing algorithms[5];
3) Survey and assessment of data fusion systems[6];
4) Establishment of engineering guidelines for algorithm selection[7,8];
5) Evaluation of data fusion technology[9]; and
6) Development of a data fusion lexicon[10].

Numerous data fusion systems have been implemented[6] and some software tool kits are becoming available to support rapid prototyping and technique evaluation (e g, Hall and Linn[11] and Hall and Kasmala[12]). Because of this legacy, it might appear that data fusion is a mature technology, and that implementation of data fusion systems involves a routine exercise in systems engineering and software development. However, design

and implementation of fusion systems remains very challenging.

It is beyond the scope of this paper to present a prescription for the successful implementation of these systems. However, we identify common problems (pitfalls) in data fusion and suggest how to avoid or mitigate these problems. The next section of this paper provides a framework by summarizing the JDL data fusion process model. Subsequently, we identify common problems or pitfalls and their effects on system implementation. In addition, advice is provided on how to mitigate or avoid these problems. Finally, recommendations are provided for research in data fusion.

2. JDL Data Fusion Process Model

The JDL Data Fusion Working Group was established in 1986 to assist in coordinating research in data fusion, and improving communications among different DoD research and development efforts. The JDL Data Fusion Working Group began an effort to codify the terminology related to data fusion. The result of that effort was the creation of a process model for data fusion[4], and a data fusion lexicon[10]. The top level of the JDL data fusion process model is shown in Figure 1.

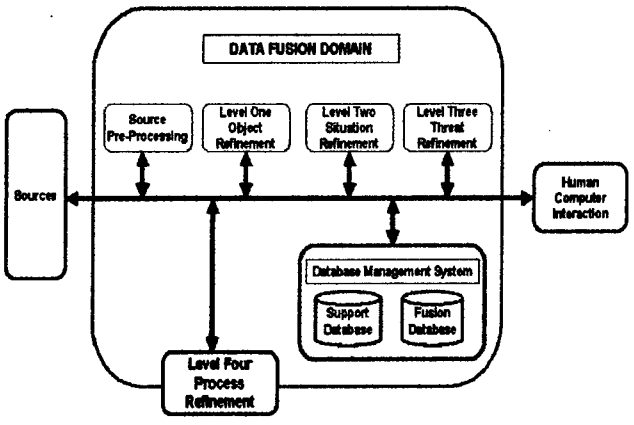


Figure 1: Top-Level JDL Data Fusion Process Model

The JDL process model [4,13] is a functionally oriented model and is intended to be very general and useful across multiple applications. The intent of the model is to assist researchers and developers in communicating about basic data fusion functions, algorithms, and techniques. The model is a *paper* model, and not intended to be used as a blueprint for system design or software development. A brief summary of each component of the process model is presented below.

Sources of information – The left side of Figure 1 indicates that a number of sources of information may be available as input to a data fusion system. These include: 1) local sensors physically associated with the data fusion system, 2) distributed sensors linked electronically to a fusion system, and 3) other data such as reference information, geographical information, etc. These input data may be in the form of scalar data (e.g., directional angles, range to target, range-rate, etc.), time series data (e.g., radar cross section versus aspect angle or acoustic spectra), images (e.g., an infrared image of a target) or textual data.

Human Computer Interaction (HCI) – The right side of Figure 1 shows the HCI function for fusion systems. HCI allows human input such as commands, information requests, analyses of inferences and reports from human operators. The HCI is the mechanism by which a fusion system communicates results via alerts, displays, and dynamic overlays of positional and identity information on geographical displays.

Source Preprocessing – Source preprocessing pre-screens data and reduces the data fusion system load by allocating data to appropriate processes (e.g., location and attribute data to Level 1, alerts to Level 3 processing, etc.). Source preprocessing may include advanced signal processing, image processing, and synthesis of complex array data to create synthetic information.

Level 1 Processing (Object Refinement) – This process combines location, parametric, and identity information to refine the representations of individual objects (e.g., emitters, platforms, weapons, or geographically constrained military units). Level 1 processing performs four key functions. First, it transforms sensor data into a consistent set of units and coordinates. Second, it refines and extends in time the estimates of an object's position, kinematics, or attributes. Third, it assigns data to objects to allow the application of statistical estimation techniques. Finally, it refines the estimation of an object's identity or classification.

Level 2 Processing (Situation Refinement) – Level 2 processing develops a description of the current relationships among objects and events in the context of their environment. Distributions of individual objects (defined by Level 1 processing) are examined to aggregate them into operationally meaningful combat units and weapon systems. In addition, situation refinement focuses on relational information (such as physical proximity, communications, causal, temporal, or other relations) to determine the meaning of a collection of entities. This analysis is performed in the context of environmental information about terrain, surrounding media, hydrology, weather, and other factors.

Level 3 Processing (Threat Refinement) - Level 3 processing projects the current situation into the future to draw inferences about enemy threats, friendly and enemy vulnerabilities, and opportunities for operations. Threat assessment is especially difficult because it deals not only with computing possible engagement outcomes, but also assessing an enemy's intent based on knowledge about enemy doctrine, level of training, political environment, and the current situation. The overall focus is on intent, lethality, and opportunity. Level 3 processing develops alternate hypotheses about an enemy's strategies and the effect on uncertain knowledge about enemy units, tactics, and the environment.

Level 4 Processing (Process Refinement) – Level 4 processing may be considered a *meta-process*, i.e., a process concerned about other processes. Level 4 processing performs four key functions. First, the data fusion process performance is monitored to provide information about real-time control and long-term performance. Second, it identifies what information is needed to improve the multi-level fusion product (inferences, positions, identities, etc.). Third, it determines the source-specific requirements to collect relevant information (i.e., which sensor type, which specific sensor, which database, etc.). Finally, it allocates and directs the sources to achieve mission goals. This latter function may be outside the domain of specific data fusion functions. Hence, Level 4 processing is shown as partially inside and partially outside the data fusion process.

Data Management – A major support function required for data fusion is data management[14]. This collection of functions provides access to, and management of, data fusion databases, including data retrieval, storage, archiving, compression, relational queries, and data protection. Database management for data fusion systems is particularly difficult because of the large and varied data managed (i.e., images, signal data, vectors, textual data, procedural information, rules, etc). In addition, data management is challenging because of the data rates for ingestion of incoming sensor data, as well as the need for rapid retrieval of data using general *Boolean* queries. Antony[14] provides an overview of database strategies for data fusion systems.

Hall and Llinas[3] provide a summary of the JDL data fusion process components and techniques. This paper also addresses the current state of the art for each processing function. Mathematical techniques for data fusion are provided by Hall[2] and by Waltz and Llinas[1]. The JDL model identifies the processes, functions, and techniques applicable to data fusion as information flows from the sources to the human operator.

3. Pitfalls in Data Fusion

The JDL data fusion process model provides a unified framework for the development of data fusion systems. However, in practice there are several pitfalls that can drastically affect the performance of the data fusion system. These pitfalls and their associated implications are summarized below.

There is no substitute for a good sensor. No amount of fusion of *bad* sensors will substitute for a single accurate sensor that measures the phenomena that you want to observe. A common misconception is that the fusion of multiple poor sensors will substitute for an accurate sensor. Under some conditions the use of marginal, non-commensurate sensors (i.e., sensors that measure fundamentally different physical phenomena such as infrared sensor, acoustic sensors, and radar) can improve the robustness of the assessment of a situation. An example is the use of multiple sensors to defeat an enemy's attempt to use camouflage or information warfare. One might use a combination of an infrared camera to characterize the heat of a tank's diesel engine, an acoustic sensor to observe the sound of a tank's engine, radar to observe a radar-cross section, and a visible image to identify a tank's size and shape. Because of the broad physical baseline, the combination of these sensors is difficult to defeat by camouflage or information warfare. However, in general, multiple, marginal-performance sensors do not combine to produce an improved result. Without special processing, it can be shown that the fusion of multiple sensors, each having a probability of correct detection or identification of less than 50 per cent, actually yields worse results that any individual sensor [15].

Downstream processing cannot make up for errors (or failures) in upstream processing. Data fusion processing cannot correct for errors in processing (or lack of pre-processing) of individual sensor data. Each sensor data stream (whether it be imagery, time series data, vector or scalar data) must be processed to perform signal characterization and representation. Each stream of sensor data must be analyzed to determine what types of manipulations and canonical transformations will best characterize and represent the data. An example is the case of feature-based pattern recognition for target identification. Failure to select good features from the sensor data cannot be overcome by the concatenation of multiple sensor feature vectors input to complex pattern classifiers. Instead, care must be taken to obtain as much information as possible from each sensor stream prior to the fusion process (whether that fusion occurs at the data, feature, or decision level).

Sensor fusion can result in poor performance if incorrect sensor information about sensor performance is used. A common failure in data fusion is to characterize the sensor performance in an *ad hoc* or convenient way (typically using static, Gaussian, zero-mean probability distributions to represent sensor performance). In the real world, sensor performance is very dynamic and non-Gaussian. Environmental conditions can cause wide variations in sensor performance (e.g., effects of terrain, atmospheric conditions, the local environment, etc). It is well known for example that acoustic sensor performance can vary by a factor of 100 depending upon the time of day and atmospheric conditions[16]. It is perhaps less well known that near-earth atmospheric conditions can seriously affect radar performance. Ironically, as sensors become more sophisticated, environmental conditions may exert a greater impact on their dynamic performance. It is easy to demonstrate that the use of incorrect error statistics for sensor performance can greatly corrupt data fusion processes. Hence, care must be taken to properly model or calibrate the dynamic performance of sensors.

There is no such thing as a magic or golden data fusion algorithm. Despite claims to the contrary, especially by their authors, there is no perfect algorithm that is optimal under all conditions. While this statement may seem obvious, there continue to be controversies in the

literature, and in the data fusion literature, about which algorithm is best, optimal, or robust. Consider, for example, the arguments between advocates of Bayesian methods versus Dempster-Shafer techniques, or advocates of multiple hypothesis tracking versus single hypothesis tracking. The reality is that for many practical applications, the mathematical assumptions upon which many of these methods are formulated are rarely satisfied. Sophisticated algorithms are easily corrupted and produce very poor results when the input data do not meet the requisite conditions[17, 18] (e.g., conditionally dependent observations, incorrect a priori probabilities, etc.). Care must be taken to ensure that the data fusion techniques selected for applications are appropriate to the available *a priori* knowledge. This issue is rarely considered in the implementation of data fusion systems.

There will never be enough training data. The development of algorithms for applications such as automatic target recognition (ATR) or identification-friend-foe-neutral (IFFN) processing often utilize *implicit* pattern recognition techniques such as neural networks or cluster algorithms. These algorithms are usually provided with examples of data (e.g., samples of the radar cross section or infrared signatures of known targets such as a tank). The algorithm is then *trained* to recognize the known targets based on the implicit patterns in the observed data (or features extracted from the data). For true generality these techniques require an enormous amount of training data.

Hush and Horne[19] provide a complete discussion and indicate that, in general, if there are n features and m classes to be identified or recognized, then the required number of independent training test cases should exceed n x m x k, where k is a number that ranges between 10 and 30. For realistic applications such as ATR the number of training cases needed would exceed several thousand. In practice of course, there is never enough data to satisfy this requirement for statistical significance. Despite a number of methods to provide a synthetic training population, other techniques must be used to obtain significance for the pattern recognizer. These include the use simulations and a special hybrid approach described by Hall and Garga[20] that involves the combination of implicit pattern recognition with explicit information such as that obtained from domain experts via fuzzy logic rules.

It is difficult to quantify the value of a data fusion system. One of the challenges of implementation of a fusion system is the ability to quantify the utility of the system. Waltz and Llinas[1] point out that there is no such thing as requirements for a data fusion system, *per se*. Instead, data fusion requirements are derived from system or operational requirements. Conceptually, a fusion system improves the ability to sense information, which in turn improves the ability to perform accurate inferences. The utility of a data fusion system must be determined by the extent to which a system supports overall mission requirements. Waltz and Llinas[1] suggest that the quantification of the value of a fusion system requires development of a hierarchy of measures of performance (MOP) and measures of effectiveness (MOE). These link quantities such as sensor performance (e.g., probability of correct identification and observation accuracy) to measures of overall mission success (e.g., quantities such as force effectiveness measures and probability of survival). The quantification of system effectiveness is a daunting task, but should be considered during system design and development to guide tradeoff analyses, and to establish realistic expectations concerning the utility of data fusion.

Fusion is not a static process. The data fusion process is an iterative dynamic process that seeks to continually refine the estimates about an observed situation or threat environment. This is explicit in the basic JDL definition of data fusion[4] and in the JDL process model. In the JDL process model described in the first part of this paper, a special level 4 process is aimed at recognizing these dynamics and adapting fusion algorithms and processing

to improve the evolving fusion products. In general, this adaptive control is an under-researched area in data fusion[21, 22]. Hall and Garga[22] suggest that the concept of Level 4 processing should be significantly extended to span the range from direct sensor management (e.g., computing of look angles for sensors and sensor controls) to adaptation to the information needs and styles of the human user of a data fusion system. This system level adaptation could provide significant improvements to the data fusion process and interpretation of results.

4. Avoiding the Pitfalls

In order to avoid the pitfalls described in the previous section, the following are some general recommendations for system implementers and designers. These are not meant to be an exhaustive set of prescriptions, nor a substitute for fundamental system engineering (involving formal analysis of the data fusion architecture, algorithm selection, requirement analysis, etc.). Nevertheless, these may provide insights for system designers and implementers.

Perform a thorough analysis of sensor technology and establish a map between observable phenomena and required inferences. In designing a data fusion system, a systematic analysis of underlying physical phenomena should be performed to determine what can be observed, and to link these observable quantities to required inferences (see for example the process suggested by Waltz and Llinas[1], p. 351). This analysis should consider both theoretical models as well as test range data in order to establish realistic expectations on sensor performance under anticipated operational conditions. The dynamic effects of the local environment and countermeasures should be considered to accurately characterize the sensor performance. This information should be considered as a vital input to the fusion algorithms along with the sensor data itself. Consideration should be given to implementation of use of smart meta-sensors to perform dynamic self-calibration[16]. Even if the sensor suite is pre-determined for a fusion application (e.g., the sensor suite on-board an aircraft), this sensor analysis should be performed. Steinberg[23] provides an example of this type of analysis.

Examine the information content of the sensor data and intelligently select algorithms for sensor pre-processing. Each sensor stream and type should be carefully analyzed to determine how to extract as much information as possible from the sensor data. Appropriate canonical transformations, filtering, feature extraction, and corrections should be made. Tradeoffs and comparisons should be made to systematically determine which algorithms are the most robust. Every step of the sensor processing flow should be analyzed and explicitly examined. Special care should be given to steps such as feature extraction and signal conditioning.

Perform a systematic algorithm selection and use rapid prototyping tools to perform tradeoffs against real data sets. Selection of algorithms for data fusion should be performed without preconceived notions about perfect algorithms. Candidate algorithms should be identified and considered based on the realistic availability of requisite *a priori* data (such as prior probabilities, etc.). The performance of the algorithms should be evaluated against real sensor data and the available computing suite for the intended operational fusion system. Designers should be wary of appeals for the use of mathematically sophisticated algorithms for which requisite information is not available. Designers may also consider the use of a hybrid approach in which multiple algorithms are used adaptively (viz., in which different algorithms are used in a dynamic sense based on the particular regime in which the fusion system is operating). Such an approach can also be used to adjudicate the use of different sensor data, based on the observational conditions[15].

Use hybrid pattern recognition methods to overcome the limitations posed by a lack of training data. For applications such as ATR or IFFN, in which insufficient

training data exists, designers should consider the use of hybrid pattern recognition techniques[20]. Rather than rely on *ad hoc* methods to artificially generate synthetic training data for a pattern classifier (e.g., a feed-forward neural network), designers should consider the approach developed by Hall and Garga[20] in which both explicit and implicit information are used for training. In that approach, explicit information from a domain expert is combined with sample test data and simulated data to establish a robust test set for the classifier. Designers should also consider an augmented approach in which a pattern classifier is combined with an automated reasoning interpreter that interprets the results of the pattern classifier in the context of a particular operational mission or observing environment. An example is the use of an automated reasoning system to interpret IFFN results based on whether information warfare is expected or not.

Measures of Performance (MOP) and Measures of Effectiveness (MOE) should be defined and computed to assess the utility of a data fusion system. In order to develop effective data fusion systems that are useful to support real applications, it is advisable to define and compute MOE and MOP. These provide a focus to ensure that the developed system will actually be of use to support a human user to make more effective decisions[24,25]. This focus on utility ensures that systems level design decisions are motivated by the fusion application and operational requirements, rather than by the *technology du jour.* In addition, MOE and MOP are of use in improving the Level 4 processing.

An intelligent Level 4 Process should be developed to monitor and improve the overall data fusion process. Significant thought and effort should be put into the Level 4 meta-process. Development of an intelligent monitor/controller for the fusion process can be of significant value in improving the data fusion system performance and in adapting the fusion performance to meet changes in the operational environment. This should be considered as an integral part of the fusion system.

5. Conclusions

Data fusion is a rapidly maturing technology with an extensive legacy. The research community is beginning to adopt common models and terminology, while system developers are beginning to reach consensus on engineering guidelines. In addition, commercial tools are beginning to appear. Despite this maturity, the implementation of effective data fusion remains a challenge. This paper describes some common misconceptions and pitfalls related to data fusion. Recommendations are provided to avoid these pitfalls.

References

[1] E. Waltz and J. Llinas, *Multisensor Data Fusion*, Artech House, Inc., Norwood, MA, 1990.

[2] D. L. Hall, *Mathematical Techniques in Multisensor Data Fusion*, Artech House, Inc., Norwood, MA, 1992.

[3] D. L. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proceedings of the IEEE*, vol. 85, No. 1, January 1997, pp. 6-23.

[4] Kessler, et al, *Functional Description of the Data Fusion Process*, Technical Report, Office of Naval Technology, Naval Air Development Center, Warminster, PA, January, 1992.

[5] D. L. Hall and R. J. Linn, "A taxonomy of algorithms for multisensor data fusion," in *Proceedings of the 1990 Tri-Service Data Fusion Symposium*, April, 1991, pp. 13-29.

[6] D. L. Hall, R. J. Linn, and J. Llinas, "A survey of data fusion systems," in *Proceedings of the SPIE Conference on Data Structure and Target Classification*, vol. 1470, Orlando, FL, April, 1991, pp. 13-36.

[7] D. L. Hall and R. J. Linn, "Algorithm selection for data fusion systems," *Proceedings of the 1987 Tri-*

Service Data Fusion Symposium, APL Johns Hopkins University, Laurel, MD, Vol. 1, pp. 100-110, June 1987.

[8] A. N. Steinberg and C. L. Bowman, "Development and application of data fusion system engineering guidelines," *Proceedings of the National Symposium on Sensor and Data Fusion (NSSDF)*, Lexington, MA, May 1998.

[9] D. L. Hall and J. Llinas, "A challenge for the data fusion community I: research imperatives for improved processing," in *Proceedings of the 7th National Symposium on Sensor Fusion*, Albuquerque, NM, March 1994.

[10] F. E. White, *Data Fusion Lexicon*, Data Fusion Sub-panel of the Joint Directors of Laboratories Technical Panel for C^3 NOSC, San Diego, CA, 1991.

[11] D. L. Hall and R. J. Linn, "Survey of commercial software for multisensor data fusion," in the *Proceedings of the SPIE Conference on Sensor Fusion and Aerospace Applications*, Orlando, FL, April 1991.

[12] D. L. Hall and G. Kasmala, "A visual programming tool kit for multisensor data fusion," *Proceedings of the SPIE AeroSense 1996 Symposium*, Orlando, FL, Vol. 2764,181-187, April 1996.

[13] A. N. Steinberg and C. Bowman, "Revisions to the JDL Data Fusion Process Model," *Proceedings of the National Symposium on Sensor and Data Fusion (NSSDF)*, Lexington, MA, May 1998.

[14] R. Antony, *Principles of Data Fusion Automation*, Artech House, Inc., Norwood, MA, 1995.

[15] D. Miller and D. L. Hall, "The use of automated critics to improve the fusion of marginal sensors for ATR and IFFN applications", *Proceedings of the 1999 National Symposium on Sensor Data Fusion* (NSSDF), Johns Hopkins University, APL, Baltimore, MD, May, 1999.

[16] D. L. Hall and D. Swanson, "Real-time data fusion processing of internetted acoustic sensors for tactical application", *Proc. 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Las Vegas, NV, pp. 443-446, 2-5 October.

[17] W.H. Press, B.P. Flannery, S.A. Teukoolsky, and W.T. Vetterling, *Numerical Recipes: The Art of Scientific Computing*, New York, Cambridge, 1986.

[18] R. Deutsch, *Orbital Dynamics of Space Vehicles*, Prentice-Hall, Englewood Cliffs, New Jersey, 1963

[19] D. Hush and B. Horne, "Progress in supervised neural networks: What's new since Lippman?", *IEEE Signal Processing Magazine*, pp 8-39, Jan. 1993.

[20] D. L. Hall and A. K. Garga, "Hybrid Reasoning Techniques for Automated Fault Classification", *Proc. Society for Machinery Failure Prevention Technology (MFPT) Society 51st Meeting*, April 1997.

[21] P. L. Rothman and S. G. Bier, "Evaluation of sensor management systems," *Proceedings of the 1991 Fusion Systems Conference*, Oct 1991, John Hopkins University APL, Baltimore, MD.

[22] D. L. Hall and A. K. Garga, "New perspectives on Level 4 processing in data fusion", *Proc. of the SPIE Aero Sense '99 Conference: Digitization of the Battlefield IV*, Orlando, FL, April 1999.

[23] A. Steinberg, "Threat management system for tactical aircraft," *Proceedings of the 1987 Tri-Service Data Fusion Symposium*, Vol. I, Johns Hopkins Physics Laboratory, Laurel, MD, pp. 32 – 542, June 1987.

[24] J. Llinas, C. Drury, W. Bialas, and A. Chen, *Studies and Analyses of Vulnerabilities in Aided Adversarial Decision Making*, Technical Report, State University of New York at Buffalo, Department of Industrial Engineering, February 1997.

[25] D. L. Hall and J. Llinas, "From GI Joe to Starship Trooper: the evolution of information support for individual soldiers," *Proceedings of the International Symposium on Circuits and Systems*, June 1998, Monterey, CA.

The Omnibus Model: A New Model of Data Fusion?

Mark Bedworth and Jane O'Brien
Jemity, PO Box 113, Malvern,
Worcestershire, WR14 3YJ, UK
Email: {mark.bedworth, jane.obrien}@datafusion.clara.net

Abstract - *Over the last two decades there have been several process models proposed (and used) for data and information fusion. A common theme of these models is the existence of multiple levels of processing within the data fusion process. In the 1980's three models were adopted: the intelligence cycle, the JDL model and the Boyd control. The 1990's saw the introduction of the Dasarathy model and the Waterfall model. However, each of these models has particular advantages and disadvantages.*

A new model for data and information fusion is proposed. This is the Omnibus model, which draws together each of the previous models and their associated advantages whilst managing to overcome some of the disadvantages. Where possible the terminology used within the Omnibus model is aimed at a general user of data fusion technology to allow use by a distributed audience.

Keywords: Data fusion architectures and models.

1. Introduction

Data fusion is a diverse field encompassing many approaches, algorithms and applications. As with any such complex endeavour the communal knowledge needs to be structured and organised in order to be effective. Over the last two decades several process models have been proposed for data and information fusion [1]. In each case the motivation seems to have been a partitioning of knowledge into sub-tasks since a common theme of these models is the prescription of multiple levels of processing within the data fusion process itself. A multi-disciplinary, international data fusion community is now developing and the issue of standardisation is becoming important. The purpose of this paper is threefold. First we establish the nomenclature for data fusion models, architectures and frameworks. Secondly, we review the main existing data fusion models identifying the advantages and limitations of each. Armed with this information we describe a new model which may be regarded as subsuming the current models. We propose that this model form the basis of a new international standard.

2. Models, Architectures and Frameworks

The conceptual organisation of our collected knowledge regarding data fusion has taken many forms. As a result a potential confusion of terminology may arise. We shall therefore define a few terms to describe the way in which data fusion algorithms may be embedded in the context of a larger system.

Three main organisational paradigms are currently in use for describing data fusion systems. These are:

- Models
- Architectures
- Frameworks

We shall describe each of these in turn, highlighting the main differences between them:

Model - we define a model, or more specifically a process model, to be a description of a set of processes. This set of processes should be undertaken before the system may be regarded as fully operational. As such it highlights the component functions which the system has but makes no statement regarding their software implementation or physical instantiation. The study of process models forms the main thrust of the remainder of this paper.

Architecture - we define an architecture to be the physical structure of the system. We make particular reference to the way in which data or information is communicated. An architecture includes the arrangement of the component parts, their connectivity and the data flows between the them. The architectural description may be high level - data fusion systems which are described as centralised, hierarchical or distributed are classified by their architecture. It may also be specific - blackboard systems [2] and common object request brokering (CORBA) [3] are specific examples of distributed architectures.

Framework – we define a framework to be a set of axioms and a reasoning system for manipulating entities based on those axioms. As such a framework provides us with a method of inferencing from a data-rich / information-poor source to produce abstract concepts which are information-rich Examples of frameworks currently used in data fusion are probabilistic reasoning, possibilistic reasoning and evidential reasoning.

The remainder of the paper will concentrate on data fusion models. Architectures and frameworks are equally important but are left for future discussions.

3. Review of Data Fusion Process Models

Data fusion has its roots in the defence research community of the early 1980's. As a result the first data fusion models were either adapted from existing military oriented process models or were designed with a distinctly military flavour [4]. More recently the use of data fusion has broadened to include industrial, medical and commercial applications. More recent models have acknowledged this migration by reducing the military terminology. However, this still exists to some extent (and needs to be changed).

3.1 The Intelligence Cycle

Intelligence processing involves both information processing and information fusion. Although the information is often at a high level, the processes for handling intelligence products are broadly applicable to data fusion in general. There are a number of well-established principles of intelligence:

- central control (this avoids the possibility of duplication). The issue of central control is really an architectural issue. The avoidance of duplication may be achieved in several ways;
- timeliness (this ensures that the intelligence is available fast enough to be useful);
- systematic exploitation (makes sure that the outputs of the system are used appropriately);
- objectivity (of the sources and the manner in which the information is processed). This is perhaps the factor most relevant to data fusion;
- accessibility (of the information);
- responsiveness (to changing intelligence requirements);
- source protection (to guarantee a source of information with increased longevity);
- continuous review (of the process and the collection strategy).

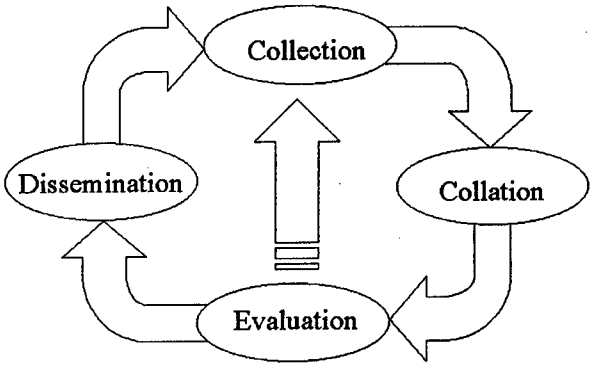


Figure 1: The UK intelligence cycle.

The UK intelligence community describes the intelligence process as a cycle, which lends itself to modeling the data fusion process [5]. The cycle itself is depicted in Figure 1. Related models exist outside the UK. The British model, unlike the American intelligence model, does not include a specific planning and direction phase that is subsumed in the dissemination process. The UK intelligence cycle comprises four phases:

Collection – collection assets such as electronic sensors or human derived sources are deployed to obtain raw intelligence data. In the world of intelligence the data is often presented in the form of an intelligence report which is already at a high level of abstraction – either free form text or in a predefined report format.

Collation – associated intelligence reports are correlated and brought together. Some combination or compression may occur at this stage. Collated reports, however, may simply be packaged together ready for fusion at the next stage.

Evaluation – the collated intelligence reports are fused and analysed. Historically, highly skilled human intelligence analysts have undertaken this process. The analysis may identify significant gaps in the intelligence collection. In this case, the analyst may be able to task a collection asset directly. More usually, however, this requirement is included in the disseminated information.

Dissemination – the fused intelligence is distributed to the users (usually military commanders) who use the information to make decisions regarding their own actions and the required deployment of further collection assets.

3.2 The JDL Model

In the JDL model, proposed by the US Joint Directors of Laboratories Data Fusion Sub-Group in 1985 [6] and recently updated [7], the processing is divided into five levels as depicted in Figure 2.

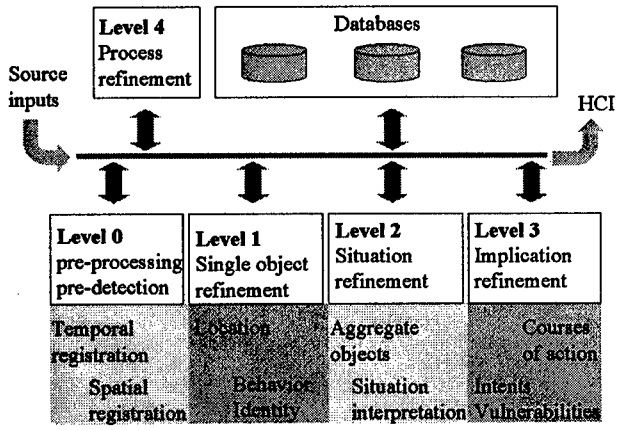


Figure 2: The JDL data fusion process model as recently updated.

Level 0 – sub-object data assessment, is associated with pre-detection activities such as pixel or signal processing, spatial or temporal registration.

Level 1 – object refinement is concerned with the estimation and prediction of continuous (*e.g.* location or kinematic) or discrete (*e.g.* behaviour or identity) states of objects.

Level 2 – situation refinement introduces context by examining the relations among entities such as force structure and communication rôles. By aggregating objects into meta-objects an interpretation may be placed on the situation.

Level 3 – implication refinement delineates sets of possible courses of action and the effect they would have in the current situation. This level also introduces the concept that the data fusion system may be operating in an adversarial domain.

Level 4 – process refinement is an element of resource management and used to close the loop by re-tasking resources (*e.g.* sensors, communications and processing) in order to support the objectives of the mission.

This model has been widely used by the US data fusion community and can now be regarded as the *de facto* standard for defence data fusion systems, at least in the US. Partly because of its popularity it is applied in a variety of ways [8] and is not always used appropriately. The JDL model was never intended to prescribe a strict ordering on the data fusion levels. This was indicated diagrammatically by the use of an information bus rather than a flow structure. Nevertheless, data fusion system designers have consistently assumed this ordering. Clearly there is a need for users to have an ordering whilst the authors of the JDL model rightly defend the need for a model which admits systems of systems with different hierarchies at different levels.

3.3 The Boyd Control Loop

The Boyd control loop [9] was first used for modeling the military command process but has since been widely used for data fusion. The Boyd (or OODA) loop possesses four phases as shown in Figure 3.

The Boyd and JDL models show distinct similarities, although the Boyd model makes the iterative nature of the problem more explicit.

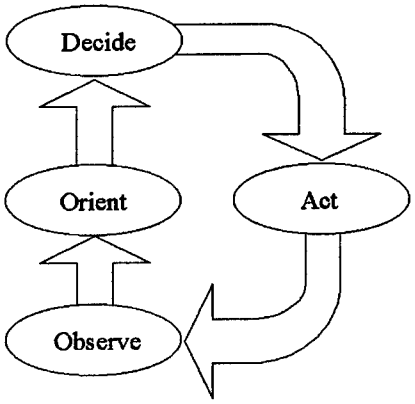


Figure 3: The Boyd (or OODA) loop, which has been used as a data fusion process model.

Observe – broadly comparable to the JDL level 0 and part of the collection phase of the intelligence cycle.

Orient – encompasses the functions of JDL levels 1, 2 and 3. It also includes the structured elements of collection and the collation phases of the intelligence cycle.

Decide – includes JDL level 4 (process refinement and resource management) and the dissemination activities of the intelligence community. It also includes much more (such as logistics and planning).

Act – has no direct analogue in the JDL model and is the only model that explicitly closes the loop by taking account of the effect of decisions in the real world.

The PEMS loop (for Predict, Extract, Match and Search) can be regarded as a perceptual specialisation of the Boyd control loop. The PEMS model has recently attracted attention for automatic target recognition and data fusion [10].

3.4 The Waterfall Model

The Waterfall model was proposed in [11] and endorsed by the UK Government Technology Foresight Data Fusion Working Party [12]. It places its main emphasis on the processing functions at the lower levels (see Figure 4). Again, similarities exist with the other models. Sensing and signal processing correspond to JDL level 0, feature extraction and pattern processing to JDL level 1, situation assessment to JDL level 2 and decision making to JDL level 3. In the Waterfall model the feedback is not explicitly depicted. This appears the major limitation of the Waterfall model which otherwise divides the data fusion process more finely than other models. The Waterfall model has been widely used in the UK defence data fusion community but has not been significantly adopted elsewhere.

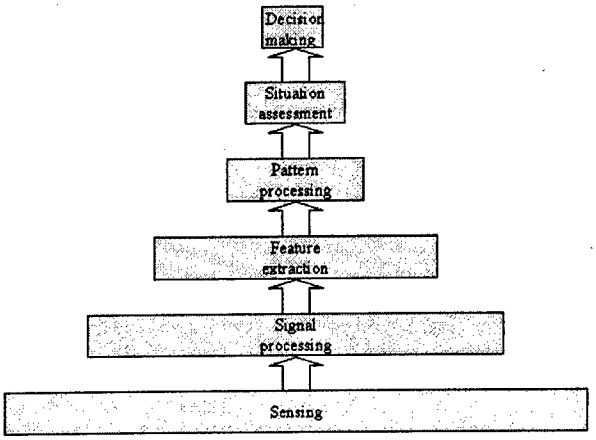


Figure 4: The Waterfall data fusion process model.

3.5 The Dasarathy Model

Many researchers have identified the three main levels of abstraction during the data fusion process as being:

- **Decisions** – symbols or belief values
- **Features** – or intermediate-level information
- **Data** – or more specifically sensor data

As was pointed out by Dasarathy in [13], however, fusion may occur both within these levels and as a means of transforming between them. In the model proposed there are five possible categories of fusion as shown in Table 1 at the end of this paper.

3.6 Model Comparison

The five models described in the preceding sections can be compared and equivalencies identified where appropriate. Table 2 shows a comparison between the process models described thus far. In some cases the equivalence is approximate. Greyed out boxes are not addressed by the specific model. It can be seen that there is some overlap in the way that the different

models sub-divide the information flow from sensors to actions. The main differences correspond to the amount of detail with which particular processes are represented. This stems from the different uses of the various models and the emphasis they place on certain aspects of the information processing and fusion chain. As can be seen from Table 2, the Waterfall model contains the finest distinction between the lower levels of abstraction, the JDL model at the medium level and the Boyd loop at the higher levels. The intelligence cycle covers all levels but in somewhat compressed detail. The Dasarathy model is based on fusion functions rather than tasks and may therefore usefully be incorporated in each of the fusion activities.

4. Requirements of a Process Model

The design and implementation of a data fusion system may usefully be regarded as a problem-solving task. As such a standard approach to problem-solving [14] may be addressed. Of particular relevance are:

- Has the problem been solved before?
- Has the same problem appeared in a different form and is there an existing solution?
- Is there a related problem with similar constraints?
- Is there a related problem with the same unknowns?
- Can the problem be sub-divided into parts that are easier to solve?
- Can the constraints be relaxed to transform the problem into a familiar one?

A process model should facilitate answers to these questions by providing a sub-division of the problem which is rich enough (and detailed enough) to allow re-use of specific knowledge whilst being general enough to allow existing solutions to be deformed into new domains. In this way we may break the goal into sub-goals and those into smaller sub-goals until we reach a set of objectives which is attainable [15,16]

The model should therefore not only represent the system under consideration but should also simplify it conceptually. The model will only be of practical use to system developers if this simplification facilitates calculations and predictions.

The foregoing review indicates shortcomings of each of the existing models. Specifically we require a model which:

- defines the ordering of processing;
- makes the cyclic nature of the system explicit;
- admits representation from multiple viewpoints;
- identifies the advantages and limitations of various fusion approaches;
- facilitates the clarification of task-level measures of performance and system-level measures of effectiveness;
- uses a general terminology which is widely accessible;
- does not assume that applications are defence oriented.

With these desires in mind we define a new process model.

5. The Omnibus Model

A unified model, to be known as the Omnibus model, is proposed. It comprises a flow chart, a dual-perspective definition and a structured repository of accumulated expertise.

The Omnibus flow chart shown in Figure 5 is based around the cyclic nature of the intelligence cycle and the Boyd control loop but uses the finer definitions of the Waterfall model, each of which can be associated with one of the levels in the JDL and Dasarathy models. In the Omnibus model feedback is explicit and the previously neglected concept of loops within loops is acknowledged. The cyclic nature of the data fusion process is made explicit by retaining the general structure of the Boyd loop. The fidelity of representation expressed by the Waterfall model is then easily incorporated into each of the four main process tasks. The points in the process where fusion may take place are explicitly located.

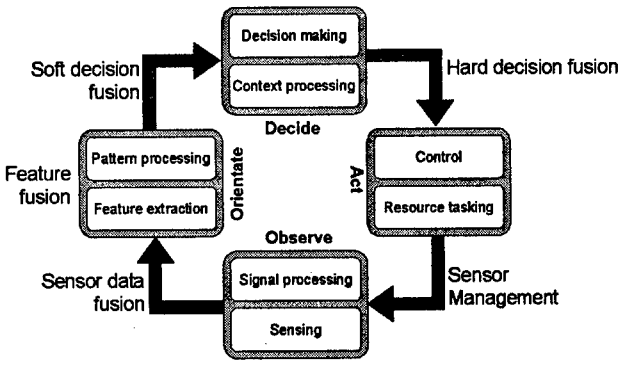


Figure 5: The Omnibus model – a unified data fusion process model.

The model is used in two ways. Firstly, it characterises and sub-divides the overall system aims to provide an ordered list of tasks. Secondly, the same structure may be used to organise the functional objectives of each such task. Using this approach a data fusion solution is categorised using a dual perspective – both by its system aim and its task objective.

The repository of communal knowledge should be structured in a top-down fashion so that analogies can be drawn at either abstract or specific levels. A part of this repository should be a list of advantages and limitations of fusion approaches and techniques. Table 3 shows an example list.

5.1 Omnibus Case Study

A technique for fusing multiple uncertain detection reports of intruder aircraft is required for embedding within an air defence data fusion system. Using the Omnibus, dual-perspective, approach this problem is categorised as:

System aim – orientation (information has been provided by the sources and the aim is to detect aircraft regardless of context)

Task objective – soft decision fusion (the reports contain symbolic data from multi-sensor system).

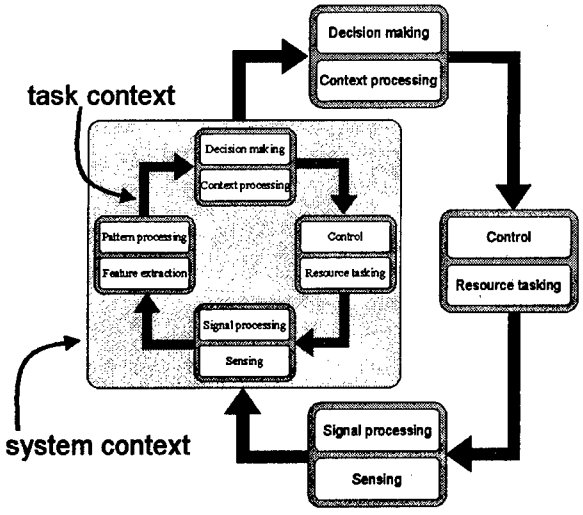


Figure 6: A system within a system representation of the air defence case study example.

This indicates that the system level communication and processing bandwidth will be moderate whilst the task-level bandwidth will be low. We may also predict the likely performance at the task level (false alarm rate for example) and begin to analyse its impact on the system-level effectiveness (cost effective reduction in own-force casualties for instance).

6. Conclusions

A common disadvantage of existing data fusion models is that they are each specifically oriented towards a military domain (to some degree). This is understandable given the origins of data fusion. However, with the increasing use of fusion techniques for industrial and commercial problems it is necessary to define a model with which the extending fusion community is able to identify.

The requirements of a process model have been re-examined and a new model, the Omnibus model, has been proposed. This comprises a process flow chart, a dual-perspective prescription for using it and a structured repository of fusion knowledge. The Omnibus model overcomes some of the main limitations of previous model whilst capitalising on their advantages. The nomenclature used is loosely based on existing notation to maximise familiarity but moves away from a defence-based scheme.

7. Recommendations

The fledgling data fusion community has now reached the maturity that warrants a re-examination of the organisation and structuring of the communal knowledge. The three main categorisation methods – models, architectures and frameworks – would lead to a less (unwarranted) duplication and substantial savings from reduced nugatory effort. Some degree of standardisation has to be a good thing in this respect.

It is suggested that over the next 12 months an internationally agreed terminology for (and descriptions of) models, architectures and frameworks be put in place. We observed that several models are currently in use by different parts of the fusion community (largely characterised by their geographic location or their application domain). Some entrenchment of ideas is also evident and the inertia of certain models will make it difficult to inject change.

It was not originally our intention to invent yet another model since we feel there are already too many. However, the Omnibus model emerged as a montage of the best aspects of existing models and therefore represents a unification rather than something new.

References

[1] M. Kokar and K. Kim, "Review of Multisensor Data Fusion Architectures", Proceedings of IEEE (1993).

[2] E. Shahbazian, E. Bosse and P. Valin, "Multi-Agent Data Fusion Workstation (MADFW) Architecture", Proceedings of AeroSense conference, SPIE Vol. 3376, pp. 60-68 (1998).

[3] M. Balci and S. Kuru, "A CORBA Based Infrastructure (CORBIS) for Sensor Data Fusion Systems", Proceedings of AeroSense conference, SPIE Vol. 3179, pp. 220-229 (1999).

[4] D. Hall and J. Llinas, "Data Fusion and Multisensor Correlation", Technology Training Corporation course (1985).

[5] A. Shulsky, "Silent Warfare: Understanding the World of Intelliegnce", Brassey's (1993).

[6] F. White, " A Model for Data Fusion", Proceedings 1st National Symposium on Sensor Fusion (1988).

[7] A. Steinberg, C. Bowman and F. White, "Revisions to the JDL Data Fusion Model", Proceedings of AeroSense conference, SPIE Vol. 3719, pp. 430-441 (1999).

[8] L. Klein, "Sensor and Data Fusion Concepts and Applications", SPIE Volume TT14 (1993).

[9] J. Boyd, "A Discourse on Winning and Losing", Maxwell AFB lecture, (1987).

[10] J. Gainey and E. Blasch, "Development of Emergent Processing Loops as a System of Systems Concept", Proceedings of AeroSense conference, SPIE Vol. 3179, pp. 186-195 (1999).

[11] M. Bedworth, "Probability Moderation for Multilevel Information Processing", DRA Technical Report DRA/CIS(SE1)/651/8/M94.AS03BP032/1 (1994).

[12] M. Markin, C. Harris, M. Bernhardt, J. Austin, M. Bedworth, P. Greenway, R. Johnston, A. Little and D. Lowe, "Technology Foresight on Data Fusion and Data Processing", Publication of The Royal Aeronautical Society, (1997).

[13] B. Dasarathy, "Sensor Fusion Potential Exploitation – Innovative Architectures and Illustrative Applications", Proceedings of IEEE, Volume 85, Number 1, pp 24-38 (1997).

[14] G. Polya, "How to Solve It", Princeton University Press (1945).

[15] R. Atkinson, R. Atkinson, E. Smith, D. Bem and E. Hilgard, "Introduction to Psychology", Harcourt Brace Jovanovich (1990).

[16] J. Anderson, "Cognitive Psychology and its Implications", New York Freeman (1985).

Input	Output	Notation	Analogues
Data	Data	DAI-DAO	*Data-level fusion*
Data	Features	DAI-FEO	Feature selection and feature extraction
Features	Features	FEI-FEO	*Feature-level fusion*
Features	Decisions	FEI-DEO	Pattern recognition and pattern processing
Decisions	Decisions	DEI-DEO	*Decision-level fusion*

Table 1: The five levels of fusion in the Dasarathy model.

Activity being undertaken	Waterfall model	JDL model	Boyd Loop	Intelligence Cycle
Command execution			Act	Disseminate
Decision making process	Decision making	Level 4	Decide	
Threat assessment		Level 3	Orient	Evaluate
Situation assessment	Situation assessment	Level 2		
Information processing	Pattern processing	Level 1		Collate
	Feature extraction			
Signal processing	Signal processing	Level 0	Observe	Collect
Source / sensor acquisition	Sensing			

Table 2: A comparison between the data fusion process models described in this paper.

Fusion level	Bandwidth	Performance	Advantages	Limitations
Hard decisions	Very low	Depends on system	Simplicity for large systems	Poor performance for small systems
Soft decisions	Low	Often good	Bandwidth / performance trade-off	Sophisticated algorithms needed for correlated sources
Features	Moderate	Good→high	High performance	Difficult to select correct features
Data	High→very high	Potentially optimal	Possibility of using physical models	High bandwidth restricts use to single platform systems

Table 3: The advantages and disadvantages of the four levels of data fusion.

DATA FUSION IN SUPPORT OF DYNAMIC HUMAN DECISION MAKING

Stéphane Paradis, Richard Breton and Jean Roy
Decision Support Technology Section
Defence Research Establishment Valcartier
2459 Pie XI Blvd. North
P.O.8800, Val-Bélair, G3J 1X5 CANADA

Abstract – *Command and control can be characterised as a dynamic human decision making process. A technological perspective of command and control has led system designers to propose solutions such as data fusion to overcome many of the domain problems. This and the lack of knowledge in cognitive engineering have in the past jeopardised the design of helpful computerised aids aimed at complementing and supporting human cognitive tasks. Moreover, this lack of knowledge has most of the time created new trust problems in designed tools, and human in the loop concerns. Solving the command and control problem requires balancing the human factor perspective with the one of the system designer and coordinating the efforts in designing a cognitively fitted system to support the decision-makers. This paper presents a triad model establishing the relationship between the three elements required for the design of a system that support dynamic human decision making: the task, the human and the technology.*

Keywords: *Command and Control, Data Fusion, Situation Awareness, Decision Making, Cognitive Engineering.*

1.0 Introduction

Command and control (C2) is defined, by the military community, as the process by which a commanding officer can plan, direct, control and monitor any operation for which he is responsible in order to fulfil his mission [Ref. 1]. Recently, a new definition has been proposed [Ref. 2] describing C2 as a dynamic human decision making process that establish the common intent and transform that common intent into a co-ordinated action.

From a human factor perspective, the complexity of military operations highlight the critical role of human leadership in C2. To resolve adversity, C2 systems (CCSs) require qualities inherent to humans such as decision-making abilities, initiative, creativity and the notion of responsibility and accountability. Although these qualities are essential, characteristics inherent to the environment in which C2 occurs, combined with the advancement in threat technology, significantly challenge the accomplishment of this process and therefore require the support of technology to complement human capabilities and limitations.

A technological perspective of C2 has led system designers to propose solutions such as data fusion to overcome many of the domain problems by fitting warships with an efficient combat system featuring a real-time decision support system (DSS). The main role of such a DSS is to aid the operators to achieve the appropriate situation awareness (SA) state for their tactical decision-making activities, and to support the execution of the resulting actions. The lack of knowledge in cognitive engineering has in the past jeopardised the design of helpful computerised aids aimed at complementing and supporting human cognitive tasks. Moreover, this lack of knowledge has most of the time created new trust problems in designed tools, and human in the loop concerns.

Solving the C2 problem thus requires balancing the human factor perspective with the one of the system designer and coordinating the efforts in designing a cognitively fitted system to support the decision-makers. This paper presents a triad model establishing the relationship between the three elements required for the design of a system that support humans: the task, the human and the technology. The concepts lying behind this model are currently being used for the design of a DSS that complement and support the human during his cognitive activities. The model allows the design of systems taking into account the human role in a dynamic decision making process such as C2.

2.0 The command and control task

C2 is the process by which commanders can plan, direct, control and monitor any operation for which they are responsible [Ref. 1]. C2 requires that the commander is aware of the tactical situation in order to make a timely decision about the best course of action to be implemented. In a naval context afloat, most tactical decisions taken within the ship's operations room are made after completing a number of perceptual, procedural and cognitive activities linked to the C2 process. The C2 process is indeed a suite of cyclic activities which mainly involves the perception of the environment and an assessment of the tactical situation, from which the decision making about a course of action and the implementation of the chosen plan will be based.

The C2 activities are performed by either humans, machines (i.e., hardware and software computer systems), or a combination of both. Characteristics of this suite of activities are described in Ref. 3 and were captured through the Boyd's Observe-Orient-Decide-Act (OODA) loop illustrated in Figure 1.

Although the OODA loop might give the impression that C2 activities are executed in a sequential way, in reality, the activities are concurrent

and hierarchically structured. The military community typically states that the dominant requirement to counter the threat and ensure the survivability of the ship is the ability to perform the C2 activities (i.e., the OODA loop) quicker and better than the adversary. Therefore, the speed of execution of the OODA loop and the degree of efficiency of its execution are the keys to success for shipboard tactical operations.

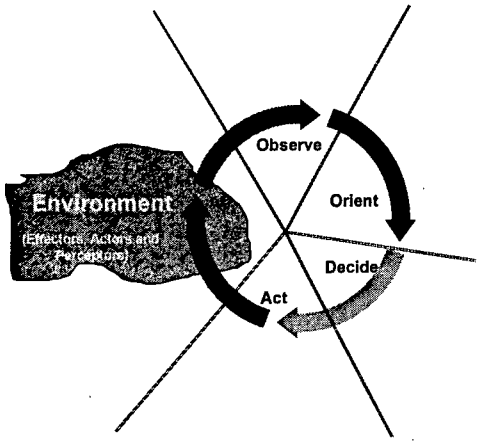


Figure 1 - Boyd's OODA Loop

C2 is characterised by ill-structured problems, changing and stressful conditions, technological advances in threat technology, the increasing tempo and diversity of open-ocean and littoral (i.e., near land) scenarios, the volume, rate, imperfect nature and complexity of the information. Most likely, the latter will be processed under time-critical conditions and, as a consequence, the risk of saturation in building the tactical picture and of making the wrong decision increases.

These characteristics inherent of the C2 domain pose significant challenges to the C2 process, to the design of future shipboard CCSs and to the combat system operators responsible to conduct this process using these systems to defend their ship and fulfil their mission.

3.0 Human factor perspective of C2

The C2 process is seen as an instantiation or an example of a dynamic human decision making process that establishes the common intent and transforms that common intent into a co-ordinated action [Ref. 2]. The first half of Boyd's loop (Observe-Orient) gathers a number of processes that mainly perceives, interprets and projects the status of the entities included in the C2 environment. Yielding from these processes is the situation awareness required to complete the decision-making process. The latter process corresponds to the second half (Decide-Act) of the OODA loop. Given the tactical situation and the available onboard resources, it decides on the best course of action with respect to own ship mission and support its implementation.

Figure 2 illustrates a theoretical model derived by Endsley of situation awareness (SA) based on its role in dynamic human decision making. SA is defined [Ref. 4] as the perception of the elements in the environment, within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future.

The first level of SA yields in the *perception* of the status, attributes and dynamics of relevant elements in the environment. If we look at our problem domain, maritime C2, the basic element relevant for the command team is any object in the environment (e.g., air, surface or subsurface targets).

Endsley describes the comprehension process as follows: "Comprehension of the situation is based on a synthesis of disjoint level 1 elements". Level 2 of SA goes beyond simply being aware of the elements that are present, to include an understanding of the significance of those elements in light of pertinent operator goals. Based on knowledge of Level 1 elements, particularly when some elements are put together to form patterns with other elements, the decision-maker forms a holistic picture of the environment, comprehending the significance of objects and events.

The third and last step in achieving situation awareness is the *projection* of the future actions of the elements in the environment. This is achieved through knowledge of the status and dynamics of the perceived and comprehended situation elements.

The situation awareness processes described by Endsley are initiated by the presence of an object in the perceiver's environment. However, processes related to situation awareness can also be triggered by a priori knowledge, feelings or intuitions. In these situations, the picture is understandable, and projections in the future are possible, if any event, which have not been perceived at this time, can be found in the environment. Hence, hypotheses related to the possible presence of an object are formulated. The perceiver then initiates search processes in the environment that confirm or invalidate these hypotheses. Note that this type of SA is possible only if mental models related to the possible objects are available.

If one compares the OODA loop with the SA model of Endsley, one sees a close resemblance. In both models one finds a *decision-making* part and an *action* part. In Endsley's model, SA is one of the main inputs for decision-making. In the OODA loop, the processes Observe and Orient provide inputs for the decision making process. One should recall, however, that situation awareness in Endsley's model is a *state of knowledge* and not a process.

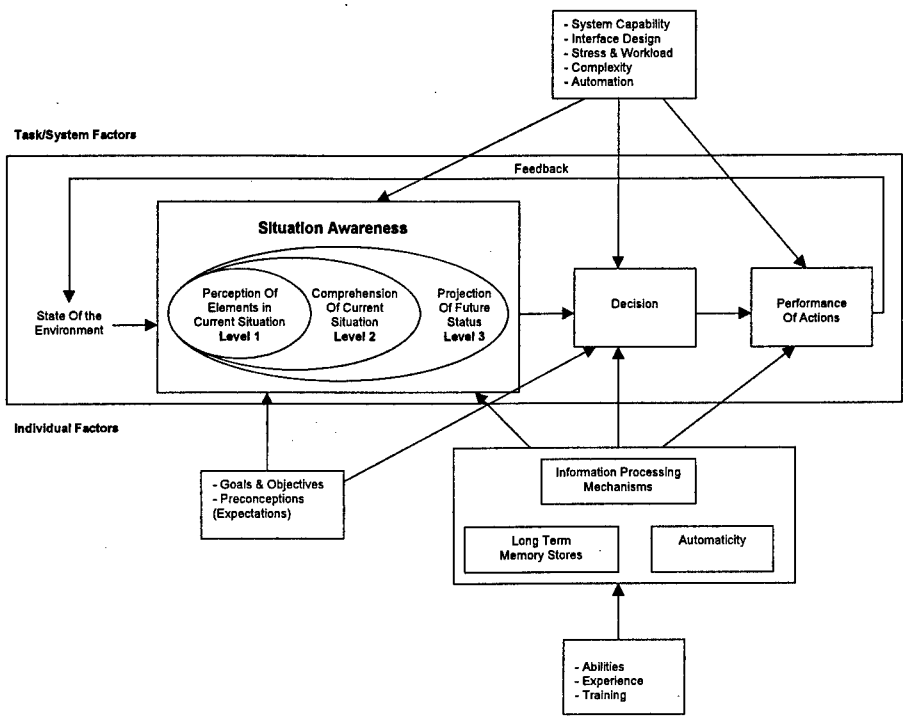


Figure 2 – SA Model in Dynamic Human Decision Making

In her theory of SA, Endsley clearly presumes *patterns* and *higher level elements* to be present according to which the situation can be structured and expressed. SA can be interpreted as the operator's mental model of all pertinent aspects of the environment (processes, states, and relationships).

There is a tight link between this mental model used to structure and express situation elements and the cognitive processes involved in achieving the levels of awareness. This link is known as the cognitive fit and requires an understanding of how the human perceives a task, what processes are involved, what are the human needs and what part of the task can be automated or supported. This understanding is crucial and only achieved via a number of specialised human factor investigations known as cognitive engineering analyses.

Cognitive engineering analyses are generally conducted by the human factor engineering community. Human factor engineering can be seen as the US counterpart of the ergonomics. According to Preece [Ref. 5], the cognitive ergonomics is a discipline that focuses particularly on human information processing and computer systems. By definition, it aims to develop knowledge about the interaction between human information processing capacities and limitations, and technological information processing systems.

The usefulness of a system is closely related to its compatibility with the human information processing. Thus, such a system must be developed according to the human information processing and human needs. A first step is the identification of the cognitive processes involved in the execution of the task. Many procedures have been developed to identify those processes. Jonassen, Hannum and Tessmer [Ref. 6] describe the task analysis as a process that is performed in many ways, in a variety of situations, and for multiple purposes. This analysis determines what the performers do, how they perform the task, how they think or how they apply a skill.

Among the procedures developed to identify cognitive processes, there are the Cognitive Task Analysis (CTA) and the Cognitive Work Analysis (CWA). There are only subtle and ambiguous differences between these two procedures. Moreover, their labels are frequently used in an interchangeable manner in the literature. However, the CWA can be seen as a broader analysis than the CTA. According to Vicente [Ref. 7], traditional task analysis methods typically result in a single temporal sequence of overt behaviour. This description represents the normative way to perform the task. However, traditional methods cannot account for factors like changes in initial conditions, unpredictable disturbances and the use of multiple strategies. The use of the traditional task analysis brings an artefact that will only support one way to perform the task.

Vicente proposes an ecological approach in which the three factors above are considered. The ecological approach, which can be seen as a CWA, takes its roots in psychological theories that were first advanced by Brunswick [Ref. 8] and Gibson [Refs.9-10]. These researchers raised the importance to study the

interaction between the human organism and its environment. The perception of an object in the environment is a direct process, in which information is simply detected rather than being constructed [Ref. 10]. The human and the environment are coupled and cannot be studied in isolation. A central concept of this approach is the notion of affordance. The affordance is an aspect of an object that makes it obvious how the object is to be used. Examples are a panel on a door to indicate "push", and a vertical handle to indicate "pull" [Ref. 5]. When the affordance of an object is obvious, it is easy to know how to interact with it. The environment in which a task is performed has a direct influence in the overt behaviour. Hence, the ecological approach begins by studying the constraints in the environment that are relevant to the operator. These constraints influence the observed behaviour. Ref. 11].

The ecological approach is comparable to and compatible with Rasmussen's abstraction hierarchy framework. Rasmussen's framework is used for describing the functional landscape in which behaviour takes place in a goal-relevant manner. This abstraction hierarchy is represented by means-ends relations and is structured in several levels of abstraction that represent functional relationships between the work domain elements and their purposes. With the ecological approach, Rasmussen has developed a comprehensive methodology for CWA that overcomes the limitations of traditional CTA by taking into account the variability of performance in real-life, complex work domains. For these reasons, the CWA seems to be the best choice to answer questions related to understanding the C2 task.

Within the design process of a system, the technological development has raised new issues and challenges. The type of issues and challenges has shifted from identifying the technological possibilities and limitations through determining how these systems must be designed to fit with the human information processing. This situation has brought the emergence of the human factor community and the development of CTA and CWA methods.

4.0 Technology perspective of C2

Command and control has been, and still is, a challenging problem to address from a technological perspective. The complexity of the C2 task opens the door to a broad range of technological solutions. Bearing in mind the scope of this paper, only the technological aspects of C2 related to the transformation and the fusion of data are addressed.

According to the Joint Directors of Laboratories (JDL) [Refs. 12-13], a complete DF system can typically be decomposed into five levels:

- Level 0 - Signal Data Refinement (source pre-processing);
- Level 1 - Object Refinement (Multi-Source Data Fusion (MSDF));
- Level 2 - Situation Assessment (SA);
- Level 3 - Threat Assessment (TA); and,
- Level 4 - Process Refinement through Resource Management (RM).

Each subsequent level of DF processing deals with a higher level of abstraction. Level 1 DF uses mostly numerical, statistical analysis methods, while levels 2, 3, and 4 of DF use mostly symbolic or Artificial Intelligence methods. Note that RM in the context of level 4 fusion is mainly concerned with the refinement of the information gathering process (e.g., sensor management).

For several years, research and development activities in DF concepts and algorithms have been conducted at the Defence Research Establishment Valcartier, leveraging from the JDL model. Lately, a number of R&D activities have been undertaken focusing on the application of DF to the design of a DSS for maritime C2 [Ref. 14]. All these efforts yielded in the derivation of two generic systems.

First, a generic MSDF system (level 1 fusion) has been derived presenting the functionalities required for the fusion of data from dissimilar sources to accomplish the tracking and the identification of the objects sensed in the environment [Ref. 15]. Figure 3 depicts the generic MSDF system.

A first cut at another generic system, illustrated in Figure 4, has also been produced. It provides the high-level functional decomposition of a multilevel Situation and Threat Assessment process (levels 2-3 fusion). This latter generic system has been derived taking into consideration some cognitive engineering and SA concepts. A detailed description of this generic system is given in [Ref. 16].

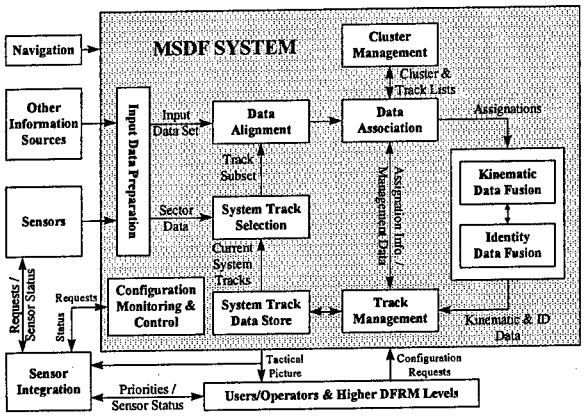


Figure 3 – Generic MSDF System

Ongoing efforts are now aiming at deriving an integrated version of these two generic systems while applying some cognitive engineering principles [Refs 17-18]. Figure 5 illustrates a framework used to

investigate issues related to the integration of all DF levels [Refs. 19-20].

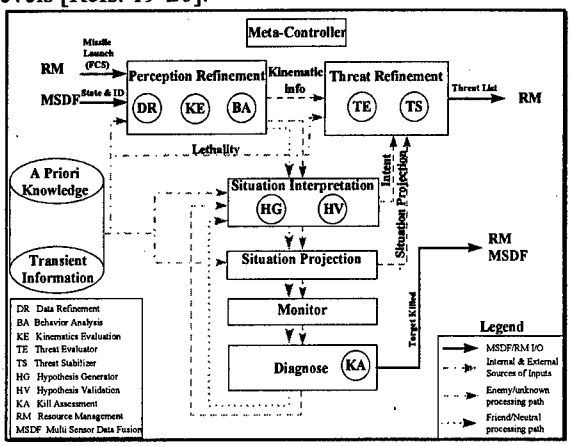


Figure 4 – Generic STA System

The JDL model, with its process refinement capability, implicitly supposes that all levels of fusion are integrated within the same framework. The levels of fusion are linked together and cannot be considered independently, in an opened loop fashion, without missing functionalities and/or reducing the quality of their results. The functional decomposition of the DF process and the quality of its results are also different whether the process is implemented as an opened or a closed loop.

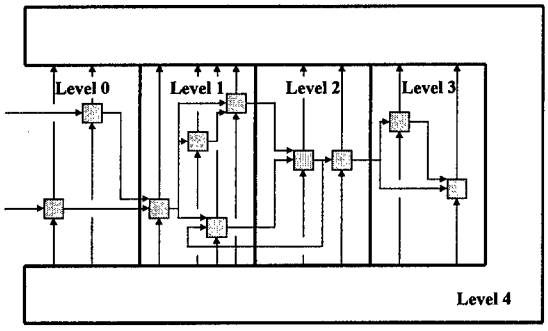


Figure 5 – Integrated Data Fusion Framework

The closed loop implementation inherently uses the notion of process refinement. This means, for instance, that the level 1 fusion process will be refined and enhanced leveraging from the results of higher levels of DF. As a result, the level 1 fusion process then benefits indirectly of contextual information. This is a major difference from the opened loop implementation where the results of the level 1 fusion process are context-free.

Clearly, the tight integration of all DF levels is essential to gain the maximum benefits from this process. Unfortunately, the R&D effort in the data fusion domain has generally been done in a fragmented way. Most of the time, the functionalities of one level of fusion have been studied independently from the other levels and, consequently, they have also been implemented on independent and opened loop test beds.

The resulting framework, illustrated in Figure 5, provides the appropriate environment to any DF sub-process and therefore allows the integration of all DF levels, and the achievement of all states of SA. The framework, from now on referred to as the integrated DF framework, is composed of a number of interconnected DF sub-processes or agents within a closed loop environment. All DF agents within the framework are required to comply with the notions of process refinement.

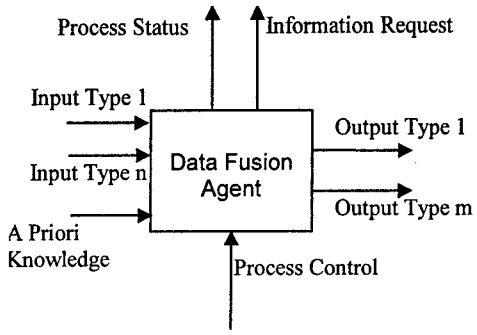


Figure 6 - Data Fusion Agent

Any DF sub-process included in the framework can be defined as an agent according to Figure 6. A set of dynamic inputs is presented to the agent along with a priori knowledge. These inputs can originate from the results of prior stovepipe processes, the results of higher level processes, the results of additional or complementary processing or, from the environment sensing process. The agent can be managed, via a process control flow, for the tuning of the parameters of its current algorithms or for the selection of alternate algorithms.

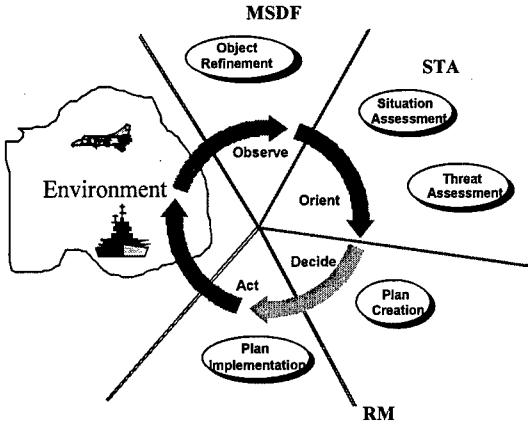


Figure 7 – Data Fusion & OODA Loop Mapping

In addition to the actual results of the process, the DF agent can output a request for additional information that would eventually be used to refine itself. Finally, a Process Status flow indicates the current status of the Data Fusion agent. This status

may indicate, for example, how much time remains before a result is expected to be available at the output.

Given the description of data fusion presented above, one can now appreciate its usefulness in support of the C2 process. Figure 7 presents the mapping of the DF process onto the OODA loop. This mapping can be seen as a system designers perspective of the C2 task.

5.0 Meeting the C2 requirements

5.1 Task-human interaction

As mentioned earlier, it is crucial, for anyone who takes part of the design process of a system, to understand what the performers do, how they perform the task, how they think or how they apply a skill. Hence, a good understanding of human resources, skills and limitations is required within the context of the task. It is necessary to understand the interaction between the human and the task. A CWA can provides this understanding. The analysis is made in isolation of any system available and considers the constraints related to the task, such as the human resources, skills and limitations. From this analysis, shortfalls and needs are identified. Obviously, these needs are closely related to human limitations.

Physical factors like stress and fatigue must also be considered when assessing human skills and limitations to perform a task. According to Proctor and Van Zandt [Ref. 21], stress refers to a physiological response to unpleasant or unusual conditions. These conditions may be imposed by the physical environment, the task performed, one's personality and social interactions. Stress situations are defined by a substantial imbalance between the demands imposed by the environment and the human's capability to successfully handle those demands. Stressful situations are created by overload and also by underload [Ref. 22]. The influence of physical factors on decision-making abilities have been investigated in the Tactical Decision Making Under Stress project (TADMUS) following the Vincennes incident [Ref. 23].

The human has limited resources and these resources are generally related to the capacity of attention. It seems that the attention is divided in limited pools of resources. There is some multiplicity of non-overlapping reservoirs (see Wickens [Ref. 24]). The pools would be related to each specific sensory modality (for a review, see Pashler, [Ref. 25]). Hence, two different tasks can be performed simultaneously if they are referring to different pools. For instance, it is possible to drive a car and talk with someone at the same moment. However, it is impossible to sing and talk simultaneously. This affirmation brings the concept of serial and parallel processing. Two different tasks that refer to different pools can be processed in parallel. However, they must be processed serially if they refer to the same pool. In the latter situation, the workload related to the two different tasks determines the complexity of the situation. The workload can be defined by the demand required by the execution of a task in function of the resources available in the pools. The workload cannot be solely defined in terms of attentional resources.

The working memory is also involved in any attentive activity. The working memory is the cognitive center responsible for problem solving, retrieval of information, language comprehension, and many other cognitive operations [Ref. 26]. To encode words in the long-term memory, the human must be attentive to these words, and the flow of presentation of the words cannot exceed the capacity of the working memory. Unfortunately, the storage and processing capacity of the working memory is limited. However, these limited resources can be expanded through practice.

The workload related to a task is thus defined by the demands imposed by the task in terms of attentional and working memory resources needed. Moreover, the human performance is closely related the workload of the task. Tasks with high workload can be seen as more complexes than task with low workload. However, strategies, practice and training can reduce the workload to a level at which enough resources are available. The idea that mental events operate automatically after a certain amount of practice is a well-entrenched doctrine of folk psychology, and it has a long history in academic psychology [Ref. 25]. According to Schneider and Shiffrin [Ref. 27], mental operations that are trained sufficiently are performed more quickly and accurately. They also undergo qualitative changes. Trained operations impose less capacity demands, providing more resources for concurrent mental activities. Trained operations also are not subject to voluntary control or conscious awareness and require little or no mental effort.

Rasmussen [Ref. 28] proposes a skill-rule-knowledge (SRK) framework including three different levels of performance in which the automation is different (see Fig. 8). At the skill-based level, human performance is governed by stored patterns of knowledge. This knowledge is acquired with practice. With a specific stimulation from the environment, a specific response is given. The link between the stimulation and the response can be seen as a reflex that requires no effort or conscious awareness. The second level is the rule-based level that is applicable to tackling familiar problems in which solutions are governed by rules (if-then-else). Processes related to this level are mainly automatics. With new situations,

the third level described by Rasmussen is involved. The knowledge-based level deals with unfamiliar situations for which actions must be planned on-line, using conscious analytical processes and stored knowledge. These processes are controlled and impose high mental workload. However, with practice and training, unfamiliar situations become familiar and can thus be solved at the rule-based level. Moreover, with extended practice, these knowledge can even become a reflex to the specific situations (skill-based level).

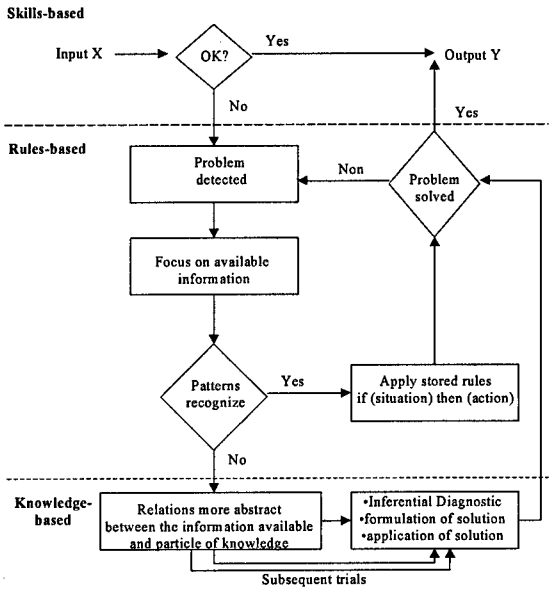


Figure 8 – Rasmussen SRK Framework

The SRK framework is compatible with the notions of bottom-up and top-down processing. The bottom-up stresses the importance of the stimulus in the environment. Data arrive from the sensory receptors and influence directly the perception of the information. The top-down processing stresses the importance of a person's knowledge and concepts in the perception process. The human knowledge about how the world is organised helps the human to perceive and understand the environment. Even if these two approaches of processing are opposite, they are not incompatible. In fact, it is probable that in any perceptual process of the environment, both are implicated. Since the top-down processing lays on the person's concepts and knowledge, this approach is compatible with the Rasmussen's theory of human performance. This processing approach is also related to the training and practice. The top-down processing happens if concepts and knowledge have been previously stored in the long-term memory.

Dreyfus [Ref. 29] proposes 5 different stages to become and expert (novice, advanced beginner, competence, proficient and expertise). However with extended practice and the use of strategies, the human may require the support of systems.

It is crucial that these systems be designed according to the human information processing. The CWA provides an understanding of how the human perceives the task and defines constraints of environment. From this analysis, it is important to identify which part of the task can be, and must be, automated, and which part of the task can and must be supported. Human shortfalls are translated as requirements for the technology community.

5.2 Task-technology interaction

As mentioned previously, C2 is a very complex and ill-defined problem within an uncooperative environment. With technological developments, it is appealing to tackle the C2 problem by providing humans with computer-based systems.

Evidently, human and machines have different capabilities for performing various tasks [Ref. 30]. On one hand, in addition to number crunching capabilities, computer-based systems have great deductive capacities. However, they can hardly make inductive reasoning. On the other hand, the human can hardly deal with several hypotheses at the same time, but has the capacity to make inductive reasoning. According to Ballas [Ref. 31], inducing hypotheses is better accomplished by humans and the validation of these hypotheses is efficiently done by computer-based aids.

According to Bainbridge [Ref. 32], the automation of processes may expand rather than eliminate problems with the human operator. Such developments may increase the complexity of the environment thereby imposing higher processing demands to the human. In such circumstances, the role of the human would shift from a controlling role toward a monitoring one. Hence, it seems that the technological development redefines the human contribution. In fact, Bainbridge suggests that the more advanced a system is, the more crucial may be the contribution of the human.

Bainbridge also raises an important point with automated systems. One can only expect the operator to monitor the computer's decisions at some meta-level, to decide whether the computer's decisions are acceptable. If the computer is being used to make decisions because the human judgement and intuitive reasoning are not adequate in the context, then, which of the decisions are to be accepted? The human in a monitoring role cannot handle the information processing and decision loop anymore. Much likely the human will not cope with the system and, consequently, won't use it due to a lack of proper understanding and/or trust.

Therefore, system designers are confronted to new challenges. The nature of the limitations to be

considered within the design process of a system is different. Limitations are more and more related to the human information processing. It is thus crucial to understand how the human perceive a task, which processes are involved, what are the human needs and which part of the task can be automated or supported.

These issues do not mean that decision support systems or automated systems are not useful. However, the way they are designed, their purposes and their interaction with the human are critical. Moreover, given the nature of the unpredictable events, it is crucial that the design process starts with a complete understanding of the environmental constraints and the human information processing. The technological perspective must be seen as the solution to human shortfalls. Hence, the design process must involve systems designers and human factors specialists.

6.0 Task/human/technology triad model

A triad approach has been proposed by Breton, Rousseau and Price [Ref. 33] to represent the collaboration between the systems designers and the human factors specialists. As illustrated in Figure 9, three elements compose the triad: the task, the technology and the human. In the C2 context, the OODA loop represents the task to be accomplished. The design process must start with the identification of the environmental constraints and possibilities by subject-matter experts within the context of a CWA.

Systems designers are introduced via the technology element. Their main axis of interest is the link between the technology and the task. The general question related to this link is: "What systems must be designed to accomplish the task?" Systems designers are also considering the human. Their secondary axis of interest is thus the link between the technology and the human. The main question of this link is: " How must the system be designed to fit with the human"? However, systems designers have a hidden axis. The axis between the human and the task is usually not covered by their expertise. From their analyses, technological possibilities and limitations are identified. However, all environmental constraints may not be covered by the technological possibilities. These uncovered constraints, named thereafter deficiencies, are then addressed as statements of requirements to the human factor community (see Fig. 10). These requirements lead to better training program, the reorganisation of work and the need for leadership, team communication, etc.

Human Factor specialists are introduced via the human element of the triad. Their main axis is the link between the human and the task, which is the hidden axis of systems designers. With a CWA, they identify how the humans perceive the task, what they have to do to accomplish the task, what strategies and resources are involved and what are the shortfalls and human limitations. Their secondary axis of interest is the same as the one for the system designers (i.e., human-technology), and their hidden axis is the link between the technology and the task, which is the main axis of the system designers. From their analyses, human possibilities and limitations are identified. However, all environmental constraints may not be covered by the human possibilities and resources. The uncovered deficiencies are then addressed as statements of requirements to the technological community (see Fig. 11). These statements become the specification of which part of the task needs support or must be automated, what the system must do, in which conditions, and how the system must interact with the operator.

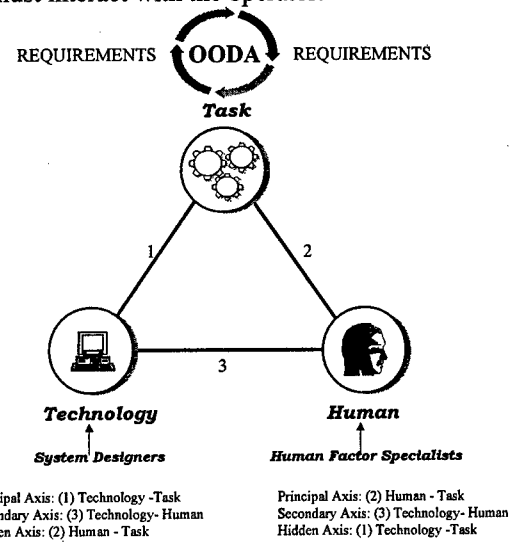


Figure 9- Task/Human/Technology Triad Model

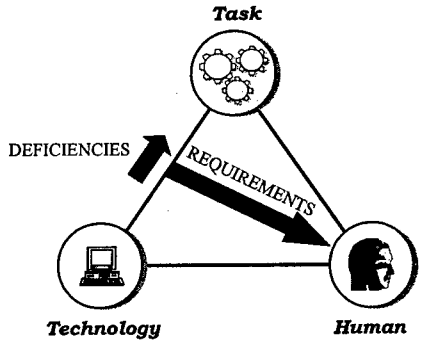


Figure 10 - Human Factor Requirements

In this context, everyone involved in the design process has its own field of intervention. The weakness of one is the strength of the other. The sets of statements of requirements produced by the systems designers and the human factor specialists are analysed within a multi-disciplinary team involving both communities. This analysis leads to one set of consolidated requirements that determines the nature

of the solution (see Fig. 12). It is very important that both types of specialists work in a close collaboration. Working in isolation would bring unrealisable requirements formulated by one part to the other.

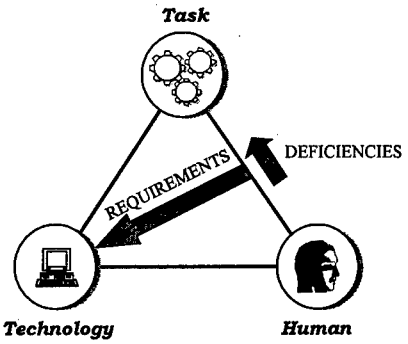


Figure 11 - Technological Requirements

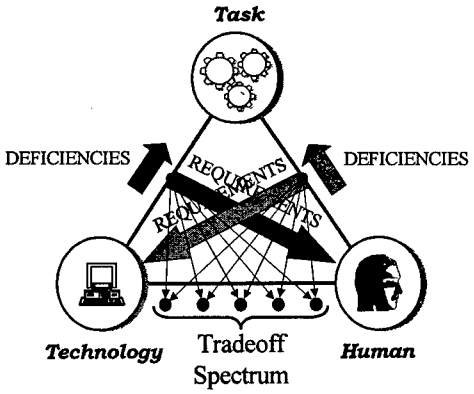


Figure 12 - Requirements Tradeoff Spectrum

Within the context of a war or tactical operations, unpredictable events are expected more frequently and are caused mainly by intelligent sources. The inductive capacity of human is then required to deal with these events. Some part of the system can be automated, but the system must be mostly design to support the human in its activities. Hence, the solution cannot be found from a complete technological perspective or a complete human perspective. It must rather be amixture of both.

Automation has changed the nature of the implication of the human. With automated systems, the human role is mainly related to the supervision of the situation. As mentioned earlier, this new role brings new problems and issues to be considerated. In particular, this situation raises the question about which part has the authority. There is no general answer to this question. A proposed approach is to delegate authority according to the situation. Chalmers [Ref. 34] proposes five modes of operator-system delegation. The human selects the mode, which applies until mode transition is triggered by a new selection. It is obvious that a good understanding of the situation is crucial to select the required mode. Each mode implies a fixed delegation of authority for all the various sub-processes for which automated support is available. Figure 13 presents these modes along with the variations in the level of work distribution and the synergy between the automation and the operator in these various modes.

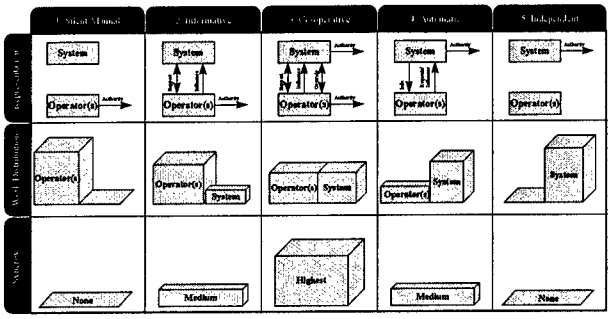


Figure 13 - Operator-System Modes of Operation

7.0 CONCLUSION

This paper described command and control as a very complex and ill-defined dynamic human decision making process within a non-cooperative environment. Data fusion is seen as a promising technological solution to tackle the C2 problem, but it can't assure that it will support adequately the human cognitive requirements usually obtained through cognitive engineering analyses. The lack of knowledge in cognitive engineering can jeopardised the design of computerised aids and, most of the time, introduce new problems such as human in the loop concerns and trust.

The paper presented a triad model establishing the relationship between the three elements required for the design of a system that supports dynamic human decision making: the task, the human and the technology. Solving the command and control problem requires balancing the human factor perspective with the one of the system designer and coordinating the efforts in designing a cognitively fitted system to support the decision-makers.

8.0 References

1. Parks, D., Johnson, B., Picard, G., Dubois, S., MacLean, J., Ouellet, J. and Craig, M., "Maritime C3I Way Ahead Working Group (MC3IWAWG) Final Report", DREV - TM - 9715, October 1997.
2. Pigeau, R., "The human in command", Defence Science & Technology,Issues #2, January 1998.
3. Hart, G. and Lind, W.S., America Can Win: the Case for Military Reform, Adler and Adler, Bethesda, Maryland, 1986.
4. Endsley, M. R., "Toward a Theory of Situation Awareness in Dynamic Systems", Human Factors Journal, 37(1), 32-64, March 1995.
5. Preece, J., Human-Computer Interaction. In Y. Rogers, H. Sharp, D. Benyon, S. Holland and T. Carey (Eds.) Addison-Wesley, England, 1994.
6. Jonassen D. H., Hannum, W. H., & Tessmer, M.

(). Handbook of task analysis procedures. Praeger Publishers : New York, 1989.
7. K.J. Vicente, A Few Implications of an Ecological Approach to Human Factors, Lawrence Erlbaum Associates, NJ, 1995.
8. Brunswick, E., Perception and the representative design of experiments (2nd ed.) Berkeley: university of California Press, 1956.
9. Gibson, J. J., The problem of temporal order in stimulation and perception. Journal of Psychology, 62, 141-149, 1966.
10. Gibson, J.J., The Ecological Approach to Visual Perception, Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ, 1979.
11. Rasmussen, J., Information Processing and Human-Machine Interaction: An Approach to Cognitive Engineering, North-Holland, New York, 1986.
12. Llinas, J. and Hall, D. L., "An Introduction to Multi-Sensor Data Fusion", Proceedings of the 1998 IEEE International Symposium on Circuits and Systems, Monterey, California, June 1998.
13. Antony, R.T., Principles of Data Fusion Automation, Artech House, Norwood, MA, 1995.
14. Chalmers, B., Roy, J., Carling, R., Paradis, S. and Bossé, É., "Study of Real-Time Issues for an Integrated MSDF/STA/RM System for the CPF: A Detailed Description of the SRTE Project", DREV-R-9629, March 1998.
15. Roy, J., Bossé, É., "A Generic Multi-Source Data Fusion System", DREV-R-9719, June 1998.
16. Paradis, S., Chalmers, B.A., Carling, R., Roy, J., Bossé, É., "Towards A Generic Model For Situation And Threat Assessment", DREV-R-9622, December 1996.
17. Treurniet, W., van Delft, J., Paradis, S., "Tactical Information Abstraction Framework in Maritime Command and Control", NATO Symposium on Modeling and Analysis of Command and Control, Paris, France, January 1999.
18. Chalmers, B. A., Burns C. M., "A Model-Based Approach to Decision Support Design for a Modern Frigate", TTCP Symposium on Coordinated Maritime Battlespace Management, Space and Naval Warfare Systems Center, San Diego, CA, USA, May, 1999.
19. Paradis, S., Treurniet, W., Roy, J., "Environment Perception Process in Maritime Command and Control", NATO Symposium on Sensor Data Fusion and Integration of the Human Element, Ottawa, Canada, September 1998.
20. Paradis, S., Roy, J., Treurniet, W., "Integration of All Data Fusion Levels Using a Blackboard Architecture", Eurofusion98: International conference on Data Fusion, Great Malvern, UK, October 1998.
21. Proctor, R. W., Van Zandt, T., Human factors in simple and complex systems, Allyn and Bacon, Needham Heights, MA, 1994.
22. Proctor, R. W., Van Zandt, T., Human factors in simple and complex systems, John Wiley & Sons, Toronto, Canada, 1983.
23. Riffenburgh, R. H., "Tactical Decision Making Under Stress Program Study: Initial Design", Naval Ocean Systems Center, San Diego, CA, 1991.
24. Wickens, C. D., Processing resources in attention. In R. Parasuraman & R. Davies (Eds.), Varieties of attention (pp. 63-102). New York: Academic Press, 1984.
25. Pashler, H. E., The Psychology of Attention. A Bradford Book. The MIT Press, Cambridge, Massachusetts, 1998.
26. Haberlandt, K., Cognitive Psychology (2nd ed.), Allyn and Bacon Publishers: Boston, 1997.
27. Schneider, W., & Shiffrin, R. M., Controlled and automatic human information processing: I. Detection, search, and attention. Psychological Review, 84, 1-66, 1977.
28. Rasmussen, J., Skills, rules, knowledge; signals, signs, and symbols; and other distinctions in human performance models. IEEE Transactions on Systems, Man, and cybernetics, Vol. SMC-13, No. 3, 1983.
29. Dreyfus, H. L., Intuitive, deliberative, and calculative models of expert performance. In C. E. Zsambok and G. Klein (Eds). Naturalistic Decision Making (pp. 17-28). Lawrence Erlbaum Associates : New Jersey, 1997.
30. Schneiderman, B., *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Second Edition, Addison Wesley Publishing Company, 1992.
31. Ballas, J. A., McFarlane, D.C., Achille, L.B., Stroup, J. L. and Kushnier, S.D., "Interfaces for Intelligent Control of Data Fusion Processing", Naval Research Laboratory, NRL/FR/5513-96-9806, April 1996.
32. Bainbridge, L., The Ironies of Automations, Automatica, vol. 19, No. 6, pp. 775-779, 1983.
33. Breton, R., Rousseau, R. and Price, W., "The integration of human factor in the design process: a triad approach", DREV-R-99XX, March 1999. (in preparation)
34. Chalmers, B., "On the Design of a Decision Support System for a Data Fusion and Resource Management in a Modern Frigate", NATO Symposium on Sensor Data Fusion and Integration of the Human Element, Ottawa, Canada, September 1998.

Implementing Knowledge and Data Fusion in a Versatile Software Environment for Adaptive Learning and Decision-Making

Nik Kasabov[1], David Tuck[2], Michael Watts[1]

[1] Department of Information Science, University of Otago, PO Box 56, Dunedin, New Zealand
Ph (+64-3) 4798319, Fax (+64-3) 479-8311, nkasabov@otago.ac.nz, mike@kel.otago.ac.nz

[2]Imaging & Sensing Team, Industrial Research Limited, PO Box 2225, Auckland, New Zealand
Ph (+64-9) 3034116, Fax (+64-9) 302-8106, email: D.Tuck@irl.cri.nz

ABSTRACT: *Data fusion is used today in many engineering and managerial applications to help resolve complex planning, control and optimisation problems. The purpose of this paper is to introduce practical and versatile tools and an environment for implementing data fusion that also provides a reverse-engineering methodology to extract comprehensible rules developed from the data. The environment has two major tools (among several others) - a fuzzy neural network FuNN, and evolving fuzzy neural network EFuNN applicable for both off-line and on line adaptive learning and rule manipulation. The EFuNNs allow for on-line fusion of variables over time sequences of information through adaptive learning. Two case study time-series applications are presented and discussed: a water flow prediction and a provisional robot control example.*

Keywords: Decision-making, On-line Prediction, Fuzzy Neural Networks, Evolving Fuzzy Neural Networks, Rule Extraction.

1. Introduction

While artificial neural networks (ANN) *per se* can provide the ability to produce a model for the mechanisms underlying the information in sources data used in decision-making and time-series prediction processes, hybrid neuro-fuzzy systems that include both learning from data and fuzzy rules manipulation, add much more to this useful property [1, 8, 16, 17]. Many past data fusion applications have utilised *ad hoc* designs at some level in the decision-making process to include explicit information or *a priori* knowledge constraints, and a structure to assist in highly dynamic applications or poorly defined problem solutions. This capability has been made available in the fuzzy neural network structures and in the hybrid connectionist-based environments described here.

One particular example for combining neural networks and fuzzy systems is the concept of fuzzy neural networks (FNN) [17, 8]. By fuzzifying a neural network, the quantisation of the inputs and outputs, through the application of membership functions, extra robustness is provided when used with redundant, noisy or incomplete input data. Further, this fuzzification technique can provide the means for extracting the information learnt in the form of rules. It is also now possible to add explicit information or *a priori* knowledge constraints to the network and thereby improve the interpretation of the rules learnt by the network, after training.

Here, two types of FNNs are illustrated as part of a hybrid software environment: the fuzzy neural network FuNN [8-11], used for off-line learning rule manipulation, and the evolving fuzzy neural network EFuNN [12-14] used for on-line real time learning and prediction.

Since the paradigm of hybrid connectionist-rule based systems was established [6] there are now several software environments that implement this paradigm. The first generation of such environments (see for example COPE [7]) implemented in a logical way, different types of ANN (such as multi-layer perceptrons, Kohonen self-organising maps [15], adaptive-resonance theory ANN [1]), to be combined with the

CLIPS-based production systems. Here, an ANN could be called for training, or for recall of the action part of the production rules [6, 7]. The second generation of such environments included fuzzy rules and fuzzy neural networks. Such an environment was FuzzyCOPE [8]. This new data fusion environment further developed the main principles of COPE [7] through a combination of the Fuzzy-CLIPS (an extension of CLIPS) developed by the NRC in Canada in 1994, http://ai.iit.nrc.ca/fuzzy/fuzzy.html and fuzzy inference and fuzzy-neural network modules at http://divcom.otago.ac.nz/com/infosci/KEL/home.htm.

The latest development in the series of FuzzyCOPE environments, FuzzyCOPE/3, allows for the extraction of a more comprehensible interpretation of the underlying rules implicit in the data used in training. It also has a module (EFuNN) for on-line learning where the inputs (sources of information) are not pre-defined and can vary during the on-line learning process, thus allowing for "on the fly" fusion of different sources of information and fuzzy rules.

2. The Fuzzy Neural Network

Fuzzy neural networks (FNNs) are connectionist models for fuzzy rules implementation and inference [8-11, 17]. However, there are a wide variety of architectures and functionality, differing in the type of fuzzy rules, type of inference method, and modes of operation. In general the architecture of these FNNs consist of five layers, Fig. 1. These layers in order are:

A. An input layer, where the neurones represent the linguistic variables of the input data;
B. A fuzzification, or condition layer, where the neurones represent the fuzzy values;
C. A rules layer, where the neurones represent the fuzzy rules;
D. An action layer, where the neurones represent the fuzzy values of the output variables, and finally;
E. An output layer, where the neurones represent the output linguistic variables.

The example illustrated in Fig.1 has two inputs, with two fuzzy membership functions (MF) each, two rule nodes, and two outputs, again with two MF each.

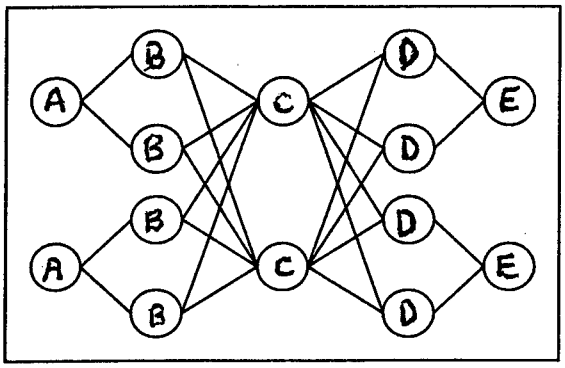


Figure 1: A general structure of a fuzzy neural network

FuNN is a FNN developed and presented in [8-11]. It is characterised by the following features: using weighted fuzzy rules [8]; modified back-propagation algorithms for training that include training with forgetting; using genetic algorithms, to improve and speed up training [2]; training with or without modifying the membership functions [11]; different types of rule extraction (e.g. simple fuzzy rules, weighted fuzzy rules, aggregated rules [10,11]); and rule insertion.

FuNNs have four basic advantages over ANNs (and standard fuzzy systems):

1. The FuNN structure is interpretable by fuzzy linguistic "if-then" rules – not so readily achieved for ANNs;
2. A FuNN is more likely to converge to a global minimum in error-weight space under arbitrary conditions, than an ANN;
3. FuNNs show a remarkable improvement in learning speed and accuracy compared to an equivalent ANN;
4. A FuNN can learn to predict signal variation well, even if it is of a chaotic signal type.

Usually, the FuNNs employ standard triangular membership functions and the number of rule nodes and rules are defined and fixed by the analyst prior to initialisation. However, FuNNs do have some difficulties when applied to on-line modelling and prediction [5], but these can be overcome by the evolving FuNNs as described below.

3. Evolving Fuzzy Neural Networks

Evolving fuzzy neural networks (EfuNNs) were introduced in [12-14]. In this extension of the FuNN architecture, the network begins with an empty rule layer. As training patterns are presented to the network, examples that are not adequately represented by the rule layer, trigger the addition of nodes to represent these new examples. Each rule node, after training, therefore represents one or several training examples.

EFuNNs have the following characteristics:

- Memory-based learning where exemplars of data are stored as they arrive at the inputs;
- Open structure – the number of the inputs and the outputs of the EFuNN can vary from example to example thus making fusion from an unknown number of sources possible in an on-line, "on the fly" mode, and;
- Local tuning of connection weights [12-14].

EFuNNs also exhibit the following advantages over conventional FuNNs:

- Rapid, one pass training;
- Good generalisation capability, both local and global;
- Robustness to forgetting, and;
- Rapid adaptation to new data.

4. The Hybrid Environment FuzzyCOPE/3

FuzzyCOPE/3 is a suite of data processing and neural network tools for the Microsoft Windows environment. FuzzyCOPE was developed by the Knowledge Engineering Laboratory of the Department of Information Science at the University of Otago. It consists of a graphical user interface built on top of a computational engine. The engine, which is encapsulated within a dynamic link library (DLL), is actually a simple command interpreter capable of creating and manipulating multiple instances of various classes of objects. These include data sets, multi-layer perceptrons, self-organising maps, and different types of fuzzy neural networks. Communication between the interface and engine is via customised formatted commands and result strings. These strings are assembled and parsed by specially written Application Programming Interface (API) libraries. This approach was adopted for maximum flexibility: it eliminates problems with handling C++ style pointers, it avoids problems with passing data in proprietary formats, it simplifies use of the engine (only the API library functions need be considered at the application level) and it lends itself readily to future expansion, such as a possible client-server architecture, or even the implementation of a specialised programming language. The FuzzyCOPE/3 environment is currently being used by more than 35 universities from all over the world as a teaching environment for courses in computational intelligence. There are also more than 200 developers of intelligent information systems using it. The environment is available from the web site at http://kel.otago.ac.nz/software/FuzzyCOPE3/

5. Case Study I - Water Flow Prediction

5.1 The Problem

This first example problem chosen for this paper was that of water flow prediction to a sewage plant (see also [8]). Given the time of day t, (0 - 23), whether or not it is a holiday (0 or 1), and the water flow over the past few hours (t-1, t-2 etc.), the task is to predict the water flow for the next hour. This is a time-series prediction problem useful for resource management. Accurate prediction of the water flow is necessary to allow for finer control of the sewage plant process.

The data is highly variable, with large differences between the hourly water flow for a workday as compared to a holiday. An extract of the data, shown in Fig.2, demonstrates the typical difference between holiday (dotted line) and workday (solid line) flows.

5.2 Experimental Data Sets

Two data sets, a training set and a testing set, were prepared. Each data set contained four input variables and one output variable. The input

variables represented the time of day, whether or not it was a holiday, and the water flow over each of the two preceding time intervals. There were 503 examples in the training set, 176 examples in the test set. Due to the requirements of the FuNN and EFuNN architectures, each data set was linearly normalised so that the values all reside within the range [0, 1].

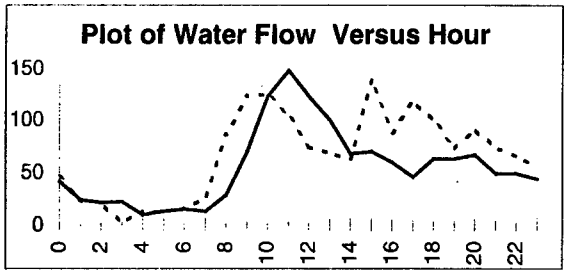


Figure 2: Extract of water flow data for holiday and workday (see text).

5.3 Off-line Training, Prediction and Rule Extraction with FuNNs

A fuzzy neural network FuNN was created within the FuzzyCOPE/3 environment. It consisted of four inputs, one for each input variable described above in 5.2. The first input had four MF attached (representing early morning, morning, afternoon and evening). The second and third inputs each had three MF attached (representing low, medium and high water flow). The final input had two MF attached (either a holiday, or not). Ten rule nodes were used, and the single output had three MF (again representing low, medium and high water flow).

This network was trained for 10,000 epochs using the backpropagation algorithm, and the results were recalled with the test data. The results of the recall are presented in Fig.3, where the actual (solid line) and predicted (dotted line) water flow are plotted.

After recall, a set of fuzzy rules was extracted. These rules seem to explain well the relationship between the input variables and the expected water flow. A set of example rules is presented below.

If <Time is EarlyMorning 4.63992> **and** <Flow_T-2 is Low 1.63653> **and** <Holiday is Is 1.77835>, **then** <Flow is Medium 3.81119>

If <Time is Morning 13.5842> **and** <Flow_T-1 is High 13.8779> **and** <Flow_T-2 is Low 5.44741>, **then** <Flow is Medium 1.86714>

If <Time is Afternoon 19.1327> and <Flow_T-1 is Low 24.77> **and** <Flow_T-2 is Medium 7.791> **and** <Holiday is IsNot 5.04419>, **then** <Flow is Medium 1.58037>

If <Time is Evening 6.96259> **and** <Flow_T-1 is High 7.72363> **and** <Flow_T-2 is Medium 3.65387>, **then** <Flow is Medium 0.955361>

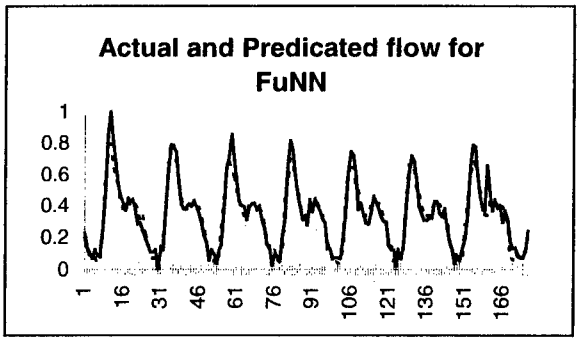


Figure 3: Plot of actual and predicted water flow for the trained FuNN.

5.4 On-line Prediction with EFuNNs

An evolving fuzzy neural network, EFuNN was first created with the same number of inputs and outputs (and input and output MFs). Because EFuNNs add rule nodes as required, the rule layer initially consisted of one node.

This network was then trained in an on-line mode, so that after the first data input vector had been presented, the network was next tested to predict the new hourly flow value. Finally, when the actual flow value became known, the input – output association was added to the EFuNN through a one-epoch adaptive training. Then the cycle repeats and the EFuNN was used to predict the next new value, etc. After the presentation of the first 75 examples two new inputs were added to the EFuNN without re-training the whole

system, these were the moving average 12 hours and the moving average 24 hours of the flow data. The EFuNN continued to grow. When the number of nodes reached 70 the EFuNN then started pruning the nodes as explained in [12-14]. A fuzzy rule for pruning was used based on the total activation of the rule nodes and the "age" (the time from creation). Fig. 4 presents the actual water flow (solid line) and the predicted (dotted line) on-line mode water flow.

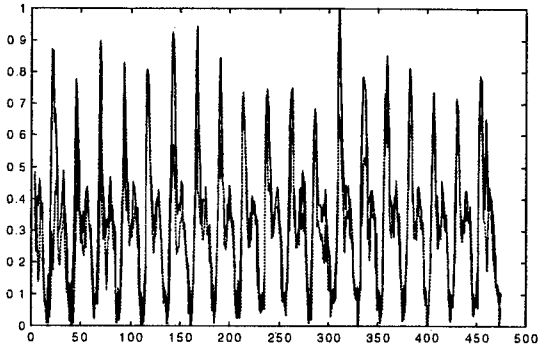


Figure 4: The actual and the predicted EFuNN on-line mode water flow.

It is clear that at the beginning the EFuNN could not predict well, not having any training or *a priori* knowledge. The more it was trained on the incoming data the better the prediction became.

EFuNN simulators are written in MATLAB and C++ and are part of the NZ-RICBIS – the New Zealand Repository for Intelligent Connectionist-based Information Systems. This is available at http:divcom.otago.ac.nz/infosci/kel/CBIIS.html

The water flow data is available from http://kel.otago.ac.nz/software.

5.5 Comparative Analysis of the Different Fusion Techniques for the Water Flow Prediction Problem

Both the FuNN and EFuNN were able to approximate the data to a reasonable degree of accuracy. However, while the FuNN required 10,000 training epochs (taking approximately 20 minutes on a 233-Mhz Pentium II), the EFuNN required only one pass through the training data, taking less than 20 seconds. It is this rapid training capability that is one of the major advantages of EFuNNs. Rules from an EFuNN can also be extracted and inserted [12-14].

6. Case Study II - On-line Robot Control

6.1 The Problem

In a New Zealand meat-works, a sheep is valued for both its pelt and meat products. Lamb meat is an important export product and the fluffy sheepskins make great souvenirs for our tourist visitors.

In order to remove the carcass pelt without damage to itself or the flesh underneath, extreme care is required in the initial cutting operation of the skin. For the purpose of this example, a new robot cutting path planner approach has been investigated. At present an algorithmic path planning robotics system has been developed and is being trialed in a New Zealand meat-works, so far showing great potential over the traditional manual butchering preparation. However, this current approach is somewhat limited by the rather restricted algorithmic method of the semi-automated implementation developed.

We have started to explore use of the FuNN tool from FuzzyCOPE/3 to first develop a model of this current algorithmic planner. Then later, if the model demonstrates success, we propose to pursue and utilise the on-line adaptation properties of the EFuNN to continue learning to compensate for the highly variable sizes and shapes of this natural product (sheep). The present the algorithmic method allows for some on-line modification to the cutting path planning, when sheep variations demand it, but only by manual operator intervention through the adjustment of certain parameters which effect the two cut intersection point in the Y-Z plane.

6.2 Experimental Data Sets

The carcass de-pelting process starts with what is termed as a "Y-cut", performed on the sheep carcass while hanging upside down on a moving

conveyor chain, Fig. 5. This skin Y-cut, really two separate cuts, begins with one front leg at the hoof and follows down that leg and across the lower neck/chest region of the carcass (also known as the brisket), terminating just past the midline of the body. A second cut is then carried out following a similar path, but mirroring the first cut, beginning at the hoof of the second front leg, continuing down it to finish just past the point of intersection with the first cut. When the completed Y- cut is performed correctly, the pelt can be pulled off the carcass as a whole piece and with minimal damage.

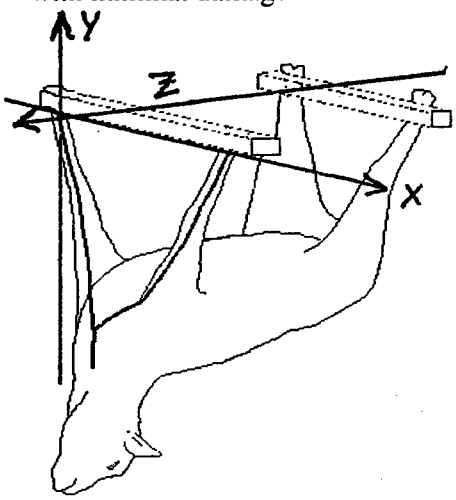


Figure 5: An example of the carcass Y-cut path.

For this second time-series study, the sheep carcass Y-cut sensor data was used, together with the algorithm path data, for training with a fixed parameter setting. Three sensors provide three-dimensional measurements of important points on the carcass so that the robotic skin cutting operation can be planned. These measurements are: the separation between the two front hooves; the highest point on the brisket; and finally the horizontal offset between the brisket and the trachea region of the neck. At present, the *ad hoc* algorithmic intersection point for the two cuts is determined by manual parameter settings. In a future development of an EFuNN multi-sensor data model to determine corrections to the algorithm calculations, we aim to fully automate this the path prediction despite the sheep variations by using the on-line learning and adaptation mode of the EFuNN.

Because the carcass is hung from an overhead conveyor line from its hooves the starting points are easily identified and provide the [0, 0, 0] reference in space for the cut. However, the meeting point of the cuts and their paths down the front legs of the carcass in space are very much dependent on the size and breed of the animal. Also, because the carcass is continually moving along the conveyor line, the cut intersection point needs to be accurately determined and tracked, although cutting is assisted by design of the hook shaped knife. The shape pulls the skin away from the flesh and helps ensure the knife just cuts through it.

6.3 Preliminary Results - Training Path Planning with FuNN

An off-line fuzzy neural network cutting path planning model is being developed using FuNN to predict the next knife position for time, t. The input consists of 12 nodes, each having five membership functions (MFs) for fuzzification. The first three inputs are the X, Y, and Z carcass sensor measurements made on each sheep as described above. The next three inputs are the x, y, and z coordinates of the last (t-1) knife position. The final two sets of three inputs are the time lagged (t-2) and (t-3) coordinate positions. Three output nodes [x_o, y_o, z_o] with 7 MFs each generate the 3-D predicted cutting path sequence.

Data for the Y-cuts, taken from 83 sheep were used for this preliminary investigation - 50 for training and 33 for testing. The curent algorithm generated the time-series sequence of 100 cutting path positions, each sheep, which control the robot arm manipulation of the cutting knife. A limited range of animals sizes and shapes were included. Each carcass cutting path data set of 100 vectors contained the 12 input data values (X, Y, Z measurements followed by the three time lags of the previous knife positions), and then the next [x_o, y_o, z_o] predicted position for the knife, to be learnt.

The best results obtained so far have been with a 15 node rule layer and after only 100 training epochs. Further experimentation is obviously

required to refine the model. Figures 6 and 7 display typical results of a single cut for one sheep, x versus y and z versus y respectively, with the actual (solid line) and the FuNN predicted (dotted line) cut paths superimposed. The average RMS differences are 4.6, 11.8 and 8.5 (mm) for the x, y, and z directions respectively for 50 carcasses.

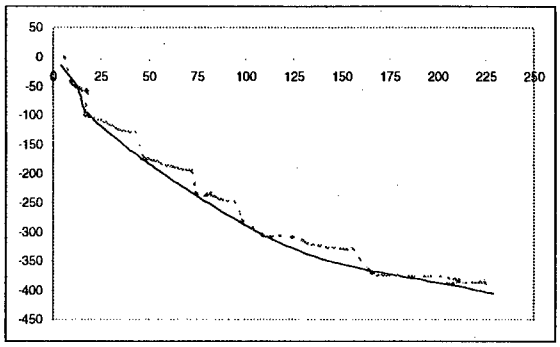


Figure 6: Typical x-y cutting path (mm) from FuNN.

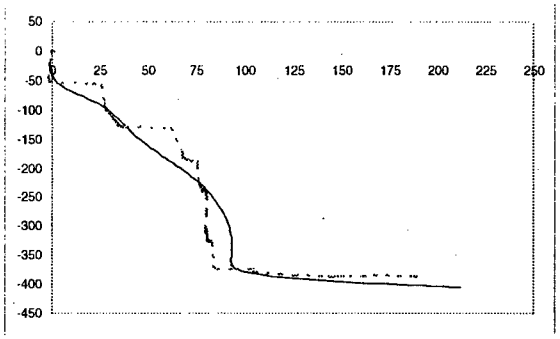


Figure 7: Resultant z-y cutting path (mm) for Fig. 6.

7. Discussion and Conclusions

Connectionist-based algorithms are robust when the appropriate techniques are used. They allow the analyst to learn relationships between the input and output variables without making assumptions about the data distribution. Thus, improving the prediction or classification accuracy is based on updating the transfer function and not manipulating the incoming data flow. Also the fuzzified connectionist-based algorithms may now require fewer training examples than traditional sensor data fusion methods. The results of ANNs and FuNNs, over fuzzy rules and more traditional statistical methods can be shown to have a distinct advantage [8]. For example, the adaptive learning algorithms enable the EFuNNs to learn relationships between input data and output data in an iterative way [12-14] and on-line. Finally, fuzzy rules may then be extracted and updated from all the classes of FuNN to help explain what the network has learned.

When using FuNNs and EFuNNs one should always refer to traditional statistical methods and compare the results with them, if possible. However, there exist disadvantages in applying statistical algorithms to determine the input-output transfer function characteristics. First, this approach requires large amounts of sample data for processing. Second, it is not capable of handling conflicting information that can arise in the transfer function it is trying to model and this cannot be updated without changing the input data - there is no feedback process for statistical algorithms to learn from *a posteriori* knowledge. For example, they do not cope well where the data distribution is bimodal or very non-normal, which are the case here. Also, the sensitivity for the separation between output states is a function of all the inputs, so closely positioned states are not well distinguished. However, statistical methods can suit some models where the data is uni-modal and normal. Then this approach has the advantages of being computationally efficient and capable of producing highly accurate results.

In the first study, two of the hybrid neuro-fuzzy modules of FuzzyCOPE/3, FuNNs and EFuNNs have been demonstrated and in the second case study a preliminary FuNN application looks promising. Work on this robotic path planning problem is to continue and it is expected that a fully automated solution can be developed. While the modules performed acceptably in both cases, it is expected that recurrent versions of these networks, scheduled to be included in the next version of FuzzyCOPE, will yield even better results.

The objective of this paper has been to promote awareness of this new and versatile data fusion, FuzzyCOPE/3 environment, and to entice others to investigate and apply it to new real world

problems. The results presented here, hopefully demonstrate the potential of this fusion environment for providing solutions to previously difficult to-solve-problems.

Acknowledgments

This research is partially supported by the Foundation for Research, Science and Technology, New Zealand through a grant UOO808 to the University of Otago, and NSOF at Industrial Research Limited. Also thanks must go to Dr Malcolm Taylor for the acquisition and generation of the robotic cutting path data used in the second case study.

References

1. Carpenter, G. S. Grossberg, N. Markuzon, J.H. Reynolds, D.B. Rosen, "FuzzyARTMAP: A neural network architecture for incremental supervised learning of analog multi-dimensional maps," IEEE Transactions of Neural Networks , vol.3, No.5, 698-713 (1991).
2. Goldberg, D.E., Genetic Algorithms in Search, Optimisation and Machine Learning, Addison-Wesley (1989).
3. Hassibi and Stork, "Second order derivatives for network pruning: Optimal Brain Surgeon," in: Advances in Neural Information Processing Systems, 4, 164-171, (1992).
4. Hech-Nielsen, R. "Counter-propagation networks", IEEE First int. conference on neural networks, San Diego, vol.2, pp.19-31 (1987).
5. Heskes, T.M., B. Kappen, "On-line learning processes in artificial neural networks", in: Math. foundations of neural networks, Elsevier, Amsterdam, 199-233, (1993).
6. Kasabov N. Hybrid connectionist rule based systems. In Artificial Intelligence IV Methodology, Systems, Applications. P.Jorrand and V.Sgurev eds. Amsterdam, North-Holland (1990) 227- 235.
7. Kasabov, N. COPE-a hybrid connectionist production system environment. In *Proceedings of the Third Australian Conference on Neural Networks (ACNN'92).* Sydney, Sydney University Electrical Engineering (1992) 135-138.
8. Kasabov, N., Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering, The MIT Press, CA, MA, (1996).
9. Kasabov, N. "Adaptable connectionist production systems". Neurocomputing, 13 (2-4) 95-117, (1996).
10. Kasabov, N., "Learning fuzzy rules and approximate reasoning in fuzzy neural networks and hybrid systems", Fuzzy Sets and Systems 82 (2) 2-20 (1996).
11. Kasabov, N., Kim, J., Watts, M. and Gray, A. Architecture for adaptive learning and knowledge acquistion, Information sciences, 101 (3-4_: 155-175 (1997).
12. Kasabov, N., "Evolving Fuzzy Neural Networks - Algorithms, Applications and Biological Motivation", in: in: Yamakawa and Matsumoto (eds), Methodologies for the Conception, design and Application of Soft Computing, World Scientific, 1998, 271-274.
13. Kasabov, N., "ECOS: A framework for evolving connectionist systems and the ECO learning paradigm", Proc. of ICONIP'98, Kitakyushu, Japan, Oct. 1998, IOS Press, 1222-1235.
14. Kasabov, N. The ECOS Framework and the ECO Learning Method for Evolving Connectionist Systems, Journal of Advanced Computational Intelligence, 2 (6) 1998, 1-8.
15. Kohonen, T., Self-Organizing Maps, second edition, Springer Verlag, 1997.
16. Tuck, D., "Preliminary CBIS Findings: Forecasting a Variable Multiple Cyclic Process with Confidence!" Proc. ICONIP'97, ANZIIS'97 &ANNES'97, Dunedin, NZ, November, 1997, Springer, Vol.2, 992-995.
17. Yamakawa, T., H. Kusanagi, E. Uchino and T. Miki, "A new Effective Algorithm for Neo Fuzzy Neuron Model", in: Proceedings of Fifth IFSA World Congress, 1017-1020, (1993).

A Hybrid Artificial Intelligence Architecture for Battlefield Information Fusion

Paul G. Gonsalves, Gerard J. Rinkus, Subrata K. Das and Nick T. Ton

Charles River Analytics, Inc.
725 Concord Avenue, Cambridge, MA 02138 USA
Email: {pgonsalves, grinkus, sdas, nton}@cra.com

Abstract *- The processing of tactical information and the associated situation assessment of the tactical battlefield is a major task for military personnel. Significant effort has been made in countering this challenge with advances in sensor capabilities and enhancements in avionics, electronics and C4I (command, control, communications, computer and intelligence) systems. This rapid evolution must be met with concomitant advances in information fusion and situation assessment. Additionally, a rapid verifiable means is needed in situ for management of sensor and information assets. Here, an on-going effort to develop a hybrid artificial intelligence architecture for battlefield information fusion is described. The architecture incorporates three distinct modules: a low-level information fusion module incorporating a fuzzy expert system manager; a situation assessment module incorporating a fuzzy logic based event detector and a Bayesian belief network component for generating probability measures of situational state; and a fuzzy expert system based module for collection or sensor management.*

Keywords: Information Fusion, Situation Assessment, Belief Networks, Fuzzy Logic

I. Introduction

The analysis of intelligence data to generate a comprehensive understanding of all tactical elements within the battlespace and their likely evolution, i.e., to achieve situation awareness is a major task for military personnel. This task naturally overlaps with and benefits from the tasking and management of the sensor/collection assets themselves. Here, we develop a hybrid artificial intelligence (AI) architecture that provides an integrated framework for analysis of information in support of enhanced tactical awareness and needs-based sensor asset management to assist in battlefield intelligence processing. The architecture's flexibility stems from combining two AI techniques for model-based approximate reasoning: fuzzy logic and the Bayesian belief networks.

Information fusion strives to combine information from multiple sources into information that has greater benefit than would have been derived from each of the contributing parts. An obvious analogy exists between fusion and human cognitive processing, in particular, the way humans process multi-sensory information (i.e., sight, sound, smell, etc.) to make inferences regarding the environment. Our hybrid AI battlefield information fusion system uses a coordinated application of two artificial intelligence technologies, fuzzy logic (FL) and Bayesian belief networks (BNs), to the problem of tactical fusion and collection management. Fuzzy logic [1] provides a means of converting low-level imprecise information in non-numerical format into mid-level knowledge units about individual battlespace entities. Belief networks [2] [3] provide a means for constructing and maintaining a hierarchical, probabilistic model linking multiple entities, at various levels, in the context of the overall mission goals, rules of engagement, etc. Evidence gathered incrementally and in real-time first undergoes FL filtering and is then applied to the appropriate node(s) of the BN. This evidence then automatically propagates throughout the BN resulting in revised probability estimates concerning the higher-level tactical situational hypotheses. Experiences from prior research efforts [4][5] have shown that this approach provides an effective solution to the problem and offers a natural framework for encoding complex tactical knowledge.

II. System Description

Figure 1 illustrates how the overall scope of the hybrid architecture for battlefield information fusion falls within the various levels of fusion [6] and other key components of tactical C4I systems. Information concerning the various entities present in the battlespace, are collected by a variety of sensor or collection assets (JSTARS, AWACS, etc.) and then fused (level one) within the architecture to generate individual target tracks and to classify and characterize targets. The situation assessment (SA) module of the architecture uses this fused track data to generate a probabilistic situational state hypothesis

from detected events. This SA information is then forwarded to level 3 impact or threat assessment and decision-aiding modules, currently outside this effort's scope. Finally, the SA information is used by the architecture's sensor or collection management module to assign, prioritize and communicate intelligence requests.

schemes, etc. For correlation, the FL Manager also specifies algorithm selection and threshold levels with final oversight of assignment. For filtering/prediction management, the manager specifies algorithm type, filter parameters and model choice. For example, the FL Manager may inspect residuals from a bank of Kalman filters to determine

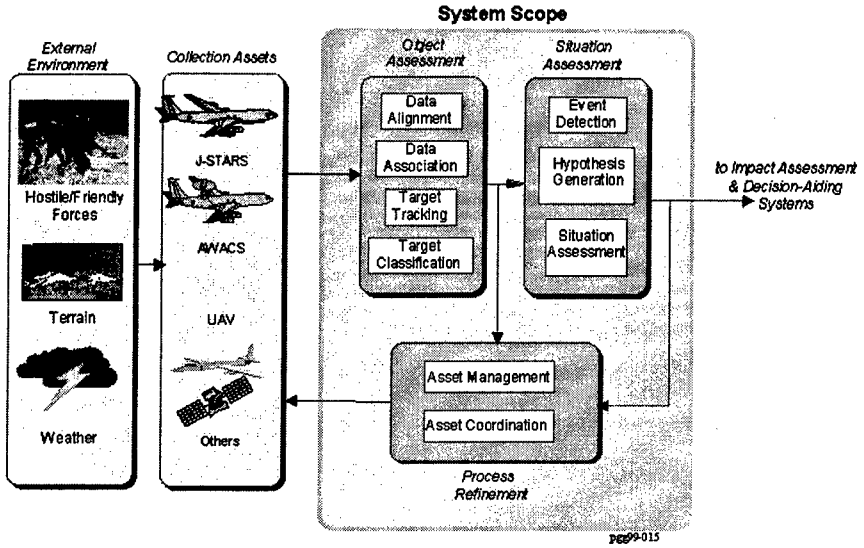


Figure 1: Scope for Battlefield Information Fusion

Figure 2 displays the overall architecture, incorporating all modules necessary to support management and control of level one fusion processing, situation assessment and enhanced collection management functionality. The system incorporates three specific and distinct modules: a fuzzy logic based level one fusion module responsible for management and control of observation-to-track gating and assignment, state estimation, and track database management; a combination fuzzy logic based event detection and belief network based level two situation assessment module responsible for generating probabilistic hypotheses for high-level situational state descriptors; and a fuzzy logic based level four collection management expert system responsible for mapping informational requirements and current state information into asset resource requests

The architecture shown in figure 2 encompasses all aspects of level 1 object assessment fusion processing including data association, state estimation, identification and track management. The Fuzzy Logic Manager for level 1 has direct responsibility for management and oversight of these level 1 functions. Specifically, data association management provides gating technique selection, gating parameter modification (e.g. gating constant for rectangular gate), use of multi-level gating the most appropriate model or for target maneuver detection. It may also update measurement noise models based on target range (i.e. increased angular measurement accuracy with decreasing range for a radar sensor) or based on sensor confidence levels. The FL Manager also monitors and controls the track identification process. Here, again algorithm selection and output monitoring are its key functions. The final element of the level 1 FL Manager is track management. Responsibilities for track management include track initialization and confirmation (based on data association results), as well as track deletion. Specific items addressed in track management include, track initiation criteria, track confirmation logic including required number of assignments and time window, and specification of last update time threshold for track deletion.

Level 2 processing within the hybrid architecture for battlefield information fusion of figure 2 has two primary functions: detection of key events and assessment of the current situation. Event detection is performed using FL reasoning in conjunction with a pre-defined library of domain-relevant events. This event library is of a broad enough nature to encompass typical tactical engagements. Event detection automatically translates information gleaned from incoming level 1 information into domain-relevant events (e.g. presence of specific

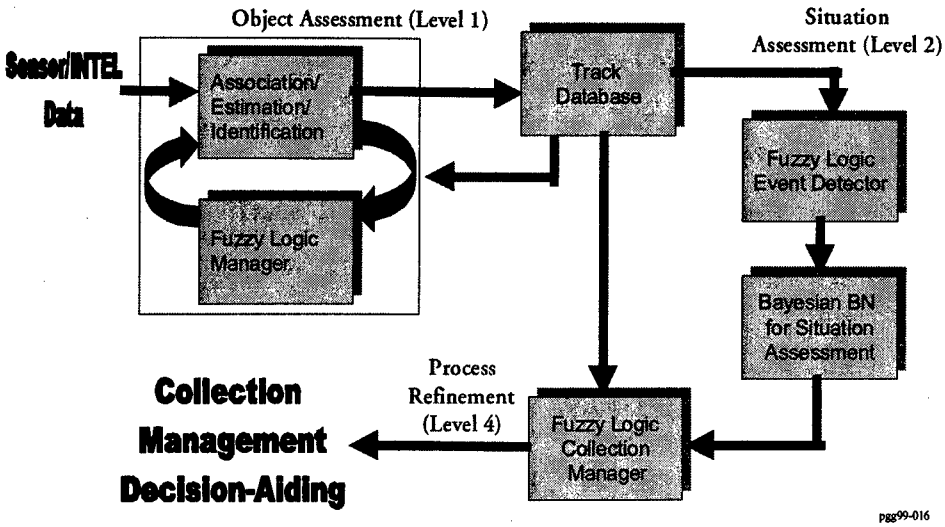


Figure 2: Hybrid AI Architecture

enemy units at a specific location), along with an associated measure of certainty of the event. Key events are then sent to a belief network that determines current situation.

At the heart of level two processing is a belief network which is a probabilistic model of the battlefield tactical situation. The belief network allows uncertain evidence concerning any of the represented battlefield and unit features to be incorporated so as to consistently update any other contingent features of the model. The network, shown in figure 3, can be interpreted as representing causal relationships between the variables. For example, a particular enemy mission (**E.Miss**) combined with enemy knowledge about where friendly forces are located (**F.Loc**) cause a rational enemy to choose a specific objective (**E.Obj**) which will maximize its utility. The possible values for variables **E.Miss** and **F.Loc** are shown next to the nodes. Similarly, the choice of a specific objective causes a rational enemy to choose a specific route or course of action (COA), denoted by the node **E.COA**, that maximizes its utility in prosecuting that objective. The bottom-most nodes (**MC-1**, **MC-2**, ...etc.), represent the belief that the enemy is present within the specific regions of the battlefield termed mobility corridors (MC). While we interpret these relationships as causal, we represent the inherent uncertainty of the battlefield environment by encoding them probabilistically. Specifically, each

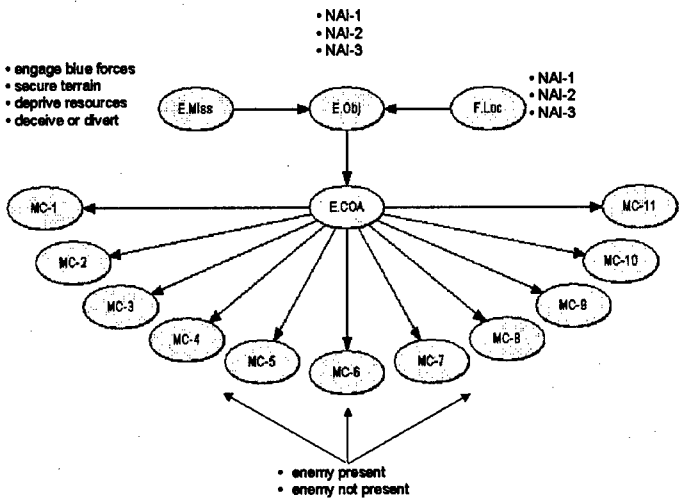


Figure 3: Belief Network for Situation Assessment

link between nodes has a corresponding conditional probability table (CPT) which encodes the probability of the child variable given the parent variable. In the more general case in which a node may have more than one parent, the node's CPT encodes its probability given all of its parents.

At level four, a Fuzzy Logic Collection Manager maps current situation Assessment State, and enemy track information, into sensor/INTEL requests. The mapping is performed using a repository knowledge of sensor/asset capabilities and enemy tactical doctrine. High-level event notifications and observations relating to asset requests are also relayed to the user. The mapping from situational state and track information to asset request is based on several appropriateness metrics including timeliness, desired classification level, availability, and geographic coverage. Timeliness refers to an asset's turnaround time to meet a given request. Classification level refers to the asset's classification capabilities, i.e., detection (find enemy units), classification (discriminate enemy units, tanks vs. APCs), and identification (type or model of tank). Availability refers to the time period in which the asset is accessible.

III. Prototype Demonstration

To assess feasibility and demonstrate the hybrid architecture for battlefield information fusion, a battlefield scenario was developed by subject matter experts covering a 24-hour period in which friendly ground forces, a mechanized infantry brigade, defend against a Soviet-like adversary consisting of a motorized rifle division (MRD). A terrain analysis/IPB stage results in a constrained set of possible enemy objectives, courses of action, and mobility corridors. Friendly intelligence-gathering assets include ground-based reconnaissance units, electronic support measures (ESM) equipment, reconnaissance aircraft and the multi-mode radar capabilities of the airborne J-STARS platform.

The level one fusion simulation consists of: a) a main window (see Figure 4) which displays the evolution of the battle; and b) a track database window that displays the current associations of individual sensor reports to tracks.

We tested three variations of our main scenario. The overall qualitative conclusions derived from these simulations can be summarized by the following [7]: a) fuzzy logic provides a natural human-like reasoning mechanism for handling uncertainty; and b) the level one Fuzzy Logic Manager was able to discriminate multi-level unit types, perform track generation and maintenance, and aggregate lower echelon units into higher echelons. In our scenario, the fusion manager was able to discriminate between battalion and regimental units. Gating and assignment control ensured reasonable track maintenance. Finally, the fusion manager could aggregate lower units into higher echelon units, e.g. battalion units into regiments.

The level two demonstration entailed the sequential posting of sensor/INTEL evidence to the BN model of figure 3. The results showed that the model was able to maintain correct hypotheses regarding the higher-level (hidden) variables, e.g., enemy objective, for the range of scenarios. These results demonstrate the feasibility of the belief network framework for modeling causal battlefield relationships. A single, integrated model combines variables of differing scales and allows probabilistic inferencing of higher-level, hidden variables, e.g., enemy objective, intent, etc., based on evidence concerning lower-level variables, e.g., enemy unit locations, types, movements, etc. The belief network formalism simultaneously allows a consistent means for combining prior information, e.g., derived from terrain analysis/IPB, weather reports, and enemy doctrine and order of battle information, with evidence gathered in real-time from sensor assets and units deployed in the battlespace.

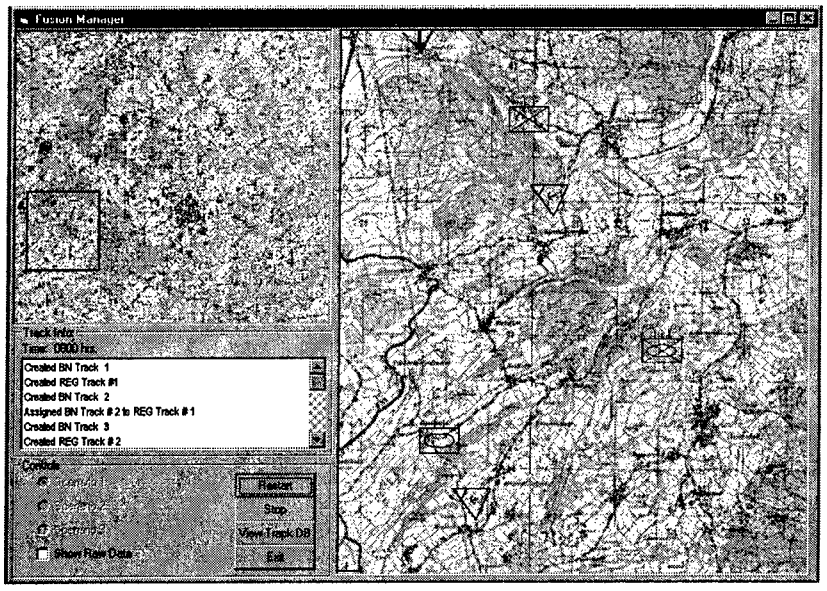


Figure 4: Main Window for Level 1 Simulation

The level four demonstration tested the fuzzy rulebase for collection or sensor management. The system displayed basic capabilities for combining hypothesized unit locations and intents with friendly intelligence requirements and asset capabilities to produce asset requests sufficient to acquire the needed intelligence. The fuzzy expert system rulebase contains over 100 rules and assumes an asset suite consisting of the JSTARS platform, including both moving target indicator (MTI) radar and imaging synthetic aperture radar (SAR), and a generic electronic support measures (ESM) platform. The rulebase uses several fuzzy variables including sensor resolution, timeliness, availability, area coverage, and a user-specified information criticality level.

Figure 5 shows the main interface window for the Fuzzy Logic Collection Manager prototype. As shown, the graphical user interface (GUI) has two sets of edit boxes, a listbox, a textbox, and two buttons. The two sets of edit boxes provide the means to directly input BN node values from level two processing. These values correspond to the presence of enemy units at the eleven mobility corridors or segments (refer to figure 3) and to the belief in the three possible enemy objectives (A, B, or C referring to NAI 1, 2, or 3, respectively). These sets of edit boxes are at the top left and top right of the main screen, respectively. Below, the set of edit boxes for segments (or mobility corridors) is a listbox in which the user can specify the criticality value for the asset request. The textbox below the label "Asset Request" is where the Fuzzy Logic Collection Manager output is displayed. Figure 5 shows the Fuzz Logic Collection Manager after inferencing. The results shown are for the case where we have ascertained (via BN belief network level two processing) enemy objective is A (or NAI 1), user criticality is low, and no substantial enemy location information. The results shown at the top of the "Asset Request" textbox map the inputs into the informational requirements. That is, the informational requirements specify the request priority level, the coverage area, and type of coverage required. As shown, since enemy objective is A, then we want enemy detection in segments or mobility corridors 9, 10, or 11. At the bottom of the textbox is listed the corresponding assets meeting the informational requirements. In this case both the MTI radar and the ESM meet the requirements.

IV. Current Work

Current efforts on extending the hybrid AI architecture for battlefield information fusion are focusing on: a) integration of the three major modules (for levels, 1, 2 and 4) to produce a full-scope system for enhanced battlefield information processing and situation assessment; b) incorporation of temporal/spatial aspects of battlefield information processing to enhance current situation assessment and to facilitate prediction of future enemy evolutions; c) evaluation of a full-scope prototype in an empirical study employing multiple tactical scenarios; d) system enhancement based on the evaluation findings; and e) specification of H/W and S/W requirements for follow-on development within fielded C4I-related information processing systems to enhance overall information fusion, situation assessment, and collection management. Additionally, a parallel effort is underway to develop a level 3 (impact assessment) component with the functionality to infer enemy intent, capabilities and vulnerabilities, and how that component could be integrated within the hybrid AI architecture.

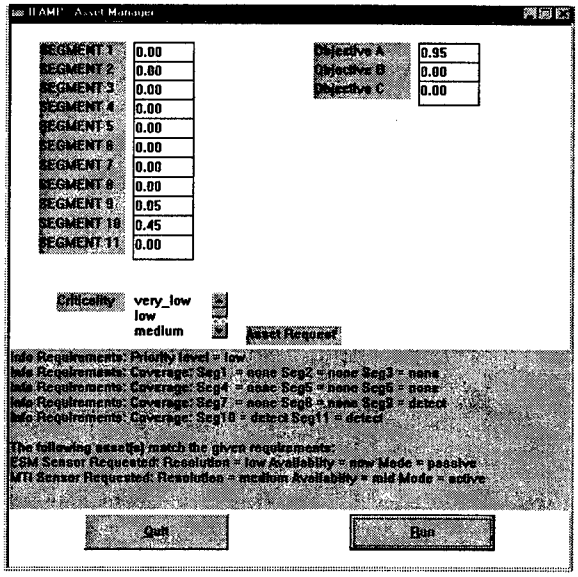


Figure 5: Fuzzy Logic Collection Manager

V. Conclusions

We have designed and developed a limited-scope prototype hybrid AI system for battlefield information fusion incorporating three modules: a fuzzy logic-based level one fusion module for low-level fusion management; a belief network-based level two situation assessment module for generating probabilistic hypotheses for high-level situational state descriptors; and a fuzzy logic-based level four collection management system for mapping information requirements and state information into asset requests. Basic system feasibility was shown by

exercising the system using variations of a specified tactical battlefield scenario.

VI. Acknowledgement

The work detailed in this paper was performed under USAF Contracts F30602-97-C-0208 and F30602-98-C-0041. The authors thank the Technical Monitor, Mr. Michael Hinman, of the Air Force Research Laboratory Information Directorate (Rome, NY) for his guidance and support.

VII. References

[1] Zadeh, L. A. (1965). "Fuzzy Sets." Information and Control 8: 338-353.

[2] Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. San Mateo, CA, Morgan Kaufmann.

[3] Lauritzen, S. and Spiegelhalter, D. (1988). "Local Computation with Probabilities in Graphical Structures and Their Applications to Expert Systems." Journal of the Royal Statistical Society B, 50(2).

[4] Miao, X., Zacharias, G., and Kao, S. (1997). "A Computational Situation Assessment Model for Nuclear Power Plant Operations." IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans, Vol. 27, No. 6 (November)

[5] Mulgund, S., Rinkus, G., Illgen, C. and Zacharias, G. (1997). "Situation Assessment Modeling and Pilot State Estimation for Tactical Cockpit Interfaces." Presented at Human-Computer Interaction International 1997, San Francisco, CA (August).

[6] Steinberg, A., Bowman, C. and White, F. (1998). "Revisions to the JDL Data Fusion Model." Presented at the Joint NATO/IRIS Conference, Quebec, Canada.

[7] Gonsalves, P., Rinkus, G. and Zacharias, G. (1998). Intelligent Fusion and Asset Management Processor (IFAMP). Final Report R97061, Charles River Analytics, Cambridge, MA.

Session WA3
Fusion for Target Tracking II
Chair: S. Musick
Air Force Research Laboratory, USA

A possibilistic approach of high level tracking in a wide area

O. WALLART, C. MOTAMED, M. BENJELLOUN
Université du Littoral Côte d'Opale
Laboratoire ASL - EA 2600
195 av P.L.King 62228 CALAIS, FRANCE

Abstract *The objective of this work is the ability to track multiple objects in a wide outdoor area. Numerous fixed vision sensors have to be spatially distributed. In general, it is not possible to cover the whole scene, so the sensors are separated into blind zones, for which we do not have any observation. Our approach that we call 'high level tracking' is based on the co-operation of sensors in order to obtain a global motion interpretation. The main difficulty is to ensure a robust matching of mobile objects perceived by several sensors from different locations at different moments. In order to model and take into account uncertainties, we have decided to use the Possibility Theory. Thus we use a measurement of necessity expressing the matching decision quality.*

Keywords: Distributed sensor, Co-operative vision system, Object tracking, Possibility Theory

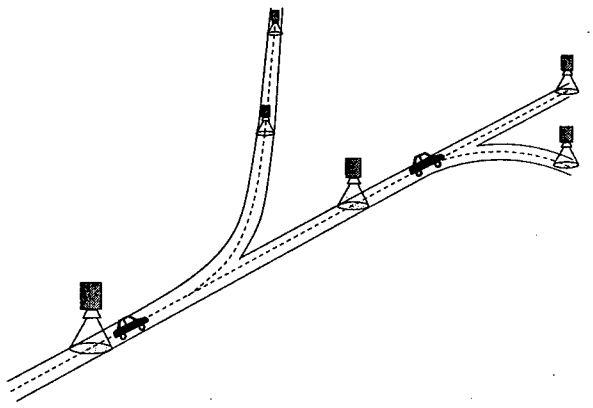


Figure 1: Envisaged application

1 Introduction

The development of distributed vision systems carrying out sites monitoring is an interesting field of investigation. Indeed, motivations are multiple and concern various domains as monitoring of specific sites (nuclear thermal power), control and estimation of flows (airport, port, motorway), continuous coverage over large battle field areas . Because of the rapid evolution in the field of data processing, communications and instrumentation, such applications become possible. Vast research programs have been launched such as VSAM(Video Surveillance And Monitoring) financed by DARPA, SMART by the European Community, CDV(Cooperative Distributed Vision) in Japan....

Our approach that we call 'high level tracking' is based on the co-operation of sensors in order to obtain a global motion interpretation. The originality of this work is the handling of uncertainties and imprecisions related to the system. They have various origins and come essentially from predictions carried out in blinds zones, i.e. for which we do not have any observation, and from sensors which operate in outdoor scenes. In order to model and take into account uncertainties, we have decided to use the Possibility Theory.

As we are interested in traffic monitoring in a urban or motorway context, an application has been envisaged (figure 1). The configuration of each sensor is tuned in order to make objects recognition task easier.

In this article, we present the multi-sensor tracking architecture that we have envisaged. Then, we explain how predictions are carried out in blinds zones, i.e. which are located between sensors. Finally, we show how data acquired by each sensor are combined in order to track mobile objects in the whole scene.

2 Architecture

In order to track mobile objects over the scene, we have decided to use several sensors which are spatially distributed and have different fields of views. As it is not possible to cover visually all the scene, sensors are separated by blind zones, i.e. for which we do not have any observation. We can mention the works of Rombaut [1] and Hutber et al [2] in multisensor tracking with small blind zones.

The use of multiple sensors gives rise to one problem : how all the sensors could be connected together (organization and architecture).

First, the architecture that we have considered is presented, then we show how communications between sensors are carried out and how the tracking is managed with this architecture.

2.1 Envisaged architecture

We have decided to use a fully decentralized architecture. This architecture has no central processing facility, no centralized communications medium. The structure of this architecture is equivalent to a network of intelligent sensor nodes. Each sensor node is autonomous, it has its own processing element and its own communications facilities. Communication can take place between any two connected sensor nodes. Each node can assimilate and receive information independently from other nodes.

This type of architecture has many advantages [3]. Among the principal ones, we can quote the facts that it is completely modular and also that it ensures the maximum benefit derived from the use of multiple sensors. In particular, it is robust to the loss of sensors. It can use different varieties of sensors working in parallel.

2.2 Communication between sensors

The co-operation between spatialy distributed sensors is based on message transmissions which present some specificities.

First, it concerns messages content. We have decided that messages must contain only the necessary information for the tracking, thus their sizes are reduced. In our applications, information stored in the messages is visual primitives (color, size, texture), dynamic characteristics and temporal predictions.

Then, it concerns messages transmission. Only sensors, likely to perceive an object, can receive messages. This approach allows an optimal management of communications and a simplification of matching process by activating only the useful resources for the tracking.

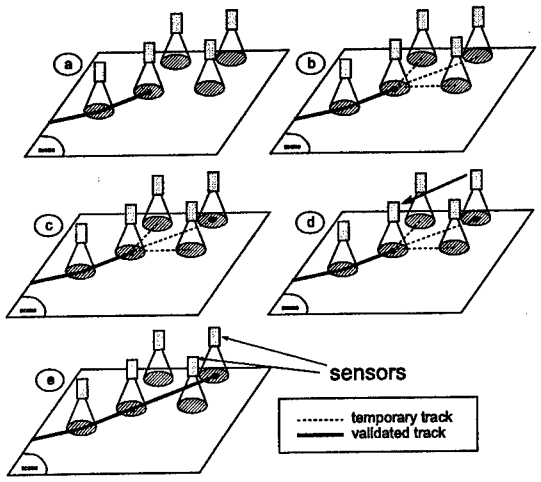


Figure 2: Tracks management

2.3 Tracks management

Tracks associated with a mobile object (i.e. initialization, maintenance and termination) are managed by the sensor which has initialized them.

As soon as a mobile object is perceived (figure 2.a) by a sensor, temporary tracks associated with possible trajectories of the object in blind zones are initialized (figure 2.b). If a close sensor recognizes the object (figure 2.c), then the sensor which has initialized temporary tracks is informed (figure 2.d) in order to valid the current track connecting the two sensors and to remove the others (figure 2.e). This management mode has been motivated to achieve efficient tracks termination.

To realise this track management, it is essential that sensors are able firstly, to predict displacement of mobile objects in blind zones and secondly, to match perceived objects with ones which are likely to be perceived called "awaited objects ".

3 Temporal prediction in blind zones

We have decided to use fuzzy temporal curves of events described by Dubois and Prade [4] [5](DOP: Domain Occurrence Possibility) (figure 3) in order

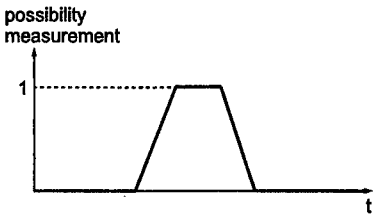


Figure 3: A fuzzy DOP

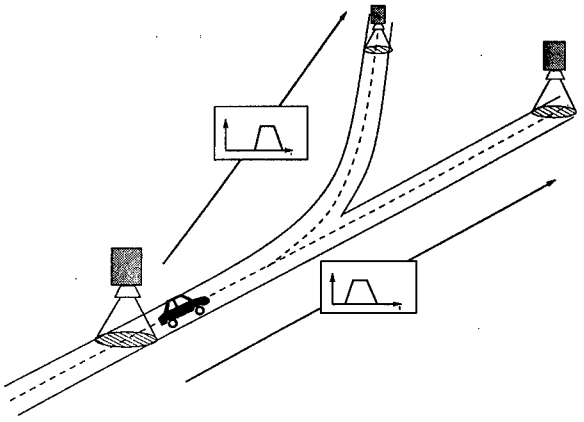


Figure 4: Transmission of possible predictions (DOP) associated to a mobile object

to predict object motion in blind zones. This choice is motivated by the fact that we work in wide outdoor scenes including blind zones. Under such conditions, we must be able to manage uncertainties related to the system. It is thus necessary to build rough models tolerating ranges of variation for the numerical parameters.

The DOP(Oi,Sj,Sk) is the prediction generated by sensor Sj, explaining the appearance possibility of a recognized object Oi in the field of a specific sensor Sk (figure 3). They are then transmitted to the closest sensor likely to perceive the mobile object (cf. figure 4).

The creation of a DOP depends on the context and dynamic characteristics of the mobile object [6]. We thus developed a method allowing automatic generation of DOP according to this knowledge.

After defining the notion of context, we propose an approach for DOP generation.

3.1 Contextual informations

The definition of the context of a process depends on the process nature and is all the additional information needed by the process to work efficiently. For our distributed interpretation system, contextual knowledge represents :

Spatial and working configuration of the scene. We decided to use maps of the scene. In this case, the scene is broken up into zones. We associate a zone with the field of view of each sensor and also with each blind area. The zones are characterised by a certain number of information :

- motion object areas located in the zone (lenght, intersection...)
- topology (unevenness..)
- rules of object movement (direction, priority)
- possible obstacles

These information are completed by the spatial relations knowledge existing between contiguous zones.

Class information for moving objects. Each module of vision tries to classify the observations. An observation is associated with a class of objects if it verifies a set of constraints. Those, on one hand, are imposed on the characteristic of each mobile object, e.g. independent of the scene, which are static (dimensions, size...) and also dynamic (speed, acceleration, possible behaviours...). On the other hand, the objects belonging to a class must verify geographical constraints related to the scene. This classification helps the tracking process first of all by reducing search areas and then by excluding abnormal situations.

Image acquisition information. The purpose of this information is to calibrate the data extracted from measurements. This information tries to transform measurements into invariant data irrespective of sightings and ambient illumination. This operation is essential for the visual recognition of an object seen by various sensors. It includes :

- camera characteristics (camera model, focal length)
- image characteristics (image type and size)
- sensor positioning (camera orientation, geolocation, ...).

Dynamic environment. It concerns the information relative to the global motion of all mobile objects located in the scene. This information can influence the object prediction. In traffic monitoring

area, it can be represented by the traffic density or congestion level. These information are either obtained on-line or can be generated by a predictive off line model. The last case can be used only when some situations appear periodically depending on specific schedules (time, day, weekend, holidays...).

3.2 DOP generation

A DOP is represented by means of a possibility distribution (π). This is a function from a reference set, here the time scale, to the real interval [0,1] which restricts the more or less possible values of a number, here a date. To synchronise the system each sensor is equiped which its own clock, all clocks having the same time base.

The width of the DOP support explains the imprecision of prediction associated to the mobile object. Indeed, this imprecision is strongly depending on the quality of knowledge resulting from the context and the dynamic characteristics of the tracked object.

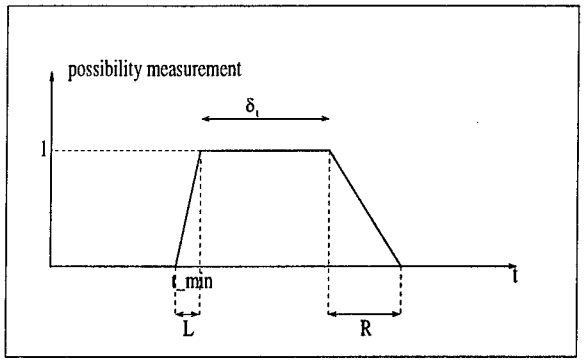


Figure 5: DOP's parameters

t_{min} represents the most optimistic time of arrival of object i in front of the sensor k from sensor j. This time depends on the maximum speed of the object by taking into account the spatial context between the two sensors. This speed is calculated thanks to the dynamic characteristics of the class of the object. This time does not take into account possible decelerations that can occur between the two sensors. The duration δ_t is based on possible decelerations related to spatial and dynamic contexts (traffic light, congestion...), as well as the constraints of object classes (acceleration variation).

In reality, as we work in an outdoor environment with many objects, and as we use a limited number of classes, these have to be roughly defined. (large / small vehicle, bicycle, human being, ...).

We know also that the various parameters related to the context are badly defined and numerous situations can't be envisaged. Under these conditions it seems more natural to build a DOP tolerating these incomplete information (c.f. figure 5). The slope L explains the approximation concerning the earliest date. It depends on the maximum speed variations within each object class. The slope R expresses all of the inaccuracies related to the spatial and dynamic contexts as well as the behaviour of the object class.

Once displacements prediction in blind zones are carried out, each sensor will try to match its observations with its awaited objects.

4 Matching

As soon as an observation is detected in front of a sensor, the latter tries to match it with its awaited objects. This operation breaks up into two stages :

- first, estimation of compatibilities between the observation and each awaited object using matching possibility measurements.
- then, the matching decision based on the knowledge of all the compatibilities.

4.1 Matching possibilities measurements

The matching process begins with the extraction of objects primitives (observations). The choice of discriminating primitives is important because objects matching is mainly based on them [7]. These primitives must be time invariant. A primitive extracted by a sensor must be logically found by another one.

After the primitives extraction, compatibilities measurements associated with each primitive are computed. They concern visual as well as temporal primitives.

Temporal compatibilities measurements take into account the observations date and the DOPs associated with awaited objects. The temporal compatibility measurement between an observation and an awaited object is equal to the value of the awaited object DOP when the observation appears in front of the sensor.

We show in figure 6 an example of temporal predictions carried out by the sensor C1. The first observation perceived in front of the sensor C2 is temporally compatible with the object O1 because its temporal compatibility measurement has a value equal to 1.

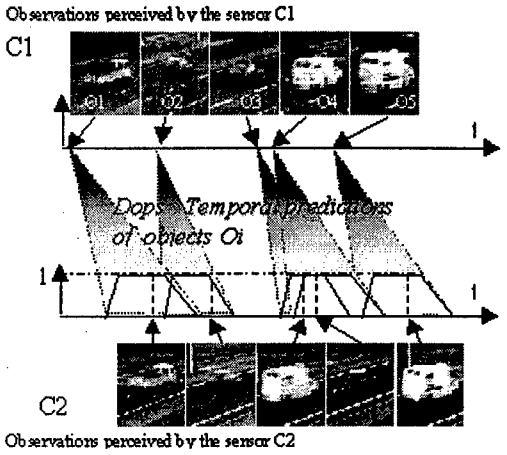


Figure 6: Temporal predictions appearance at sensor C2 of objects detected by sensor C1

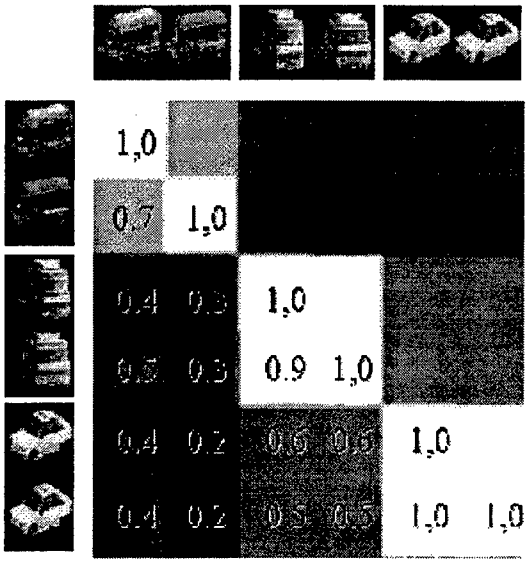


Figure 7: Visual compatibilities

Note : In figure 6, temporal information does not allow the discrimination of objects, concerning the object O4, two observations are temporally compatible. In this case, only visual compatibilities permit the matching.

Visual primitives employed are color, compactness and size. Figure 4.1 illustrates compatibilities between isolated objects using color histogram (test of the χ^2 between histograms). The test of the χ^2, presented below, is used to determine the similarity between the histograms of images I and H.

$$\chi^2 = \sum_j \frac{(h_j^I - h_j^H)^2}{(h_j^I - h_j^H)^2}$$

Once compatibilities measurements have been calculated, global degrees of compatibilities are computed between each perceived object and each awaited object. These degrees are the results of the combination of the various degrees of compatibility associated with the primitives. Each of these degrees is estimated using the distance existing between the values of the primitives of the perceived object and those of the awaited ones. Each degree takes a value ranging between 0 and 1, a value 1 means a total compatibility. The combination is based on a possibilist approach by taking into account the visual compatibilities and the temporal compatibility.

We have tested several possibilistic operators. Our choice was directed towards a type of operator supporting compatibility measurements favourising some specific situations, i.e. very strong compatibilities and incompatibilities (for example "X.Y"). Global compatibility degrees , noted P(Oi), between a perceived object and all the awaited objects Oi represent some possibility measurements of association [8].

4.2 Matching decision

The decision strategy for the matching between an observation and one possible awaited object exploits the global compatibilities degrees of the whole awaited objects. Two sets Ω and Ω' containing the candidates for the matching are built. Ω represents the set of awaited objects having a good compatibility with the observation. The subset Ω' extracted from Ω represents the set of dominant candidates (c.f. figure 8).

We present in figure 9 two decision matching examples. In the first example, the association possibility of the awaited object O1 is high, so O1 is stored in the set Ω. Moreover, as it presents a high necessity measurement (N= 0.65), O1 is also stored in the set Ω'. The necessity measurement expresses the uniqueness of the solution. The necessity measurement is important when the possibility measurement of the object is important with respect to the other objects. As Ω' contains only one object, then the association of this object can be realised with the observation. On the other hand in the second case, as two objects belong to Ω', matching

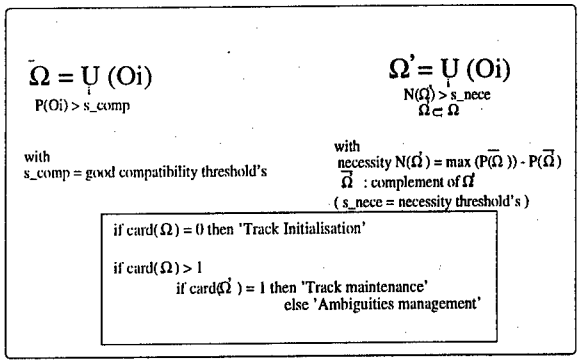


Figure 8: Matching algorithm

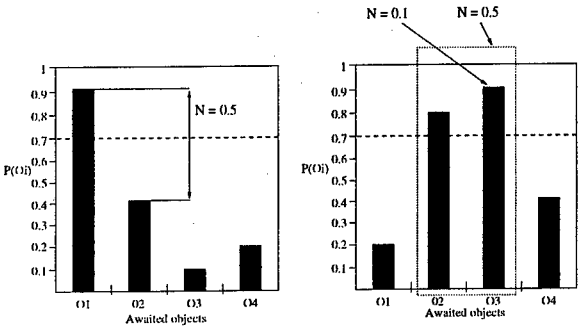


Figure 9: Two examples of decision matching

decision cannot be realised, decision is deferred.

The various decisions of our tracking system are:

4.2.1 initialisation of track

When there is a low compatibility between a perceived object and awaited objects, visual as well as temporal level, a new object is created and its track is initialised.

4.2.2 track maintenance

When only one awaited object is compatible with the perceived object, then there is no ambiguity. In this case, the object is well recognized and its track is maintained. This matching decision quality is considered by a necessity measurement expressing the certainty associated with the decision.

4.2.3 Ambiguous situations management

In some applications, when traffic is dense, many mobile objects can present visual and behavioral similarities. It is then necessary to manage this type of situation as well as possible. If a group of awaited objects presents a strong similarity degree, i.e. a high possibility measurement, then we decide, in order to avoid errors, to associate temporally the group Ω' with the observation. As a significant doubt exists between these objects, the individual necessity measurement of each candidate is low and the matching decision is deferred thanks to a multiple hypotheses tracking (MHT) approach. This MHT is limited in time according to the application configuration in order to control the propagation of the ambiguities in the system. When one object belonging to the group Ω' is accuratly perceived by one of the sensors, it can be removed from Ω'. The group's size reduction enables to reduce ambiguity of the matching decision.

4.2.4 Tracks termination

If an awaited object does not appear in a sensor observation zone after a given period, then thanks to the DOP associated with this object, which has its own lifespan, the track will be terminated. This termination operation is controlled by the sensor which has generated the associated DOP.

5 Conclusion

We have presented a tracking system in an extended scene by a geographically distributed multisensors approach. The interpretation system must be able to manage the objects tracking on the whole scene in presence of blind zones. The originality of our approach is based on the co-operation of autonomous vision modules. The communication is carried out using messages containing useful information for the global tracking. This latter is possible only if the system can match mobile objects by several sensors. The matching considered here is carried out by taking into account of temporal and visual compatibilities. Temporal compatibilities are controlled using DOP (Domain Occurrence Possibility) curves expressing the appearance possibility of an object in front of a given sensor. The matching between objects is carried out within a possibilistic framework. It is based on the calculation of global compatibility degrees between perceived objects and awaited one and followed by an estimation of necessity measurement taking into account of all the possible associations expressing matching quality.

References

[1] M. Rombaut and D. Meizel. Dynamic data temporal multisensor fusion in the prometheus prolab2 demonstrator. In *IEEE International Conference on Robotics and Automation*, page 3676, San Diego, 1994.

[2] D. Hutber, S. Moisan, C. Shekhar, and M. Thonnat. Perception interfacing via multisensor data fusion and multi-program supervision for the prolab2. In *The 7th IFAC/IFORS Symposium of Advance Technology*, number 1, pages 20–44, 1994.

[3] M.S.Y. Rao, H.F. Durrant-Whyte, and J.A. Sheen. A fully decentralized multi-sensor system for tracking and surveillance. *the international journal of robotics research*, 12(1):20–44, 1993.

[4] D Dubois and H. Prade. Processing fuzzy temporal knowledge. *IEEE transactions on Systems, man, and cybernetics*, 19:729–744, 1989.

[5] O. Wallart, C. Motamed, and M. Benjelloun. Temporal knowledge for cooperative distributed vision. In *seventh IEE Conference on Image Processing and its applications*, Manchester, 1999.

[6] F. Brémond and M. Thonnat. A context representation for surveillance systems. In *Workshop on Conceptual Descriptions from Images*, Cambridge, 1996. European Conference on Computer Vision (ECCV).

[7] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattren Recognition (CVPR94)*, Seattle, 1994.

[8] D. Dubois and H. Prade. *Théorie Des Possibilités, Applications À la Représentation Des Connaissances En Informatique.* Masson, Paris, 1988.

Searching Tracks *

J.-P. Le Cadre ,
IRISA/CNRS ,
Campus de Beaulieu, 35 042, Rennes, France
e-mail :lecadre@irisa.fr

Abstract

Search theory is the discipline which treats the problem of how best to find the optimal distribution of the total search effort which maximizes the probability of detection. Even if the general formalism of search theory will be of constant use subsequently, we shall consider now a radically different problem. In the "classical" search theory, the target is said detected if a detection occurs at any time of the time frame. Here, the target track will be said detected if elementary detections occur at various times. That means that there is a test for acceptation (or detection) of a target track and that the problem is to optimize the allocation of the search effort for track detection. Keywords: Search theory, optimization, duality, detection

1 Introduction

Search theory is the discipline which treats the problem of how best to search for an object when the amount of searching efforts is limited and only probabilities of the object's possible position are given. An important literature has been devoted to this subject, including surveys [2] and books [3], [4], [5]. The situation is characterized by three data: (i) the probabilities of the searched object (the "target") being in various possible locations; (ii) the local *detection probability* that a particular amount of local search effort should detect the target: (iii) the total amount of searching effort available. The problem is to find the *optimal* distribution of this total effort, i.e. which maximizes the probability of detection. Major steps in the development of search theory have been summarized in a prospective form by Stone [1].
However, even if the general formalism of search theory will be of constant use subsequently, we shall consider now a radically different problem. The problem is to detect target tracks. In the "classical" search theory, the target is said detected if a detection occurs at any time of the time frame. Here, the target track will be said detected if elementary detections occur at various times, and this is the *fundamental* difference. That means that there is a test for acceptation (or detection) of a target track. Track detection is also associated with a spatio-temporal modelling of the target track. Moreover, we shall not consider (in general) bounds relative to the search effort at each period. The bound is relative to the global search effort.
The paper is organized as follows. In section 2, the optimization framework is presented; followed by the general formulation of the search problem (see section 3). In section 4, we deal with the 2-period search problem, for the "AND" detection rule. Then, the optimization problems are detailed and solved, while they are extended to the n-period search in section 5. Another detection rule is considered in section 6, the "MAJORITY" detection rule. Section 7 is of a different nature since we consider here the general problem of search for Markovian tracks. The two-sided search problem is considered in section 8.

2 The optimization framework

The major part of this paper is centered around the following (primal) optimization problem :

$$\mathcal{P}\left\{\begin{array}{l}\min\ -P \quad \text{with : } P=\sum_\theta\ F\left(x_{1,\theta},x_{2,\theta},\cdots,x_{n,\theta}\right)\ ,\\ \text{where :}\\ F\left(x_{1,\theta},x_{2,\theta},\cdots,x_{n,\theta}\right)\triangleq f\left(p(x_{1,\theta})p(x_{2,\theta})\cdots p(x_{n,\theta})\right)\ ,\\ \text{under the resource constraints :}\\ \sum_\theta\ x_{1,\theta}+x_{2,\theta}\cdots+x_{n,\theta}=\Phi\ ,\\ x_{1,\theta}\geq 0\ ,x_{2,\theta}\geq 0\ ,\cdots,x_{n,\theta}\geq 0\ \ \forall(\theta)\ .\end{array}\right. \tag{2.1}$$

*This work has been supported by DCN/Ingénierie/Sud, (Dir. Const. Navales), France

In 2.1, $x_{k,\theta}$ represents a research effort, affected to the cell indexed by the parameter θ, at the search period k. The index k takes its values in the subset $\{1, \cdots, n\}$. The parameter θ takes its values in a multidimensional space, characterizing the target trajectory (e.g. initial position and velocity) and the n-dimensional vector $\mathbf{X}_\theta \triangleq (x_{1,\theta}, x_{2,\theta}, \cdots, x_{n,\theta})^*$ represents the effort vector associated with the target trajectory (or track) indexed by θ. Furthermore, $p(x_{k,\theta})$ is the elementary probability of detection in the cell (k, θ), for a search effort $x_{k,\theta}$; while f is a given differentiable function. The following simple remarks are then fundamental :

- the functional $F(x_{1,\theta}, \cdots, x_{n,\theta})$ is a differentiable functional of the variables $x_{k,\theta}$,
- the constraints are qualified since they are linear,
- the "hard constraint" is the equality constraint (i.e. $\sum_\theta x_{1,\theta} + x_{2,\theta} \cdots + x_{n,\theta} = \Phi$), the inequality constraints being *implicitely* taken into account.

A fundamental assumption is made in all the search theory literature : the detection functionals $F(x_{1,\theta}, x_{2,\theta}, \cdots, x_{n,\theta})$ are concave. In turn, the objective functional P is also concave. This assumption is central for proving the necessity of the "classical" optimality conditions for the search plan. Unfortunately, this assumption is not at all valid in our context.

These considerations lead us to consider and use basically the dual formalism. The following dual function is considered :

$$\begin{cases} \psi(\lambda) = \inf_{x_{1,\theta}, \cdots, x_{n,\theta}} \mathcal{L}(\lambda) \, , \\ \text{where :} \\ \mathcal{L}(\lambda) = -P + \lambda \left(\sum_\theta x_{1,\theta} + x_{2,\theta} \cdots + x_{n,\theta} - \Phi \right) . \end{cases} \tag{2.2}$$

We stress that, in our framework, the function $\psi(\lambda)$ may be explicitely determined on the subset defined by the inequality constraints. The dual problem $(\mathcal{D})$ then takes the following form :

$$\mathcal{D} \; : \; \max_\lambda \; \psi(\lambda) \, . \tag{2.3}$$

The decisive benefits of this approach are :

- the maximization of $\psi(\lambda)$ is an (unconstrained) *monodimensional* [1] problem,
- the function $\psi(\lambda)$ is differentiable,
- from the solution $\underline{\lambda}$ of the dual problem, the solution $\underline{\mathbf{X}}$ of the primal problem $\mathcal{P}$ is deduced (say $\underline{\mathbf{X}}(\underline{\lambda})$). The couple $(\underline{\lambda}, \underline{\mathbf{X}})$ is a saddle point of the primal-dual problem.

[1] In the case of a unique "hard" resource constraint

3 Modelling and formulation of the problem

In a large part of this article, we shall make the assumption that the target motion is rectilinear and uniform. So, in this case, the target trajectory is completely defined by its initial position vector (i) and a velocity vector (v), i.e. $\theta \equiv (i, v)$. Assumptions of our search problem are as follows :

- A target moves in a search space consisting of a finite number of search cells $C_t = \{c_{\theta,t}\}_\theta$ in discrete time $\mathbf{T} = \{1, 2, \cdots, n\}$. We further assume that the sequence of (searched) cells $\{c_{\theta,t}\}_t$ is completely defined by the parameter (θ) [6] (conditionally deterministic motion). Thus, the mapping $c_{\theta,1} \rightarrow c_{\theta,2} \cdots \rightarrow c_{\theta,n}$ is a bijection. In the simpler case (rectilinear motion of the target), this function mapping is simply a translation of vector v .
- The search effort applied to cell $c_{\theta,t}$ is denoted $x_{t,\theta}$ $(x_{t,\theta} \geq 0)$.
- The conditional probability of detecting the target given that the target is in the cell $c_{\theta,t}$ and that the search effort applied to this cell is $x_{t,\theta}$ is $p(x_{t,\theta})$. This probability is a classical exponential law, i.e. $p(x_{t,\theta}) = 1 - \exp(-w_{t,\theta}\, x_{t,\theta})$. The term $w_{t,\theta}$ stands for the particular conditions of detection (visibility) for the cell $c_{\theta,t}$.

4 The 2-period search for the "AND" track detection rule

First, we shall deal with the two period search problem (i.e. $n = 2$). More specifically, we shall say that the target track has been detected if the target has been detected at *each* (temporal) period of the search . We then have to solve the following search problem :

$$\mathcal{P} \begin{cases} \min \; -P \quad \text{where: } P = \sum_\theta g_1(\theta)\, p(x_{1,\theta})\, p(x_{2,\theta}) \, , \\ \text{under the constraints :} \\ \sum_\theta (x_{1,\theta} + x_{2,\theta}) = \Phi \quad , x_{1,\theta} \geq 0 \, , x_{2,\theta} \geq 0 \, , \; \forall(\theta) \, . \end{cases} \tag{4.4}$$

In the above equation $x_{1,\theta}$ (respectively $x_{2,\theta}$) denotes the search effort applied to the cell $c_{\theta,1}$ (respectively $c_{\theta,2}$). Then, we form the Lagrangian of

the primal problem 4.4, i.e. :

$$\begin{aligned}\mathcal{L}(\lambda) &= -\sum_{\theta} g_1(\theta)\,(1-e^{-w x_{1,\theta}})\,(1-e^{-w x_{2,\theta}})\ , \\ &\quad +\lambda\left(\sum_{\theta} x_{1,\theta}+\sum_{\theta} x_{2,\theta}-\Phi\right)\ , \\ &\quad -\sum_{\theta}\mu_{1,\theta}x_{1,\theta}-\sum_{\theta}\mu_{2,\theta}x_{2,\theta}\ , \\ &\quad \mu_{1,\theta}\geq 0,\quad \mu_{2,\theta}\geq 0\ .\end{aligned}$$

In order to apply the Karush-Kuhn-Tucker conditions of optimality (KKT for the sequel), we must consider two cases.

4.1 KKT optimality conditions and their consequences

case 1 $(x_{1,\theta}>0)$

In this case, the KKT condition $\{\mu_{1,\theta}\ x_{1,\theta}=0\}$ implies $\{\mu_{1,\theta}=0\}$. Then, the KKT stationarity condition (for the Lagrangian) simply results in :

$$\frac{\partial}{\partial x_{1,\theta}}\mathcal{L}(\lambda)=-w\,g_1(\theta)\,e^{-w\,x_{1,\theta}}\,(1-e^{-w\,x_{2,\theta}})+\lambda=0\ . \tag{4.5}$$

From 4.5, we note that the assumption $x_{1,\theta}>0$ implies $x_{2,\theta}>0$, otherwise the multiplier λ should be zero. Indeed, if $\lambda=0$ then it is easily seen (see 4.5) that the value of the dual function $\psi(\lambda)=\inf_{(x_{1,\theta},x_{2,\theta})}\mathcal{L}(\lambda)$ is $-\infty$. Since, we have to maximize $\psi(\lambda)$, we see that λ is necessarily *strictly* positive (see 4.5 for the sign). Thus, 4.5 implies the validity of the following equation :

$$\frac{\partial}{\partial x_{2,\theta}}\mathcal{L}(\lambda)=-w\,g_1(\theta)\,e^{-w\,x_{2,\theta}}\,(1-e^{-w\,x_{1,\theta}})+\lambda=0\ . \tag{4.6}$$

By collecting 4.5 and 4.6, and denoting $X_{1,\theta}=e^{-w\,x_{1,\theta}}$,$X_{2\theta}=e^{-w\,x_{2,\theta}}$, we obtain :

$$\begin{aligned}&X_{1,\theta}\,(1-X_{2,\theta})=X_{2,\theta}\,(1-X_{1,\theta})\ , \\ &\text{so, that :} \\ &X_{1,\theta}=X_{2,\theta}\quad \text{i.e. } x_{1,\theta}=x_{2,\theta}\ .\end{aligned} \tag{4.7}$$

The above equality is fundamental for solving the problem.

case 2 $(x_{1,\theta}=0)$

Assume now that $x_{2,\theta}>0$, then the KKT condition (relative to $x_{2,\theta}$) should imply (see 4.6, with $x_{1,\theta}=0$) :

$$\frac{\partial}{\partial x_{2,\theta}}\mathcal{L}(\lambda)=\lambda=0\ , \tag{4.8}$$

and, in turn, that the multiplier λ should be zero. Under this assumption, the value of $\psi(\lambda)$ is $-\infty$. Hence, we can restrict to the *strictly* positive values of λ, which means that the assumption $x_{1,\theta}=0$ implies $x_{2,\theta}=0$. Indeed, the hypothesis $x_{2,\theta}>0$ should imply the validity of 4.6 and, in turn, the multiplier λ should be zero since we assume the nullity of $x_{1,\theta}$, which contradicts the fact that λ is strictly positive.

4.2 Solving the dual problem

In conclusion, the following result has been stated : $x_{1,\theta}=x_{2,\theta}$. So that, we have now to deal with the following (simplified) optimization problem :

$$\mathcal{P}\begin{cases}\min\ -P\quad \text{where : } P=\sum_{\theta} g_1(\theta)\ (p(x_{1,\theta}))^2\ , \\ \text{under the constraints :} \\ \sum_{\theta} x_{1,\theta}=\Phi/2\quad ,x_{1,\theta}\geq 0\ ,\ \forall(\theta)\ .\end{cases} \tag{4.9}$$

Again, we examine the necessary conditions induced by the KKT theorem. Now, we consider the reduced Lagrangian functional $\mathcal{L}(\lambda)$ given by :

$$\mathcal{L}(\lambda)=-\sum_{\theta} g_1(\theta)\,(1-e^{-w x_{1,\theta}})^2+\lambda\left(2\sum_{\theta} x_{1,\theta}-\Phi\right)\ . \tag{4.10}$$

This form of the Lagrangian corresponds to the relaxation of the positivity constraints relative to the search variables $\{x_{1,\theta}\}$, which are *implicitely* taken into account by restricting our search to positive values of the variables $x_{1,\theta}$. Under the assumption that $x_{1,\theta}$ is *strictly* positive and differentiating $\mathcal{L}(\lambda)$ relatively to $x_{1,\theta}$, we then obtain :

$$\begin{aligned}&\frac{\partial\mathcal{L}(\lambda)}{\partial x_{1,\theta}}=-2\,w\,g_1(\theta)\,e^{-w\,x_{1,\theta}}\,(1-e^{-w\,x_{1,\theta}})+2\,\lambda=0\ , \\ &\text{or, equivalently :} \\ &X_{1,\theta}\,(1-X_{1,\theta})=\frac{\lambda}{w\,g_1(\theta)}\ .\end{aligned} \tag{4.11}$$

Equation 4.11 is a second order equation (in $X_{1,\theta}$), allowing us to determine $\underline{x}_{1,\theta}$, for a given value of λ . Note that we restrict to the roots (0 or 2) of 4.11 lying inside the interval $[0,1]$, and select the root (denoted $\underline{X}_{1,\theta}(\lambda)$) which minimizes the reduced Lagrangian functional $\mathcal{L}(\lambda)$ [2]).

[2]Note that we must test and compare the value of $\mathcal{L}(\lambda)$ not only for the roots of 4.11, but also for its lower bound (i.e. $X_{1,\theta}=1\Leftrightarrow x_{1,\theta}=0$

We have now to deal with the maximization of the dual functional defined by :

$$\begin{aligned}\psi(\lambda) &= -\sum_{(\theta)_+} g_1(\theta)\left(1-\underline{X}_{1,\theta}(\lambda)\right)^2 , \\ &\quad +\lambda\left(2\sum_{(\theta)_+}\underline{x}_{1,\theta}(\lambda)-\Phi\right) , \\ \underline{x}_{1,\theta}(\lambda) &= -\tfrac{1}{w}\ln\left(\underline{X}_{1,\theta}(\lambda)\right) \text{ if : } \underline{x}_{1,\theta}(\lambda)>0 ,\end{aligned} \tag{4.12}$$

where the symbol $(\theta)_+$ denotes the values of the index for which 4.11 has a root inside $[0,1]$. The maximization of $\psi(\lambda)$ is rather easy since it corresponds to an unidimensional search for a concave and differentiable function. In turn, the is no duality gap.

Notation 1 *The (spatio-temporal) index (θ,t) for which the research efforts are strictly positive are denoted $(\theta,t)_+$ (t : index of the search period); $(\theta)_+$ for the first search period.*

5 The n-period search for the "AND" track detection rule

Quite similarly to the 2-period search, we assume that the probability of detection of the track is the product of elementary detection probability of detection (i.e. at each period) and is thus given by [3] :

$$\begin{cases} P=\sum_\theta g_1(\theta)\,p(x_{1,\theta})\,p(x_{2,\theta})\cdots p(x_{n,\theta}) , \\ p(x_{k,\theta})=\gamma_k\left(1-e^{-w_{k,\theta}\,x_{k,\theta}}\right) \quad k=1,\cdots,n , \end{cases} \tag{5.13}$$

and the optimization problem is again :

$$\mathcal{P}\begin{cases} \min\ -P , \\ \text{under the constraints :} \\ \sum_\theta\left[x_{1,\theta}+\cdots+x_{n,\theta}\right]=\Phi , \\ x_{1,\theta}\geq 0,\cdots,x_{n,\theta}\geq 0 ,\ \forall(\theta) . \end{cases} \tag{5.14}$$

Assume $x_{1,\theta}\neq 0$, then by a reasoning strictly identical to the 2-period case, we deduce that $x_{2,\theta}\neq 0,\cdots,x_{n,\theta}\neq 0$. The optimality equations deduced from the KKT conditions then yield the following (non-linear) system of n equations :

$$\begin{cases} \gamma_1 X_{1,\theta}\left(1-\gamma_2 X_{2,\theta}\right)\cdots\left(1-\gamma_n X_{n,\theta}\right)=\frac{\lambda}{w_{1,\theta}\,g_1(\theta)}=\alpha_1 , \\ \gamma_2 X_{2,\theta}\left(1-\gamma_1 X_{1,\theta}\right)\cdots\left(1-\gamma_n X_{n,\theta}\right)=\frac{\lambda}{w_{2,\theta}\,g_1(\theta)}=\alpha_2 , \\ \vdots \\ \gamma_n X_{n,\theta}\left(1-\gamma_1 X_{1,\theta}\right)\cdots\left(1-\gamma_{n-1} X_{n-1,\theta}\right)=\frac{\lambda}{w_{n,\theta}\,g_1(\theta)}=\alpha_n \end{cases} \tag{5.15}$$

Consider now the above system, dividing row (1) by row (p) and denoting $Y_{1,\theta}\triangleq\gamma_1 X_{1,\theta},\cdots,Y_{p,\theta}\triangleq\gamma_p X_{p,\theta}$, we obtain :

$$\begin{aligned}&\frac{Y_{1,\theta}\left(1-Y_{p,\theta}\right)}{Y_{p,\theta}\left(1-Y_{1,\theta}\right)}=\frac{\alpha_1}{\alpha_p} , \\ &\text{i.e. : } Y_{p,\theta}=\frac{\alpha_p Y_{1,\theta}}{Y_{1,\theta}(\alpha_p-\alpha_1)+\alpha_1} .\end{aligned} \tag{5.16}$$

Consequently, $x_{p,\theta}$ is deduced from $x_{1,\theta}$, itself given by:

$$x_{1,\theta}=\frac{1}{w_{1,\theta}}\left[ln\left(\frac{\gamma_1}{Y_{1,\theta}}\right)\right]^+ .$$

The problem is thus reduced to the determination of $\underline{x}_{1,\theta}$. From 5.16 we have $1-X_{p,\theta}=\left[\alpha_1\left(1-X_{1,\theta}\right)\right]/\left(X_{1,\theta}(\alpha_p-\alpha_1)+\alpha_1\right)$. Inserting this equality in 5.15, we see that $\underline{X}_{1,\theta}$ is a root of the following n-th order polynomial equation :

$$\alpha_1{}^{n-2}\,X_{1,\theta}\left(1-X_{1,\theta}\right)^{n-1}-\prod_{i=2}^{n}\left(X_{1,\theta}\left(\alpha_i-\alpha_1\right)+\alpha_1\right)=0 . \tag{5.17}$$

The value of $\underline{X}_{1,\theta}(\lambda)$ is the root of 5.17 which minimizes the Lagrangian, deduced from 5.13; where $\underline{x}_{2,\theta},\cdots,\underline{x}_{n,\theta}$ are determined (from $\underline{x}_{1,\theta}$) by 5.16. The computation load is relatively modest. From $\underline{x}_{1,\theta}$, the dual function $\psi(\lambda)$ is deduced, i.e. :

$$\psi(\lambda)=-\sum_{(\theta)_+}\prod_{k_+}\gamma_k\left(1-X_{k,\theta}\right)+\lambda\left(\sum_{(\theta,k)_+}x_{k,\theta}-\Phi\right) . \tag{5.18}$$

The problem is simply to determine the value of λ which maximizes the concave function $\psi(\lambda)$.

So far, the problem has been considered in its full generality. To illustrate the previous calculations, assume now that the visibility coefficients $\{w_{1,\theta},\cdots,w_{n,\theta}\}$ are equal altogether, i.e. $:p(x_{k,\theta})=\gamma\left(1-e^{-w\,x_{k,\theta}}\right)\quad k=1,\cdots,n$ then the optimality equations 5.15 and 5.16 simply reduce to $Y_{1,\theta}=\cdots=Y_{n,\theta}$, so that $X_{1,\theta}=\cdots=X_{n,\theta}$ and the probability of track detection as well as the dual function $\psi(\lambda)$ become :

$$\begin{cases} P &= \sum_\theta g_1(\theta)\left[\gamma\left(1-e^{-w\,x_{k,\theta}}\right)\right]^n . \\ \psi(\lambda) &= -\sum_{(\theta)_+} g_1(\theta)\left[\gamma\left(1-\underline{X}_{1,\theta}(\lambda)\right)\right]^n \\ &\quad +\lambda\left(n\sum_{(\theta)_+}\underline{x}_{1,\theta}(\lambda)-\Phi\right) . \end{cases} \tag{5.19}$$

Again, we have to deal now with a simple monodimensional optimization problem, involving a concave functional.

[3] The scalar $w_{k,\theta}$ stands for the possibly changing visibility conditions from one period to another one

Let us denote $\Phi(\lambda)$ the optimal value of the (total) search effort for a given λ; then the following result holds :

Proposition 1 *$\Phi(\lambda)$ is a decreasing function of λ .*

Proof : Denoting θ, the track parameter, the Lagrangian $\mathcal{L}(\lambda)$ of the constrained problem is $\mathcal{L}(\lambda,\theta) = -P + \lambda(\sum_{i=1}^{n} x_{i,\theta} - \Phi)$ ($P = \sum_\theta g_1(\theta) p(x_{1,\theta}) \cdots p(x_{n,\theta})$); so, that : $\frac{\partial \mathcal{L}(\lambda)}{\partial x_{i,\theta}} = -\frac{\partial P}{\partial x_{i,\theta}} + \lambda$, and consequently :

$$\lambda_2 > \lambda_1 \Rightarrow \frac{\partial \mathcal{L}(\lambda_2)}{\partial x_{i,\theta}} \geq \frac{\partial \mathcal{L}(\lambda_2)}{\partial x_{i,\theta}} , \qquad (5.20)$$

hence $\underline{x}_{i,\theta}(\lambda_1) \geq \underline{x}_{i,\theta}(\lambda_2)$ $(\forall i,\ \theta)$; and in turn $\Phi(\lambda_2) \leq \Phi(\lambda_1)$.

6 The "MAJORITY" rule for track detection :

Up to now, our analysis has been restricted to an "AND" rule for track detection. For numerous applications, a MAJORITY rule is also quite realistic. This means that a track is said detected if a "sufficient" number of elementary detections occur "along" the track. We have now to face specific problems. First, it is difficult to give a general formulation (for the general n-period search) of the detection rule. Second, the optimization problems become far more intricated.

6.1 The 3-period case and the "MAJORITY" track detection rule

The detection function is modified in order to take into account a majority rule ("MAJORITY") for decision [7]. More precisely, the track is said to be detected if the target is detected *at least* at 2 periods. With this rule, the probability of detection becomes :

$$\begin{aligned} P \;=\; & \sum_\theta g_1(\theta)\,[\beta_{0,2,3}\,P_{0,2,3}\,, \qquad (6.21)\\ & +\;\; \beta_{1,2,0}\,P_{1,2,0} + \beta_{1,0,3}\,P_{1,0,3} + \beta_{1,2,3}\,P_{1,2,3}] \,. \end{aligned}$$

In 6.21, the notation $P_{0,2,3}$ corresponds to the following hypothesis: no detection at period 1, detection at periods 2 and 3, idem for $P_{1,2,0}$ and $P_{1,0,3}$. The notation $P_{1,2,3}$ corresponds to a detection at each period. Finally, the weights $\beta_{0,2,3}, \cdots, \beta_{1,2,3}$ are related to the information "gain" associated with an elementary event. This gain may be expressed in terms of quality of the estimated track, probability of correct association, etc. Thus, the elementary detection terms $P_{0,2,3}\,, \cdots, P_{1,2,3}$ have the following form :

$$\left\{ \begin{array}{ll} P_{0,2,3} = & e^{-w\,x_{1,\theta}}\,(1-e^{-w\,x_{2,\theta}})\,(1-e^{-w\,x_{3,\theta}})\,,\\ P_{1,2,0} = & e^{-w\,x_{3,\theta}}\,(1-e^{-w\,x_{1,\theta}})\,(1-e^{-w\,x_{2,\theta}})\,,\\ P_{1,0,3} = & e^{-w\,x_{2,\theta}}\,(1-e^{-w\,x_{1,\theta}})\,(1-e^{-w\,x_{3,\theta}})\,,\\ P_{1,2,3} = & (1-e^{-w\,x_{1,\theta}})\,(1-e^{-w\,x_{2,\theta}})\,(1-e^{-w\,x_{3,\theta}})\,. \end{array} \right. \qquad (6.22)$$

Defining the reduced Lagrangian as $\mathcal{L}(\lambda) = -P + \lambda\,(\sum_\theta\,(x_{1,\theta} + x_{2,\theta} + x_{3,\theta}) - \Phi)$, we adopt the following notations for the sake of simplicity [4] :

$$\left\{ \begin{array}{l} \beta_{0,2,3} \equiv \delta_1,\ \beta_{1,2,0} \equiv \delta_3,\ \beta_{1,0,3} \equiv \delta_2,\ \beta_{1,2,3} \equiv \delta^*\,,\\ X_{1,\theta} = e^{-w\,x_{1,\theta}} \equiv y_1, \cdots, X_{3,\theta} = e^{-w\,x_{3,\theta}} \equiv y_3\,. \end{array} \right. \qquad (6.23)$$

Assuming that y_1, y_2, y_3 differ altogether from 1, the KKT conditions yield :

$$y_3 = \left[\frac{y_1\,(\delta^* - \delta_1 - \delta_2) + \delta_2 - \delta^*}{y_1\,(\delta^* - \delta_1 - \delta_3) + \delta_3 - \delta^*}\right]\,y_2\,. \qquad (6.24)$$

Then, inserting $y_3 = f(y_1)\,y_2$ (see 6.24) in 6.22, we obtain the following 2-th order equation :

$$(a - b\,y_1)\;y_2^2 + \left(c - d\,y_1^2\right)\;y_2 + \left(e\,y_1^2 + f\,y_1\right) = 0\,,$$

where the coefficients (a, b, c, d) are easily calculated. In this case ($x_{k,\theta} \neq 0$; $k = 1, 2, 3$), the distribution of the search efforts is completely determined by the optimality equations. For instance, from 6.24 we obtain $y_3 = f(y_1)\,y_2$ and $y_2 = f'(y_1)$. The optimal value of y_1 is thus the value of y_1, solution to the non-linear equation in y_1 deduced from 6.24 by replacing y_2 and y_3 by their expressions in terms of y_1 only; $\underline{y}_1$ is then the root of this non-linear equation which minimizes the Lagrangian.

Also from the optimality equations, we see that the nullity of the search effort at two periods (i.e. $y_k = y_{k'} = 1$ for $k \neq k'$) results in the nullity of the total search effort (i.e. $y_1 = y_2 = y_3 = 1$). So, we must consider the cases where the search effort is null at one period. In this case, only

[4] The index of missed detection is the index of δ

two optimality equations are valid. Consider for instance (other cases are completely similar), the case $x_{2,\theta} = 0$, then we obtain $\delta_2\, y_1\,(1-y_3) = \frac{\lambda}{g_1}$.

6.2 The n-period search and the "MAJORITY" track detection rule

We shall now restrict to the following track detection rule. The track is said detected if, at least, $(n-1)$ elementary detections occur (for a n-period search). Thus, the probabilities of the following events are considered :

$$\left\{\begin{array}{lll} P_1 & \equiv P_{0,2,\cdots,n} & = y_1 \prod_{i=2}^{n}(1-y_i)\,, \\ P_2 & \equiv P_{1,0,2,\cdots,n} & = y_1 \prod_{i=1,\neq 2}^{n}(1-y_i)\,, \\ & \vdots & \\ P_n & \equiv P_{1,2,\cdots,n-1,0} & = y_n \prod_{i=1}^{n-1}(1-y_i)\,, \\ P_* & \equiv P_{1,2,\cdots,n} & = \prod_{i=1}^{n-1}(1-y_i)\,. \end{array}\right. \tag{6.25}$$

For the sake of simplicity, the following assumptions are made: the (detection) coefficients (i.e. $\beta_{0,2,\cdots,n}$, $\beta_{1,0,2,\cdots,n}, \cdots, \beta_{1,2,\cdots,n}$, see 6.23) are equal [5]. Let us first assume that the search efforts are non-zero for all the periods (i.e. : $x_1 \neq 0, \cdots, x_n \neq 0$), then the KKT conditions result in :

$$(y_2 - y_3) \prod_{i\neq 2,3}^{n} \left[\frac{y_1}{(1-y_1)} + \frac{y_4}{(1-y_4)} + \cdots + \frac{y_n}{(1-y_n)}\right] = 0\,. \tag{6.26}$$

Since the term between brackets is well defined and non-zero, we deduce from 6.26 that $y_2 = y_3$, and more generally considering the difference equations obtained by substracting row $(i+1)$ to row i in the optimality equations , we have $y_1 = y_2 = \cdots = y_n$. Moreover, we can prove that the search efforts (for a given track parameter $\{\theta\}$) is either zero for all the periods or zero for *at most* one period. The rest of the derivation is identical to the 3-period case.

7 Search for Markovian tracks :

We consider now search for a markovian target. The classical optimization framework we used previously is here useless, due to the complexity of elementary events. Instead, we shall use the Brown's approach [8], where a sequence of search plans is incrementally generated. For the sake of simplicity, our approach is restricted to the AND detection rule.

The target is moving among a finite number of cells. Let the set of cells be C (at each time period). The target occupies one cell during each of the time periods so its path is decribed by $\omega = (\omega_1, \omega_2, \cdots, \omega_n) \in C^n$ so that the target be detected (for the AND detection detection rule) is :

$$P = \sum_{\omega\in\Omega} g(\omega) \prod_{i=1}^{n} \left[1 - \exp\left(-w(\omega_i, i)\, x(\omega_i, i)\right)\right]\,. \tag{7.27}$$

Thus, we have to deal with the following problem :

$$\mathcal{P}\left\{\begin{array}{l} \min\ -P \quad \text{where: } P \text{ is given by 7.27}\ , \\ \text{under the constraints :} \\ x(c_i,i) \geq 0 \ \text{ and : } \sum_{c_i\in C_i} x(c_i,i) \leq L_i\,. \end{array}\right. \tag{7.28}$$

Sufficient conditions may be derived from the results of Stone (see [3]). However, a direct solution to the optimality conditions seems quite unfeasible. It is then worthy considering the following factorization of $P(\mathbf{X})$ ($\mathbf{X}$: search plan) :

$$\begin{array}{l} P(\mathbf{X}) = \sum_{c,i} P(c,i,\mathbf{X})\ \left[1 - \exp\left(-w(c,i)\, x(c,i)\right)\right]\,, \\ \text{where :} \\ P(c,i,\mathbf{X}) = \sum_{\omega\in\Omega;\, \omega_i=(c,i)}\ g(\omega) \prod_{j\neq i} \left[1 - \exp\left(-w(\omega_j, j)\, x(\omega_j, j)\right)\right. \end{array} \tag{7.29}$$

The problem is thus immersed in a stationary framework, in which $P(c,i,\mathbf{X})$ represents the probability that the search has been successful at all the periods different from i, for all the target paths passing by the cell (c,i) at the period i. This corresponds to the reallocation problem [8]. So, the main problem then consists in effectively calculating $P(c,i,\mathbf{X})$. For that aim, we consider the following recursion, rather similar in its spirit, to the Brown's one [8] :

$$\begin{array}{rcl} \text{reach}(c,i,\mathbf{X}) & = & \sum r(\omega_1)\, t(\omega_1,\omega_2)\cdots t(\omega_{i-1},c)\,, \\ & & \times \prod_{j=1}^{i-1} \left[1 - \exp\left(-w(\omega_j,j)\, x(\omega_j,j)\right)\right]\,, \\ \text{surv}(c,i,\mathbf{X}) & = & \sum t(c,\omega_{i+1})\cdots t(\omega_{n-1},\omega_n)\ s(\omega_n)\,, \\ & & \times \prod_{j=i+1}^{n} \left[1 - \exp\left(-w(\omega_j,j)\, x(\omega_j,j)\right)\right]\,, \\ P(c,i,\mathbf{X}) & = & \text{reach}(c,i,\mathbf{X})\ \text{surv}(c,i,\mathbf{X})\,. \end{array} \tag{7.30}$$

Previously, Ω was small enough to practically enumerate its elements (e.g. *conditionallly deterministic motion*). Here, this is not feasible

[5] As seen previously (see section 6.1), this assumption does not reduce the generality of our approach.

since we must consider all the (markovian) paths $\omega = (\omega_1, \omega_2, \cdots, \omega_n)$. The terms reach$(c, i, \mathbf{X})$ and surv$(c, i, \mathbf{X})$ are themselves determined by the Brown recursion [8].

8 Two-sided search

Up to now, our efforts have been exclusively devoted to the one-sided search, which means that decisions are only made by the searcher. For the two-sided search, game theory is generally used. Here, the following game is considered : the strategy for player 1 (target) is a distribution of $g_1(\theta)$, while for player 2 (searcher) it is the distribution of efforts (i.e. $\mathbf{X}_\theta$)[6]. Let us denote $\mathbf{G}_1$ the vector representing the distribution of $g_1(\theta)$, we are now considering the following problem :

Determine the vectors $\mathbf{G}_1^*$, $\mathbf{X}^*$ such that :
$$P(\mathbf{G}_1^*, \mathbf{X}) \leq P(\mathbf{G}_1^*, \mathbf{X}^*) \leq P(\mathbf{G}_1, \mathbf{X}^*) \ , \ \forall \ (\mathbf{X}, \mathbf{G}_1) \ . \tag{8.31}$$

The detection function $P(\mathbf{G}_1, \mathbf{X})$ being given by 2.1. Equivalently $\mathbf{X}^*$ and $\mathbf{G}_1^*$ are the solutions to the min-max problem $\max_{\mathbf{X}} \min_{\mathbf{G}_1} P$. Restricting, first, to the AND detection test, we have thus to deal with the following optimization problem :

$$\mathcal{P} \left\{ \begin{array}{l} \min_{\mathbf{G}_1} \min_{\mathbf{X}} \{ - \sum_\theta g_1(\theta) \prod_{i=1}^n \gamma_k \ (1 - e^{-w_{k,\theta}\, x_{k,\theta}}) \} \\ \text{under the constraints :} \\ \sum_k \sum_\theta x_{k,\theta} = \Phi \ , \ \{ x_{k,\theta} \geq 0 \ , \forall \ (k, \theta) \} \\ \sum_\theta g_1(\theta) = 1 \ . \end{array} \right. \tag{8.32}$$

If, furthermore, the following assumption is made (γ_k=cst, $w_{k,\theta} = w_\theta$); then the above problem may be explicitely solved.

Proposition 2 *The elements of* $\mathbf{G}_1^*$ *and* $\mathbf{X}_1^*$ *are determined by the following equation:*

$$\begin{array}{l} x_{1,\theta} = \cdots = x_{n,\theta} \ (\forall \theta \in \Theta) \ , \ \textit{and} \ \forall \theta \in \Theta \ , \\ x_{1,\theta} = \frac{\Phi}{n} \left(\sum_{\theta \in \Theta} w_\theta^{-1} \right)^{-1} \textit{and} \ g_1(\theta) = \left(\sum_{\theta \in \Theta} w_\theta^{-1} \right)^{-1} w_\theta^{-1} \end{array} \tag{8.33}$$

Proof Let us consider the reduced Lagrangian defined by :

$$\mathcal{L}(\lambda, \mu) = -P + \lambda \left(\sum_k \sum_\theta x_{k,\theta} - \Phi \right) + \mu \left(\sum_\theta g_1(\theta) - 1 \right) , \tag{8.34}$$

then the KKT conditions $\frac{\partial \mathcal{L}}{\partial x_{k,\theta}} = 0$ yield $x_{1,\theta} = \cdots = x_{n,\theta}$ $(\forall \theta)$. Furthermore, the condition $\frac{\partial \mathcal{L}}{\partial g_1(\theta)} = 0$ implies :

$$\mu = (1 - X_{1,\theta})^n \ \rightarrow w_\theta \, x_{1,\theta} = \text{cst} \ , \ \forall \theta \in \Theta_+ [7] \ . \tag{8.35}$$

The constant itself is determined by the constraint $\sum_\theta x_{1,\theta} = \Phi / n$, yielding :

$$x_{1,\theta} = \frac{\Phi}{n} \left(\sum_{\theta \in \Theta_+} w_\theta^{-1} \right)^{-1} w_\theta^{-1} \ , \ \forall \theta \in \Theta_+ . \tag{8.36}$$

Then from $\frac{\partial \mathcal{L}}{\partial x_{1,\theta}} = 0$, we deduce :

$$\begin{array}{l} -w_\theta \, X_{1,\theta} \, (1 - X_{1,\theta})^{n-1} + \lambda = 0 \ , \text{so, that :} \\ g_1(\theta) = \text{cst} \ w_\theta^{-1} \left(X_{1,\theta}^{-1} - 1 \right) \ , \ \theta \in \Theta_+ . \end{array} \tag{8.37}$$

The constant is determined by the constraint $\sum_\theta g_1(\theta) = 1$ and the above expression (8.36) of $x_{1,\theta}$, yielding :

$$g_1(\theta) = \left(\sum_{\theta \in \Theta_+} w_\theta^{-1} \right)^{-1} w_\theta^{-1} \ , \ \forall \theta \in \Theta_+ . \tag{8.38}$$

Consider now $\mathbf{G}_1^*$, the vector formed with $g_1(\theta) = \left(\sum_{\theta \in \Theta} w_\theta^{-1} \right)^{-1} w_\theta^{-1}$ (i.e. θ in the *whole* set Θ), as well as $\mathbf{X}_1^*$ the vector with components $x_{1,\theta} = \frac{\Phi}{n} \left(\sum_{\theta \in \Theta} w_\theta^{-1} \right)^{-1}$. Then the value of the elementary detection term (i.e. $(1 - e^{-w_\theta \, x_{1,\theta}^*})$) is independent of θ, hence $P(\mathbf{G}_1, \mathbf{X}^*) = P(\mathbf{G}_1^*, \mathbf{X}^*)$ $(\forall \ \mathbf{G}_1)$. The second inequality is a consequence of the two following inequalities, themselves resulting from KKT conditions :

$$\left\{ \begin{array}{l} g_1^*(\theta) \, w_{1,\theta} \, X_{1,\theta}^* \left(1 - X_{1,\theta}^* \right)^{n-1} \leq \lambda \ , (1) \ , \\ \sum_{\theta \in \Theta} x_{1,\theta}^* \left[\lambda - g_1^*(\theta) \, w_{1,\theta} \, X_{1,\theta}^* \left(1 - X_{1,\theta}^* \right)^{n-1} \right] = 0 \ (\end{array} \right. \tag{8.39}$$

Now, there exists at least one value of θ (say θ_0) such that x_{1,θ_0}^* be strictly positive. From 8.39,(2), we deduce that $\lambda = g_1^*(\theta_0) \, w_{1,\theta_0} \, X_{1,\theta_0}^* \left(1 - X_{1,\theta_0}^* \right)^{n-1}$. Let c' be the constant term $w_{1,\theta} \ g_1^*(\theta)$; then λ is strictly inferior to c' $(X_{1,\theta}^* < 1)$ and, in turn, 8.39,1 yields :

$$c' \, X_{1,\theta}^* \left(1 - X_{1,\theta}^* \right)^{n-1} < c' \ . \tag{8.40}$$

[6] For the notations, we refer to section 2.

[7] Θ_+ is the subset of Θ corresponding to $g_1(\theta) > 0$.

Therefore $x^*_{1,\theta}$ is strictly positive, whatever θ, and $\Theta = \Theta_+$.

Thus, we note that, under the above assumption, the two-sided search problem has an explicit and remarkably simple solution (see [9]). Furthermore, remark that the optimal searcher and target strategies are proportional. Quite intuitively, this strategy is such that the product $w_\theta g_1(\theta)$ remains constant. In the general case (i.e. $p(x_{k,\theta}) = \gamma_k \left(1 - e^{-w_{k,\theta}\, x_{k,\theta}}\right)$), a direct resolution to the primal problem 8.32 is unfeasible; however the problem may be easily solved by the dual approach. More precisely, eqs 5.15, 5.16 are still valid.

9 Simulation results

For the sake of brevity, complete results may be found elsewhere and we shall restrict here to the optimization procedure which is illustrated by fig. 1 (AND, 10 periods). Notice that $\Phi(\lambda)$ is a decreasing function of λ, and that the maximum value of $\psi(\lambda)$ corresponds to the value of the total search effort (i.e. $\Phi(\underline{\lambda}) = 1000$). This result is important since it proves that there is no duality gap. Finally, the probability of detection P is plotted on the bottom subfigure. Again, it is a monotonic (decreasing) function of λ.

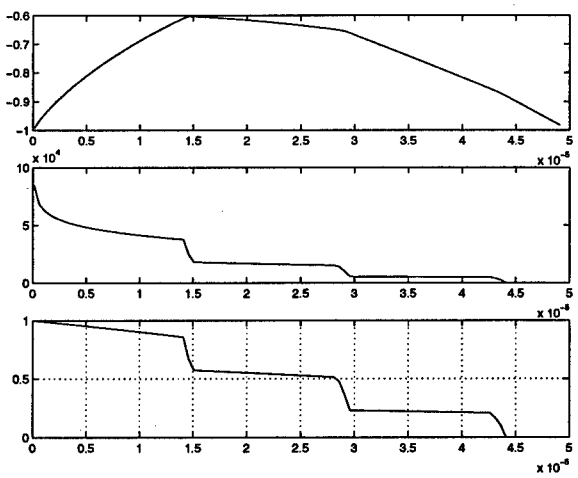


Figure 1: Top subfigure : values of the dual function $\psi(\lambda)$, versus λ. Middle subfigure : values of the total search effort $\Phi(\lambda)$, versus λ. Bottom subfigure : probability of detection, versus λ.

10 Conclusion

The problem under consideration was the optimization of the search effort for detecting tracks. The problem formulation is tightly related to the definition of the track detection criterion. Various definitions have been considered (namely AND and MAJORITY), as well as the corresponding optimization problem. In order to develop feasible methods, we focused on discrete (both in time and space) optimization . Under simple constraints (relative to the distribution of the search effort), the dual formalism appears as a feasible and versatile approach.

References

[1] L.D. STONE, What's Happened in Search Theory Since the 1975 Lanchester Prize?. OPERATIONS RESEARCH, vol.–37, no 3, May–June 1989, pp. 501–506.

[2] S.J. BENKOVSKI, M.G. MONTICINO and J.R. WEISINGER, A Survey of the Search Theory Literature. NAVAL RESEARCH LOGISTICS, vol.–38, pp. 469–491, 1991.

[3] L.D. STONE, *Theory of Optimal Search*, 2-nd ed. . Operations Research Society of America, Arlington, VA, 1989.

[4] K. IIDA, *Studies on the Optimal Search Plan.* Lecture Notes in Statistics, vol. 70, Springer-Verlag, 1992.

[5] A.R. WASHBURN, *Search and Detection*, 2-nd ed. . Operations Research Society of America, Arlington, VA, 1989.

[6] L.D. STONE and H.R. RICHARDSON, Search for Targets with Conditionally Deterministic Motion. SIAM J. Appl. Math. , vol.–27, no.–2, pp. 239–255, September 1974.

[7] P. K. VARSHNEY, *Distributed Detection and Data Fusion.* Springer-Verlag, 1997.

[8] S.S. BROWN, Optimal Search for a Moving Target in Discrete Time and Space. OPERATIONS RESEARCH, vol.–28, no 6, Nov.–December 1980, pp. 1275–1289.

[9] T. NAKAI, Search Models with Continuous Effort Under Various Criteria. Journal of the Operations Research Society of Japan, vol.–31, no.–3, pp. 335–351, Sept. 1988.

Passive Sonar Fusion for Submarine C^2 Systems

Pailon Shar* **and** **X. Rong Li†**

Department of Electrical Engineering
University of New Orleans
New Orleans, LA 70148, USA
Phone: (504)280-7416, Fax: (504)280-3950, xli@uno.edu

Abstract-The most important sensors for gathering target information onboard a submarine are passive sonars. Problems concerning fusion of these passive sonars are discussed. Three typical passive sonars - passive noise sonar, passive ranging sonar and acoustic pulse surveillance sonar – are supposed to constitute a passive sonar system for data fusion. This paper is concerned mainly with problems of significance in system development, such as tactical application background, special fusion techniques and own-ship maneuver considerations.

Key Words: fusion, sonar, submarine, sensor, command and control.

1. Introduction

For tactic reasons, passive sonars are considered to be the most important sensors onboard a modern submarine, for which stealth is vital. Basic submarine underwater operations, such as surveillance, search, detection and tracking, are usually guided by passive sonars. Almost all of modern passive sonars are capable of processing multiple targets. They can detect, sort, track, record and display many targets simultaneously. When several such passive sonars are introduced on the same platform to form a multisensor system, fusion techniques are needed to handle this multisensor multitarget problem. This is the task of a unit known as fusion center, which is part of the command and control (C^2) system. Fusion center receives and processes the multitarget information from the sensors. The information received is usually in large amount, of miscellaneous type, inaccurate and could even be misleading. The output of the center is more concise, more accurate and more meaningful tactically.

A modern submarine is usually equipped with many other sensors, e.g. radars and ESM, in addition to passive sonars. Fusion center should handle all these sensors, not just passive sonars. For the fusion system to be effective, it is important to coordinate the passive sonars and the other sensors. In reference [1], a fusion framework of a hierarchic structure for all the submarine sensors is proposed. It is suitable for systems with special groups of sensors that need to be handled relatively independently. Because of the importance of passive sonars and the similarity of their information, they can be treated as a group. Fusion may be conducted among themselves at first, then with other sensors or groups. This structure, among other things, makes submarine sensor fusion unique. In this framework, it is evident that the passive sonar fusion system, which is the major topic of this paper, is one subsystem of the entire sensor fusion system.

Meanwhile, special requirements and problems arise from the overwhelming importance of passive information and the passive property of the information itself, and need to be satisfied or treated specially when such a passive fusion subsystem is developed. These specialties are exactly what interest us in this paper.

Suppose the passive sonar system is composed of three typical passive sonars onboard submarines: passive noise sonar, passive ranging sonar and acoustic pulse surveillance sonar. Information collected by these sensors can basically be classified into two categories: positional information and characteristic information. Positional information reflects target position and motion, such as bearing, distance, course and velocity. Characteristic information includes target type and identity. The techniques to process them are quite different. This is concerned mainly with the former type of information.

2. Passive Sonar Systems and Tactical Background

There are plenty of common techniques, devices, software and systems that can be used to develop military systems. Adjustments have to be made, how-

* Also named Peilun Xia, visiting scholar, on leave from Ocean University of Qingdao, P. R. China.

† Supported by ONR via Grant N00014-97-1-0570, NSF via Grant ECS-9734285, and LEQSF via Grant (1996-99)-RD-A-32.

ever, due to the special requirements of a particular system. These requirements are usually put forward by the system itself and the tactical environment to which the system is supposed to be exposed. Meeting these requirements is a basic prerequisite of system development. In fact, the importance of understanding the sensor system itself and its application background, especially the tactical background, can never be overemphasized. System developers should bear this in mind in the entire process of system development.

2.1 Passive Sonars

Passive noise sonar is the fundamental sensor of a submarine. It serves both as search sensor and as attack sensor. For positional information, noise sonar provides the angle-of-arrival (azimuth angle, or bearing) measurement of an acoustic source. This bearing information is the basic information source for a submarine. Needless to say, a comprehensive modern passive noise sonar can provide much more information than bearing. The accuracy of bearing measurements is relatively good. Under some disadvantageous conditions, however, such as in shallow water, high water temperature, complex sea current, other sudden changes in the underwater acoustic transmission media, the measurement error can grow significantly.

The inability of the noise sonar to provide distance information is compensated by passive ranging sonar. Ranging sonar has three or four groups of hydrophone symmetrically mounted on both flanks of the submarine. It provides passively both bearing and distance information by processing the time-of-arrival differences between the hydrophone groups. The problem is that the range measurement error is usually large, especially at the beginning of detection, and it is also geometrically correlated. Target distance and the relative bearing of the target to the submarine have a significant impact on the ranging error. The larger the distance, the larger the error. In addition, the error is the smallest when the target is on the beam of the submarine. The farther away the target is from the beam, the large the error. The ranging error sometimes is so large that the detected distance information cannot be directly used for fire control purposes.

Acoustic pulse surveillance sonar intercepts acoustic transmissions from active sonars. It can provide bearing information of the detected pulses. Other information such as frequency, pulse length and pulse repetition frequency, is also available. The bearing measurement error is much larger than (usually several times of) its counterpart of the other two sonars. That is why its positional information plays a minor role in the fusion system.

The detection regions of the three sonars are quite different. Acoustic pulse surveillance sonar is omnidirectional. Its detection range is the largest of the three. Passive noise sonar usually has a sector of blind zone around the stern of the submarine, because its array is usually located in the bow sonar dome. Its detection range is smaller than that of the acoustic pulse surveillance sonar, but larger than that of the passive ranging sonar. Passive ranging sonar has two sector blind zones around the stern and the bow, respectively. Its detection range is the smallest. Fig. 1 illustrates the detection zones of these sonars.

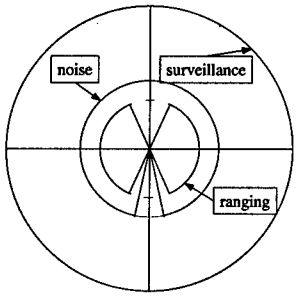


Figure 1. Detection zones of the sonars

Generally speaking, the ability of all these sonars to resolve or distinguish multiple targets is much weaker than their radar counterparts. This is also mainly due to the disadvantageous physical media. And the resolution is seriously affected by factors such as environment, geometry and signal intensity, other than sonar's own physical properties. All these factors should be considered and treated properly when the fusion system is developed.

2.2 Tactical Background

The most typical scenario of a multitarget engagement is a submarine versus military force formation (battle group) case. In this case, the targets are formatively scattered. Since the movability (speed) of a marine formation is limited and the separations between the targets are usually large enough (compared with air battle groups), it is quite often true that the first sensor contact involves only one target (and most likely made by the passive noise sonar). Gradually, as the formation gets closer, other targets enter the sight of the sensors, also caught by noise sonar first. This is a picture quite different from that of an air engagement

with radars as major sensors. In an air battle engagement case, the speed of the aircraft formation is so high that the first radar contact is quite possibly the whole formation, which is a dense target problem. From this point of view, it seems much easier to handle the sonar problem than the radar one. Unfortunately, this is not true because the sonar case has its own problems. The number of targets may be smaller, the requirements on reaction time may be not so stringent, but the available information usually has much poorer quality, is inadequate and quite often is of only a passive type.

In addition, when the real engagement begins, which means that the targets notice the existence of the submarine, the situation becomes complicated immediately. Counter actions begin. The formation begins to change. Targets begin to maneuver. They begin to counter detect the submarine by using every possible measure. Before long, they may launch weapons, hard or soft. Only at this time, the real challenge for the sensor system as well as the fusion system comes.

Of the three sonars, the operation of the acoustic pulse surveillance sonar is peculiar. It depends not only on the sonar itself but also on the operation of the active sonars onboard the targets. For the target warships to use active sonars, tactically it often means that they have noticed the submarine threat. If this is the case, the upcoming military actions will be hardly predictable. Although it is very difficult to cope with such a situation, and it seems to be a task more suitable for human intelligence, the fusion system should at least has some measures for this situation.

3. Single-Sensor Multitarget Processing

It is essential to the fusion system that each sensor processes its multitarget positional information effectively. The prerequisite of excellent performance of any fusion system is that each single sensor can provide well-sorted multitarget information within its own domain. The most important positional information passive sonars can get is target bearing sequences. Therefore, the fusion problem is usually bearing-to-bearing fusion or bearing-to-track fusion. There is no ideal tool for such fusion problems, although many powerful techniques are available, which are, however, more suitable for track-to-track fusion problems. In view of this, single-sensor processing is particularly important.

According to the fusion structure proposed in [1], the main goal of single-sensor processing of positional information is to separate multitarget measurements into distinguishable measurement sequences or tracks. The original measurements might be incomplete, tangled with each other, and of course inaccurate, or might be simply false alarms. The basic procedure for such a multitarget processing problem for each sonar may be nothing special but the concrete techniques are not so common. Fig.2 illustrates the processing procedure of single-sensor multitarget information.

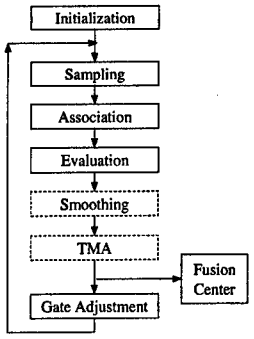


Figure 2. Single sensor processing procedures

3.1 Initialization

System initialization is very important in that it affects the effectiveness of the system significantly. A poorly initialized system can take much longer time to get the desired results than that of a well-initialized one. Sometimes a system could even collapse because of bad initializations. For this passive sonar fusion system, initializations mainly include two aspects. One is the determination of the initial gate size for the measurement association process. The other is the initialization of the association algorithm itself, if the algorithm is a recursive one. Algorithm initialization is a widely studied problem (see, e.g., [2,3]), and thus will not be discussed here.

Two types of measurements - bearings-only and bearings plus ranges - are involved in this system. Correspondingly there are two types of gates. For the bearings-only case, the shape and size of the gate are determined by the bearing gate, which is of a sector shape. For the bearings plus ranges case, the shape and size of the association gate are confined to the bearing gate and the range gate. The mostly widely adopted shape is a ring sector, although other shapes, such as rectangles, can also be used.

Passive noise sonar and acoustic pulse surveillance sonar belong to the bearings-only category. The gate initialization - i.e., the determination of the initial bearing gate size - is not as easy as it appears. It is evident that an optimal size would depend upon many

factors, such as the sampling interval, the speeds and courses of and the distance between the target and the own-ship, the measurement error level and the resolution capability of the corresponding sonar. Most of these factors are not obtainable and thus it is impossible to get a perfect gate size. In practice, conservative measures are taken to get a larger gate. For example, the speeds of the target and the own-ship are replaced by their maximum possible values.

For passive ranging sonar, the sizes of the initial bearing gate and range gate should be determined. Conservative measures are also needed in this case to account for the initial uncertainties. For example, at the beginning, the distance measurement error may be much higher than the normal level, for the distance processor of the sonar may be not yet stable. Factors like this have to be taken into account when determining the gate size. Anyway, sector ring shaped gate is a very common gate. Its counterpart can be easily found in other sensor fusion (e.g., radar fusion) applications.

3.2 Association

In each step, new measurements should be evaluated to determine if they could be associated with any existing sequences or tracks, or simply a starting point of a new sequence or track. When the association gate is determined, this should not be a difficult problem, for which many algorithms are available (see, e.g., [3]). What is important is to develop an algorithm that is acceptable from an engineering point of view. A common approach is to modify an existing algorithm according to the particular requirements of the application.

3.3 Evaluation of Track or Sequence Quality

At the end of each step in the recursive process, each sequence or track should be evaluated in some way. The evaluation result is used to decide as to maintain, modify or abandon the existing sequences or tracks, or to initiate new sequences or tracks. Practically, some simple yet effective techniques are used in real system development. For example, a credit accumulator may be designed to serve as such an evaluator for each sequence or track. For each step, if there is a new measurement that is successfully associated with a particular sequence or track, a certain number of credits are added to the corresponding accumulator. Otherwise, the credits are lowered. Relying on the credit number, a sequence or track may be declared as a false one, a possible one, a conformed one, or discarded one, etc. The thresholds can be determined by offline simulations and underwater trial tests.

3.4 Smoothing and TMA

For a conformed sequence or track, further processing like measurement sequence smoothing and target motion analysis (TMA) can be done to improve the association result. However, it is not necessarily conducted at this stage. With more processed information available, smoothing and TMA may be done more effectively in the fusion center. The fact that bearings-only TMA is difficult and time consuming due to poor observability [4] makes it probably better to handle it in the fusion center. That is why the corresponding boxes of these two parts in Fig. 2 are drawn in dashed lines.

3.5 Gate Adjustment

With more and more information poured in, the picture becomes clearer and clearer. It is very natural that the association gate, usually the gate size only, should be adjusted, although the shape also can be changed. The size can be reduced gradually, i.e., step by step. It can also be reduced periodically. Sometimes it needs to be enlarged when a normal association fails. Albeit seemingly easy, this problem can be troublesome. In practice, however, to determine when and how to adjust the associate gate is a problem of more engineering than theoretical. So engineering tools, such as simulation and trial and error, are always available and are powerful weapons for fighting against this problem.

4. Multisensor Fusion

Multisensor fusion is the fusion center's task. Because the input data from each sensor may be bearing sequences or tracks, three possible fusion forms exist: bearing-to-bearing, bearing-to-track and track-to-track fusion. Which form the fusion center should take depends on the type of data it can get. If bearings-only TMA is not done at the sensor level, which means noise sonar and surveillance sonar can not provide track data, then track-to-track fusion is not possible in this case, because only ranging sonar can provide track data. Even if bearings-only TMA is conducted at the sensor level, track-to-track fusion is not the only fusion form. Bearings-only TMA sometimes can not provide a unique track solution (e.g., before an own-ship maneuver), or can only provide a poor solution (e.g., shortly after an own-ship maneuver, or more generally, under poor observability conditions) [4]. Bearing-to-bearing fusion or bearing-to-track fusion is still necessary in these cases. Anyhow, bearing-to-

bearing fusion and bearing-to-track fusion are more fundamental in passive sonar fusion applications.

Detailed techniques for the aforementioned fusion forms have been introduced in [1].

Fusion results can be sent back to sensor level processors to improve their performances. This feedback channel can also be used by the sensors to help each other. The fact that the detection radius of noise sonar is usually larger than that of ranging sonar makes it quite possible that the multitarget information has already been well processed (e.g., initiated, classified) by the noise sonar before the ranging sonar can detect the target. In this case, the ranging sonar information can be used to refine and enforce the results of the noise sonar. On the other hand, the result of the noise sonar can be used by the ranging sonar to improve its own multitarget information. The poor quality of the bearing measurements makes it very difficult for the surveillance sonar to finish the multitarget positional information processing by itself. The help from the other two sonars and the fusion center is very valuable.

5. Own-Ship Maneuver

Own-ship maneuver is very important in multisensor multitarget tracking. It is also a difficult problem because many factors must be taken into account and not fewer requirements need to be considered. For example, at the initial phase, own-ship maneuver is mainly concerned with enhancing the sensors' capability to detect and distinguish multiple targets. The corresponding requirements, however, differ significantly for different sensors.

Own-ship maneuver in a multitarget environment is quite different from that of a single target. In the single target case, the goal of maneuver is to maximize the degree of the system observability. From a more practical point of view, the criterion is to find maneuver strategies so that the solution of the system converges in the shortest period of time. This has been shown to be a difficult problem. It is further complicated when other basic practical considerations are taken into account, such as ensuring ideal observation of the tracking sensor and ideal target and own-ship geometry for the possible forthcoming attack or other tactical operations.

The multitarget case is no doubt much more challenging. Theoretically, the maneuver optimization criterion for a multitarget system can be defined as maximization of the so-called global degree of observability of the tracking system, which is an index used to measure the comprehensive ability of the system to track all the targets as a whole. However, to use such a criterion to optimize own-ship maneuver strategies may be difficult. First, it is next to impossible to define such a global degree of observability due to the complexity of the problem. As a matter of fact, even the degree of observability for the single target case is still not perfectly defined. Secondly, it would be very difficult to get precise and optimal results that are physically meaningful using this criterion. Thirdly, the implementation of such optimal maneuver strategies, if exist, is very difficult, if not impossible, in practical situations.

Some compromise measures may be taken to cope with this problem. For example, instead of trying to maximize the global degree of observability of the system, a practical alternative is to maximize the degree of observability of a single-target system that involves only the most interesting target. Since it is almost impossible to obtain the states of all targets simultaneously, a surely reasonable solution would be to try to get the state of the most interesting target. How to select the most interesting target is a problem, but not a difficult one. In fact, there are several choices, including the one with the highest signal to noise (S/N) ratio, the one with the fastest rate of bearing changes, the one that exhibits the most serious potential threat, to mention a few. As such, the complicated problem of own-ship maneuver optimization for multitarget tracking is converted into the simpler problem of maneuver optimization for single-target tracking. While a really optimal solution to the single-target tracking problem is still difficult to obtain [5,6], there exist at minimum many rule-of-thumb maneuver strategies that are effective and can be easily implemented (see, e.g., [7]).

Similar to the single-target case, observability is sometimes not the only concern. There might be many other things that should be considered. In practice, the objective of own-ship maneuver in a multitarget environment varies from case to case. For example, when targets are detected by the noise sonar only, which means they are still out of the reach of the ranging sonar. If the range information is needed urgently, the maneuver strategies should be those that get the targets into the detectable zone of the ranging sonar as soon as possible. The resultant maneuver strategies out of this requirement should be quite different than those from the bearings-only observability approach.

For the passive ranging sonar, the requirements are relatively simple. The basic rule is that putting most

targets or the most interesting target on or around the beams of the submarine. In some cases, however, this is not enough. Sensor properties, application environment, and even tracking algorithms can affect own-ship maneuver strategies. For example, under some ideal conditions the detected distance information is highly reliable. Maneuver is not necessary if this is the case. When the detected distance is not so ideal, some algorithms weigh the detected bearing information much heavier than the detected distance information. These algorithms are relatively close to those bearings-only tracking algorithms and distance information plays a supplementary role. Own-ship maneuver strategies no doubt should be also close to those strategies for bearings-only tracking in such cases.

Because the operation range of a passive ranging sonar is relatively small, maintaining stealth while maneuvering is another important concern.

Under some more complicated circumstances, e.g., the targets are also aware of the existence of the submarine, maneuver is not mere a fusion concern any more. It is more a tactical problem in this case. The real decision making burden is left for the commander of the submarine, although some maneuver strategies may be recommended.

6. Some Further Considerations

The corner stone of the hierarchic fusion structure recommended in [1] is distributed processing. It is well known that centralized systems have some advantages over distributed systems, such as higher accuracy. The recommendation of the distributed instead of centralized structure has been justified in [1]. In fact, such a centralized system is very difficult, if not impossible, to realize. For a centralized sonar fusion to be really superior in aspects such as accuracy, the input information has to be directly from the hydrophones of all the sonar arrays. This is almost impossible, especially if the sonars and the fusion system are developed by different manufacturers. In addition, the complexity of underwater acoustic signal processing makes the task of fusing all this tremendous amount of information in a central fashion unbearably tough. Besides the fact they are easy to realize, distributed fusion systems have many nice properties, such as more flexibility and better survivability, that are extremely important for military systems and can well compensate for the possible loss of accuracy.

The coordination of the passive fusion system and the other related systems is another problem that needs attention. Closely or loosely, directly or indirectly, passive sonar fusion system is connected to many other systems, such as other sensor systems, C^2 system, navigation system, weapon system, steering system. The information flow between these systems is very complicated, especially during intensified engagements. The system might collapse if it is not well designed to handle this problem effectively. It is intrinsically a problem of information flow control and management. There are many commercial systems and techniques for this problem, but careful selection and adaptation is required.

7. Conclusion

Passive sonar fusion is the basic and key component of submarine sensor fusion. There are many distinctive features in such a fusion system that need to be properly treated. Only some major aspects have been presented. Several problem-solving principles for system development have also been discussed. It should be emphasized that a modern passive sonar system could be more complex than the model system used in this paper [8]. The system may include more passive sonars, and they may be more diversified. The structure of the system itself may be quite different. In some systems, the sonars are completely independent. There is no information channel at the sensor level. Some other systems, however, are highly synthesized. All their component sonars are connected and organized by data buses, which means the systems themselves are distributed. While the realizations of these systems can be quite different, the basic principles and considerations should be similar.

References

1. P. Shar and X. R. Li, "Some Considerations of Submarine Sensor Fusion," *Proc. of 1998 Int. Conf. on Information Fusion (FUSION'98)*, Vol. II, Las Vegas, Nevada. July 1998.
2. Y. Bar-Shalom and X. R. Li, *Estimation and Tracking*, Artech House, 1993.
3. Y. Bar-Shalom and X. R. Li, *Multitarget-Multisensor Tracking*, YBS Publishing, 1995.
4. S. C. Nardone and V. J. Aidala, "Observability Criteria for Bearings-Only Tracking," *IEEE T-AES-17*, No. 2, July 1981.
5. J. P. Helferty and D. R. Mudgett, "Optimal Observer Trajectories for Bearings-Only Tracking by Minimizing the Trace of the Cramer-Rao Lower Bound," *Proc. of the 32nd Conf. on Decision and Control*, San Antonio, Texas, Dec. 1993.
6. J. M. Passerieux and D. Van Cappel, "Optimal Observer Maneuver for Bearings-Only Tracking," *IEEE T-AES-34*, No. 3, July 1998.

7. B. J. MacCabe, "Accuracy and Tactical Implications of Bearings-Only Ranging Algorithms," *Operation Research*, Vol. 33, No. 1, Jan.-Feb. 1985.
8. N. H. Guertin and R. W. Miller, "A-RCI—The Right Way to Submarine Superiority," *Naval Engineers Journal*, Mar. 1998.

A PDAF with a Bayesian Detector

Ruixin Niu and Peter Willett *
Electrical and Systems Engineering
U-157, University of Connecticut
Storrs, CT 06269 USA

Abstract *Practical detection systems generally are operated using a fixed threshold, optimized to the Neyman-Pearson criterion. An alternative is Bayes detection, in which the threshold varies according to the ratio of prior probabilities. This prior information is available in a tracking situation, but appears little used. The effect here is of a depressed detection threshold near the predicted measurement. The PDAF must be modified if used with such a detector. We provide this modification and test: it is considerably better than that the PDAF both in terms of tracking accuracy and effort, and in the former offers only slight degradation relative to the PDAF using amplitude information.*

Keywords: Target tracking, PDAF, Detection

1 Introduction

Most target tracking systems work with the data they are given. By this is meant that measurements from a detection "front-end" processor are interrogated for threshold exceedances, and these "hits" are delivered to the tracking algorithm. For the most part the threshold is set and fixed according to a false-alarm criterion, that there should be on average a specified number of false hits per unit volume. There have been studies relating the tracking performance to this threshold, and suggesting threshold-settings for optimized performance for a given expected signal-to-noise ratio (SNR). Further, there has been some research indicating that considerably improved performance is achievable when some amplitude information (AI) is delivered to the tracker along with the measurements and their locations [3, 4].

The above two points have largely been investigated as they pertain to the probabilistic data association filter (PDAF) [1]. The PDAF is a particularly simple and successful target tracking algorithm: it is predicated on the assumptions that the best one-step estimation of the target's location should be sufficient, and that once this estimation is accomplished the target's true location should be afforded a Gaussian distribution about its estimated value. The key to this paper, as alluded to by the PDAF's Bayesian nature, is in this Gaussian "posterior" distribution on the target's location.

Communication between the signal processing front-end and the PDAF is presently one-way. In this paper we allow two-way communication; or, perhaps more appropriately, "feedback" from the tracker to the detector. The form of this feedback is of the posterior distribution on the target's location. From the detector's point of view this is "prior" information for its hypothesis tests (i.e. its matched filters), as represented in figure 1. Thus, a detector using this operation ceases to be used in a Neyman-Pearson mode and becomes Bayesian, and from a practical point of view this amounts to a threshold which is depressed near where a target is expected to be and elevated where it is unexpected – this is illustrated in figure 2.

In this new approach there are fewer false-alarms than previous, and these are no longer uniformly distributed in space. Thus, the PDAF must be modified accordingly, which we do in this paper. The performance of the PDAF so obtained is compared favorably to that of the original PDAF, and turns out to be essentially equivalent to that of the version of

*willett@engr.uconn.edu

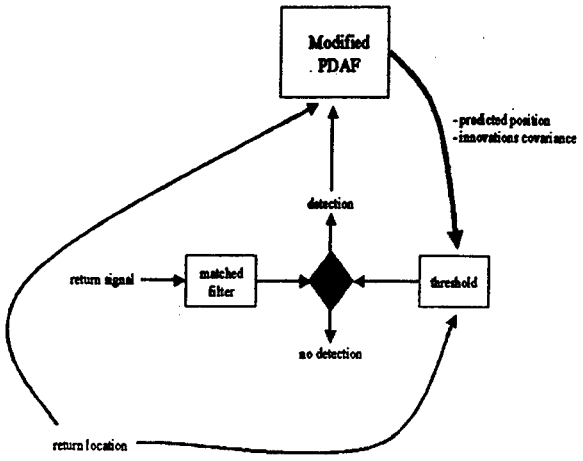


Figure 1: Representation of flow of data within proposed system. A signal return, from a known location is matched-filtered and its magnitude compared to a threshold – a threshold exceedance, along with its location, is passed to the tracker, a modified PDAF. The threshold itself is determined as a function of the predicted location of the target, the innovation covariance, and the location of the return.

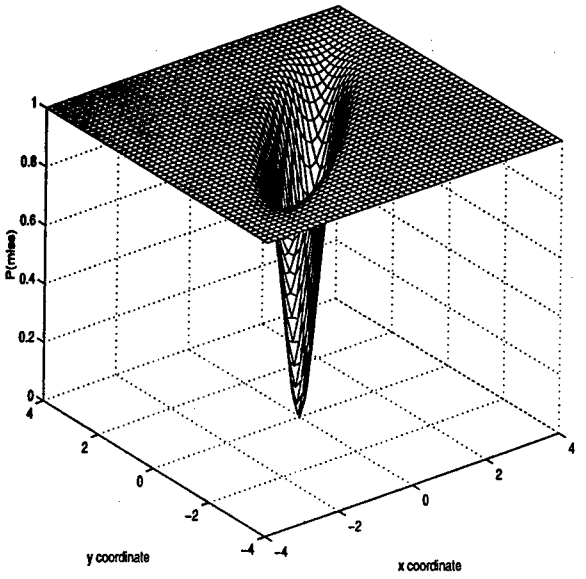


Figure 2: Illustration of the effect of a position-dependent threshold. The x and y coordinates are those of the innovation; that is of the one-step predicted measurement subtracted from the return location. The z coordinate shows the probability that a return of a given strength be missed, as a function of its normalized innovation.

the PDAF which incorporates amplitude information (the PDAF-AI).

2 Development of the New PDAF-BD

At the outset let us note the informing feature of the PDAF: it is entirely optimal *except* that after each scan its posterior track probability density function – ideally a mixture of Gaussian pdf's – is converted to a single Gaussian mode having the same mean and variance. Thus, at each scan, estimation is built upon a Gaussian prior, converted to a Gaussian mixture posterior, which is then forced back to Gaussianity for the succeeding scan.

Operation of the PDAF based on one scan of data (the k^{th}) can be summarized as:

1. Predict state at scan k from prior at scan $k-1$.
2. Form "innovations" (ν's), all measurements at scan k minus the predicted value.
3. Gate the measurements according to the innovations and the innovations covariance (**S**). Remove observations outside the gate.
4. Calculate the association probabilities (β's), meaning the posterior probabilities that each surviving measurement is the true one, or that none is.
5. Use these β's to form a synthetic "innovation", and use the Kalman gain formula to update the track.
6. Use the β's to calculate a modified estimation covariance (**P**).

It has recently been shown [4] that the use of amplitude information can be of significant benefit to the PDAF. That is, instead of "measurements" consisting simply of the locations of threshold exceedances, these are augmented by information as to how much the threshold was exceeded. Thus, it may be expected that a strong target return would be more recognizable as such than if this confidence information were thrown away by the detector, and in fact this is so. It is interesting that the PDAF

structure is little altered by the presence of amplitude information: the only change is to the calculation of the β's.

This extends to the PDAF-BD: although the signal-processing (meaning the thresholding) is different from either PDAF or PDAF-AI, the only variation is again in the calculation of the β's. In the following we show how to form these.

First, however, let us agree on the standard tracking terminology that

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{F}\mathbf{x}_k + \mathbf{v}_k \\ \mathbf{y}_k &= \mathbf{H}\mathbf{x}_k + \mathbf{w}_k \end{aligned} \tag{1}$$

in which $\mathbf{x}$ is the target state, $\mathbf{y}$ the measurement, and that the Gaussian noises are independent, white, and have

$$\begin{aligned} E\{\mathbf{v}_k\mathbf{v}_k^T\} &= \mathbf{Q} \\ E\{\mathbf{w}_k\mathbf{w}_k^T\} &= \mathbf{R} \end{aligned}$$

as their associated covariance matrices.

2.1 The Statistical Testing

We assume that a test of presence or absence of a target at location $\mathbf{z}_k(l)$ is to be performed. The hypotheses are:

$$\begin{aligned} \mathbf{H}: \quad & Pr(A_k(l) > a) = e^{-a} \\ \mathbf{K}: \quad & Pr(A_k(l) > a) = \frac{1}{1+\rho} e^{-a/(1+\rho)} \end{aligned} \tag{2}$$

in which $\mathbf{A}_k(l)$ is the corresponding amplitude (magnitude-square output of a matched filter, with a Swerling I target fluctuation model implicit), and ρ is the signal to noise ratio.[1] The usual implementation is according to the Neyman-Pearson criterion [5], that the probability of detection be maximized subject to a constraint on the false alarm rate, and the resulting test can easily be shown to be a comparison of $\mathbf{A}_k(l)$ to a fixed threshold. From the Bayesian viewpoint, the appropriate test is

$$\frac{f_K(\mathbf{A}_k(l))}{f_H(\mathbf{A}_k(l))} \underset{K}{\overset{H}{\lessgtr}} \frac{Pr(H)[c_{KH} - c_{HH}]}{Pr(K)[c_{HK} - c_{KK}]} \tag{3}$$

[1] In the formulation given it is apparent that the returns are assumed perfectly pre-normalized such that the target-absent mean is unity. If some other target-model – such as a CA-CFAR distribution – is desirable, then the succeeding development must be modified. This modification is straightforward.

in which $Pr(j)$ denotes the hypothesis j ($\in \{H, K\}$), $f_j(\cdot)$ is the pdf given hypothesis j, and c_{ij} refers to the cost of making decision i when j is true. We note that these costs are not easily available.

The "prior" probabilities $Pr(H)$ and $Pr(K)$ are not well-posed: the latter amounts to the probability that a target is located *exactly* at the test's coordinates $\mathbf{z}_k(l)$ given the prior tracking information, and this is zero. If a sampling grid of resolution cells is available, then the quantity can be calculated; but since the answer is configuration-specific we prefer to avoid this, and simply note that

$$\frac{Pr(K)}{Pr(H)} \propto e^{-\frac{1}{2}\nu_k(l)^T \mathbf{S}_k^{-1} \nu_k(l)} \tag{4}$$

in which

$$\nu_k(l) = \mathbf{z}_k(l) - \mathbf{H}\mathbf{F}\hat{\mathbf{x}}_{k-1|k-1} \tag{5}$$

is the *spatial* innovation of the l^{th} measurement at time k, and $\hat{\mathbf{x}}_{k-1|k-1}$ is the estimate of the state given data up to scan $k-1$. It should be noted that the prior probabilities are of *isolated* tests, meaning that each observation is tested separately, as is fair.

At any rate, from (3,3,4) we have with reference to figure 1 the test:

$$\mathbf{A}_k(l) \underset{K}{\overset{H}{\lessgtr}} \frac{\rho+1}{2\rho} \nu_k(l)^T \mathbf{S}_k^{-1} \nu_k(l) + \eta \tag{6}$$

in which η is a tunable parameter.

It is useful to calculate the resulting probability of detection as

$$\begin{aligned} P_d &= \int Pr(detection \mid \nu) f(\nu) d\nu \\ &= \int e^{-\frac{1}{\rho+1}[\frac{\rho+1}{\rho}\nu^T \mathbf{S}^{-1}\nu + \eta]} \\ &\quad \times \frac{1}{\sqrt{2\pi|\mathbf{S}|}} e^{-\frac{1}{2}\nu^T \mathbf{S}^{-1}\nu} d\nu \\ &= \frac{1}{\sqrt{2\pi|\mathbf{S}|}} e^{-\eta/(1+\rho)} \int e^{-\frac{1}{2}\nu^T [\frac{\rho}{\rho+1}\mathbf{S}]^{-1}\nu} d\nu \\ &= \left(\frac{\rho}{\rho+1}\right)^{n_z/2} e^{-\eta/(\rho+1)} \end{aligned} \tag{7}$$

in which n_z is the dimension of the measurement.

2.2 The Probability of the Number of False Alarms

In the PDAF (and PDAF-AI) the number of false alarms is assumed Poisson, and hence we have the probability mass function (pmf) of the number of false alarms in a "volume" V as

$$\mu(m) = \frac{(\lambda_{pdaf}V)^m}{m!}e^{-\lambda_{pdaf}V} \quad (8)$$

in which λ_{pdaf} is the average number of false alarms per unit volume. The expression for μ is necessary in the evaluation of the β's. The above is so simple that it may seem strange to devote much space to it; but in the case of the PDAF-BD the number of false alarms is controlled by the detection thresholding, and hence the answer is not straightforward.

We begin by assuming that there is an underlying Poisson process with parameter λ, which produces "points" at which detection decisions can be made. (This λ is in general *not* the same as λ_{pdaf}, since not all points produce detections. In fact, assuming the hypothesis model (3) and that the basic PDAF uses a detection threshold τ_{pdaf}, we have $\lambda = \lambda_{pdaf}e^{\tau_{pdaf}}$.) Assume that the Poisson process has produced an event with an innovation ν; then we have as the probability of a false alarm

$$\begin{aligned} P_{fa} &= \int_V Pr(\text{false alarm}|\nu)f(\nu)d\nu \quad (9)\\ &= \int_V e^{-\frac{\rho+1}{2\rho}\nu_k(l)^T\mathbf{S}_k^{-1}\nu_k(l)-\eta}\frac{1}{V}d\nu \\ &= \frac{1}{V}e^{-\eta}\sqrt{2\pi\left|\frac{\rho}{\rho+1}\mathbf{S}\right|} \end{aligned}$$

if V is sufficiently large.

Now, assume that the underlying Poisson process has generated n points in the volume V. The probability that there are m threshold exceedances (false alarms) is binomial with parameter P_{fa}, provided that $n \geq m$. That is, we have for the probability that there are m false alarms

$$\begin{aligned} \mu(m) &= \sum_{n=m}^{\infty}\frac{n!}{m!(n-m)!}(P_{fa})^m(1-P_{fa})^{n-m} \\ &\quad \times\frac{(\lambda V)^n}{n!}e^{-\lambda V} \\ &= \frac{(\lambda V P_{fa})^m}{m!}e^{-\lambda V}\sum_{l=0}^{\infty}\frac{(\lambda V)^l}{l!}(1-P_{fa})^l \\ &= \frac{(\lambda V P_{fa})^m}{m!}e^{-\lambda V P_{fa}} \quad (10) \end{aligned}$$

in which P_{fa} is given in (9). Note that VP_{fa} is independent of V, as expected. Our final result: the number of false alarms under the new detection model is again Poisson, but with a modified parameter.

2.3 Calculation of the β's

Assuming that at the present scan we have $m \in \{0, 1, \ldots\}$ threshold exceedances, then according to [1] we define the events $\theta \in \{0, 1, \ldots, m\}$ such that $\theta = i$ means that the i^{th} measurement is target-generated and the others are false-alarms; and $\theta = 0$ that all measurements are false. We denote by $\beta(\theta)$ the probability, conditioned on all measurements at scan k (and naturally, implicitly, on m), that θ is true.

We first require the conditional observation probabilities

$$f(\mathbf{z}_k(l) \mid \theta = l, m) = \frac{1}{\sqrt{2\pi\,|\mathbf{S}_k|}}e^{-\frac{1}{2}\nu_k(l)^T\mathbf{S}_k^{-1}\nu_k(l)} \quad (11)$$

and for $p \neq l$

$$\begin{aligned} &f(\mathbf{z}_k(l) \mid \theta = p, m) \\ &= \frac{Pr(\text{FA at } \mathbf{z}_k(l)|\mathbf{z}_k(l))f(\mathbf{z}_k(l))}{Pr(\text{FA}} \\ &\propto e^{-\frac{\rho+1}{2\rho}\nu_k(l)^T\mathbf{S}_k^{-1}\nu_k(l)-\eta} \\ &= \frac{e^{-\frac{1}{2}\nu_k(l)^T[\frac{\rho}{\rho+1}\mathbf{S}_k]^{-1}\nu_k(l)}}{\sqrt{2\pi\left|\frac{\rho}{\rho+1}\mathbf{S}_k\right|}} \quad (12) \end{aligned}$$

in which "FA" refers to a threshold exceedance by a realization from the underlying (factitious) Poisson point process. This is particularly interesting: under the new detection model any randomly-chosen false-alarm has a Gaussian distribution. For the standard PDAF model, this probability is naturally uniform.

Overall, then, we get (see [1, 2])

$$\begin{aligned} \beta(0) &= c_1 f(\{\mathbf{z}_k(l)\}_{l=1}^m \mid \theta = 0, m) \quad (13)\\ &\quad \times P(\theta = 0|m)\mu(m) \end{aligned}$$

$$= c_2 \frac{e^{-\frac{1}{2}\sum_{l=1}^{m} \nu_k(l)^T [\frac{\rho}{\rho+1}\mathbf{S}_k]^{-1} \nu_k(l)}}{\left(\sqrt{2\pi \left|\frac{\rho}{\rho+1}\mathbf{S}_k\right|}\right)^m} \times (1-P_d)\frac{(\lambda V P_{fa})^m}{m!} e^{-\lambda V P_{fa}}$$

and for $\theta \neq 0$

$$\begin{aligned}\beta(\theta) &= c_1 f(\{\mathbf{z}_k(l)\}_{l=1}^m \mid \theta, m) P(\theta|m)\mu(m) \\ &= c_2 \frac{e^{-\frac{1}{2}\sum_{l\neq\theta} \nu_k(l)^T [\frac{\rho}{\rho+1}\mathbf{S}_k]^{-1} \nu_k(l)}}{\left(\sqrt{2\pi \left|\frac{\rho}{\rho+1}\mathbf{S}_k\right|}\right)^{m-1}} \\ &\quad \times \frac{e^{-\frac{1}{2}\nu_k(\theta)^T \mathbf{S}_k^{-1} \nu_k(\theta)}}{\sqrt{2\pi |\mathbf{S}_k|}} \\ &\quad \times \frac{P_d}{m}\frac{(\lambda V P_{fa})(m-1)}{(m-1)!} e^{-\lambda V P_{fa}}\end{aligned} \tag{14}$$

where P_d and P_{fa} are given by (7) and (9) respectively. Since these add to unity, it is simpler and computationally more appealing to write them as

$$\begin{aligned}\beta(0) &= c_3 \left(\frac{\rho+1}{\rho}\right)^{n_z/2} \frac{1-P_d}{P_d} \\ &\quad \times \frac{1}{\lambda e^{-\eta}\sqrt{2\pi \left|\frac{\rho}{\rho+1}\mathbf{S}\right|}}\end{aligned} \tag{15}$$

and for $\theta \neq 0$

$$\beta(\theta) = c_3 e^{-\frac{1}{2}\nu_k(\theta)^T \mathbf{S}_k^{-1} \nu_k(\theta)} \tag{16}$$

Since the β are probabilities, we must have

$$\sum_{\theta=0}^{m} \beta(\theta) = 1 \tag{17}$$

which suffices to specify the normalizing constant c_3.

An illustration is plotted in figure 3, a plot of β versus normalized innovation $\nu \mathbf{S}^{-1} \nu$ in the case that there is but one threshold exceedance. Two items are of note. The first is that β_1 is relatively large; but this is as it should be, since in effect the detection model is "pre-screening" the detections. The second is that $\beta(1)$ actually *increases* with as the distance from the predicted measurement increases. This may appear anomalous, but

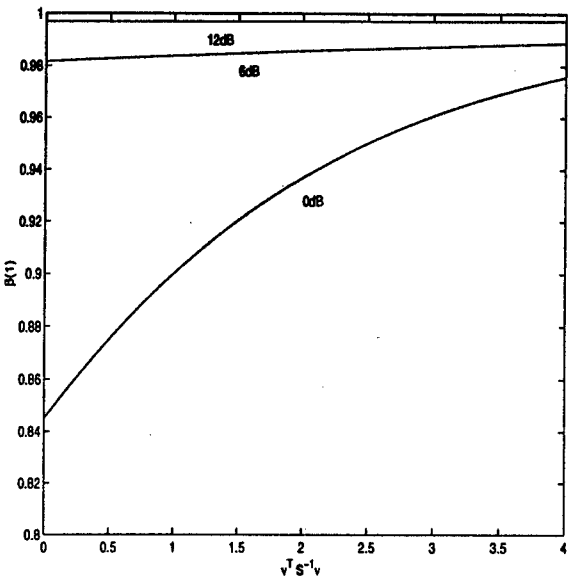


Figure 3: An example of the value of $\beta(1)$ (the posterior probability that a threshold exceedance is target-generated) given that $m = 1$, for various signal to noise ratios (ρ).

in fact given the thresholding a measurement more spatially "surprising" must have had a larger amplitude to exceed its threshold, and hence is apparently more likely to have come from the target. The case of two threshold exceedances is explored in figure 4.

3 Comparison

In this section we compare the PDAF, the PDAF-AI, and the PDAF-BD. For our simulations we choose a two-dimensionally kinematic model with direct discrete-time process noise. According to the kinematic model we have

$$\mathbf{F} = \begin{Bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{Bmatrix}$$

$$\mathbf{H} = \begin{Bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{Bmatrix}$$

$$\mathbf{Q} = \sigma_p^2 \begin{Bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} & 0 & 0 \\ \frac{\Delta t^3}{2} & \Delta t^2 & 0 & 0 \\ 0 & 0 & \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ 0 & 0 & \frac{\Delta t^3}{2} & \Delta t^2 \end{Bmatrix}$$

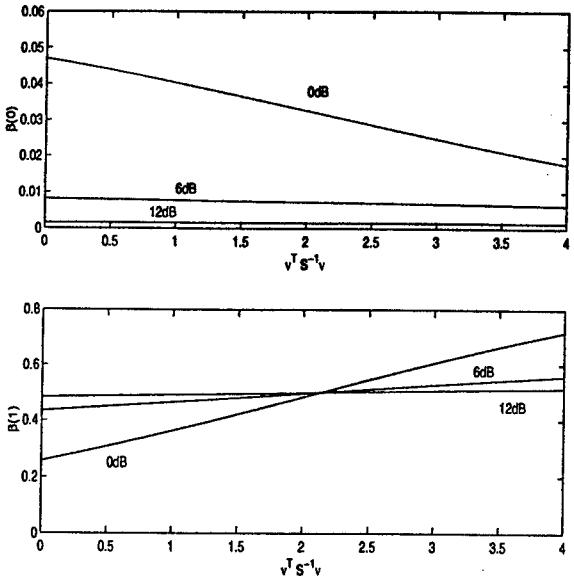


Figure 4: An example of the value of $\beta(0)$ (above) and $\beta(1)$ (below), given that $m = 2$ and $\nu(2)^T \mathbf{S}^{-1} \nu(2) = 2$, for various signal to noise ratios (ρ).

$$\mathbf{R} = \sigma_m^2 \begin{Bmatrix} 1 & 0 \\ 0 & 1 \end{Bmatrix} \tag{18}$$

Measurements are of position-only, and the model is in all respects linear. True detections are generated along with an associated amplitude; thresholding of this amplitude determines whether or not there is a miss. Some notes and parameter values:

- We have chosen $\Delta t = 30$ seconds, a fast but not-unreasonable scan rate for active sonar.
- We have $\sigma_m = 10, 100$ meters, corresponding to a constant-frequency pulse with length of the order of one-tenth and one second, respectively.
- We explore process noises $\sigma_p = .1, .01$ meter/second^2.
- We explore signal to noise ratios $\rho = 6$ and $12dB$.
- We explore various clutter densities, $\lambda \in \{10^{-5.5}, 10^{-6}, 10^{-6.5}, 10^{-7}\}$.
- We choose a track length $T = 100$ scans of data.
- Tracks are initialized by two-point differencing.

Targets begin their trajectories at scan $k = 0$ with position coordinates (0,0) and velocity coordinates (5,5) meters per second, corresponding to 13.8 knots. A typical, but somewhat self-serving, track output is given in figure 5.

Most studies of tracking performance are parameterized by P_d and by the clutter return density λ. In this case we cannot use the former, since for the new approach P_d is not constant; hence we use the signal to noise ratio ρ instead. Each of the schemes takes as a parameter the detection threshold, given simply by τ for the PDAF and PDAF-AI, but in a more implicit fashion by η in the PDAF-BD. We have no particular insight at present as to how η should be chosen, hence we adopt the simple expedient that the aggregate probabilities of detection for all three schemes be the same, meaning

$$\eta = -(1+\rho) \log \left[P_d(pdaf)((\rho+1)/\rho)^n z \right] \tag{19}$$

from (7), and in which $P_d(pdaf) = e^{-\tau/(1+\rho)}$. The explicit appearance of τ here amplifies the fact that independent specification of λ may be incompatible with σ_m – there is in fact through σ_m an implied resolution cell grid at which threshold-exceedances are interrogated, and, for example, $\lambda = 10^{-5} m^{-2}$ and $\sigma_m = 100m$ makes very little sense indeed. Thus, we have adopted the convention that

$$\tau = -\log \left[12 \sigma_m^2 \lambda \right] \tag{20}$$

with the intuition that resolution cells be square and of side $\sqrt{12}\sigma_m$. Since with the probabilities of detection the same in all schemes we must have $\eta < \tau$, and since the Poisson process assumed for the PDAF-BD must be more dense than that assumed for the PDAF and PDAF-AI, our simulations generate a Poisson process with spatial density $\lambda e^{\tau - \eta}$. For each clutter point so generated we also form an amplitude variate with distribution

$$f(a) = \begin{cases} e^{\eta - a} & a > \eta \\ 0 & a \leq \eta \end{cases} \tag{21}$$

which is thresholded using τ for the PDAF and PDAF-AI, and using (6) for the PDAF-BD.

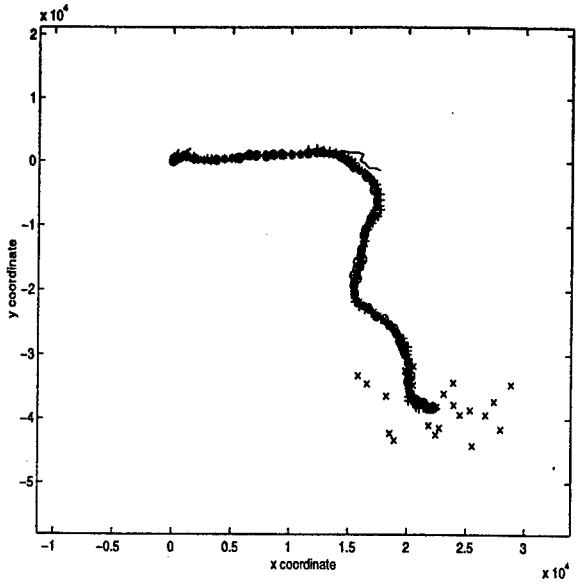


Figure 5: Example of track with $\rho = 4$, $\lambda = 10^{-6}m^{-2}$, $\sigma_p = 0.1$, $\sigma_m = 100$. The PDAF loses track early; the PDAF-AI somewhat later. The PDAF-BD holds track for the full 100 scans.

σ_m	σ_p	ρ (dB)	λ	PD	AI	BD
10	0.1	6	$10^{-5.5}$	0	0	0
10	0.1	6	10^{-6}	0	1	0
10	0.1	6	$10^{-6.5}$	0	5	0
10	0.1	6	10^{-7}	0	11	0
10	0.1	12	$10^{-5.5}$	0	65	30
10	0.1	12	10^{-6}	0	66	68
10	0.1	12	$10^{-6.5}$	5	64	68
10	0.1	12	10^{-7}	36	61	66
10	0.01	6	$10^{-5.5}$	0	71	71
10	0.01	6	10^{-6}	1	62	63
10	0.01	6	$10^{-6.5}$	3	57	59
10	0.01	6	10^{-7}	5	47	50
10	0.01	12	$10^{-5.5}$	94	95	98
10	0.01	12	10^{-6}	93	94	98
10	0.01	12	$10^{-6.5}$	93	93	96
10	0.01	12	10^{-7}	92	92	95
100	0.1	6	$10^{-5.5}$	0	39	11
100	0.1	6	10^{-6}	0	74	77
100	0.1	6	$10^{-6.5}$	0	81	78
100	0.1	6	10^{-7}	0	78	75
100	0.1	12	$10^{-5.5}$	0	73	77
100	0.1	12	10^{-6}	87	91	99
100	0.1	12	$10^{-6.5}$	91	93	99
100	0.1	12	10^{-7}	93	95	99
100	0.01	6	$10^{-5.5}$	76	97	88
100	0.01	6	10^{-6}	88	99	99
100	0.01	6	$10^{-6.5}$	91	99	99
100	0.01	6	10^{-7}	94	98	98
100	0.01	12	$10^{-5.5}$	96	97	100
100	0.01	12	10^{-6}	99	99	100
100	0.01	12	$10^{-6.5}$	100	100	100
100	0.01	12	10^{-7}	100	100	100

Table 1: The in-track percentage for various situations. The latter three columns refer to the PDAF, the PDAF-AI and the new PDAF-BD.

The results are given in tables 1 and 2, respectively the *in-track* percentage and tracking RMSE. A simulation is judged "in-track" if at the end of 100 scans the true and estimated positions are less than $\sqrt{2}(4\sigma_m)$ apart. The tracking error is the RMSE *over the whole track* for those simulations which are in-track at their conclusion. What we notice from table 1 is that the PDAF-BD has essentially the same tracking performance as the PDAF-AI, that in general considerably better than the PDAF – this is true except for a few very difficult situations. This is also true for the RMSE, although in this case, since the RMSE is only calculated for those tracks which are good, the "straight" PDAF appears to perform well. It is arguable that the PDAF-BD has a slight advantage over the PDAF-AI. This at first seems impossible, since the information given to the PDAF-AI should be a super-set of that given to the PDAF-BD; but we assume this is due to the fact that the thresholds τ and η have in no sense been optimized, and that given to the PDAF-BD may happen to be a little more advantageous.

4 Summary

The usual target tracking model is of separation between detection and tracking subsystems. In the absence of information from the latter, the former has little choice but to do the best job it can: it provides Neyman-Pearson optimal performance, the most powerful test subject to a constraint on false-alarm rate. If there *is* some information flow from tracker to detector, particularly in terms of predicted measurement location and association confidence (innovations covariance), then a *Bayesian* detector is appropriate: then difference is not in the statistic tested, but rather in the threshold. In fact, assuming that the prior probability is Gaussian (which fits with the PDAF assumptions, hence our use of this model) the threshold is proportional to the normalized innovation, and hence is lowest near where a detection is expected, at the predicted measurement.

In the paper the threshold shape has been derived, and appropriate modification to the

σ_m	σ_p	ρ (dB)	λ	MSE_p	MSE_a	MSE_b
10	0.1	6	$10^{-5.5}$	—	276	—
10	0.1	6	10^{-6}	—	467	—
10	0.1	6	$10^{-6.5}$	—	834	—
10	0.1	6	10^{-7}	—	1355	—
10	0.1	12	$10^{-5.5}$	—	103	121
10	0.1	12	10^{-6}	—	129	127
10	0.1	12	$10^{-6.5}$	149	159	141
10	0.1	12	10^{-7}	209	190	160
10	0.01	6	$10^{-5.5}$	33	68	63
10	0.01	6	10^{-6}	59	95	85
10	0.01	6	$10^{-6.5}$	87	127	108
10	0.01	6	10^{-7}	123	176	152
10	0.01	12	$10^{-5.5}$	21	21	19
10	0.01	12	10^{-6}	23	23	21
10	0.01	12	$10^{-6.5}$	26	26	24
10	0.01	12	10^{-7}	28	29	26
100	0.1	6	$10^{-5.5}$	—	225	314
100	0.1	6	10^{-6}	—	284	243
100	0.1	6	$10^{-6.5}$	—	367	324
100	0.1	6	10^{-7}	—	500	450
100	0.1	12	$10^{-5.5}$	195	145	207
100	0.1	12	10^{-6}	173	154	147
100	0.1	12	$10^{-6.5}$	181	169	159
100	0.1	12	10^{-7}	194	188	174
100	0.01	6	$10^{-5.5}$	122	105	118
100	0.01	6	10^{-6}	123	115	111
100	0.01	6	$10^{-6.5}$	142	133	127
100	0.01	6	10^{-7}	163	155	148
100	0.01	12	$10^{-5.5}$	100	92	96
100	0.01	12	10^{-6}	92	91	88
100	0.01	12	$10^{-6.5}$	85	84	82
100	0.01	12	10^{-7}	88	88	86

Table 2: The RMSE for various situations. The latter three columns refer to the PDAF, the PDAF-AI and the new PDAF-BD.

PDAF (we call it the "PDAF-BD") made. Simulation has revealed that the performance is considerably better than that of the PDAF, and is only slightly degraded relative to the PDAF-AI – that version of the PDAF appropriate to possession of full amplitude information for *all* returns.

We note:

- Since, in effect, only detections close to the predicted measurement are allowed, the PDAF-BD is less of a computational load than the others. In fact, our simulations have shown this to be only a small difference.
- As far as we are aware there is at present no detection system which allows non-constant thresholding, at least not on the scale proposed here. Thus, this work, if at all useful, is several generations ahead of its platform. We have no intention of trying to "over-sell" the idea, and hope that we do not seem as if we are.
- As future work we intend to explore maneuver, particular the IMM. This should prove interesting since the different maneuver models should ideally use different thresholds.
- As future work we intend to explore multiple targets and multiple sensors.

As to the last bullet above, the new scheme may prove particularly useful for *detection fusion*, since the information required to be communicated among platforms is quite abbreviated, and is limited to location (no amplitude).

Acknowledgment

This work was supported by AFOSR under contract F49620-95-1-0299, and by ONR under contract N00014-97-1-0502.

References

[1] Y. Bar-Shalom, X.R. Li, *Estimation and Tracking: Principles, Techniques and Software*, Artech House, Inc., 1993.

[2] Y. Bar-Shalom, X.R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*, YBS Publishing, 1995.

[3] T. Kirubarajaran and Y. Bar-Shalom, "Low Observable Target Motion Analysis Using Amplitude Information", *to be published in IEEE Transactions on Aerospace and Electronic Systems*, July 1996.

[4] D. Lerro and Y. Bar-Shalom, "Interacting Multiple Model Tracking with Target Amplitude Feature", *IEEE Transactions on Aerospace and Electronic Systems*, April 1993.

[5] H.V. Poor, "An Introduction to Signal Detection and Estimation", *Springer-Verlag*, 1988.

A Depth Control Pruning Mechanism for Multiple Hypothesis Tracking

Jean Roy
Decision Support Technology Section
Defence Research Establishment Valcartier
2459 Pie-XI Blvd. North, Val-Bélair
Québec, Canada, G3J 1X5
jean.roy@drev.dnd.ca

Nicolas Duclos-Hindié, Dany Dessureault
Groupe Informission Inc.
1260 Lebourgneuf Blvd., Office 250
Québec, Canada, G2K 2G2
nicolas@informission.ca
dany.dessureault@informission.ca

Abstract - *Computational requirements represent the main drawback of the Multiple Hypothesis Tracking (MHT) data association algorithm. To reduce these requirements, the number of hypotheses must be limited through the use of pruning methods. This paper presents a depth control pruning mechanism making hard decisions on the origin of input data elements contained in the hypothesis tree. Inherently, the MHT uses later input data to aid in evaluating prior correlation decisions. Ultimately though, a final decision has to be made. The depth control mechanism is used to transfer the assignment of an input data element from the "hypothetical" section of the hypothesis tree to the "definitive decision" section. The waiting period is determined by the number of target observation attempts made that can be used to resolve a particular assignment. The occurrence, the duration, the quality and the result of a target observation attempt are concepts discussed in the paper. Some depth control pruning simulation results are also presented.*

Keywords: multiple hypothesis tracking, MHT, data association, pruning, hypothesis tree

1.0 Introduction

From the point of view of tracking multiple targets in a cluttered environment, the data association process can make either hard decisions or soft decisions about which of a number of hypotheses best describes the origin of input data elements received from a sensor. A hard decision is a definitive assignment to one and only one origin, while a soft decision allows the data to be assigned to multiple origins, with each candidate assignment having a measure of uncertainty. The soft decision approach typically results in multiple association hypotheses being maintained until additional input data elements have been collected and there is enough information data available to reduce the uncertainty and to substantiate or refute the prior hypothetical assignments. In principle, this approach should lead to the most accurate association results. However, the computational requirements necessitated to retain multiple interpretations of the situation represent the main drawback of the standard (hypothesis oriented) Multiple Hypothesis Tracking (MHT) data association algorithm (Refs. 1-8).

To reduce these computational requirements, the number of data association hypotheses must be limited (sometime sacrificing optimal Bayesian inference) through the use of hypothesis pruning and combining methods. In terms of pruning, both the width and the depth of the hypothesis tree (i.e., the number of hypotheses maintained and the number of levels in the tree respectively) can be controlled. This paper presents a depth control pruning mechanism that forces hard (or definitive) decisions on the origin of input data elements contained in the hypothesis tree.

The paper is organized as follows. Section 2.0 discusses the hypothesis tree of the MHT and the dynamic data structure used to implement it. Section 3.0 gives a brief introduction to width pruning while section 4.0 describes the depth control pruning mechanism in length. Section 5.0 discusses the concept of target observation attempts and, finally, section 6.0 presents some depth control pruning simulation results.

2.0 The hypothesis tree

Central to the MHT approach is the formation of a hypothesis tree. The discussion in this paper focuses on the standard MHT algorithm implementations (i.e., those that support explicit hypothesis propagation over time as in Refs. 1-2) as opposed to the implementations based on structured branching (SB/MHT, Refs. 3-8).

When there is no limitation, all possibilities concerning the origin of received input data elements are enumerated as alternative hypotheses organized in a tree (Fig. 1). These hypotheses contain groupings of some input data elements into tracks, and the identification of other such elements to be false targets (Refs. 1-2). As a new set of input data elements is received, a new set of data association hypotheses is formed by extending the existing prior hypotheses of the tree with the feasible correlation hypotheses that account for all possible origins of the new input data elements.

The growth of hypotheses must however be limited if the MHT implementation is to be feasible. Hence, before a new hypothesis is created, the candidate track must typically satisfy a set of conditions (e.g., require that an input data element

satisfies a gating relationship before an assignment to a track is made, etc.).

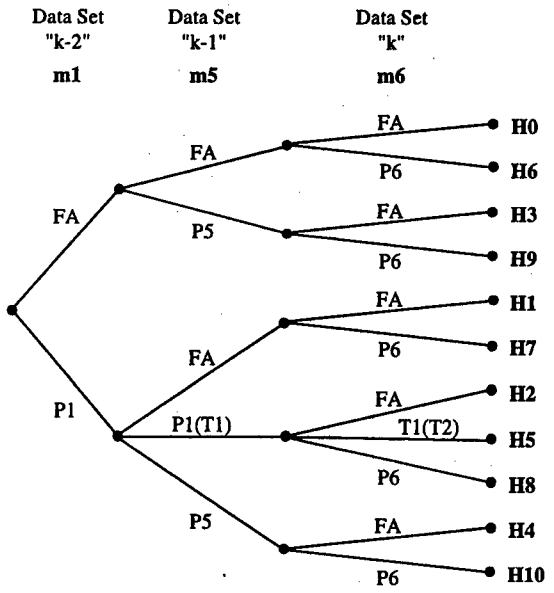


Figure 1. A typical hypothesis tree

2.1 Dynamic data structure

Figure 2 illustrates how a dynamic data structure using pointers and other software constructions can be used to represent the actual architecture of a typical hypothesis tree (such as the one presented in the previous subsection).

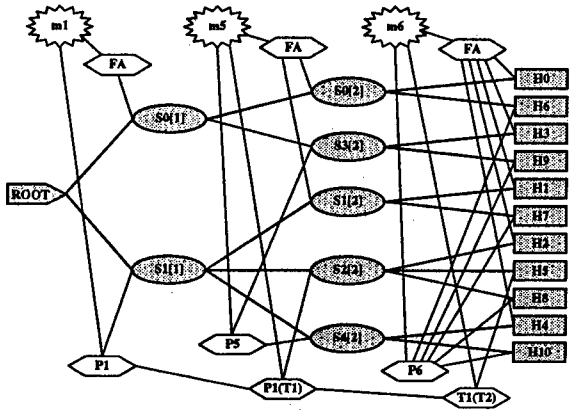


Figure 2. Typical hypothesis tree implementation

The data structure is made of three main types of data records: input data element, affectation and tree node. Each input data element record corresponds to one measurement. Each affectation data record is used to store one possibility for the interpretation of the origin of a measurement (i.e., false alarm, new target or track update). Considering Fig. 2 for example, measurement m1 is "affected" to the potential track P1 in hypothesis H7 while it is "affected" to a false alarm in hypothesis H9.

There are three kinds of tree nodes: root, sub-interpretation and hypothesis. Since a data association hypothesis is indeed a unique interpretation of the origin of each measurement, the nodes of the tree are considered to be sub-interpretations, each one being applicable to a specific measurement. Hence, a given hypothesis is represented in the data structure as one particular sequence of sub-interpretations (considered together as one possible global interpretation of the origin of all the received data) that are linked from the root node up to a special sub-interpretation node (called hypothesis) at the other end of the tree.

Each input data element record is linked to one or more affectations representing the possibilities for this measurement. Each affectation is linked to only one input data element. However, an affectation can be linked to one or more tree nodes and one or more child affectations. A child affectation is an affectation of a measurement to the update of a previously established track. This concept of parent-child affectations is used to represent the different track families of the hypothesis tree.

A level exists in the hypothesis tree for each input data element. Levels may also be created to accommodate targets whose existence is known a priori. Fake "input data elements" provided by the initialization procedure are then affected to these known targets. The level of each sub-interpretation in the tree is indicated between brackets (level 0 being the root level), and each sub-interpretation has also a number that follows the hypothesis numbering scheme up to that level (Fig. 2).

Note that the hypothesis numbering follows the scheme described in Refs. 1-2. One important aspect of the standard numbering scheme is that an internal system track, once created by the assignment of an input data element to it, can only progress towards its deletion by the track management system (as a result of a decrease in its quality, or because of the pruning of some relevant branches of the hypothesis tree). This is so because the track will keep its number (the one that has been used at its creation) only if it is not updated; the internal track number will change (thereby creating a "new" internal system track) as soon as the track is considered updated in any given hypothesis.

3.0 Width pruning

If implemented without severe limitation mechanisms, the MHT algorithm requires an ever-expanding memory as more data are received and processed. Hence, the growth of the hypothesis tree must clearly be limited for a feasible implementation on a computer (Refs. 1-2). The goal is a data

association algorithm that requires a minimum amount of computer memory and execution time while retaining nearly all the accuracy of the optimal procedure.

As discussed above, the hypotheses may be considered as branches of a tree. The hypothesis reduction techniques may thus be viewed as methods of either pruning or binding together these branches. In terms of pruning, both the width and the depth of the hypothesis tree (i.e., the number of hypotheses maintained and the number of levels in the tree respectively) can be controlled. Reference 2 describes four width-pruning approaches in details (i.e., probability, probability sum and ratio of probabilities thresholding, and fixed number).

4.0 Depth control pruning

Inherently, the MHT uses later input data to aid in evaluating difficult prior correlation decisions concerning prior input data. Hence, for each new input data element, the MHT generates soft association decisions and then waits (i.e., defers the final decision as to the right assignment) until further observations resolve the matter as best as possible. Based on this fundamental principle, one could be tempted to let the hypothesis tree grow forever (i.e., retain all hypotheses) with the conviction that the bigger the tree is (i.e., the longer the waiting period is), the better the information available is to make an educated decision on the origin of a particular input data element. Ultimately though, a final, hard decision has to be made for the system to be practical. This is the main consideration behind the depth control pruning mechanism discussed in this paper. That is, it is useless to accumulate evidences about the occurrence of an event if no decision is made about it at the end of the day.

This hard/soft decision concept leads directly to the notion of hard and soft zones in the hypothesis tree. A particular tree level is thus said to be in the "soft decision zone" of the hypothesis tree when there are multiple alternatives for the interpretation of the origin of the corresponding input data element. When there is only one option left for the explanation of the origin of an element, then the corresponding level is said to be in the "hard decision zone" of the hypothesis tree. Figure 3 is a graphical illustration of this zone concept. Note, however, that the situation depicted in Fig. 3 (i.e., hard zone on the left and soft zone on the right) is purely academic. Plausibly, since an effort is made to keep the arrival sequence of the input data elements intact in the tree, the definitive assignments and the hypothetical affectations would be mixed and spread over the entire length of the data structure in a realistic example. This has no consequence on the results.

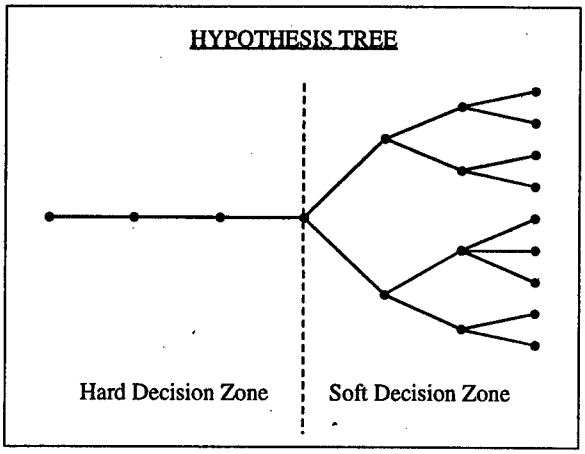


Figure 3. The hard and soft zones concept

In view of the considerations above, the depth control pruning mechanism is a set of rules used to transfer (logically only) the interpretation of the origin of an input data element from the hypothetical (or soft) decision zone of the hypothesis tree to the definitive (or hard) decision zone. One should note that, although the assignments attached to the hard decision zone are final, there is a reason to keep the input data elements, affectations and tree nodes in this zone for some time after they have been transferred by the depth control procedure. Any affectation must be kept in the hypothesis tree (whatever the zone it is in) for as long as the track it represents is still "reproductive". By definition, a reproductive track is one that can still be considered for association with new input data from the sources. Therefore, a reproductive track can eventually generate new tracks, which are considered as its "children", and the affectation matching such a track must thus be kept to ensure the consistency of the growth of the hypothesis tree.

A track that is marked for deletion by the track management logic is not considered anymore for association with new input data from the sources; such a track is, by definition, a "sterile" track. Note that a track may become sterile as its quality falls below some minimum or as a result of a pruning operation on the hypothesis tree (i.e., when the only hypotheses left in the tree are the ones where the track has already been updated with an input data element). The rule for a sterile track is that it must be kept in the hypothesis tree if it is still in the soft decision zone of the tree (since a final decision about the best interpretation of the origin of the corresponding input data element has not been made yet). Hence, an affectation (and the related input data element and tree node) can be ultimately removed from the tree if 1) it is in the hard

decision zone and 2) the corresponding track becomes sterile. In a sense, this last pruning operation is the ultimate "depth control" step limiting the size of the overall tree (i.e., not only the size of the soft zone).

The decision to transfer a tree level from the soft zone to the hard zone can be based on the monitoring of discrete or continuous parameters. For the discrete version, the waiting period for triggering the depth control pruning mechanism is determined by the number of target observation attempts made that can be used to resolve a particular assignment. That is, the waiting period is set by the observation attempt depth, not by the physical depth of the hypothesis tree.

The physical depth is useless to settle a correlation conflict between different tracks for a given input data element if none of the other data elements has something to do with the one being resolved. With a scanning sensor for example, if 10 observations were received at the end of a given scan, then the hypothesis tree will be augmented with 10 new levels. In such a case, although the tree may be considered as "deep", no truly educated decision can be made about any assignment of the 10 new observations. The 10^{th} observation of the data set doesn't tell anything about how the 1^{st} should be interpreted. And this is true of any of the 10 observations. However, if these observations were received in 10 distinct scans, then the reception of say the 6^{th} observation could help with the resolution of the assignment of say the 1^{st}. Similarly, the reception of say the 10^{th} observation could help with the decision on the explanation of the origin of say the 6^{th}. This is so because later scans constitute additional target observation attempts that have been made, each one producing some result (hit or miss), and that can thus be used to substantiate or refute prior data associations that are still considered hypothetical.

With respect to the actual implementation, a hard decision is made at one level of the tree (i.e., for the explanation of the origin of a specific input data element) when all affectations at this level have received a prescribed number of target observation attempts (a configurable parameter). When this is the case, the affectation having the highest likelihood (as determined by the sum of the likelihood of all hypotheses ensuing from this affectation) is retained as the best, final assignment for the input data element. All hypotheses linked to the other affectations of the element are then pruned from the tree. This procedure greatly reduces the number of hypotheses to be maintained.

The mechanism described above requires that a count be kept for an affectation (i.e., for the track created by the affectation) of the number of subsequent observation attempts that have been made and that are relevant to this affectation. The results of these attempts (hits or misses) are reflected in the hypothesis tree by the actual hypotheses following the affectation, and their likelihood.

Obviously, the higher the number of sources reporting on a target is (i.e., the higher the observation attempt rate is), the faster hard decisions can be made on the affectations. This is an immediate benefit of sensor data fusion.

5.0 Target observation attempts

The notion of "target observation attempts" is at the heart of any track management system and it is also the key concept behind the depth control pruning mechanism. A target observation attempt is defined as an opportunity to acquire information for the maintenance of a track on a hypothesized target entity. The occurrence, the duration, the quality and the result of a target observation attempt are important concepts that are discussed next.

5.1 Observation attempt occurrence

Some logic must determine when target observation attempts will occur (or should have occurred). This is a very important issue that has multiple facets: the use of scanning type sensors (some potentially reporting data based on a spatial decomposition into sectors), the use of multiple sensors, the use of agile beam sensors (e.g., electronically scanned antennas), etc. Taking into account all of the aspects above, a mechanism is required to determine the observation opportunities for each individual track with respect to each individual source. A very accurate model could quickly become very complex. Note, however, that the complexity of this model must not be greater than the one of the MHT implementation that one is trying to simplify.

5.2 Observation attempt duration

Very often, there is a significant duration associated with any target observation attempt when using a scanning type of sensor. This time interval results from the uncertainty on the estimated kinematics properties of the targets. The scanning sensor must sweep the totality of the area of uncertainty for a target before an observation attempt occurrence is declared for this target.

5.3 Observation attempt quality

It is very important to assess the quality of an observation attempt in order to derive a meaningful interpretation of the result of this attempt. For example, one *should not be* surprised when a particular target *is not detected* if the observation conditions for

this target are really *bad*. Similarly, if the observation conditions for a particular target are really *bad*, then one *should be* surprised if this target *is actually detected*; the received input data element is probably a false alarm in this case. Finally, if the observation conditions for a particular target are really *good*, then one should question the existence of a hypothetical target if this target *is not detected*.

Factors typically taken into account in the evaluation of the quality of observation attempts include:

Sensor-Target Geometrical Factors: A target may be momentarily obscured by terrain obstacles (e.g., the earth curvature, a mountain, etc.) or it may have left the coverage of the sensor (e.g., the elevation coverage). If a sector based report grouping mechanism is used, it may happen that a target is not in the current sector of interest (e.g., the target may have already been observed in a previous sector, or it may eventually be observed in a subsequent sector). These factors have an impact on the probability of detection value (P_D) and the density of new objects per attempt per unit of volume (i.e., β_{NT} and β_{FT}). Note that the uncertainty on the estimated kinematics properties of a target must be taken into account with the geometrical factors.

Sensor System and Environmental Factors: Sensor configuration parameters (transmitter power, scan mode, blind zones, etc.) and environmental conditions (e.g., sea state, rain, etc.) affect sensor performance (i.e., P_D).

Track Duration/Length Factors: A target may have left the coverage of the sensor (e.g., a target with a radial outbound flight profile) or may have disappeared (e.g., the target has been destroyed).

Hence, for each observation attempt that is made, some process must determine if the attempt is a good one or not. Note that the concepts of occurrence and quality of observation attempts are tightly coupled. Should the occurrence of an attempt with a null quality still be considered an occurrence? Once again, the complexity of the quality model must not be greater than the one of the MHT implementation that one is trying to improve.

5.4 Observation attempt result

Basically, there are two possibilities for the result of an attempt:

No Detection: The observation attempt has been unsuccessful. This is called a "missed observation attempt", or, more simply, a miss.

Sensor Data Available: The sensor has provided some data as a result of the attempt. Generally however, there is ambiguity as to the origin of the sensor data provided. An input data element may originate from a target that was already known and monitored (the element could thus be used to update the corresponding track), or it may originate from a new object (i.e., a new target previously undetected or a false alarm).

In any case, one has to assess the result taking into account the quality of the attempt that has been made.

6.0 Depth control simulation results

Two simulation examples have been produced, using the CASE_ATTI (Concept Analysis and Simulation Environment for Automatic Target Tracking and Identification) test bed developed at Defence Research Establishment Valcartier (DREV), to illustrate the behavior of the depth control pruning mechanism. This test bed provides the algorithm-level test and replacement capability required to study and compare the technical feasibility, applicability and performance of advanced, state-of-the-art sensor fusion techniques (Ref. 9).

6.1 First example: depth control impact

The first example has been designed to illustrate the impact of the depth control pruning mechanism on the computer resources requirements for the MHT. A very simple target-tracking scenario has been defined for this example. The scenario features two targets that appear one after another to illustrate the growth and decay of the hypothesis tree. The first target appears after 50 s of simulation. Its initial position is x = -31.25 km, y = 25 km from the origin, at an altitude of 1 km. The target then travels along the x-axis (positive direction) at a constant speed of 250 m/s for 100 s. It then disappears from the simulation. After another 50 s without a real target, the second target appears at t = 200 s, x = 6.25 km, y = 25 km, at an altitude of 1 km. It also travels along the x-axis (positive direction) at a constant speed of 250 m/s for 100 s. The second target then disappears from the simulation. After another 50 s without a real target, the scenario ends at t = 350 s.

During the whole scenario (i.e., from 0 to 350 s), a simulated scanning sensor, located at the origin, samples the environment at a rate of 60 RPM. This is an "academic" type of sensor simulation where the probability of detection for the targets has been set to a constant value of 1. Hence, when a target is present, it is detected and a measurement of the target position is produced once every second. The simulated sensor also generates false measurements, uniformly distributed in the overall coverage of the sensor, at an average rate of one per scan.

The resulting simulated data, shown in Fig. 4, was used twice to feed a target tracking system running the MHT. The track confirmation logic was set to three

hits out of five attempts, while the track deletion criteria was set to 10 s without a track update. In the first run, the Depth Control Metric Threshold (DCMT) of the depth control pruning mechanism was set to 5 observation attempts. In the second run, the same threshold was set to 1. In both cases, the tracking system successfully formed a firm, accurate track on each of the two targets, without generating any false track. However, the resources required by the MHT were not the same for the two runs.

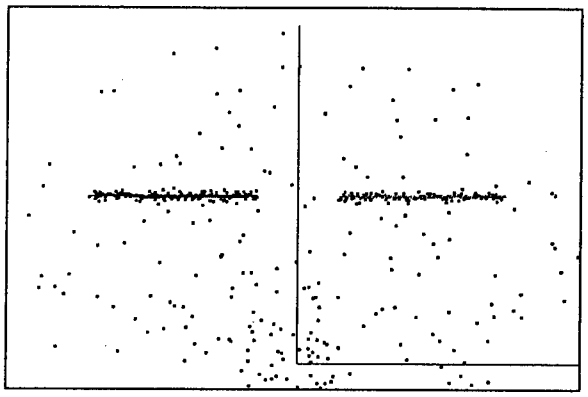


Figure 4. Simulated data for the first example

Three parameters, i.e., the depth of the hypothesis tree, the number of hypotheses maintained and the number of internal system tracks stored in the track database, were monitored for each run. Figure 5 shows the evolution of the depth of the hypothesis tree during the first run. Both the depth of the hard decision zone alone and the total depth of the tree (i.e., hard and soft zones) are shown. One can clearly identify on the graph the two time intervals where the real targets were present in the scenario (i.e., [50, 150] and [200, 300]). The number of tree levels augmented significantly during these intervals when the tracking system was not fed with false measurements alone. The maximum number of tree levels attained during the run was 16 (the minimum was obviously 1), while the average number of levels was 8.44.

Note that the number of levels in the hard decision zone was exactly 1 when a real target was present (indeed, it took 5 s after the appearance of the target to attain 1, and it took 10 s after its disappearance to go back to 0), while it is 0 when the tracking system processes only false measurements. This is in line with the fact that, in the MHT implementation used, when a false alarm affectation is selected (by the hard decision procedure) as the most likely interpretation of an input data element, then it is immediately removed from the tree instead of being transferred into the hard zone. It is also in line with the fact that an affectation to the real target becomes sterile as soon as all the branches of the tree where it could have a child (i.e., the subsequent branches with false alarms or assignments to other tracks) are pruned. Such an affectation is thus also removed from the tree as soon as it is transferred into the hard zone.

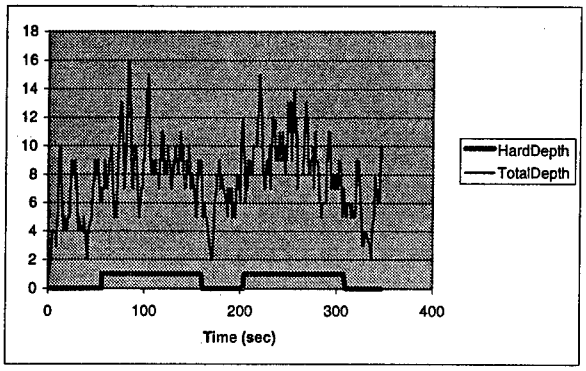


Figure 5. Tree depth (DCMT = 5)

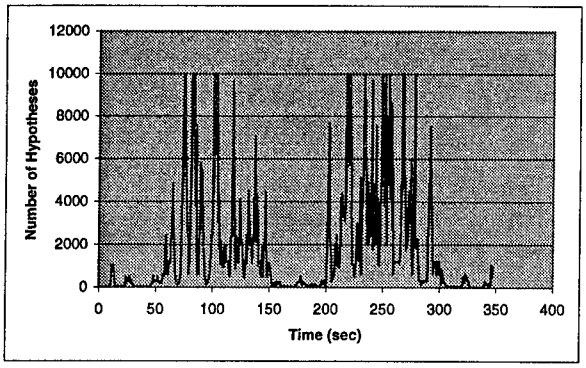


Figure 6. Hypotheses maintained (DCMT = 5)

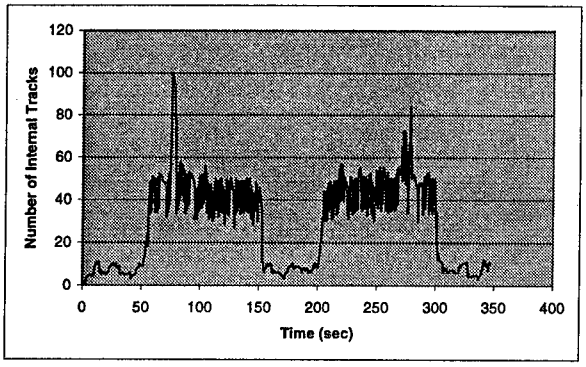


Figure 7. Tracks stored (DCMT = 5)

Figure 6 shows the evolution of the number of hypotheses maintained in the tree during the first run. Again, one can clearly identify on the graph the two time intervals where the real targets were present. The maximum number of hypotheses allowed in the tree (a configurable saturation parameter of the MHT) was set to 10,000. This maximum was reached a number of times during the intervals where the real targets were present. Note that the maximum allowed could have

been set to a much lower value without degrading the tracking performance. However, we wanted to illustrate how an uncontrolled MHT can be resource demanding. Hence, the maximum number of hypotheses attained during the run was 10,000 (the minimum was obviously 2), while the average number was around 2400. Note that during the portions of the simulation without a target, the number of hypotheses maintained was always a power of 2 (e.g., 16, 64, 1024, etc.), which was not the case in the other segments.

Finally, Fig. 7 shows the progress of the number of internal system tracks stored during the first run. The maximum number of tracks was 100 (the minimum was obviously 1, a new potential track), while the average number was around 32. During the intervals where the real targets were present, the average number of tracks was around 40.

Figures 8 to 10 show the evolution of the same three parameters for the second run, with the DCMT set to 1 observation attempt. One can without a doubt see that the depth control pruning procedure greatly reduces the computer resources requirements for the MHT; the size of the hypothesis tree maintained has been significantly reduced. Figure 8 shows the progress of the depth of the hypothesis tree during the second run. It is not as easy to identify on the graph the two time intervals where the real targets were present. The maximum number of tree levels attained was 8 (again the minimum was obviously 1), while the average number of levels was 3.33. Note that the number of levels in the hard decision zone of the tree fluctuated more than in the first run, reaching a peak value of 3 while maintaining an average value close to 1.

The number of hypotheses maintained (Fig. 9) has also been radically reduced, and no saturation condition was observed. The maximum number of hypotheses maintained during the second run was 96 (again the minimum was obviously 2), while the average number was around 10. Note that during the portions of the simulation without a target, the number of hypotheses maintained was not always a power of 2, reflecting the higher difficulty of the MHT to maintain the data association accuracy. Finally, Fig. 10 shows the progress of the number of internal system tracks stored during the second run. The maximum number of tracks was 11 (the minimum was obviously 1, a new potential track), while the average number was around 5. During the intervals where the real targets were present, the average number of tracks was around 6.

As a final remark for the first example, note that the edge of the transitions from one interval to the other (target, no target) were not as sharp in the second run with the DCMT set to 1 as they were in the first run with the DCMT set to 5. In a sense, this reflects the association discrimination power of the MHT when it is allowed to keep more information to make the final decision.

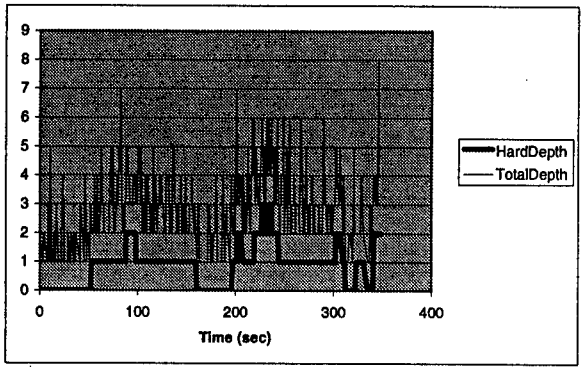


Figure 8. Tree depth (DCMT = 1)

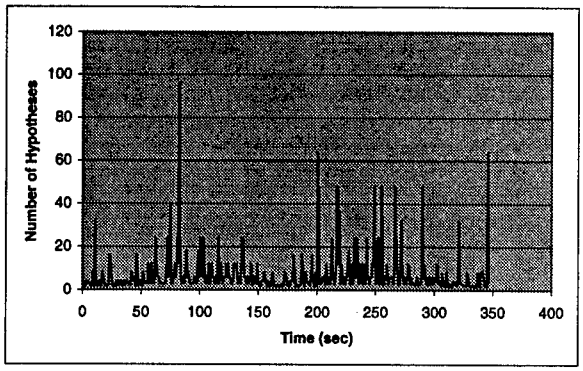


Figure 9. Hypotheses maintained (DCMT = 1)

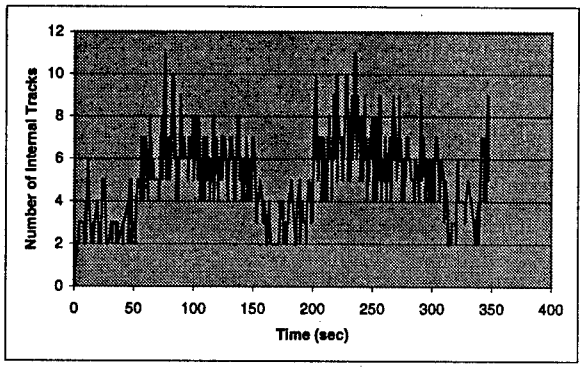


Figure 10. Tracks stored (DCMT = 1)

6.2 Second example: optimal DCMT setting

The second example has been designed to illustrate the trade-off between the data association accuracy and the computer resources requirements for the MHT. Again, a very simple target-tracking scenario was defined for this example. The scenario features two closely spaced targets flying in parallel (with a separation of 600 m), along the x-axis (positive

direction), at about 25 km from the sensor. The two targets were observed during 100 s by a simulated scanning sensor located at the origin and having a scan rate of 60 RPM. The probability of detection was set to 0.8. The standard deviations of the measurement errors for the simulated sensor were 500 m in range and 0.02 radian in bearing. No false alarms were generated.

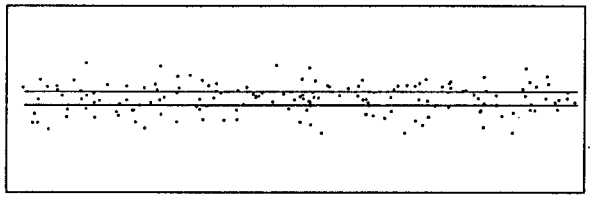


Figure 11. Simulated data for the second example

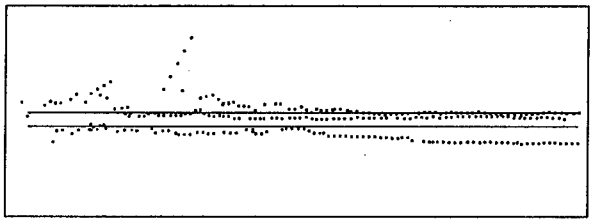


Figure 12. Tracking results: nearest neighbor

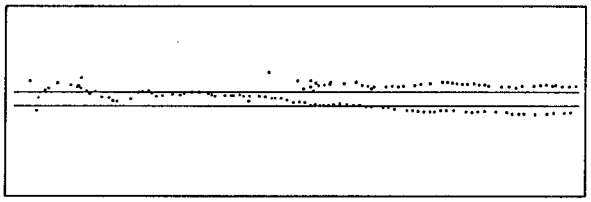


Figure 13. Tracking results: MHT with DCMT = 1

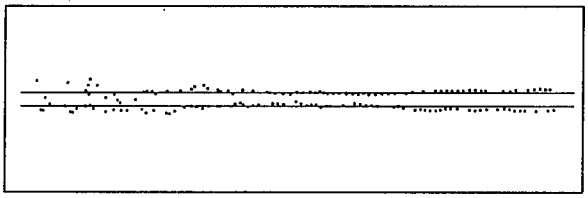


Figure 14. Tracking results: MHT with DCMT = 5

The resulting simulated data shown in Fig. 11 were used three times to feed a target tracking system running a nearest-neighbor (NN) data association algorithm for the first run, and the MHT for the last two runs (with the DCMT set to 1 and 5 respectively). The track confirmation logic was set to three hits out of five attempts, while the track deletion criteria was set to 10 s without a track update.

Figures 12 to 14 show the tracking results for the three runs. One can see that the tracking system using the NN algorithm (JVC technique) had a hard time tracking the two targets (Fig. 12). Three firm tracks were established on the two targets. In the first half of the run, the tracks were very unstable. During the last portion of the run, two tracks followed the same target, while the third one diverged from the other target.

Figure 13 shows the tracking results for the second run, with the DCMT set to 1 for the MHT. This time, two tracks were established. However, one of the tracks was only confirmed after about 50 s of simulation, while the other exhibited a track seduction behavior (i.e., the track initially followed one target, then the other target, then the first target again, etc.). The maximum number of tree levels attained during this run was 4 while the average number of levels was around 3. The maximum number of hypotheses attained was 20 while the average number was around 9. The maximum number of internal tracks was 14 while the average number was around 10.

Finally, Fig. 14 shows the tracking results for the third run, with the DCMT set to 5 for the MHT. In this case, the tracking system successfully tracked the two targets for the whole duration of the run, without generating any false track. The maximum number of tree levels attained during the third run was 9 while the average number of levels was 6.6. The maximum number of hypotheses attained was 100 (i.e., the saturation condition set for this run) while the average number was around 58. The maximum number of tracks was 166 while the average number was around 100.

This example clearly demonstrated that there is a trade-off between the data association accuracy (and consequently the tracking stability and accuracy) and the computer resources requirements of the MHT. An optimal setting for the DCMT parameter remains to be found that would result in a balance between tracking performance and resources utilization.

7.0 Conclusion

To reduce the computational requirements of the MHT data association algorithm, the number of hypotheses must be limited through the use of pruning methods. This paper presented a depth control pruning mechanism making hard decisions on the origin of input data elements contained in the hypothesis tree. It is used to transfer the assignment of an input data element from the soft decision zone of the hypothesis tree to the hard decision zone. The waiting period is determined by the number of target observation attempts made that can be used to resolve a particular assignment. Obviously, the higher the number of sources reporting on a target is (i.e., the higher the observation attempt rate is), the faster hard decisions can be made on the affectations. This is an immediate benefit of sensor data fusion. The occurrence, the

duration, the quality and the result of a target observation attempt are concepts that were discussed in the paper. A model is required to determine the observation opportunities for each individual track with respect to each individual source, and to evaluate the quality of the attempts. However, a very accurate model could quickly become very complex. Clearly, the complexity of this model must not be greater than the one of the MHT implementation that one is trying to improve.

Some depth control pruning simulation results were presented. Two simulation examples have been produced using the CASE_ATTI test bed developed at DREV. The first example has been designed to illustrate the impact of the depth control pruning mechanism on the computer resources requirements for the MHT. Results showed without a doubt that it is possible to greatly reduce the computer resources requirements for the MHT with the depth control pruning procedure while, in some conditions, keeping the accuracy of the tracking process. In particular, it is manifest that if the depth of the tree is well controlled, then the width pruning mechanisms may never have to be used (i.e., the saturation conditions may potentially never be met).

However, the second simulation example presented showed that a trade-off between the data association accuracy (and consequently the tracking stability and accuracy) and the computer resources requirements of the MHT has to be made when the situation portrays high potential for correlation ambiguities.

Further work is required to better characterize the depth control pruning mechanism in order to find the optimal depth of observation attempts that would maximize the data association accuracy and minimize the resources requirements. The option to adaptively select the optimal depth for a given environment must also be investigated.

8.0 References

1. Reid, D.B., "An Algorithm for Tracking Multiple Targets", IEEE Transactions on Automatic Control, Vol. AC-24, pp. 843-854, December 1979.
2. Blackman, S.S., "Multiple-Target Tracking with Radar Applications," Artech House, Norwood, MA, 1986.
3. Demos, G.C., Ribas, R.A., Broida, T.J. and Blackman, S.S., "Applications of MHT to Dim Moving Targets", SPIE Proc., Vol. 1305, Signal and Data Processing of Small Targets, 1990.
4. Werthmann, J.R., "Step-By-Step Description of a Computationally Efficient Version of Multiple Hypothesis Tracking", SPIE Proc., Vol. 1698, Signal and Data Processing of Small Targets, 1992.
5. Blackman, S.S., Dempster, R. and Broida, T.J., "Design and Evaluation of Multiple Hypothesis tracking for Infrared Surveillance Systems", SPIE Proc., Vol. 1698, Signal and Data Processing of Small Targets, 1992.
6. Chan, D.S.K., Harrison, D.D. and Langan, D.A., "Tracking in a High-Clutter Environment: Simulation Results Characterizing a Bilevel MHT Algorithm", SPIE Proc., Vol. 1954, Signal and Data Processing of Small Targets, 1993.
7. Chan, D.S.K. and Langan, D.A., "Performance Results of the Bi-Level Multiple Hypothesis Tracking (MHT) Algorithm for Two Crossing Targets in a High-Clutter Environment", SPIE Proc., Vol. 2235, Signal and Data Processing of Small Targets, 1994.
8. Blackman, S.S., Dempster, B.J., Tucker, G.K. and Roszkowski, S.H., "Application of Multiple-Hypothesis Tracking to Shipboard IRST Tracking", SPIE Proc., Vol. 2759, Signal and Data Processing of Small Targets, 1996.
9. Roy, J.M.J., Bossé, É. and Dion, D., "CASE_ATTI: An Algorithm-Level Testbed for Multi-Sensor Data Fusion", DREV Report No. 9411, May 1995.

Efficient Multidimensional Data Association for Multisensor-Multitarget Tracking Using Clustering and Assignment Algorithms*

M. R. Chummun, T. Kirubarajan, K. R. Pattipati and Y. Bar-Shalom
Electrical and Systems Engineering Department
University of Connecticut
Storrs CT 06269-2157, USA
ybs@ee.uconn.edu

Abstract *In this paper we present a fast multidimensional data association technique based on clustering and assignment algorithms for multisensor-multitarget tracking. An important part of a multisensor-multitarget tracking algorithm is data association and assignment-based methods have been shown to be very effective for this purpose. In assignment, candidate assignment tree building consumes 95%–99% of the CPU time. In this paper, we present the development of a fast data association algorithm, which partitions the problem into smaller sub-problems, resulting in significant computational savings. This hierarchical clustering algorithm is illustrated on 2- and 3-dimensional full-position measurements (active sensors) and on angle-only measurements (passive sensors). Simulation results show that the computational load can be reduced by 20–80 times, depending on sensor type and problem sparsity, over the standard multidimensional assignment approach without clustering.*

Keywords: Multitarget tracking, data association, multidimensional assignment algorithms, clustering, angle-only tracking.

1 Introduction

The problem of data association, that is, deciding which measurement came from which target in a multitarget tracking problem in the presence of false alarms and missed detections, has been studied extensively. Some of the proposed solutions to this complex problem include the Nearest Neighbor algorithm, Probabilistic Data Association (PDA), Multiple Hypothesis Tracking (MHT) and assignment [2]. These algorithms vary widely in their complexity and the resulting performance.

Data association using a multidimensional algorithm, where the measurements in the last S scans are associated with the list of tracks (S lists — S-dimensional association, denoted as S-D), has been shown to be a practical and feasible alternative to MHT *without* the exhaustive enumeration. In assignment, the association between the lists of measurements and the list of tracks is formulated as a global discrete optimization problem, subject to certain constraints, where the objective is to minimize the overall cost of association. While finding the optimal assignment is an NP-hard problem for $S > 2$, a number of near-optimal modifications with (pseudo-)polynomial complexity have been proposed [3].

The major challenge to overcome in the S-D assignment for tracking is that of solving the ensuing NP-hard multidimensional assignment problem. In particular, an algorithm that determines the optimal solution is not only arduous, but also impractical for even fairly small-sized problems. A multistage Lagrangian re-

*Supported by ONR Grant N00014-97-1-0502 and AFOSR Grant 49620-97-1-0198.

laxation approach can be used to solve the S-D assignment problem as a series of 2-D assignment problems, which are solvable in (pseudo-) polynomial time and is thus fast. However, for fairly large scenarios with many sensors and hundreds of targets, this too can become inefficient. S-D assignment tree was also used in passive (angle-only) multisensor-multitarget direction finding problems [3, 5], where candidate tree building took about 99% of the time. This provides the motivation for finding ways to build the candidate assignment tree more efficiently.

In this paper, we present an efficient technique based on clustering to significantly reduce the CPU time of the S-D assignment algorithm by partitioning the assignment problem into smaller subproblems. A clustering technique is used to screen for improbable candidate associations and reject them while forming the candidate assignment tree resulting in a smaller candidate tree. Solutions based on the clustering approach are developed for different types of sensors, namely, for passive (angle-only) and active (full position) sensors in two and three dimensional space — different sensor configurations require different metrics for forming the clusters.

This paper is organized as follows. In Section 2, the data association problem via multidimensional assignment is discussed. In Section 3, the hierarchical clustering approach for active and passive sensors is presented. Simulation results for different target-sensor configurations are given in Section 4.

2 Assignment Algorithm

In a multisensor-multitarget scenario, we have an unknown number of targets, which can be either mobile or stationary, in a surveillance region and a known number of sensors observing different areas of this region at different times. The sensors, which typically have nonunity detection probabilities, can be on moving platforms or fixed. In either case, it is assumed that their locations are known at any time. The sensors can be either active, i.e., full position measurements (polar or Cartesian) are available or passive, i.e., angle-only or line-of-sight (LOS) measurements are available. For data association with S-D assignment problems, S synchronized scans (or frames or lists) of measurements are used in the static case, while for dynamic problems $(S-1)$ consecutive (most recent) scans of measurements are used together with the list of tracks — in both cases the association is among S lists of data.

The goal is to associate the measurements and estimate the target states in terms of position and possibly velocity and acceleration. In S-D assignment, the data association is formulated as an optimization problem where the objective is to minimize the total cost of associating the measurements subject to certain feasibility constraints. The cost of each candidate association is usually calculated based on the state-to-measurement relationship with the help of a state estimator [2]. The S-D assignment algorithm finds the most likely set of S-tuples such that each measurement is assigned to at most one track (target), or declared false, and each track receives at most one measurement from each list. When a track is not associated with any of the received measurements in a scan, it is said to have been associated with the "dummy" measurement. For example, when a candidate association does not contain any measurement from a scan (list), it is associated with the "dummy" measurement from the list.

Though finding the best candidate assignment is the most important phase of data association, it typically takes only about 5% of the CPU time when compared with the time taken for forming the candidate assignment tree. For example in [3] and [5], where multisensor-multitarget angle-only tracking was considered (using synchronized frame from several sensors[1]) the cost of each candidate associa-

[1]This is a static association resulting in full position "composite measurements", which then have to be associated across time by the dynamic associator/estimator. This static association problem is the most difficult one because its dimension is the number of sensors (7 in

tion is obtained by solving a hypothesis testing problem and a generalized likelihood ratio is derived.

Most of the computing time is spent on maximizing the negative log-likelihood function optimized via a Conjugate Gradient algorithm, such as the Fletcher-Reeves or Polak-Ribiere-Polyak Algorithm, which takes 95% of the computation time. Note that in order to evaluate the total cost of association, one needs to evaluate the negative log-likelihood ratio and then maximize it for every possible candidate association, i.e., all combinations of measurements. Thus if we have a scenario with 3 lists with, say, 10 measurements each (including a dummy measurement to handle a missed detection), we need to compute 1000 (10^3) possible candidate associations and hence 1000 cost evaluations (1000 maximizations of the log-likelihood functions). We can limit the number of Conjugate Gradient maximizations, and thus reduce the time complexity of the whole algorithm, by removing unlikely candidate associations using a gating method, for example. A chi-square validation test is used to reject candidate associations that fall outside the validation gate. This test rejects those combinations which give goodness of fit inconsistent with the noise covariances (acceptance interval space) [2]. In this paper, the idea of pruning the candidate association tree is taken a step further. A clustering algorithm is used to select only those candidate associations that are most likely to be matched together using a Euclidean distance criterion.

3 Clustering Algorithm

The clustering technique used to prune the number of candidate associations is a hierarchical algorithm which groups the measurements according to two parameters: a distance metric and the sensor-origin of the measurement. The distance metric, which can be the Euclidean

[5]) while the dynamic one has, typically, a much lower dimension because time depth beyond 3 scans yields negligible marginal returns

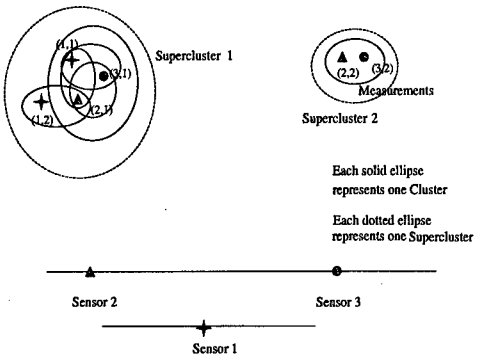


Figure 1: Cluster formation with multisensor-multitarget data

distance between two points or two angles measurements, decides which set of measurements should be in the same cluster.

Our definition of a cluster is the largest set of measurements, each coming from a different sensor, which satisfy the distance criterion — a cluster from S lists may contain fewer than S measurements. Each cluster is a candidate association with the largest number of received measurements. To reduce spurious clusters, a cluster is required to have a minimum cardinality, which can be defined by the user. With this definition of a cluster, note that measurements from a target (obtained with different sensors) may be placed in different clusters. Also, a measurement can be placed in different clusters — multiple candidate associations per measurement. In order to handle this, clusters are merged into superclusters when measurements in different clusters satisfy the distance criterion. Thus, a supercluster is the largest set of clusters with possible cross-associations across clusters. Now the measurements can be partitioned into a number of superclusters and the corresponding candidate assignment trees (smaller in size than the complete assignment tree) within each supercluster can be formed. Also note that measurements in a cluster may have come from different targets. This is handled automatically when the candidate tree is formed within a supercluster — all possible[2]

[2]Note that each candidate association has to be part of a cluster. However, even with a cluster the candidate

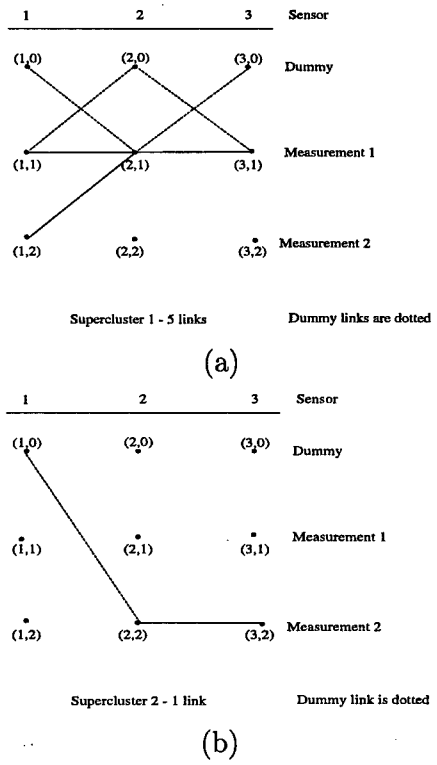


Figure 2: Candidate association tree with clustering corresponding to (a) supercluster 1 (b) supercluster 2

candidate associations within a supercluster are considered. Instead of a full assignment tree consisting of all the measurements from all the sensors as in the case without clustering, one ends up with a number of smaller, sparser trees.

To illustrate the clustering approach, consider a scenario with 3 sensors illustrated in Figure 1 where the measurements (full position in this case) from different sensors are denoted by different symbols. There are 6 measurements in total, two from each sensor, but spread over the surveillance space of interest. If we require at least two measurements to identify a target, then we have 6 possible candidate sets represented by the cluster ellipses in the figure. Note that 5 clusters contain two measurements (plus a dummy accounting for a possible miss in each case) and one cluster contains 3 measurements — the measurements (1,1),

association is not full.

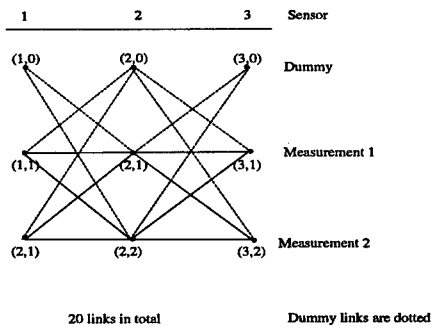


Figure 3: Full candidate association tree (without clustering)

(2,1) and (3,1)[3]. The measurements (1,2),(2,1) and (3,1) and cannot form a 3-measurement cluster because measurements (1,2) and (3,1) are far from each other (the distance between (1,2) and (3,1) is greater than the Euclidean decision threshold). Even though measurements (1,1) and (1,2) are close to each other, they do not form a cluster because they originate from the same sensor.

The candidate association trees corresponding to superclusters 1 and 2 are shown in Figure 2. The full candidate association tree without clustering for the same scenario is shown in Figure 3. It can be seen that with clustering the number of possible candidate associations is only 6 whereas without clustering it would have been 20. Thus the total number of cost computations is only 6 and so is the number of expensive log-likelihood maximizations. This saving increases substantially as the number of sensors and targets increases. Although, the savings depend on the sparsity of the scenario, the complexity can never be greater than the original case.

3.1 Clustering Solutions

In the following section the above algorithm is applied to different scenarios. The algorithm is described for both passive (angle-only) and active (full position) cases (2-d and 3-d in both

[3]The pair (i, j) denotes measurement j from sensor i.

cases[4]). The measurements used can be either in polar or Cartesian coordinates. Different clustering techniques are required for these cases.

3.1.1 Active Sensors

An active scenario is one in which the sensors measure all the coordinates of position, typically in a 2-d scenario or a 3-d space. This is the easiest case to which the clustering technique can be applied. The algorithm reduces to clustering points in 2-d or 3-d space. To do so, any one of the many existing clustering algorithms can be used [4]. The algorithm proposed in [6] divides the space of interest iteratively into smaller sections, until each point in the space is contained in one section. During the division, the neighbors of the resulting segments are formed. This algorithm has complexity $O(n \log n)$. In our case, however, in the formation of the clusters, we need to take care of the fact that at most only one measurement from a particular sensor can be placed in the same cluster.

3.1.2 Passive Sensors

A passive scenario is one in which only partial observations of a target position are available. That is, we might know only the direction in which the target lies, but not its position. In this case, it is more difficult to apply clustering techniques. We will consider the 2-d and 3-d cases separately.

2-d scenarios. In this case, we have only one scalar for each measurement (since two scalars define the full position). Each target is defined as lying along a line in a certain direction. Note that in this case, given two sensors, we can find the potential position of the target by computing the point of intersection of any two LOS measurements (two lines) from different sensors. However, any two points of intersection are equally probable to be a valid target — this results in the well-known ghosting problem (see, e.g., [2]). However, a third (or fourth, fifth, etc.) sensor will reduce the number of probable targets if we consider the intersection points of measurements from the third sensor and the first two sensors. Thus the whole problem is reduced to first computing the points of intersection between the LOS measurements for any two sensors and then clustering the intersections. LOS measurements relating to the same target from different sensors will intersect in the same region (ideally at the same point in a no-measurement-noise scenario). Thus their points of intersection will lie in a close area. By first computing these points and then clustering them, we can identify the LOS measurements that potentially come from the same target.

However, the number of possible target positions in this case is far larger than in the active case. If we have N sensors, for example, and each has m measurements, then the number of possible target positions can be m^N, assuming no missed detections. In case we have missed detections and a target can be identified by fewer measurements than there are lists, we will have even more candidate sets.

Another problem is that since each angle measurement is with respect to the corresponding sensor position, the density of lines close to the sensor is larger than that far from the sensor. Thus the possible intersection points near the sensor locations are relatively closer to each other than those far from the sensor. A solution to this problem is to use a variable cluster size in the clustering algorithm. A small gate is used closer to the sensor locations and the size is increased as we move away from the sensors. The size of this gate is inversely proportional to the proximity of the sensors. A heuristic function to define the size of such a variable gate is an absolute logarithmic function, defining a wedge-type function. Note that if the sensors are spread around the targets, then the gate size needed will be almost the

[4]To avoid ambiguity 2-d and 3-d indicate the dimension of the physical space in which the targets are. In contrast, the S-D assignment is of dimension S since it associates elements from S lists

same everywhere in the space of interest. We can still, however, define an approximate function for the gate size, to fit each scenario by contour mapping the sensors.

3-d scenarios. In a 3-d passive case, we assume that we have at least two parameters (out of three, which define the exact position) from a target — typically two angles that define the LOS. Thus each measurement again defines the target as lying along a line. The problem with such a case is that finding the points of intersection between the measurements from different sensor lists does not help much. Due to noise, the measured LOS lines are off the true LOS and thus may never intersect. In this case, the clustering algorithm is modified as follows.

Instead of clustering the measurements directly, we do so indirectly. The dihedral angle of a plane is defined as the angle between this plane and a reference plane. Let us consider a scenario with two sensors with two LOS measurements for two different targets each. A plane, called the target plane, can be passed through the LOS measurement line and the line containing the two sensors. This plane is unique for each measurement. Now, the angle between this plane and the ground defines a unique dihedral angle, α_1. However, the dihedral angle between one of the LOS measurements of the other sensor will lie in the proximity of α_1 since this LOS measurement also defines the same target. As a result, clustering the dihedral angles leads to clustering the respective LOS measurements.

In the case where we have more than two sensors, the dihedral angles can be computed pairwise between any two sensors. The cost of computation increases, but we also get the added advantage that cross-associations of pairs of dihedral angles improves the accuracy of the clustering algorithm.

3.1.3 Polar Coordinate Systems

The clustering algorithm described in [6] is an algorithm whereby points are clustered in the rectangular (Cartesian) coordinate system, by dividing the space of interest iteratively and keeping a list of neighbors of each subspace formed. A similar algorithm is used in the above case. It can be suitably modified to cluster points in polar coordinates too — this avoids the conversion of polar measurements to Cartesian. In this case, instead of breaking the space into rectangular boxes, we can divide the space into cones and sections of these. The respective sub-regions are thus delimited by the arcs and radii.

4 Results

The effectiveness of the clustering algorithm approach combined with multidimensional assignment is demonstrated on a number of scenarios with different number of targets and sensors. In all cases, significant improvement in CPU times is noticed. CPU times are obtained for the problems published in [3] and [5].

4.1 Active Sensors

A 3-dimensional scenario in polar coordinates with 3 sensors and a variable number of targets is used to compare the performance with and without clustering. The measurement noise standard deviations were, for the elevation, $\sigma_\theta = 0.1$ rad, and for the range, $\sigma_r = 8$ m. The probability of detection $P_D = 0.9$ and false alarm density $P_{FA} = 0.0005$, which gave about 2 false alarms for a batch of 15 targets.

The CPU times are presented in Table 1 together with the improvement factors with clustering. From the table, we note that as the number of targets is increased, the improvement factor is higher. This is because the algorithm without clustering has to process a larger number of potential candidate associations. This increases non-linearly as explained earlier. The computational load in the approach with clustering increases at a lower rate as a function of the number of targets. Also, the association results obtained in both cases are nearly identical. For larger numbers, some minor differences appear because at higher density, measurements from different

No. of Targets	CPU times (clustering)	CPU times (w/o clustering)
10	0.05	1.05
15	0.09	2.97
20	0.19	6.90
25	0.34	13.68
50	1.16	71.68
100	6.78	550.28

Table 1: CPU times for an active scenario (3-sensors) in 3-d space

No. of Targets	CPU times (clustering)	CPU times (w/o clustering)
10	0.06	1.04
15	0.11	3.20
20	0.20	5.90
25	0.35	11.65
50	1.86	123.24
100	8.25	564.24

Table 2: CPU times for a passive scenario (3 sensors) in 2-d space

targets end up in the same cluster. The cluster distance parameter can be varied to control this effect.

4.2 Passive Sensors

The 2-dimensional passive scenario is presented in greater detail because it is a more complex (and common) situation. First, the number of targets is varied keeping the number of sensors fixed, and then the number of sensors is varied keeping the number of targets fixed.

The sensors measure only the azimuth of the target from the corresponding sensor. The standard deviations of the sensors used are $\sigma_\theta = 0.1$ rad, with a detection probability, $P_D = 0.95$ to 0.98 and false alarm density $P_{FA} = 0.0001$. The number of targets is from 10 to 100 and their positions are randomly placed in the space of surveillance. The CPU times are listed in Table 2 for different number of targets.

Now consider the case where the number of sensors is varied while the number of targets is kept constant at 15. The sensors are the arranged along perimeter of a circle, with radius 2 km. The CPU times are listed in Table 3 for different number of sensors and 15 targets.

No. of Sensors	CPU times (clustering)	CPU times (w/o clustering)
3	0.15	8.15
4	0.94	28.76
6	5.85	125.92
8	17.39	320.84

Table 3: CPU times a for passive scenario (2-d) with different number of sensors and 15 targets.

The observations noted for the active case in Section 4.1 apply here as well. In addition, we note that the average time taken in the active case with (without) clustering is 0.09 (2.97) vs. the passive case of 0.15 (8.15). This is because, in the passive scenario, one has to process a larger number of possible candidate solutions because of the availability of only partial measurements. Also, as the number of passive sensors increase, the number of possible candidate associations increase, and there is a decrease in the improvement factor. For 3-d passive sensors, similar observations are made.

4.3 A Dynamic Problem

An m-best S-D Data association algorithm to solve a dynamic (moving target) problem was considered in [5]. Here we make a comparison to the CPU times obtained in [5] without clustering and demonstrate the advantages of our clustering technique. In [5] the dynamic estimation problem is preceded by solving a sequence of static problems at different points in time. The m-best S-D algorithm solves each static problem separately as an S-D problem, and finds the m-best assignment solutions. Using a JPDA-like technique, a probability of being correct is assigned to each solution (which consists of full position "composite" measurements). This information along with the m-best solutions are used with a state estimator in a dynamic 2-D assignment algorithm to estimate the states of the moving targets over time.

In [5] this algorithm was applied to a 7-sensor, 5-target problem. The sensors record scans of measurements at 10 time instances. The detection probability is 0.9 and the false alarm rate is 0.8/radian. Thus with 5 targets, the average number of detection per sensor scan is 7. To identify a target, a candidate association must include at least 4 non-dummy measurements (majority vote). This scenario results in a large number of candidate associations to process. The LOS measurement error standard deviation, σ_θ, was 2.0°.

The clustering algorithm for passive cases was applied to the above mentioned problem. The clustering technique was used in forming the candidate associations prior to assignment in each of the static problems at the different time instances. An improvement of 7 times in the CPU times was noted, which represents a significant saving.

5 Conclusions

In this paper we presented an efficient approach to multidimensional data association for multisensor-multitarget tracking. Data association via multidimensional assignment is a an NP-hard problem. Even with near-optimal approximations, the computation times can be very high. This is especially true for passive target tracking problems, where, due to target state-to-measurement nonlinearities, the numerous candidate cost evaluations are rather expensive. In this case, 95%–99% of the CPU time for data association is typically spent on forming the candidate assignment tree. The clustering approach, which breaks the whole assignment problem into smaller, more manageable subproblems, reduces the time taken to form the candidate assignments via a "divide-and-conquer" technique. By using the Euclidean distance as a measure to prescreen candidate assignments, the assignment tree is made sparser. Also, by grouping the measurements into clusters, a number smaller problems are solved instead of solving one big problem .

A hierarchical clustering approach, in conjunction with an assignment algorithm, was developed for tracking multiple targets using 2-dimensional and 3-dimensional active (full position) and passive (angle-only) measurements. Simulation results on different target-sensor geometries and configurations showed significant improvement in computation time *without* altering the association or the estimation processes. Depending on the scenario, CPU time is reduced by about 20–80 times over the standard multidimensional assignment approaches without clustering.

References

[1] Bar-Shalom, Y., and Li, X. R., *Estimation and Tracking: Principles, Techniques and Software*, Boston, MA: Artech House, 1993. Reprinted by YBS Publishing, 1998.

[2] Bar-Shalom, Y., and Li, X.R., *Multitarget-Multisensor Tracking: Principles and Techniques*, Storrs (CT 06269-2157): YBS Publishing, 1995.

[3] Deb, S., Yeddanapudi, M., Pattipati, K. R., and Bar-Shalom, Y., "A Generalized S-D Assignment Algorithm for Multisensor-Multitarget State Estimation", *IEEE Trans. Aerospace and Electronic Systems*, Vol. 33, No. 2, pp. 523-538, April 1997.

[4] Jain, A. K., and Dubes, R. C., *Algorithms for Clustering Data*, Prentice Hall, 1988.

[5] Popp, R. L., Pattipati, K. R. and Bar-Shalom, Y., "m-best SD Data Association Algorithm for Multitarget Tracking", to appear in *IEEE Trans. Aerospace and Electronic Systems.*

[6] Vaidya, P. M., *An O(nlogn) Algorithm for the All-Nearest-Neighbors Problem*, Discrete and Computational Geometry, Vol. 4, pp. 101-115, 1989.

Session WA4
Biological and Linguistic Models for Fusion
Chair: George Chapline
Lawrence Livermore National Laboratory, CA, USA

Verb Disambiguation through the Fusion of Two Independent Systems

Abolfazl Fatholahzadeh
Dépt. d'informatique
École Supérieure d'Électricité
2, rue Édouard Belin, 57078 Metz, France

Sylvain Delisle
Dépt. de mathématiques et d'informatique
Université du Québec à Trois-Rivières
Trois-Rivières, Québec, Canada G9A 5H7

Abstract *This paper presents the fusion of two independent systems, namely TSGR (Target Sequence Generation by Refinement) and D&H (DIPETT & HAIKU). The former is designed for translation purposes and can quickly compute the most probable meaning, in the target language, of multi-sense verbs appearing in the same paragraph. The later is a text analysis system which performs syntactic and case-based semantic analysis. The fusion is dictated by the fact that D&H can provide the type of information needed on input by TSGR, i.e. semantic cases associated to clauses. However, because both systems use different sets of semantic cases, this integration becomes an interesting and non-trivial problem.*

Keywords: Word sense disambiguation, Case-based semantic analysis, constraint, refinement, concept coherence.

1 Introduction

The discrimination of word senses, word sense disambiguation, is of prime importance for all areas involving computerized language analysis, including corpus-based research, lexical studies, information retrieval, machine translation, natural language processing, studies of style and theme, authorship attribution, and applications such as hypertext browsing.

This paper outlines the fusion of two independent systems, namely TSGR and D&H. TSGR is designed for translation purposes and can quickly compute the most probable meaning, in the target language, of multi-sense verbs appearing in the same paragraph. However, TSGR requires that the text be hand-coded in terms of semantic (case) relationships. On the other hand, D&H is a text analysis system which performs syntactic and case-based semantic analysis and greatly facilitates the identification of semantic relationships that occur in the sentences of a text.

Since both TSGR and D&H use different sets of semantic (case) relationships and since we also want these two systems to collaborate in a coherent way, finding a fusion method for the integration of both systems is an important problem which happens frequently during the development of complex softwares. This is what we discuss in the following sections.

The rest of the paper is organized as follows. Sections 2 and 3 describe the TSGR system and the D&H system, respectively. The fusion process will be explained through the examples of section 4. We conclude in section 5 and identify further problems that we intend to investigate in the future.

2 The TSGR System

TSGR is designed to determine the exact meaning in the target language of multi-sense verbs appearing in the same paragraph. Let us take the following example, called thereafter the *Lake-Example*, taken from "The Peasant and the Watterman" [9]: "A peasant was chopping a tree in the woods by the lake. He dropped his axe and it fell with a splash into the water. Quickly he

Table 1: Several senses of the verb 'chop'

	Description	French	Turkish
1	*vi* to make a quick stroke or repeated strokes with a sharp instrument *he has been chopping in the woods for an hour.*	couper	kes
2	*vt* to cut into or sever by repeated blows of a sharp instrument *he was chopping a tree in the woods.*	couper	kes
3	*vi* to hit with a short downward stroke *he chopped with his hand.*	frapper	vur
4	*vt* to hit with a short downward stroke *he chopped the ball with the club.*	frapper	vur
5	*vt* to cut into bits, mince *she chopped the meat with a robot.*	hacher	kıy
6	*vi* to change direction *the wind is chopping about.*	changer direction	yön değiştir
7	*vt* to reduce *we chopped more than USD 1,000 off the budget.*	baisser	azalt

dove into the lake."

In this example, there are at least 1848 possible candidates to be considered as the global (paragraph) meaning.

$$7(chop) \times 11(drop) \times 8(fall) \times 3(dive) = 1848$$

The aim here is to instantiate these four verbs in a target language without loosing their right meanings. Table 1 shows several senses of the verb chop, where vt and vi stand for transitive verb and intransitive verb, respectively.

Each particular sense of a verb may have a different corresponding translation in the possible target language. Determining the correct instantiations of the verbs in the target language is carried out by TSGR (Fatholahzadeh & Güvenir [6]) which makes use of two separate methods, namely, Concept Coherence (Alterman [1]) and Refinement (Güvenir & Ernst [8]).

In the above example, verbs 'drop', 'chop', and 'hold' are concept coherent because they mutually define one another. A part of 'chopping wood' is 'holding an axe', and in order to have 'dropped something' one must first have 'held it'. The couples $(drop, chop)$ and $(chop, hold)$ are called concept coherent in the *theory of event/state concept coherence* advocated by Alterman. According to this theory, the representation of a narrative text can be generated by a process of matching text against a dictionary of concepts, which are related by a small set of relation-types, and using the organization of the concepts in the dictionary to organize the instances of event/state concepts which appear in the text. Event concept coherence is a property of the dictionary. It is determined as a function of the distance between two concepts. Two terms in the dictionary are event concept coherent if there exists a path between two concepts in the dictionary.

All knowledge about the relationships between two concepts in TSGR's dictionary is stored as a graph. The nodes of the graph represent the concepts (e.g. holding, choping, etc.), and the arcs represent the relations between concepts. Relations between nodes are stored in a quadruple, which has the following template:
[*Relation Event/State*1 *Event/State*2 (*Constraints*)]

The first argument states the kind of relationship that exits between two concepts. The second and the third arguments give the names of the two concepts being related, and the last argument is an optional list of constraints. The constraints specify the required matches between the case arguments of the concepts, as in the following relation:

```
[coor hop hold ((match AGT1 AGT2)
                (match INS1 OBJ2))].
```

Table 2: Concept coherence and its relations

Relation	Abbr.	Description
Class-subclass	sc	Property inheritance relation.
Sequence-subsequence	subseq	One event is part of another, and it occurs for a subinterval time.
Coordinate	coor	One event has parts that co-occur over the same time interval.
Antecedent	ante	One event must necessarily occur before another event.
Precedent	prec	One event with some regularity occurs before another event.
Consequent	conseq	One event always, necessarily, occurs immediately after the other.
Sequel	seq	One event follows another with some regularity.

The relational form given above roughly states that there exists a coordinate relationship between chopping and holding. To establish this relationship, the instrument (i.e. INS) of 'chopping' must match the object (i.e. OBJ) of 'holding' and the agents (i.e. AGT) of both concepts must match.

The instantiation of a concept is accomplished by matching the associated case notation of event against the dictionary. Hence, given a phrase such as the event "A peasant was chopping a tree in the woods by the lake", it is converted to a case notation which acts as the input to the TSGR system:

```
(chop  AGT peasant  AE tree
       LOC woods-by-the-lake)
```

The definition of the case arguments are given in Table 3. The cases are meant to account for the fact that the concept 'chop' includes 'an agent who performs the action', 'the entity affected by chopping' and, optionally, 'a place where chopping occurs'.

Table 2 shows 7 relations that Alterman used in TSGR. There is one taxonomic relation: class/subclass (*sc*). For instance, a subclass of 'working' is 'chopping'. Two relations are partonomics: sequence/subsequence (*subseq*) and coordinate (*coor*). For example, 'travel' has three subsequences: 'depart', 'move', and 'arrive'. The corresponding event concepts between 'chopping' and 'holding' are in a coordinate relationship.

Four of the relations are temporal: antecedent (*ante*), precedent (*prec*), consequent (*cons*), and sequel (*seq*). An antecedent of 'dropping' is 'holding'. Sometimes before 'drinking' it is first necessary to open the container. Then, a precedent of 'drinking' is 'opening'. A consequent of 'dropping' is 'falling'. Sometimes when two objects 'hit' each other, one of them 'breaks'. Then, in the event "the cup hit the floor and broke", the relationship between 'hit' and 'break" is sequel.

3 The D&H System

Analysis in D&H consists of recognizing semantic relationships signalled by surface linguistic phenomena. The system uses as little a priori semantic knowledge as possible. Instead, it performs detailed syntactic analysis using publicly available part-of-speech lists and lexicons and produces a tentative semantic analysis. This analysis is proposed to a participating user who usually approves the system's proposal. In the case of an incorrect or incomplete analysis, the user may also be required to supply elements of the semantic interpretation. Such new elements will be learned by the system and will facilitate future processing of similar situations.

3.1 The DIPETT Parser

Syntactic analysis in D&H is performed by DIPETT (Domain Independent Parser of English Technical Texts), a broad coverage Definite Clause Grammar parser whose rules are based primarily on Quirk et al. [10]. DIPETT takes an unedited, untagged text and automatically produces a single initial parse of each sentence. This first good parse tree is a detailed representation of the constituent structure of a sentence. If DIPETT is unable to produce a single complete parse of a sentence within a

Table 3: Case attributes used in TSGR

Case	Abbr.	Description
AffectedEntity	AE	the entity affected by an event.
Agent	AGT	the entity which instigates the action.
Beneficiary	BEN	the entity on which the event has a secondary effect.
Destination	DES	the location of a thing at the end of a motion.
Instrument	INS	the tool used in performing the action.
Location	LOC	the place where an event occurs.
Object	OBJ	the thing moved or transferred.
Source	SRC	the location of a thing at the beginning of a motion.
Recipient	REC	the receiver in a transfer of possession.
StateOf	SOF	the entity which the state describes.
Theme	THM	an event or a state embedded in a perception or communication.
Time	TIM	the time of an event.

time limit imposed by the user, it will attempt to produce parses for fragments within the sentence, such as clauses and phrases. Delisle [4] presents a complete discussion of DIPETT and related parsing issues.

3.2 The HAIKU Semantic Analyzer

Given the parse trees produced by DIPETT, the HAIKU semantic analyzer [5] identifies the semantic relationships expressed by related syntactic constituents. The semantic relationships are expressed at three levels: between connected clauses, within clauses (between a verb and each of its arguments) and within noun phrases (between a head and each of its modifiers). The clause level relationships (CLRs) are assigned to connected clauses, the cases are assigned to verb-argument pairs and the noun modifier relationships (NMRs) are assigned to modifier-noun pairs. The cases that HAIKU assigns to verb-argument pairs appear in Table 4: these are the semantic relationships we will be mostly concerned with in the rest of this paper.

There are several observations to make about these semantic relationships. They constitute an amalgam of similar lists used by researchers in discourse analysis, case and valency theory. We identified an initial set of relationships and then did an extensive survey of the lexical items that mark them. This survey identified several omissions and redundancies in the lists. We further validated the relationships by checking their coverage on real English texts. Details of the construction process and validation of our list of cases appear in [3]. The next observation is that HAIKU does not depend on these particular lists of relationships. The techniques it uses at each level of analysis would work with any other closed list of semantic relationships.

HAIKU tries to assign semantic relationships with a minimum of a priori hand coded semantic knowledge. In the absence of such precoded semantics, HAIKU enlists the help of a cooperating user who oversees decisions during semantic analysis. To lessen the burden on the user, the system first attempts automatic analysis. It compares input structures to similar structures in the text for which semantic analyses have been stored. Since it does not have access to a large body of pre-analyzed text, HAIKU starts processing from scratch for a text (or a collection of texts) and acquires the needed data incrementally[1].

Clause level relationships are assigned whenever there are two or more connected finite clauses in a sentence. For each clause, DIPETT provides a complete syntactic analy-

[1]An alternative to starting analysis from scratch would be to accumulate the semantic analyses from session to session. The extent to which the acquired knowledge from one text (or domain) would be useful in the analysis of a different text is a consideration for future work.

Table 4: Semantic relationships in HAIKU

Accompaniment	Locationto	Agent	Manner	Beneficiary	Material
Cause	Measure	Content	Object	Direction	Opposition
Effect	Order	Exclusion	Orientation	Experiencer	Purpose
Frequency	Recipient	Instrument	Time_at	Location_at	Time_from
Location_from	Time_through	Location_through	Time_to		

sis including tense, modality and polarity (positive/negative). It gives us the connective (usually a conjunction) and the type of syntactic relationship between the clauses: coordinate, subordinate or correlative. The CLR analyzer looks up the connective in a dictionary that maps each connective to the CLRs that it might mark. Since the connectives are a small closed class, the construction of such a marker dictionary is not a large knowledge engineering task. Once constructed, it can be used for any text. Using the subset of CLRs, HAIKU holds competitions between each pair of relationships based on the syntactic features of the clauses. The CLR with the most points after all competitions is the one suggested to the user for approval. Within a clause, the parser identifies the main verb and its arguments: subject, direct and indirect objects, adverbials and prepositional phrases. From this information, the case analyzer builds a case marker pattern (CMP) made of the symbols psubj, pobj, piobj, adv and any prepositions attached to the verb. To assign cases to the arguments of a given verb, the system compares the given verb+CMP to other verb+CMP instances already analyzed. It chooses the most similar previous verb+CMP instances and suggests previously assigned cases for this verb+CMP. Delisle et al. [5] and Barker et al. [3] describe case analysis and the cases in detail. Within noun phrases, the parser identifies a flat list of premodifiers and any postmodifying prepositional phrases and appositives. The NMR analyzer ([2]) first brackets the flat list of premodifiers into modifier-modificand pairs and then assigns NMRs to each pair. NMRs are also assigned to the relationships between the head noun of the noun phrase and each postmodifying phrase. To pick the best NMR, the system first finds the most similar modifier-modificand instances previously analyzed. Next, it finds the NMRs previously assigned to the most similar instances and selects one of these relationships to present to the user for approval.

4 Fusion

As we mentioned earlier, TSGR can quickly determine the exact meanings in the target language of multi-sense verb appearing in the same paragraph, provided that the input takes the form of hand-coded cases associated with the current (input) sentence. More precisely, the hand-coded representation of a sentence is a set of main verbs and their respective case-value pairs, one for each clause in the current sentence. D&H has the ability to produce exactly this kind of input. In its present version, human intervention may be necessary for clauses containing new syntaxico-semantic patterns but, for a good part, all the user has to do is approve the system's suggestions. Therefore, the integration of D&H with TSGR significantly facilitates the user's task and, also, ensures that case-based semantic analysis is performed in a more consistent and coherent fashion.

The integration of D&H with TSGR involves taking the hand-coded input of TSGR associated to a paragraph, and refining it into a sequence of D&H cases, which collectively constitute a semantic representation for the given paragraph. Given a TSGR case, the task of refinement requires finding one or more relevant cases among those of D&H. To keep the coherence between both systems, the refinement

process should result in the same or a very similar sense. For instance, the combination of two 'location to' cases of D&H via a spatial proposition, namely 'by' is relevant to the LOC(ation) case of TSGR, because our fusion method produces such a result (see below). By contrast, if there is only one 'location to' case in the semantic representation at hand, its relevant case in TSGR is DES(tination). A set of rewriting rules is used in our fusion method for determining the relevant case to be used in TSGR's knowledge base (KB). A rewriting rule uses the familiar 'if-then' form, where the condition and conclusion parts correspond to the cases of D&H and TSGR, respectively. Here is an example of such a rewriting rule:

> IF $Case_i = Lat$ and $Case_{i+1} = Lat$
> THEN apply Combining-Preposition rule.

The purpose of *Combining-Preposition* is to allow the assignment of a collection of function-specific words to a *generic* information item. This can best be explained using the first phrase of the Lake-Example:

> (S1): A peasant was chopping a tree in the woods by the lake.

Given (S1), how is the assignment of the four words (*wood,by,the,lake*) to the 'location information' realized? This information can be determined by the derivation rules that are actually 'inverted Fillmorian transformations': starting from the syntactic function of a given noun phrase, the rules will derive a case as a semantic function of that noun phrase. Then, the combining rules, e.g., $LOC1 = wood$ and $LOC2 = lake$, via a spatial preposition (such as 'by') can give the new information: $LOC =$ woods-by-the-lake.

The reader may have noticed that this rule can help to determine the exact meaning of the word 'by' in the target language. According to the Webster English dictionary, the word 'by' has fourteen different usages. As a preposition, this word can be used for relating the following contexts: 'near', 'via', 'agency', 'mean', 'according to' (ex. by my watch), 'in measuring number' (ex. by degrees), 'during', 'time', 'to the extent of', 'in oaths' (ex. by God), etc.

It is interesting to see how we can integrate into TSGR the output of D&H for the above phrase, namely (S1). First of all, at the syntax level, (S1) is ambiguous because the prepositional phrase (PP): "in the the woods by the lake" is ambiguous. Indeed, (PP) can either modify 'chopping' (verb) or 'tree' (noun). When submitted to HAIKU, after full syntactic analysis with the DIPETT parser, we obtain the following final (parsing) result:

```
CURRENT SUBJECT: "a peasant"
CURRENT VERB   : chop
CURRENT COMPL  : "a tree in the woods
                  by the lake"
```

followed by this (partial, here) interaction between the user and HAIKU for semantic interpretation.

> please enter the new CP (e.g. agt-obj-tat), or enter 'h' to see the current input string and CMP, or CR to abort ["new CP"/h/CR]? **agt-obj-lat-lat**
> CMP & CP will be paired as follows: psubj/agt pobj/obj in/lat by/lat;

where, CP and CMP stand for Case Pattern (or, semantic pattern) and Case-Marker Pattern (or, syntactic pattern), respectively. Note that in the above interaction, HAIKU's request is dictated by the fact that it has no suggestions to offer.

The output of HAIKU gives us four cases, i.e. Agt-Obj-Lat-Lat. By using the above third and fourth cases, both labeled LAT and the Combining-Preposition rule, we get 'LOC = woods-by-the-lake'. Since the AGT case in D&H and TSGR are used in the same sense, then we are allowed to write 'AGT = peasant'.

A remaining question is how to assign the value of OBJ given by D&H, namely the word 'tree', to an appropriate case in TSGR? This question can efficiently be answered using the restriction list for verbs. The restriction list has two parts: *'must exist'* and *'must not exist'*, respectively. This list helps us to accept/reject those cases included in each part for the clause at hand. Note that the first part (e.g., 'must-exist') contains only the obligatory

Fusion ($\mathbf{SRP(c)} = \{\mathbf{SRS_1(c)}, \cdots, \mathbf{SRS_n(c)}\}$)

n	The number of sentences in a paragraph.
SRP(c)	The semantic representation produced by HAIKU.
$\mathbf{SRS_i(c)}$	The semantic representation of the ith sentence of the paragraph, where : $\mathbf{SRS_i(c)} = \{\mathbf{Verb_i}\ ((\mathbf{Case_{i1}}\ \mathbf{Value_{i1}}) \cdots (\mathbf{Case_{im}}\ \mathbf{Value_{im}}))\}$

Ouput: **LRC** i.e. List of relevant TSGR cases.

1. Let the set of relevant cases be empty i.e. $\mathbf{LRC} = \emptyset$
2. While the semantic representation of a paragraph is not empty (i.e. $\mathbf{SRP(c)} \neq \emptyset$) do:
   (a) Choose the first semantic representation i.e. $\mathbf{SRS_i(c)} \in \mathbf{SRP(c)}$.
   (b) Collect the cases of $\mathbf{SRS_i}$ into a list of case candidates and call it $\mathbf{LC_i}$.
   (c) While $\mathbf{LC_i}$ is not empty do:
      i. Choose the first candidate, i.e. $\mathbf{Case_j}$.
      ii. If a relevant case has already been found with respect to the current case (e.g. $\mathbf{Case_j} \in \mathbf{LRC}$), then do collect it into **LRC** and goto (v), else goto (iii).
      iii. Apply the associated combination rule, if the case(s) of that rule satisfies the condition then goto (iv), else goto (v).
      iv. Select the conclusion part of the selected rule as the relevant case and insert it into **LRC**.
      v. Remove $\mathbf{Case_j}$ from the list of candidates.
   (d) End of While.
3. End of While. Return **LRC**

case. Other optional cases, like *time* are also accepted by TSGR. If the verb has multiple-meanings, as is the case for 'chop', then for each sense a list is available in TSGR. For example, the restriction list associated to the first and second senses of the verb 'chop' (see Table 1) is, respectively:

```
[[AGT] [OBJ, AE, SRC, DES, SOF]]
[[AGT, AE] [SRC, DES, OBJ, SOF]]
```

Since HAIKU asserts the existence of four cases, then AE is assigned to 'tree'. All of the cases derived by D&H can be integrated into the cases representation. Therefore, we do not need anymore the hand-coded information as it was initially the case with the TSGR system.

To further illustrate the fusion process, let us walk through the second sentence of our Lake-Example.

(S2): He dropped his axe and it fell with a splash into the water.

HAIKU's output for (S2), in terms of case-relationships can be rewritten under the following form:

```
(HMC2) [Drop (Agt  He)(Obj  axe)]
(NMC2) [Fall (Expr (his axe))(Manr splash)
            (Lto water)]
```

where (HMC2) and (NMC2) stand for the head main clause (e.g. "He dropped his axe") and the next main clause (e.g. "It fell with splash into the water.") with respect to (S2).

The cases Agt and Obj have the same behavior both in TSGR and D&H, therefore we do not need to make the conversion. By contrast, the cases Expr (Experiencer), Manner and Lto (Location to) are converted to AGT, Manner and DES, respectively. The formal description of the fusion algorithm is shown above.

5 Discussion and Conclusion

In this paper, we have described the fusion of two systems, namely TSGR and D&H for a

verb disambiguation task. Given an input sentence, D&H derives the syntactic and semantic representations. Once this is done, we take this output as the input to achieve the disambiguation of the multi-sense verbs in the target language for translation purpose by TSGR, because TSGR has the ability to quickly disambiguate, in the target language, the multi-sense verbs appearing in the same paragraph.

The contribution of this paper to the area of fusion is the following. The fusion of the two systems frees the user from the non-trivial task of providing TSGR with the hand-coded input. And because both systems complement one another, our work also shows that new useful systems can be produced via an appropriate fusion process. In the situation we presented here, the fusion process is materialized by an interface expressed under the form of a set of rewriting rules.

Although we have considered only the analysis side of natural language processing (NLP) here, it is also important to mention the benefit of the integration of D&H into TSGR for the generative side of NLP. The information provided by D&H is rich enough to help generate the translation in the target language by TSGR because both syntactic and semantic information of D&H contain required elements of the generation model, i.e. morphology, syntax, semantics, which are complementary to the knowledge available in TSGR.

Another approach to fusion that we can use is to exclude totally the cases used in TSGR. That is to say, instead of expressing the arguments of the match function (section 2) with the cases of TSGR, we can directly use the appropriate cases of D&H. This could be done in the same spirit as the work presented here and is a good candidate for further investigation.

References

[1] R. Alterman. A Dictionary Based on Concept Coherence. *Artificial Intelligence*, **25**, 153-186, 1985.

[2] K. Barker. Trainable Bracketer for Noun Modifiers. *Lecture Notes in AI #1418*, Springer, R.E. Mercer and E. Neufeld. editors, 1998, 196-210. Proceedings of the Twelfth Canadian Conference on Artificial Intelligence, Vancouver, Canada, 18-20 June 1998.

[3] K. Barker, T. Copeck, S. Delisle & S. Szpakowicz. Systematic Construction of a Versatile Case System. *Journal of Natural Language Engineering*, **3**, 279-315, 1997.

[4] S. Delisle, *Text processing without A-Priori Domain Knowledge: Semi-Automatic Linguistic analysis for Incremental Knowledge Acquisition.* Ph.D. thesis, TR-94-02, Department of Computer Science, University of Ottawa, 1994.

[5] S. Delisle, K. Barker, T. Copeck & S. Szpakowicz. Interactive Semantic analysis of Technical Texts. *Computational Intelligence*, **12**, 273-306, 1996.

[6] A. Fatholahzadeh & H.A. Güvenir. Verb Instantiation by Concept Coherence and Refinement. In R. Mitkov et al. editors, 1997, 252-257. *Proceedings of Int. Conf. on Recent Advances in Natural Language Processing,* 11-13 Sept. 1997, Tzigov Chark, Bulgaria.

[7] H.A. Güvenir & V. Akman. Problem Representation for Refinement. *Minds and Machines, 2, 267-282, 1992.*

[8] H.A. Güvenir & G.W. Ernst. Learning Problem Solving Strategies Using Refinement and Macro Generations. *Artificial Intelligence*, **44**, 209-243, 1990.

[9] E. Protter(Ed.), *A Children's Treasury folk and Fairy Tales.* Channel, Great Neck, NY, 1961.

[10] R. Quirk, S. Greenbaum, G. Leech & J. Svartvik, *A Comprehensive Grammar of the English Language.* London: Longman, 1985.

Query Evaluation and Information Fusion in a Retrieval System for Multimedia Documents

I. Glöckner and A. Knoll
Faculty of Technology, University of Bielefeld
P.O. 10 01 31, 33501 Bielefeld, Germany

Abstract *Despite their predominant application in robotics, the utility of methods for information fusion is not limited to sensor-based fusion tasks. The paper presents an information retrieval (IR) system for multimedia weather documents which makes use of linguistic fusion methods and a semantically rich retrieval model based on methods from fuzzy set theory. The computational problem of how to efficiently organize the query evaluation process is solved by object-based mediation and asynchronous parallel invocations both of the document evaluation and fusion methods.*

Keywords: Information fusion, information retrieval, mediators, multimedia systems

1 Introduction

Today's information search services do not fully exploit the wealth of information offered. One of the reasons is that the mutual contribution of a document's parts to its content are not considered, nor are relationships of documents (e.g. hyperlinks).

A number of attempts to utilize the rich document structure of hypertext documents for querying have been proposed, for example W3QL [1] and FLORID [2]. However, the structure (partitioning in sections etc.) of a document is only indirectly related to its content. In particular, users typically know the precise structure of the individual documents satisfying their information need only *after* having found these documents. Web query languages like WebSQL [3] support the search for hypertext links. But again, users querying the IR system know the precise (hyperlink) structure of desired documents only *after* they have found the relevant documents. Therefore a *gradual measure* of a pair of documents being related could prove useful, supported by methods for processing imprecise information.

Federated IR systems aim at providing uniform access to a number of networked and possibly heterogeneous information sources. A typical architecture for information integration is depicted in Fig. 1. It mediates access to a complex system of multiple and possibly very heterogeneous information sources through "wrappers" in such a way that the illusion of a local database with rich informational content emerges. The results of the individual wrappers are merged by the *mediator* component [4] into a global logical view. Examples of such systems are HERMES [5], SIMS [6], and TSIMMIS [7].

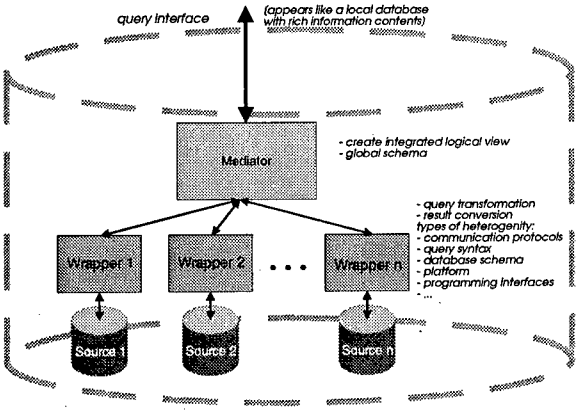


Figure 1: Information integration architecture

In the evolving field of *content-based image retrieval* [8, 9, 10], images are analysed for features like structure, color distribution (histograms or correlograms), texture etc., which all correspond to the signal level of the image (as opposed to the semantic level). Like string matching in text retrieval, these methods are not restricted to specific

domains. However, the results obtained are not yet comparable to that of text retrieval.

The low filter quality of today's generic techniques for multimedia retrieval suggests another strategy for building high-quality search services for multimedia documents, namely that of combining a substrate of generic methods for document description and information fusion with domain-specific methods which are taylored to a chosen field of application. The current concept of broad-coverage search engines is thus contrasted with that of a search service specialized to a topic area of general interest, such as weather, geography, sports, or vacation, which in this area provides search facilities on a new level of quality. These considerations lead to the following profile of a *high performance query server* (HPQS):

- *natural language (NL) interface*, to help occasional users formulate their search interest;
- *on-line search* of the document base under the user query: the complex modes of NL querying may frequently not be anticipated through pre-computed descriptors and necessitate the application of direct-search methods;
- *scalability:* acceptable response times must be ensured even for large data sets;
- *evaluation and combination* of pieces of information extracted from different sources, by applying methods for information fusion.

In the following section, we shall briefly introduce the HPQS system, and then concentrate on aspects of information fusion and query mediation.

2 The HPQS system

Fig. 2 depicts the architecture of the HPQS system [11]. The user interacts with the system via a graphical user interface (Java applet); natural language queries are typed into a query mask using the keyboard.[1] The morphological and syntactical analyses are carried out by the *natural language interface*, which generates a semantical representation of the query content. This representation is purely declarative, i.e. not directly executable. The subsequent *retrieval module* hence applies domain specific transformation rules which translate the declarative representation into a sequence of executable database queries. These trigger the generic evaluation and information fusion functionality as well as additional application methods. Execution of the generated database queries is controlled by the *multimedia mediator* which optimizes response times by maintaining a cache for storage and reuse of intermediate search results. The use of a *parallel media server* coupled with *dedicated high-speed VLSI processors* for image and text search ensures acceptable response times even when a computationally expensive online analysis of the mass data has to be performed.

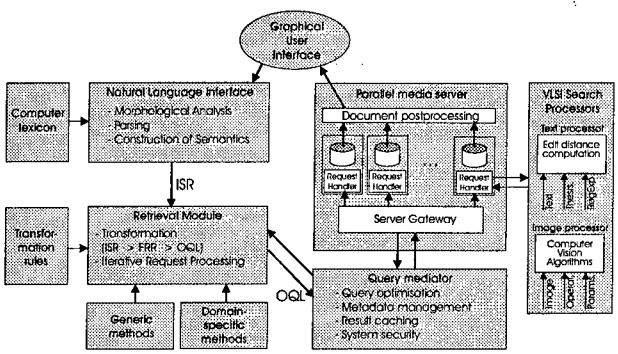


Figure 2: Architecture of the HPQS system

As the prototypical application of HPQS, we have chosen meteorological (weather information) documents. The range of meteorological documents used in our system comprises textual weather reports (ASCII and HTML), as well as satellite images and various weather maps (colour images). Query types in this application scenario include the following:

- What is the weather like in Bielefeld?
- Is it more often rainy on Crete than in southern Italy?
- Show me pictures of cloud formation over Bavaria!
- In which federal states of Germany has it been humid but warm last week?
- There were how many sunny days in Berlin last month?

The system accepts questions in exactly this form as text strings.

[1] i.e. speech input is not yet supported.

3 Formal retrieval representation

The retrieval component of the HPQS system utilizes a formal retrieval representation (FRR) which combines generic FRR methods (search techniques for documents of all relevant media and methods for information fusion) and domain-specific methods (which implement domain concepts). The FRR is syntactically identical to ODMG-OQL (Object Query Language); the FRR functionality is provided by generic and application-specific classes in the object-oriented database schema of the mediator. The generic part of FRR comprises:

- an elaborate *text-search component* (based on the dedicated VLSI processors for approximate full-text search);
- *image analysis primitives* (partly implemented in VLSI hardware);
- *discrete and parametrized fuzzy sets* and corresponding connectives from fuzzy set theory;
- *fuzzy quantifiers* which provide a numerical interpretation of quantifying expressions in NL queries.

Fuzzy quantifiers also prove useful in weighted information fusion tasks, i.e. for combining pieces of information according to numerical degrees of relevance (see below).

The generic FRR can be extended by domain-specific methods, which provide an interpretation for NL domain concepts based on the raw document data. The HPQS prototype has been taylored to the meteorology domain by implementing cartographic projections of the considered image classes; objective ("more than 20 degrees") and subjective ("warm") classification of temperature readings; estimation of cloud-top height and cloud density in satellite images; determination of degrees of cloudiness ("sunny"); and other domain concepts. In the same way that text-matching provides only a very coarse, but often still useful, approximation of text-understanding, we attempt to model only that portion of the domain concepts which must be captured to restrict the search to useful query results.

Table 1 displays the FRR sequence generated for an example query. The results of the query are shown in Fig. 3.[2]

Generated FRR

```
q_311:  element(select x.shape from x in FederalStates
where x.name = "Bavaria")
q_312:  select i from i in MeteoFranceImages where
i.date.ge(1997,8,1,0,0,0) and i.date.lower(1997,8,8,0,0,0)
q_313:  select i.pred from i in q_312 where i.pred <> i
q_314:  select ImageAndRelevance(image:i,
relevance:q_311.rateGreaterEqual(0.7, i.cloudiness().
sunny().negation().germanyProjection())) from i in q_312
q_315:  select ImageAndRelevance(image:i,
relevance:q_311.rateGreaterEqual(0.7,
i.cloudiness().sunny().germanyProjection()))
from i in q_313
q_316:  select ImageAndRelevance(image:i.image,
relevance:i.relevance.min(j.relevance))
from i in q_314, j in q_315
where j.image = ((HpqsMeteoFranceImage)i.image).pred
q_317:  select f.relevance from f in q_316
q_318:  select f from f in in q_317 order by 1
q_319:  HpqsGreyValSeq(greyval_sequence:
o2_list_GreyVal(q_318)).determineThreshold()
q_320:  select ImagesAndRelevance(image:f.image,
pred:((HpqsMeteoFranceImage)f.image).pred,
succ:((HpqsMeteoFranceImage)f.image).succ,
relevance:f.relevance)
from f in q_316 where f.relevance.ge(q_319) = 1
```

Table 1: FRR sequence generated for query: *"Show me pictures of cloud formation over Bavaria in the first week of August 1997!"*

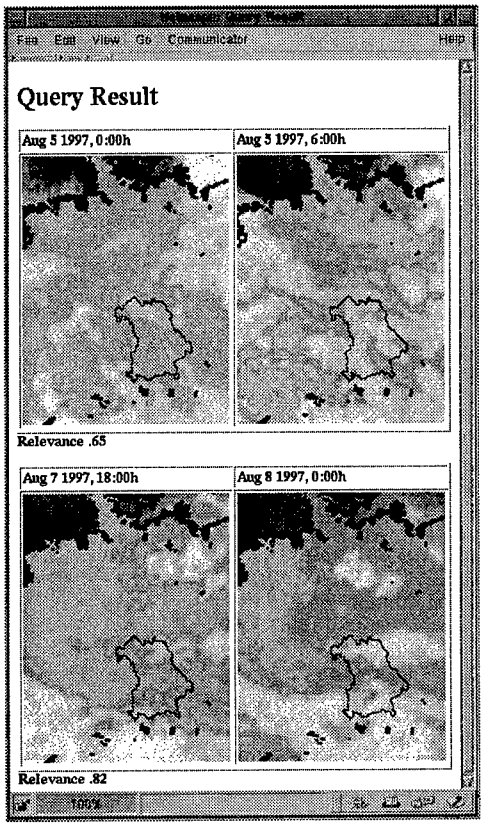


Figure 3: Result of example query

[2] see [11] for a description of the search process.

4 Linguistic information fusion

In [12], we have pointed out that providing natural language access to a multimedia retrieval system cannot be accomplished merely by adding an NL frontend to an existing retrieval "core". This is because the modes of information combination expressible in natural language are not restricted to the Boolean connectives supported by traditional retrieval systems. In particular, vague quantifying expressions (fuzzy quantifiers) like *most, almost everywhere*, are often used in NL queries to express accumulative criteria such as *"almost all of Southern Germany is cloudy"*. In this example, we have a set E of pixel coordinates. Each pixel $e \in E$ has an associated relevance $\mu_{X_1}(e) \in \mathbf{I} = [0,1]$ with respect to the fusion task, which in this case expresses the degree to which pixel $e \in E$ belongs to Southern Germany, and each pixel has an associated evaluation $\mu_{X_2}(e) \in \mathbf{I}$ which expresses the degree to which the pixel is classified as cloudy (see Figs. 4 and 5). The mappings $\mu_{X_1}, \mu_{X_2} : E \longrightarrow \mathbf{I}$ can be viewed as membership functions representing fuzzy subsets $X_1, X_2 \in \widetilde{\mathcal{P}}(E)$ of E, where $\widetilde{\mathcal{P}}(E)$ is the fuzzy powerset of E. Our goal is to provide a mapping $\widetilde{Q} : \widetilde{\mathcal{P}}(E) \times \widetilde{\mathcal{P}}(E) \longrightarrow \mathbf{I}$ which, for each considered satellite image, combines these data to a numerical result $\widetilde{Q}(X_1, X_2) \in \mathbf{I}$ as requested by the NL expression "almost all".

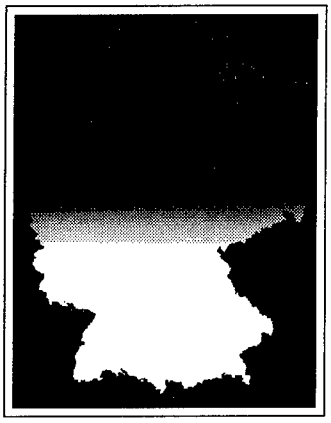


Figure 4: A possible definition of X_1 = southern_germany (Pixels with $\mu_{X_1}(e) = 1$ are depicted white)

Apparently, an operator which implements "almost all" yields adequate results only if it captures the *meaning* of "almost all". We have therefore decided to base our solution to the fusion problem on (a) the Theory of Generalized Quantifiers (TGQ [13]), which has developed important linguistic concepts for describing the meaning of NL quantifiers; and (b), methods from fuzzy set theory, known as *fuzzy linguistic quantifiers* [14, 15], which are concerned with aspects of fuzziness involved, i.e. the use of concepts without sharply defined boundaries ("Southern Germany", "cloudy", "almost all"). Our investigation of existing approaches to fuzzy quantification [14, 16, 17] based on criteria of TGQ has led us to reject these approaches because of their inconsistency with linguistic facts. Building on TGQ, we have formulated a set of axioms which characterizes mathematically sound models of fuzzy quantification; in addition, we have presented a model of the axioms [18]. In [19], we have shown that this approach is computational by presenting a histogram-based algorithm for the efficient evaluation of the resulting operators.

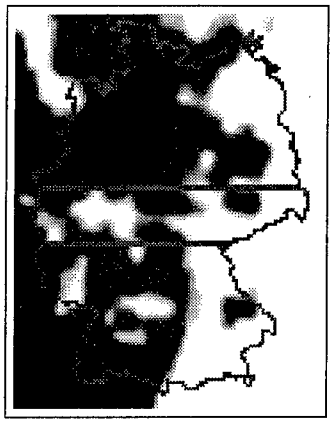


Figure 5: Fuzzy image region X_2 = cloudy (Pixels classified as cloudy are depicted white. The contours of Germany, split in southern, intermediate and northern part, have been added to facilitate interpretation)

In our system, we are currently using these operators for the fusion of fuzzy sets of pixels (local quantification) or fuzzy sets of time points (temporal quantification), see Table 2. We are hence utilizing spatio-temporal relationships between extracted pieces of information in order to compute a combined evaluation of the documents

Quantification over local regions
few clouds over Italy
many clouds over southern Germany
more clouds over Spain than over Greece
cloudy in Northrine-Westphalia (implicit)
Quantification over regions in time
almost always cold in the last weeks
more often sunny in Portugal than in Greece
hot in Berlin in the previous week (implicit)

Table 2: Examples of fuzzy quantification in the meteorology domain

of interest. This type of relationship might look different from those established by hypertext links, and from intra-document relationships (between parts of a composite document). However, all of these relationships can be deployed for retrieval purposes only if suitable methods for information fusion are available. Fuzzy quantifiers are promising in this respect because they are both human-understandable and sufficiently powerful to handle the required two-dimensional fusion problem (data to be combined plus weights of relevance). The basic aptitude of fuzzy quantifiers for combining search ratings of a document's parts to a global evaluation has recently been demonstrated by Bordogna&Pasi [20].

5 Mediation and query evaluation

In the HPQS system, we have only one (but a very complex) information source, viz. the parallel media server. The tasks of the HPQS mediator include:

- abstraction from details of the parallel media server, e.g. socket-based communication protocol and query syntax;
- making optimal use of the *parallelism* available in the external source;
- establishing a well-structured view of the multimedia system, which to the retrieval module (the mediator's client) should appear like an object-oriented ODMG database;
- maintenance of a *proxy state* of the external document base: method invocations can only be delegated to the parallel media server if the documents to which these methods should be applied are known to the mediator;
- *materialization* of results of method invocations, in order to avoid redundant computations by reusing query results of a result cache.

The efficient organisation of method invocations on the external source is of particular importance to the HPQS system because a large number of documents (and hence of instances of document evaluation and information fusion tasks) must be processed with acceptable response times. The problem is that the database executes OQL queries sequentially, and cannot directly benefit from the parallel processing abilities of the media server.

The first HPQS mediator, described in [21], makes use of blockwise request execution in order to benefit from the parallelism in the media server source. The transformation of ODMG-OQL queries to the mediator into simpler queries which can be executed in parallel will be illustrated by an example. The mediator might e.g. receive the query

```
select ImageAndRelevance( image : I,
   relevance :
      BAY.rateGreaterEqual(0.7,
      I.cloudiness().sunny()
      .negation()
      .germanyProjection()))
   from I in q_18
```

By means of query transformations, it decomposes the query in a sequence of elementary queries:

```
R1: select I.cloudiness()
    from I in q_18
R2: select I.sunny() from I in R1
...
```

These simple queries are transformed into blocks of requests and transmitted to the media server, which executes them in parallel and returns the set of results to the mediator. Using such blockwise parallel calls, the example query is executed as depicted in Figs. 6 and 7. The nodes (circles) represent elementary requests (individual method invocation given on particular choice of parameters). The dependency structure of the requests is

represented by arcs (a complex expression depends on its subexpressions in the sense that it can only be evaluated once each of its subexpressions have been evaluated).[3]

In the figures, we have assumed that nine images are to be processed and that eight processing nodes are available on the parallel server. With blockwise evaluation, execution starts which the block request to compute x.`cloudiness`() for the nine images (requests $A_1 \ldots A_9$), which is sent to the parallel server; the mediator then suspends processing until the parallel server returns the results for the whole block of requests. Having obtained the results of the first block, the mediator then initiates processing of the second block $B_1 \ldots, B_9$, to compute y.`sunny`() on all results y of the first block, etc. As witnessed by Figs. 6 and 7, this

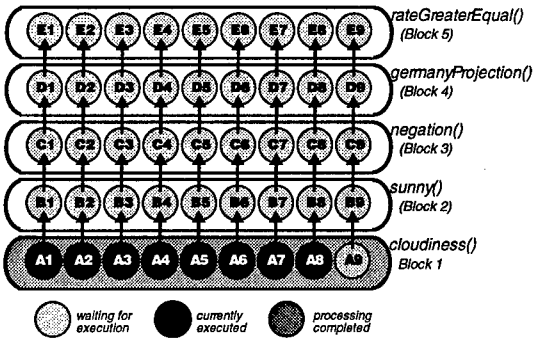


Figure 6: Blockwise Parallel Execution A

blockwise parallel evaluation does not make optimal use of the computing resources. Assuming that each request in the first block needs about 10s processing time, the parallel server will execute the requests $A_1 \ldots A_8$ in 10s. However, it needs another 10s to process request A_9 (Fig. 7). Only after 20s, the result of the block can be sent to the mediator, and processing of the second block can be initiated. This behaviour is suboptimal because when executing A_9, only one work node is active, and the other seven work nodes are idle, although the results of $A_1 \ldots A_8$ are available so that execution of $B_1 \ldots B_8$ could be started.

[3]In our example, the dependency structure is a chain, but with multiplace functions, it becomes a forest (set of trees). If intermediate results are re-used by a caching mechanism (as is done by the mediator), the structure becomes a directed acyclic graph.

The blockwise evaluation approach requires the mediator to parse OQL queries and reformulate these into blocks of requests to the media server.

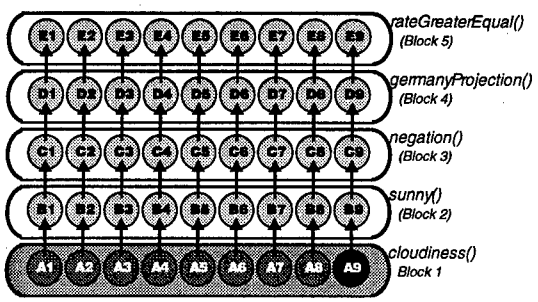


Figure 7: Blockwise Parallel Execution B

In order to avoid the intricacies of OQL analysis and translation, and to make better (i.e. more fine-grained) use of the parallel computing resources, we have decided to build an alternative mediator for the HPQS system based on parallel asynchronous method invocations. This approach rests on the following considerations. We can leave the database application unchanged (i.e. still executing sequentially) and still profit from parallel execution on the media server only if the act of initiating or triggering a request is *decoupled* from the processing of the request. In the alternative medi-

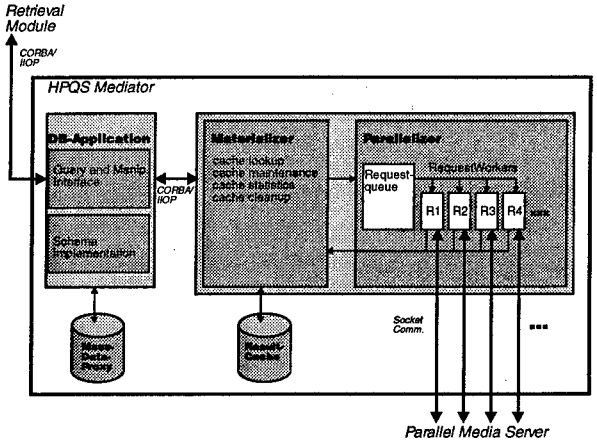


Figure 8: Asynchronous Execution Architecture

ator (see Fig. 8), we have the database *trigger* the requests sequentially: triggering is a non-blocking call which immediately returns with a *result key*. If the request cannot be found by the materializer in its result cache, it is inserted into a request queue. The parallelizer makes use of a number of `Re-`

questWorkers (one for each processor node of the parallel server) which fetch requests from the queue and cater for their execution on the parallel server.

It is sufficient for the database to know the result key to initiate further requests. Only when direct access to the computed result is necessary (e.g. in order to display a result image), it performs a "fetch" call on the result key to obtain the computed data. These fetch calls are blocking and wait until the result is available.

Snapshots of the parallel asynchronous execution of the example query are presented in Figs. 9, and 10. The database executes the query (i.e. triggers requests) using its "normal" execution order, which respects the dependencies of the requests. The requests are hence triggered, and inserted into the request queue, as indicated by the small numbers beneath the nodes in Fig. 9. The policy for ob-

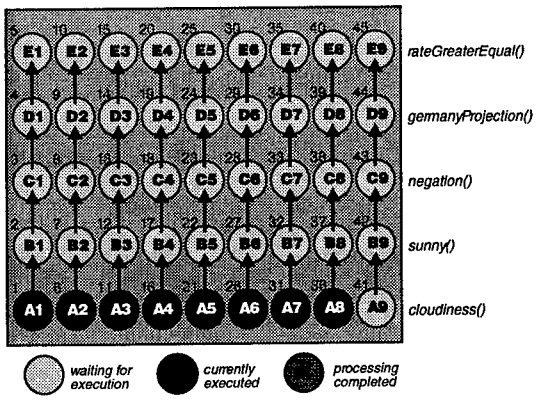


Figure 9: Asynchronous Parallel Execution A

taining the request to be executed from the queue is to select the "oldest" request all dependencies of which are satisfied (in the sense that the results for all arguments are available). The precise execution order with the parallel asynchronous evaluation strategy is hence dependent both on the insertion order into the queue and on the termination order of requests as processed by the parallel server. The initial configuration of processed requests as depicted in Fig. 9 is typical because the database triggers much faster than the requests can be executed. A later processing state is depicted in Fig. 10.

When the processing of a request is completed, the corresponding RequestWorker immediately selects the next request to be processed from the queue. We achieve a better utilisation of the parallel computing resources because it it avoided that processor nodes be idle.

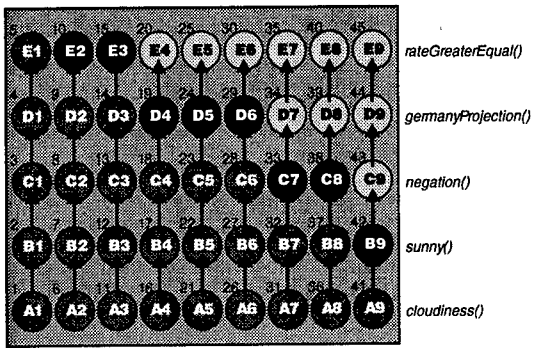


Figure 10: Asynchronous Parallel Execution B

6 Discussion

We have presented a system architecture suitable for building high-quality multimedia search services for restricted (but in principle arbitrary) topic areas. By providing an NL interface, technical barriers in accessing the system are removed. The imprecision and vagueness of NL queries must be handled because an adequate system behaviour can only be achieved if these factors do not result in system failure or implausible results. We have therefore developed a semantically rich retrieval model based on methods methods from fuzzy set theory. Emphasis has been put on linguistic methods for information fusion (viz. fuzzy quantifiers). Apart from our use of these methods to utilize spatio-temporal relationships, such methods are a prerequisite of combining the contents spread over the parts of a multimedia document, and of utilizing relationships established by hypertext links in a broad range of other applications.

HPQS supports online search and thus offers versatile ways of querying: there is no restriction to pre-computed descriptors and their Boolean combinations. We have combined several techniques in order to ensure acceptable response times, in particular *parallelisation of method invocations*, by utilizing a parallel asynchronous evaluation strat-

egy, and the use of *materialization*, which yields a speed-up for frequent queries (or subqueries) comparible to that of traditional indexing.

Although HPQS makes use of only one data source (the parallel media server), the information provided by the various document types is partially overlapping. For example, satellite images of different weather satellites (Meteosat, NOAA) or weather maps of different meteorological services can all be used to compute estimates of the degree of cloudiness at a given geographical location, and the results obtained can either support each others or contradict. Existing mediators like HERMES [5] have chosen to handle such cases by conflict resolution rules which specify a priority ordering on the sources, in order to *select* one of the conflicting pieces of information. We are currently working on the problem of *combining* (rather than selecting) such overlapping and possibly contradictory data, based on our linguistic methods of information fusion.

References

[1] http://www.cs.technion.ac.il/~konop/w3qs.html.

[2] B. Ludäscher, R. Himmeröder, G. Lausen, W. May, and C. Schlepphorst. Managing semistructured data with FLORID. *Information Systems*, 23(8):589–612, 1998.

[3] http://www.cs.toronto.edu/~websql/.

[4] G. Wiederhold. Mediators in the architecture of future information systems. *IEEE Computer*, 25(3):38–49, 1992.

[5] http://www.cs.umd.edu/projects/hermes/.

[6] http://www.isi.edu/sims/.

[7] http://www-db.stanford.edu/tsimmis/tsimmis.html.

[8] A. Del Bimbo and P. Pala. Visual image retrieval by elastic matching of user sketches. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 19(2):121–132, 1997.

[9] V. Castelli, L. Bergman, C. Li, and J. Smith. Search and progressive information retrieval from distributed image/video databases: the SPIRE project. In Nikolaou and Stephanidis [22].

[10] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkhani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: The QBIC system. *IEEE Computer*, 28(9), September 1995.

[11] A. Knoll, C. Altenschmidt, J. Biskup, H.-M. Blüthgen, I. Glöckner, S. Hartrumpf, H. Helbig, C. Henning, Y. Karabulut, R. Lüling, B. Monien, T. Noll, and N. Sensen. An integrated approach to semantic evaluation and content-based retrieval of multimedia documents. In Nikolaou and Stephanidis [22], pages 409–428.

[12] I. Glöckner and A. Knoll. Fuzzy quantifiers for processing natural-language queries in content-based multimedia retrieval systems. TR97-05, Technische Fakultät, Universität Bielefeld, 1997.

[13] J. Barwise and R. Cooper. Generalized quantifiers and natural language. *Ling. and Phil.*, 4:159–219, 1981.

[14] L.A. Zadeh. A computational approach to fuzzy quantifiers in natural languages. *Computers and Math. with Appl.*, 9:149–184, 1983.

[15] Y. Liu and E.E. Kerre. An overview of fuzzy quantifiers. (I). interpretations. *Fuzzy Sets and Sys.*, 95:1–21, 1998.

[16] A.L. Ralescu. A note on rule representation in expert systems. *Information Sciences*, 38:193–203, 1986.

[17] R.R. Yager. Families of OWA operators. *Fuzzy Sets and Systems*, 59:125–148, 1993.

[18] I. Glöckner. DFS – an axiomatic approach to fuzzy quantification. TR97-06, Techn. Fakultät, Univ. Bielefeld, 1997.

[19] I. Glöckner, A. Knoll, and A. Wolfram. Data fusion based on fuzzy quantifiers. In *Proc. of EuroFusion98*, pages 39–46, 1998.

[20] G. Bordogna and G. Pasi. A fuzzy information retrieval system handling users' preferences on document sections. In D. Dubois, H. Prade, and R.R. Yager, editors, *Fuzzy Information Engineering*. Wiley, 1997.

[21] J. Biskup, J. Freitag, Y. Karabulut, and B. Sprick. A mediator for multimedia systems. In *Proceedings 3rd International Workshop on Multimedia Information Systems*, Como, Italia, Sept. 1997.

[22] C. Nikolaou and C. Stephanidis, editors. *Research and Advanced Technology for Digital Libraries: Proceedings of ECDL '98*, LNCS 1513. Springer, 1998.

Web Data Compression for Information Collection: Organization , Navigation and Filtering with Linguistic Relationships of inclusion.

Omar LAROUK
Maître de Conférences, Université de Dijon,
High School of Information Science and Library (ENS SIB)
Villeurbanne, France, E-mail: larouk@enssib.fr

Abstract - This paper describes an approach to the design and implementation of an information retrieval capable of providing an search of users. Textual analysis is a part of information treatment systems. The access to digital data through WEB servers is facilitated by search engines. A number of Internet search engines provide classified search directories (alphabetical index, WEB guides, etc..) . Following request, the user visualizes masses of the obtained WEB pages. However, the selection of documents becomes very difficulty due to no-relevant of the obtained documents. Generally, the user visualizes the first pages but he doesn't consult the hundred ones. It is a difficult to analyzing the pertinence of documents obtained. He has to have some tools that allow to filter the information of all web pages. The aim of the present paper is to suggest a method of filtering based only on the address URL, titles, abstracts. This filtering will allow to constitute a set of filtered solutions in order to improve the reformulation of the question (request). This step is a part of the user profile modeling as a tool in order to access to information. This filtering will allow to constitute a totality of solutions between the framework of the modeling of needs oriented of the user. The module is using classification algorithms to extract more relevant 'terms' in titles and abstracts, given texts accepted and rejected interactively by the user in the process of filtering. The problem of information searching in texts is mainly a linguistic problem. The objective is to construct a system of automatic indexing that uses the model of Noun Phrases (NP). The couples intensional prédicate/NP are used from retrieval, navigation and filtering the solutions captured from the WEB. The questions, that are asked now, are : Can they play the role of descriptors of textual databases? How to organize them in Documentary Indexing System for the future research of information ? The paper describes a simple method of selecting the 'good result' and proposes an algorithm for organizing future optimal search.

Key Words - Process of Filtering. Natural language processing. Ways finding. Schema of interrogation. Relationships of inclusion. Noun Phrases (NP). Intentional predicate. Algorithms. Quantitative analysis. WEB (search engine). User profile modeling. Competitive information

1. Introduction

Access to the information through the WEB servers is very extremely used by seekers. Following a request that is formulated by means of an exploitation engine, the user receives on this screen masses of WEB pages. The user visualizes tools that allow to filter the information of all pages WEB. With the widespread stored information in Web, it is becoming increasingly important to use automatic methods for filtering such information (Belkin & Croft 1992). The goal is to propose a method of filtering based on address URL[1], titles and abstracts. The aim is to suggest:

- It is about obtaining an environment to analyze the information produced during the process of cooperation or resulted from automatic treatment.
- The linguistic approach of indexing indicates that the meaning is included in the document.

This approach favors the textual analysis (reflection of the information producer) in order to end in a representation of meaning. This study is based on linguistic techniques to optimize the following aspects :

- On improvement of automatic indexing based on an extraction of text references in order to make a good representation of its content.
- On adequate analysis of the request users in order to satisfy its informational needs.

2. Natural Language Understanding for Information Retrieval : reference and indexing

2.1. Noun Phrase (NP) : Referential function

The indexing of a document is a representation of the document so as to facilitate the obtaining of the included information. It is the passage from the textual document to internal representation (Blair, D.C., 1990). This representation has to have the semantic characteristics of this document. It has been shown that the NP[2] can be defined as a continuation of *free predicates* (Larouk, 1993a) that is constructed around a name. The NP makes a direct reference to an extralinguistic element in a fixed universe as like in the following example.

<1> /The station/ **<NP>**=*<The+station >=< quantifier + predicate >*

According to Le Guern, it has been seen that the NP_s are the *themes.* Thus, it is possible to make a correspondence between extracted NP of a text by a system and the *descriptors* that result from a manual indexing (Le Guern,1992). The extraction of NP is therefore determinate to be able to optimize an automatic indexing (Antoniadis & al., 1988), (Metzger, 1988), (Smeaton & Van Rijsbergen 1988).

2.2. Intentional predicate

The quantifier and the central *predicate* are vital for obtaining the NP. Consequently, it is around a central predicate the other neighborhood elements organize. It is often represents by a name as in next examples:

<2> / The *policy* economic / = / The (*policy*economic*) /

<2> / The [quantifier] policy [intensional predicate] economic [intensional predicate] /

The central predicate *''policy''* is an intentional element. However, it is possible to consider it as an *open intentional predicate* in order to access to the NP after its referential closing down in a documentary research. In addition, the study of elements around *intensional predicate* can give interesting information on continuity of the analysis of the type of quantifier, at the proximity, can avoid to do false analysis. We have seen that the quantification allows the actualization of simples predicates or complex predicates (*policy * economic*) (Larouk,1993a).

2.3. Closing operation of complex predicate : Appurtenance relations in a NP

The fooling example show that some information on the predicates around of syntagm center are possessed and also on that NP that are included in other (NP).

<3> *< The policy economic of <France>* NP > NP

The NP *<France>* is *included* in the NP_sentence:

<The *policy*economic* of France> NP

This appurtenance relation determines some levels. It has been shown that it is possible to define several inclusion levels with set theory (Larouk,1994). Therefore, it is possible to attribute to NP *level 1* if it is *simple*, *level 2* if it contains a simple NP, level 3 it contains a NP of level 2 and so on...*(thus of continuation).* On one hand, it can be thought that this process can be extended to other levels, on the other hand, it seems that this processes is limited in *French* (Le Guern,1992). In the framework of management of answers, the information provides by the automatic system on the inclusions between NP can be useful for oriented interrogations. The advantage of this viewpoint, by grouping referential objects in textual set, is to illustrate the composition of intentional predicates.

3. Linguistic representation of semantic hierarchy and Interrogation

3.1 Classic mode of interrogation

Documentary research is the mode that seems to match better for the user. The users questions in natural language with will be explained by IRS[3] in order to return the most relevant answers of system. In order to compare a question with the stocked documents in the database, the request will be analyzed according to the classical formalization (Salton (G), McGill (M.J), 1983), so that, its referential terms can be extracted. Therefore, the extraction of content can be carried out by logical representation. In this case, the provider solution to the user is that witch answers its request (and only this one).

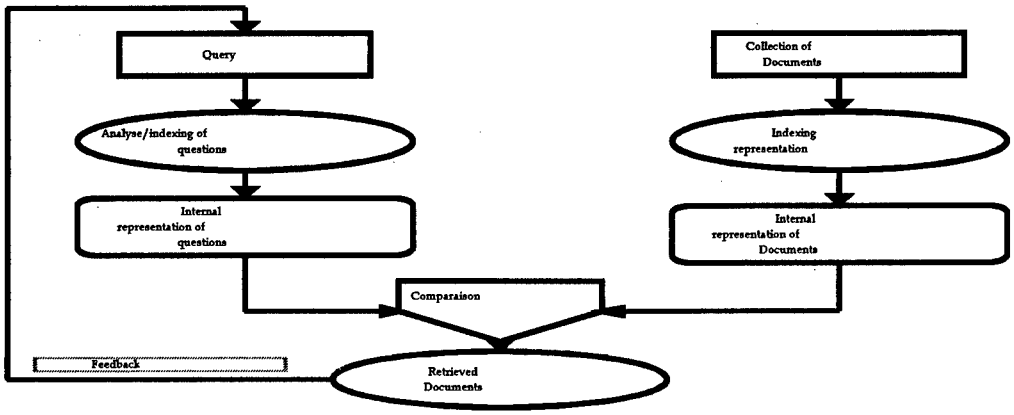


fig.1: Classical model of SRI [Salton-83, Smeaton-88, Blair-90, Belkin&Croft-92).]

3.2. Schema of interrogation : Other mode of research based on appurtenance relations

The suggested interrogation schema are based on logical approach and was developed in previous work (Larouk, 1993 a and b, 1994). The difference was made between intensional free predicates and closed predicates (NP). However, this distinction allows to analyze the interrogation problem according as these elements are intentional properties without reference to a fixed universe (*intensional logic*) or are referential functions linked well linked to well defined with the true value (*classic logic*).

3.2.1. Hierarchic informational levels

The pertinence could be tried to relations between the NP. The information levels are found in the appurtenance bonds between the words of textual sequence as shown below

<4> */Les conditions de travail des salariés des entreprises de la capitale /*

<4> *< /The/ conditions/ of/ / work / of /the /workers / of / the/ enterprises /of/ the /capital /*

[_Les conditions de travail _des salariés _des entreprises de _la capitale] ***level 0***

NP_1[*la capitale*] ***level 1***

NP_2[*les entreprises de la capitale*] ***level 2***

NP_3[*les salariés des entreprises de la capitale*] ***level 3***

NP_4[*Les conditions de travail des salariés des entreprises de la capitale*] ***level 4***

We can see the following characteristics (Larouk,1994) :

- We call NP_4 of level *4* : **[macro_NP_final]**. This final NP4 is the NP that contains all the other NP with low level.
- **NP_3** et **NP_2** are called respectively **[macro_NP]** of *level 3* and *level 2*;
- **NP_1** is called **[micro_NP]** of *level 1* ;
- The **intentional predicates** have of *level 0.*

This gradual process of levels determine one inclusion between NP. This relation reflects the links between referential objects of textual structure.

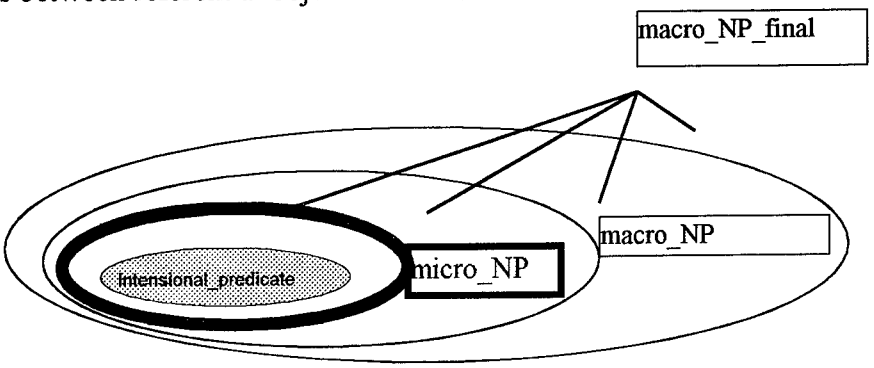


fig.2: Inclusion relationships in macro_NP_final

3.2.2. Different schema of interrogation and retrieval :

Documentary systems, that treat textual chains, are very formalized. However, these systems suit well to designers and formed users (Rich, 1984). But some problems subsist for no specialist even if they are helped by assistance systems of research that's why the possibility for questioning an information databases in natural language is the object of several studies (Copestake & Sparck-Jones, 1990), (Larouk & Bouché 1993).The next approach gives the choice between many research strategies. The notions of *intensional_predicate, micro_NP, macro_NP, macro_NP_final* are used to introduce the different navigation paths.

*3.2.2.1. **Filtering Interrogation** : Choice of navigation path*

- **Information Retrieval** *by intentional_predicate* : The database has to provide to the user all the NP, in priority, that contain the *intensional predicate* as the center of NP. If these documents do not answer to the needs of the user, then it is to possible to provide him all the NPs with upper lever (*greater level*) which contains the *intensional predicate* as the center of NP. In the case, where this *intensional predicate* appears in the shape of complex word, at first the *micro_NP* which contains this intensional predicate is proposed (or prompted) in order to avoid noisy[4] solutions.

- **Information Retrieval** *by micro_NP* : The database must to provide to the user all the NP, in priority, that contain the *micro_NP*. If these documents do not reply to needs of the users, then we can provide him with the NP of upper level (*macro_NP_final*) that contains this *micro_NP* or lower level

- **(Information Retrieval** *by macro_NP* : The databases must to provide to the user all the NP, in priority, that contain the *macro_NP*. If there are many documents that reply to the needs of users, it is possible to select the NP in this macro_NP with lower level. This operation of information reduction can be continued until to *micro_NP* of low level.

- **Information Retrieval** *by macro_NP _final.* The databases must to provide to the user all the NP, in priority, that contain the

macro_NP_final. If these documents do not reply to needs of the users, it is possible to select the NP in this macro_NP with lower level and thus of continuation.

Filtering set : Schema of Filtering Interrogation

The previous different interrogation manners are summarized in the following schema:

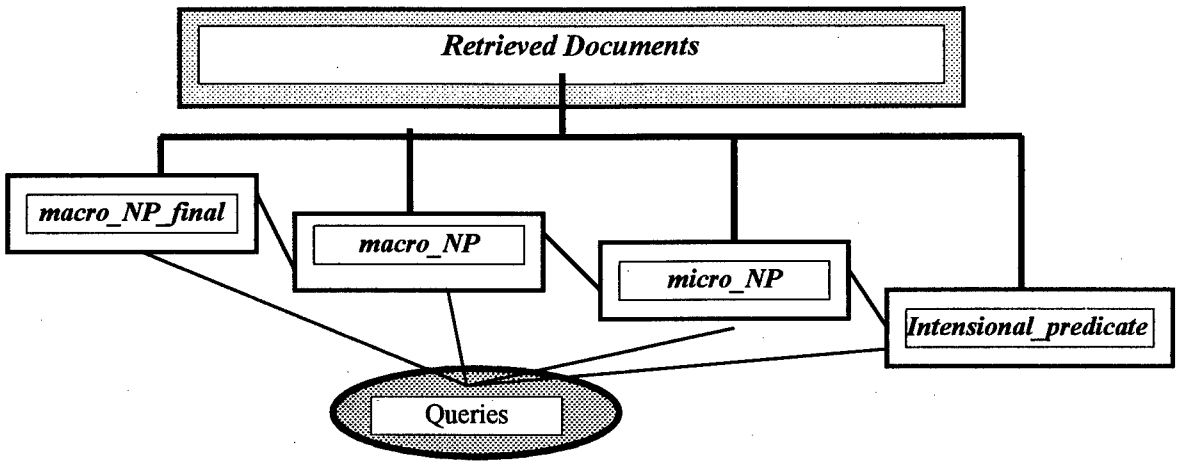


fig.3: Interrogation and Retrieval : Different path of navigation

The schema illustrated a set of solution even of the most noisy and gives the choice to the user to satisfy his demand (request). The manner permits to user to mark the susceptible solutions of his demand. In order to achieve this marking , the user has to be able to move in the structure produced by different levels. This is what we call the filtering of answers in *cooperative/collaborative* mode. To measure the importance of NP relations in indexing documents by the search engines, next natural questions are tested on the web.

3.2.1. Search engines

Search engines have developed in order to look for information stored on the Web (Lardy,1996). Two types of robots are apart (distinguished) : *the indexes* and *the descriptors.*

- Indexes engines coves all web servers, and enrich automatically (enlarge) the directory by indexing the contents (titles, abstracts, texts)
- Descriptors engines that have titles as basis or descriptions provided by the *designer-web.*

Among search engines that combine the two techniques, it there has *Francité, WebCrawler, Excite,...*

3.2.2. Results of test[5] on the WEB

modes		*Questions*
intensional predicate	Q_1	capitale
micro_NP	Q_2	*la capitale*
macro_NP_final	Q_3	*les conditions de travail des salariés des enterprises de la capitale*
predicates combined by (ET)	Q_4	*conditions ET travail ET salariés ET entreprises ET capitale*
predicates combined by (AND)	Q_5	*conditions AND travail AND salariés AND entreprises AND capitale*

	Search engines	*Results of interrogation by :*				
		intensional predicate : *Q1*	micro_NP *Q2*	**macro_NP_final** *Q3*	**ET** *Q4*	**AND** *Q5*
1.	*HotBot*	**43973**	**41401** **la=4306052** capitale=43973	217 (WEB Europe) 108 (WEB North America) **0**	336 (WEB Europe) 108(WEB North America) **7210546**	217 (WEB Europe) 108 (WEB North America **3807868**
2.	*InfoSeeK*	13183	2168144	693294	1956916	1956916
3.	*ltaVista 04.12.97*	12667	12761	**418759**	242238	138699
4.	*ltaVista 30.05.98*	172800	172800	**2047037**	1875540	1578770
5.	*Lycos*	4055	4055	0	0	0
6.	*SwissSearch*	2125	2125	11454	11454	11459
7.	*Magellan*	942	97153	177 681	60203	60203
8.	*WebCrawler*	765	107857	275 645	0	0
9.	*ogpile (Thunderstone)*	320	71	0	0	0
10	*Ecila*	200	200	200	0	0
11	*Nomade*	67	67	55077	4924	133641
12	*geocities*	66	66	0	0	0
13	*EUREKA*	60	32	63	4	344
14	*Carrefour.net*	58	59	0	0	0
15	*MetaCrawler*	38	25	23	1	0
16	*Excite(dogpile)*	10	10	10	0	0
17	*YelloWeb*	6	6	0	0	0
18	*Yahoo (Fr)*	themes=3 sites=42	themes=3 sites=42	themes=3 sites=42	themes=3 sites=42	themes=3 sites=42
19	*Lokace*	10396	(la) = 530821 capitale)=10396	*see Q3*	*see Q4*	*see Q5*
20	*Francité**	656	656	*see Q3*	*see Q4*	*see Q5*

- **Result for the question Q3**

	Search engines	**les**	**conditions**	**(2) de**	**travail**	**(2) des**	**salariés**	**entreprises**	**la**	**capitale**
1.	*Lokace*	*434312*	60036	643208	96116	491567	7651	85240	530821	10396
2.	*Francité*	Ø	1314	Ø	3443	Ø	242	4524	Ø	663

- **Result for the question Q4**

		ET	***conditions***	***travail***	***salariés***	***entreprises***	***capitale***
1.	*Lokace*	**502050**	60036	96116	7651	85240	*10396*
2.	*Francité*	Ø	1314	3443	242	4524	*663*
3.	*DejaNews*	**894744**	72937	6397	525	1697	*1428*

- **Result for the question Q5**

	Search engines	**AND**	***conditions***	***travail***	***salariés***	***entreprises***	***capitale***
1.	*Lokace*	**116433**	60036	96116	7651	85240	*10396*
2.	*Francité*	Ø	1299	3391	236	4434	*656*

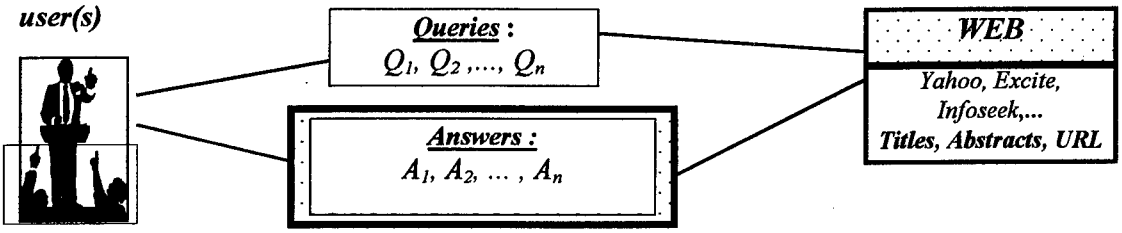
The problem of interrogation in natural language can generated the no-pertinent information for the research. However, if the user uses an important textual sequence a very long sentence such as the question Q2, the system gives false information *(HotBot gives 4306052 results for the quantifier* **'la'** **)** and *AltaVista gives for the question Q3 :*

	Questions	*Answers*	*Dates*
1.	*les conditions de travail des salariés des enterprises de la capitale*	**2047037**	*in 30.05.98*
2.	*'' les conditions de travail des salariés des enterprises de la capitale ''*	Ø	*in 30.05.98*

This situation produces the ambiguities because of the number free predicates *(the, of, each, etc.)* component of the request. We notices that the answers from search engines *(Yahoo, Nomade, ...)* presents a structure of metadata. This structure is constituted of *categories : titles, under-titles, abstracts* and *URL.*

3. Quantitative measures and Oriented filtering of solutions issued by WEB

4.1. Modeling of oriented needs of users in cooperative mode



By tapping the corresponding URL to question the WEB servers for the same question each, one will adopt a formulation that is proper to concerning *the search engine, the serves,* and *the composition of his questions*. The users has tendency to be oriented towards the server that he has already used although is competence on the other servers. Long since it's well known that the result of an indexing has to serve in an interrogation .The scheme of filtering the relations between queries and answers by the users is giving by :

a) Choice of search engine
b) Question(s) on search engine
c) Obtained all solutions indexed of the questions
d) Filtering of the solutions decided by the user in mode cooperative, *(if failure)*

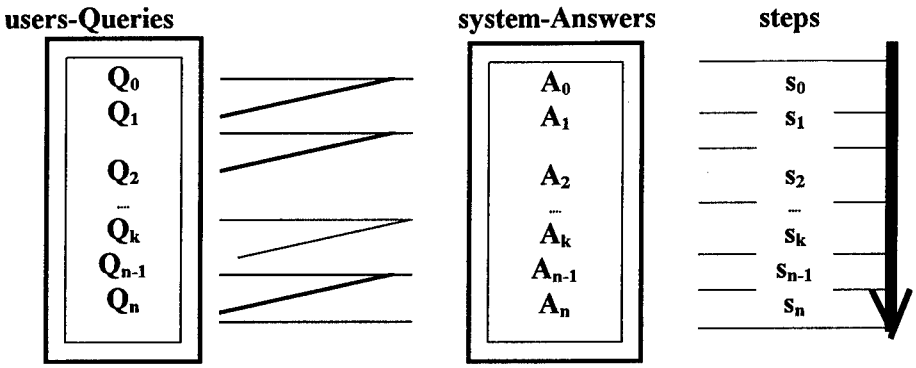


fig.4. Schema representing the process of filtering of queries by users.

A_n is the filtered final answers with the oriented needs by steps :

$S = < s_1, s_2, \ldots, s_{k-1}, s_k, s_{k+1}, \ldots, s_{n-1}, s_n >$ with $A_n = \alpha_n + \beta_n$

$A_n = \alpha_n + \beta_n$ where β_n is the set of no-pertinent solutions (rejected by users : $\beta_n = \varnothing$)

α_n is the set of pertinent solutions (accepted by users : $\alpha_n \neq \varnothing$)

The interrogation flexible consists to offering the user the possibility to eliminate the "parasite" solutions and to reformulate the request. The graduate process of filtering steps permits to reduce the set of solutions. The user constitute the database contains the solutions. However, at the present time, the result of a research on the WEB is not exploitable because of the great number of answers. It will be useful to find other tools to filter his information masses. In the present paper, we propose to use the an mathematical measurement linked to relations between the questions and the title, the abstract and URL address. The criteria permits to capture the reference of documents.

4.2. Selection of the documents with quantitative implication

The aim is to limit the noise by analyzing the answers issued by web. The captures (under the shape) ASCII of a file that resulted from an interrogation that mainly present the following structures : *titles, abstracts, URL address(documents)*. The classification of the obtained answers leans on the quantitative measurements. However, it allows to reformulate the questions in cooperative mode by means of weighting. We only choose the search engines that index the abstract and the text of document. This The implications are :

choice will permit the evaluation of, *on the one hand* the implication relation between the question and the title and, the implication between the question and the abstract, and the relation between the question and the document, *on the other hand,* the implication between the question and the *all answers captured.* However, it seems that the keywords of index files of the most robots are extracted from documentary databases and from the indexed servers.

$$\lambda f(Q \rightarrow t, a, d) = \left(\lambda t(Q \rightarrow t) + \lambda a(Q \rightarrow a) + \lambda d(Q \rightarrow d)\right) * N$$

where λ_f is the total frequency of predicate in all solutions *(N answers captured from WEB)*

If λ_f is great, so the implication of the question in the title, the abstract and the document URL address will be strong. This criterion will be used, in order to classify the answers and to permit to oriented requests, in priority, to *URL address* of WEB.

Question	Title	Abstract	Document *(obtained from URL address)*	***all answers*** captured (N solutions)
Intensional predicate ex : /capitale	λ_t is the number of occurrence of predicate in the title λ_t (Question → title)	λ_a is the number of occurrence of predicate in the abstract λ_a(Question → abstract)	λ_d is the number of occurrence of predicate in the document λ_d (Question → document)	λ_s is the number of occurrence of predicate in all documents captured (in N solutions) λ_f (Question → all_answers)

5. Elaboration of DataBases with Semi-Structured data stemming from WEB

5.1. Answers Filtering Process : Filtering algorithm

The process consists of filtering the answers and presenting them in an order to facilitated the decisional choices of user in cooperative mode. The analysis of predicates in analyzed answers has permitted to notice, that a descriptors is shown at once in the title several times and in the abstract that has a strong probability to be a *"good descriptor"* during a new research. In the present process, the user intervenes after the sentence of statistic indexing to eliminate the *parasite* solutions and then to reorient this request on the set of solutions the following procedure :

a) Choice of search engine.
b) Question(s) on search engine.
c) Obtained all solutions indexed of the questions.
d) Downloading (files ASCII and HTML) with address URL, titles, abstracts, etc..
e) Typographic Filtering the texts downloading (ASCII and HTML) by user *or/and* automatic treatment.
f) Distinction of parts of the file to process (titles, abstracts, URL) and parts to delete by user *or/and* automatic treatment.
g) Calculation of the implication of the question Q in the title (λ_t)
h) Calculation of the implication of the question Q in the abstracts (λ_a)

i) Calculation of the implication of the question Q in the document downloading by URL (λ_d)

j) Calculation of the final implication $\lambda f(Q \to t,a,d) = (\lambda t(Q \to t) + \lambda a(Q \to a) + \lambda d(Q \to d)) * N$

k) Presentation of solutions in an order (λ_f very great).

l) Final classification : Database Semi-Structured (only *URL address, titles, abstracts*).

m) New search strategies on the Database Semi-Structured (only *URL address, titles, abstracts*).

n) Navigation of the users in the semi-structured documents.

o) Relevant document downloading helps by the URL.

p) Control the solutions by the user *{if failure : question(s) on other search engines}.*

This one would orient the user towards the optimal request that would permit to capture the final document. The quantitative method that the relations between the questions and the captured elements (titles, abstracts, documents) will be used in order to construct hierarchic classification.

5.2. Constitution de databases of strategic information (indexed databases)

This modeling of users needs follows a preview process. A solution would be to present all answers (*even de most noisy* for intensional predicates) and to let the choice to user to satisfy his demand. Other solution would be to determine the *NP of the question* and he compare them to *NP solutions of titles* and *abstracts* in order to improve the filtering.

The goal is to make an syntactical analysis on the contents of tittle, and downloaded abstracts and to represent the *NP solutions* in order to construct of database filtered an indexed databases for the information research.

This optic obliges the databases to provide to the user on all the NP that answers his question. This optic of *marking* the set of solution would permit to filter the information due to the existence of a mark. When the system produce different solutions, the user has to select the best solutions and/or to call automatic analysis by *agents of filtering* (Foltz & Dumais 1992). To reaper the information, the strategies based on the algorithm of classification allows the filtering. It should be noted that the access to content of document is not obtained by such methods. The general process will be completed by a linguistic procedure of filtering :

a) Filtering lexical of intensional predicates (simple or complex) of texts of Database Semi-Structured.

b) Syntactical analysis on the texts of Database Semi-Structured (only *URL address, titles, abstracts).*

c) Classification by order of NP in titles, in abstracts and in the documents.

d) The user consults the list of NP titles in priority, *{if failure then go to e)}*

e) Presentation of solutions in an order (λ_f very great).

f) Constitution de databases of *strategic information.*

g) Choice and Evaluation.

h) Future queries tested on the Database Semi-Structured.

6. Conclusion : Perspective of this research

This study propose two complementary methods to conceive documentary system. The data first one emphasis capturing textual data with quantitative algorithm of filtering based on the measure of implication between the question and the titles, abstracts and documents. The second one adopts a method that focuses on the role of the user and on his knowledge filter the relevant answers from the Web.

Information Retrieval which is known as documents, is the process of locating and retrieving documents that are relevant to the user queries. The approach which allows the user to navigate and inspect the database documents captured

according his demand. The future search strategies proceed on the Database Semi-Structured (only *URL address, titles, abstracts*) to look for relevant document. The aim is to constitute de databases of strategic information.

In the case of an IR, there is no correspondence between the set of reference (NP) that the user wants and the set of reference that the system is going to suggest to him. To limit the noisy/silence problem, we have to call the linguistic tools. We suggested some elements to study the different schema of interrogation The notions of *intensional_predicate, micro_NP, macro_NP, macro_NP_final* are used to introduce the different navigation paths. The research will be oriented toward complying of linguistics techniques with filtering tools.

References

Belkin, N.J ; Croft, W.B. (1992). *Information Filtering and Information Retrieval : Two Sides of the Same Coin ?* .In : Communication of the ACM. Vol.35, No.12. pp. 29-38.

Blair, D.C. (1990). Language and Representation in Information Retrieval. Elsevier. Amsterdam.

Copestake, A., Sparck-Jones, K. (1990). *Natural Language Interfaces To Databases*. The Knowledge Engineering review, n°5, part 4.

Foltz, P.W, Dumais, T. (1992). *Personalized Information Delivery : An Analysis of Information Filtering Methods*. In : Communication of the ACM. Vol.35, 1992, No.12, pp. 51-60.

Grishman, R. (1984). Natural Language Interfaces. Journal of the ASIS , 35. Pp. 291-296.

Lardy J-P. (1996). *Les outils de recherche d'informations sur Internet*. Publications ADBS. Paris.

Larouk O. (1997). *Logico-semantic and Statistics applied to textual data in information retrieval : Quantitative Mathematics (Set theory, Logics, Statistics) and Algorithms applied on textual data in Information Retrieval*. in the Third International Conference on Quantitative Linguistics. **QUALICO' 97-IQLA**; Research Institute for the Languages of Finland. pp. 135-143.

Larouk O. (1994). *Extraction de connaissances à partir de documents textuels : Traitement automatique de la coordination (Connecteurs et Signes de ponctuation)*. Thèse de Doctorat, spécialité : Informatique, Université Claude Bernard Lyon 1 ; 359 p.

Larouk O. (1993). *Application of Non-Classical Logics to Optimize Textual Knowledge Representation in an Information Retrieval System*. in HEURISTICS : THE JOURNAL of Knowledge Engineering. Volume 6, Number 1 Spring 1993. Gaithersburg, MD- USA; pp. 25-37.

Larouk O. (1993). *Linguistico-Statistical and Logics Applied for Documentary System : Algorithm of Correction punctuations signs*. ACM-SIGAPP'93 : APPIED COMPUTING: States of the Art and Practice. ACM Press-NY, Indianapolis. pp. 737-744.

Larouk O. Bouché R. (1993). *Apports des logiques et de la linguistique dans la conception d'interface de Bases de données textuelles*. in Pluridisciplinarité dans les Sciences Cognitives. HERMÈS. Paris. pp. 142-160.

Le Guern M. (1992). *Un analyseur morpho-syntaxique pour l'indexation automatique*. Le Français Moderne; tome LIX; n° 1. pp. 22-35.

Metzger J.P. (1988). *Syntagmes Nominaux et Information Textuelle*. Thèse d'Etat ès sciences. Université C. Bernard Lyon I.

Rich, E. (1984). *Natural Language Interfaces*. Computer. September.

Salton G, Mc Gill M.J. (1983). *Introduction to modern information retrieval* . Mc Graw-Hill; New York.

Smeaton A.F, Van Rijsbergen C.J. (1988). *Experiments on Incorporating syntactic processing of User Queries into a Document Retrieval Strategy*. ACM-SIGIR'88; Grenoble; June 13-15. pp. 31-51.

---

1 URL (Uniform Resource Locator)
2 NP (Noun Phrase)
3 IRS (Information Retrieval System)
4 In IRS, *Noise* is represented by the selection of inappropriate documents. *Silence* is represented by relevant documents which have been not selected.
5 This test was realized le 4 December 1997 on the WEB.

Session WB1
Knowledge-Based Techniques for Information Fusion and Discovery
Chairs: Ray Liuzzi and Craig Anken
Air Force Research Laboratory, USA

Domain specific document retrieval using n-word combination index terms

David B. Johnson, MS Wesley W. Chu, PhD

Department of Computer Science
University of California, Los Angeles

Abstract ***Traditional text based information retrieval is based on isolated keywords or word stems. Without a context, words are frequently ambiguous. The ambiguity of isolated words decreases the precision of information retrieval tasks and additional contextual words may increase precision. This motivated us to develop a method to extract word combinations from text documents. We define an "n-word combination" as n words that co-occur in the same context. Brute force methods to calculate n-word combinations are limited to small documents. Our technique uses the structure (e.g., sentences) of the document to limit the search for the word combinations, thus it can scale to large documents.***

The n-word combinations can be used to represent documents via a vector space model. We have used the resulting model to perform document retrieval tasks. We have compared the precision and recall of the n-word combination model with that of the traditional isolated keyword or word stem vector space models. Our results reveal that using n-word combinations to model documents can significantly improve the precision of query results.

Keywords: Domain-specific information retrieval, text data mining, medical application

1 Introduction

Information retrieval systems index free text documents using keywords. These systems have been shown to be useful to access general document spaces most recently for indexing and accessing the World Wide Web. We are exploring the indexing and access of domain specific information sources where additional information may be available that could be used to improve the information retrieval process.

The field of medicine provides a vast set of text documentation, including medical literature and a variety of patient medical documents. Medical specialties are well-defined, often with their own specialized vocabulary. These specialties provide a test-bed for the exploration of domain specific information retrieval.

In a medical teaching facility, on-demand and interactive teaching material based on real patient population data can significantly enhance the ability of instructors to teach students, house-staff, and other colleagues. Currently, these teaching files are manually indexed by anatomical site and disease process. However, it may be difficult to locate different kinds of a particular disease. Furthermore, the static nature of these teaching files do not facilitate the incorporation of recent medical cases nor do they enable automated cross referencing of patient files with existing teaching cases.

This paper describes a method to select and use multi-word combinations as indexing terms. A set of thoracic radiology medical reports was selected for our test domain.

2 Related Work

Automated information retrieval follows three general steps:

1. Index term selection
2. Encoding documents
3. User query processing

An *indexing term* is defined as a set of unique words that characterize some feature found in the document set. Documents are encoded or modeled using the indexing terms that appear in them. User

queries are processed by mapping the query to the indexing terms previously extracted from the document set and then matching the query to individual documents.

Term selection is also important in text-based knowledge discovery systems [1]. These systems data mine text documents to discover patterns which hold across many documents. Patterns can take the form of one term or phrase that co-occurs with a second term or phrase [1]. The frequency of word or word n-gram occurrence can also be used by experts to find useful information as well as anomalies in data sets [5].

Index term selection Automatic information retrieval systems often select indexing terms based on their ability to differentiate documents rather than content. Typically, indexing terms are selected based on their frequency of use in the corpus. Intuitively, words which are used frequently do not differentiate documents well, as they appear in a large subset of the corpus. Experimental results have shown that neither high nor low frequency words work well for indexing documents [15].

Various methods exist for selecting or normalizing indexing terms. *Stop word lists* are used to eliminate frequent words. *Stemming* is used to normalize words with similar meaning to a common prefix (e.g., the word "masses" is stemmed to "mass"). Using statistics derived from the document set, weights can be added to the indexing terms to reflect their individual classification power.

Encoding documents For indexing purposes, documents can be represented by the set of indexing terms found in them. Each indexing term can be considered a *character* of the document set, where each character has a single well-delineated meaning or definition.

Individual documents can be represented as an n-dimensional *document vector*, where each vector term represents one of the indexing terms selected from the corpus. Documents are encoded using the n-dimensional document vector by assigning positive values to those vector terms which correspond to indexing terms found in the document. Vector terms corresponding to indexing terms which do not appear in the document are assigned a null value [14].

The similarity of two documents can be measured using a variety of methods. In retrieval systems utilizing a vector space model, one frequently used measure is the cosine of the angle between two vectors (i.e., query vector and document vector). The cosine of the angle between vectors has been shown to perform better than using the Euclidean distance between two vectors as a measure of similarity between document vectors [15].

Query processing can be performed by transforming the query terms into the n-dimensional vector used to model the documents, forming a *query vector*. The query vector is compared to each of the document vectors, forming a list of similarity measures. Using the list of similarity measures, documents can be ranked and returned to the user.

In many systems, the indexing terms used to represent documents consist of individual word stems. The word stems, isolated from other words, may be ambiguous, thus are not well suited to serve as well-delineated *characters* of the document set. For example, the isolated word "mass" may refer to any mass and is not specific to any anatomical location. Using "mass" as a character of the document set, while separating reports with the "mass" character from those that do not contain this character, does not differentiate between specific mass lesions (e.g., right upper lobe mass).

This is especially true in radiology reports where frequently all anatomy examined in the study is described. Using isolated word stems, it is frequently impossible to fully classify particular medical findings. For example the following six words refer to a hypothetical patient's lungs {clear, left, lobe, lung, mass, right}. In this example, words are isolated and word order does not effect the "meaning" of the set. Using no other information it may be possible to infer that one lung is clear and the other contains a mass. The ambiguities introduced by isolated word stems decreases our ability to interpret the individual findings.

To accurately index and access patient reports by disease and anatomy, a system should be able to differentiate between various anatomical structures and involved findings. Research has provided

evidence that multi-word or phrase indexing terms may improve retrieval performance [2, 10, 3].

Previous research has focused on short phrases (i.e., 2-words) applied to general information retrieval test beds. The results from this research was limited in that phrases were calculated by combining terms taken from *anywhere* in the document [2]. As the size of a document can be large (e.g., 100s to 1000s of words), brute force calculation of all word combinations in the document can not be easily managed.

A system called INDEX used n-grams to extract content from legal documents [10]. The developers of INDEX noted that the number of potential n-grams is large, and may include many "meaningless" phrases. To solve this problem, they used human experts to eliminate useless phrases. Recognizing that this solution does not scale well, a second system (INDEXD) was developed. Instead of human experts, INDEXD used a dictionary associating word stems with a list of similar terms. Associated terms were normalized to a common term, thus increasing the frequency of common "meaningfull" phrases [10].

These results are promising, showing that multiple word indexing terms may provide important contextual information, thus improving retrieval performance. Still the data sets used were general in nature, making text mining difficult. We are using domain specific data sets, which may enable the extraction of common patterns which can be used to improve retrieval. This additional information will enable the system to more accurately model the content of the individual documents, without the use of external knowledge sources such as a dictionary of associated terms.

3 Method

Several methods exist for defining multi-word indexing terms. An n-gram is defined as an ordered sequence of n words taken from a document. For example, "several methods" and "methods exist" are the first two bi-grams of the last sentence. Given a document d of length l there are $(l - n)$ n-grams in d. By providing context lacking from isolated words, n-grams may more accurately model the content of documents.

Two other factors may influence the effectiveness of n-grams as indexing terms. First, n-grams are dependent on word order, thus "right upper lobe mass" is not equivalent to "mass right upper lobe." Second, n-grams are limited by word proximity, requiring that words appear next to one another in the original text. For example, in the text sample: "a mass is seen in the right upper lobe," here, "mass" and "right upper lobe" will only appear together if a 8-gram is used to model the text. Removing typical stop words from the sample results in "mass seen right upper lobe," still for the finding and anatomy descriptions to appear together requires an n-gram with a minimum length of 5 terms.

N-grams may improve retrieval precision by providing additional context over isolated words. However, reliance on the original document's word order as well as word proximity may decrease retrieval recall and may require longer n-grams to be used to model documents.

We define an n-word combination as an unordered collection of n words taken from a document. Given the text sample: "right upper lobe mass," there are 6 different 2-word combinations, including "right upper" and "upper mass." Unlike n-grams, n-word combinations (n-combos) do not depend on word order or proximity. Any set of n words can form an n-combo.

Removing the restriction on word order and proximity dramatically increases the number of potential n-combos. Given a document d of length l, there are $l!/(n!(l-n)!)$ n-word combinations in d. As the length of the document grows, the number of n-combos grows dramatically (e.g., a 100 word document has the potential of 3,921,225 4-combos, a 200 word document has 64,684,950). Brute force calculation of all possible n-word combinations in a document, even for relatively small n, is too time and space expensive. In order to use n-word combinations, some method of limiting the search space must be defined. Furthermore, a method to select which n-combos should be used as indexing terms must be developed.

Although each document has a central theme (e.g., a medical report describes an individual patient), the concepts useful for indexing

are described in the individual sentences of the document. Limiting the search scope to individual sentences will dramatically decrease the time and space required to calculate n-word combinations, while focusing on relevant indexing terms. Furthermore, stop word lists can be employed to factor out those words that do not carry any semantic significance, further reducing the search space. Finally, statistical information concerning n-combos can be used to focus the search for subsequent $(n+1)$-word combinations, such that if the n-word combination is infrequent, the $(n+1)$-combo is also infrequent.

4 Implementation

Indexing term selection While indexing terms must distinguish different documents, the terms must, more importantly, distinguish between different meaning or content. Although some n-word combinations may provide key information allowing better modeling of documents, other combinations will not be useful for document retrieval. For example in the sentence:

A 3cm right upper lobe mass is noted.

"right upper mass" is a useful indexing term, while "right upper noted" will generally not be useful. Clearly, the number of possible n-word combinations in each document is large. To decrease the storage requirements necessary to model each document, methods to filter and select word combinations are necessary.

Several standard methods are used to normalize the terms used for indexing. First, a short list of stop words is utilized to remove common words. Second a stemming algorithm, as implemented by the SMART system, is used to normalize each word to a common stem [14].

In order to limit calculations, only words found together in a single sentence will be combined. Using this rule, the system will not combine a word found only in one sentence (e.g., first sentence) with a word found only in another sentence (e.g., second sentence). This single rule dramatically decreases the space and time requirements of the system.

In each document, indexing terms are extracted by first identifying sentences and words in the document. A pre-processor transforms documents into sentences. A set of rules are used to define sentence boundaries. The rules define a sentence as a set of words followed by a period. Other rules account for other uses of the period, for example in numerical measurements, or general formating (such as lists). These rules attempt to minimize false sentence partitioning.

Each sentence is processed sequentially. First, individual words are extracted from the sentence. Next, stop words are removed and each word is stemmed [13]. Individual word statistics, as processed by an earlier stage, are used to filter out infrequent words. Words that appear more than once in the sentence are combined. Finally, the remaining words are sorted in alphabetical order, forming a sentence word list.

Word combinations are calculated using the sentence word list. Although the earlier stages decrease the number of possible word combinations, we have found that space considerations are much more of a problem than time issues. Although rare, long sentences can result in a large number of n-combinations. In these cases, many of the resulting n-combinations only appear once, thus they are not useful for indexing. To filter out these word combinations, each combination is evaluated before storage.

Each n-word combination is evaluated for storage by examining each of its "child" word combinations. A "child" combination for a n-word combination is defined as any of the $(n-1)$-word combinations that can be derived from the n-word combination. For an n-word combination, there are n $(n-1)$-word combinations. Frequency statistics are maintained for each set of word combinations. If any one of the $(n-1)$-word combinations does not meet a frequency threshold, the n-word combination is discarded. Each remaining n-word combination is considered an indexing term.

Table 1 shows the most frequent 2- and 3-word combinations extracted from a set of thoracic radiology reports. As can be seen from the list, disease (e.g., mass) and anatomy (e.g., upper lobe) can be linked by word combinations.

2-word combo	3-word combo
lobe upper	lobe right upper
lobe right	lobe mass upper
right upper	left lobe upper
left lobe	lobe lower right
lobe lower	left lobe lower

Table 1: Frequent 2- and 3-word combinations

Encoding documents Documents are modeled using a vector space model, where each vector term corresponds to a single indexing term. Indexing terms in a vector space are of the same length. The system uses multiple vector spaces, or models, to encode each document, each vector space corresponding to a different length indexing term.

In general, we are use three models to encode and represent documents: single word index terms, two word index terms and three word index terms.

Query processing Similar to existing information retrieval systems, queries can take the form of free-text natural language queries, or may include a sample document. The query is processed by transforming the query text using the vector space models which have been used to encode the document set. As each document is encoded using multiple models, the query is similarly modeled.

For each document in the corpus, a similarity value is calculated comparing the document to the user's query. For a corpus of size n documents with each document encoded using m vector space models, $n \cdot m$ similarity values are calculated. A combined similarity between each document and the query can be calculated by summing the similarity measures from each representation.

5 Evaluation

Traditional information retrieval evaluation has a number of well documented problems [7]. One problem is associated with the determination of relevance [4, 8]. Traditionally, a set of experts define queries and the set of relevant documents that match each query. It has been described that a user's criteria for relevance is much more difficult to describe and often does not match those of the experts [12].

To evaluate n-word combinations we have developed a simple definition of relevance based on the absence or presence of medical findings. Medical documents often describe abnormal anatomy or medical findings. These descriptions often include the anatomy involved and the type of finding. In creating medical teaching files it would be useful to specify the anatomy involved and the type of finding, for example the query "right upper lobe mass" would return all patient reports that describe a right upper lobe cancer mass. Using these queries it is simple to determine if a particular document is relevant or not. Only those documents specifically describing the anatomy and finding in question are considered relevant.

Thoracic radiology reports frequently describe abnormal lung anatomy (e.g., cancer) and enlarged lymph nodes. The text descriptions used to describe medical findings are often similar. The two lungs are referred to as the right and left lungs. Each lung has an upper and lower lobe, with the right lung having a middle lobe. Medical findings are frequently described in proximity to surrounding anatomy or other findings, thus each document may contain descriptions referring to both lungs and several lobes. Indexing using single word terms may be easily confused by other descriptions in the document.

We have used two methods to compare the results of the four methods. Retrieval performance is traditionally measured using recall and precision metrics. *Recall* is the ratio of the number of relevant documents returned over the total number of relevant documents. *Precision* is the ratio of the number of relevant documents returned over the total number of documents returned [3]. To compare the recall and precision of each technique, we use an 11-point precision / recall graph [15]. Also, we compare the retrieval effectiveness (E), defined as the harmonic mean of recall (R) and precision (P) for each of the techniques and term size [16]. Retrieval effectiveness (E) is defined as:

$$E = \frac{2}{\frac{1}{R} + \frac{1}{P}} \quad (1)$$

The value of E ranges from 0 to 1. $E = 0$ when no relevant documents are retrieved and $E = 1$ when only all relevant documents are retrieved.

For a given query and retrieved document set, a maximum value of E can be found, representing the optimal combination of precision and recall. The value of E increases as both recall and precision increase [16].

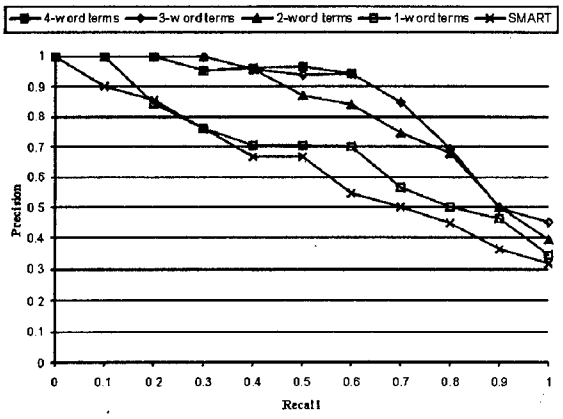


Figure 1: Query - "right upper lobe mass"

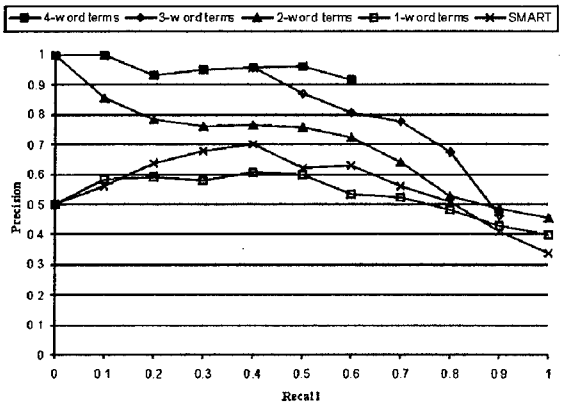


Figure 2: Query - "left upper lobe mass"

To evaluate the system, 178 thoracic radiology reports were processed. The data set was originally used as a test-bed for extracting key features using natural language processing techniques [9]. Recently, the data has been used for text data mining experiments [5]. Within the document set, the average document length is 11.2 sentences with an average sentence length of 13 words. In total 1988 sentences were processed. Each document was encoded using 1-word, 2-word, 3-word and 4-word combinations. Each document was also processed and indexed using the SMART system [14].

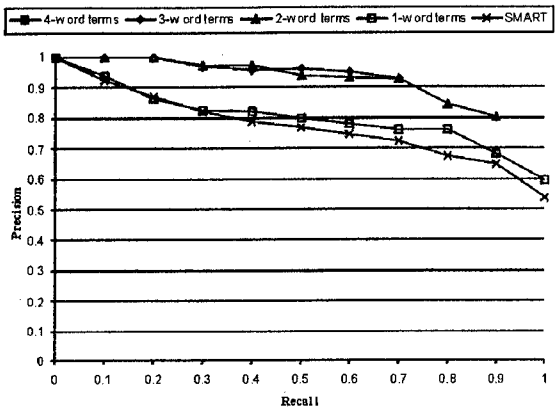


Figure 3: Query - "upper lobe mass"

To explore the retrieval performance of both systems given the bilateral similarity of the two lungs, two queries differing only by the lung of interest were selected, a third query combining the results of the other two queries was also processed. The queries both request documents describing upper lobe cancers, one for the right lung the second for the left. These queries expose some of the difficulties to accurately retrieve documents where single word indexing terms are used to describe several different medical findings (e.g., upper – can refer to either lung). In radiology images, tumors are seen as large opaque regions and are often described in the resulting radiology report as a "mass." To reflect the domain the following three queries were processed and indexed by each system:

Q1 "right upper lobe mass"

Q2 "left upper lobe mass"

Q3 "upper lobe mass"

Figures 1, 2, and 3 compares the precision and recall for each of the three models. The results show that multi-word indexing terms (2 and 3-word indexing terms in Figures 1, 2, and 3) can improve retrieval results over isolated single word indexing terms (i.e., SMART system and 1-word indexing terms).

Table 2 shows the retrieval effectiveness for each of the techniques used. The results show

that multi-word indexing terms improve effectiveness compared to the single-word methods (i.e., SMART system and 1-word indexing).

The prototype is implemented using the Java programming language [6]. Initial indexing of the test set of 178 reports calculating 1-word, 2-word and 3-word combinations and encoding each of the documents completes in less than 5 minutes using standard PC hardware. Response time for subsequent queries, where queries are processed and displayed to the user, is under 2 seconds. The evaluation runs were performed using a 200 Mhz AMD K6 processor, running Windows NT, using the Java 2 developer's kit [11].

6 Discussion

Our prelimiary evaluation reveals that n-word combinations can be effectively used for indexing and providing access to medical documents. The results show that multiple word combinations, within the context of a well defined domain, provides better retrieval performance for some queries than isolated word stems. Although the number of possible word combinations that can be calculated and used as indexing terms can be very large, the experiments have shown that within a well-delinated domain the number is manageable using the filtering techniques described here. Furthermore, while it is not possible to model complex concepts using isolated words, word combinations allow certain concepts to naturally appear (e.g., "left lobe mass").

For a specific subject area (e.g., thoracic radiology) with a uniform vocabulary, word combinations appear to capture the semantics of the document better than isolated word stems, thus may better model the content of the document. By joining multiple words as indexing terms, the model may decrease ambiguities that exist in isolated word stems, thus better serving as *characters* for the document set.

While our preliminary results are promising, further evaluation is necessary. First, a wider set of queries must be tested to better understand the strengths of this form of indexing. Also methods must be developed extending the retrieval improvements found in domain specific document sets to more general document sets. Furthermore, using external knowledge sources (e.g., UMLS Metathesaurus, SNOMED) to assist filtering the word combinations extracted by the system may further improve performance.

While partitioning documents by sentences works well in most cases, long sentences can impact the run time of the system. We would like to perform further studies of document partitioning to determine its influence on retrieval performance. One partitioning scheme would use a moving fixed-sized window to segment the document currently being processed. This method would not rely on sentence partitioning, thus would not be affected by long sentences. Furthermore, as the window would have a fixed size, calculating maximum run-time and space requirements would be simplified.

While we have run the system to calculate up to 8-word combinations, these combinations do not appear to be useful for indexing. First, user queries are often limited to a few words, thus large word combinations are not useful for short queries. An example of this can be seen in processing query Q3 (see Figure 3 and Table 2). Furthermore, word combinations may expose specific concepts (e.g., right upper lobe mass), however adding words to these combinations may not necessarily increase meaning.

7 Conclusion

Information retrieval systems can be used to automatically index free text reports, thus providing access to associated data such as patient images. The type of indexing term used can have a significant impact on retrieval.

While, traditional methods use isolated word stems to model documents, we have developed a prototype system that uses word combinations automatically extracted from documents to index the document set. Multiple vector space models are used to allow ranked retrieval of patient reports.

Preliminary evaluation results of the method is promising, showing that multi-word combinations provide better retrieval performance than isolated

Indexing method	Retrieval Effectiveness (F)		
	Q1	Q2	Q3
SMART	.6087	.6296	.7512
1-word terms	.6667	.6154	.7895
2-word terms	.7458	.6916	.8587
3-word terms	.7879	.7579	.8199
4-word terms	.7500	.7857	NA

Table 2: Retrieval effectiveness (F)

word stems. While isolated word stems may be ambiguous, word combinations may better capture the content of the document, thus improving performance.

References

[1] C Clifton and R Steinheiser. Data mining on text. In *Proceedings. The Twenty-Second Annual International Computer Software and Applications Conference (Compsac '98)*. IEEE Comput. Soc, 1998.

[2] JL Fagan. The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval. *J. Amer. Soc. for Information Sci.*, 40(2):115–132, 1989.

[3] WB Frakes and R Baeza-Yates, editors. *Information Retrieval Data Structures & Algorithms*. Prentice-Hall, Inc., 1992.

[4] TJ Froehilich. Relevance reconsidered – towards an agenda for the 21st century. *J. Amer. Soc. for Information Sci.*, 45(3):124–134, 1994.

[5] JA Goldman, WW Chu, DS Parker, and R-M Goldman. Tdda, a data mining tool for text databases: A case history in a lung cancer text database. In *1st International Conference on Discovery Science*, 1998.

[6] J Gosling, B Joy, and G Steele. *Java Language Specification*. Addison-Wesley, 1996.

[7] SP Harter. Variations in relevance assessments and the measurement of retrieval effectiveness. *J. Amer. Soc. for Information Sci.*, 47(1):37–49, 1996.

[8] SP Harter and Hert CA. Evaluation of information retrieval systems: approaches, issues, and methods. In *Annual review of information science and technology*, volume 32. Information Today, 1998.

[9] DB Johnson, RK Taira, AF Cardenas, and DR Aberle. Extracting information from free text radiology reports. *International J. on Digital Libraries*, 1:297–308, 1997.

[10] LP Jones, Jr. EW Gassie, and S Radhakrishnan. INDEX: The Statistical Basis for an Automatic Conceptual Phrase-Indexing System. *J. Amer. Soc. for Information Sci.*, 41(2):87–97, 1990.

[11] Sun Microsystems. Java 2 software developer's kit. *www.java.sun.com/products/jdk/1.2/*.

[12] TK Park. The nature of relevance in information retrieval: An empirical study. *Library Quarterly*, 63(3):318–351, 1993.

[13] G Salton. *Automatic Text Processing*. Addison-Wesley, 1989.

[14] G Salton, L Allan, and C Buckley. Automatic structuring and retrieval of large text files. *Communications of the ACM*, 37(2):97–108, 1994.

[15] G Salton and MJ McGill. *Introduction to modern information retrieval*. McGraw-Hill, 1983.

[16] WM Shaw Jr., R Burgin, and P Howell. Performance standards and evaluations in IR test collections: vector-space and other retrieval models. *Information Processing & Management*, 33(1):15–36, 1997.

IMPACT: Intelligent Mining Platform for the Analysis of Counter-Terrorism

Sherry E. Marcus
21st Century Technologies Inc.
8716 North Mopac Suite 310, Austin, TX
78759 USA
Email: sem@cais.com

Darrin Taylor
21st Century Technologies Inc.
8716 North Mopac Suite 310, Austin, TX
78759 USA
Email: dtaylorz@aol.com

Abstract

We present the IMPACT system - Intelligent Mining Platform for the Analysis of Counter Terrorism. IMPACT is a knowledge discovery system designed to provide new methodologies for the identification and detection of terrorist related events. Unlike conventional data mining and knowledge discovery systems, the tracking and targeting of terrorist related events is much more complex. For example, the event associated with an individual purchasing fertilizer and renting a truck in the same day does not qualify that individual as a terrorist, yet other factors related to that individual could significantly raise the chances of the likelihood that event being a terrorist related. The goal of IMPACT, therefore, is to filter out and identify events from diverse and heterogeneous data sources in order to identify potential terrorist event.

I. Introduction

The first component of IMPACT is an agent based Terrorist Profile Generation Facility that aggregates and pre-computes key quantitative or qualitative data indices multi-dimensionally to dynamically generate an N-dimensional OLAP data cube for analysis. Profiles are defined by the analyst using a JAVA-compliant Web browser. Once the profile is submitted, the appropriate OLAP analyses are continually computed and appropriate notifications are sent to the analyst.

The second component of IMPACT, the Dynamic Data Miner, provides data mining capabilities that could not be identified solely by the OLAP data cube. The Dynamic Data Miner uses case based, predictive, and inference based reasoning to identify new links and relationships. Clustering methods are deployed where applicable. In addition, we develop a special facility that identifies and links suspicious names, identification numbers, and organizations.

The third component of IMPACT is the Terrorist Network Identification Facility that identifies new relationships by linking disjoint (or seemingly unconnected) subnetworks. This component uses singular value decomposition (SVD), and a variety of graph theoretic related algorithms to prune and identify relevant subnetworks of information.

Finally, the Temporal Link Finder identifies terrorist related events that many somehow be linked by time. We demonstrate temporal reasoning algorithms that can identify and link complex relationships about time.

II. Terrorist Profile Generation Facility

The purpose of a profile is to provide the analyst with the capability to define within a browser a suspicious activity (or modus operandi) within the terrorist domain. Figure 1 provides a rudimentary example of this generation. In figure 1, a basic profile has been defined that states, an entity is suspicious if that entity has all the components required to build a certain class of bombs and if that entity has purchased paramilitary weapons.

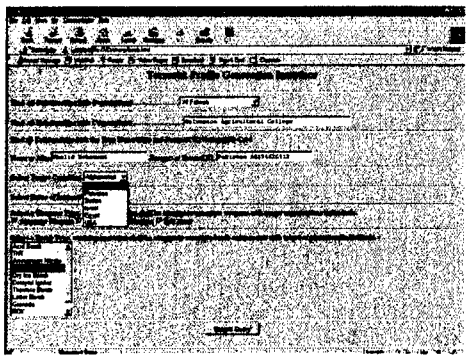


Fig.1: Example of Terrorist Profile

In figure 1, "entity" can be a person, organization, corporation, etc. that has been entered by the analyst or is currently within a profile that is being tracked and is "similar" to the profile that has been submitted. In addition, any entity that is manually entered by the analyst is tracked as "suspicious" within that analyst's profile. In other browser windows, it is possible for the analyst to generate new profiles on the fly.

The code segments below describe the query language used in the implementation of profiles. These rules specify that an entity is deemed a suspicious electronic site if the number of "suspicious" people above that use it exceeds a given threshold, and the site is owned by Group X. This can be expressed in our language as follows:

More-surv-site(S) ***if*** suspicious-user-count(S)=N & N >= threshold & Owner(S)=Company & From(Company,Country) Rogue(Country).

Suspicious-user-count(S)=**COUNT(SELECT DISTINCT** S.Name **FROM** suspicious S, site-users U **WHERE** S.name = U.name)

Owner(S) = **(SELECT** Owner-id **FROM** corporate-db **WHERE** company-name = S).

From(Company) = **(SELECT** Country **FROM** corporate-db **WHERE** company-name like Group X.)

Rogue(Country) = **(SELECT** Country **FROM** rogue-countries).

Once an analyst has specified the profiles and modifications, IMPACT will automatically generate a set of instantiated knowledge based rules, cases, and relations specifying both the users explicit interests (represented by the profiles) as well as the user's implicit interests (represented by the various modified versions of the profiles). This information will then be used as input for additional data mining activities.

The process of expanding a profile involves the following steps:

1. Each profile gives rise to a tree with that profile as the root of the tree.
2. Each node in any such tree is labeled with a query.
3. If we consider a partial tree T of this form, the leaves of T may be "expanded" by making one modification to the query labeling the leaf.
4. Each link in the tree also has an associated "cost." The higher the cost of the path from a root node to a query node, the less interesting the query is.

In short, the search space may be described by iteratively performing the following steps. First, create one tree for each seed query with that seed query as the root. For now, the root of each tree is also a leaf. Now expand each leaf to create children. Repeat this process, ignoring leaves whose cost exceeds a threshold cost.

In general, though it is useful to view the relaxation process in terms of tree expansions, it is not wise to implement them this way. One reason for this is that the same query may end up occurring in multiple

trees, and expanding the same query many times over is wasted effort. We will build upon the well known A* algorithm in the construction of such trees so that we can efficiently support two operations:

1. Find all relaxations of the seed queries whose cost lies below a given threshold cost.
2. Find the top k-relaxations of the seed queries (in terms of lowest cost).

Based on these approximations, we shall be able to represent profiles based on certain major dimensional features that maximize the aggregate to be computed and apply OLAP data cube methods and computations.

III. Dynamic Data Mining

Each profile shall be managed as a "case" within the case base. The "case-base" used by this module consists of a set of past profiles that were deemed suspicious. As new cases rarely match existing cases exactly, this module will attempt to use *profile-modification rules* to convert a set of submitted profiles into an existing suspicious case. Each profile modification rule has a cost. Suppose we apply rule *r1* to a set of profiles *P1*. The result of this application is a new set of profiles *P2*. The smaller the cost of *r*, the more "similar" *P2* is to *P1*. Of course, we may now apply another conversion rule, *r2* to P2, to get a new profile *P3*. The case-based reasoning module will attempt to determine the similarity between a submitted profile and an existing profile by converting a set SubP of submitted profiles into an existing set of profiles, ExP, in the case base.

- *Similar Object Replacement:* The submitted profile is deemed similar to an existing profile if a substitution of an appropriate object in the submitted profile yields the known profile.
- *Case Instantiation:* The submitted profile may correspond to be a "part" of an existing profile.
- *Case Merging:* As stated above, a set of submitted profiles may jointly correspond to a profile in the case base (or the other way around). Similarity based profile retrieval and matching methods must be able to dynamically determine which profiles will be merged. This will expand the range of possible database fields which are relevant to a specific query.
- *Link Addition:* A submitted profile is similar to an existing profile if inserting a valid link causes the profiles to become similar.

IV: Terrorist Network Identification Facility

We will start this section with a scenario that this tool is designed to solve. If all the electronic data that that identifies all the known members of Group X, their relationships with other individuals and organizations, their travel, their purchases (and so on) were described graphically, the result would look like a messy tangled network of links and nodes that would appear nonsensical to the keenest of analysts. Some of the nodes on the network may be innocent individuals, others may be cutout companies, while yet others might be legitimate businesses that have no knowledge of any illegal/suspect activity, but are being used (unbeknownst to them).

The Graph Theoretical Path Generation and Validation Tool provides an environment within which the network connections between all these disparate entities may become untangled, validated, queried, and browsed. In addition, sub-networks that are apparently unconnected will be hypothesized, based on data mining utilities, to be connected. Thus, the objective of this tool is to (1) identify the critical links that exist and (2) hypothesize the existence of new links based on unconnected sub-networks.

The Graph Creation and Update Module component of this tool is responsible for two tasks. First, it will

examine external databases and data sources, and determine which entities are linked to which other entities. Figure 2 below is as an example of such a network and linkage between entities on a network. A pointer back to the original document that generated the link must validate each link on the graph.

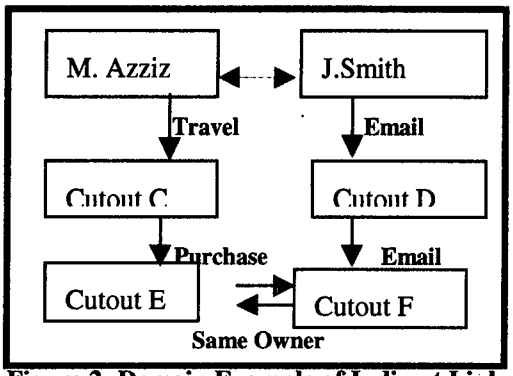


Figure 2: Domain Example of Indirect Linkage

This module will be responsible for managing updates to the network, as new data becomes available. As there may be numerous different links between two entities, and as a single investigation may involve thousands of entities, with millions of connecting links, the task of designing efficient, scalable data structures becomes a formidable one.

We approach the task of generating such graphs using the well known mathematical concept of Singular Valued Decompositions (SVDs). SVDs have been used extensively for clustering where one may want "similar" entities to be clustered together. Based on this clustering method, links between clusters can then be hypothesized. We will outline the approach in the next few paragraphs.

Associated with our graph G representing transactions is a massive, *implicit matrix* M_G. If graph G has k nodes in it, then M_G is a (K x K) matrix. The (i,j)'th entry of this matrix is set to 0 when the i'th entity in the graph G and the j'th entity in the graph G are not directly linked together. If the i'th entity in the graph G and the j'th entity in the graph G are linked together, then this implicit matrix contains the number *of direct edges in G between the* i'th and j'th entities. For example, suppose the i'th entity in graph G represents John Smith and the j'th entity is Mohamed Hashimi. Then the (i,j)'th entry of this matrix M_G is the number of direct links between John Smith and Mohamed Hashimi.

The technique of singular valued decomposition takes the matrix M_G and splits it into three matrices. That is, it rewrites the (K x K) matrix M_G as a product of three matrices of size (K x R), (R x R) and (R x K) such that:

- The product of these three matrices is identical to M_G and
- The (R x R) matrix is a non-increasing diagonal matrix, i.e. all its entries are zero except for the diagonal entries, and those entries are in descending order.

Figure 3 below shows this situation. The diagonal matrix produced by the SVD contains the R most significant links within the network. As one walks down the diagonal, the strengths of the links between the entities involve decreases. Thus, we can capture the most important links and clusters by breaking the large "K x K" matrix into a smaller "R x R" matrix and using the values within the R x R matrix and the rows in the K x R matrix to identify the highly relevant links.

There is a plethora of well-known techniques to efficiently compute singular valued decompositions of massive matrices. In the case when the matrices are sparse, a variant of the SVD called the semi-discrete decomposition may also be used. Once this is done, we may as we have described above, the SVD technique (and its variants) have been successfully applied to a wide variety of applications where different entities need to be "linked" together. These include:

- The well known Latent Semantic Indexing technique developed by Bellcore for indexing and retrieval of massive collections of textual documents – in this application, the goal was to consider that two documents are "linked" if they are on the same topic.
- The organization of multimedia data stores where different multimedia objects need to be clustered together based on "similarity" – here, two media objects are considered to be linked together if they satisfy a similarity requirement.

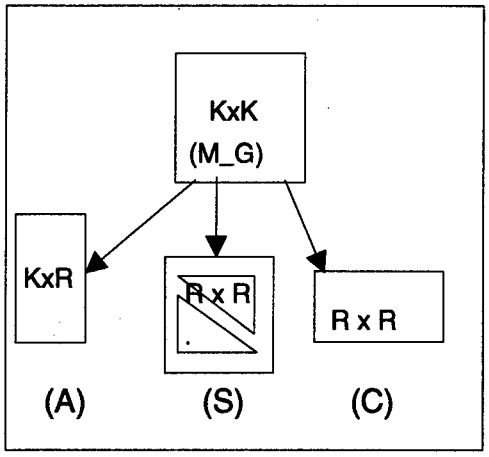


Figure 3: Singular Value Decomposition.

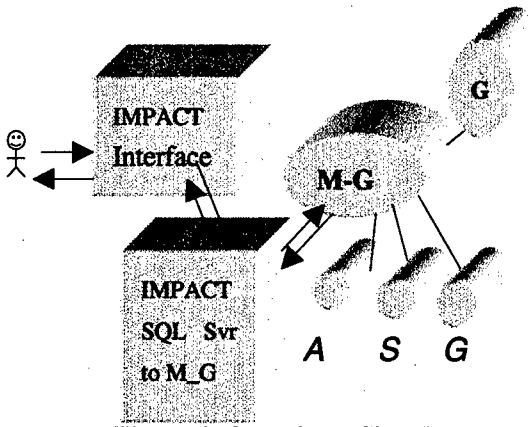


Figure 4: Querying a Singular Value Decomposition Matrix for Link Hypothesis Generation

Suppose, for example, an analyst wants to find the ten "highest cost" paths linking A and B that go through C. Using graphical user interface, he can generate the following query.

SELECT MAX(10) PATH P BY COST FROM GRAPH G WHERE P CONTAINS C.

Or suppose the analyst wants to find the ten highest cost paths linking A and B that go through C, but which do not contain D (e.g. D may be a hardware chain that has been investigated and that may have been completely exonerated). He may express this via the following query.

SELECT MAX(10) PATH P BY COST FROM GRAPH G WHERE P CONTAINS C and NOT(P CONTAINS D).

Cost, in this scenario may be set by ranking the types of edges within the graphs based on their type. (Such as e-mail, relationship, purchase, or similar link.) The implementation of these queries *on top of the underlying graphs* rather than on top of a set of relations (as in classical SQL) requirse:

The identification of a core set of operations on the graph. These core operations include:

- Given a node N in the graph, find all its immediate neighbors.
- Given nodes N1,N2 in the graph, and an integer k, find the k cheapest paths between N1, and N2.
- Given nodes N1,N2 in the graph, and an integer k, find the k most expensive paths between N1, and N2.
- Given nodes N1,N2 in the graph, and a cost c, find all paths between N1 and N2 in the graph with cost less than c.
- Given nodes N1,N2 in the graph, and a cost c, find all paths between N1 and N2 in the graph with cost greater than or equal to c.
- Given nodes N1,N2 in the graph, and a database query condition C on paths, find all paths

between N1 and N2 that satisfy condition C.

It is important to note that using this set of core operations, we can express extremely powerful queries through Boolean combinations of the above. Fortunately, each of the above graph theoretical operations is amenable to implementation through well studied techniques in graph algorithms.

The **Graph Browsing Module** will allow the analyst, to browse the initial graph G and/or its decompositions (G_A,G_S,G_C), effectively. For example, the analyst may wish to zoom in on one section of the graph, and/or view all attributes associated with one or more links. He may also want to merely browse all paths in the graph connecting entities A and B. These types of browsing operations are scalably supported by this module.

- **Link Hypothesis Generation Tool**

In many cases, two or more "clusters" or "groups" or "subnetworks" within a network, detected by the singular valued decomposition (SVD) analysis, may be linked together in some way, unbeknownst to the analyst. The analyst may wish to explore alternative possible groupings (to those produced by the SVD) in one of two modes:

1. He merely asks the SVD to generate other potential groupings that he can browse or
2. He can request an evaluation of the possibility that certain disparate groups are in fact strongly linked.

The task of the Link Hypothesis Generation Tool is to facilitate such explorations by the analyst. Proximity of clustered networks represents how closely related the SVD algorithm believes they are to each other.

When evaluating the possibility of a link between two groups (sub-networks within a graph), the Link Hypothesis Generation Tool will generate different hypotheses to support such linkages, and present these to the user. Hypothesizing that two "groups" (as determined by SVD) are linked is equivalent to saying that SVD should have "merged" these two groups. Generally, speaking, two entities are in the same group (w.r.t. SVD splits) if the distance between the row-vectors in the matrix A described earlier in this proposal is below some threshold value. Given a set of entities, SVD attempts to group them together so that for any two entities in a given group, this property is satisfied. However, such a split may be made in many ways, and SVD may arbitrarily choose one. When the analyst asks that the possibility that two disparate groups are actually one be investigated, it is possible to adapt the SVD to merge these groups together – however, to accomplish the merge, one of two things needs to happen:

- *Some new links should be hypothesized.* This tells the analyst that these new, hypothesized links are potential new areas of investigation (to see if the links hypothesized really can be substantiated through physical evidence), or
- *Some existing links are stronger than they were thought to be.* In terms of SVDs, this means that some existing links were thought to be less strong than they really are. In other words, two entities may have been placed in different groups because they were not thought to be well linked, but further investigation (and further evidence) may prove a more solid link.

In the first case, our Link Hypothesis Generation Tool will hypothesize different sets of links that cause the two groups to be merged, and rank these new link sets in descending order of plausibility – of course, the analyst can change this ranking. In the second case, the Link Hypothesis Generation Tool will identify entities in the two groups that caused the

two groups to be split apart in the SVD. Typically, these two entities will have a relatively low-strength link between them (though not too low). It will then hypothesize that these link strengths are higher than predicted by the original data.

V: Temporal Link Finder

One of the fundamental parameters that must be taken into account by any serious effort at identifying links between multiple entities is the temporal dependencies between events. For example, consider the following sequence of events:

1. John Smith and Mohammed Hashim, known member of Group X engage in a "chat" session in an MCI chat room at 10:30pm EST on July 27, 1997.
2. An electronic funds transfer of $20,000 is made from account #277789018 in the Grand Cayman Bank to account #81910182 of ABC Corp, a major arms dealer and supplier to Group X. Intelligence data indicates that the former bank account belongs to John Smith and the latter to an unidentified individual. This is done on July 29, 1997, at 11:00am EST.
3. An international funds transfer is made from account #81910182 of ABC Inc in Switzerland, to account #91728292 in the Bank of Qatar to another unknown individual. This is done on August 1, 1997, at 9:00am.
4. On August 2, 1997, John Smith and Vladimir Zhirovski exchange email in which Zhirovski reports that Carlos Orojuelo has received payment for the goods.

Examining the above temporal sequence of events in the above scenario, we may lean towards the following hypothesis that the four events jointly indicate a payment (for some unspecified goods).

1. The payment is being made by Mohammed Hashim to Carlos Orojuelo.
2. John Smith works for Mohammed Hashim of Group X.
3. Vladimir Zhirovski works for Carlos Orojuelo.
4. Account #91728292 in the Bank of Qatar is somehow linked to either Vladimir Zhirovski or to Carlos Orojuelo.

The Temporal View Tool allows one to "zero in" on certain temporal patterns or intervals that he is interested in analyzing further. For example, the analyst may request that the SVD network be restricted only to events that occurred between July 25, 1997, and August 5, 1997 and that all sequences involving a payment (direct or indirect) between two people be reported, where the payment exceeds $5,000, and where at least one "questionable" bank is involved in the transaction.

In effect, this analyst specifies a temporal view – this view reflects those aspects of the Terrorist Network Identification Facility that she is interested in examining more closely. Temporal views, in effect, allow the analyst to provide logical specifications reflecting his expertise in the domain of investigation – in response, she expects the Temporal View Tool to effectively group together sequences of such events into "groups" or "clusters" that reflect possible transactions of the sort she is interested.

In order to implement temporal views, we enhance the well known view mechanism in commercial database systems to accommodate viewing networks. In commercial databases, views are specified by conditions over relational data. However, in our case, the conditions must be evaluated over a SVD network, not over relational databases, because all interactions being monitored during the investigation are stored largely within the SVD network we have described earlier. We use a graphical user interface through which the analyst may specify her

constraints. For example, the criteria articulated in the proceeding example may be specified as follows:

SET VIEW V1 **TO** SVD-Net(A) **REFINE BY** **Time** > July 25, 1997 **AND Time** < Aug. 5, 1997.

This causes V1 to be a view that reflects all transactions in the SVD network that occurred between the stated dates. To further focus on transactions involving transfers over $5,000 and involving "questionable" banks or organizations, this definition may be further refined as follows:

SET VIEW V2 **TO GROUPS GWHERE** G **HAS PATH** P **AND** P **HAS LINK** L **AND** Transaction(L)=electronic transfer **AND** Amount(L) > 5000 USD **AND** G **HAS NODE** N **AND** Questionable(N).

In the above specification, the predicate Questionable(N) may be defined as a standard view on a relational database system such as the Oracle Web Server on which our Phase I development has been carried out.

IV. Conclusions

This paper identifies some of the key technologies that can be deployed in the knowledge discovery and data mining process with respect to the data mining domain. Such a task is complicated due to the nature of the domain. However, with the vast increase of integrated database systems, much work can be accomplished through the deployment of on-line profiles or modus operandi.

References

[1];Rakesh Agrawal, et. Al Proceedings of the 1993 ACM SIGMOD international conference on Management of data , 1993, Pages 207 – 216

[2] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer andR. Harshman. (1990) "Indexing by Latent Semantic Analysis", Journal of the American Society for Information Science,
Vol. 41, pages 391-407

[3] Sylvain Létourneau; Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence , 1998, Page 1178

[4] Shian-Hua Lin, Chi-Sheng Shih, Meng ChangChen, Jan-Ming Ho, Ming-Tat Ko, and Yueh-Ming Huang; Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval , 1998, Pages 241 – 249

[5]Sunita Sarawagi, Shiby Thomas, and Rakesh Agrawal; Proceedings of ACM SIGMOD international conference on Management of data , 1998, Pages 343 – 354

[6] Usama Fayyad, and Ramasamy Uthurusamy; Commun. ACM 39, 11 (Nov. 1996), Pages 24 –26

[7] Clark Glymour, David Madigan, Daryl Pregibon, and Padhraic Smyth; Commun. ACM 39, 11 (Nov. 1996), Pages 35 - 41

Supporting Thorough Knowledge Discovery With Data-Aware Visualizations

Terrance Goan
Stottler Henke Associates Inc.
1660 South Amphlett Blvd., Suite 350,
San Mateo, CA 94402 USA
Email: goan@shai-seattle.com

Laurie Spencer
Stottler Henke Associates Inc.
1660 South Amphlett Blvd., Suite 350,
San Mateo, CA 94402 USA
Email: lauries@shai-seattle.com

Abstract

Much of the research in the area of Knowledge Discovery in Databases (KDD) has focused on the development of more efficient and effective data mining algorithms, but recently issues related to Human-Computer Interaction (HCI) have drawn significant attention. One very promising set of work seeks to improve system usability through the use of direct manipulation techniques to provide for the flexible utilization of data and tools. In this paper we describe our efforts to compliment this work by developing result visualization techniques for a variety of classes of data mining algorithms that act not only as end products, but provide direct inputs to future iterations in the KDD process. By building this feature on top of a unified, persistent, and visualizable knowledge and workflow representation system, we provide users with a high degree of flexibility while simultaneously permitting thorough and systematic knowledge discovery processes.

Key Words: knowledge discovery, human-computer interaction, visualization.

I. Introduction

Massive datasets arise naturally as a result of automated monitoring and transaction archival. Military intelligence data, stock trades, bank account deposits and withdrawals, retail purchases, medical and scientific observations, and spacecraft sensor data are all examples of data streams continuously logged and stored in extremely large volumes. Unfortunately, the sheer magnitude and complexity of data being stored acts to conceal valuable information that may lie below the surface, making manual analysis infeasible. Even computer-aided statistical analysis techniques are currently of limited practicality in fusing data from such vast resources. However, over the past decade, researchers have responded by developing KDD (Knowledge Discovery in Databases) tools that can greatly improve prospects for uncovering interesting and useful patterns from such large data collections.

In recent years, substantial progress has been made in developing highly efficient and effective data mining algorithms and information rich data visualization techniques. These techniques can yield important information about hidden patterns in very large datasets, but, in the end, a user's ability to uncover interesting and useful patterns remains limited by human cognitive capacity, and that capacity remains easy to overwhelm with today's KDD tools.

The need for more attention to the human-computer interaction aspects of KDD has not gone unnoticed. A review of the recent research literature reveals a number of efforts to improve the usability of knowledge discovery and data mining systems, including providing: access to numerous KDD tools through a single interface, data mining algorithm selection advice, user guidance through automated planning, and impressive exploratory data visualizations. In this paper we describe our efforts to compliment these previous efforts by making iterative and interactive KDD processes more intuitive through the use of data-aware visualizations that enable a set of novel exploratory operations. The primary goal of our investigation is to reduce cognitive load on the users, and free them to explore their data in an efficient, systematic and thorough manner.

Of particular interest to us is improving the usefulness of *integrated* systems that seek to support the entire knowledge discovery process: target dataset creation, data cleaning and preprocessing, data mining, and results reporting. Since the exploratory process associated with KDD proceeds in a data-driven manner, it is crucial that these tools are seamlessly integrated so as to allow flexible utilization of tools and operation chaining. Especially useful to intensive knowledge discovery processes is the ability to intuitively utilize the results of data mining operations in subsequent exploration steps. To date, such capabilities have largely been

neglected to the significant detriment of users.

IKODA (Intelligent KnOwledge Discovery Assistant) utilizes "data aware" visualization techniques that lie atop a unified and persistent knowledge representation and provide a mapping between graphical objects and the underlying data resources. This approach results in the ability to perform direct manipulation operations such as drag-and-drop transfer of data between tools and a unique capability to explore data mining results. Unlike previous integrated KDD systems, our IKODA's visualizations act as interactive tools rather than simple information displays. More specifically, IKODA's visualizations of data mining results (e.g., decision trees and automatically created data clusters), can be manipulated and used directly to form new datasets that feed future data mining operations. The resulting "recursive" knowledge discovery capability represents a substantial step forward in reducing KDD tool complexity while simultaneously increasing flexibility and efficacy.

In this paper we describe how the HCI techniques being developed for IKODA compliment existing interactive visualization techniques to provide for a very flexible KDD capability. Further, we will describe how IKODA's extensible and persistent knowledge representation supports thorough exploration by allowing the user to label, organize, and utilize meaningful data models throughout future knowledge discovery sessions.

II. Exploratory Data Visualization

Task oriented data mining is an important capability, and has received a significant level of attention in the research community. However, it has been recognized that the formation of precise knowledge discovery goals is often difficult and time-consuming [1]. For this reason, it is important to support data-driven exploration that can provide users with the basis for subsequent goal-driven data mining. To accomplish this, it is useful to exploit the human's unmatched perception and reasoning abilities in detecting patterns through the utilization of exploratory data visualization techniques [2].

The objective of data visualization techniques is to represent very large numbers of data items in a way that facilitates the detection of interesting and potentially useful patterns. Exploratory visualization techniques take many different forms including scatter plots [3], parallel coordinates [4], icon-based techniques (e.g., [5]), hierarchical techniques (e.g., [6]), and distortion-based techniques (e.g., [7]).

Another class of these visualization techniques, and the one most related to our effort, are those that are dynamic and interactive (e.g., filtering (see [8] and brushing [3]). Derthick [9] presents an interactive visualization environment called Visage that has provided significant inspiration for our work on IKODA. Visage makes pervasive use of direct-manipulation techniques to provide the user with the means to explore data via several tightly coupled and customizable visualization tools.

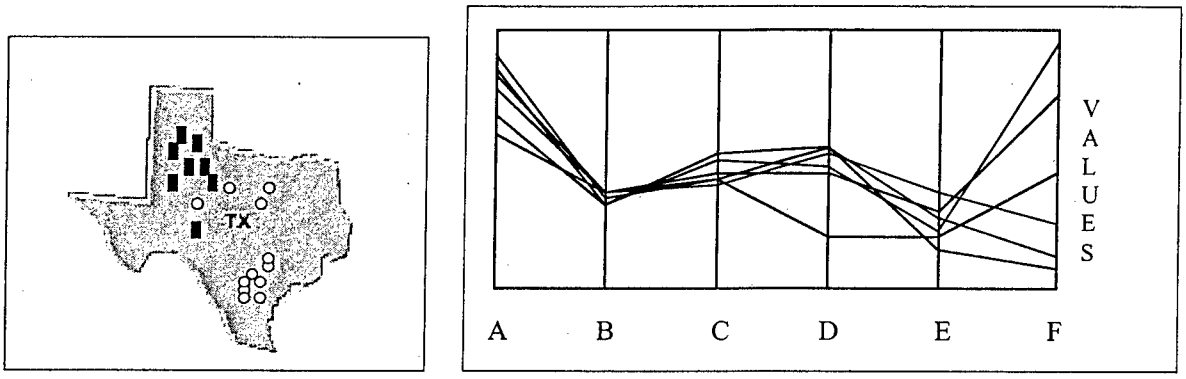


Figure 1a & 1b. Figure 1a shows a "data" visualization technique that would be compatible with techniques such as brushing, where the control on the display of particular points depends on user set threshold values on attribute values. Figure 1b shows an example of a parallel coordinates visualization of data points. In this display, the lines correspond to attribute values for individual data points.

While we share the goal of utilizing direct manipulation pervasively in IKODA to provide an intuitive means of interacting with multiple tools and visualizations, we augment data visualization with "data mining result" visualization (henceforth "result visualization") and exploration. Figures 1a and 1b clarify this distinction. Whereas data visualization techniques are used to explore the relationships between large volumes of data points, result visualization techniques enable exploration of the data models output by data mining algorithms.

III. Usable Data Mining Result Visualizations

The primary focus of our research effort has been to compliment the excellent previous work done in developing interactive data visualizations, by creating interactive data result visualizations. Traditional data mining result visualizations are largely static and act much more as a fixed end result of an autonomous procedure, than as intermediate steps within an ongoing knowledge discovery process. Integrating interactive result visualizations is a crucial step in creating a truly exploratory environment.

We separate interactive functions into two classes; recursive data mining operations, and result manipulation operations. Recursive data mining refers to the process of building deriving models such as decision trees, and then using the natural data partitions that are described by these models as the input for the next round of knowledge discovery operations. Result manipulation operations refer to actions that change the nature of a particular result in order to develop a deeper understanding of its nature. We discuss these two classes of operations in detail below.

IV. Recursive Data Mining

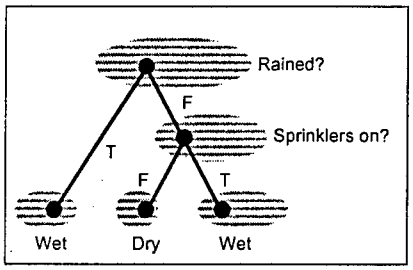


Figure 2. Decision Tree "result" visualization where the gray regions represent the underlying data partitions.

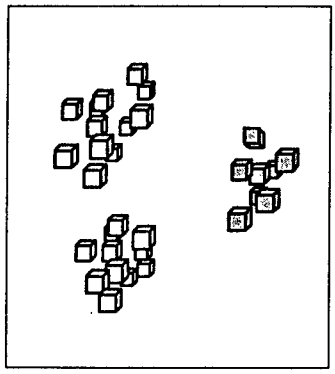


Figure 3. Three identified clusters represent an obvious opportunity for recursive data mining.

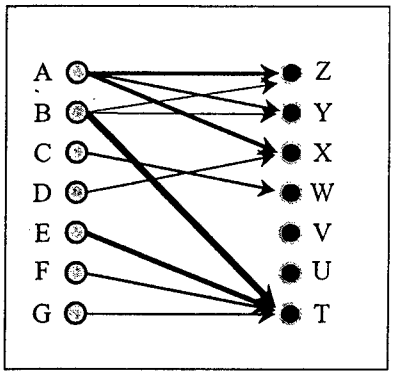


Figure 4. This visual representation of 2-term association rules offers three types of data sets. These are associated with a rule's antecedent, conclusion, and the intersection of the two.

Figures 2, 3 and 4 demonstrate the data partitions created by different types of data mining algorithms. In each case, the derived models provide a useful means for directed search in smaller subsets of the original dataset. Decision trees, for example, are developed by repeatedly dividing the training set based on differences in selected attributes. Clearly, parts of a generated decision tree may be particularly interesting to a user and therefore deserve further examination. IKODA makes the utilization of the generated data partitions easy by allowing the user to select, drag, and drop collections of internal nodes of a decision tree (representing the underlying data subsets) into other tools (e.g., data visualization and data mining tools). In fact, the range of recursive data mining operations that may be useful in a given search is very broad. Some of these include:

- Labeling identified data clusters and utilizing them to form decision trees in order to develop a concise description of how those clusters differ

- Examining subcluster structure by rerunning the clustering algorithm on particular clusters or unions of clusters
- Uncovering data clusters in the data behind high confidence association rules

Obviously, a user could accomplish these operations by other means, but providing the capability to flexibly explore the derived data mining results through direct manipulation leads to substantial time savings and therefore an improved ability to conduct thorough investigations. We will discuss the chaining of recursive data mining operations further after the next section in which we describe the second class of interactive result visualization operations.

V. Result Manipulation

Another natural class of exploratory operations on our result visualizations are those that manipulate the nature of the derived data model. For example, the data partitioning nature of decision trees makes them malleable in a number of different exploratory ways. A user might select a particular interior node and change its characteristics to determine how it would change the structure of the subtree. In particular, one might conduct the following operations before reconstructing the subtree:

- Alter the split attribute at the root of the subtree
- Manually re-discretize variables
- Remove attributes from the data subset
- Form new composite attributes

A second group of manipulation operators under development will work with IKODA's K-means clustering algorithm. In particular, the user will be able to merge and/or divide clusters through direct manipulation. These operations would act to alter the similarity measures used to conduct our clustering and therefore provide the user with the ability to provide IKODA with feedback as to the quality of the developed clusters.

VI. Workflow Management Support

In order to succeed at our goal of providing users with both a highly flexible and usable exploratory KDD tool, it is necessary that we provide some means for tracking the many paths a user may follow in pursuit of thoroughly understanding his data. For this reason we have incorporated a data-aware process monitoring system that allows users to keep track of their search paths and avoid repeating operations unnecessarily.

Figure 5 shows how a particular exploratory path can be represented and reused in IKODA. Here the user forms an initial dataset from which he generates 2-item association rules. The user identifies what appears to be some interesting clustering of high confidence rules and decides to examine the associated data points in an alternative manner, so he selects a collection of rules and drops them into IKODA's K-means clustering tool. This step creates a number of clusters, two of which are worth further exploration. At this point the user seeks to understand how the two classes differ, so he drags those clusters to our decision tree algorithm. The user can then manipulate the structure of the tree, and thereby create new tree instances, in order to better understand how the clusters differ. Note that through our workflow management diagram the user can also return to previous steps and continue his exploration from there.

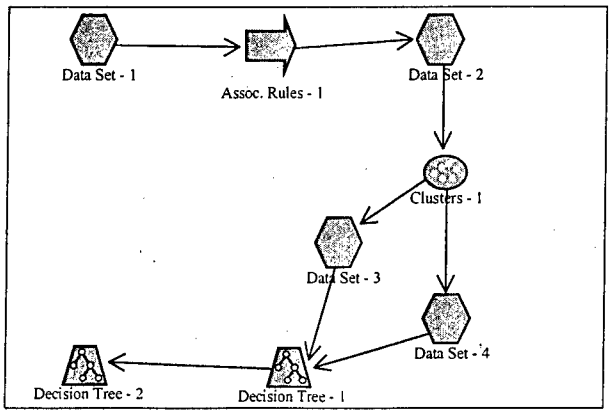


Figure 5. IKODA's workflow monitoring system follows the same direct manipulation principles used elsewhere. This display allows users to keep track of their search path. Note that data sets 2, 3 and 4 are created implicitly by the user's actions.

VII. Future Directions

The development of IKODA is still in an early stage and much remains to be done. SHAI is currently working to expand IKODA's set of data mining algorithms and result visualization techniques. We expect that for each data mining algorithm class there will be a number of visualization styles that can be utilized to facilitate interactive exploration. We are also seeking to expand the range of result manipulation operations that are supported by IKODA. In addition, SHAI is examining the range of classes of data mining algorithms whose results can be fruitfully explored via visualization. It is clear at this point that certain classes of algorithms are more amenable to direct interaction than others. Finally, we hope to further improve IKODA's interaction with the user by integrating techniques that will allow users to provide the resident data mining algorithms with domain knowledge in order to guide their search and improve utility (see [10] & [11]).

VIII. Conclusions

In this paper, we have argued that KDD is an interactive process that can benefit greatly from the better exploitation of the perception and reasoning capabilities of the human user. In particular, we showed how data visualization techniques can be complimented by the use of interactive data mining result visualization techniques. The described techniques provide users with an efficient means of exploring derived data models that therefore allows for an improved understanding of identified patterns. Finally, we illustrated how several data mining tools and visualization techniques can be used together to explore data for useful patterns.

References

[1] Engels, R. Planning Tasks for Knowledge Discovery in Databases; Performing Task-Oriented User-Guidance. In *Proceedings of Knowledge Discovery in Databases*, 1997.

[2] Keim, D. Visual Techniques for Exploring Databases, *Tutorial, Knowledge Discovery in Databases*, 1997.

[3] Becker, R. A., and Cleveland, W. S. Brushing Scatterplots. *Technometrics* 29(2), 1987.

[4] Inselberg, A. and Dimsdale, B., "Parallel Coordinates: A Tool for Visualizing Multi-Dimensional Geometry," *Proceedings of the First IEEE Conference on Visualization*, 1990.

[5] Pickett, R. and Grinstein, G. Iconographic Displays for Visualizing Multidimensional Data, *In Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, 1988.

[6] Johnson, B. Visualizing Hierarchical and Categorical Data, *Ph.D. Thesis, Department of Computer Science, University of Maryland*, 1993.

[7] Sarkar, M. and Brown, M. Graphical Fisheye Views, In Communications of the ACM, 37(12), 1994.

[8] Fishkin, K. and Stone, M. Enhanced Dynamic Queries via Movable Filters, In *CHI Conference Proceedings*, 1995.

[9] Derthick, M., Kolojejchick, J. A., and Roth, S. F. An Interactive Visualization Environment for Data Exploration. In *Proceedings of Knowledge Discovery in Databases*, 1997.

[10] Goan, T. Supporting the User: Conceptual Modeling & Knowledge Discovery. In *Proceedings of ER'97 Preconference Workshop - Conceptual Modeling: Historical Perspectives and Future Directions*, 1997.

[11] Pazzani, M. Comprehensible Knowledge Discovery: Gaining Insight from Data. In *Proceedings of the First Federal Data Mining Conference and Exposition*, 1997.

Semi-automatic Integration of Knowledge Sources

Prasenjit Mitra
Department of Electrical Engineering
Stanford University
Stanford, CA, 94305, U.S.A.
mitra@db.stanford.edu

Gio Wiederhold, Jan Jannink
Computer Science Department
Stanford University
Stanford, CA, 94305, U.S.A.
{gio, jan}@db.stanford.edu

Abstract *Integration of knowledge from multiple independent sources presents problems due to their semantic heterogeneity. Careful handling of semantics is important for reliable interaction with autonomous sources. This paper highlights some of the issues involved in automating the process of selective integration and details the techniques to deal with them. The approach taken is semi-automatic in nature focusing on identifying the articulation over two ontologies, i.e., the terms where linkage occurs among the sources. A semantic knowledge articulation tool (SKAT) based on simple lexical and structural matching works well in our experiments and semi-automatically detects the intersection of two web sources. An expert can initially provide both positive and negative matching rules on the basis of which the articulation is to be determined and then override the automatically generated articulation before it is finalized. The articulation may be stored or generated on demand and is used to answer customer queries efficiently.*

Keywords: information integration, knowledge discovery

1 Introduction

1.1 Need for an Ontology Algebra

Queries posed by end-users can, often, not be answered from a single knowledge source but require consulting multiple sources. When those sources are independent, the terms they use are not constrained to be mutually consistent. The semantic heterogeniety of these sources must be resolved in order to present a reliable and consistent view of the world.

The traditional approach to dealing with multiple knowledge sources is to create a unified schema or to merge the contents of these sources into an unified knowledge base [1], [2], [3]. Such an effort has two phases, first merging all the entries based on identical spelling and then manually resolving any recognized semantic mismatches. For instance, the erroneous match of *nail*, used in a human anatomy, with its use in carpentry must be undone. More complex are cases where definitions change over time, for instance types of cholesterol as disease-causing agents.

The merging approach of creating an unified source is not scalable and is costly. The process must be repeated when the sources change [4]. Furthermore, in certain cases a complete unification of a large number of widely disparate knowledge sources into one monolithic knowledge base is not feasible due to unresolvable inconsistencies between them that are irrelevant to the application. For a particular application, resolution of inconsistencies between a pair of knowledge sources is typically feasible, but it becomes nearly impossible when the objective is undefined and the number of sources is large.

Due to the complexity of achieving and maintaining global semantic integration, the merging approach is not scalable. We have adopted a distributed approach which allows the sources to be updated and maintained in-

dependent of each other. Articulations are generated between the sources that serve specific application objectives. Only the related articulations need to be updated when a term in the intersection of sources are changed and other articulations will remain as they were.

1.2 The Ontology Algebra

An *algebra* has been defined to enable interoperation between ontologies [5]. *Ontologies* are knowledge structures which explain the nature and essential properties and relations between terms present in a knowledge source. The *operators* in the algebra operate on selected portions of the ontologies. *Unary* operators like *filter* and *extract* work on a single ontology and are analogous to the *select* and *project* operations in relational algebra. They help us define the interesting areas of the ontology that we want to further explore. *Binary* operators that take as input two ontologies and outputs another ontology include *union*, *intersection* and *difference*. *Intersection* is the most crucial operation since it identifies the articulation over two ontologies.

1.3 Model of Articulation

In our model of articulation there are the sources, maintained autonomously by their experts, and applications, which need to use these sources. Linkages between sources are achieved through articulation contexts, which contain the terms that are needed to reach the sources and the rules which resolve semantic differences among them. The articulation contexts are created and maintained by articulation experts. For example, if an application has to access information from city maps, using local coordinates ranging from A1 to M19, and information from images, using latitude and longitude, the expert will provide the matching rules. When local maps are reissued, the coordinates will change if the city has grown, and these rules must be updated. Since the sources, say the local map, can be accessed remotely, we do not need to move all of the map detail into a knowledge base, as long as we can refer to its contents reliably. That means we do not need to update the coordinate mapping when the map is updated, say to indicate new buildings, but only when a major revision which changes its coordinates is made.

We find similar cases of interoperation requirements in purchasing of goods from multiple sources, in shipping, in genomics, etc. In all these instances we can also identify experts who must handle the interoperation of the diverse sources, although they have not been provided with specific tools for their task.

1.4 Organization

Section 2 discusses a tool (SKAT) that computes the articulation and introduces an example application based on government websites of NATO countries. Section 3 discusses the issues involved in computing the intersection and the techniques used in matching. Section 4 highlights the results obtained by matching two example NATO graphs. Section 5 points out the other uses of the matching tool. The last two sections conclude the paper and acknowledge other contributors to the work.

2 The Semantic Knowledge Articulation Tool

The articulation between two ontologies is determined using a semi-automatic *articulation tool*(SKAT). We use a subset of KIF [6], a simple first order logic notation to specify the declarative rules. The steps involved in computing the articulation are as follows:

1. The expert supplies SKAT with a few initial rules which indicate the terms that need to be matched and ones that must not be matched. For example, a rule

   ```
   (Match US.President FRG.Chancellor)
   ```

   would indicate that we want the US President to be matched with the German Chancellor. Similarly, a rule like

(Mismatch Human.nail Factory.nail)

would indicate that we do not want the term nail from the Human ontology match with the term nail in the Factory ontology. Apart from declarative rules, the expert has the option of supplying matching procedures that can be used to generate matches.

2. SKAT suggests matches and the articulation based on the supplied matching rules and the matching procedures approved of by the expert.

3. The articulation expert a) approves, b) rejects, or c) marks as irrelevant the suggested match or the rule used to compute the match.

4. SKAT now creates the correct rules and computes an updated articulation. The knowledge gained from the rejected or irrelevant rules and matches is stored by SKAT so as to avoid suggesting the same matches later.

2.1 An example: NATO websites

In order to demonstrate the concepts we have built a SKAT prototype that can be used by an application to identify the articulation between multiple web sources. Websites can be thought of as structured as a graph with a root page which has links to other related pages. Each website is structured differently and loosely represents an ontology. Initially the labelled graph structure of each website is constructed where each page is a node and all the links found on the page are modelled as outgoing arcs. A *label* is constructed for each arc from the title of the page that it points to and the anchor text found along with the link. We will illustrate our algorithm using examples that have been extracted from the government websites of NATO countries and show the matched nodes of the graph that constitute the articulation. (Figures 1, 2).

3 Intersection and Matching of Ontologies

The Intersection operator takes two ontologies and finds the correspondence of terms obtained from one ontology with that obtained from the other based on a set of rules. A major process in determining the intersection is to find this correspondence or *matching*. We will highlight the issues in matching and their solutions using object graphs obtained from the websites of NATO countries. It might be worthwhile to explore the types of mismatches that exist between ontologies that need merging. The typical mismatches that exist in such object graphs are as follows:

- *Term Semantic Mismatches*: these types of mismatches occur because two terms from two different ontologies refer to different concepts. Alternately, two different terms obtained from different ontologies might refer to the same concept. For the purposes of this work, we assume that within one particular ontology the same term consistently refers to the same concept.

- *Structural Mismatches*: these types of mismatches occur because the same term in one source matches multiple terms in another and causes one node in a graph match with many in the other. For example, the Prime Minister of a country might be holding the defence ministry too, whereas, in some other country the Prime Minister and the defence minister are different. In this case the node in the first graph will match against two nodes in the second. In order to allow for such mismatches between ontologies we allow a node in one graph to match multiple nodes in the second.

- *Instance Mismatches*: these mismatches occur because in an instance of a class in one source is not an instance of the same class in the second source. For example, one university considers its grad-

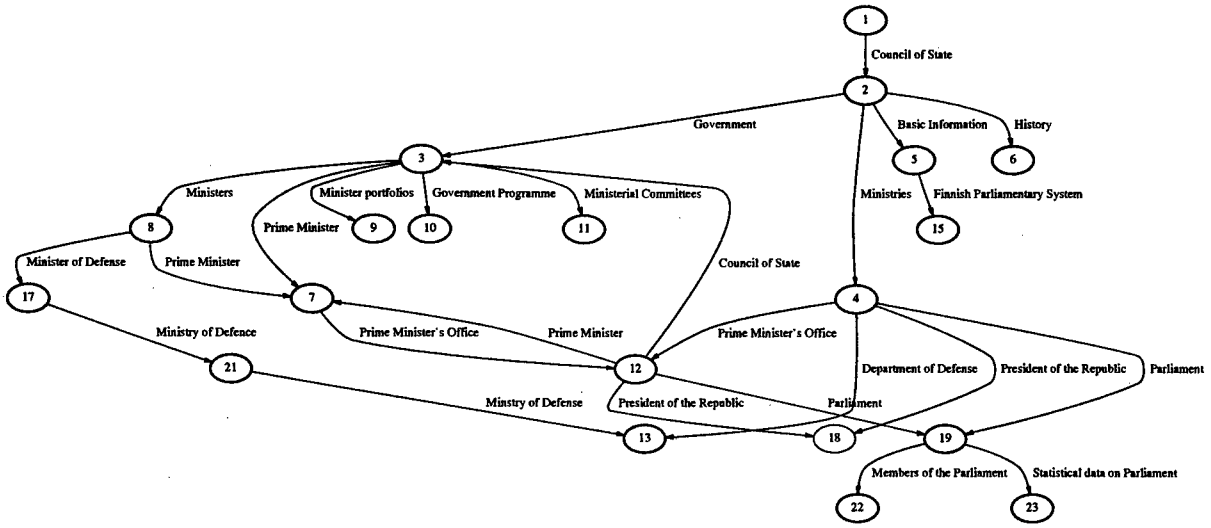


Figure 1: Partial Graph of Finland

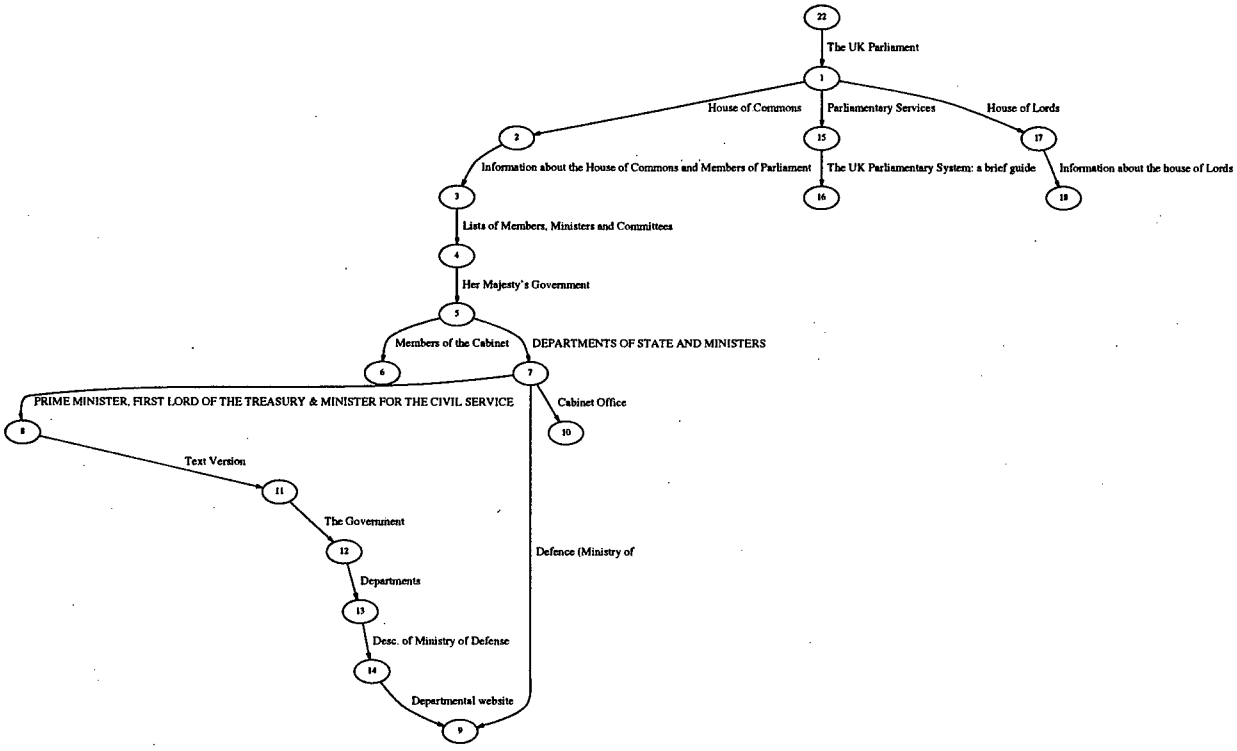


Figure 2: Partial Graph of UK

uate students who hold assistantships as its employees, whereas another one does not. The US President is a part of the the government, however, the Finnish President is not. The Finnish Prime Minister is the head of the government and the President is a ceremonial head of state. The articulation rules should explicitly specify what matches we want to generate in such cases. In the absence of articulation rules, no matches will be generated.

- *Granularity Mismatches*: these occur when we have two matching nodes and the great grandchild of a node matches with the grandchild of the other node. This results because in the first graph a concept has been organized into a more elaborate hierarchy than in the second one. The expert can conservatively choose not generate any match between the intermediate nodes in the two graphs or decide to allow matching both the child and grandchild of the node in the first graph with the child node in the second one.

3.1 Expert Aided Matching

To generate the articulation the two ontologies need to be matched based on a set of *matching rules* specified by the expert. We do not automatically assume that terms from two different contexts match. This is necessary to avoid errors that occur due to the simplistic assumption that similarly spelled words have precisely matched meanings. If the expert believes that terms occurring in two different contexts are the same and relevant to the application, the expert can indicate that these terms mean the same across ontologies. The system will then generate the required matching rules. It is, however, our intent to generate an intersection that is minimal, i.e., it should contain only as much information as is necessary to compose knowledge from the two sources and answer queries posed by a particular application. We believe this approach will increase the precision of answers and reduce subsequent maintenance cost.

However, in cases where the ontologies are very closely related, most terms spelled the same might be referring to the same concept. In such a case, as an optimization, the expert might initially point out those terms spelled the same but are semantically different and then write a rule saying that whetever has not been indicated as mismatches till now are matches. In our example with NATO graphs we use the latter optimization.

3.2 Rule Based Semantic Mismatch Resolution

We envisage our higher-level rules to preprocess the terms in the labels and to determine the similarity of the terms. Such rules can declare global matching operations, as matching specified terms, requiring a dictionary or table lookup, or the use of a procedural function which matches terms. Such resources can be designed by the expert to satisfy recurring needs. However, rules that are sufficiently general in nature, especially procedural functions, can generate specific false matches which must be rejected or marked as irrelevant by the expert. If the rules are tuned towards the specific application contexts, fewer false matches will be generated.

3.2.1 Preprocessing Rules

SKAT initially attempts to match nodes in the two graphs based on their labels. Certain erroneous labels, especially, in the case of a single node having multiple labels, may be weeded out. In the graph for Canada, which has not been included due to space constraints, we had a node labelled 'Parliamentary System' twice and 'Senate' once and hence the former label was selected using a simple voting scheme. A better way is to analyze the document using IR techniques and determine what the referenced document contains. For most cases keeping both labels and matching with either to detect a correspondence does not generate too many false matches. Once again, the expert

indicates whether we take the more conservative voting approach or the more generous approach of keeping all labels.

Another important preprocessing step is the removal of stopwords and stemming of words to their root words. The expert can choose and edit a list of stopwords and either provide a stemming procedure or specify a table (or individual rules) of root words.

SKAT uses rules specified by the expert to resolve semantic mismatches between two ontologies. In our example, the expert provided a matching procedure that simply matches terms from two ontologies if they are spelled similarly. However, before such a procedure is called, the expert had to take care of certain issues which otherwise would have produced false matches.

3.2.2 Context Identifier Tagging Rules

For the instance-level semantic mismatches as discussed above, we require to differentiate between the President of the US and that of Finland since, they are semantically different, yet the same term 'President' might have been used for both in the two graphs. This is achieved by a set of preprocessing rules, that simply indicate which terms need to be labelled with the context identifier tag. The labels of the nodes that pertain to the presidents of the two countries in the two graphs are tagged with the name of the contexts, i.e., they now become US.President and Finnish.President.

3.2.3 Context Identifier Removal Rules

The matching is performed based on a certain criteria. In our example graphs while matching between countries the expert might want to match the parliament nodes of two countries e.g., we want to match the node labelled 'Finnish Parliamentary System' with that labelled 'The UK Parliamentary System' in the graph pertaining to Finland. Therefore, a set of preprocessing rules can be supplied that reduce the labels such as 'Finnish Parliamentary System' to 'Parliamentary System' and thereby enable the matching.

3.2.4 Term Based Matching Rules

In our example an on-line dictionary or table specified by the expert can be searched to find matches among the terms. The terms that match in the two labels are given weights based upon their frequency of occurrence in the sources and other heuristics. A similarity score between two labels is calculated based on the match between terms in the labels. If the similarity score is above a threshold then the labels are matched.

Apart from the simple rules that simply mention the two terms that should be matched, the expert can supply more complex rules.

```
(Instance-Of Country UK)
(Instance-Of Country Finland)
(=> (and (Instance-Of Country ?Country1)
         (Instance-Of Country ?Country2))
    (Match ?Country1 ?Country2))
```

The first two sentences indicate that UK and Finland are countries and then provides a general purpose rule to match two countries. This has the added advantage that in order to match all countries to each other we do not have to list all combinations of matching rules. When we want to add another country all we need to do later on is to add the information that that country is an instance of Country.

3.2.5 Structure Based Matching Rules

Matching rules can also be based upon the structural similarity of the graphs. Parts of ontologies, represented by sub-graphs, can be matched based upon the similarity of their hierarchical structure. The matching rules are expressed as functions which take in the entire graphs and generate the correspondence between nodes of the two graphs.

Matching based solely upon structural similarity works well for sources that are structurally very similar. However, between most ontologies there is a fair degree of structural dissimilarity. Thus matching rules based only

on the structure of the ontologies produce a variety of false matches. Therefore, the matching procedure first matches the nodes based on their labels and using this set of matches and the structural similarity, it generates further matches. In the strictest version, two nodes are matched if all their parent nodes and children nodes match and such a perfect match is given more weight.

The expert can specify the number of matches she is interested in generating and if the perfect structure match is not sufficient to identify the articulation then it is progressively relaxed. Nodes that do not match perfectly but have a low weight (i.e., a few of the parents and/or children match) are then accepted as being matched.

3.3 Other Matching Rules

In our example, instead of basing the matching only on the labels, SKAT can analyze the entire documents (i.e., web pages) and try to detect the similarity of the pages based on the words occurring in the page. A standard Information Retrieval algorithm can be used to generate such matches.

3.3.1 Spurious Match Resolution Rules

Techniques, such as automatic stemming, that were used previously can generate spurious matches. Words such as 'minister' and 'ministry' might have been preprocessed to be reduced to 'minister' and therefore, will match. However, for our government articulation we want to preserve the difference between the two. These sanity checking rules can be explicit statements of mismatches like

```
(Mismatch Minister Ministry)
```

provided by the expert that would indicate that we do not want certain matches. The expert can also indicate that certain phrases should not be preprocessed. Sanity checking rules like

```
(=> (and (Instance-of Person ?X)
         (Instance-of Office ?Y))
    (Mismatch ?X ?Y))
```

can also be used.

3.4 Performance Issues

Due to the more complex general purpose processing rules (the ones shown above that involve implication and logical inference), the performance of a system like SKAT can be really slow. Therefore, our prototype implementation of SKAT does not use those types of rules.

The structural matching procedures mentioned above are quadratic in complexity with respect to the number of nodes being matched. In case of very large graphs where such a complexity is unacceptable, structural slices of the graphs are matched against each other. It is expected that terms near the root of one ontology will match terms near the root of another and so on. Thus slicing the graphs should still generate an adequate articulation. The lexical matching is done by sorting all the terms in the labels in each graph and then matching is done in linear time. However, if sorting the terms in the entire set of labels is unacceptably slow for very large graphs, they too can be sliced and then matched.

4 Results

For the example graphs of Finland and UK the following is the match generated by SKAT using a simple lexical and term matching algorithm:

```
"Finnish Parliamentary System"
-> "The UK Parliament"
"Ministers"
-> "Lists of .. Ministers .."
"Government"
-> "Her Majesty's Government"
"Government"
-> "The Government"
"Ministers"
-> "Departments .. Ministers"
"Prime Minister"
-> "Prime Minister, ..  Service"
```

```
"Ministry of Defence"
-> "Defence (Ministry of"
"Parliament"
-> "The UK Parliament"
```

where the first label refers to a node in the Finland graph and the second to that in the UK graph. As we can see, most matches are correct and intuitive. Using the structural matching algorithm we are able to generate the matches between

```
"Ministries"
-> "Departments"
"Ministries"
-> "Desc. of Ministry of Defense"
```

This was after the requirement for perfect structural match was relaxed and granularity mismatches were not resolved conservatively. Since the Government sites and the Department of Defence sites in both the graphs matches, the nodes in the paths between these two nodes i.e. 'Ministries' and 'Departments' were matched. The unwanted match generated 'Ministries' with 'Desc. of Ministry of Defence' is the price we pay for relaxing the perfect match requirement. A more conservative approach would generate no structural match.

5 Other Uses of the Matching Tool

For our examples, the generated articulation consisted of extracted structural information from the government sites of the NATO countries, and the important nodes selected from them. The result is a partial skeleton of the entire website, providing a taxonomy of the site, as well as of governmental structure.

The government websites were laid out in a hierarchical fashion and the portions of the hierarchy that were common across sites were extracted out. The common portion of the second graph can be restructured so that the nodes are arranged according to the hierarchical structure of the common portion of the first graph. This restructuring, creates a view of the second graph on the lines of the first and can be stored. It is now easier to answer user queries that require searching multiple websites since we have reformatted the articulation of the sites into one common structure.

SKAT can be used to extract information from a website by supplying a template graph whose nodes are labelled with terms of interest. For example, a graph constructed from one of the existing government ontologies can be used as a template graph and its articulation with the Finland graph will give us the nodes in the Finland graph that correspond to the terms in the template graph. A simple adaptation of SKAT can thus be used as a template-based querying tool wherein the answer will be structured according to the provided template.

Since SKAT extracts structural information from an ontology it can be used to create a new ontology. If a graph has little structure we can compute the articulation of this graph with an already existing structured graph. Using the articulation we can structure portions of the first graph. Given the huge amount of information present in today's World Wide Web, one can just supply SKAT the root node of a country's graph or a reference ontology and a set of webpages and those pages will be automatically structured.

Once web pages from distinct sources are consistently structured, queries by end-users will be reliably answered. Misleading matches will be avoided and many new matches, that are now based on verified semantic identities will be created.

6 Conclusion

Applications and their decision-makers benefit from broad access to information, but the information is widely dispersed and difficult to integrate reliably.

We are addressing an important problem in the use of the many diverse knowledge sources that are available to our applications.

By keeping the intersection as small as feasible we reduce the maintenance costs for the ap-

plications and maximize the autonomy of the sources. By allowing sources to remain autonomous we can take advantage of the maintenance efforts made by independent, specialized experts.

Tools, as SKAT, to create modest and managable articulations of these sources for well-understood applications allow application experts to maintain the linkages needed for reliable interoperation. Such reliability is a requirement for business-transactions, since we cannot expect human filtering and matching to occur with regular, repeated operations. The investment made once, by the articulation expert, will pay off every time disjoint domains are used to process an order or make a business decision.

7 Acknowledgements

Thanks are due to P. Srinivasan for their help in preparing the draft of the paper and comments and suggestions to improve it.

References

[1] C.A. Knoblock, S. Minton, J.L. Ambite, N. Ashish, P.J. Modi, Ion Muslea, A.G. Philpot, and S. Tejada. Modeling web sources for information integration. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence, Madison, WI*, 1998.

[2] M.R. Genesereth, A.M. Keller, and O.M. Duschka. Infomaster: An information integration system. In *ACM SIGMOD '97*, pages 539–542. ACM, 1997.

[3] A.T. McCray, A.M. Razi, A.K. Bangalore, A.C. Browne, and P.Z. Stavri. The umls knowledge source server: A versatile internet-based research tool. In *Proc. AMIA Fall Symp*, pages 164–168, 1996.

[4] D.E. Oliver, Y. Shahar, E.H. Shortliffe, and M.A. Musen. Representation of change on controlled medical terminologies. In *Proc. AMIA Conference*, Oct. 1998.

[5] Gio Wiederhold. An algebra for ontology composition. In *Proceedings of 1994 Monterey Workshop on Formal Methods*, pages 56–61. U.S. Naval Postgraduate School, September 1994.

[6] M.R. Genesereth and R.E. Fikes. *Knowledge Interchange Format,*. Reference Manual, 1992.

Knowledge Discovery and Data Mining Using an Electro-Optical Data Warehouse

P. Bruce Berra
PBB Systems and Wright State University, Box 246, Keene, NY 12942
Pericles A. Mitkas,
Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523-1373
Raymond A. Liuzzi,
Air Force Research Laboratory/IFTB, 525 Brooks Road, Rome, NY 13441
Lorraine M. Duvall
Ramsey Ridge Enterprises, Box 246, Keene, NY 12942

Abstract

In this research we postulated an Electro-Optical Computer Architecture (EOCA) that could be used t o evaluate the potential for increased performance and functionality of knowledge discovery and data mining systems that deal with very large multimedia data/knowledge bases. The postulated EOCA is composed of a number of individual holographic associative processors that could perform operations in parallel and could house terabytes of data. With regard to text and numeric data mining, we concentrated on association rules and a number of their variations since many of their operations can be common to other data mining techniques such as classification and clustering. We described these techniques mathematically as timing equations. Utilizing these equations as well as the equations that described the EOCA, we assessed the feasibility of implementing such data mining techniques on the electro-optical architecture. We concluded that great potential exists for orders of magnitude speedup in the data mining of very large text and numeric databases. In fact, some of our results indicate that the association rules algorithm can be evaluated in a matter of a few seconds for a terabyte database. In addition, we investigated the feasibility of the execution of image data mining on the postulated architecture. The results were comparable to those discussed above and therefore quite encouraging. While great potential exists, further research and development is required.

Introduction

In recent years considerable demand has developed for user oriented distributed multimedia management information systems that are able to manage terabytes of data. These systems must provide rich and extended functionality so that new, complex, and interesting applications can be addressed. The need for these systems exists in a multitude of fields including medicine, education and training, defense, business, manufacturing, arts and entertainment, space, as well as a number of other important areas. These applications place considerable importance on the management of diverse data types including text, images, audio and video. As these systems have developed, a wealth of data, information and knowledge has become resident within these vast repositories. This has given rise to a variety of new techniques that have as their objective the extraction of knowledge and information from these repositories [THU97].

Knowledge Discovery and Data Mining (KDDM) is the iterative process of efficiently and effectively finding patterns in data which are relevant to end users. The KDDM process incorporates many methods, tools and techniques from multiple fields to produce effective and usable results, ranging from machine learning techniques from the artificial intelligence field to visualization methods from the human computer interaction field to data warehousing techniques from the database world to provide multi-dimensional data analysis. Data mining is the major computational part of the process that provides algorithms for finding these patterns. There are a number of approaches to data mining including association rules, general characteristics and summaries, classification, clustering, temporal or spatial temporal and path traversal patterns [CHE96].

Optics may be able to help solve some of the very large multimedia data/knowledge base problems. Photons, which have some very attractive properties, such as high speed, non-interference, and a high degree of inherent parallelism can advantageously replace electrons in some processing operations. Optical systems can accommodate a large number of parallel, high-bandwidth channels, thus providing solutions to various interconnection problems. In addition, optical storage devices have very high storage densities and considerable research and development activities are underway to develop devices with read rates in the hundreds of megabytes per second range [MIT98a].

In the research reported here we postulated an Electro-Optical Computer Architecture (EOCA) that could be used to evaluate the potential for increased performance and functionality of knowledge discovery and data mining systems that

deal with very large multimedia data/knowledge bases. The postulated EOCA is composed of a number of individual holographic associative processors that could perform operations in parallel and could house terabytes of data. This system was used to assess the feasibility of implementing such data mining techniques on the electro-optical architecture and to obtain order of magnitude performance data.

Optical Storage, Interconnection and Processing

The state of the art of electronic computing enjoys considerable maturity. In contrast, optics as applied to digital computing is very young and has yet to make its mark. One of the objectives of digital optics is to replace electrons with photons whenever appropriate in a computing environment. As discussed above, the motivation for this is that optics possesses some very attractive properties including massive parallelism, high speed, low power consumption and noninterference of light beams [BER89, 90, GUI96].

In terms of storage, optical disks of various types are in wide use because of their large storage densities even though their access times are slower than magnetic disks. However, with suitable modification to read multiple tracks simultaneously, data rates on the order of hundreds of Mbytes/s are possible [PSA90]. Since electronic computers are designed to deal with magnetic disk transfer rates, they will have difficulty with these increased rates. This dictates that we keep the data in optical form and process them to the fullest extent possible so that, on conversion to electronics, the data rate will be within the capabilities of the electronic computer but more content rich. In this way we hope to increase the performance of the system without disturbing the large investment in systems and user software.

The continuing interest in optical memories is well justified by the potential for high-density storage and for parallel access to two-dimensional pages of data. Optical memories can store as much as 8 terabits/cm^3, (i.e., approximately 931 GBytes of information). Using wavelength domain multiplexing this figure can be increased by 2-3 orders of magnitude.

Volume Holographic Memories

Most ultra large database and knowledgebase systems used in knowledge discovery and data mining store data on magnetic or optical disks and employ indexing techniques to avoid or minimize disk accesses. Various clustering and accessing techniques are used to reduce response time. Even so, when the joint requirements of ultra large databases and very short response times are imposed, existing technologies degrade rapidly. In these cases, the ability to call forth and operate on large pages of data in parallel from a page-oriented holographic memory (POHM) would offer a profound advantage over serial operation. The basic concept of page-oriented holographic memory is quite simple. Many small spatially discrete holograms are recorded on a single substrate in a page format that can hold millions of bits per page. They are constructed in such a way that whenever a laser beam illuminates one of these small holograms, the data are read out in parallel in two dimensions. Volume holographic memories can store hundreds of thousands of these pages in photorefractive crystals using a combination of spatial, angular, peristrophic or wavelength multiplexing techniques [HON95, PSA95, PSA98]. An electrooptic or acoustooptic deflector can be used to address any of these stored pages within microseconds.

Since volume holographic memories have large storage capacities they are prime candidates for the storage of large amounts of data and information including multimedia as well as relational databases. Because of their associative nature [MIT94] they are well suited for accessing data at high speeds. The associative mode provides the ability to search the entire contents of the memory by presenting a search argument and receiving the location of the matching elements

It is safe to assume that optical memories and especially holographic memories represent a promising solution for applications requiring high volume storage, such as: knowledge discovery, relational databases, image processing and in general, a number of research issues currently under consideration in the multimedia field. These applications typically require a high degree of parallelism for processing data. Most of the data operations required by these applications are single-instruction, multiple-data (SIMD) operations. Thus, optical memories and parallel computing have a common characteristic, namely parallelism.

In most conventional computer architectures the processing elements are separated from the data store. Usually a storage hierarchy is employed to move the desired data up the hierarchy to ultimate use by the processor. However, in data intensive processes fast memory is generally not available in abundant supply and large data transfer overhead is incurred. In order to mitigate these effects the processor in memory model offers considerable advantage. In this case processors are integrated with the memory and operations are performed in situ with results being the only data transferred out of memory. While this model is very desirable, it has not been fully realized primarily because of the high cost

involved. Examples of systems that move in the direction of this model generally move multiple processors closer to the memory and employ some form of parallel processing. They do not, however, actually integrate processing capabilities with memory. In the case of holographic memory at least part of the desirable attributes of the processor in memory model are realized. That is, the memory tends to be very large which is very desirable for large data/knowledge base applications. In addition, the associative processing capabilities allow for some processing of data in memory; namely searching for data that match given search arguments exactly or, in some cases, finding the best match of images.

In order for holographic memory to completely meet the requirements of the processor in memory model considerable additional capability must be added so that arithmetic as well as logical operations can be performed. However, an intermediate system with a broad range of search capabilities would find wide application in the data/knowledge base field. And even with just the exact match capability many applications can be enhanced. For instance, many complex queries have exact match components that, with some query optimization, can be performed first thus reducing the size of the data/knowledge base needed for further processing. It is certainly true that one can construct queries that are void of exact match components, but the vast majority of queries do have one or more exact match components. And in the case of knowledge discovery and data mining many of the algorithms can be enhanced through the use of count data.

Significant advances in the field of page-oriented holographic memories have taken place over the last five years and several prototypes have been demonstrated. Companies such as IBM, Lucent Technologies, Rockwell, and others have pursued the technology, even though Universities continue to play a crucial role in new developments and innovations.

The team at IBM Almaden is heading the NSIC/DARPA/University/Industry Photorefractive Information Storage Materials (PRISM) and Holographic Data Storage Systems (HDSS) consortium. During the past five years a large variety of materials and system configurations have been tested in a specially designed holographic memory tester [BUR98]. Up to 10,000 data pages have been stored in a volume of 1 cm^3. At resolutions of up to 1,000 x 1,000 (1 Mbit) per page, the total storage density reaches a significant 10 Gbits/cm^3. A system that will employ spatial multiplexing may raise this capacity 50-100 times (with some increase in volume). Even more impressive are the data rates that have been demonstrated: 1 Tbit/sec burst and 100 Gbits/sec sustained. For 1 Mbit pages, the frame rate, that includes the (non-mechanical) access time, must range between 100 kHz and 1 MHz. At these rates, the detector array that receives the holographic memory output becomes the bottleneck. Charge-coupled devices (CCD) designed for display applications are a totally inadequate interface. Schaffer and Mitkas at Colorado State University have explored the use of CMOS smart photodetector arrays that can combine light detection and conversion with some preprocessing, such as demodulation, error control, and even some form of data selection [SCH98a]. A prototype chip was fabricated capable of performing parallel error detection and correction of 2x2 cluster errors at frame rates of 5 MHz [SCH98b]. A full size chip should be able to output corrected data at up to 100 Gbits/sec. Other research teams have considered and implemented CMOS arrays of active pixel sensors.

The media used most frequently include photorefractive crystals (iron-doped lithium niobate, barium titanate, stoichiometric lithium niobate, etc.) or photopolymers. Crystals can be used in a volumetric form while both crystals and polymers can be arranged on a disk form. Companies such as Holoplex, Rockwell, Optitek, and Lucent Technologies have all demonstrated working prototypes at small form-factors (down to a 3x4x5 " black box).

Recording data holographically is invariably slower than reading them. In fact, writing cycles may be several times longer than readout cycles depending on the material and the available optical power.

The main advantage of holographic memories, that is, their ability to perform associative searches, has not been fully explored as yet. We know that associative recall with analog data works nicely and that recent experiments have demonstrated good associative recall when binary and other digital data are used. It is not known, however, to what extent, in terms of total capacity and search argument size, holographic associative processing is effective and reliable. In this work we have taken some small positive steps in the direction of showing that holographic associative processing can be effective.

Volume Holographic Database System

A computer-controlled angular-multiplexing photorefractive-based volume holographic memory has been used to store database records, search through the records, and recall the information stored in the memory [GOE96]. Figure 1 depicts the Volume Holographic Database System (VHDS) that was used in the experiments. To record information

we load the data into the spatial light modulator (SLM), create a unique reference angle through the reference beam generation arm, and then open shutters SH1 and SH2. After a predetermined time the shutters are closed, at which point the interference pattern of the two beams has been successfully recorded in the photorefractive crystal. This process is repeated until all the information has been stored. To recall pages we generate a unique reference angle,

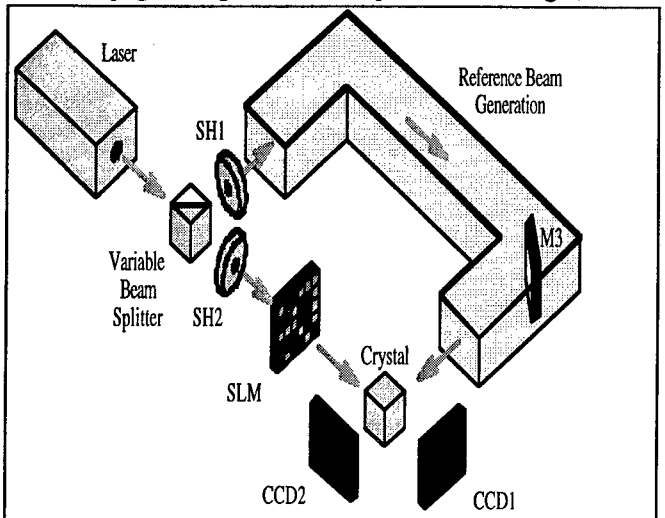


Figure 1. Volume Holographic Database System (VHDS)

open shutter SH1, and then capture the data on the camera CCD1. This process can be repeated as needed. The most important aspect of this system is its ability to search every record stored in the memory in a single step; the associative property. To perform searching, we must first have multiple pages of data stored within the memory. With the data in place, we load the SLM with a search argument, open shutter SH2, and capture an image of the reference beam plane on CCD2. Using this image we can determine the angular "address" of the desired information. The search argument that is presented to the VHDS can range in size from an entire page of data to just a small section of a page. This gives us the ability to search for a very specific record, or to search for multiple records that contain similar information.

In this work up to 800 pages were successfully recorded in one cm^3 of $Fe:LiNbO^3$ with each page comprising one record of a relation with data fields containing: last name, first name, affiliation, address, city, zip code, and telephone number. Records ranged in length from 98 to 210 characters. These characters were modulated to a binary format using a 2-out-of-15 encoding scheme and a multiblock row and column parity code. Tests were successfully performed on both modes of operation; addressed recall and associative recall. To test addressed recall the VHDS was presented with angles that corresponded to specific pages and then the output of the memory at CCD1 was examined to determine if the correct image was indeed recovered. The results showed that the 800 holograms were successfully recorded and that any page could be reconstructed.

In testing the associative recall they explored how both the search argument and the data stored in the memory affect reconstruction of reference beam planes [MIT98b]. How the number of characters in the search argument, the number of matches, the position, the orientation, and size of the search argument affect recall were also examined. It was determined that when the number of characters in the search argument decreased, the intensity of the correct hit dropped thereby setting a lower bound on the number of characters that are required in the search argument. However, this lower bound is well within the operational limits of the system. It was also shown that it is possible to find multiple pages containing similar data.

Electro-Optical Computer Architecture

Since we are interested in data mining applications, which are heavily based on content-based searches, a system similar to the VHDS forms the basic building block of the proposed Electro-Optical Computer Architecture (EOCA). We call this block a holographic associative processor (HAP) since it is an improved VHDS. The EOCA employs many HAP blocks arranged in groups. Each group will store related data (i.e., relations of the same database, images of the same collection, or video sequences). Certain HAP blocks are reserved for storing and searching index files for faster data access and more efficient data manipulation. Different data types can be stored in the pages of the same recording. For example, pages of binary alphanumeric data can be interleaved with pages of digitally encoded imagery or gray-scale images.

The need for data modulation and error coding to ensure industry acceptable corrected bit error rates ($<10^{-14}$) will reduce the user capacity of the system. A 1 Mbit page with a 40% overhead for modulation and error control will be able to accommodate roughly 75,000 ASCII characters. This number can be contrasted with typical page sizes in electronic systems of .5, 1 and 2 Kbytes. Thus, with 75 Kbytes/page a variety of combinations can be accommodated from all tuples of the same relation to interesting mixes of various types of data.

In our analysis of the potential of EOCA, we select parameter values from the ranges given below. Other parameters are defined as needed.

Number of Pages / Spatial Location:	1,000 - 10,000
Number of Spatial Locations:	1 - 100
Page Resolution:	1000x1000 (1 Mbit)
Access Time (page to page):	1 - 10 μs
Access Time (location to location):	100 μs - 1 ms
Frame Readout Time:	1 μs - 1 ms
Reference Beam Profile Readout Time:	1 - 10 μs
Frame Recording Time	1 ms - 1 sec

Structure of the EOCA

Shown in Figure 2 is an overall block diagram of an Electro-Optical Computer Architecture. This architecture serves as the basis of our evaluation of the potential performance

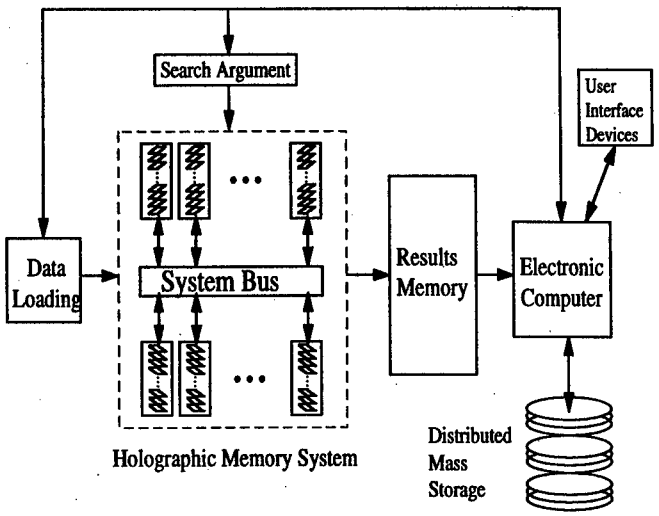


Figure 2. Electro-Optical Computer Architecture

and functionality improvement that such a system can bring to the knowledge discovery and data mining environment. The optical system consists of many HAP blocks. These blocks are connected together in order to form an ultra large multimedia data warehouse that can house terabytes of data. In this section we characterize the system in terms of memory sizes, bandwidths, speeds, scalability, degree of parallelism, etc.

Holographic Memory System (HMS) Response Time

For the holographic system considered here the total storage capacity per module is determined from the following equation:

$$S_{total} = N_{bits/char.} \times N_{char/page} \times N_{pages} \times N_{SL}$$

where $N_{bits/char.}$ is the number of bits that it takes to represent a character, $N_{char/page}$ is the number of characters that can be placed on a page, N_{page} is the number of pages that can be placed in a spatial location, and N_{SL} is the number of spatial locations. S_{total} represents the total capacity of the memory. However, the effective capacity is smaller since it will require more than 8 bits to store a character (byte) of data. Since we are most interested in very large data/knowledge bases we will assume a large system. Thus, if we assume 10,000 pages per spatial location, two spatial locations, 1000 HAP's operating in parallel and one megabit/page, we will have a 20-terabit system. Allowing bits (40%) for parity and error correction and converting to bytes we would have a 1.5-terabyte capacity system. As with any system, design tradeoffs are required. For instance, in the case of increased spatial locations, we would be able to have fewer storage elements but search times would be increased.

Knowing that most operations in a database environment involve the retrieval of a record or group of records per request, it is more useful to discuss the response time of the system than the data rate, which is a commonly used performance metric. We define the response time here as the time between the point a request for data is made and the point when the desired information becomes available. This is directly affected by the system components, the type of data access (addressed or associative), and the possibility of having to reread a page of information due to double errors.

The response times of the system components are defined as $T_{shutter}$, T_{angle}, T_{SLM}, T_{CCD}, and T_{decode} for the shutter, generation of the angularly-encoded reference beam, SLM, CCD detector array, and decoder, respectively. T_{CCD} is the total response time of the CCD array which includes both the integration time (the time over which optical power is integrated on the array) and the time to read all pixels from the detector.

Address-based retrieval is performed by generating the reference beam (i.e. deflecting to the desired angle), illuminating the crystal, and then detecting and decoding the output. Thus, the addressed retrieval response time, T_{Addr}, is

$$T_{Addr} = T_{angle} + T_{shutter} + T_{CCD} + T_{decode}$$

A fast deflector (such as an acoustooptic device) can be set in only a few microseconds and decoding can be done in a parallel fashion within microseconds. The shutter, SLM, and CCD, however, have response times on the order of milliseconds. Thus, T_{angle} and T_{decode} can be eliminated from the equation and the equation for T_{Addr} is approximated by

$T_{Addr} = T_{shutter} + T_{CCD}$

For associative retrieval, the search argument must first be generated on the SLM, the output reference beams must be detected, and finally each matching page retrieved by address. Thus, T_{Assoc}, the associative retrieval response time, is the time to load the SLM with the search argument plus the time to detect the location(s) of the matching page(s) plus the time to retrieve and process those pages. Again ignoring, T_{angle} and T_{decode} we have

$$T_{Assoc} = 2\,T_{shutter} + T_{SLM} + T_{CCD} + k\,N_{Rec}\,(T_{CCD} + T_{post})$$

where k is the selectivity factor equal to the percentage of records which match the selection criterion ($k \leq 1$), N_{Rec} is the total number of records in the database, and T_{post} is the time required to do any necessary post-retrieval processing to determine an exact match with the search argument.

This analysis is valid only for purely angularly multiplexed systems. If spatial multiplexing is also employed to increase capacity, the address based retrieval response time does not change, but the associative retrieval response time is directly affected. For spatio-angular multiplexed systems, the search process must be carried out for each of N_{SL} locations. Thus, T_{Assoc} becomes:

$$T_{Assoc} = 2\,T_{shutter} + T_{SLM} + N_{SL}\,T_{CCD} + k\,N_{Rec}\,(T_{CCD} + T_{post})$$

where we have assumed that the response time of the deflector used to direct the search argument to the next location is on the order of T_{angle} and have neglected it in the third term.

The majority of retrievals in a database environment are content-based, so we are primarily interested in T_{Assoc}. It is important to note that the search time in the HAP does not vary with the number of search criteria, unlike electronic database machines. That is, a search for the name 'Smith' and a search for both the name 'Smith' and the zip-code '68405' are performed equally fast since all records and attributes are searched simultaneously.

In order to generate some insight into the capabilities of the HAP we assume some values for the terms in T_{Assoc}. In the following calculations we will assume that $T_{shutter} = 3$ msec, $T_{SLM} = 3$ msec, $T_{CCD} = 1$ msec and $T_{post} = 1$ msec. In the case of performing any complex query for a count of the number of hits as described above $T_{Assoc} = 11$ msec. It is important to note that what is retrieved at this point are hologram locations that represent the pages that contain the search argument(s). The number of hits will yield the number of qualifying pages. In the association rules data mining technique, the algorithm can be executed by just counting the number of hits. We expect that this approach will yield two to five orders of magnitude reduction in time.

If we then desire the pages, we can estimate the time to retrieve them from the HAP by selecting a value for the selectivity factor k and knowing the number of records in the system. If we assume that the number of records is one per page then there are 20,000 records per HAP. With a selectivity factor of $k = 0.01$ then $T_{Assoc} = 411$ msec. With 75,000 characters per page this is an effective transfer rate of 36 megabytes per second. Standard magnetic disks have transfer rates on the order of five megabytes per second. It is important to note that with improved optical components the read out rate of the HAP can be increased considerably.

Electro-Optical Computer Architecture Response Time

The main strength of the EOCA is associative access. That is, we can search all pages in memory for responders to an arbitrarily complex query and determine page positions in one scan of the memory. Thus, we can search a terabyte database in a matter of milliseconds. From the mirror angles we can obtain the number of responses to the query and with multiple scans of the EOCA we can obtain all of the data we need to execute the association rules algorithm. With the EOCA the potential exists to render the time to execute the association rules algorithm negligible. From the peaks in the reference beam profile we can determine the pages in memory that have produced hits and they can be read out if needed or they can be accessed from secondary storage on the sequential front end computer and further processing performed.

Referring to Figure 2 note that all HAP units operate in parallel. Thus, for an arbitrarily complex query the electronic computer would broadcast the search argument to all HAP's via the system bus. They would execute in parallel and collect the responding hologram position data at each HAP. The count could be determined at each HAP with a local processor or the hologram positions could be transferred to the electronic computer for determination of the count. In executing association rules, a local processor could collect the results of many passes and do some preprocessing prior to sending the results to the electronic computer.

The timing equation for executing a single search, T_S, on the EOCA for a complex query is composed of a) a query broadcast time, T_B, b) a search, T_{Assoc}, without readout (the first three terms), and c) a collection of the hologram positions from the

HAP's and their transfer to the electronic computer for further processing T_{trans}. Thus,

$$T_S = T_B + T_{Assoc} + T_{trans}$$

T_B takes a few microseconds and T_{Assoc}, based on previous calculations, is 11 msec. T_{trans} will depend upon how many HAP's have registered hits. However, suppose they all do. To transfer the hologram positions from a single HAP would require a few microseconds. Since there are 1,000 HAP's in the EOCA we would expect the transfer time to take a few msec. Thus, the entire process would only take order of milliseconds to complete. For association rules, depending upon the number of queries to the EOCA that would be required, the algorithm could be executed in a matter of seconds.

Database of Transactions **Determine Relationships**

Trans	A	B	C	D
1	X	X	X	
2		X		X
3	X	X		X
4	X	X		X
5	X			X
6		X	X	X
7	X	X		X
8		X	X	X
9	X	X		
10	X	X	X	

Trans	A	B	C	D	Set Support at 50%:
10	7	9	4	7	A, B and D Qualify

AB	AD	BD	
6	4	6	AB and BD Qualify

ABD	
3	ABD Does Not Qual.

The significant relationships are: A and B; B and D

Figure 3. Association Rule Example

Shown in Figure 3 is an example database of transactions that is used to illustrate the capabilities of the EOCA in solving the association rules problem. There are ten transactions that have A, B, C and D as possible values. In relational database parlance we have a single relation with the transaction number as primary key and the presence or absence of the values A, B, C and D in the four domains. One can view this in a commercial application as the fact that the customer purchased A, B, and C in transaction 1, another customer purchased B and D in transaction 2, and so on. In mining for association rules we would like to know the strength of the relationship between and among the items purchased in all of the transactions.

We first set the level of support or strength of relationship that we are interested in. Here we choose 50%. That is, if the percentage of transactions that include an item is 50% or greater, then we look further for associations between and among all of those items. In this case we see that A, B and D meet our criteria. We now look for associations between products and find that AD and BD qualify, but AD does not. Finally, the relationship ABD does not qualify.

In executing this algorithm using sequential computing, the database would have to be accessed many times or multiple indexes would have to be established depending upon the approach taken to solving the problem. Using the EOCA, the timing equation given below would determine the time to produce all of the necessary count data and then it would be a simple matter to determine all possible associations.

$$T_{AR} = T_B + k \sum_{i=1}^{n} \binom{n}{i} (T_{assoc} + T_{trans}) + T_{Calc}$$

In this equation k is the number of tuples per page since we will have to perform multiple searches if we have more than one tuple per page; n is the number of domains in the transactions (four in the above example), while the sum of combinations gives all possible combinations of the domain values (A, B, C, D, AB, AC, ..., ABCD). T_{assoc} is as before and T_{trans} is the transfer time from each HAP to the sequential computer under the assumption that results are transferred after each search of the EOCA. If the results are all collected first and then transferred this term would be larger but outside the parenthesis yielding a smaller value overall. However, the calculation of the associations T_{Calc} would be impacted since this operation could not commence until all the data in the EOCA were collected and transferred. In the above equation it is assumed that the transfer of the partial results will be provided to the sequential computer for processing as they become available and the transfer time and calculation time can be overlapped. Thus, the time required for T_{Calc} is just the time to process the results from a single interrogation of the EOCA.

If we assume that there are four domains, 1.5 terabytes in the EOCA, 10 tuples per page and T_{Calc} is 10 msec, then T_{trans} is 10 msec. T_{AR} for this example is about 3.6 seconds. Although not a valid comparison, just to transfer 1.5 terabytes of data from magnetic disks would take days.

Thus, it is clear that the use of the associative property in the EOCA has great potential for speeding up association rule processing. However, we must still bear in mind that holographic memories are not yet widely available, they take a long time to load, and, of course, have other problems that must be solved before they can become a main stream computer system reality. But, nonetheless, great potential exists which clearly warrants continued investigation.

Similar difficulties arise with clustering, so additional research needs to be performed to more completely measure the effectiveness of the HAP in executing these data mining algorithms.

References

[BAB97] Babbitt, W. R., "Persistent Spectral Hole-Burning Memories and Processor," in Proceedings of the 1996 Workshop on Data Encoding for Page-oriented Optical Memories, P.A. Mitkas, Editor, Colorado State University College of Engineering, 1997.

[BER89] Berra, P.B., Ghafoor, A., Mitkas, P.A., Marcinkowski, S., Guizani, M., "Optics and Supercomputing," Proceeding of the IEEE, (invited paper) Vol. 77, No 12, December 1989.

[BER90] Berra, P.B., Brenner, K.H., Cathey, W.T., Caulfield, H.J., Lee, S.H.and Szu, H. "Optical Database/Knowledge Base Machines," Applied Optics, Vol. 29, January 1990.

[BER98] Berra, P. B.,Mitkas, P. A. and Liuzzi, R. A., "Dynamic Data Mining Using an Electro-Optical Data Warehouse," IEEE Information Technology Conference, Syracuse, NY September 1-3, 1998.

[BUR98] G. W. Burr, W.-C. Chou, M. A. Neifeld, H. Coufal, J. A.Hoffnagle, and C. M. Jefferson, "Experimental Evaluation of User Capacity in Holographic Data StorageSystems," Applied Optics, Vol. 37, 1998, pp. 5431-5443.

[CHE96] Chen, M-S., Han, J. and Yu, P. S. "Data Mining: An Overview from a Database Perspective," IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No.6, December, 1996, pp. 866-883.

[GOE96] Goertzen, B. J. and Mitkas, P. A, "Volume Holographic Storage for Large Relational Databases," Optical Engineering, Vol. 35, No. 7, July 1996, pp. 1847-1853.

[GUI96] Guilfoyle,P., Hessenbruch ,J., Stone,R., Liuzzi,R.A., and Berra, P.B., "Implementation of Relational Algorithms on a Digital Optoelectronic Processor", Dual-Use Technologies & Applications Conference, Syracuse, New York, June 3-6, 1996.

[HON95] Hong, J. H., McMichael, I., Chang, T. Y., Christian, W. and Paek, E. G. "Volume Holographic Memory Systems: Techniques and Architectures," Optical Engineering, Vol. 34, No. 8, Aug. 1995, pp. 2193-2203.

[MIT98a]P. A. Mitkas, G. Betzos, and L. J.Irakliotis, "Optical Processing Paradigms for Electronic Computers," IEEE Computer, Vol. 31, No 2, February 1998, pp. 45-51.

[MIT98b] P. A. Mitkas, G. Betzos, S. Mailis, and N. Vainos,"Characterization of Associative Recall in a Volume Holographic Database System for Multimedia Applications,"SPIE Proceedings, Vol. 3388, pp. 198-208, 1998.

[PSA90] Psaltis, D., Neifeld, M. A., Yamamura, A., and Kobayashi, S. "Optical Memory Disks in Optical Information Processing," Applied Optics, **29,** 14 (10 May 1990), pp. 2038-2057.

[PSA95] Psaltis, D. and Mok, F. "Holographic Memories," Scientific American, Vol. 273, No. 5, Nov. 1995, pp. 70-76.

[PSA98] Psaltis, D. and Burr, G. "Holographic Data Storage," IEEE Computer, Vol. 31, No. 2, 1998, pp. 52–60.

[SCH98a] Schaffer, M. and Mitkas, P. A., "Requirements and Constraints for the Design of Smart Photodetector Arrays for Page-oriented Optical Memories," IEEE Journal of Selected Topics in Quantum Electronics, 1998.

[SCH98b] M. E. Schaffer and P. A. Mitkas, "Smart Photodetector Array for Page-oriented Optical Memory in 0.35-um CMOS," IEEE Photonics Technology Letters, Vol.10, No. 6, June 1998, pp. 866-868.

[THU97] Thuraisingham, B., Nwosu, K. C. and Berra, P. B. (editors), Multimedia Database Management Systems: Research Issues and Future Directions, Kluwer Academic Publishers, 1997.

Acknowledgement

This research was partially supported by AFRL/IFTB SBIR Contract No. F30602-98-C-0076 entitled "Dynamic Data Mining Using an Electro-Optical Data Warehouse."

Knowledge Discovery and Knowledge Bases: Problems and Opportunities

Vinay K. Chaudhri and Marie E. DesJardins
Artificial Intelligence Center
SRI International
333 Ravenswood Ave, EJ225
Menlo Park, CA 94025

Knowledge bases (KBs) can enable Knowledge Discovery from Databases (KDD) by providing a natural, object-oriented representation of an application domain, a powerful query language that can manipulate schema as well as the ground facts, and an easy to use graphical interface that can support interactive exploration [1, 2, 3] KDD, in turn, can enable the construction of a KB by semiautomated derivation of rules of domain knowledge or by starting from a KB and refining it based on the data in a database. This two-way interaction presents a multitude of opportunities, and we attempt to address some of them.

Many KDD engines use automatic statistical or machine-learning mechanisms to search for implicit patterns in data. The overall KDD task faced by an analyst, however, involves many other activities in addition to what is offered by the core KDD engine. The input necessary for a KDD engine is not usually available in the required format, and in most cases, has to be prepared by processing the data in an existing database. For example, while analyzing the commodities exported by a country, the export data may be available for each product (such as beef, chicken, etc.), but the input to the KDD engine needs to be represented in terms of abstract categories of products (such as animal products). In such a situation, an ontology categorizing commodities can significantly aid an analyst in preparing the data for input to the KDD engine. KDD tasks are usually iterative and involve experimenting with categories at different levels of abstraction. Frame Representation Systems, such as Ocelot, and graphical browsing and editing tools, such as the GKB-Editor [KCP99], are natural tools for hierarchical representation, display, and selection of knowledge. Their utility is significantly enhanced with an interface to a commercial database management system supported by a system such as PERK (Persistent Knowledge) [2].

Large knowledge bases (KBs), such as the Cyc KB, the Sensus ontology, or the Ontolingua ontology library, are expensive to build [4, 5, 6]. The output of a KDD task can contribute significantly to KB development. Many KDD tasks extract association rules from data, which can be integrated directly into a KB. If these newly learned rules are determined to be inconsistent with existing rules in the KB, this serves as an indicator of potential errors in the existing rules, or in the data that was used to generate the new rules. In other cases, a KB may contain causal rules that do not have associated probabilities indicating the strength of causation. Probabilistic KDD tools can use empirical data to assign probabilities to these rules.

In summary, leveraging KB systems with KDD tools will permit more effective knowledge understanding by providing KB support to prepare data for the KDD process, and using the output of the KDD tools to refine the contents of the KB.

References

[1] Vinay K. Chaudhri and Peter D. Karp. Querying Schema Information. In *Proceedings of the 4th International Workshop Knowledge Representation Meets Databases (KRDB'97)*, pages 4–1 to 4–6, Athens, Greece, 1997.

[2] Peter D. Karp, Vinay K. Chaudhri, and Suzanne M. Paley. A Collaborative Environment for Authoring Large Knowledge Bases. *Journal of Intelligent Information Systems*, 1999. To appear.

[3] R. J. Brachman, P. G. Selfridge, L. G. Terveen, B. Altman, A. Borgida, F. Halper, T. Kirk, A. Lazar, D. L. McGuiness, and L. A. Resnick. Knowledge Resentation Support for Data Archaeology. In *Proceedings of the First International Conference on Information and Knowledge Management*, Baltimore, MD, 1992.

[4] Douglas B. Lenat and R.V. Guha. *Building Large Knowledge-based Systems: Representation and Inference in the Cyc Project.* Reading, MA, Addison-Wesley Publishing Co., 1989.

[5] K. Knight and S. Luk. Building a Large-Scale Knowledge Base for Machine Translation. In *Proceedings of the National Conference on Artificial Intelligence*, Seattle, WA, August 1994.

[6] Adam Farquhar, Richard Fikes, and James P. Rice. A Collaborative Tool for Ontology Construction. *International Journal of Human Computer Studies*, 46:707–727, 1997.

Exploring Reusability Issues in Telemetry Knowledge Bases

Mala Mehrotra
Pragati Synergetic Research Inc.
145 Stonelake Ct
Yorktown VA. 23693
mm@norfolk.infi.net

Abstract In spacecraft telemetry, expert systems technology is being used to manage the complexity generated by the greater number of complex measurands. The telemetry subsystems usually have multiple configurable roles; hence, there are similar rule bases in existence for different subsystems. There is a need to extract reusable components from such systems so that they can be adapted and integrated for newer missions. A semi-automated tool, such as Pragati's MVP-CA (Multi-ViewPoint Clustering Analysis) tool, can provide a valuable aid for comprehension, maintenance, integration and evolution of these expert systems by structuring a large knowledge base in various meaningful ways. The similarity in existing telemetry rule bases is exploited by applying the MVP-CA tool to "mine" the knowledge existent in them. This knowledge can serve as a handle to fuse information from different rule sets and formulate new rule sets for further mission planning activities. We will discuss issues about indexing, retrieval and adaptation of the rule sets by describing a support architecture needed in the MVP-CA tool for investigating the identification of potentially reusable clusters and linking it with case-based reasoning technology.

Key Words: expert systems, clustering, reusability, case-based reasoning

1. Introduction

The increased number and complexity of spacecraft mission measurands and the evolution of ground systems architectures that support multiple configurable roles have emphasized the need to alleviate the mission operator workload. Rule-based expert systems are a common technology used to manage this complexity; yet a rule set created for a particular mission is often developed in a stand-alone, ad hoc manner. The consequence of this practice is that rule-based systems are redeveloped each time the system changes [1]. Moreover, due to the critical nature of these applications, much more stringent standards have to be imposed now on their ability to provide reliable decisions in a timely and accurate manner. Pragati's Multi-ViewPoint-Clustering Analysis (MVP-CA) tool provides a framework for clustering large, homogeneous knowledge-based systems from multiple perspectives [8]. It is a semi-automated tool allowing the user to focus attention on different aspects of the problem, thus providing a valuable aid for comprehension, maintenance, verification and validation (V&V), integration and evolution of knowledge-based systems.

The MVP-CA tool has recently been adapted for clustering telemetry knowledge bases. We present here some preliminary results of applying the MVP-CA tool on some telemetry expert systems. In particular, results exposing verification and validation (V&V) problems in the rule bases have been discussed in [11,12]. We will briefly discuss here our next step of extracting reusable components in a systematic manner by proposing an integration of the MVP-CA tool with case-based reasoning (CBR) technology. Issues relating to indexing, retrieval and adaptation of the rule sets can be addressed effectively when the two technologies are integrated.

2. Motivation

Expert systems are increasingly being used as intelligent information specialists in cyberspace, both for civilian and military applications. In spacecraft telemetry, expert systems technology is used to manage the complexity generated by the greater number of complex measurands [7]. Spacecraft satellite telemetry (sub) systems have a unique characteristic in that they usually have multiple configurable roles; hence, there are similar rule bases in existence for different subsystems. As new missions get planned the number of such rule bases with similar structures keeps growing. Also, as new knowledge evolves due to new technology in the market, these systems have to be adapted to incorporate/reflect the changes in technology. Each mission has its own rule set to be applied and each one of them has the potential to grow into a monolithically large unmanageable system. The phenomenon of "add a rule each time" to take care of different situations in any expert system, leads very quickly to an uncontrolled proliferation of rules in the expert

system. Due to the data-driven nature of expert systems, as the number of rules of an expert system increase, the number of possible interactions between the rules increases exponentially. The complexity of each pattern in a rule compounds the problem of management of rules even further. Documentation has the danger of becoming obsolete very quickly, as software developers do not always have the necessary discipline to keep updating their documentation. Furthermore, defining any requirements or specifications up front in such a rapid prototyping and iterative development environment, even though they are desirable, becomes an impractical and moot question. Even if they were specified, as any software, conventional or knowledge-based becomes more complex, common errors are bound to occur through misunderstandings of specifications and requirements [2].

It is therefore desirable to have an analysis tool that exposes a developer to the current software architecture and semantics of the knowledge base in such a dynamically changing development environment, so that the knowledge base can be comprehended at various levels of detail. To achieve this goal, the knowledge in the system has to be suitably abstracted, structured, and otherwise clustered in a manner that facilitates software engineering activities [5,6]. Hence, **by exposing the knowledge contained in the knowledge-based** system through the Multi-ViewPoint Clustering Analysis tool, we formulate a basis for addressing reusability, maintainability, and reliability issues for such systems.

3. Multi-ViewPoint Cluster Analysis (MVP-CA) Technology

Existing approaches to structuring systems are limited in a major way. They only provide a single viewpoint of a system. We believe that **no one single structuring viewpoint is sufficient to comprehend a complex system**. In this paper we show the feasibility of applying Pragati's Multi-ViewPoint-Clustering Analysis (MVP-CA) methodology on satellite telemetry rule-based systems for reusability. MVP-CA framework has the potential to be extended to incorporate case-based retrieval and adaptation technology for reusability of clusters generated through the MVP-CA tool.

Our approach hinges on generating clusters of rules in a large rule base, which are suggestive of mini-models related to the various sub domains being modeled by the expert system. These clusters can then form a basis for understanding the system both hierarchically (from detail to abstract) and orthogonally (from different perspectives). An assessment can be made of the depth of knowledge/reasoning being modeled by the system which can pave the way for adapting the clusters for new specifications in new systems.

3.1 Overview of the MVP-CA Tool

Pragati's Multi-ViewPoint-Clustering Analysis (MVP-CA) tool provides such a framework for clustering large, homogeneous knowledge-based systems from multiple perspectives. It is a semi-automated tool allowing the user to focus attention on different aspects of the problem, thus providing a valuable aid for comprehension, maintenance, integration and evolution of knowledge-based systems. The generation of clusters to capture significant concepts in the domain seems more feasible in knowledge-based systems than in procedural software as the control aspects are abstracted away in the inference engine. It is our contention that the MVP-CA tool can form a valuable aid in exposing the conceptual software structures in such systems, so that various software engineering efforts can be carried out meaningfully, instead of in a brute-force or ad-hoc manner [2,10]. In addition, insight can be obtained for better reengineering of the software, to achieve run-time efficiency as well as reduce long-term maintenance costs. It is our intention to provide a comprehension aid base first, through our MVP-CA tool, for supporting all these software engineering activities. The MVP-CA tool consists of a Cluster Generation and a Cluster Analysis Phase. Together they help analyze the clusters so that these clusters can form the basis for any software engineering activities.

The multi-viewpoint approach utilizes clustering analysis techniques to group rules that share significant common properties and then it helps identify the concepts that underlie these groups. In the Cluster Generation Phase the focus is on generating meaningful clusters through clustering analysis techniques augmented with semantics-based measures. In this phase, the existing rule base along with a concept focus list feeds into a front end interpreter. The interpreter parses the rule base and transforms it into an internal form required by the clustering tool. The clustering algorithm starts with each rule as a cluster. At each step of the algorithm, two clusters which are the most "similar" are merged together to form a new cluster. This pattern of mergings forms a hierarchy of clusters from the single-member rule clusters to a cluster containing all the rules. "Similarity" of rules is defined by a set of heuristic distance metrics for defining the distance between rules.

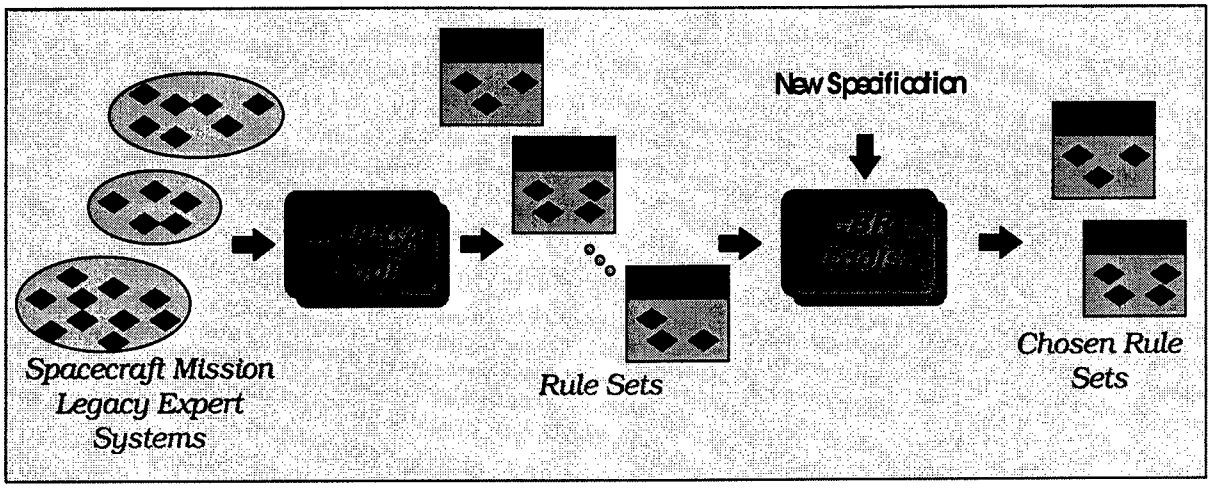


Figure 1: Using MVP-CA technology and CBR tools for management of telemetry rule sets

One of the most significant ways a user can effect the clustering process is through his choice of a distance metric. Distance Metric measures the relatedness of two rules in a rule base by capturing different types of information for different classes of expert systems [3,4]. There are five different distance metrics that we have implemented so far. Classification systems yield easily to a data-flow grouping and hence information is captured from the consequent of one rule to antecedent of other rules. This defines our ***data-flow*** metric. In a monitoring system since the bulk of domain information required for grouping is present in the antecedents of rules, the ***antecedent*** distance metric captures information only from the antecedents of rules. Alternatively, grouping the rule base on information from the consequents alone, gives rise to the ***consequent*** metric. The ***total*** metric is general enough and captures information from both sides of rules to take care of systems where a combination of the above programming methodologies exists.

4. Reusability of Rule Sets

In the MVP-CA-based environment, it is envisioned that legacy expert systems can be clustered into rule sets of semantically related rules as shown in Figure 1. Once we have a mechanism for decomposing the expert systems in various meaningful ways, relevant rule sets from different expert systems can be retrieved and assimilated through case-based retrieval (CBR) and analogical reasoning techniques [14]. In fact, the sets of rules could be "wrapped" in such a manner that commercial CBR tools could be used to retrieve the relevant rule sets as and when required. Once the appropriate rule set has been retrieved through the Cluster Interface Definition (CID), they can be adapted for the new mission's functionality as needed. For the new evolving prototypes, providing insight into the continually changing models through the MVP-CA tool can prove to be a valuable aid in their transition to an operational stage. Such an environment could then support the orderly and reliable transition of evolving, complex, knowledge-based system software in the satellite telemetry domain, so that such systems can be reused for new scenarios.

This environment will focus on the issues of long-term maintenance, reusability and evolution of mission-specific rule sets in spacecraft telemetry systems. Preliminary investigation is currently under way to study how case-based retrieval and storage techniques could be used effectively for storing and retrieving such CIDs defining the rule clusters. Our research efforts address the possibility of providing a software environment which enables semi-automatic detection, storage, retrieval and adaptation of these rule sets so that reusability of existing rule sets can be addressed across missions in a systematic and disciplined manner [13]. It is envisioned that some form of case-based storage and retrieval techniques will be incorporated into the MVP-CA methodology for reuse of rule sets, so that a full-scale prototype environment can be built. Such an environment will alleviate the developer from the tedious burden of manually inspecting large and complex legacy rule bases before building a new rule base for the next similar mission.

Rule#	Rule Description
* 138	INCL BT -5 5 => INCLIN = EQTRL
* 139	INCL BT 5 30 V INCL BT -5 -30 => INCLIN = L_INCLIN
* 140	INCL BT 30 60 V INCL BT -30 -60 => INCLIN = I_INCLIN
* 141	INCL BT 60 80 V INCL BT -60 -80 => INCLIN = H_INCLIN
* 142	INCL BT 80 90 V INCL BT -80 -90 => INCLIN = PLR
* 173	PERIGEE BT 96 145 ^ APOGEE BT 320 480 ^ INCL BT 80 100 => ORBIT = LOW1
* 175	PERIGEE BT 200 300 ^ APOGEE BT 200 300 ^ INCL BT 45 70 => ORBIT = STS57L
* 177	PERIGEE BT 280 420 ^ APOGEE BT 280 420 ^ INCL BT 45 70 => ORBIT = STS57H
* 179	PERIGEE BT 480 720 ^ APOGEE BT 480 720 ^ INCL BT 45 70 => ORBIT = ERBS
* 174	PERIGEE BT 135 205 ^ APOGEE BT 185 276 ^ INCL BT 85 105 => ORBIT = LOW2
* 180	PERIGEE BT 630 770 ^ APOGEE BT 630 770 ^ INCL BT 90 105 => ORBIT = L^SAT
* 181	PERIGEE BT 735 900 ^ APOGEE BT 750 920 ^ INCL BT 90 110 => ORBIT = DMSP
* 183	PERIGEE BT 800 980 ^ APOGEE BT 820 1000 ^ INCL BT 90 110 => ORBIT = IRAS
* 176	PERIGEE BT 240 360 ^ APOGEE BT 240 360 ^ INCL BT 20 35 => ORBIT = STS28L
* 178	PERIGEE BT 400 600 ^ APOGEE BT 400 600 ^ INCL BT 20 35 => ORBIT = STS28H
* 185	PERIGEE BT 400 600 ^ APOGEE BT 31600 47500 ^ INCL BT 52 78 => ORBIT = MOLNIYA
* 182	PERIGEE BT 700 860 ^ APOGEE BT 700 860 ^ INCL BT 95 120 => ORBIT = GEOSAT
* 184	PERIGEE BT 1035 1265 ^ APOGEE BT 1080 1320 ^ INCL BT 81 99 => ORBIT = NOVA
* 186	PERIGEE BT 15900 23800 ^ APOGEE BT 16300 24600 ^ INCL BT 50 75 => ORBIT = GPS
* 187	PERIGEE BT 28600 43000 ^ APOGEE BT 28600 43000 ^ INCL BT 0 10 => ORBIT = GEOSYNC

Figure 2: Candidate reusable SEAS rule group

Even though our ideas are being applied to telemetry applications primarily, the methodology for reusability being advocated here can be transitioned to other knowledge-based applications areas such as, medical, forensics, civil engineering and others. Also the clustering methodology in the MVP-CA technology is not dependent on any particular knowledge representation scheme or the language of the knowledge-based system; hence, the MVP-CA methodology can be integrated into any environment that encapsulates domain knowledge in a regular form.

4.1 Reusable Rule Sets in Telemetry Systems

In our preliminary study we have manually identified several rule sets from telemetry systems that could be viable candidates for reusability. Informal discussions with some domain experts in telemetry systems have corroborated our results. Currently, from the MVP-CA tool we aid the user in detecting potentially reusable clusters and then provide him with an infrastructure to first describe the cluster in free-form text and then ask him to describe it with a few keywords. We also ask of the expert how he/she envisions retrieving it. In other words, what is the most likely manner in which another domain expert may want to recall this cluster in future.

A representative stable group for the concept of inclination (INCL) from Aerospace's SEAS rule base is presented in Figure 2. *Spacecraft Environmental Anomalies (SEA-ES)* is an expert system developed by The Aerospace Corporation, Space and Environment Technology Center for use in the diagnosis of satellite anomalies caused by the space environment. The satellite anomalies to be detected by the rule base ranges from surface charging, bulk charging, single-event effects, total radiation dose, and space-plasma effects. Various parameters play a role in the determination of these anomalies such as, orbit of the satellite, the local plasma and radiation environment, satellite-exposure time, hardness of the circuits and their components etc.

The cluster in Figure 2 shows the relationship between different orbit types, inclination types, perigee and apogee. Concepts such as inclination are a supporting domain concept in this rule base, that the MVP-CA tool allows us to identify through the clustering of rules. The key features for indexing for such a cluster will be INCL, INCLIN, PERIGEE, APOGEE and ORBIT. The interrelationships across these concepts could be documented in an annotations window, and their various possible values or value ranges could get represented through templating, discussed in the next section. These rule sets can thus become viable candidates for potential reuse. In future if another type of orbit needs to be specified, the developer needs to retrieve this cluster, and be careful of not infringing one of the already specified ranges for the various orbit types.

XTE knowledge base from NASA provided a very rich environment for finding reusable rule clusters. X-Ray Timing Explorer (XTE) is an expert system written in GenSAA (Generic Spacecraft Analyst

```
76  rcvr-1 lock2search
77  rcvr-1-search2lock
78  rcvr-2-lock2search
79  rcvr-2-search2lock
```

```
(defrule rcvr-1-lock2search ""
   (Mission XA1CARLK#XTE_DECOM
      ?r1&:(neq ?r1 LOCK))
   ?x1 <- (Inferred SC-Rcvr-1-Lock
      ?r3&:(neq ?r3 Search))
=>  ...
   (SendMessage "MessageWindow" Status
    "Reciever 1 went from Locked to Search"))

(defrule rcvr-1-search2lock ""
   (Mission XA1CARLK#XTE_DECOM  LOCK)
   (Mission XA1RCVLK#XTE_DECOM  LOCK)
   ?x1 <- (Inferred SC-Rcvr-1-Lock  Search)
   (Inferred valid-telemetry  valid)
=>  ...
   (SendMessage "MessageWindow" Status
```

Figure 3: Reusable Cluster from XTE rule base for Receiver Lock and Search

Assistant), which is a superset of Clips. GenSAA was built by NASA to serve as a development and application environment for building expert systems at various NASA control centers. XTE is a health and safety monitoring rule base, checking the various onboard subsystems on the satellite, such as attitude and control system, power subsystem, thermal subsystem, solar array subsystem, spacecraft data subsystem, transponder subsystem, and many others.

A couple of representative clusters from this knowledge base are presented below. In Figure 3, we present a group of rules, which set the *Receiver* in one of two modes, *lock or search.* Contents of this group of rules for Receiver 1 is presented in an abbreviated form. Receiver 2 had the same set of rules for the different mode switches as well. By highlighting the similarity across the rules in this set, the MVP-CA tool brings to our attention, the high level functionality of the rule set. If this functionality can be captured in a template form we can generate more such sets of rules for different receivers.

In Figure 4 we present another representative cluster from the XTE knowledge base which watches the telemetry and statistics monitor (TSM). A close inspection of the rules themselves in Figure 4 reveals the potential reuse capability of such a rule set. The CID definition for this rule set would have to incorporate a general name, for example, "TSM watch" rules for indexing purposes. (Notice a possible anomaly in rule *tsm-0-22-watch* that is really watching the range, 0 through 16 instead of 0 through 22, as suggested by the rule name.) We would like to match up the CID definitions obtained from domain experts, with features to be utilized for our case-based indexing scheme for the clusters.

MVP-CA tool's contribution, in the context of reusability of software systems, is to ease the process of populating repositories of reusable components by semi-automatically flagging rule sets in existing knowledge bases.

4.2 Adaptability of Telemetry Rule Sets

The overall environment for reuse as envisioned in the MVP-CA tool is conceptualized as a problem space, which is indexable by CIDs and a solution space, which stores the adaptable and reusable clusters. As a new specification comes in, the CBR technology enables us to pull out the relevant clusters through retrieval algorithms. The adaptable cluster is then pulled out and a new rule cluster for the new mission is formulated.

In our case of adapting the rule clusters to the problem at hand, we would have to identify the parameters, which will take on different values for different missions. We illustrate this aspect by handworking through a rule set, shown in Figure 5, which we obtained during our interactions with the NASA flight engineers. Since they were contemplating on putting it in their reuse repository, we chose to work with them on templating such a cluster. This set of rules is part of a background monitoring system and its functionality is to basically infer the telemetry data quality from the main-frame data quality (MF-QUALITY). These rules are a part of TPOCC (Transportable Payload Operations Control Center) where the process XTE_DECOM is defined and active. Each of the rules in the set basically checks if the frame synchronization is in place, and what types of data-quality are being obtained from the main frame. It then asserts the deduced fact and sends the appropriate message. Thus, there are certain portions of the code, which are, like the constants of an equation; the rest are the variable parameters.

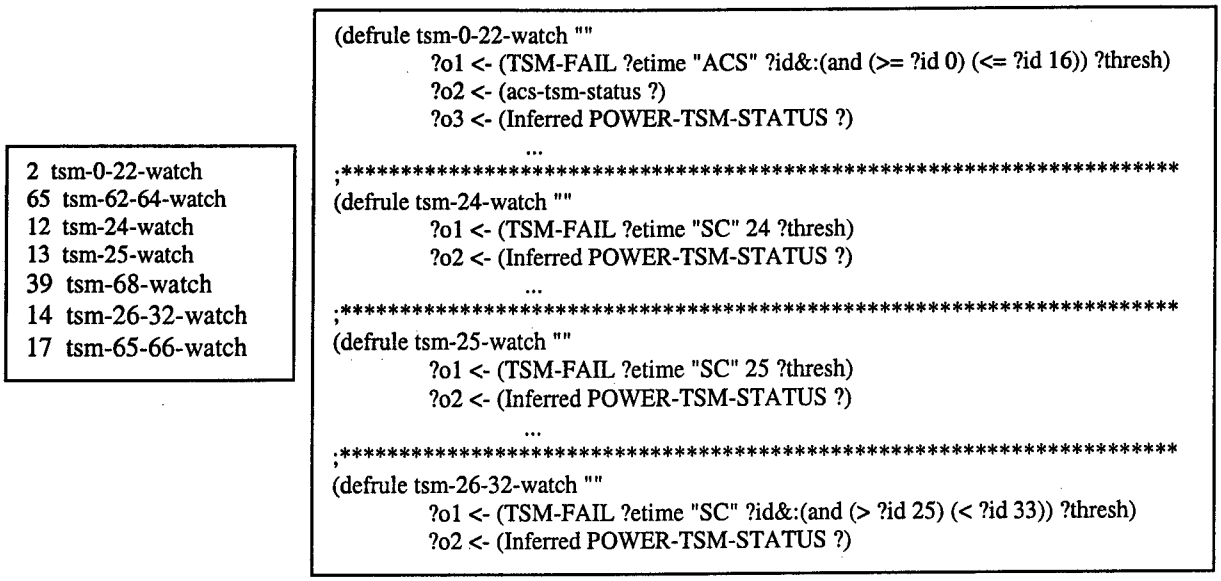


Figure 4: Telemetry and Status Monitoring Reusable Cluster from the XTE rule base

One of the most practical ways in which such information about a cluster can be captured is through the generation of a template for the cluster [13]. Since the underpinnings of a reusable cluster will necessarily be the degree of similarity of rules within that cluster, trying to encapsulate this knowledge in a template form is a first step towards making the cluster reusable.

The challenge in this situation was to locate the static or constant portions of the code and set it off from the parameterizable or variable portion of the code. In other words, when two rules are deemed similar to a certain degree, one would like to know to what extent and what type of similarity it is. It is postulated that given such a group, it is feasible to create a template, which would look like the one shown in Figure 6. A new <name-of-rule> is generated for each of the different rule cases for checking data quality. For this set of rules the Mission name, MF_QUALITY and process name, XTE_DECOM, is fixed; hence we did not parameterize it. However, in building the indexing scheme for such a representative cluster we may want this to be filled in as a slot in the attribute fields as shown in Figure 7. Thus, Figure 7 represents a higher level of abstraction for the cluster, than Figure 6. The former is a means of storing and retrieving the cluster templates. Once retrieved, the necessary open slots can be instantiated with the new mission needs and names. Thus we can build a case library of such rule sets, indexable through the CIDs, such as given in Figure 7, and we can then, retrieve for the user, relevant parameterizable templates which could be adapted for the situation at hand. Such templates would abstract the structure of the rule set and can be used for generation of new rule sets. A possible set of cluster-identification parameters is shown in Figure 7.

M. Wolverton and B. Hayes Roth's [14] work on Knowledge-Directed Spreading Activation seems to be a very applicable technology in our context for case retrieval in the following manner. It retrieves analogical cases stored in a large semantic network by using task-specific knowledge to guide a spreading activation search to a case or concept in memory that meets a desired similarity criterion. Both similarities and dissimilarities guide the search process. Thus, if knowledge about clusters and their (dis) similarities with each other could be stored in an appropriate fashion in the CID, this technology could be overlaid on the CBR commercial tool's functionality so as to make it applicable for retrieval of rule sets generated from large multi-use knowledge-base systems through the MVP-CA tool.

We showed the feasibility of taking a CID for a representative cluster such as specified above, and populating the case base with the appropriate features to index into the clusters. An index is really a piece of information about the cluster that can be stored in a computational data structure so that it can be searched and retrieved quickly. We do provide a mechanism in our interface to store unindexed information as well with each cluster, because it may provide contextual information that could be of value to the user, but which may not play a role in the retrieval process.

```
CONTINUOUS BGM RULE DESCRIPTION:  Trigger when MF_QUALITY is Good
(defrule tlm_qual1
  (bgm-rule tlm_qual1 on)
  (Mission MF_QUALITY#XTE_DECOM ~Nodata)
  (Inferred fsync_lock_occurred yes)
  =>
  (AssertFact "Inferred Telem_Quality Good")
  (SendMessage "MessageWindow" Status "MF_QUALITY indicates Telem quality is Good.") )

;CONTINUOUS BGM RULE DESCRIPTION:  Trigger when MF_QUALITY is Bad
(defrule tlm_qual2
  (bgm-rule tlm_qual2 on)
  (Mission MF_QUALITY#XTE_DECOM ~Good)
  (Inferred fsync_lock_occurred ~yes)
  =>
  (AssertFact "Inferred Telem_Quality Bad")
  (SendMessage "MessageWindow" Warning "MF_QUALITY indicates Telem quality is not Good.")

;CONTINUOUS BGM RULE DESCRIPTION:  Trigger when MF_QUALITY drops out
(defrule tlm_qual3
  (bgm-rule tlm_qual3 on)
  (Mission MF_QUALITY#XTE_DECOM ~Good)
  (Inferred fsync_lock_occurred ~yes)
  (Inferred Telem_Quality Good)
  =>
  (AssertFact "Inferred Telem_Quality Bad")
  (SendMessage "MessageWindow" Warning "MF_QUALITY indicates Telemetry has dropped out.")
```

Figure 5: Reusable Rule Set in XTE Rulebase

5. Conclusions

We have shown that the MVP-CA prototype tool is able to extract various views of expert systems through the clustering of rules. The rule clusters form a basis for understanding the system for various software engineering activities because they are suggestive of various rule-models inherent in the software system. Information can be fused from various reusable clusters to develop new mission systems. Even though the technolgy has been applied to expert systems, it is applicable to any information system which has a regular grammar.

Given the successful development of the MVP-CA tool, software developers will be in the position to leverage the knowledge of existent systems in building new ones in a reliable and efficient manner.

Acknowledgements

This research was supported by a Phase I SBIR Air Force contract F29601-98-C-0102. Our thanks to Ross Wainwright from AFRL and Dr. Sergio Alvarado from Aerospace as well as Walter Truszkowski and his colleagues at NASA Goddard Space Flight Center, for providing us their knowledge bases and valuable technical assistance.

References

[1] S.J.Alvarado. An Evaluation of Object-Oriented Architecture Models for Satellite Ground Systems. *In Proceedings of the Ground System Architecture Workshop, GSAW98.* The Aerospace Corporation. El Segundo, CA. http://sunset.usc.edu/GSAW/1998

[2] K. L. Bellman and D. O. Walter. Analyzing and correcting knowledge-based systems requires explicit models. *In AAAI-88 Workshop on Verification, Validation and Testing of Knowledge-Based Systems, July 1988.*

[3] B. Chandrasekharan. Generic tasks in knowledge-based reasoning: High-level building blocks for expert systems design. IEEE Expert, Fall 1986.

```
defrule <name-of-rule>
    (bgm-rule <name-of-rule> on)
   (Mission MF_QUALITY#XTE_DECOM <data-quality-value>)
   (Inferred fsync_lock_occurred <f-l-value>)
 =>
   (AssertFact "Inferred Telem_Quality <i-t-value>")
   (SendMessage "MessageWindow" <status-value>
          "MF_QUALITY indicates Telem quality is <t-value>.") )
```

Figure 6: Reusability of XTE Rules using templates

Cluster-identification: *Input data-quality derives telemetry-quality*		
Parameters:	Mission-name:	*MISSION*
	Process-name:	*XTE-DECOM*
	Input-data-source:	*MF-Quality*
	Output:	*Telem-quality*
	.....	

Figure 7: Cluster Indexing, Retrieval and Adaptation in XTE

[4] W. J. Clancey. Classification problem solving. In Proceedings, National Conference on Artificial Intelligence, pages 49-55, 1985.

[5] R. J. K. Jacob and J. N. Froscher. A software engineering methodology for rule-based systems. *IEEE Transactions on Knowledge and Data Engineering*, 2(2): 173-189, June 1990.

[6] C. Landauer. Correctness principles for rule-based expert systems. *Journal of Expert Systems with Applications*, 1:291-316, 1990.

[7] S. Lindsay. COTS AI Technologies for Satellite Control. *Proceedings of the Ground System Architecture Workshop, GSAW98*. The Aerospace Corporation. El Segundo, CA. http://sunset.usc.edu/GSAW/1998

[8] M. Mehrotra and C. Wild. Analyzing knowledge-based systems using multi-viewpoint clustering analysis. *Journal of Systems and Software*, 29:235-249, 1995.

[9] M. Mehrotra. Requirements and capabilities of the multi-viewpoint clustering analysis methodology. In Notes for the IJCAI-95 Workshop on Verification, Validation and Testing of Knowledge-Based Systems, Montreal, Canada, August 1995.

[10] M. Mehrotra. Application of multi-viewpoint clustering analysis to an Expert Systems Advocate Advisor. Technical Report FHWA-RD-97022, Federal Highway Administration, Pragati Final Report, Yorktown, VA., August 1996.

[11] M. Mehrotra, S. Alvarado and R. Wainwright Laying a Foundation for Reusability of Knowledge Bases in Spacecraft Ground System. *In Proceedings of the Ground System Architecture Workshop, GSAW99*. The Aerospace Corporation. El Segundo, CA. http://sunset.usc.edu/GSAW/1999.

[12] M. Mehrotra, S. Alvarado and R. Wainwright. Laying a Foundation for Software Engineering of Knowledge Bases in Spacecraft Ground Systems. To Appear in Proceedings of FLAIRS-99 Conference to be held May 3-5th, 1999 in Florida.

[13] S.R. Turner and P. Mangan. Reusable Expert Systems. In *Proceedings of the Fourth International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems*, 780-784. June 1991.

[14] M. Wolverton and B. Hayes-Roth. Retrieving semantically distant analogies with knowledge-directed spreading activation. *In Proceedings, National Conference on Artificial Intelligence*, pages 56-61, 1994.

Thesaurus Entry Extraction from an On-line Dictionary *

Jan Jannink
Computer Science Department
Stanford University
Stanford CA, 94305, U.S.A.
jan@db.stanford.edu

Abstract *The diversity and availability of information sources on the World Wide Web has set the stage for integration and reuse at an unparalleled scale. There remain significant hurdles to exploiting the extent of the Web's resources in a consistent, scalable and maintainable fashion. The autonomy and volatility of Web sources complicates maintaining wrappers consistent with the requirements of the data's target application. This paper describes the ArcRank model of relationships between nodes in a directed labeled graph, such as hypertext. The paper presents a ranking algorithm for directed arcs, and the algorithm for extraction of hierarchical relationships between words in a dictionary. Using ArcRank we create a thesaurus style tool to aid in the integration of texts and databases whose content is similar but whose terms are different. These algorithms complement handcrafted thesauri, by determining more complete relationships between words, although they are less specific. Exploiting hierarchies of relationships between words paves the way for broadening and related term queries in web-based repositories.*

Keywords: relationship rank, semantic heterogeneity, thesaurus, extraction

* This work was supported by a grant from the Air Force Office of Scientific Research (AFOSR).

1 Introduction

The principal obstacle in integrating information from multiple sources is their semantic heterogeneity. The most easily recognized form of heterogeneity is when different terms are used to mean the same thing: lexical heterogeneity. Even so, there is no algorithmic procedure to authoritatively resolve problems of lexical heterogeneity. However, we still desire assistance in determining semantically related terms.

Our experiments use an on-line version of the 1913 Webster's dictionary that is available through the Gutenberg Project [1]. The original dictionary is a corpus of over 50 MB containing some 112,000 terms, and over 2,000,000 words in the definitions alone. We have been working on the problem of automatically extracting thesaurus entries, using the following graph structure: each head word and definition grouping is a node, each word in a definition node is an arc to the node having that head word.

After accounting for the most common problems in constructing the graph, a naive script mis-assigns over five percent of the words, because of differences between the actual data in the dictionary and its assumed structure. Errors in the computation of the graph would affect any subsequent computation of related terms for the thesaurus application. Therefore, we set a goal of 99% accuracy in the conversion of the dictionary data to a graph structure.

Using a novel algebraic extraction technique we were able to generate such a graph structure and then use it to create thesaurus entries for all words defined in the structure including *stop words* such as 'the', 'a', 'and' that most sytems specifically list so as to ignore. The thesaurus engine, based on our relationship ranking technique, constructs more complete repositories than manually constructed thesauri, although they are less specific. It is a potentially important tool for systems integration experts.

1.1 Related work

Some early work on constructing taxonomies[2] and extracting *semantic primitives* [3] used a graph generated from the dictionary definitions. Examples of *lexical knowledge bases* that relate terms according to some two dozen relationships, are the handcrafted WordNet [4], and MindNet [5]. MindNet is generated by phrase parsing in the dictionary.

PageRank[6] is the algorithm that underlies the material in this paper. Algorithms that operate on a matrix representation of word graphs include LSI [7] and *hubs and authorities* [8]. WHIRL [9] attempts database integration using novel IR based textual similarity queries.

1.2 Motivation

The starting point for this work is the hypothesis that structural relationships between terms are relevant to their meaning. These relationships become interesting when all items in the domain of interest contain them, and are organized according to them. Dictionary definitions form a closed domain in the sense that the set of words used in definitions are defined elsewhere in the dictionary. This property leads to a directed labeled graph representation of the dictionary. Nodes of the graph model definitions, head words are labels for the nodes, and a word in a definition represents an arc to the node having that word as a label. Notable collections which are not closed include encyclopedias, which cover a set of terms equivalent to the dictionary nouns, and search engines, which return documents for all but stop words.

At first glance, the PageRank model of Web structure does not lend itself to direct application in non-hypertextual domains. However, we have found that a related model, which we call ArcRank, is useful for extracting relationships between words in a dictionary. This model expresses the importance of a word when used in the definition of another. The attraction of using the dictionary as a structuring tool is precisely that head words are distinguished terms for the definition text. This extra information allows types of analysis that are not currently performed in traditional data mining, and IR, where no term is assigned as 'head word' of a document. Interestingly, we now find that this new analysis may also be applied to document classification and the ranking of results of mining queries.

2 Background

In this section we present the basis of our dictionary structuring techniques. Before presenting the ArcRank measure, we present the PageRank algorithm, and the variants we have used in our experiments.

2.1 Graph Extraction

Substantial manipulation is required to bring the dictionary data into a format ready for generating a graph [10]. Head words and definitions are in a many to many relationship since head words have variant spellings and definitions have multiple differing senses. Other problems in the transformation process are listed below.

- syllable and accent markers in head words
- misspelled head words
- accents and special characters
- mis-tagged fields
- common abbreviations in definitions (etc.)
- stemming and irregular verbs (Hopelessness)
- multi-word head words (Water Buffalo)
- undefined words with common prefixes (Un-)
- undefined hyphenated and compound words (Sea-dog)

Table 1: PageRank

input: directed graph, *output*: scored node list

1. Make adjacency list representation of directed graph
2. Make rank array of size $|n|$ for graph nodes
3. Set (round 0) rank $p_{0s} = 1/n$ for all nodes s
4. While `rankchange` > `threshold` (round i)
5. For nodes s in $\{1 \cdots |n|\}$ (ranking step)
6. For arcs $a_{s,t}$ in s's adjacency list a_s
7. Transfer rank $p_{is}/|a_s|$ from source s to target t
8. For nodes s in $\{1 \cdots |n|\}$ (adjustment step)
9. Normalize, if needed, rank p_{is} wrt to total rank
10. Compute `rankchange` from previous iteration
11. Return final values from rank array

For example, when a conjugated verb form appears as a head word we use it for generating graph arcs. Otherwise we stem definition words until we find a head word that matches. Also, whenever we find instances of a multi-word head word in the definitions, we prefer it over the individual words for generating a graph arc. Since words often appear multiple times in a single definition we allow multiple arcs between graph nodes. Dealing with undefined terms and spelling errors is the most complex issue in the graph generation, and accounts for the quasi-totality of the structural errors in the graph. In the following we define the algorithms that run on the graph structure.

2.2 PageRank

The PageRank algorithm forms the basis of the ranking technique described in this paper, and is important to define before discussing the ranking of arcs. Table 1 below is a pseudocode description of the algorithm:

This algorithm is a flow algorithm which assumes no capacity constraints on the arcs between nodes. All nodes begin with an initial ranking, in our case a constant $1/|n|$, where $|n|$ is the number of nodes in the graph. At each iteration, nodes distribute their rank to their neighbors on outgoing arcs, and receive rank from neighbors on incoming arcs. The total outgoing flow from a node is never greater than its rank, $\sum_t a_{s,t} \leq p_s$, nor is any individual $a_{s,t}$ ever less than zero. The intuition behind the flow is that more richly connected areas of the graph carry larger capacity, and therefore nodes in these areas maintain a higher rank. The rank flow of nodes in strongly connected aperiodic graphs is shown to converge to a steady state [11]. Steady state flow is desirable, because it allows us to assert stable relationships between nodes in the graph. In practice, we accept variability in the flow between nodes, so long as the total variability over the entire graph lies below a threshold.

In general graphs, nodes and clusters of nodes with only outgoing arcs act as sources which lose all of their rank. Likewise, nodes with incoming arcs only act as sinks for the rank of their neighbors. The dictionary graph contains both source and sink nodes: sources are words which are never used in other words' definitions, sinks are words whose definitions are not found in the dictionary. In our application sinks consist of misspellings, proper nouns such as geographical and Latin species names, and scientific formulae, which we do not consider. In PageRank the rank of sources, sinks and weakly connected clusters do not reflect their structural differences well. In our algorithm the final rank of a node should be defined in such a way that when any two nodes have a distinct pattern of connections, then their rank will differ. We adapt the algorithm from Table 1 in one of the following three ways so that sources and weakly connected clusters preserve some rank at each iteration.

1. redistribute $b\%(b/100)$ of total graph rank before each iteration
2. limit rank transfer to a fraction $1/c$ of a node's rank
3. add a self-arc $a_{t,t}$ (node t is both source and target) to nodes

By selecting a non zero threshold for termination of PageRank, and one of the above adaptations, we ensure that all graph nodes preserve a non zero rank. We show here that, given a node t, at iteration i with rank p_{it}, the following holds:

Theorem 1 $\boxed{\forall t \in G, p_{it} > 0}$

Proceeding by induction, we have: by definition, at the initial iteration, $p_{0t} = 1/n > 0$. *Assuming the property holds at iteration i, the following holds:*

$$p_{i+1t} = \{b/100, 1/c, p_{it}/(|a_t| + 1)\} + \sum_{v \neq t} a_{v,t}$$

Since, by definition all quantities on the right hand side are positive and greater than zero, p_{i+1t} *is greater than zero. As indicated by the equation, this property holds for each PageRank variant enumerated above.*

We see that PageRank for dictionary terms represents the transitive contribution of each term to the definitions of all of the dictionary terms. We capitalize on this property to compute the relative importance of terms with respect to each other. This measure is a feature of the arcs between nodes, or equivalently in the dictionary, the usage of terms in the definitions of others.

2.3 Relative Arc Importance

In the dictionary application, PageRank suffers from some inherent limitations. First of all, PageRank is inherently a node oriented algorithm. The top ranked nodes are the common conjunctions and prepositions, which convey little conceptual meaning, and are commonly considered stop words by other applications. It is clear that on its own, PageRank is insufficient to conceptually organize the dictionary structure. We may consider an extension to PageRank which assigns to each arc the amount of rank that flows across it at each iteration. As an absolute measure, this extension is also unsatisfactory, because it favors flows between the most highly ranked terms, that is, between stop words. Besides this obvious extension, there appears to be no self-evident technique to extract an absolute arc-based measure from PageRank.

However, our original goal is to identify the most important arcs for a given individual node. By casting our ranking problem in terms of our original goal we see that rather than an absolute measure, a relative measure between nodes is preferable. For any term in the dictionary, the words that signify the most in their definition should correspond to the arcs in the graph which are most significant in a ranking of arcs. Hence we arrive at the relative measure of arc relevance. Given an edge e, having source node s with rank p_s, target node t with rank p_t, and given $|a_s|$ outgoing arcs from s, the arc relevance r for e is defined as:

$$r_e = \frac{p_s/|a_s|}{p_t}$$

When s and t share several (m) edges $e_1 \ldots e_m$, we sum the arc ranks to compute the importance of t in the definition of s:

$$r_{s,t} = \sum_{e=1}^{m} \frac{p_s/|a_s|}{p_t}$$

$r_{s,t}$ measures the relative contribution of the rank of s to the rank of t which we show has desirable properties, such as:

Theorem 2 $\boxed{0 < r_{s,t} \leq 1}$

This follows directly from Theorem 1 and the definition of p_t, *since both numerator and denominator must be positive and* $p_t = \sum_v p_v/|a_v| = p_s/|a_s| + \sum_{v \neq s} p_v/|a_v| \Rightarrow p_t \geq p_s/|a_s|$.

Note that the arc importance measure is an indicator valid only in the immediate local vicinity of the end points of the arc. There is no reason to expect it to be globally commensurate. Having established an arc importance measure we are ready to present the ArcRank algorithm, and walk through a hierarchical set of relationships the algorithm uncovers.

3 ArcRank

In the previous section, we have computed a relative measure of arc importance. Here we show how to rank it with respect to both the source and target nodes, to promote arcs which are important to both endpoints. We discuss the repository we construct using ArcRank, and compare it to other systems.

3.1 ArcRank Algorithm overview

The ranking of an arc according to the arc importance metric defined above is typically different at the source and the target node. Indeed, it is possible for the highest arc importance value of arcs from a source node to be the lowest value for arcs coming into the target node. ArcRank, defined in Table 2 below, computes a mean of the ranked importance of arcs, so as to promote arcs which are important both to the source nodes and to the target nodes.

Table 2: ArcRank

input: triples (source s, target t, arc importance $v_{s,t}$)

1. given source s and target t nodes
2. at s, sort v_{s,t_j} and rank arcs $r_s(v_{s,t_j})$
3. at t, sort $v_{s_i,t}$ and rank arcs $r_t(v_{s_i,t})$
4. compute ArcRank: $\text{mean}(r_s(v_{s,t}), r_t(v_{s,t}))$
5. Rank Arcs *input*: sorted arc importance
   - sample values $\{0.9, 0.75, 0.75, 0.75, 0.6, 0.5, \ldots, 0.1\}$
   - equal values take same rank $\{1, \mathbf{2}, \mathbf{2}, \mathbf{2}, \ldots\}$
   - number ranks consecutively $\{1, 2, 2, 2, \mathbf{3}, \ldots\}$

Other rank numbering techniques resulted in skewed output. Competition style ranking, which counts equal values equally, but orders subsequent values differently, disadvantages arcs to nodes with many in-arcs. Given the same sample values from the above, the boldface value in the list here shows where this ranking differs: $\{1, 2, 2, 2, \mathbf{5}, 6, \ldots\}$. Also, computing rank as a fraction of the total number of ranks: $\{1/n, 2/n, \ldots, \mathbf{n/n}\}$ favors arcs to nodes with a larger number of distinct ranks.

The ArcRank algorithm is more space intensive than PageRank, because it is arc oriented, but is fast and easily made into a disk based version. It essentially requires two passes through the data, and storage for twice the number of arcs. In the course of developing ArcRank, we derived a further extension to PageRank. The idea is to vary according to the arc importance ratio the amount of a source node's rank transfered to the targets. Tuning this optimization properly strengthens strong relationships, weakens less important ones. The additional cost is minimal, and requires ranking arcs and summing ranks per node, before pushing value across arcs.

3.2 The Webster's Repository

The repository we have built [12] has a very general structure, and it is defined by usage alone. There are no preimposed limitations, based on grammatical models, as to how terms relate. As it is very general, the structure also sidesteps problems of parsing the part of speech for each term and handling general negation. This repository is the only one which does not exclude stop words, and as a result we are able to find that stop words most strongly relate to each other. On the down side, the type of relationship expressed in the repository is not always self evident, especially since many definitions and terms are now obsolete. Also, the accuracy of the ArkRank measure increases with the amount of data, and much of the dictionary contains very sparse definitions. Due to this sparseness we often find that ArcRank will promote arcs to lower ranked targets. Also, misleadingly, the sparseness of data makes a simple metric of ranking sources by the paucity of arcs work well, when it would otherwise fail.

3.3 Comparison to Other Systems

MindNet is not publicly available, but its scale is 159,000 head words and 713,000 relationships between head words. Its development began in 1992, and it supports 24 different relationships between terms. It appears that it suffers from problems, both in terms of accuracy and completeness of extraction.

WordNet has been in development since 1990, and its design has been elaborated since 1986. Its current revision, **WordNet 1.6** was released in 1998, and includes four principal data files, and a number of executables to aid in searching and displaying the data. Of the existing electronic lexical tools, WordNet is the one that most closely resembles the Webster's repository.

The relationships WordNet defines between terms are more precise, as they were manually entered, however there are necessarily fewer of them, and they are far from exhaustive. Also, since the design of WordNet long preceded its implementation, artificial concepts, such as non-existant words, and artificial categorizations, , such as non-conforming adjectives, were introduced when the repository was built. These constructs are a valid ad hoc approach to make the terms conform to the design, but they do not arise out of the usage of the language. WordNet carefully distinguishes between senses of a term, and separates a term into multiple entries when it may be used as different parts of speech, i.e., to *run* vs. a computer *run* vs. a *run* salmon. The Webster's repository only distinguishes senses of a term based on usage, not on grammar. Another significant difference between the two structures is that the data in WordNet is separated by lexical categories, whereas the Webster's repository allows any relationship between terms to exist. Table 3 makes some simple numerical comparisons between the two systems.

Having compared the repositories numerically, it is necessary to illustrate with an example what the Webster's repository provides. Specifically, it relates terms without defining the type of relationship, just the importance of the relationship. The following section gives an example of terms relating to transportation.

4 Word Relationships

In this section we examine some subgraphs that emerge from the repository data after applying the ArcRank measure. For lack of space we can not cover the full array of relationships present in the dictionary, which extend even to stop words for the other repositories.

4.1 Browsing the Webster's Repository

It is instructive to browse through the repository to get an idea of how it organizes the dictionary terms. The example below is prompted by an interest in developing a transportation ontology to support logistics applications. We start at the term **Transport** as shown below in Figure 1. The general form of graphs generated using the repository, such as Figure 5, frame a term by terms used in its definition above and terms that use it in their definition below. These terms are placed from left to right in order of their ArcRank measure. No more than the two dozen most significant associated terms are displayed: the label for the central term contains a count of incoming and outgoing arcs of the form <*outgoing, incoming*>. In addition to the ArcRank measure on arcs, each term has an associated PageRank value. Arcs and Term borders are dotted when the arc's direction is the reverse of the PageRank ordering of its end points.

In Figure 1, which has been further pruned for clarity, we see that the term **Convey** is used in transport's definition. When we next examine the term graph for convey, Figure 2, we find transport, along with transported, and cargo which are also significant for the logistics ontology. Other terms in the set illustrate the more general nature of convey as compared to transport.

Further browsing in the repository takes us to the graph for **Carry** in Figure 3. Note how

Table 3: Comparison of Webster Repository and WordNet 1.6

Name	Size	Comment
Webster	96,800 terms	~ four man months of effort
	112,897 distinct words	(including variant spellings)
		error rates
		<1% of original input (spelling errors, etc.)
		<0.05% incorrect arcs (hyphenation)
		<0.05% incorrect terms (spelling)
		0% artificial terms
WordNet 1.6	99,642 terms	2 profs, students, volunteers, 8-12 years
	173,941 word senses	(including numbers, repetition of terms)
	66,025 nouns	**disjoint files**
	12,127 verbs	
	17,915 adj.	
	3,575 adv.	
		error rates
		~0.1% inappropriate classifications
		~1-10% artificial & repeated terms

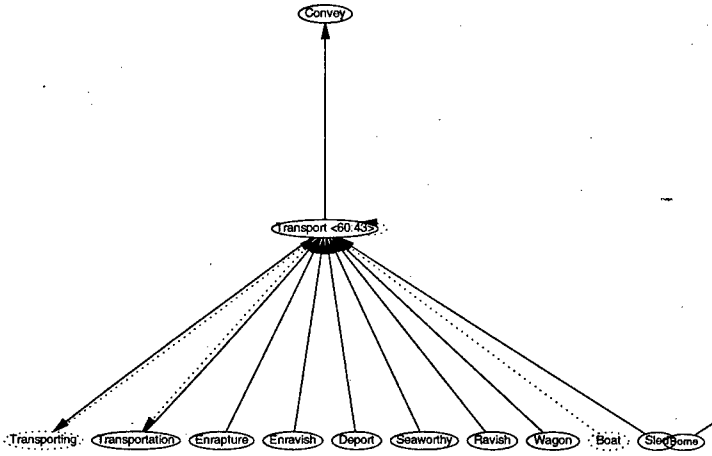


Figure 1: Terms Relating to Transport

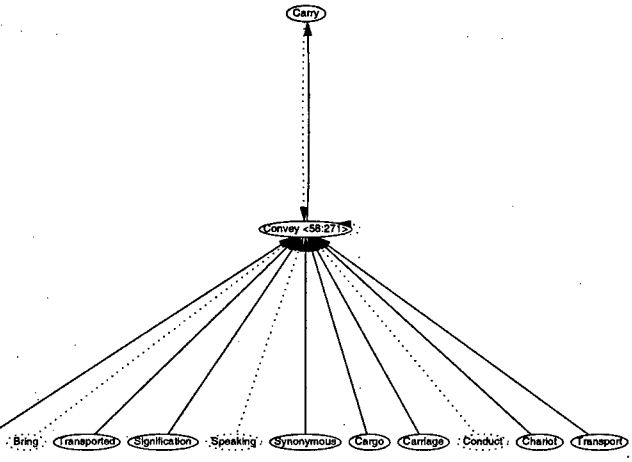


Figure 2: Convey Generalizes Transport

carry subsumes convey in the sense of transport, and that the term transported is also in its set of terms. We expect too that **Hold** expresses a more general notion relating to carry.

Starting from transport in the other direction, we select **Wagon** and consider Figure 4. Wagon is not a specialization of transport, although transport does subsume it: a wagon is one of a number of forms of transport. We see that terms such as **Car** and **Vehicle** also shown in Figure 4 represent the generalization relationship for wagon. Also, terms such as **Charioteer**, **Caravan** and **Wheelwright** relate to wagon without being specializations. bf Locomotive is however a specialization, and we next consider the graph in Figure 5.

The graph for locomotive illustrates a spectrum of relationships between terms, some of which are altogether unexpected, such as locomotive's relationship to the term **Appendix**. A glance at the definition of locomotive reveals that a reference to an illustration in the appendix of the dictionary appears inappropriately in the definition field of the term. The other associated terms all respect some subsuming or entailment relationship to locomotive.

Having traveled through a very small sam-

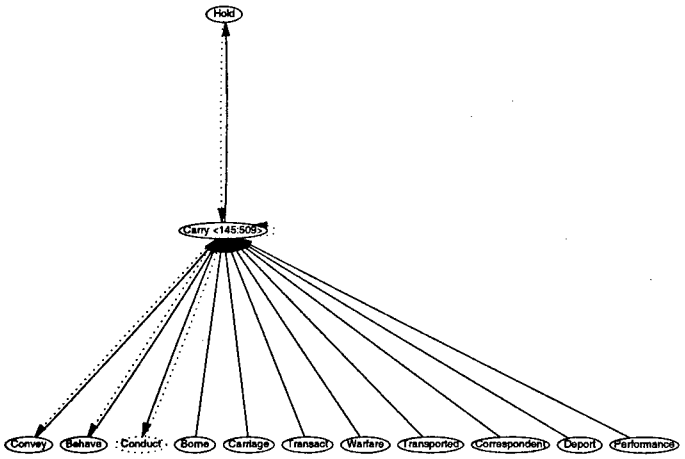


Figure 3: Carry Subsumes Convey

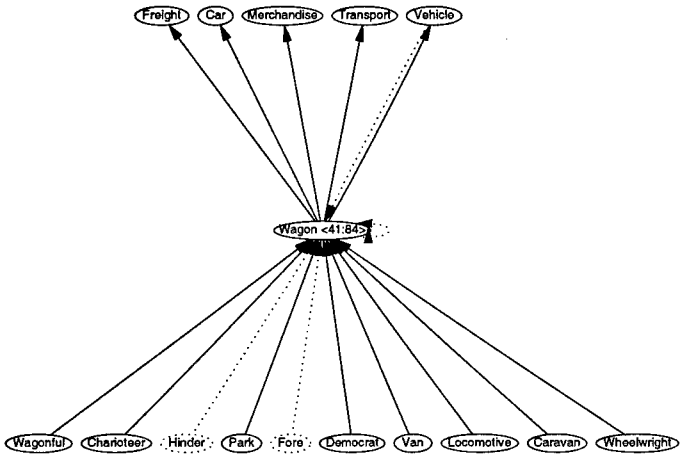


Figure 4: Wagon as a Means of Transport

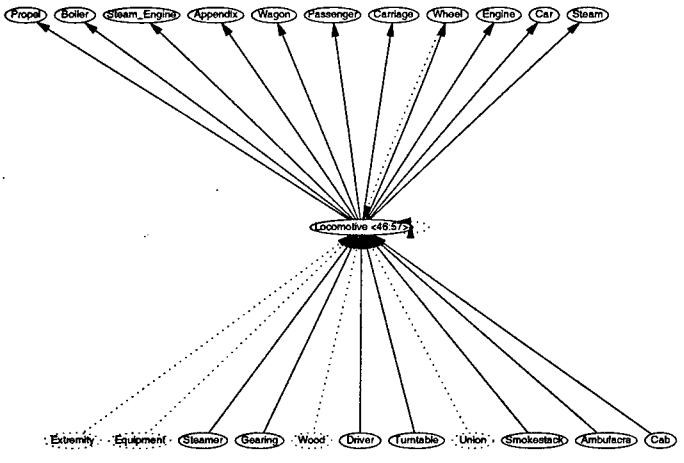


Figure 5: Locomotive Specializes Wagon

ple of the structure of the repository, it becomes clear that the ordering itself is not sufficient to automatically extract the significant terms relating to a given term. The algorithm to achieve this is the basis for the application we are building on top of the repository, and discussed in the following section. As it turns out the rankings provided by PageRank and ArcRank enable an efficient extraction procedure to maintain structure that is confirmed by relationships with other terms.

5 Applications

In this section we discuss applications of these new algorithms, and current directions of our research.

5.1 Relation Extraction

Having a repository with rank relationships between terms, it becomes possible to extract groups of related terms based on the strengths of their relationships. In particular, we are interested in extracting three relationships: *subsuming*, *specializing* and *kinship*. The kinship relationship is a similarity relationship broader than synonymy. We are able to achieve this extraction using a new iterative algorithm, based on the Pattern/Relation extraction algorithm [13], as follows in Table 4.

Table 4: Extract Relation

input graph with ArcRank computed, & seed arc set, *output* local hierarchy based on seed arc set

1. Compute set of nodes that contain arcs comparable to seed arc set
2. Threshold them according to ArcRank value
3. Extend seed arc set, when nodes contain further commonality
4. If node set increased in size repeat from 1.

The output of the algorithm computes a set of terms that are related by the strength of the associations in the arcs that they contain. These associations correspond to local hierarchies of subsuming and specializing relationships, and the set of terms are related by a

kinship relationship. The algorithm is naturally self-limiting via the thresholds.

This approach allows us to distinguish senses of terms when they engender different structures according to the algorithm. Indeed, the senses of a word such as *hard*, are distinguished by the choice of association with *tough* and *severe*. Also, ranking the different senses of a term by the strength of its associations with other terms allows us to uncover the principal senses of a term.

We are currently investigating the utility of the ArcRank algorithm for traditional document classification applications, as well as to rank the association rules resulting from data mining queries. We are also using the results of the relation extraction algorithm to aid in the resolution of semantic heterogeneity in our ontology algebra research.

6 Conclusion

In this paper we have presented algorithms for ranking relationships represented in a graph structure. We have applied these algorithms to a graph extracted from an on-line dictionary to uncover the strongest relationships between dictionary terms, as given by term usage, rather than grammatical categorization. We consider this repository an adjunct, not a replacement, for handcrafted thesauri, to aid in the integration of disparate information sources, by reducing the effects of their lexical heterogeneity.

References

[1] PROMO.NET. Project Gutenberg. http://www.gutenberg.net/, 1999.

[2] R. Amsler. *The Structure of the Mirriam Webster Pocket Dictionary.* PhD thesis, University of Texas, Austin, 1980.

[3] D. P. Dailey. On the search for semantic primitives. *Computational Linguistics*, 12(4):306–307, 1986.

[4] G. A. Miller and *al.* Five papers on WordNet. Technical Report 43, Cognitive Science Laboratory, Princeton University, 1990.

[5] S. Richardson, W. Dolan, and L. Vanderwende. MindNet: acquiring and structuring semantic information from text. In *Proceedings of COLING '98*, 1998. ftp://ftp.research.microsoft.com/pub/tr/tr-98-23.doc.

[6] L. Page and S. Brin. The anatomy of a large-scale hypertextual web search engine. *Proceedings of the 7th Annual World Wide Web Conference*, 1998.

[7] S. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41(6):391–407, 1990.

[8] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998. http://simon.cs.cornell.edu/home/kleinber/auth.ps.

[9] William W. Cohen. Integration of heterogeneous databases without common domains using queries based on textual similarity. In *ACM SIGMOD '98*, pages 201–212. ACM, 1998.

[10] J. Jannink and G. Wiederhold. Ontology maintenance with an algebraic methodology: a case study. In *To appear: Proceedings, AAAI Workshop on Ontology Management*, 1999. http://www-db.stanford.edu/SKC/papers/summar.ps.

[11] R. Motwani and Raghavan P. *Randomized algorithms.* Cambridge University Press, New York NY, 1995.

[12] Jan Jannink. Webster's dictionary repository. http://skeptic.stanford.edu/data/, 1999.

[13] S. Brin. Extracting patterns and relations from the world wide web. http://www-db.stanford.edu/ sergey/booklist.html, 1998.

Session WB2
Hardware for Information Fusion
Chair: Adrian Stoica
NASA Jet Propulsion Laboratory, CA, USA

Extended Logic Intelligent Processing System For Sensor Fusion

Adrian Stoica, Tyson Thomas, Wei-Te Li, and Taher Daud
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

James Fabunmi
AEDAR Corporation,
P.O. Box 1469, Landover, Maryland

Abstract - The paper presents the hardware implementation and initial tests from a low-power, high-speed reconfigurable sensor fusion processor. The Extended Logic Intelligent Processing System (ELIPS) is described, which combines rule-based systems, fuzzy logic, and neural networks to achieve parallel fusion of sensor signals in compact low power VLSI. The development of the ELIPS concept is being done to demonstrate the interceptor functionality, which particularly underlines the high speed and low power requirements. The hardware programmability allows the processor to reconfigure into different machines, taking the most efficient hardware implementation during each phase of information processing. Processing speeds of microseconds have been demonstrated using our test hardware.

Keywords: sensor, fusion, processor, hardware, fuzzy, expert, neural, networks, reconfigurable.

1. Introduction:

1.1. A general need for sensor fusion processors:

With the advent of high-performance sensors and increased processing power more real time applications are now possible. Novel architectures, algorithms, and hardware are required to address the challenges of high sensor bandwidth and the often noisy, sometimes contradictory data present in these new applications. The problem of using more sensors with higher data rates is combined with the need for faster response in real time scenarios, which demands higher levels of computational power. The traditional approach is to build/use increasingly powerful general-purpose processors. Yet, classical algorithms for fusing data (originating in preponderant Bayesian approaches) face challenges in addressing the sensor-fusion problem and more novel approaches, such as the ones coming from the computational intelligence research, can complement or replace the traditional schemes.

Computational intelligence techniques, such as fuzzy logic and neural networks combined with the more traditional Artificial Intelligence paradigm of expert systems proved efficient in solving a category of problems for which an accurate mathematical formulation of models was either not feasible or practically impossible to compute in useful time. The most pertinent examples of such problems are in pattern recognition and decision-making applications. These techniques are essentially parallel, and thus it is natural to build dedicated processors efficient for these types of operations, which would function in stand-alone mode or as co-processors to provide high-speed computation on massive amounts of data in parallel mode. While these processors can be built both in digital or analog hardware, the massive amount of interconnection lines of a parallel implementation and the power requirements encountered in certain space, military or commercial applications such as hand-held devices make the idea of an analog ASIC processor preferable. An example of such an application requiring low power and fast processing of sensor data is associated with the discrimination performed onboard interceptors.

1.2. Discriminating Interceptor Technology requirements for an on-board sensor fusion processor:

The Ballistic Missile Defense Organization (BMDO) is conducting the Discriminating Interceptor Technology Program (DITP) for the development of advanced and enabling fast frame seeker capabilities. The challenge for the technology is to combat more complex future threats facing the National and Theater Missile Defense (NMD/TMD). The objective is to develop miniaturized interceptor components and subsystems to meet serious space, weight, and power constraints [1]. In this regard, part of a major effort is directed towards the development of new sensor data fusion processing technology that will particularly address high speed and on-board autonomy. This capability can achieve earlier target acquisition, thereby extending the time-to-engage and reducing the dependence on the external battle management and off-board surveillance assets[1].

Once the initially required off-board battle management intelligence is provided to the seeker, the primary goal of the DITP is to exploit the multi-phenomenological sensor data obtained from on-board LADAR and infrared detector arrays for threat engagement via development and integration of real-time sensor fusion algorithms and processors. The overriding hypothesis is that sensor data fusion at three levels (i.e., signal, feature, and decision) is

necessary to improve its capability and to accommodate a wide variety of missions and targets.

In order to meet the challenge of compact, low power, and high-speed on-board data processing, a novel intelligent sensor data fusion processing architecture, termed the Extended Logic Intelligent Processing System (ELIPS), has been developed. ELIPS integrates the analog hardware technology of neural networks, fuzzy logic, and expert rule processing with the conventional digital processing using a host computer. The individual modules are designed to be reconfigurable and cascadable. In addition, the overall architecture has been developed to be flexible enough for rerouting of signals to any required processing module by having an interconnecting network with switching arrays.

This paper briefly describes the ELIPS concept and architecture, focusing more on the hardware implementation of the individual ELIPS component modules. Experiments with test chips implementing ELIPS modules illustrate the performance of the analog ASIC implementation.

2. Fuzzy, Expert, And Neural Computation:

Expert systems have been employed in a variety of sensor fusion applications; a recent example is detailed for guiding the user in defining the architecture for the sensor fusion system[2]. Fuzzy logic and neural networks are also becoming widely accepted in the sensor fusion community as techniques with proven capabilities in sensor fusion applications[3-4].

Conditional rule-based systems are using rules of the form "IF a is A AND b is B THEN y is Y' where a, b, and y are the input and output variables respectively, and A, B, Y are classes - in particular fuzzy classes/sets. Thus, a rule-base system can be seen as accepting input data from measurements or preprocessing and providing outputs as transformed by the rules. In particular the outputs could be associated with classes to which the inputs cluster and the magnitude of the outputs associated to the degree of membership to these classes. (Another possible interpretation is that the numbers represent the confidence in the classification, e.g. 70% confidence that the object is target 1, 20% that it is target 2, 10% confidence that it is a decoy.)

New concepts from fuzzy set theory have revitalized the use of rule-base systems, which can cope with the imprecision in matching antecedent clauses. The main operations of fuzzy reasoning are fuzzification, rule evaluations and defuzzification. Fuzzification transforms a crisp input to a degree of membership to a fuzzy set and certain rules are evaluated depending on which fuzzy sets are matched. For certain problems such as classification, this is the end of fuzzy reasoning - the output results are fuzzy sets and degrees to which they are matched. For example, the output result can be that input signals match the characteristics of target A to 0.8 extent, targets B in degree 0.4 and decoys in degree 0.3; sometimes this can be (improperly) expressed as probabilities, i.e., there is 80% chance/probability/ confidence that object is target A, etc. If the desired output is a crisp one, for example an output control signal - the output sets and the associated degrees of memberships are transformed by a defuzzifier into a crisp value. Amongst the most popular methods for defuzzification is the center of gravity method, which requires mainly additions and multiplication.

Neural networks are parallel computation structures characterized by somatic operation between inputs and weights and somatic operations aggregating the weighted inputs and usually passing them through a nonlinear function. Different neural architectures were explored, with different ways of interconnecting the neurons in feed-forward only or in recurrent mode as well, and with a variety of learning rules.

Requirements for fast processing, compact or low power implementation lead to efforts for developing various hardware implementations. The nature of computations involved in fuzzy reasoning is essentially parallel (for example, rule evaluations are independent of each other and can be calculated concurrently). Therefore, a dedicated parallel H/W solution is preferable to a S/W solution on a general-purpose processor and even to a RISC processor with fuzzy-oriented instructions such as VY86C570 (70-microsecond inference speed)[5] or Motorola's 68HC12 (the 1st standard microcontroller family with a comprehensive fuzzy logic instruction set, and the 1st 16-bit engine for fuzzy logic)[6]. Ideally one would want to preserve high versatility of general-purpose processors while reaching low-power high-speed operation. Analog offers the advantage of lower power consumption. While better precision can be obtained in digital implementations, precise computations are not required for fuzzy processing; usually 8 bits are considered sufficient for most applications. (This is because membership functions representing fuzzy classes are usually defined by humans, who can not and do not specify fuzzy set borders with high precision - usually with less than 8 bits). Specific implementations of fuzzy processors are described in the literature[7-11].

The same parallelism is true for neural processing, and ideally H/W implementations should be parallel for maximum efficiency. Similarly for fuzzy expert systems, large number of interconnections and low power justify analog VLSI neural processors. A detailed justification of analog neural processors is presented in Ref. [12].

3. ELIPS Concept And Architecture:

The main assumption behind ELIPS is that fuzzy, rule-based and neural forms of computation can serve as the main primitives of an "intelligent" processor. Thus, in the same way as classic processors are designed to optimize the hardware implementation of a set of fundamental operations, ELIPS is developed as an efficient implementation of computational intelligence primitives, and relies on a set of fuzzy set, fuzzy inference and neural modules, built in programmable analog hardware. The hardware programmability allows the processor to be reconfigured into different machines, taking the most efficient hardware implementation during each phase of information processing.

The ELIPS architecture (Figure 1) is designed to accomplish, for the first time, a fully parallel implementation and seamless integration of three artificial/computational intelligence technologies[13]: (1) membership-function-based fuzzy logic; (2) rule-based expert systems; and (3) massively parallel artificial neural networks. In its initial demonstration, ELIPS will perform functions of discrimination, recognition, tracking, and homing [1]. It is necessary to develop a design that is hardware-implementable using very large scale integration (VLSI) technology. Additionally, it should provide an ultra low power embodiment in a compact package, with an unprecedented signal processing speed (10 to 15 microseconds for each operation), at least three orders of magnitude faster compared to a conventional digital machine (e.g. several milliseconds on a personal computer, PC).

ELIPS is envisaged as a synergistic processor incorporating four processing modules illustrated in Figure 1. PFN and PRN refer to Programmable Feed-forward and Recurrent (feedback) Neural networks, respectively, FSP is a Fuzzy Set Processor, and MERP stands for Multistage Expert Rule Processor. ELIPS modules are destined to work cooperatively in a variety of configuration sequences. For example, to implement fuzzy expert reasoning as a processing sequence of PFN, FSP, and MERP modules, fuzzification is performed by FSP, rule evaluation is done by MERP, while defuzzification (when needed) is done using the PFN.

4. Elips Building Blocks And Their Hardware Implementations:

4.1. The neural (PFN and PRN) modules:

Neural network modules are implemented around a neural chip-architecture developed at JPL[12,14]. The chip, termed NN64, consists of a 64 x 64 array of 8-bit synapses with 8-bit local static memory, 64 neurons, and registers for data and control. The chip is designed to implement a feed-forward or a recurrent neural network with various network topologies with up to 64 neurons.

4.1.1 Functional description of analog processing in NN64: The 64 analog voltage inputs first get converted to currents by a row of V-I converters at the top of the 64 x 64 synaptic array. Each V-I circuit actually produces two currents: I and 16 x I. These signals are then broadcast down each column for each of the 64 inputs so that all the synapses in a column receive the same input.

The building block for the NN64 array is a current-mode multiplying analog to digital converter (MDAC) which forms the basis of the synapse (Figure 2). A byte, which controls switches D1 to D7 to scale current copies of the input, is stored in a local static memory (SRAM) for each synapse. By switching in different multiples of the input current and adding them together, the input current is effectively multiplied by the digital weight stored in the local SRAM. The most significant bit (MSB) of the digital weight (D8+/D8-) controls the sign of the product by steering the synapse output current so that it is either sunk or sourced through the output node. Synapses on the same row have their outputs summed by attaching them all to the same wire. These 64 signals, one for each row of the array, are then sent to 64 separate neurons where they are either processed through the neuron or sent directly out, depending on how the neurons are programmed. If the neuron is on, the current is converted to a voltage through a small resistor and applied to a small differential amplifier that outputs a voltage. Should the neuron be off, the output current is routed directly out off the chip as a current.

4.1.2. Digital programming of NN6: The synapses are loaded single row at a time. The data for a given row is clocked into a 64 long 8-bit wide shift register, one byte at a time. After 64 clock cycles, the data for an entire row of synapses is ready to be loaded into the local memory of each MDAC. A 6-bit row address is supplied and an active-low load signal is asserted, which dumps the data into the synapses on the row specified. Alternatively, a synchronous loading scheme may be used. This method employs a single bit shift register to act as a token ring and specify consecutive rows for loading. When reset is asserted, the top of the token ring corresponding to row 1 is set while the rest of the shift register is reset. As data is clocked in, a 6-bit counter keeps track of how many bytes have been loaded. When the carry-out of the counter indicates that the entire data has been loaded, a load signal is automatically generated that activates

the row on its rising edge and passes the token to the next row on its falling edge. In this way the entire array of synapses can be loaded from the top row down by simply clocking in 4096 bytes of data. Neurons are also programmed with a single bit shift register. If a control signal is asserted, all neurons are automatically bypassed since the entire register is reset. Otherwise, a single bit is clocked 64 times by a special clock. The register loads from the bottom up so that the first data loaded corresponds to the first row neuron. More details on the NN64, including its configuration as a recurrent neural network can be found in the literature[14]. The chip was tested in a variety of applications where neural networks proved efficient. A particular application was interpretation of visual input data for automatic tracking of a path by a mobile robot[13].

4.2. The fuzzy set processor (FSP) module:

The main function of a fuzzy set processor is signal transformation, which can be interpreted as,

- fuzzification - i.e. association between an input crisp signal and a degree of membership to a fuzzy set/class, or
- signal conditioning/ non-linear transformation, coordinate transformation.

The FSP was designed as a processing module with 16 inputs of 5 membership classes each. The chip has 16 analog voltage inputs and 16x5 outputs, and allows digital programmability of the membership functions for each input variable. The membership functions have trapezoidal shape, with programmable parameters for the legs and slopes as illustrated in Figure 3. The position of the legs can be specified with 8-bit resolution and the slope with 5-bit resolution. The equations that describe the output of a trapezoidal membership function are:

If $X <= A$, $Y = Low$

If $A < X =< (CD+AB)/(B+C)$, $Y=MIN(BX-AB + Low, High)$

If $(CD+AB)/(B+C) < X < D$, $Y=MIN(-CX + CD + Low, High)$

If $X> = D$, $Y= Low$,

where A is the location of the left leg, B is the unsigned slope of the left leg, C is the unsigned slope of the right leg, and D is the location of the right leg. The chip design currently uses Low = 1 volt and High = 4 volts with Vdd = 5 volts.

The schematic diagram in Figure 3 details the processing path of a single membership function circuit (MFC). While inputs and outputs are in voltage mode for external compatibility, the internal MFC implementation is in current-mode. The input voltage enters the first processing block, which is a Voltage to Current (V/I) converter. Currents proportional to the digital values of the legs, A and D, are generated in Multiplying Digital to Analog Converters (MDACs). The current corresponding to the left leg gets subtracted from a copy of the input current, while a different copy of the input current gets subtracted from the right leg current. The resulting currents, which correspond to the left and right sides of the trapezoid, enter their appropriate Dividing Digital to Analog Converter (divDAC) where the signals are divided by 5-bit digital values to scale the slopes. The minimum of the two resulting values is then selected which chooses the side that is along the trapezoid. The top of the trapezoid is achieved by taking the minimum of the resulting current and the full-scale current, and this result is converted to the voltage output of the MFC. A test chip for 2 input variables with 5 membership functions calculating the degree of membership has been implemented and tested. A variety of membership functions generated by the chip is illustrated in Figure 4.

Signals obtained from the chip are also illustrated below in a discrimination task. The results are compared with the software implementation and show accurate reproduction in hardware of the results obtained by simulation. Figure 5 shows an example of how the membership functions are used to separate the spaces containing targets and decoys. The software simulated membership function shapes are compared with the programmed hardware output of the membership as shown in the lower graph in Fig. 5. The variables are transformations of some measured parameters characterizing target and decoy signals. The software results show that signals processed using these membership functions would result in discrimination of targets and decoys, as well as targets of different types based on available DITP data. Figure 5 shows discrimination between two targets. Similarly, discrimination distinguishing targets from decoys was also performed successfully by programming the chip. The hardware tests show that the fuzzification/discrimination of this type would take less than a microsecond.

4.3. The multistage expert-rule processor (MERP) module:

The main function of a rule processor is to evaluate matches between input data and classes of knowledge (the satisfaction of certain conditions by the input) and prescribe the implications for such cases. The general structure of processing in MERP is by inference on a collection of rules of the form:

Rule 1. IF a_1 is A_{11} AND a_2 is A_{12} AND ...a_m is A_{1m} THEN y is Y_1

...

Rule n. IF a_1 is A_{n1} AND a_2 is A_{n2} AND ...a_m is A_{nm} THEN y is Y_n

where A_{ij} are fuzzy sets or their complements, i.e. if A_{lm} is a predetermined trapezoidal membership function/fuzzy set and A_{fk} is its complement then A_{fk} = NOT(A_{lm}). Consider the degree of membership/ matching a fuzzy set/class being calculated by the FSP, and thus "a is A" being replaced with u, which is the degree to which "a is A". The complement is commonly calculated either as the difference to unity, i.e. NOT(u) = 1-u, or as the maximum of all other classes except the one to be complemented, i.e. if classes covering input space are u1,u2,u3,u4 then the complement is NOT(u3) = MAX(u1,u2,u4). We built test circuitry to calculate the complement in both ways but only the second version was so far integrated within a rule-system chip. The conjunction AND is treated as the MIN operator. Thus, the antecedent "a_1 is A_{n1} AND a_2 is A_{n2} AND ...a_m is A_{nm}" can be read after fuzzification as (u_{n1} AND u_{n2} AND u_{nm}) and calculated as u_n = MIN(u_{n1}, u_{n2}, ..., u_{nm}). The collection of rules in the rules base can be read as Rule1 OR Rule 2 OR...Rulen; several rules may refer to the same conclusion/class. The logical connective OR is calculated as MAX, thus the degree of supporting an output class is the maximum of all the degrees of supporting that class coming from different rules in the rule-base.

The processing stages calculating complement, conjunction and disjunction are reflected directly in the MERP architecture presented schematically in Figure 6. Stage 1 calculates the complement by MAX operation; Stage 2 calculates the conjunction within the same rule by MIN operator; Stage 3 calculates the disjunction of all rules that refer to the same conclusion by MAX operator. The controls specify which components are selected for MIN and MAX in different rules.

The MERP module is designed as a processing module with 16 inputs with 5 membership classes each; a complement is calculated for each membership class inside the module. The module supports rules with up to 64 conjunctions; up to 128 rules can be programmed in the module and 32 decisions can be obtained as outputs. The implementation of the MERP module is performed in four development phases allowing testing of various circuits (such as analog MIN and MAX circuits) and system/integration solutions before a full-scale more expensive chip is attempted. Figure 7 shows test results from a fabricated MIN circuit (the upper waveforms are the input and the lower one is the output, which is the minimum of the two).

A smaller version of MERP (called miniMERP) with 2 inputs and 4 rules was laid out on a test chip. The chip was fabricated and tested successfully. The propagation time of a signal from inputs to output was around two microseconds. Phase 3 of development consists in integrating 8 analog inputs, 40 membership functions and 9 rules circuits on the same Fuzzy Expert System (FES) chip. The membership functions are digitally programmable trapezoids. The rules are digitally programmed to select from various membership functions for each input variable, including membership function complements. Each rule performs a conjunction amongst selected membership functions and their complements (one per variable). All analog circuitry is current-mode and the rule output currents are available in parallel on nine separate lines. The chip was fabricated and is currently under test.

4.4. Integration of ELIPS components

Efforts are ongoing for testing the synergistic operation of ELIPS components before the final cut-off design. In this sense a board is prepared to test a Hybrid Neuro Fuzzy Expert System (NFES).

4.4.1 Hybrid Neuro Fuzzy Expert System (NFES): A new test chip, termed ELIPS3, contains the second generation Membership Function Circuit (MFC) which is a voltage input/output circuit that uses current-mode processing and is digitally programmable with a generic trapezoidal shape membership function. ELIPS3 contains ten MFCs, five of which are associated with each of the two input variables. Another test chip, termed FES1, contains a similar circuit for the membership function processing but the I/V output conversion is eliminated and the current is directly passed to the rule circuits, which are part of the MERP. Current-mode rule circuits process the membership function information on the same chip before creating as output the conclusions of nine different digitally programmed rules. The rules are conjunctive (AND) and complemented or non-complemented membership function values may be used for processing. FES1 contains forty membership function circuits with five associated with each of eight input variables. Each of the nine rules may be configured to process any combination of complemented or non-complemented membership values from any of the eight input variables.

4.4.2 FANN Board: The Fuzzy-Artificial Neural Networks (FANN) test-board was designed to test the FES1 fuzzy-expert chip as well as to allow configurations of neural and fuzzy systems that combine two NN64 chips and four FES1 chips. The board also includes four analog multifunction converters capable of performing defuzzification processing and enabling a fuzzy system entirely in hardware. A photograph of the test-board is shown in Figure 8. The different system architecture configurations are achieved by setting the appropriate

jumper blocks, while the membership function shapes, rules, and neural network weights can be programmed through the computer interface. LabVIEW Full Development System 5.1 software is used to program the FANN via National Instruments ATMIO64E-3, PCI-DIO-96, and AT-AO-10 interface boards, which provide the required analog and digital I/O. The LabVIEW Fuzzy Toolbox is used to provide a high-level user interface for programming the FES1 chips, allowing the user to specify a high-level fuzzy system that then gets translated and downloaded to the fuzzy hardware on the FANN board.

The board allows 4 FES chips to be mounted on it, such that up to 36 rules can be programmed. In addition, the board incorporates the design for testing of the neural network chips, with 2 NN64 chips and a group of 16 quad - A/D chips. The board aims to play multiple roles, allowing:

- the test of the FES and NN64 chips individually,
- the test of the chips in tandem configuration, e.g. FES followed by NN64, etc.
- the test of the fusion algorithm in hardware, using the neural chips.

5. Conclusions:

Current technology allows the realization of a sensor fusion processor as a multi-chip module (MCM). A trade-off is to be made between the performance and cost of such a processor. Computational intelligence elements such as fuzzy reasoning and neural networks technology are considered fundamental for a sensor fusion chip. Several test chips implementing components of the ELIPS sensor fusion architecture have been fabricated in analog VLSI hardware and demonstrated processing times of the order of microsecond for a variety of tasks, such as target classification from preprocessed data.

6. Acknowledgment:

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by the Ballistic Missile Defense Organization through an agreement with the National Aeronautics and Space Administration. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

7. References:

1. B. Figie, et al., "Discriminating Interceptor Technology Program (DITP): Sensor Fusion for Improved Interceptor Seekers," AIAA/BMDO Missile Sciences Conference, Session 8: Ballistic Missile Defense Interceptor Technology, 1996.
2. B. V. Dasarathy and S. D. Townsend, "GIFTS – A Guide to Intelligent Technology Selection", Proc. International Conf. on Multisource-Multisensor Information Fusion, H. Arabnia and D. Zhu (Eds) Las Vegas NV, July 6-9, 1998, CSREA Press, pp. 65-72.
3. Y. Xia and J. Wang, "Recurrent Neural Networks for Shortest-Path Routing" International Conf. on Multisource-Multisensor Information Fusion, H. Arabnia and D. Zhu (Eds) Las Vegas NV, July 6-9, 1998, CSREA Press, pp. 237-244.
4. D. Zhu and B. Zhang, "Fuzzy Sensor Data Fusion in GPS Vehicle Positioning", International Conf. on Multisource-Multisensor Information Fusion, H. Arabnia and D. Zhu (Eds) Las Vegas NV, July 6.-9, 1998, CSREA Press, pp. 259-266.
5. Togai InfraLogic, Inc., "FCA Chip", http://www.ortech-engr.com/fuzzy/fcachip.html,
6. C. von Altrock, "Motorola Semiconductor and Inform Software Corp. release fuzzy logic tools for 68HC11 and 68HC12 families" http://www.fuzzytech.com/e_presmo.htm, 1997.
7. S. Guo, and L. Peters, "A High-Speed, Reconfigurable Fuzzy Logic Controller," IEEE Micro, 15: (6) 65-65, 1995.
8. H. Huertas, et al., "Integrated Circuit Implementa-tion of Fuzzy Controllers," IEEE J. Solid-State Circuits, 31: (7) 1051-1058, 1996.
9. J. Fattaruso, et al., "A Fuzzy Logic Inference Processor," IEICE Trans. Electron., E77C: (5) 727-732, 1994.
10. M. Sasaki, et al., "Current-Mode Analog Fuzzy Hardware with Voltage Input Interface and Normalization Locked Loop," IEICE Trans. Fundamentals, E75-A: (6) 650-654 June 1992
11. A. Kandel and G. Langholz, "Fuzzy Hardware: Architecture and Applications", Kluwer Academic Pub., Jan 1998.
12. S. Eberhardt,. et al, "Analog VLSI Neural Networks: Implementation Issues and Examples in Optimization and Supervised Learning," IEEE Trans. Indust. Electron. v39 (6):p. 552-564, 1992.
13. T. Daud, et al., "ELIPS: Toward a sensor fusion processor on a chip," Proc. SPIE/AeroSense Conf., vol. 3719, 1999, Orlando, FL, pp 209-219.
14. T. Duong, et al., "Learning in Neural Networks: VLSI Implementation Strategies", In Fuzzy Logic and Neural Networks Handbook, Ed: C.H. Chen, McGraw-Hill, 1996, pp. 27.1-27.48

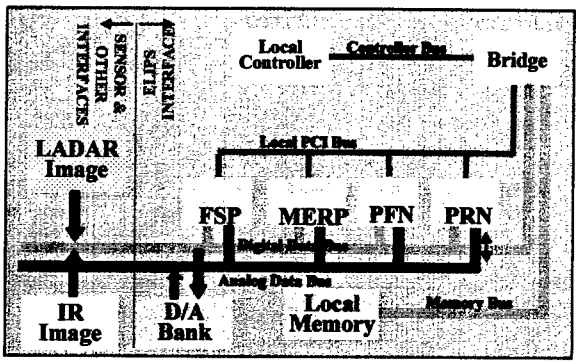


Figure 1. ELIPS architecture with modules

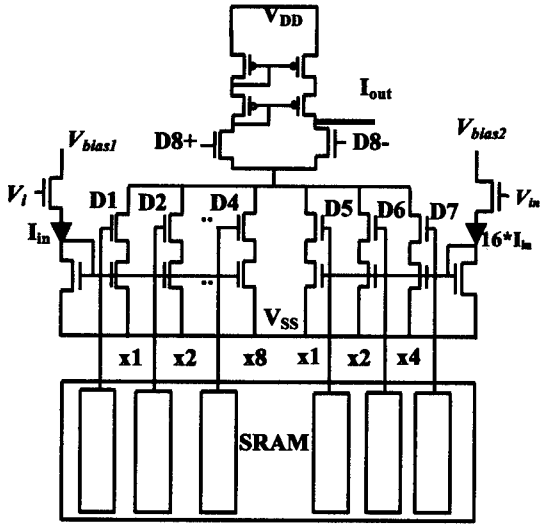


Figure 2. Circuit for the 8-bit synapse

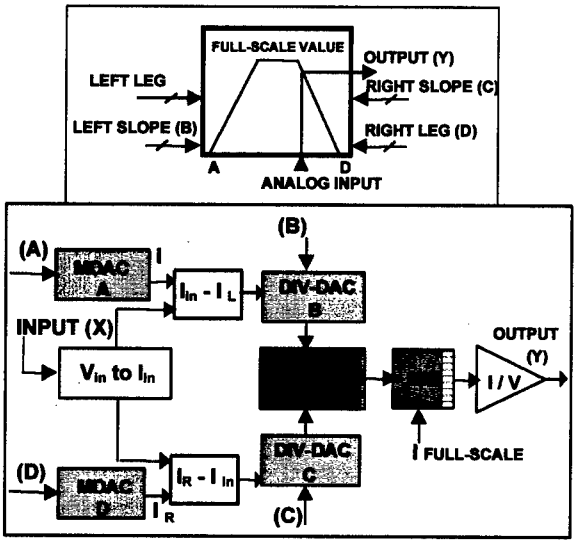


Figure 3. Block diagram of HW implementation for a MFC.

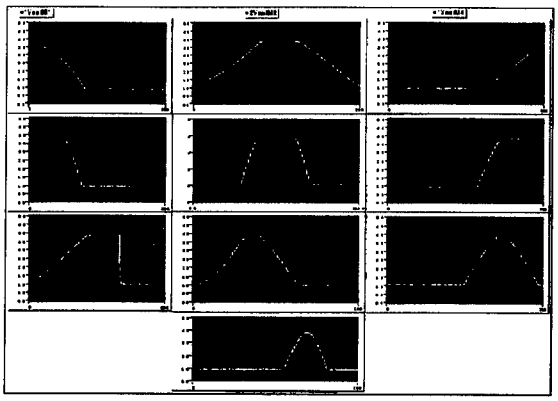


Figure 4. A variety of membership function shapes generated on the MFC test chip

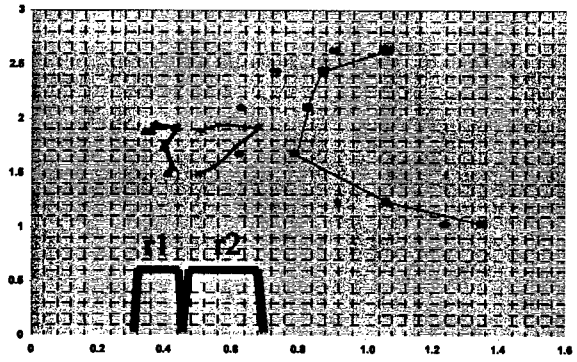


Figure 5 (a). A simulation result showing the required trapezoidal membership functions for discrimination of two targets r1 and r2.

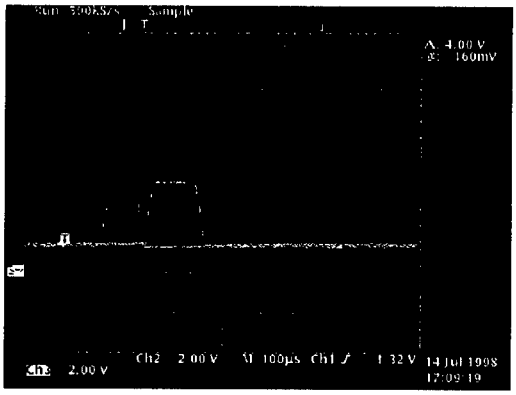


Figure 5 (b). Membership function circuit test result showing identical membership functions.

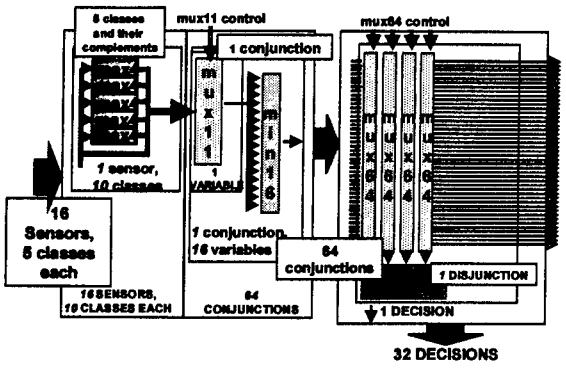


Figure 6. A schematic of the MERP architecture

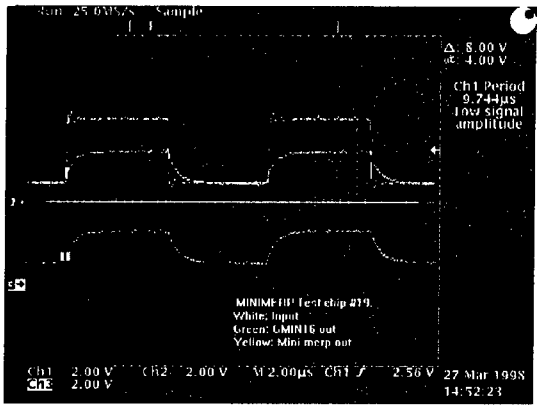


Figure 7. Propagation delay test on a miniMERP circuit. Bottom curve is the output, as the smaller of the two inputs.

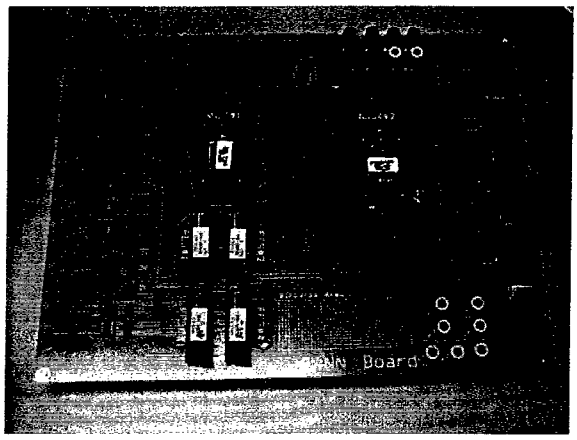


Figure 8. A photograph of the Fuzzy-Artificial Neural Networks (FANN) test-board populated with two NN64 and four FES1 chips.

High Performance Embedded Computing with Configurable Computing Machines

Prinya Atiniramit, John Davies and Peter Athanas
Virginia Tech
Department of Electrical and Computer Engineering
Blacksburg, Virginia 24061

Abstract - *Computationally speaking, sensor fusion problems can be characterized by three properties: (1) extraordinarily large I/O requirements, (2) repetitive operations on huge data sets, and (3) a large number of computational operations per point -- far beyond the capabilities of general purpose and digital signal processors. Configurable computing machines (CCMs) are emerging as a technology capable of providing high computational performance on a diversity of applications, including multidimensional signal processing, simulation acceleration, and computer graphics. High performance is achieved by rapidly reconfiguring the functionality and interconnectivity of the computing resources to match the computational requirements of specific applications. With this approach, specific application properties such as parallelism, locality, and data resolution can be exploited by creating custom operators, pipelines, and interconnection pathways. This paper illustrates these properties with an application in wireless communications.*

1 Introduction

Characteristics of the signal processing tasks associated with an advanced wireless receiver are well matched to the capabilities offered by CCM technology. Collectively, digital receiver algorithms seem to share the following properties: (a) repetitive operations are performed on huge data sets, (b) the dominant computations are conducive to very deep computational pipelines, (c) a moderate amount of latency can be tolerated, and (d) different environmental conditions require different signal processing, which in turn require distinct computational structures (time-varying computation). This paper presents an embedded solution for high-performance signal processing using configurable computing technology. Emphasis is placed on the design methodology for implementing large and intricate stream oriented signal processing tasks.

1.1 Embedded Computing in Soft Radios

The superior qualities of digital hardware over its analog counterparts, in terms of precision, stability, and flexibility, has led to the transition of communication systems from an analog implementation to a digital implementation. An extension of this trend is the software radio in which the major functions can be altered through software. A "soft" radio can not only be programmed, but can also alter the hardware.

The software radio has numerous advantages [2]. It is possible to have multimode terminals that can handle more than one standard. Traditionally, dual mode operation requires multiple sets of hardware, increasing the size and cost of the radio, an approach referred to as the "velcro radio." However, a software radio could change the modes on the same piece of hardware simply by altering the algorithms implemented on the radio. The number of discrete components are reduced since many of the traditional radio functions, like synchronization, modulation, and coding, can be integrated into one chip.

Software radios also reduce the cost associated with manufacturing and testing the radios. It is possible to precisely predict the performance of digital hardware unlike analog hardware. Furthermore, analog components frequently show a drift over time in characteristics.

Once the radio is fabricated, the time of the design cycle is reduced, since most of the existing hardware can be used. Total software reconfigurability also makes it possible to transmit upgrades to the mobile receiver *over the air*. The radio should have the capability to perform self-diagnosis, thus reducing the need for human intervention and increasing reliability.

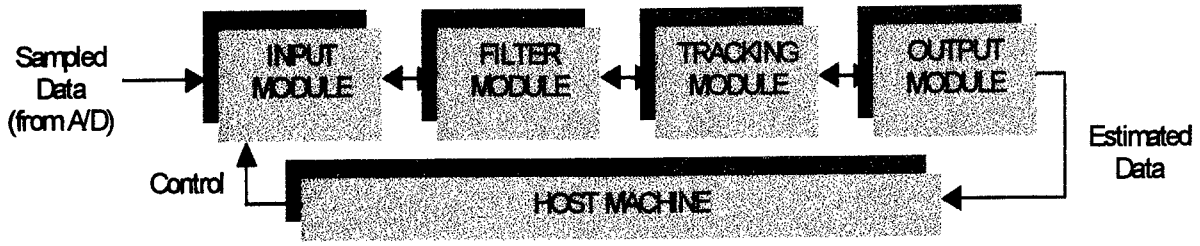


Figure 1: Adaptive receiver radio structure.

Recently software radio designers have begun to explore the use of reconfigurable computing in implementing the radios. FPGA based hardware has been used in the Speakeasy radios. The Modular Multifunction Information Transfer System (MMITS) Forum extends the concept of the Speakeasy program to build every generic communication using an open architecture [5]. The Spectrum Ware project [9] applies a software-oriented approach to wireless communication and distributed signal processing. Virtual radios are implemented that directly sample wide bands of the downconverted RF spectrum and process these samples in application software that runs on generic PCs.

1.2 Use of FPGAs in Software Radios

Software radios require complex signal processing at very high speeds. While DSPs provide the maximum flexibility and a quick design cycle, they are not efficient in terms of power consumption and system area. Multi-DSP computing platforms are quite common, but achieving inter-processor communication is often complicated, which reduces the scalability of the system when dealing with the already limited I/O bandwidth and high sample rates associated with wideband signals. ASICs, on the other hand, give the most efficient implementation of a given circuit. However, they have little flexibility, high cost, and a long design cycle.

A good design is obtained by matching the available resources to the needs of the system. FPGAs help in achieving this match while retaining flexibility in the final product. FPGAs at times can also help conserve silicon area since one chip can be configured to perform more than one function and the configurations can be changed on the fly. Situations where the use of FPGAs in digital signal processing applications is most beneficial are in systems with high sample rates, short word length, large data sets, easy pipelining and simple control requirements. FPGA based DSP designs run faster as the word width decreases since the word length on the FPGA can be set exactly to the required length. In very high order FIR filters, the algorithm can be implemented in parallel decreasing the time required for the operation. The lookup table architecture of FPGAs provides a fast and efficient way to build correlators. The property that distinguishes configurable computing from rapid prototyping is the capability of a configurable computing application to change functionality during execution, or run-time reconfiguration. Rapid reconfiguration provides the illusion of having a much larger (virtual) hardware platform. A good overview of the implementation and performance of DSP algorithms on Xilinx FPGAs, using different word lengths and different amounts of parallelism, is given in [8].

1.3 Stream Based Modular Design

The software radio prototype illustrated here is based on a concept called the *stream based* modular design process. The stream based design process provides a means to exploit the processing power attainable through deep pipelining while still maintaining some degree of flexibility. The algorithm to be implemented is first represented as a data flow graph. The data flow graph is then decomposed into smaller computational primitives called modules. Each module performs a unique subset of the overall processing on the data and passes the data and control information to the next module. An

analogy can be drawn to the assembly line process where each module performs a specific task as the component moves forward in the assembly line.

The overall architecture of the soft radio is shown in Figure 1. Once the signal is digitized to an intermediate frequency, the rest of the processing is performed on a reconfigurable platform. The bits are packed into packets ready for stream processing. Each of the reconfigurable processing modules has a similar structure and is designed to decode and act on the incoming data stream packets, as defined by the configuration of that particular module. The level of reconfiguration can vary from changing the high-level parameters of a unit with static functionality to reconfiguring the device at a primitive logic level.

A stream is comprised of both programming information and data to be processed. When a module encounters programming information, it is extracted and stored locally, and the module's parameters and operation are changed accordingly. This feature makes it possible to modify the low level parameters and functionality through high level software and also enables inter-module communication.

Valid data entering the system has a program header that provides program flow information, i.e., information about the operations to be performed on the data, and about how the data is linked together. The header of a stream packet also contains information indicating whether the bits are valid and whether the packet contains data or program information. When the system is idle or if there are no users in the system, then the valid bit in the header is not set.

Running on each module are three sets of pipelines and a state machine, which interprets how the packet is channeled through the pipelines as shown in Figure 3. If the valid bit and the program bit are set, the information is sent to the configuration pipeline to configure the module for the following data. The modules maintain the configuration to which they are set and act accordingly on valid data until the configuration is changed. The bypass pipeline is present to ensure that data is not corrupted by the module. It is important that each of these pipelines have the same amount of delay. At the end of the pipeline, the stream packet is reconstructed with the updated header and routed to the next module.

2 CCM-based Soft Radio

The radio presented here is intended for direct sequence CDMA systems. Adaptive algorithms including variants of clipped LMS (Least Mean Square) algorithms have been selected for the equalizer module. The approach adopted is to divide the system into several sub-modules, each of which performs a specific and well-defined task. The block diagram of the prototype software radio being is shown in Figure 1, where each of the functions are entirely configurable. The first module is the *INPUT* module, which is responsible for buffering sampled data, constructing packets, and controlling the system. This module can receive samples from the receiver front-end, or the control signals from the host PC.

When data is received, the module constructs data packets and sends them to the

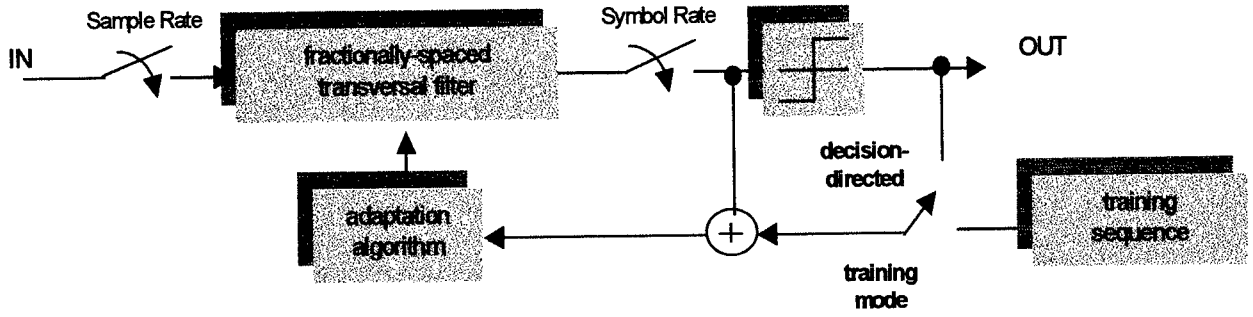


Figure 2: Complex-weight fractionally-spaced linear adaptive receiver.

next, *FILTER*, module, which performs the adaptive filter operation according to the type of packets received.

The third module, *TRACKING* module, is responsible for timing-recovery. After the timing is recovered, the module sends data back to the *INPUT* module to properly control the coefficient adaptation operation.

The *OUTPUT* module, the last module, buffers the estimated data and sends this data to the host PC.

This paper focuses on the signal processing aspects of the radio application, which are primarily embodied in the *FILTER* and *TRACKING* modules. The remainder of this section focuses on these two modules.

2.1 Adaptive LMS Equalizer Receiver

The modified model, which has lower complexity, seems to give us more favor. The structure of the new model is shown in Figure 2.

The received signal is converted to baseband and sampled at the front-end of the receiver. This sampled received signal, r(n), is then passed to the fractionally-spaced transversal filter.

The length of this filter has to be enough to keep one full symbol of signal. That is, if the process gain is N, and there are p samples per chip, the length of the filter has to be more than Np.

The filter operation is computed as,

$$\tilde{y}(n) = \tilde{w}^T(n)\tilde{r}(n)$$

where the received signal vector is,

$$\tilde{r}(n) = \begin{bmatrix} \mathrm{Re}\{r_0(n) & r_1(n) & \ldots & r_{Np}(n)\} \\ \mathrm{Im}\{r_0(n) & r_1(n) & \ldots & r_{Np}(n)\} \end{bmatrix}$$

and the coefficient vector is,

$$\tilde{w}(n) = \begin{bmatrix} \mathrm{Re}\{w_0(n) & w_1(n) & \ldots & w_{Np}(n)\} \\ \mathrm{Im}\{w_0(n) & w_1(n) & \ldots & w_{Np}(n)\} \end{bmatrix}.$$

The output, $\tilde{y}(n)$, is then passed to the decision device. The estimated data for the n^{th} symbol is equal to $\hat{d}(n) = \mathrm{sign}\{\mathrm{Re}[\tilde{y}(n)]\}$.

The filter weights adapt to the properties of the communications channel. The coefficients are updated every symbols to minimize the difference (error) between the output of the filter and the reference signal, d(n), or as

$$e(n) = d(n) - \tilde{y}(n).$$

This mode of operation is called training mode. The other mode, decision-directed mode, perform a similar task except that in this mode the error is computed from the estimated data, $\hat{d}(n)$, instead of the reference signal. The coefficient adaptation algorithm is summarized simply by the relationship,

$$\tilde{w}(n+1) = \tilde{w}(n) + \upsilon e(n)\tilde{r}(n),$$

where μ is a step-size, which define how fast the coefficients can adapt.

2.1.1 System-Level Architecture

There are some parameters of the adaptive filter that can be changed in real-time to improve the overall performance. This implies incorporating some form of flexibility of the system. In this example, it is done using a stream-based approach. In this system, there are three types of packets; data, control, or configuration packets.

The data packets contain the samples of the received signal (from the A/D converter), which will undergo normal filter operations.

The control packets govern specific operations of the filter. From the previous section the filter operation consists of two basic tasks. The first task is *filter output calculation*, which is common for every type of digital filters. The second task is the *coefficient adaptation operation*, which is an operation that applies only for adaptive filters, and happens every symbol period, while the first operation occur every sample. Since the filter consists of two operations, we define the more-often operation, the filter output operation, as a default operation. This makes the second one, coefficient adaptation, a special operation – whenever the filter receives data packet, it performs the filter output calculation, the default operation. However, if the control packet is

received, the system switches to coefficient adaptation operation, the special operation.

Note that the implemented filter calculates the output every sample even though it is theoretically enough to do so once every symbol period. This is because it is assumed that the system is asynchronous, and a tracking module is needed to perform timing-recovery. This tracking system, in turn, needs input, which is the output of filter, every sample.

The last type of packet, the configuration packet, defines the parameters, e.g. filter-length and step-size, of the filter.

The packet format adopted in the adaptive radio is shown in Figure 3. The packet contains four fields: VALID, TYPE, ADDRESS, and PAYLOAD. These fields are defined as follows:

- VALID field (1 bit): defines the validity of this packet when it is set to '1'.
- TYPE field (1 to 4 bits): defines the type of packets, data, control or configuration packets.
- ADDRESS field (4 bits): Identifies the module to be configured. Applies only to configuration packets.
- The PAYLOAD field (10 bits): contains samples of the received signal for data packet, or configuration information for configuration packet. The PAYLOAD field does not apply for the control packets.

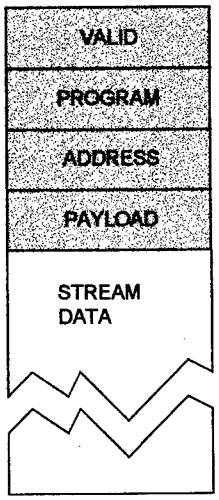


Figure 3: Stream format for programming signal processing modules.

The implemented system is built on an FPGA-based platform. The platform is accessible via PCI bus and controlled through the API functions provided by the vendor. There are 32 XC4028, XILINX 4000 series, on the platform.

2.2 Acquisition and Tracking

The Acquisition and Tracking (A/T) module performs spread spectrum symbol synchronization; as the module's name suggests, it does so using two distinct functions. Acquisition detects the presence of a user's signal and assesses this signal's initial code phase. Due to such phenomena as clock drift between transmitter and receiver, this code phase may not be constant; tracking updates the assessment of the signal's slowly changing code phase. Prior to user detection, only the acquisition function operates. However, when the module detects a user the acquisition sub-module continues to process the incoming data; it operates in parallel with the tracking sub-module to validate the user's continued presence — should the user disappear, tracking halts and the acquisition sub-module restarts, waiting for the user's reemergence.

The A/T module takes the full output of the LMS equalizer as its input; this data will peak when user data is in phase with the despreading code. Also, the A/T module is notified when the LMS equalizer calculates its error and updates its weights—i.e. when the Input Module believes the user is in phase. The A/T module uses this information as a reference for its phase corrections; it returns a signal indicating the relative offset of the Input Module's code phase estimate with the actual code phase as calculated by the A/T module. The A/T module provides the despread user data as its output.

The acquisition sub-module encompasses two functions: user detection and estimation of the user's initial code phase. For user detection, the module supports two configurable algorithms: a signal magnitude threshold and a maximum search for a persistent peak. To detect a user through thresholding, the module sums the absolute value of the incoming data over one symbol period. The module compares this sum to a threshold: if the signal is purely noise, the sum will be below the threshold; if a user is present, the power of the added signal will produce a higher sum, exceeding the threshold. This user detection scheme is very simple to implement, but setting the level of the threshold is problematic. In a fading channel, the sum of one symbol period

could drop below the threshold even when a user is present. Similarly, in a high noise environment, a threshold that is too low could result in false positives—indicating the presence of a user when there is none. These problems can be addressed through the use of a maximum search for a persistent peak, as this technique does not require a threshold. This technique searches over one symbol period for largest magnitude peak; if this peak occurs consistently at the same phase for several symbol periods, a user is present, while irregular maximum peak locations indicate the user's absence. Although this technique is immune from the problems of thresholding, it assumes that the largest magnitude peak will always correspond to the code phase; depending on channel noise characteristics, this may not always be the case. A peak would always occur at the user's phase, but higher peaks could sporadically occur at other phases.

In order to address the shortcomings of these techniques, both described methods are available in the A/T module as crucibles of user presence. The persistent peak test is relaxed such that a small number of nonmaximum peaks at the current phase location do not invalidate the user's position. Such a persistent peak test in combination with a low threshold will permit very few detection errors. Alternately, either method can be disabled to adjust as channel conditions dictate.

The second function of acquisition—the initial estimation of the user's phase—extends naturally from the use of a maximum search for a persistent peak. To assess peak persistence, the algorithm must store the location of this peak. If the algorithm detects a peak, this location is considered the initial code phase. The A/T module subsequently notifies the Input Module of the appropriate phase. If the acquisition sub-module's configuration only considers a threshold for user detection, the module still needs to search for a maximum peak to determine the initial code phase—even though this data does not affect user detection.

Once the acquisition sub-module detects a user and assesses its initial code phase, the tracking sub-module initiates operation. The drift of the user's code phase should be quite slow compared to the symbol rate; therefore, the phase should change by no more than one chip per symbol following acquisition. Unlike the acquisition sub-module, the tracking module does not need to search the entire symbol period for the maximum. Rather, it only needs to consider the samples immediately preceding and following the current code phase. These are the early and late samples, respectively; the attention paid to these symbols dubs this tracking algorithm *early-late tracking*. Should the peak occur at the early sample, the tracking algorithm indicates to the Input Module that the LMS Equalizer's calculate error/update weights should be advanced by one sample; similarly, a peak at the late sample indicates that the update should be delayed by one sample.

In order to compensate for phase drift, a modification to the acquisition algorithm is necessary: when assessing the most persistent peak, it should consider a peak persistent even if it drifts by one sample per symbol. Otherwise, the peak will be properly tracked, but the acquisition algorithm will falsely indicate that the user is no longer transmitting because of the peak drift. Should the tracking sub-module detect a phase drift, it will update the acquisition module's estimate of the peak's location.

The implementation of the A/T module is straightforward. For the acquisition sub-module, a register maintains a sum of the samples of the present symbol for threshholding. An additional pair of registers maintains the value and location of the maximum value in the present symbol; the location is compared to the location of previous maximum, and if the peak is persistent, a counter is incremented to assess its persistence level. For the tracking sub-module, the magnitudes at the current codes phases, as well as the early and late magnitudes, are registered and compared to each other to indicate any present symbol drift.

3 Conclusions and Future Research

The soft radio architecture can be implemented with various reconfigurable device technologies. Virginia Tech has fabricated a customized FPGA-like devices called Colt and Stallion, which are suited for the signal processing tasks of software radios [7]. The Colt/Stallion is an experimental run-time reconfigurable device and

utilizes self-directing streams that allocate and configure device resources to accomplish a given task. The streams contain a program header that contains the necessary configuration information for each resource required. The Colt/Stallion FPGA is capable of reconfiguring all or portions of the chip in a fraction of a microsecond. This will allow for more effective computation per area of silicon. The prototype designs on the FPGA testbed will be migrated in parts to the Colt/Stallion processors.

References

[1] Joe Mitola, "The software radio architecture," *IEEE Communication Magazine*, pp. 26-38, May 1995.

[2] Peter M Athanas, J. H. Reed and W. H. Tranter, "A prototype software radio based on configurable computing," *Advancing Microelectronics*, pp. 34-39, 1998.

[3] R. Michael Buehrer, *The application of multiuser detection to cellular CDMA*, Ph.D. Dissertation, Virginia Polytechnic Institute and State University, June 1996.

[4] John J. Proakis, *Digital communications*, McGraw-Hill, 3rd edition, 1996.

[5] Andy Ivers and Dave Smith, "A practical approach to the implementation of multiple radio connfigurations utilizing reconfigurable hardware and software building blocks," *Proceedings MILCOM*, vol. 3, pp. 1327-1332, 1997.

[6] Steve Swanchara, "An FPGA based multiuser receiver employing parallel interference cancellation," Master's Thesis, Aug 1998.

[7] R. Bittner and P. Athanas, "Wormhole Run-time reconfiguration," *ACM/SIGDA International Symposium on FPGAs*, Monterey, CA, pp. 79-85, Feb. 1997.

[8] Chris Dick, "High performance communications using FPGAs," *International Symposium on Advanced Radio Technologies*, Chapter 11, Sept. 1998.

[9] Spectrum ware home page, http://www.tns.lcs.mit.edu/SpectrumWare/home.html

[10] M. Valenti and B. D. Woerner, "Performance of Turbo Codes in Interleaved Flat Fading Channels with Estimated Channel State Information," *Vehicular Technology Conference*, Ottawa, Ontario, Canada, May 18-21, pp. 66-70, 1998.

[11] William H. Tranter and Kurt L. Kosbar, "Simulation of communication systems," *IEEE Communications Magazine*, Vol. 32, No. 7, pp. 26-35, July 1994.

Wavelet Neuron Filter with the Local Statistics Oriented to the Pre-processor for the Image Signals

Noriaki Suetake Naoki Yamauchi * Takeshi Yamakawa †

Department of Control Engineering and Science
Kyushu Institute of Technology
Iizuka, Fukuoka, 820-8502 Japan
{suetake, yuchi}@tsuge.ces.kyutech.ac.jp, yamakawa@ces.kyutech.ac.jp

Abstract *We propose a novel nonlinear filter which is based on a framework of a linear FIR filter and the wavelet neuron (WN) model, and employs the local statistics such as a variance of signal levels in the filter window. The proposed filter is synthesized by a learning method which guarantees optimal design caused by employing the WN model. The proposed filter is effective for the various applications such as the noise elimination, sharpening and so on, because their functions are determined by the pairs of target and input signals in the learning. The effectiveness and validity of the proposed filter is verified by applying it to the preprocessing of the image signals.*

Keywords: wavelet neuron, nonlinear filter, linear FIR filter, noise elimination, sharpening, image signal preprocessing.

1 Introduction

Linear filters have been the primary tools for signal processing. They are easy to design, and they offer excellent performance in many cases. This is particularly the case for spectral separation where the desired signal spectrum is significantly different from that of the interference. In many situations, however, it is necessary to process signals with sharp edges and thus wide spectrums. Unfortunately, linear smoothing filters also smooth signal edges. Furthermore, a linear filter can not totally eliminate impulse noise. A median filtering has been recognized as an effective alternative to the linear smoother in some applications[1]. In particular, the moving median of a time or spatial series has been shown to preserve edges or monotonic changes in trend, while eliminating impulse noise. A median filter is included in the class of order statistic (OS) filters, as well as a α-trimmed filter and a midrange filter and so on[2][3]. These filters achieve noise elimination well by using information of the noise distribution, although they are effective only for specific types of noise. However, OS filters have a disadvantage with respect to the preservation of the signals, because they lose information of signal patterns by sorting signal levels in the filter window. For the realization of both of restoration of the signal and noise elimination, both of pattern and statistical information in the filter window should be reflected to a design of a filter.

From this point of the view, we propose a novel nonlinear filter named the local statistics employed wavelet neuron (LSWN) filter. The LSWN filter is based on the WN model[4] and a framework of a linear finite impulse response (FIR) filter, and employs the local statistics in the filter window. The LSWN filter achieves both of the elimination of noise and high restoration of the signal simultaneously. Moreover, it is effective not only for the

*The author is presently working at Kyusyu Matsushita Electric Co,. Ltd., Minoshima, Hakata-ku, 812-0017, Fukuoka, Japan.

†The author is also with the Fuzzy Logic System Institute (FLSI), Kawatsu, Iizuka, 820-0067, Japan.

elimination of noise but also other functions such as sharpening of the image. In this paper, we verify the effectiveness and the validity of the LSWN filter by applying it to the preprocessing of the image signals.

2 The Local Statistics Employed Wavelet Neuron Filter (LSWN)

In this section, the LSWN filter is discussed. Consider the following observation mechanism:

$$y_k = x_k + v_k, \tag{1}$$

where y_k is a noisy observation, x_k an original signal, and v_k an observation noise of an arbitrary distribution type.

2.1 Extension of a FIR Linear Filter by the WN Model

Here, we extend a linear FIR filter by applying the WN model to its framework. This extension is referred to as the WN filter. The WN model has been proposed by one of the authors in 1994[4]. This WN the synaptic characteristics of which is nonlinear and represented by a set of wavelets and the weight corresponding to each wavelet has high ability of generalization with a guarantee of the global minimum.

Fig.1 shows the structures of the WN filter and the wavelet synapse (WS) model. The output of the WN filter of length N operating on a sequence $\{y_k\}$ for N odd is given by:

$$\hat{x}_{wn_k} = \sum_{i=1}^{N} f_i(y_k(i)) \tag{2}$$

with

$$f_i(y_k(i)) = \sum_{a=0}^{p}\sum_{b=0}^{a} \Psi_{a,b}(y_k(i)) w_{i_{a,b}}, \tag{3}$$

where $(\cdot)$ represents time/space sequence in the filer window. $y_k(1), \cdots, y_k(N)$ correspond to $y_{k-M}, \cdots, y_{k+M}$, respectively, and $M = (N-1)/2$. $\Psi_{a,b}(u)$ are wavelets that are generated by the mother wavelet $\Psi(u)$ as follows:

$$\Psi_{a,b}(u) = \begin{cases} 1 & a = 0 \text{ and } b = 0, \\ \Psi(au - b) & \text{otherwise.} \end{cases} \tag{4}$$

where a is a scaling parameter, and b is a shifting parameter. And this WS employs the the compactly supported wavelet shown in Fig.2 as the mother wavelet. It is represented by the following equation:

$$\Psi(u) = \begin{cases} \cos \pi u & -0.5 \le u \le 0.5, \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

The wavelet distribution is illustrated in Fig.3, where the level p stands for a reciprocal scaling parameter.

The learning of the weights $w_{i_{a,b}}$ is achieved so that the following error function $E_{wn}(\hat{x}_{wn_k}, t_k)$ becomes minimum:

$$\begin{aligned} &E_{wn}(\hat{x}_{wn_k}, t_k) \\ &= \frac{1}{2}\sum_{k=1}^{K}(\hat{x}_{wn_k} - t_k)^2 \\ &= \frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{a=0}^{p}\sum_{b=0}^{a}(\Psi_{a,b}(y_k(i)) \cdot w_{i_{a,b}} - t_k)^2, \end{aligned} \tag{6}$$

where t_k is a target signal and K is the length of target signal. The gradient descent method is employed here. Eq.(6) is unimodal function, because it is parabolic with respect to weights $w_{i_{a,b}}$, that is, this WN filter guarantees the global minimum[4].

After the learning is completed, the set of weights of the WN filter is optimized where the filtering output is as close to the ideal signal as possible.

2.2 LSWN Filter

The proposed LSWN filter is shown in Fig.4, where the outputs from the WN filter and the local statistics calculator are fed to the Input-Correlated WS model. In this case, the output of the filter is obtained as an output of the Input-Correlated WS model. Fig.5 shows the structure of the Input-Correlated WS model.

In the proposed filter, the estimation of an original signal $\hat{x}_{lswn_k}$ is given by:

$$\hat{x}_{lswn_k} = f(\hat{x}_{wn_k}, \sigma_k^2) \quad (7)$$

with

$$f(\hat{x}_{wn_k}, \sigma_k^2) = \sum_{c=0}^{q}\sum_{d=0}^{r} \Psi_{c,d}(\hat{x}_{wn_k}, \sigma_k^2) w_{c,d}, \quad (8)$$

where $\hat{x}_{wn_k}$ is given by Eq.(2). σ_k^2 is a variance of the signal levels in the filter window and calculated by following equation:

$$\sigma_k^2 = \frac{1}{N}\sum_{i}^{N}(y_k(i) - \frac{1}{N}\sum_{j}^{N} y_k(j))^2. \quad (9)$$

σ_k^2 is employed as the information of the local statistics in the LSWN filter. $\Psi_{c,d}(u_1, u_2)$ are wavelets in two dimensional space (u_1, u_2) that are generated by the mother wavelet $\Psi(u_1, u_2)$ shown in Fig.6. It is a compactly supported wavelet and represented by the following equation:

$$\Psi(u_1, u_2) = \begin{cases} \cos \pi u_1 \cos \pi u_2 & -0.5 \le u_1, u_2 \le 0.5, \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

The wavelet distribution is illustrated in Fig.7, where (a)level 0, (b)level 1 and (c)level 2 are shown for examples. In Eq.(8), the level q stands for a reciprocal scaling parameter and $r = (q+1)^2$ a shifting parameter in two dimensional space (u_1, u_2).

The learning of the LSWN filter is achieved so that each of error function of WN filter $E_{wn_k}(\hat{x}_{wn_k}, t_k)$ (Eq.6) and the following error function of Input-Correlated WS $E_{lswn}(\hat{x}_{lswn_k}, t_k)$ becomes minimum simultaneously:

$$E_{lswn}(\hat{x}_{lswn_k}, t_k) = \frac{1}{2}\sum_{k=1}^{K}(\hat{x}_{lswn_k} - t_k)^2. \quad (11)$$

The same target signal t_k is employed for both learning processes of the WN filter and Input-Correlated WS here. From Eq.11, it is also easily understood that the global minimum is reached in the learning of the Input-Correlated WS.

3 Experimental Results

The attempt is made to verify the effectiveness and the validity of the LSWN filter by applying it to the noise elimination and the sharpening of the images. In the experiments, the signals in the filter window are numbered and fed to a filter as shown in Fig.8. Here, the filter window size N is 25 (=5×5 pixels). All of p, q and r are 12.

3.1 Noise elimination

Here, the LSWN filter is applied to the noise elimination for the images of machine printed capital characters and human faces.

3.1.1 Machine printed capital characters

Learning process In this experiment, we employed the images of machine printed capital character 'F' shown in Fig.9(a) and (b), which are constructed with 50×50 pixels and the resolution is 8 bits/pixel gray-level, as a target image and the input image for the learning process of the LSWN filter, respectively. The input image shown in Fig.9(b) is the target image corrupted by both of a Gaussian noise N(0,200) and an impulsive noise (5%), the elimination of which is very difficult for the conventional filters.

Testing process After the learning has been completed, the performance of the LSWN filter is tested for the images of machine printed capital characters. For example, we show the results of noise elimination for the images of machine printed capital character 'E' (50×50 pixels, 8 bits/pixel gray-level). Fig.9(c) shows an original image 'E'. Fig.9(d) shows the input image which is the original image corrupted by both of a Gaussian noise N(0,200) and an impulsive noise (5%), similar to Fig.9(b). The root mean square error (RMSE) of the input image shown in Fig.9(d) is 39.21. The RMSE is calculated by the following equation:

$$RMSE = \frac{1}{P \times Q}\sqrt{\sum_{i=1}^{P}\sum_{j=1}^{Q}(y(i,j) - x(i,j))^2}, \quad (12)$$

where $P \times Q$ is an image size, $y(i,j)$ the input image and $x(i,j)$ the original image. Fig.10(a) shows the result of filtering by the LSWN filter. The RMSE of the LSWN filter is 1.23. For comparison, an optimized linear FIR filter, an optimized OS filter and the WN filter are examined and the results of filtering by them are shown in Fig.10(b),(c) and (d), respectively. The RMSE of an optimized linear FIR, an OS filter and the WN filter are 11.22, 5.47 and 4.53, respectively. The superiority of the LSWN filter to other filters is easily understood.

3.1.2 Human facial image

In order to verify the effectiveness of the LSWN for the images which include complicated signal patterns, we employ the facial images in this experiment. Here, we employed the images (120×160 pixels, 8bits/pixel gray-level) shown in Fig.11(a) and (b), as a common target image and the input image for the learning process of the LSWN filter. The input image shown in Fig.11(b) is the target image corrupted by both of a Gaussian noise N(0,100) and an impulsive noise (1%).

After each learning has been completed, the performance of the proposed filters is tested for the facial image. We show the results of noise elimination for the facial image. Fig.11(c) shows an original facial image (120×160, 8 bits/pixel gray-level). Fig.11(d) shows the input image which is the original image corrupted by both of a Gaussian noise N(0,100) and an impulsive noise (1%), similar to Fig.11(b). The RMSE of the input image shown in Fig.11(d) is 14.18. Fig.12(a) shows the result of filtering by the LSWN filter. The RMSE of the LSWN filter is 6.03. For comparison, an optimized linear filter, an optimized OS filter and the WN filter are examined and the results of filtering by them are shown in Fig.12(b), (c) and (d), respectively. The RMSE of the an optimized linear FIR filter, an optimized OS filter and the WN filter are 7.23, 6.92 and 6.67, respectively. The result of the LSWN filter is superior to other filters.

3.2 Sharpening of images

In this experiment, we employed the images of machine printed capital character 'F' shown Fig.13(a) and (b) (50×50 pixels, 8 bits/pixel gray-level), as a target image and the input image in the learning process of the LSWN filter. The input image shown in Fig.13(b) is the target image corrupted by both of a Gaussian noise N(0,200) and an impulsive noise (5%) after smoothing by a mean filter, widow size of which is 25 (5×5).

After the learning has been completed, the performance of the LSWN filter is tested for the machine printed capital characters. For example, the results of the sharpening for the capital character image 'E' (50×50 pixels, 8bit/pixel gray-level) are shown here. Fig.13(c) and (d) show an original image and an input image which is the original image corrupted by both of a Gaussian noise N(0,200) and an impulsive noise (5%) after smoothing it by a mean filter, window size of which is 25 (5×5) pixels, similar to Fig.13(b). The RMSE of the input image shown in Fig.11(d) is 40.18. The results of sharpening by the LSWN filter is shown in Fig.14(a) where the RMSE of the LSWN filter is 6.59. For comparison, the WN filter are examined and the results of sharpening by it are shown in Fig.14(b). The RMSE of the WN filter is 8.48. From Figs.14(a), it is clear that the sharpening of the image is achieved well. The conventional filters are much less effective than the WN filter and the LSWN filter, because they don't have high mapping ability between input and output, like the WN model.

4 Conclusions

In this paper, we propose a novel nonlinear filter which is based on a framework of a linear FIR filter and the wavelet neuron (WN) model, and employs the local statistics such as a variance of signal levels in the filter window. The LSWN filter is optimally designed and implemented by learning which guarantees convergence to the global minimum.

Through the experiments of the noise elimination and the sharpening of images, we confirmed that the LSWN filter significantly achieves both of high noise elimination and high restoration of the signal, simultaneously.

One of main characteristics of the proposed filter is that it is applicable to arbitrary image signal preprocessing. Many of traditional filters are confined to specific use. On the other hand, our filter proposed here is effective not only for noise elimination but also for sharpening and other various applications. This feature of the proposed filer is derived from that its function is determined by the pairs of target and input signals in the training. If we prepare a typical training set of images for some practical purposes, we can tune the filter to be suitable for the purposes.

Furthermore, the proposed filter does not require a complicated algorithm, and it architecture is very simple. It has highly potential applications to a wide range of practical signal processing.

References

[1] J. W. Turky, "Nonlinear (nonsuperposable) Methods for Smoothing Data," Congr. Rec. 1974 EASCON, p.673, 1974.

[2] C. Alan Bovik, Thomas S. Hung and David C. Munson, "A Generalization of Median Filtering Using Linear Combinations of Order Statistics," IEEE Trans. Acoust. Speech, and Signal Process., Vol.ASSP-31, No.6, pp.1342-1350, 1983.

[3] Digital Signal Processing Handbook (in Japanese), Edited by IEICE, Ohm Inc., 19 93.

[4] T. Yamakawa, E. Uchino and T. Samatsu, "Wavelet Neural Networks Employing Over-Complete Number of Compactly Supported Non-Orthogonal Wavelets and Their Applications," Proc. of the 1994 Int. Conf. on Neural Networks, pp.1391-1396, 1994.

[5] T. Yamakawa, Japanese Patent Application, No.TOKU-GAN-HEI 11-39641, 1999.

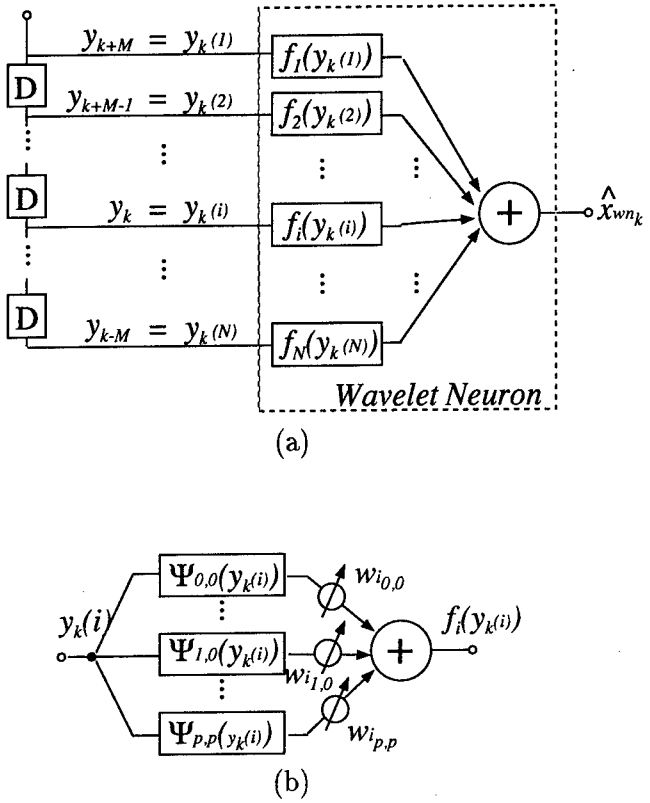


Fig. 1 The structures of the WN filter and the WS model. (a)The structure of WN filter. (b)The Structure of the WS model.

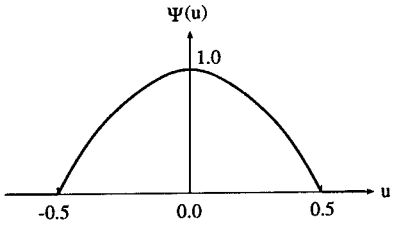


Fig. 2 The shape of a mother wavelet.

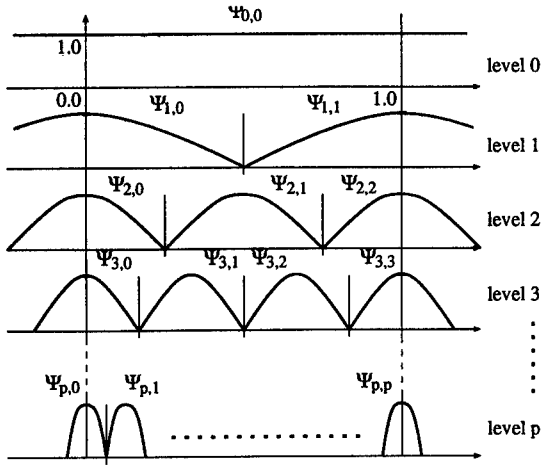


Fig. 3 The shape of a mother wavelet.

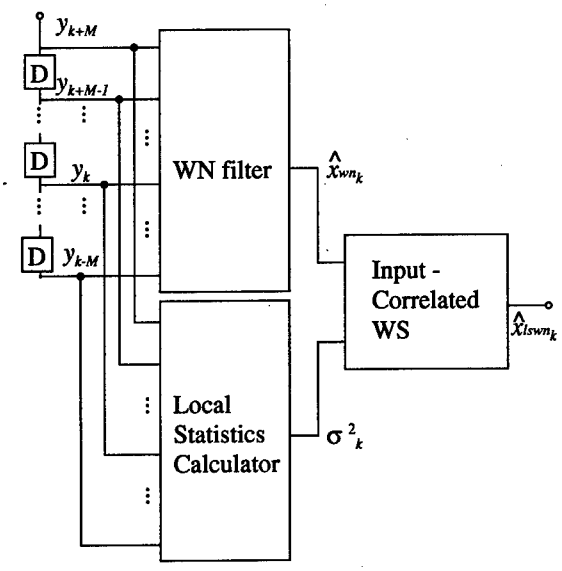


Fig. 4 The structure of the LSWN filter.

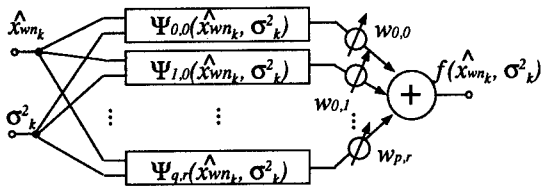


Fig. 5 The structure of the Input-Correlated WS model.

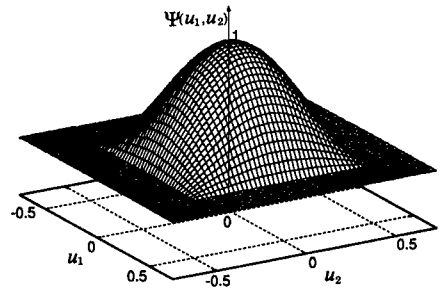


Fig. 6 The shape of a two-dimensional mother wavelet.

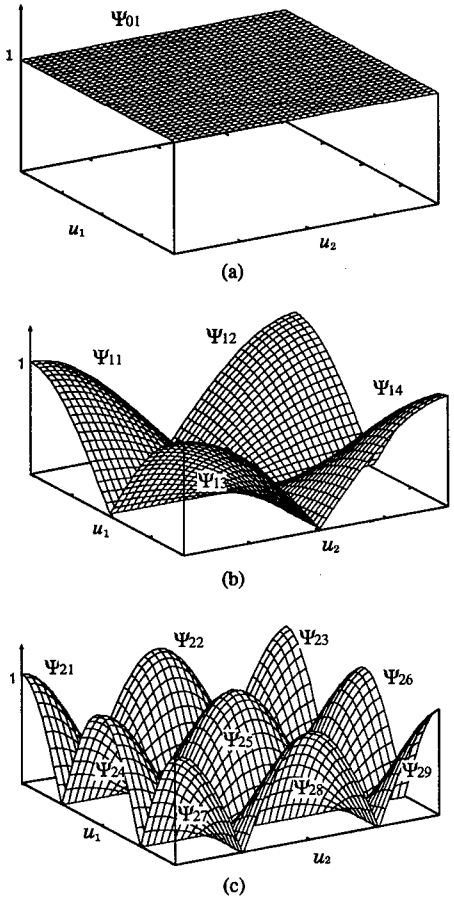


Fig. 7 The arrangement of two-dimensional wavelets. (a)Level 0. (b)Level 1. (c)Level 2

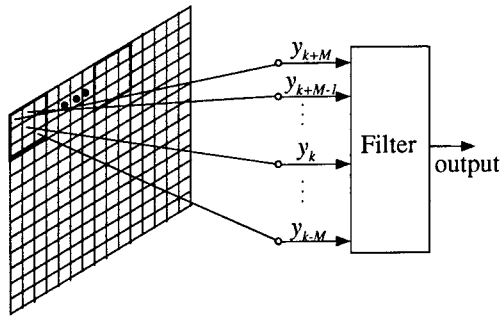


Fig. 8 How to input the image data to the filters.

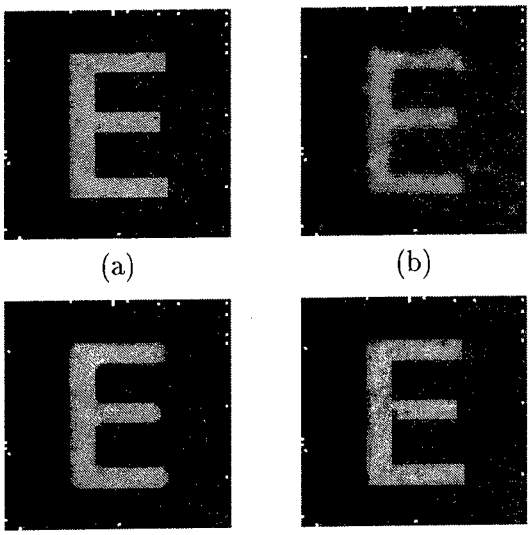


Fig. 10 The results of the filtering for the machine printed character 'E'. (a) The LSWN filter. (b) An optimized linear FIR filter. (c)An optimized OS filter. (d)The WN filter.

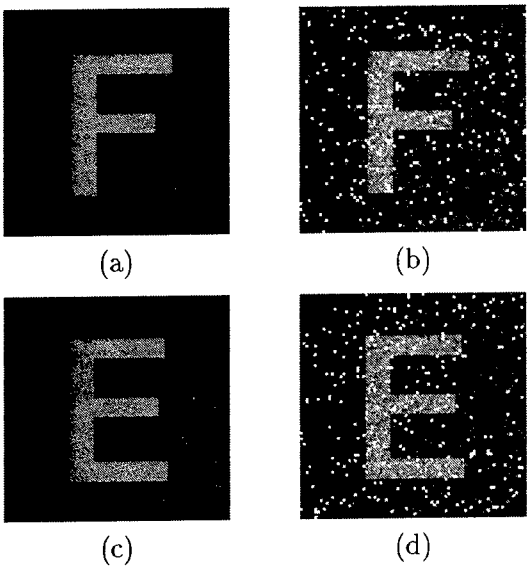


Fig. 9 The images employed in the filtering of machine printed capital characters. (a) A target image in the learning. (b)An input image in the learning. (c) An original image in the filtering. (d)An input image in the filtering.

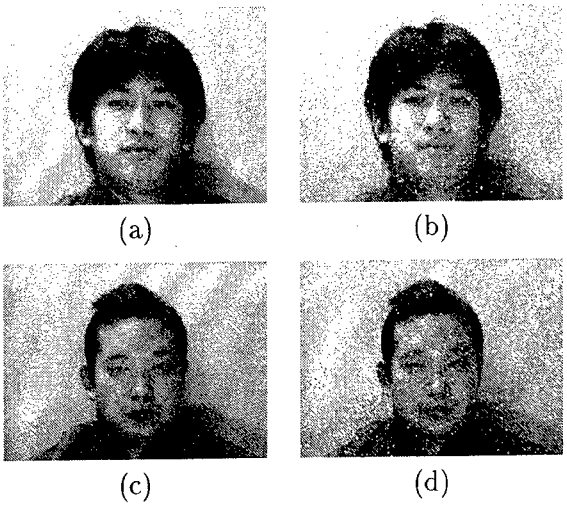


Fig. 11 The images employed in the filtering of the the facial image. (a) A target image in the learning. (b)An input image in the learning. (c) An original image in the filtering. (d)An input image in the filtering.

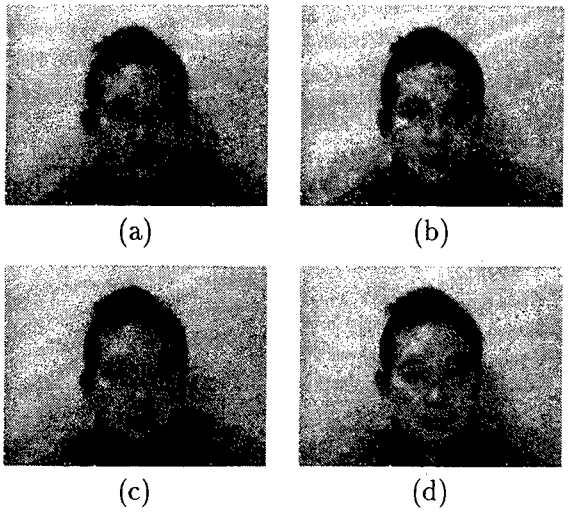


Fig. 12 The filtering results for the facial image. (a) The LSWN filter. (b) An optimized linear FIR filter. (c) An optimized OS filter. (d) The WN filter.

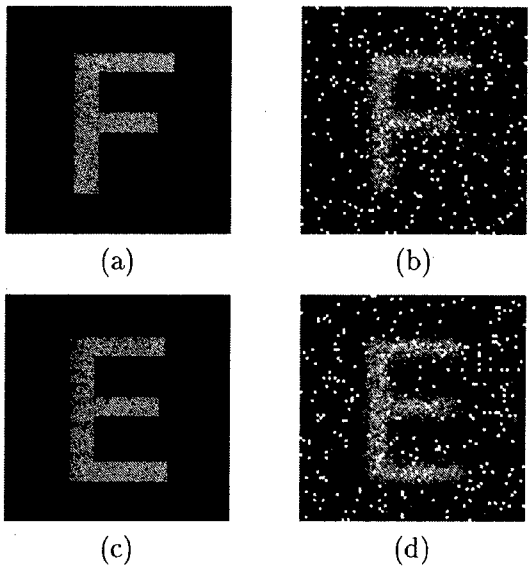


Fig. 13 The images employed in the sharpening of machine printed capital characters. (a) A target image in the learning. (b) An input image in the learning. (c) An original image in the sharpening. (d)An input image in the sharpening.

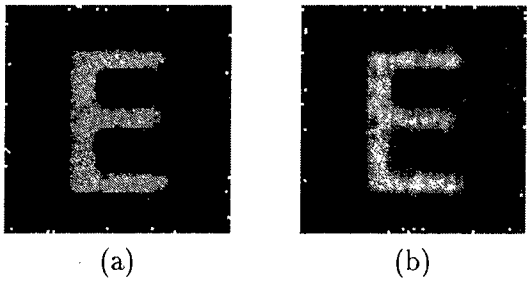


Fig. 14 The results of the sharpening for the machine printed character 'E'. (a) The LSWN filter. (b)An optimized linear FIR filter. (c) An optimized OS filter. (d) The WN filter.

Reconfigurable Architecures and Systems for Real-time Low-level Vision

A. Benedetti, P. Perona

California Institute of Technology
Division of Engineering and Applied Science
Pasadena, CA 91125
{arrigo,perona}@vision.caltech.edu

Abstract ***A novel system architecture that exploits the spatial locality in memory access that is found in most low-level vision algorithms is presented. A real-time feature selection system is used to exemplify the underlying ideas, and an implementation based on commercially available Field Programmable Gate Arrays (FPGA's) and synchronous SRAM memory devices is proposed. The peak memory access rate of a system based on this architecture is estimated at 2.88 G-Bytes/s, which represents a four to five times improvement with respect to existing reconfigurable computers.***

Keywords: Low-level vision, Reconfigurable architectures, Tracking.

1 Introduction

It is well known that real-time processing of video streams is a most expensive task from a computational point of view, due to the high amount of information to be processed. At a resolution of 640 × 480 pixels and 30 frames/sec, for example, the bandwidth of a single monochrome NTSC video stream is 9.2 M-Bytes/sec. The bandwidth of a color video signal is three times as much. Even when simple operations on pixel neighborhoods need to be carried out on such a data stream, the high bandwidth requirements rule out the use of conventional processors. For this reason, general purpose or dedicated massively parallel supercomputers based on the Single Instruction Multiple Data (SIMD) paradigm have long been advocated as a cure to this problem [1]. Massively parallel systems, however, have failed so far to provide a cost effective and flexible solution to the development and widespread use of vision systems, due to the physical constraints preventing their use on the field and their million-dollar price tags. Application Specific Integrated Circuits (ASIC's) have been widely used to implement low-level vision systems. Although they offer good performance, ASIC's do not lend themselves to rapid prototyping of systems and their development has high non-recurring engineering costs. Field Programmable Gate Arrays (FPGA's) emerged as a new technology for the implementation of digital logic circuits during the mid 80's. The basic architecture of an FPGA consists of a large number of Configurable Logic Blocks (CLB's) and a programmable mesh of interconnections [2]. Both the function performed by the logic blocks and the interconnection pattern are specified by a configuration stored in Static RAM (SRAM) memory cells scattered across the chip. This configuration can be specified by the circuit designer and easily changed at any time. In the beginning FPGA's were mostly viewed as large Programmable Logic Devices, thus they were usually employed for the implementation of the "glue-logic" used to tie together complex VLSI chips like microprocessors and memories used to build general purpose computer systems. In the late 80's and early 90's it became clear that the ability to change electrically the logic functions of FPGA's at almost any point

during operation could open an entirely new spectrum of applications in the field of high performance computing. Accelerators built using arrays of reconfigurable devices proved to boost the speed of several applications by up to three orders of magnitude, comparing favorably with supercomputers [3]. Recently, we have designed and demonstrated a 2-D feature selection system implemented on a commercially available FPGA-based reconfigurable computer [4]. This system is composed of a camera, a video decoder, an array of 6 Xilinx FPGA's and an interface to a host PC. This system is, to the best of our knowledge, the only feature selection system developed using reconfigurable devices. During this process we have learned several lessons:

- The use of an array of FPGA's to accomplish a given task adds a level of complexity to the design process, due to the need of manually partitioning the system across several chips. Moreover, signals crossing the boundary between neighboring chips incur additional latency, degrading system performance.

- Most low-level vision tasks can be accomplished by simple local operations performed across the image, which for the most part map nicely onto FPGA architectures. Although FPGA's lack native floating-point support, by carefully implementing these algorithms floating-point operations can generally be avoided.

- The majority of low-level vision algorithms process the image through a series of independent pipelined stages operating on local pixel neighborhoods of similar size (e.g. gradient computation, followed by nonlinear operations).

- Performance of real-time image processing systems is limited by the throughput of memory and I/O channels.

Based on these motivations, and the need felt by many practitioners in the computer vision community, we have designed a novel system level architecture tuned to real-time processing of video streams. This architecture exploits the locality of data access found in low-level vision algorithms and the recent availability of high pin count FPGA devices to partition in an optimal way memory and computation resources. The system that we envision is a PCI expansion board for a PC featuring a high density reconfigurable device, several synchronous SRAM memories and a digital interface for a high resolution progressive-scan video camera. A conservative estimate of the memory bandwidth that we will be able to achieve using off-the-shelf synchronous SRAM memory devices is 2.88 G-Bytes/s at a 60 MHz memory clock rate, which represents a four to five times improvement with respect to existing reconfigurable computers [5].

2 Requirements of real-time image processing systems

Image processing tasks carried out by low-level vision systems require both memory and computation resources. Memory resources are needed to feed the data to be processed to computation resources in a steady flow, and vary according to the nature of the space where the operation is defined. Spatial operations take into account every pixel of the image and require the availability of the pixel values belonging to a neighborhood defined by some geometric shape. Suppose that a pixel stream is transmitted in raster scan order by a video decoder, and that at every clock cycle a new pixel is available. The simple structure presented in Fig. 1 will make the values of the pixels belonging to a 3×3 square window available to computing resources. This window will slide across the entire image, covering a different pixel neighborhood at every clock cycle. This structure is composed of several First In First Out (FIFO) memories and registers synchronized with the video decoder. For a $k \times k$ square neighborhood the length of the FIFO is $M - k + 1$, where M is the

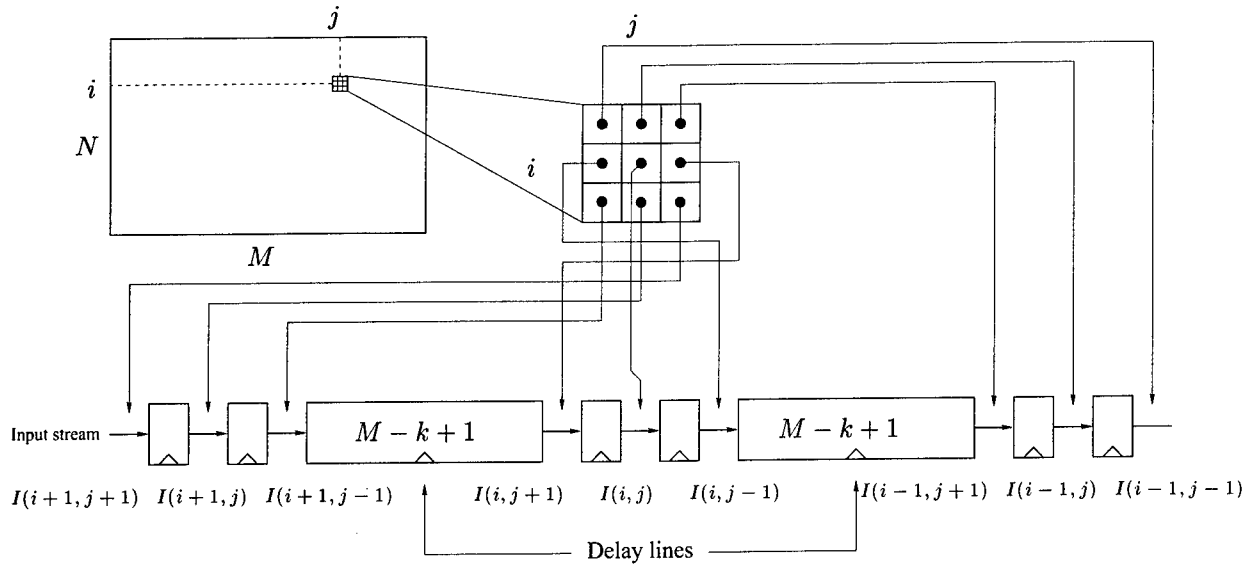


Figure 1: Formation of a 3×3 pixel neighborhood.

width of the image and usually $k \ll M$. In most FPGA architectures registers are abundant, and their implementation does not require excessive area. FIFO memories, however, require an excessive amount of CLB's when implemented as long shift register chains. In the Xilinx XC4000 FPGA architecture, for instance, each CLB contains two flip flops. At NTSC resolution, forming a 3×3 neighborhood would require six 8 bit registers and two 8 bits wide and 638 stages deep FIFO's, for a total of 5128 CLB's. On the other hand, the configurable logic blocks found in the XC4000 architecture can be configured as 34 bit SRAM cells, thus bringing the the number of required CLB's down to 302. However, when we consider operations requiring the pixel values of several frames, like filtering a video signal in the time domain, even last generation FPGA devices are not able to provide enough memory resources. The mechanism for neighborhood generation presented in Fig. 1 is easily adapted to the scheme employing an external RAM memory, as exemplified in Fig. 2. The two delay lines are here implemented by writing to the external RAM the pixel value entering the first FIFO memory and reading the values corresponding to the output of the FIFO's. The read addresses are obtained by decrementing the write address by $M - k + 1$, and after each pixel clock cycle they are incremented according to

$$\begin{aligned} l_W &= (l_W + 1) \mod 2^q, \\ l_{R_1} &= (l_{R_1} + 1) \mod 2^q, \\ l_{R_2} &= (l_{R_2} + 1) \mod 2^q, \\ &\vdots \\ l_{R_{k-1}} &= (l_{R_{k-1}} + 1) \mod 2^q, \end{aligned}$$

where q is the number of address lines of the memory device. Obviously, $2^q \geq (k-1)(M-k+1)$ must hold. According to this scheme, for every pixel clock cycle one memory write and $k-1$ memory read cycles are issued. Typical values for the pixel clock frequency are in the $12 \div 40$ MHz range, while off-the-shelf synchronous SRAM's are usually clocked at 100 MHz. This means that, according to image resolution, two or three cascaded delay lines will usually fit into a single external memory device.

3 A reconfigurable architecture for low-level vision

The data flow of many image processing systems can be decomposed as a sequence of op-

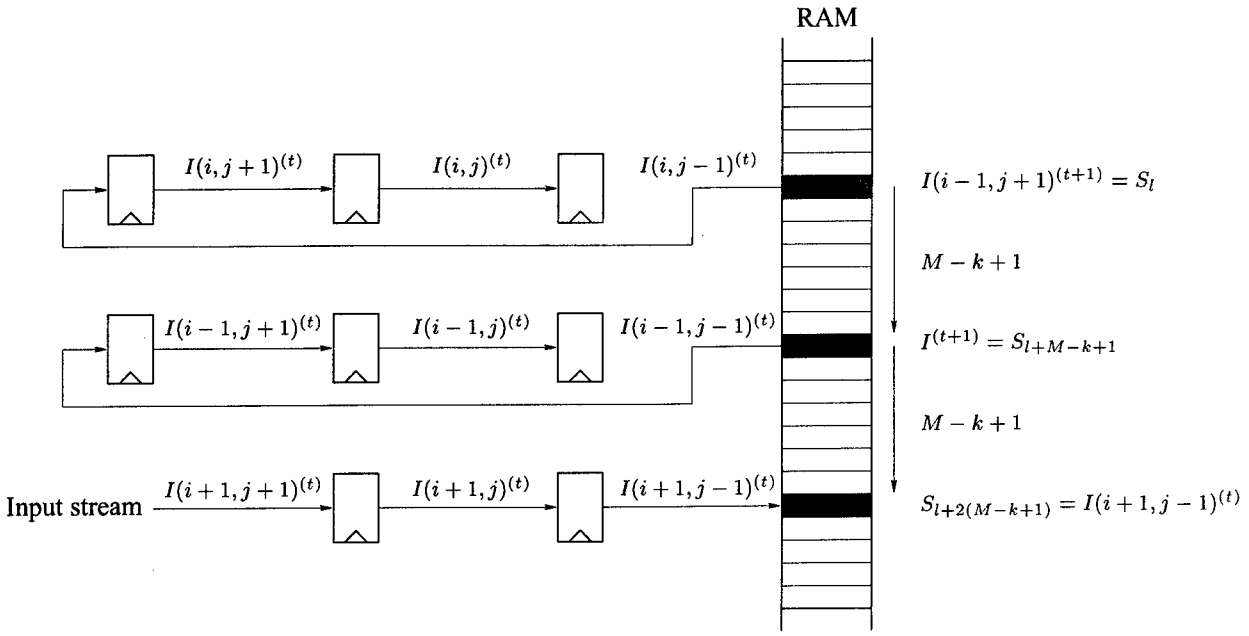


Figure 2: Building 3×3 pixel neighborhoods by external SRAM memory and internal CLB memory.

erations on sets of data whose organization resembles that of the initial image. The first stage of the feature selection system [4] depicted in Fig. 3, for example, computes the image gradient components I_x and I_y by convolving the input image respectively with the kernels $\begin{bmatrix} -0.5 & 0 & 0.5 \end{bmatrix}$ and $\begin{bmatrix} -0.5 \\ 0 \\ 0.5 \end{bmatrix}$. The next operation is the calculation of $(I_x)^2$, $(I_y)^2$ and $I_x \cdot I_y$, which are defined for every pixel in the image. Then a, b and c, defined by $a = \sum_{l=1}^{k^2} \left(I_x^l\right)^2$, $b = \sum_{l=1}^{k^2} I_x^l \cdot I_y^l$, $c = \sum_{l=1}^{k^2} \left(I_y^l\right)^2$, where the sum is extended over the pixels of a 3×3 neighborhood, are computed in parallel by three chains of adders interleaved with pixel and line delay elements in order to build a 3×3 mask in the $(I_x)^2$, $I_x \times I_y$ and $(I_y)^2$ planes. The rest of the system presented in Fig. 3 calculates the value of $P(\lambda_t) = (a - \lambda_t)(c - \lambda_t) - b^2$ by time-multiplexing a signed multiplier and performs the test expressed by

$$P(\lambda_t) > 0 \quad \text{and} \quad a > \lambda_t.$$

If the current 3×3 window passes the test, a red pixel is sent to the video encoder, meaning that that the window contains a "good" feature, otherwise the pixel value from the input stream is transmitted to the video encoder unchanged. For each intermediate operation of the algorithm, like the calculation of the gradients I_x and I_y and the coefficients a, b, c, memory resources are necessary to build the pixel neighborhood, whose content is shifted across the "images" associated with the input variables. For the sake of clarity, we will consider a $k \times k$ pixels square neighborhood, and will later relax this assumption. At every clock cycle the current values associated with the neighborhood feed a pipelined function block, computing some (arithmetic) function of the input data. The only constraint imposed on this block is that, after an initial latency of one or more clock cycles, it must generate an output data stream synchronous with the input data stream. The total latency introduced by this stage is thus given by the sum of the latency of the pipelined function block and the number of cycles needed to fill the delay lines so that the central pixel of the neighborhood corresponds to the first pixel of the input stream. Due to these latency periods, the output stream will be delayed with respect to the input stream. It is convenient to express this

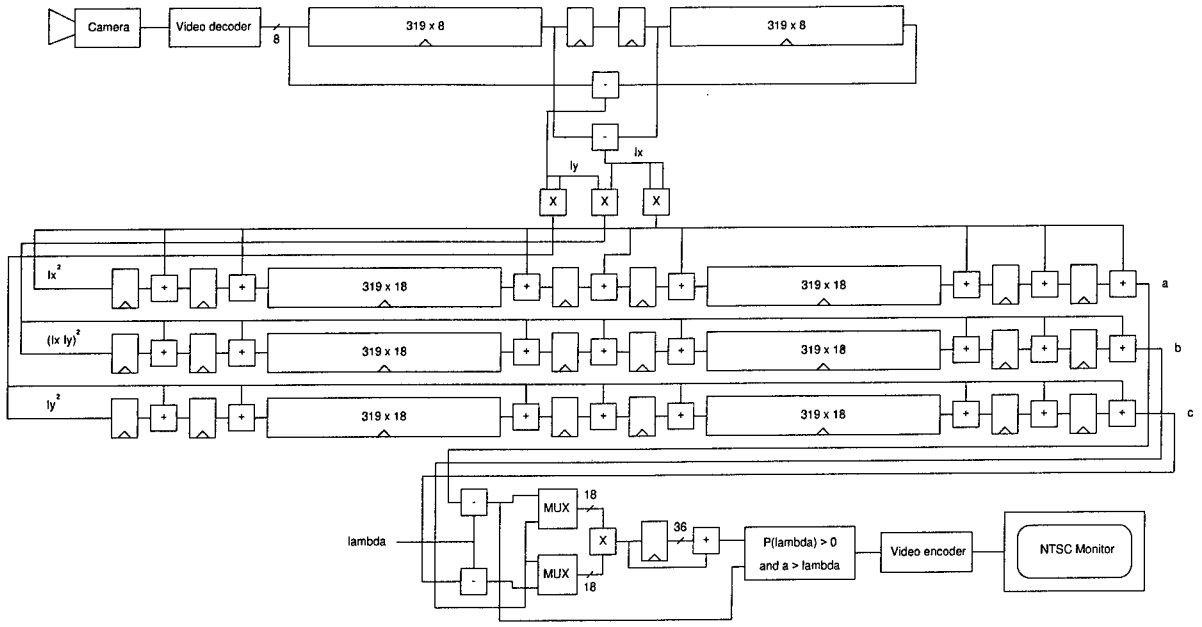


Figure 3: Schematic logic diagram of the real-time feature selection system.

phase shift in terms of an horizontal and a vertical components, which represent respectively the number of vertical pixel columns and horizontal scan lines by which the output stream has to shifted in order to be aligned with the input stream. Some processing stages, like those computing $(I_x)^2$, $I_x \times I_y$ and $(I_y)^2$, do not need memory resources since they compute numbers that are associated with individual pixels. Most stages, however, process pixel neighborhoods, thus a modular and efficient scheme for their generation is of the utmost importance for real-time video processing. In the architecture that we propose, the memory resources used to build pixel neighborhoods are provided by external synchronous SRAM memory devices, addressed according to the scheme presented in Fig. 2. The use of external memory devices has several important impacts on the design of the system. The most critical section of the system in terms of timing requirements is the FPGA to memory interface, which is clocked at up to 100 MHz, the maximum system clock frequency supported by most current generation FPGA's. The rest of FPGA logic can run at the slower pixel clock rate, usually in the $12 \div 40$ MHz range. In addition, the FPGA to memory interface can be easily generated from a high level specification of the algorithm that is being mapped. There is an additional key observation that can be exploited to further increase the memory bandwidth of a system based on this architecture. As shown in Fig. 2, the SRAM addresses are generated according to a fixed pattern, and their offset is $M - k + 1$, where M is the width of the image in pixel units and k is the size of the neighborhood. Different neighborhood sizes, denoted by k_m, may be used at the different P stages of the algorithm by taking

$$k = \max_{m=1,\ldots,P} k_m$$

and adjusting the length of the FIFO's used in each processing stage by inserting $k - k_m$ additional registers inside the FPGA. Using this strategy, the address increment is fixed indeed, and this property can be exploited to increase the memory bandwidth of the system as follows. First, we observe that memory devices are addressed according to a fixed and repeating pattern:

1. FPGA writes data to memory location

l_{W1},

2. FPGA reads from memory location $l_{R1} = l_{W1} - (M - k + 1)$,

3. FPGA reads data from memory location $l_{R2} = l_{R2} - (M - k + 1)$,

4. ...,

5. FPGA reads data from memory location $l_{R_{k-1}} = l_{R_{k-2}} - (M - k + 1)$,

6. Increment pointers to read and write locations,

7. Go to 1.

This property allows us to share the q address lines driving the memory devices. Let us put our attention to a high density and high pin count re-programmable device recently developed by Xilinx, the XC40125XV FPGA. The total number of I/O pins available to the user of this device is 448. If we dedicate 32 of these pins to communication with the digital camera and video monitor and 32 pins to communication with the PCI bus interface chip, the remaining 384 are available for interfacing with external memory chips. Up to 12 128K $\times$ 32 bit memory devices can be connected to the main FPGA. The number of FIFO memories that we will be able to fit in a single memory device depends on the widths of the data paths and on the constraint given by the fact that the delay lines implemented in the same device are necessarily cascaded. An estimate of the memory bandwidth that we will be able to achieve using this architecture, accessing the memory at a conservative 60 MHz clock rate, is thus 2.88 G-Bytes/s. This rate, represents a four to five times improvement with respect to existing reconfigurable computers. We emphasize that sharing the address lines is instrumental to achieve such a bandwidth. In fact, without sharing the address lines the maximum number of memory devices that we can connect to the FPGA drops from 12 to 7, and the bandwidth decreases by the same factor.

4 Conclusions

We have presented a novel reconfigurable architecture dedicated to fast prototyping of real-time low-level vision systems. An observation related to the mechanics of pixel neighborhood generation permits to increase almost by a factor of two the bandwidth of the communication channel between computation and memory resources. By exploiting this idea, an improvement of four to five times with respect to existing reconfigurable computers is achieved. We foresee the application of this architecture in general real-time signal-processing tasks, control systems for autonomous vehicle guidance, vision-based human-machine interfaces as well as in other applications not related to computer vision.

References

[1] C.-L. Wang, P. B. Bhat, and V. K. Prasanna. High-Performace Computing for Vision. *Proceedings of the IEEE*, 84(7):931–946, Jul. 1996.

[2] S. D. Brown, R. J. Francis, J. Rose, and Z. G. Vranesic. *Field-Programmable Gate Arrays*. Kluwer Academic, New York, 1992.

[3] D. Buell (editor). *Splash 2: "FPGA's in a Custom Computing Machine"*. IEEE Computer Society Press, 1996.

[4] A. Benedetti and P. Perona. Real-time 2-D Feature Detection on a Reconfigurable Computer. In *Proceedings of the 1998 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'98)*, pages 586–593, Santa Barbara (CA), Jun. 1998.

[5] J. Woodfill and B. Von Herzen. Real-Time Stereo Vision on the PARTS Reconfigurable Computer. In *Proceedings of the IEEE Symposium on FPGA's for Custom Computing Machines*, pages 201–210, Napa (USA), Apr. 1997.

1ms Sensory-Motor Fusion System with Hierarchical Parallel Processing Architecture

Masatoshi Ishikawa, Akio Namiki, Takashi Komuro, and Idaku Ishii

Department of Mathematical Engineering and Information Physics
University of Tokyo
7-3-1, Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
E-mail : ishikawa@k2.t.u-tokyo.ac.jp
URL : http://www.k2.t.u-tokyo.ac.jp/~sfoc/

Abstract *There is a growing interest in sensory-motor integration for realizing new behavior of intelligent robots and there must be some processing architectures integrated with the detection function. When viewed from the system as a whole, a parallel processing architecture is produced in which a part of the processing is distributed among the sensors. As a result of this, it is strongly required that a new hierarchical parallel distributed processing be introduced, corresponding to such a processing architecture. From such a viewpoint, this paper considers mainly the processing architecture for sensory information in robotics from new viewpoints such as massively parallel processing vision, high speed vision, active vision, and sensory-motor fusion. In addition, some demonstrations of grasping are presented as applications, and the perspectives of future sensor technology are discussed.*

Keywords: hierarchical parallel processing architecture, sensory-motor fusion, vision chip, grasping

1 Introduction

There is a growing interest in sensory-motor integration for realizing novel behavior of intelligent robots and mechanical systems. The key to the realization of high-level behaviors is sensory information processing technology such as sensor data fusion and hierarchical parallel processing architecture. With recent progress of the integration of electronic circuits, great changes will occur in the role and the techniques of the sensor and sensory information processing.

The most important point to be noted is that with the progress of such integration the computation cost is exceeded by the communication cost. In other words, the sensor is no longer considered simply as a hardware device for transforming a physical value to an electrical value, as in conventional sensors, but rather as an information processing module including sensory information processing.

In such a design, there must be some processing architectures integrated with the detection function. When viewed from the system as a whole, a parallel processing architecture should be necessarily introduced into the system in which a part of the processing is distributed among the sensors [1]. From such a viewpoint, this paper considers mainly the processing architecture for sensory information in robotics based on massively parallel processing vision, high speed active vision, and high speed sensor fusion.

New theory for constructing high speed sensory-motor fusion system using hierarchical parallel processing is proposed on high-speed sensory feedback. In addition, a 23 degrees of freedom robot system with high speed vision, force sensors is shown as an experimental platform.

By using the system, some demonstrations

of high speed grasping, tracking, and some applications such as robotics, human interface and virtual reality, are presented. In the demonstration, tracking, reaching grasping impedance control, and some application tasks are integrated into a unified algorithm.

Lastly, the perspectives of future sensory information processing and fusion technology are discussed.

2 Hierarchical parallel processing architecture

In this section a system architecture suitable to fuse sensor information is discussed by analyzing the real world environment. There are two important features in the real world, as follows:

(A) Flexibility under multiple conditions

A system should have flexibility to complete various tasks under various conditions. The process should be suitably changed according to the condition, for example an object's position, an object's shape and an object's motion.

To implement this, a hierarchical parallel processing architecture with several types of sensor is valid. Multiple types of sensory-motor fusion processing coexist in one system based on it. As a result, flexibility in multiple environments is realized.

(B) Responsiveness to dynamic changes

In the real world, the environment changes dynamically and is possible that the object moves at high speed and sudden accidents happen.

To overcome this, motion control based on high-speed sensory feedback is effective. High-speed sensory feedback means to return feedback of external sensor information at a rate higher than the rate of control. Because the system can recognize an external environment in real time, responsiveness to dynamic changes in the real world environment is realized.

We adopt an architecture in which both flexibility and responsiveness are realized. This is a hierarchical parallel architecture in which each element consists of high-speed sensory feedback within 1ms as shown in Figure.1. Because each feedback process is completed within 1ms, adjustment to various conditions is realized at high speed.

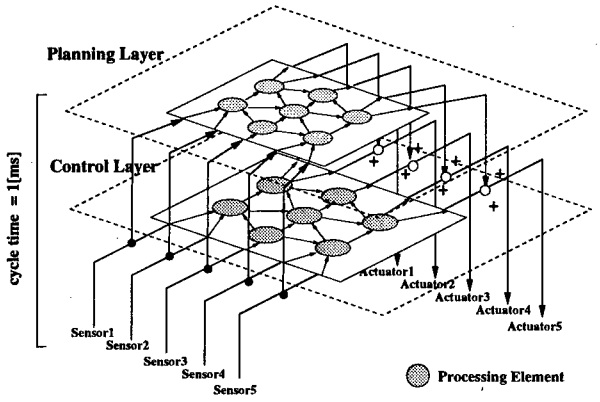


Figure 1. Hierarchical parallel processing architecture based on high-speed sensory feedback

In general the cycle time of 1ms is necessary to prevent mechanical resonance in robotic control. In our architecture we decided that the cycle time of each sensory feedback should be 1ms to ensure stable motion control.

As a related research Albus proposed a hierarchical parallel architecture based on the model of humans [2], and Brooks proposed a behavior-based hierarchical architecture consisting of layered sensory feedback modules [3]. We adopt a similar hierarchical parallel architecture, but responsiveness based on high-speed sensory feedback is not considered in these architectures.

3 1ms sensory-motor fusion system

Using the idea of a hierarchical parallel architecture, we have developed a system called the "1ms Sensory-Motor Fusion System" to realize high-speed sensory feedback and fusion of sensory information. This system exhibits high performance processing of all sensory feedback, including visual feedback, with a cycle time of

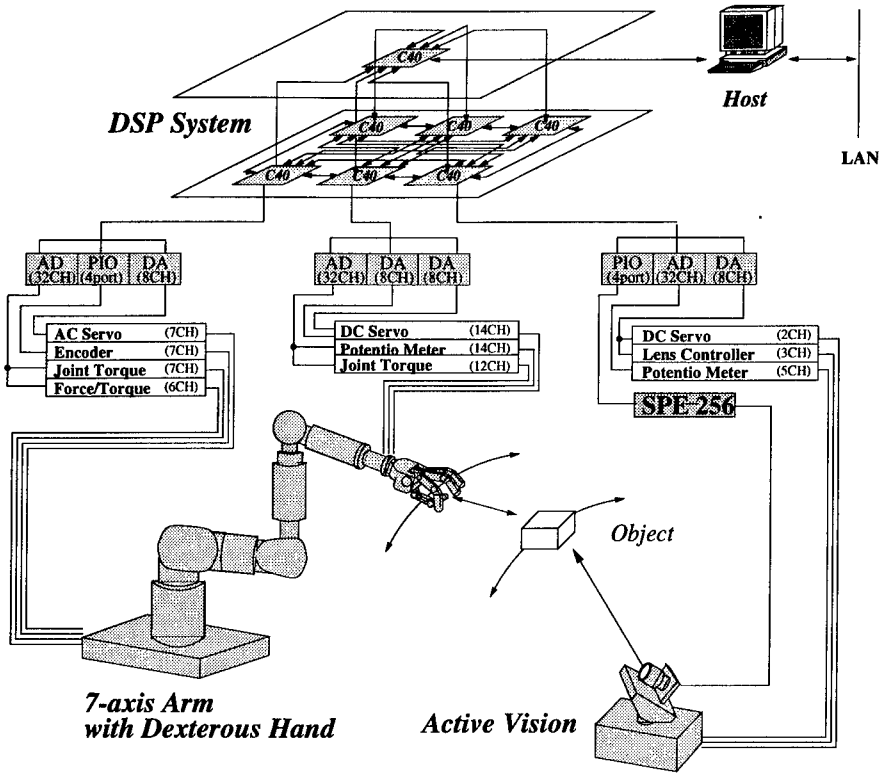


Figure 2. Architecture of 1ms sensory-motor fusion system

1ms. Because the processing result is directly used to control the manipulator, each task is realized with high responsiveness. Figure 2 shows the system components and Figure 3 shows a photograph of the system.

3.1 DSP parallel processing system

The DSP subsystem is the main part for fusion processing of sensory feedback within 1ms. It has a hierarchical parallel architecture consisting of 7 DSPs connected to each other, and many I/O ports are installed for inputing various types of information in parallel. In this system we use a floating-point DSP TMS320C40 which has high performance (275 MOPS) and 6 I/O ports (20 Mbytes/sec). By connecting several C40 processors, a low bottle-neck hierarchical parallel architecture is realized.

And In DSP system the following I/O ports are prepared; ADC (12 bit, 64 CH), DAC (12 bit, 24 CH), and Digital I/O (8 bit, 8 ports). These I/O ports are distributed on several DSPs to minimize the I/O bottleneck so that sensor signals are input in parallel.

A parallel programming development environment has been prepared in which multi-process and multi-thread programming is easily realized. This function is useful to program parallel sensory feedback.

3.2 High-speed active vision

The active vision subsystem consists of a vision chip system called SPE-256 and a 2-axis actuator moved by DC servo motors. SPE-256 consists of a 16×16 array of processing elements (PE) and PIN photo-diodes (PD). The output of each PD is connected with a corresponding PE. Each PE is a 4-neighbor connected SIMD based processor which has a 24 bit register and a bit-serial arithmetic logic unit capable of AND, OR, and XOR operations etc. Because the visual processing is perfectly exe-

cuted in parallel, high-speed visual feedback is realized within 1ms [6].

The actuator part of the active vision subsystem has two degrees of freedom; pan and tilt. This is used to move the sensor platform and this is controlled by a DSP assigned for active vision control.

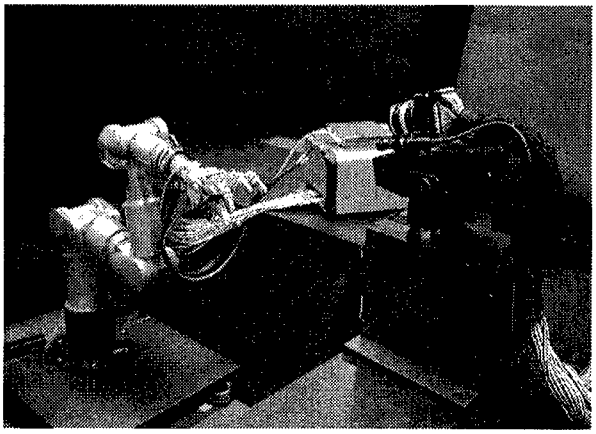


Figure 3. 1ms sensory-motor fusion system

3.3 Multi-fingered dextrous hand-arm

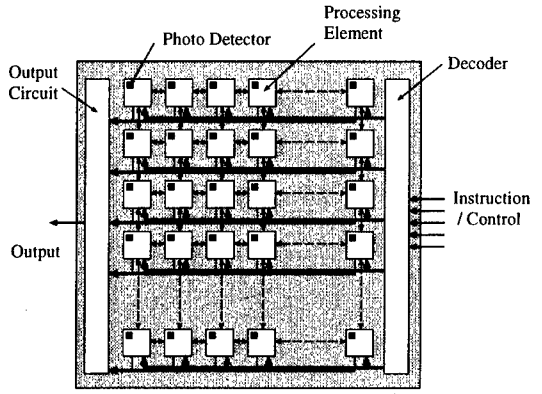
The hand-arm subsystem is a 7-axis manipulator with a dextrous multi-fingered hand. The multi-fingered hand has 4 fingers and 14 joints. Its structure is similar to a human hand, in which a thumb finger is installed opposite to the other three fingers. Each joint is controlled by DC servo motors in a remote place using a control cable consisting of an outer casing and an inner wire. Each joint of the hand has a potentiometer for position control and a strain gage for force control.

The arm has 7 joints controlled by AC servo motors. An encoder is installed in each joint and a 6-axis force/torque sensor is installed at the wrist.

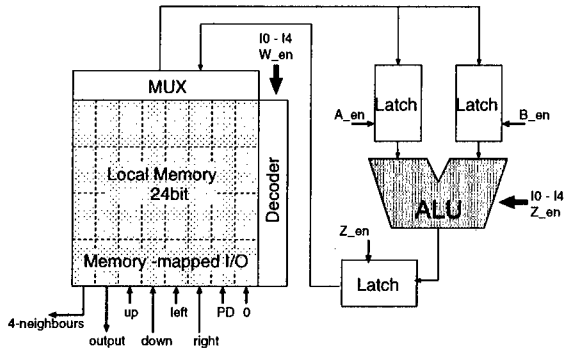
4 Vision chip

For real-time machine vision such as robot control using high speed visual feedback, traditional vision systems have an I/O bottleneck problem due to scanning and transmitting a large amount of image data, and the sampling rate is limited to video rates (NTSC 30 Hz / PAL 25 Hz). To solve this problem, we have developed the SPE-256. But this is a prototype scale-up model and an integrated architecture in one chip is needed.

For this reason we have developed a next generation vision chip architecture called S^3PE (Simple and Smart Sensory Processing Elements) [4]. In the vision chip architecture, photo detectors (PDs) and parallel processing elements (PEs) are integrated in a single chip without the I/O bottleneck, and the parallel PEs have general-purpose processing capability and are controlled by programs using digital circuits for real-time machine vision in robot control.



(a) the whole chip



(b) PE

Figure 4. Block diagram of vision chip architecture S^3PE

The block diagram of the whole chip is shown in Figure 4(a). General-purpose PEs are arranged in a massively parallel 2D array. Each

PE is directly connected to a PD, an output circuit, and its four neighboring PEs. Image signals from the PDs are A/D converted and transmitted in parallel to all the PEs. Instruction codes are decoded, transmitted to all the PEs, and executed simultaneously (SIMD type processing). The calculated result is transmitted to the output circuit and feature values such as moments are extracted and transmitted to an external system.

Table 1. Number of steps and time of sample programs on S^3PE

algorithm	steps(time[1])
4-neighbor edge detection (binary)	11 (0.72 μs)
4-neighbor smoothing (binary)	14 (1.0 μs)
4-neighbor edge detection (8bit)	70 (5.6 μs)
4-neighbor smoothing (8bit)	96 (7.7 μs)
4-neighbor thinning (binary) [2]	23 (1.9 μs)
Convolution (3×3, binary input)	40 (3.2 μs)
Convolution (3×3, 4-bit input)	372 (30 μs)
Poisson equation (4-neighbor, 4-bit) [3]	63 (5.0 μs)

[1] Calculated regarding an instruction cycle of 80 ns
[2] The process is repeated 10 times
[3] The process is repeated 200 times

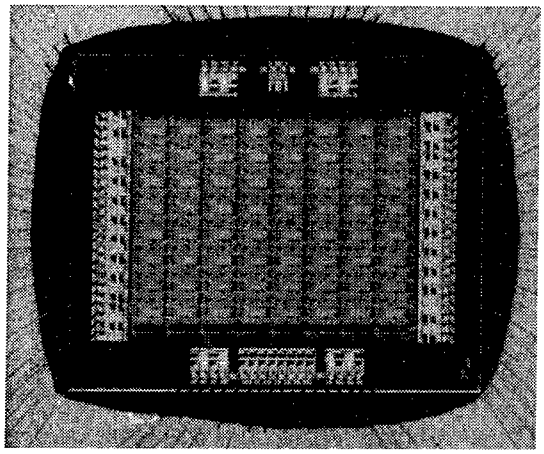


Figure 5. Photograph of the test chip

The block diagram of the PE is shown in Figure 4(b). Each PE has an ALU, a local memory, and three registers. The ALU consists of a full adder, four multiplexers and a D-flipflop for holding a carry bit and can execute 10 logical and 8 arithmetic binary operations. Multi-bit operations are implemented by repeating single operations serially (bit serial operation). The local memory has a 5-bit address space and consists of a 24-bit RAM and an 8-bit memory-mapped I/O which is connected to a PD, the output circuit, and four-neighboring PEs. Each bit can be randomly accessed.

In the vision chip, the main operation of the PEs is 2D pattern processing. In other words, 2D to 2D pattern transformations can be done in the PEs. Therefore, the total amount of data is still large. If the 2D pattern data were directly output to external pins, we would face the I/O bottleneck problem again. To avoid this problem, we introduced an output circuit which extracts feature values such as moments. To integrate the circuit together with PEs, a compact and homogeneous circuit design using digital circuits is required.

As shown above, the vision chip with the S^3PE architecture has general-purpose processing capabilities and can implement various algorithms. We developed some sample programs for the S^3PE and simulated them using a vision chip simulator we developed. The sample programs and the results of simulations are shown in Table 1. Assuming an instruction cycle of 80 ns, all of these programs are executed in much less than 1 ms, which is enough for robot control.

For the requirement to integrate digital PEs and analog PDs together on a single chip, and also to make the total area of the circuit as small as possible, a full custom design is necessary. The test chip fabricated in 1997 has 8×8 PEs and PDs in an area of 4.1 mm×3.7mm using a 0.8μm CMOS process. An SRAM technology is used in the local memory in this design. The number of transistors for the PE is 437 per pixel. Figure 5 shows a photograph of the chip.

It is estimated that 32×32 pixels can be integrated in 9.1 mm×7.9 mm using the same process. More than 64×64 pixels will be integrated using more recent processes. We have developed a test chip using a 0.35μm CMOS process.

We have already realized many applications such as target tracking, human interface using high speed vision system using vision chip

architecture[5, 6, 7, 8, 9, 10].

5 High speed grasping using visual and force feedback

We have realized grasping as an application of the 1ms sensory-motor fusion system[9]. The main aim is to realize high responsiveness to dynamical motion of a manipulated object by high speed visual feedback and force feedback with contact.

Figure 6 shows the block diagram of the grasping algorithm and Figure 7 shows a system configuration in high speed grasping. The manipulator with the dextrous hand and the active vision system are located face-to-face. Manipulated object moves between the manipulator and the active vision system, and the hand catches it by observing its position. Here we use two dimensional image features for the X-Z plane as visual feedback information.

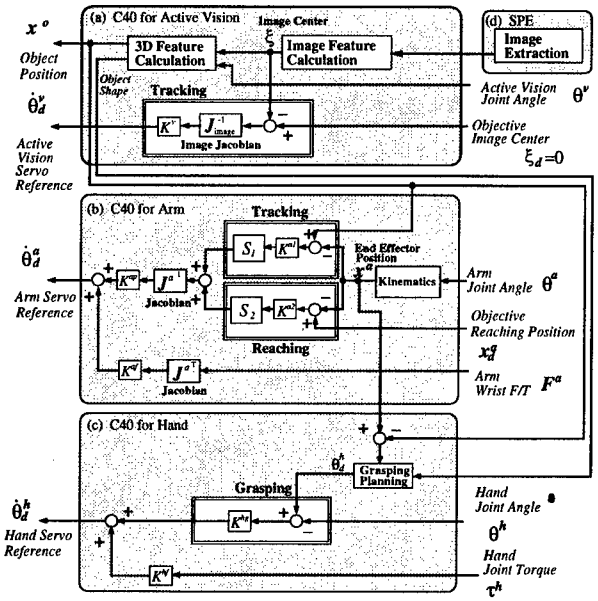
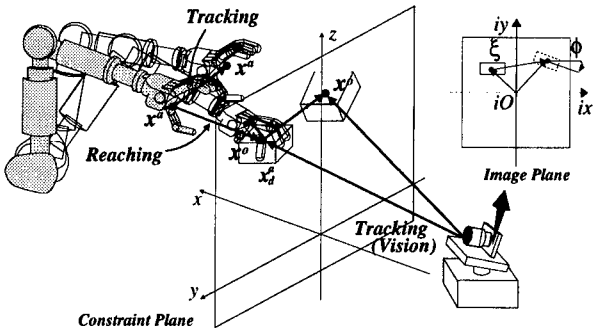


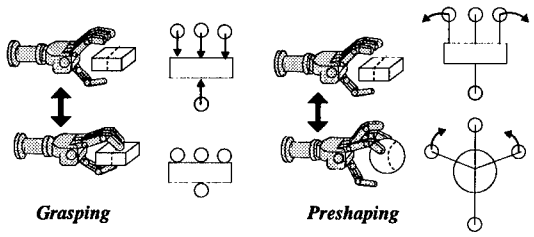
Figure 6. Algorithm of high speed grasping

Four feedback loops are executed in parallel to realize high performance processing in the high speed grasping system.

(a) Tracking (Active Vision): Tracking is done to acquire reliable object information. The active vision system is controlled so that the center of the observed object is always kept in the center of the image plane.



(a) Motion of the arm and the active vision



(b) Motion of the hand

Figure 7. Motion of High Speed Grasping

(b) Tracking (Arm): By canceling the object motion,tracking of the arm is done to keep the arm in a position suitable for grasping. In the algorithm, the relative position errors and the relative orientation error between the hand and the object observed by active vision are maintained at zero on the Y-Z plane.

(c) Reaching (Arm): Reaching of the arm is done to control the relative position between the hand and the object. In the algorithm, the arm moves from the initial position to the grasping position along the X axis. The initial position and the trajectory along the X axis can be given beforehand because the motion along the X axis is orthogonal to the tracking motion of the arm using visual information.

(d) Grasping (Hand): Grasping of the hand is done according to the relative distance between the object and the end-effector. Force sensor compliance control is used to realize stable grasping at each joint. The hand shape can be suitably adjusted for grasping according to the object shape obtained by visual information.

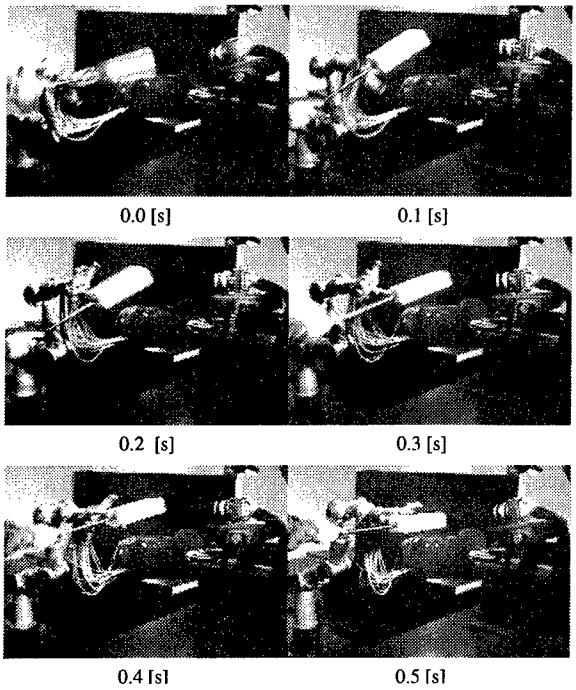


Figure 8. Experimental result: grasping of a hexahedron

These four feedback controls are executed in parallel. Each cycle time of the feedback loops is less than 1.5ms, and adequate responsiveness to the real world is achieved without using prediction.

The experimental result is shown in Figure 8 as a continuous sequence of pictures[10, 11]. All sensory feedback is executed in parallel according to the object motion at high speed: tracking motion of the active vision, tracking and reaching motion of the arm, and grasping motion of the hand. In Figure 9 a close-up view of the same motion is shown. In this figure tracking is executed from 0.0ms to 0.5ms and both reaching and grasping motion start at 0.5ms and all motion is completed at 0.8ms. Then in Figure 10 a close-up view of the grasping motion of a spherical object is shown. It is shown that the shape of the hand is changed to a suitable shape for grasping of a sphere.

In Figure 11 the trajectory of the hand is shown when grasping and releasing are alternately executed. In this figure, the Y axis position of the hand and the object show the tracking motion, and the X axis position of

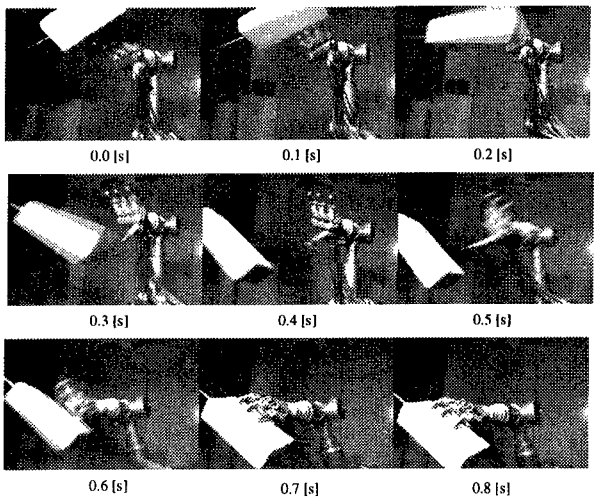


Figure 9. Experimental result: grasping of a hexahedron

the hand and objective trajectory for reaching motion show the reaching motion. This figure shows that both responsive tracking by visual feedback during the releasing phase and stable grasping by visual and force feedback during the grasping phase are realized.

In these experiments, because the object is moved by a human hand, its trajectory is irregular and difficult to predict. Using the speed of the sensory feedback this problem is solved.

6 Conclusion

This paper is based on the idea that parallel processing and high speed sensory information processing should be positively introduced into sensor feedback systems and an architecture is discussed using some applications.

References

[1] Masatoshi Ishikawa: The Art of Sensing, *Tutorial note of Int. Symp. on Measurement and Control in Robotics*, 1994

[2] J.S. Albus: Outline for a theory of intelligence, *IEEE Trans. on Systems, Man, and Cybernetics*, 21(3):473–509, 1991.

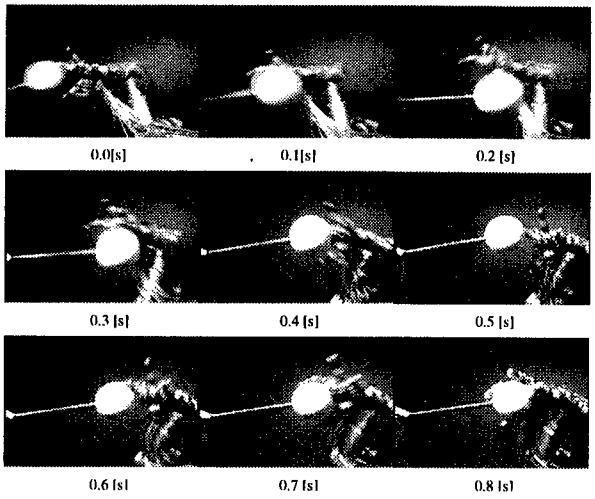


Figure 10. Experimental result: grasping of a sphere

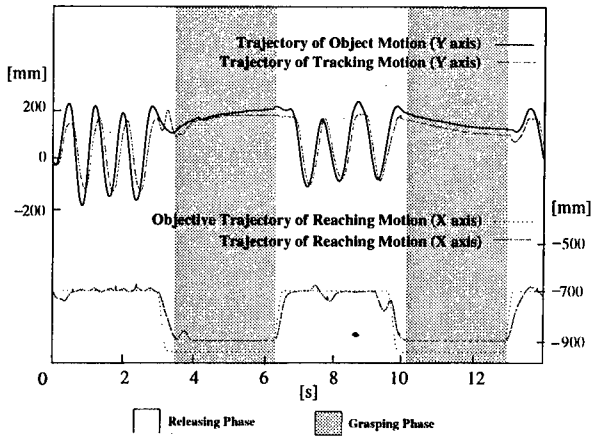


Figure 11. Feedback response

[3] R.A. Brooks: A robust layered control system for a mobile robot, *IEEE Journal of Robotics and Automation*, RA-2(1):14–23, 1986.

[4] Takashi Komuro, Idaku Ishii, and Masatoshi Ishikawa: Vision Chip Architecture Using General-Purpose Processing Elements for 1ms Vision System, *Proc. IEEE Int. Workshop on Computer Architecture for Machine Perception*, pp.276-279, 1997.

[5] Yoshihiro Nakabo, Idaku Ishii, and Masatoshi Ishikawa: High Speed Target Tracking Using 1ms visual Feedback System, *Abst. Video Proc. IEEE Int. Conf. Robotics and Automation*, p.6, 1996.

[6] Idaku Ishii, Yoshihiro Nakabo, and Masatoshi Ishikawa: Target Tracking Algorithm for 1ms Visual Feedback System Using Massively Parallel Processing, *Proc. IEEE Int. Conf. Robotics and Automation*, pp.2309-2314, 1996.

[7] Yoshihiro Nakabo and Masatoshi Ishikawa: Visual Impedance Using 1ms Visual Feedback System, *Proc. IEEE Int. Conf. Robotics and Automation*, pp.2333-2338, 1998.

[8] Takashi Owaki, Yoshihiro Nakabo, Akio Namiki, Idaku Ishii, and Masatoshi Ishikawa: Real-time System for Virtually Touching Objects in the Real World Using a High Speed Active Vision System, *Abst. and Ref. Video Proc. IEEE Int. Conf. Robotics and Automation*, p.2, 1998.

[9] Akio Namiki, Yoshihiro Nakabo, Idaku Ishii, and Masatoshi Ishikawa: High Speed Grasping Using Visual and Force Feedback, *Proc. IEEE Int. Conf. Robotics and Automation*, pp.3195-3200, 1999.

[10] Akio Namiki, Yoshihiro Nakabo, Idaku Ishii, and Masatoshi Ishikawa: 1ms Grasping System Using Visual and Force Feedback, *Abst. and Ref. Video Proc. IEEE Int. Conf. Robotics and Automation*, p.12, 1999.

[11] http://www.k2.t.u-tokyo.ac.jp/~sfoc/

Soft-Computing Integrated Circuits for Intelligent Information Processing

Tadashi Shibata
Department of Frontier Informatics
School of Frontier Science
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656 Japan
shibata@ee.t.u-tokyo.ac.jp

Masakazu Yagi
Deaprtment of Information and Communication Engineering
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656 Japan
goat@if.t.u-tokyo.ac.jp

Masayoshi Adachi
Deaprtment of Electronic Engineering
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656 Japan
adachi@if.t.u-tokyo.ac.jp

Abstract- Despite the enormous power of present-day computers, digital systems can not respond to real-world events in real time. Biological systems, while being built with very slow chemical transistors, are very fast in such tasks like seeing, recognizing, and taking immediate actions. This paper deals with the issues of how we can build real-time intelligent systems directly on silicon. An intelligent LSI system based on the psychological model of a brain is proposed. The system stores the past experience in the non-volatile analog vast memory and recalls the maximum likelihood event to the current input using the association processor architecture, where circuits are working in the analog/digital-merged decision making principle. Hardware-friendly algorithms have been developed to deal with real-time image recognition problems based on the association processor archiotecture.

Key Words: Association, neuron MOS, recognition, vector quantization.

1. Introduction

Over the past decade, we have witnessed a phenomenal progress in the computer technology. It is now possible to enjoy the super computer performance of some 15 years ago with our laptop PC's. With such overwhelming computational powers of present-day digital systems, however, it is not possible to respond to real-world events in real time. Namely, seeing, recognizing, and taking immediate actions are almost impossible for digital computers, while they are just effortless tasks for human beings, or biological systems in general. It is worth pointing out that biological systems are built with very slow chemical transistors, typically operating nine to ten orders of magnitude slower than short channel transistors available in current VLSI technology. We are missing something essential.

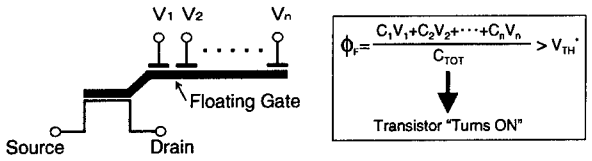


Fig. 1. Concept of neuron MOSFET (neuMOS or □MOS for short).

The strategy of our tackling the subject is in three folds. Firstly, the functionality of an elemental transistor is enhanced. Namely, the conventional MOS transistor working as a simple switch in digital circuits is replaced by a functional device and assigned more jobs to carry out at the very elemental transistor level. The subject is described in § 2. Secondly an association processor architecture has been developed as the hardware core of intelligent data processing. This is the realization of our very naïve model of a brain that recalling of the maximum likelihood event in the past memory is the bases of recognition [15,16]. The hardware model and its application to some practical problems are discussed in § 3 and § 4, respectively. In § 5, the third part of our strategy is presented, concerning the development of hardware-friendly algorithms, i.e., the algorithms for recognition that are most efficiently conducted in the association processor architecture. We have developed a very versatile method of extracting characteristic vectors from image data. The new vector representation has been applied to medical X-ray image diagnosis as well as to the recognition of handwritten patterns. Some preliminary results are presented.

2. Functionality Enhancement in Elementary Device

The concept of Neuron MOS Transistor (neuMOS or vMOS for short) [1] is shown in Fig. 1. The floating gate potential is determined by the multiple input signals via capacitance coupling and controls the on and off states of the transistor. Due to its functional similarity to the neuron model [2], the device bears the name. Applications of vMOS to binary digital circuits [3-7], real-time reconfigurable logic gates [3,5], self-learning neural networks [8], image processing [9,10], and analog multipliers [11] have been demonstrated.

Usually vMOS' are utilized in a CMOS inverter configuration to form a logic gate[3,4]. The accuracy of multivalue logic computation as well as the reduction in the power dissipation has been achieved by the

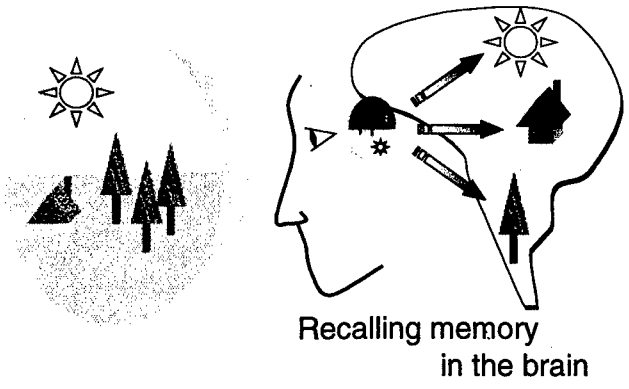


Fig. 2 "Seeing" is not objects imaging onto the retina but recalling of past memory triggered by the stimuli on the retina.

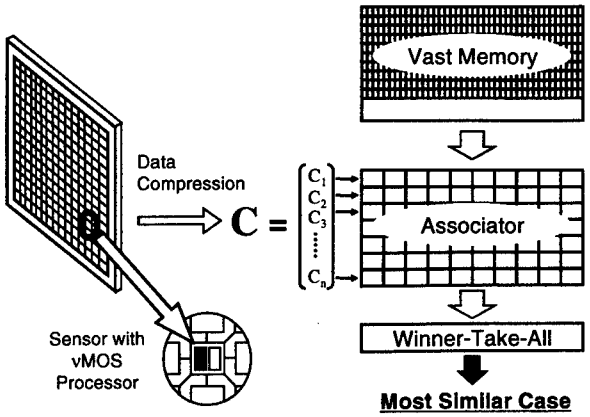


Fig. 3. Hardware recognition model.

introduction of clocked vMOS schemes [12-14].

3. Associatrion Processor Architecture and vMOS Circuits Implementation

3.1 Right Brain Computing Model

What are the intelligent functions to be implemented on integrated circuits? See Fig. 2. "Seeing and recognizing objects" is a very intelligent function of our brains. Then, what does "seeing" mean? "Seeing" is not mere optical imaging of objects onto the retina but that memorized images in the brain are recalled with their full richness of details triggered by the stimuli produced on the retina. Recalling past memory in immediate response to the current sensory inputs is the very bases of recognition. Based on this postulate, or so to speak a psychological brain model, we are tackling the subject of building "intelligent" electronic systems on silicon [15].

Our hardware recognition model is schematically illustrated in Fig. 3 [16]. An image captured on a two-dimensional pixel array is compressed into a characteristic vector consisting of a relatively small number of analog/mulitivalued variables each representing one of the salient features of the image by a respective code number. Then the association processor performs a parallel search for the most similar code vector in the vast memory where past experience is stored as template vectors. The association is conducted by calculating the distances between the input code vector and the stored template vectors and searching for the minimum distance vector by the winner-take-all (WTA) circuitry [17]. In building such systems, the analog/digital merged computation scheme using vMOS circuitry is utilized as a guiding principle.

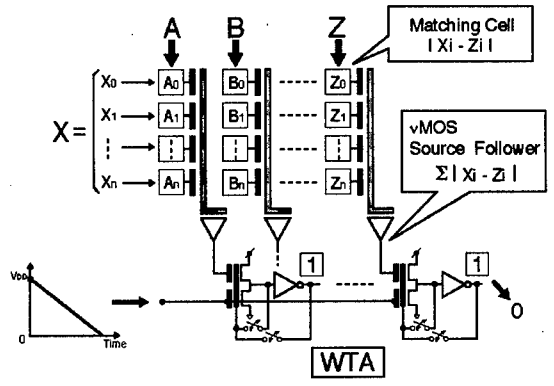


Fig. 4. Basic architecture of vMOS associator.

3.2 vMOS Association Processor

The architecture of the vMOS association processor is shown in Fig. 4 where $\boldsymbol{X}$ is an input vector and $\boldsymbol{A}$-$\boldsymbol{Z}$ template vectors down loaded from the vast memory. At each matching cell, the absolute value of difference $|X_i - Z_i|$ is calculated and transferred to the floating gate of a vMOS source follower and accumulated. Therefore the output of the vMOS source follower yields the Manhattan distance, the dissimilarity measure between the input vector and the template vector. The WTA is composed of vMOS inverters having two-equally weighted inputs. At time $t = 0$, all vMOS inverters are in on state. This is because V_{DD} is fed to one of the inputs and a non-zero distance value to the other, thus biasing the inverter above the threshold of $V_{DD}/2$. When the common voltage is ramped down, the vMOS inverter receiving the smallest distance value turns off firstly. At this moment, the feedback loop in each inverter is closed and the state of the inverter is frozen. The location of the smallest distance vector is identified by a flag appearing at the off-state inverter. Substantial computation is conducted by analog processing which is immediately followed by binary decision. This analog/digital-merged decision making operation

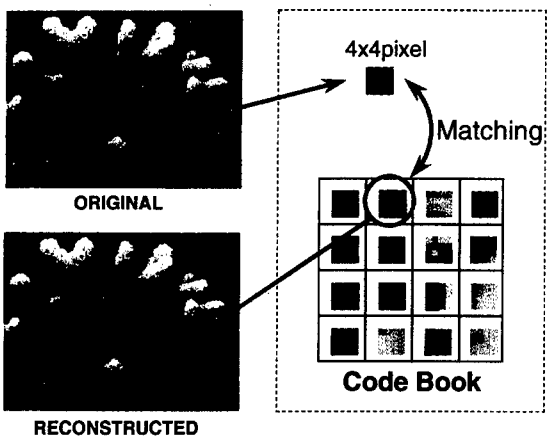


Fig. 5. Vector quantization (VQ) algorithm for image compression.

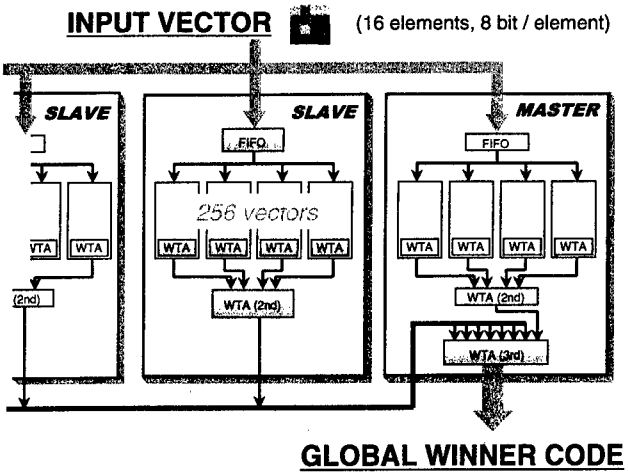


Fig. 6. Block diagram of VQ chip module.

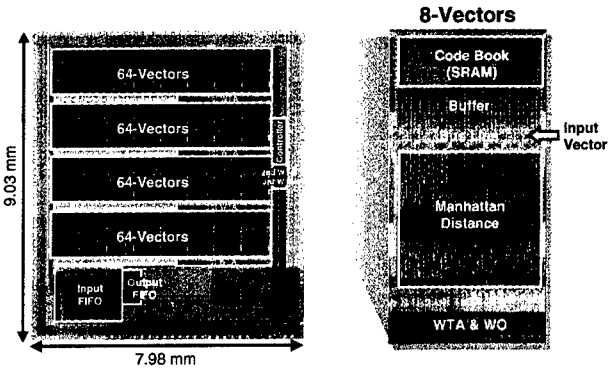


Fig. 7. Digital VQ processor for 256-template-vector parallel matching.

is an essential feature of the νMOS circuitry.

The absolute value circuit is simply composed of two floating-gate NMOS' connected at their source terminals [18,19] to form a source-follower MAX circuit. In order to achieve a mass storage of knowledge in the form of analog template vectors, a high-precision analog EEPROM technology has been developed [20]. The chip does not require time-consuming write/verify cycles [21] to write mulitivalue or analog data in the cell.

4. Applications of Association Processor Architecture

4.1 Vector Quantiation (VQ) Processor for Motion Picture Compression

As a straightforward application of the association processor architecture, the vector quantization (VQ) chips have been developed for motion picture compression and about three orders of magnitude faster performance has been demonstrated as compared to typical CISC processors. The VQ chips were implemented in conventional CMOS digital circuitry employing a fully parallel SIMD architecture [22,23] as well as in the νMOS circuitry [24], resulting in the eight times higher integration density in the νMOS implementation. This is briefly described in the following.

4.2 VQ Algorithm

The vector quantization (VQ) [25] algorithm employed in the system is explained in Fig. 5. A fragment taken from the original picture (4x4 pixels for instance) is an abstract pattern of gray patches, which can be approximated by one of the template patterns stored in the code book. Thus the pixel data are compressed to the code number of the template. Although the algorithm is straightforward, the template matching is an extremely expensive computation. However, this is what the association processor can carry out most efficiently.

4.3 Digital VQ processor

In order to prove the VQ algorithm is effective for motion picture compression, we first implemented a VQ processor in a pure digital CMOS technology. The most important concern of the system is the real-time encoding of motion pictures. In order to encode a 640×480 full color picture in a 4:1:1 format within 33 msec, a single VQ operation must be completed within 1.1 μsec. Our strategy toward this end is as follows. Firstly a fully parallel SIMD architecture has been employed. Secondly a single VQ operation is conducted in two pipeline stages, each pipeline segment consisting of 19 cycles. As a result, a single VQ operation is finished in every 1.1 μsec at a clock frequency of 17 MHz. Thirdly the chip is extendible to 8-chip master-

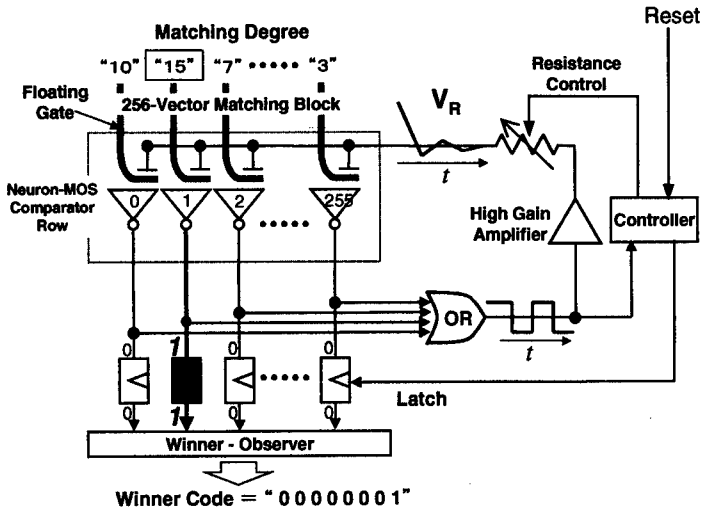


Fig. 8. vMOS WTA architecture.

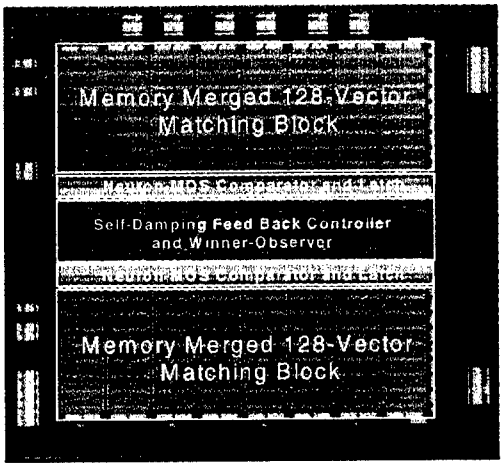


Fig. 9. Photomicrograph of vMOS VQ chip.

slave configuration, enabling us to perform a fully parallel search for maximum 2048 template vectors in 1.1 μsec.

Fig. 6 shows the block diagram of the VQ chip module, which is composed of eight VQ chips, namely one master chip and seven slave chips. Each VQ chip stores 256 template vectors in the embedded SRAM. The input vector is given to all the chips at the same time and the parallel search for the minimum-distance template vector is carried out in three stages of competition using digital winner-take-all (WTA) circuits.

Fig. 7 shows a photomicrograph of the chip fabricated in a 0.6-μm single-poly triple-metal CMOS technology. A single VQ operation for 2K template vectors on typical CISC processors requires roughly 1.2 M operations. This number was derived from the estimation: (38 operations/element) × (16 elements/vector) × (2048 vectors/VQ) = 1.2 M operations/VQ. The present VQ system in the eight-chip configuration can do this job in 1.1 μsec, which is equivalent to a CISC processor performance of about 1000 GOPS (1.2M operations/ 1.1 μsec).

4.4 vMOS VQ Processor

An analog vector quantization processor has been also developed using the neuron-MOS (vMOS) technology [24]. In order to achieve a high integrating density, the template-merged matched cell [19] is employed in the absolute value circuitry. A new-architecture vMOS winner-take-all (WTA) circuit has been developed to resolve the trade-offs between the search speed and the discrimination accuracy.

In Fig. 8, the WTA architecture is illustrated. All 256 comparator outputs are fed to an OR gate and its output is fed back to the reference voltage terminal of each comparator, thus forming a multiple-loop ring oscillator. The loop gain is controlled by the variable resistance inserted in the loop. At the start of WTA activation, all the vMOS comparators turn on and the OR output starts an 1-to-0 transition. This transition is fed back to all comparators and provide them with a descending reference voltage. If one of the comparators upsets, the OR gate upsets also and starts a 0-to-1 transition. Detecting this transition, the controller increases the value of the variable resistance. In this manner the feed back gain is step-by-step reduced and the winner search accuracy is gradually increased from the coarse search with a low scan rate to the fine search with a high scan rate. As a result, the discrimination accuracy of 5mV has been achieved in five scan steps.

A photomicrograph of the analog VQ processor chip is shown in Fig. 9. The chip was built in a 1.5-μm double-polysilicon CMOS technology and has the chip size of 7.2mm×7.2mm. A single chip contains 256 16-element template vectors. This is equivalent to one eighth of the chip size of our previous digital CMOS implementation (built in a 0.6-μm CMOS technology) when the same design rules are assumed for both chips.

4.5 CDMA Matched Filter

The fully-parallel self-correlation matching technique based on the vMOS association processor architecture was first developed for the motion vector detection [16]. This principle has been extended to build a CDMA matched filter, one of the key components in the next-generation WB-CDMA wireless communication systems [26]. In this application the templates are binary vectors representing the short PN (pseudorandom noise) codes with varying phase shifts.

The chip architecture is shown in Fig. 10. An

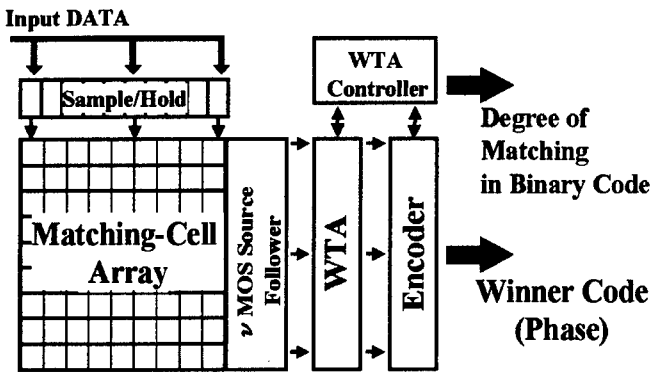


Fig. 10. Block diagram of vMOS matched filter.

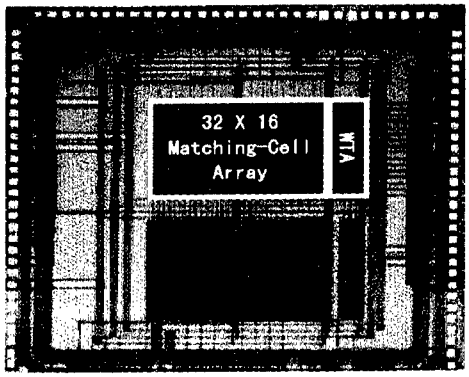


Fig. 11. Photomicrograph of a test chip of the vMOS MF fabricated in a 0.6-μm double-polysilicon triple-metal CMOS technology.

input signal train captured by sample and hold circuits is simultaneously matched with a group of templates having all possible shifts in the phase of an identical PN (pseudorandom noise) code. The maximum correlation is detected by fully parallel comparison using the binary-search vMOS winner-take-all circuit. Such a parellel architecture enables us to perform very fast peak detection as well as the detection of second or third correlation peaks arising from multi-path delays. A photomicrograph of the test chip fabricated in a 0.6-μm double-poly triple-metal CMOS technology is shown in Fig. 11.

5. Characteristic Vector Extraction from Images

So far we have been discussing the hardware implementation issues of the association processor architecture and have demonstrated its powerful nature in several practical applications. In the following the application of the architecture to image recognition problems is presented.

5.1 Linear Vector Formation

Image data are usually represented by a two-dimensional array of pixel data, i.e., by a matrix, containing voluminous data. Effective dimensionality reduction in the input image while retaining the characteristic features is the most important concern. In order to fit the problem to the association processor architecture in Fig. 3, we must generate a one-dimensional array of numerals, which we call hereafter "a linear vector." The two linear vectors representing two resembling images must be closer in the vector space. A new linear vector representation method we have developed is described in the following.

An image of 64×64 pixels was first subjected to pixel-by-pixel spatial filtering to extract four-direction edges, i.e., horizontal, vertical and $\pm 45^{\circ}$. The detected edges are indicated by digital flags at their loca-

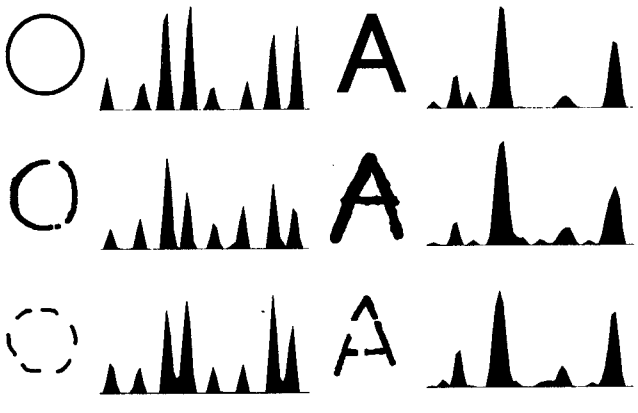


Fig. 12. Linear vectors representing circles and letter A's.

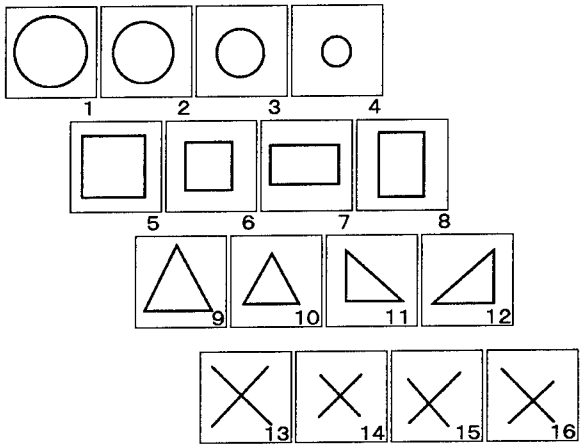


Fig. 13. Template patterns memorized for pattern matching.

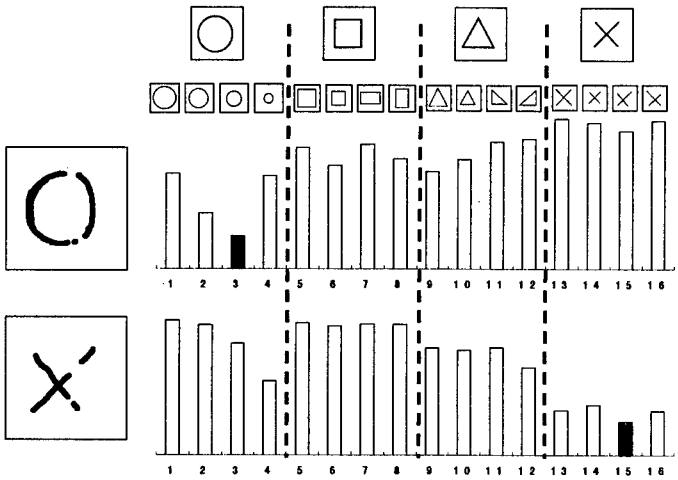


Fig. 14. Manhattan distance between presented pattern and each template vector.

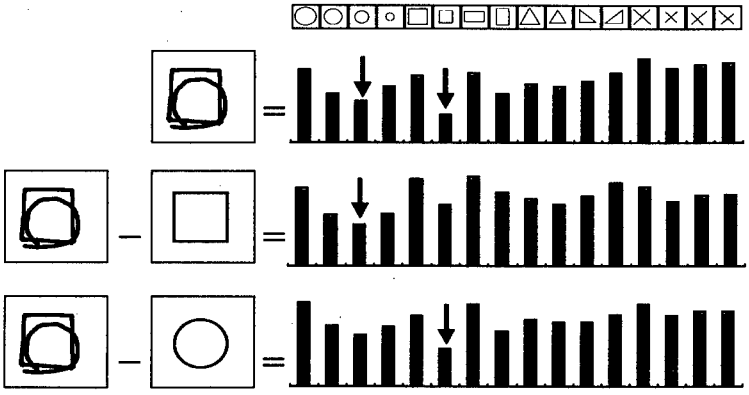


Fig. 15. Separation of overlapping patterns. When unknown pattern is presented, ○ and □ are recalled as the 1st and 2nd candidates. When the template of the recalled □ is subtracted from the input, ○ shows the strongest response. When ○ is subtracted, □ shows the strongest response.

tions, thus generating four feature maps from an original image. However, the representation is still two-dimensional and needs to be reduced to one-dimensional representation. For this purpose, we have introduced a new technique called "Principal Axis Projection." By Principal Axis Projection (PAP), we mean the flag bits are accumulated in the direction normal to the edge detection gradient, namely the horizontal edge flags are projected onto vertical axis, the vertical edges to horizontal axis, and $\pm 45^\circ$ edges to respective 45° -direction axes parallel to their edge detection gradients. The projection data obtained in each direction are reduced to a 16-element vector after merging and spatial averaging of the sum results. The four 16-element vectors obtained from four directions are cascade-connected to form a 64-element vector in the order of horizontal, $+45^\circ$, vertical, -45° , which we call a characteristic vector of the image.

5.2 Recognition of Simple Patterns

The powerful nature of the vector representation obtained by the PAP method is exemplified in Fig. 12, where the representations for hand-written patterns and characters are shown. The vectors representing the same pattern, i.e., letter A's or circles, all look alike. It is worth noting that one of the two hand-written A's is drawn in thick lines while the other is in thin lines, but resultant vectors look almost the same. This is due to the procedure of retaining only edge information by flag bits and summing and averaging them.

In order to test the performance in the pattern matching, linear vectors are formed from 16 simple patterns as shown in Fig. 13, and used as templates. The matching results are shown in Fig. 14 where the Manhattan distance between the input image and the templates are shown. Even with such distortions in the presented images, correct patterns are recalled as the shortest-distance vectors. So far the recognition of overlapping patterns is a very difficult problem. However, the present linear-vector forming technique has been successfully applied to such a difficult recognition problem as is demonstrated in the following.

Fig. 15 shows what happens when the system was presented with two hand-written patterns overlapping each other. The top row represents the distance between the input image and each template vector. The shortest and the second shortest indicated by arrows are a circle and a square, thus recalling correct candidates contained in the original image. How can such candidates be separated? In the middle row, the template vector of the square is subtracted from the vector of the input image and the residue is again matched with templates. Then the circle is recalled as the most similar. When the template vector of the circle is subtracted in the vector space, the square becomes the most similar template. From such observations we can infer that the original image presented is an overlapping of circle and square patterns.

5.3 Application to Medical X-ray Image Analysis

Automatic cephalometric landmark identification on radiographs is an important subject in establishing a fully computerized cephalometric analysis. The linear-vector formation technique developed in this work has been applied to this subject. In the following the preliminary results are presented.

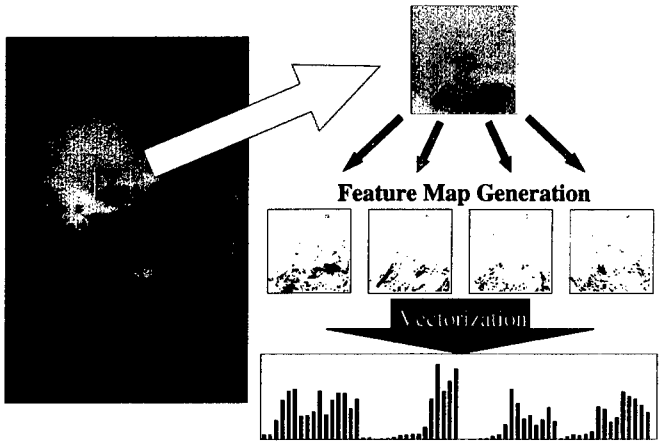


Fig. 16. Linear vector formation from radiograph of Sella (pituitary grand).

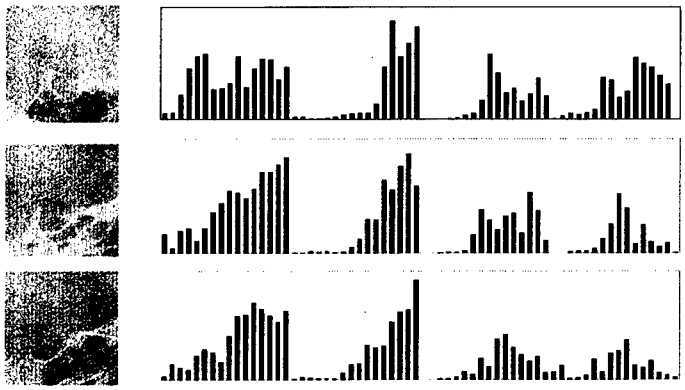


Fig. 17. Vector representation of pituitary grand images.

Fig. 16 represents the procedure of forming a linear vector from the image of Sella (pituitary gland) . Since the image is not a simple binary but a delicate gray scale image, the threshold value in the edge detection filtering process was determined taking the local intensity distribution into account. In Fig. 17, are shown the linear vectors formed from Sella images of three different patients. Evidently the vectors look very similar in shape and seem to work for identification by vector matching. In order to investigate the performance, the landmark identification experiments were carried out based on the vector formation method developed here.

Eight samples of cephalometric radiographs obtained by digital roentgen were prepared for experiments. One of the samples was selected as an input image for identification and the others were used as templates. Template vectors were generated by taking a 64×64 pixel block containing the image of Sella and transformed to a 64-diensional linear vector according to the procedure illustrated in Fig. 16. Using the seven template vectors as a template group, the position of Sella in an input image was detected by scanning the template group over the search area of 320× 240 pixels. Namely, at each point in the search area, the 64×64 pixel block is converted to a 64-dimensional linear vector and matched with the template group, and the highest score (the shortest distance) within the template group was recorded. The top 50 highest-ranking points were selected as candidates and indicated on the radiograph as shown in Fig. 18. The top 25 are indicated by white dots and the next 25 are by black dots. The procedure was repeated for all of the eight samples. The results are shown in Fig. 18.

Except for samples #8 and #11, nearly correct locations are identified. In sample #8, in addition to the correct location, false positions are also identified with higher rankings. After examining the matching results, it was found that the false identification is due to the similarity between the image at the false position and the template generated using the image of sample #4. We feel their similarity is acceptable to our eyes. This indicates that the pattern recognition based on the present vector representation is in some sense very analogous to our human processing and is likely to make mistakes like humans. In sample #11, the results are totally false. This is due to the fact that the sample itself is very different from others. Certainly we need more samples for templates and appropriate statistical manipulations on template vectors. The study on the subject is in progress.

The same procedure was conducted for identification of Nasion and the results are presented in Fig. 19. The results are much better than for Sella identification. It is interesting to note Nasion is characterized by its unique feature that clear curved lines running vertically and dark less-structured images on the right. It seems that this fact contributed to facilitating the vector-matching search.

Although the experiments are still in a preliminary stage, the present results are not very bad and seem promising. At present, these experiments are carried out by simulation on workstation and it takes a lot of time. The computation time for forming a single linear vector takes several minutes and the matching with a large number of templates takes much longer time. The design of a special hardware engine for feature map generation is in progress now using the vMOS technology. Our target is to finish the vector formation within a 1 msec.

6. Conclusions

The association processor architecture has been developed as a hardware core conducting the right-brain computation on silicon integrated circuits. The

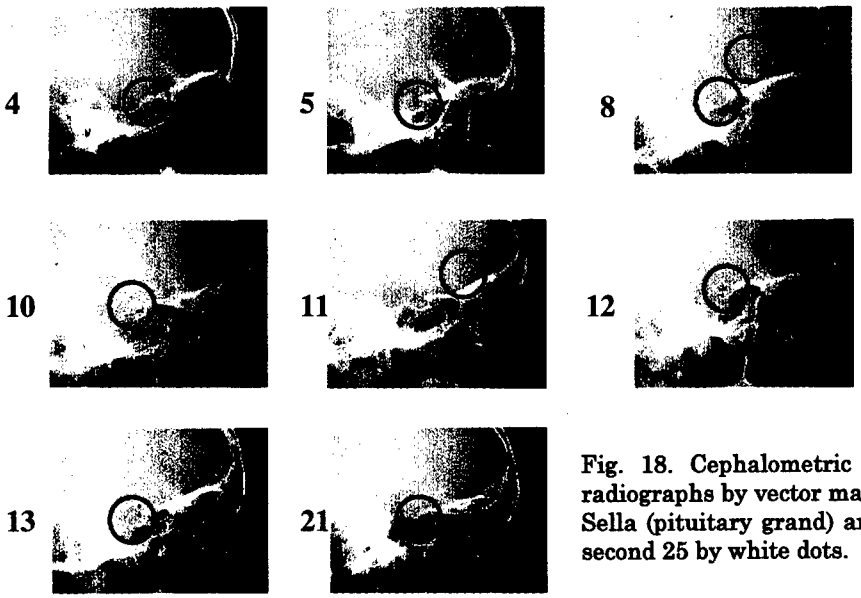


Fig. 18. Cephalometric landmark identification on radiographs by vector matching. Top 25 candidates for Sella (pituitary grand) are marked by black dots and second 25 by white dots.

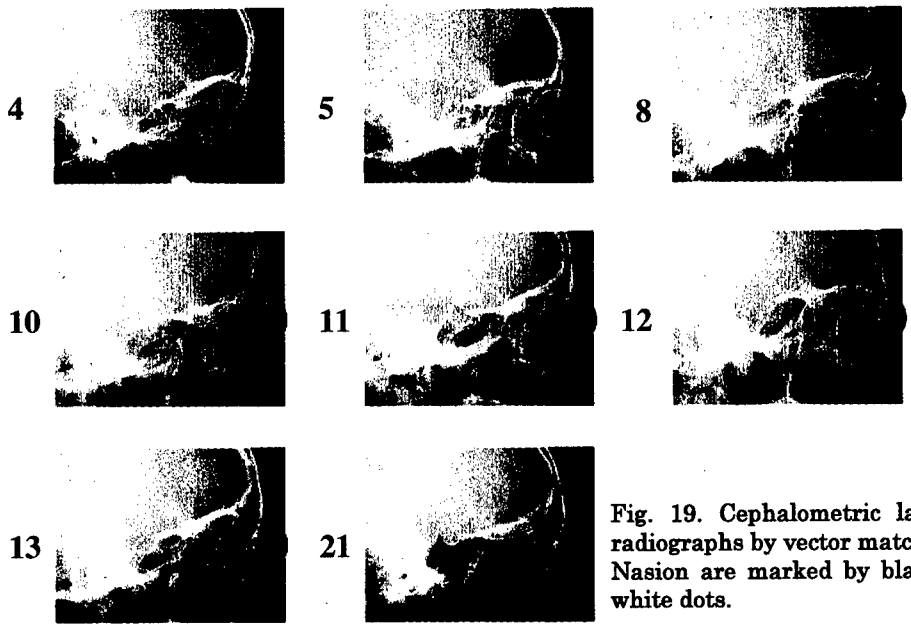


Fig. 19. Cephalometric landmark identification on radiographs by vector matching. Top 25 candidates for Nasion are marked by black dots and second 25 by white dots.

architecture has been applied to image recognition problems as well as to a number of practical applications and its powerful nature has been demonstrated. The architecture we have developed here will work for a general-purpose system and the specific application will be implemented in the system by installing template vectors deliberately prepared for each application.

Acknowledgement

The authors would like to thank Prof. Kenji Takada, Department of Orthodontics, Faculty of Dentistry, Osaka University for providing us with cephalometric radiograph samples as well as his expertise knowledge of landmark identification. Some of the chips presented here were fabricated in the chip fabrication program of VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Nippon Motorola LTD., Dai Nippon Printing Corporation, and KYOCERA Corporation and also in collaboration with Rohm Corporation and Toppan Printing Corporation.

References

[1] T. Shibata and T. Ohmi, "A functional MOS transistor featuring gate-level weighted sum and threshold operations," IEEE Trans. Electron Devices, Vol. 39, No. 6, pp. 1444-1455 (1992).

[2] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," Bull. Math. Biophys., vol. 5, p. 115 (1943).

[3] T. Shibata and T. Ohmi, "Neuron MOS binary-logic integrated circuits: Part I: "Design fundamentals and soft-hardware-logic circuit implementation," IEEE Trans. Electron Devices, Vol. 40, No. 3, pp. 570-576 (1993).

[4] T. Shibata and T. Ohmi, "Neuron MOS binary-logic integrated circuits: Part II, Simplifying techniques of circuit configuration and their practical applications," IEEE Trans. Electron Devices, Vol. 40, No. 5, pp. 974-979 (1993).

[5] T. Shibata, K. Kotani, and T. Ohmi, "Real-time reconfigurable logic circuits using neuron MOS transistors," in ISSCC Dig. Technical papers, Feb. 1997, FA 15.3, pp. 238-239 (1993).

[6] W. Weber, S. J. Prange, R. Thewes, E. Wohlrab, and Andreas Luck, "On the application of the neuron MOS transistor principle for modern VLSI design," IEEE Trans. Electron Devices, Vol. 43, No. 10, pp. 1700-1708 (1996).

[7] K. Ike, K. Hirose, and H. Yasuura, "A module generator of 2-level neuron MOS circuits," in the Proceedings of the 4th International Conference on Soft computing, Methodologies for the conception, design, and Application of Intelligent Systems (World Scientific, Singapore, 1996) pp. 109-112.

[8] T. Shibata, H. Kosaka, H. Ishii, and T. Ohmi, "A neuron MOS neural network using self-learning-compatible synapse circuits," IEEE J. Solid-State Circuits, Vol. 30, No. 8, pp. 913-922 (1995).

[9] Jun-ichi Nakamura and E. R. Fossum, "Image sensor with image smoothing capability using a Neuron MOSFET," in Charge-Coupled device and solid State optical sensors IV, Proc. SPIE Vol. 2172, pp. 30-37 (1994).

[10] M.Ikebe, M. Akazawa, and Y. Amemiya, "vMOS Cellular-Automaton devices for intelligent Image Sensors," Proceedings of the 5th International Conference on Soft Computing and Information/Intelligent Systems, 16-20 Octover, 1998, Iizuka, Fukuoka, "Methodologies for the Conception, Design and Applications of Soft Computing," Vol. 1 (T. Yamakawa and G. Matsumoto, Eds.) pp. 113-117.

[11] H. R. Mehrvarz and C. Y. Kwok, "A large-input-dynamic-range multi-input floating-gate MOS four-quadrant analog multiplier," IEEE Journal of Solid-State Circuits, Vol. 31, No. 8, pp. 1123-1131, August 1996.

[12] K. Kotani, T. Shibata, M. Imai, T. Ohmi, "Clocked-neuron-MOS logic circuits employing auto-threshold-adjustment," in Digest of Technical papers, 1995 IEEE International Solid-State Circuits conference (ISSCC), San Francisco, FP 19.5, pp. 320-321 (1995).

[13] K. Kotani, T. Shibata, and T. Ohmi, "DC-Current-Free Low-Power A/D Converter Circuitry Using Dynamic Latch Comparators with Divided-Capacitance Voltage Reference," 1996 IEEE International Symposium on Circuit and Systems (ISCAS 96), Vol. 4, Atlanta, pp. 205-208, May (1996).

[14] K. Kotani, T. Shibata, and T. Ohmi, "CMOS Charge-Transfer Preamplifier for Offset-Fluctuation Cancellation in Low-Power, High-Accuracy Comparators," Digest of Technical papers, 1997 VLSI Circuit Symposium, Kyoto, June, pp. 21-22 (1997).

[15] T. Shibata and T. Ohmi, "Neural Microelectronics," Technical Digest, International Electron Devices Meeting (IEDM) 1997, Washington D. C., pp. 337-342.

[16] T. Shibata, T. Nakai, N. M. Yu, Y. Yamashita, M. Konda, and T. Ohmi, "Advances in neuron-MOS applications," in ISSCC Dig. Technical Papers, Feb. 1996, SA 18.4, pp.304-305.

[17] T. Yamashita, T. Shibata and T. Ohmi, "Neuron MOS winner-take-all circuit and its application to associative memory," in ISSCC Dig. Technical Papers, Feb. 1993, FA 15.2, pp. 236-237.

[18] M. Konda, T. Shibata, and T. Ohmi, "Neuron-MOS Correlator based on Manhattan distance computation for event recognition hardware," 1996 IEEE International Symposium on Circuit and Systems (ISCAS 96), Vol. 4, Atlanta, pp. 217-220, May (1996).

[19] M. Konda, T. Shibata, and T. Ohmi, "A compact memory-merged matching cell for neuron-MOS association processor," Proc. MicroNeuro 97, Sept. 1997, Dresden, pp. 174-180.

[20] Y. Yamashita, T. Shibata, and T. Ohmi, "Write/Verify Free Analog Non-Volatile Memory Using a Neuron-MOS Comparator," 1996 IEEE International Symposium on Circuit and Systems (ISCAS 96), Vol. 4, Atlanta, pp. 229-232, May (1996).

[21] J. Hemink, T. Tanaka, T. Endo, S. Aritome, and R. Shirota, "Fast and accurate programming method for multi-level NAND EEPROMs," in 1995 Sym. VLSI Technology, Kyoto, Dig. Technical papers, pp. 129-130.

[22] T. Shibata, A. Nakada, M. Konda, T. Morimoto, T. Ohmi, H. Akutsu, A. Kawamura, and K. Marumoto, "A fully-parallel vector quantization processor for real-time motion picture compression," in ISSCC Dig. Tech. Papers, Feb. 1997, pp. 236-237.

[23] A. Nakada, T. Shibata, M. Konda, T. Morimoto, and T. Ohmi, "A fully-parallel vector quantization processor for real-time motion picture compression," IEEE Journal of Solid-State Circuits, Vol. 34, No. 6, June 1999 (to be published).

[24] A. Nakada, M. Konda, T. Morimoto, T. Yonezawa, T. Shibata and T. Ohmi, "Fully-Parallel VLSI Implementation of Vector Quantization Processor uisng Neuron-MOS Technology," to be published in IEICE Trans. Electronics.

[25] A. Gersho and R. M. Gray, "Vector quantization and signal compression," Kluwer Academic Publishers, Boston, 1992.

[26] A. Okada and T. Shibata, "A Neuron MOS Parellel Associator for High-Speed CDMA Matched Filter," to be presented at 1999 IEEE International Symposium on Circuit and Systems (ISCAS 99), Orlando, Florida, May 30-June2, 1999.

A Novel Image Enhancement Method Based on Intuitive Evaluation

Keiichi Horio, Takuma Haraguchi and Takeshi Yamakawa
Department of Control Engineering and Science,
Kyushu Institute of Technology
Iizuka, Fukuoka, 820-8502, Japan
{horio,takuma}@tsuge.ces.kyutech.ac.jp,yamakawa@ces.kyutech.ac.jp

Abstract *In this paper, we propose the new network which obtains the input/output relationship based on the user intuition. The intuitive evaluation of user is very important to evaluate the performance of human friendly information fusion systems. The self-organizing relationship (SOR) network proposed by the authors can extract the input/output relationship based on the evaluation function by unsupervised learning. By employing user intuition instead of the evaluation function of SOR network, the input/output relationship based on the intuitive evaluation of the user can be constructed. The effectiveness and validity of the proposed intuitive evaluation based SOR network by applying to the image enhancement.*

Keywords: intuitive evaluation, input/output relationship, self-organizing relationship network, image enhancement

1 Introduction

In the field of the engineering, the objectivity of information has been emphasized, and the information including the intuition or subjectivity has not been treated, because it has been the ones which lacks the generality. In recent years, increasing necessity of treating the system which relates to human, reduction of the intuition or subjectivity looks for the inconsistency between the knowledge of the theories and the real condition, and narrows the range of application of the theories[1].

On the other hand, the contrast of an image has an impact upon the intuitive impression of the user. In order to enhance images, many methods are proposed[2]-[4]. In almost all of these methods, the contrast of an image is represented as the evaluation function, and the original image is transformed to satisfy the evaluation function. However it is very difficult to represent the contrast of an image by discursive evaluation function, thus the transformed images sometimes do not accord with the user intuition.

In this paper, the new image enhancement method, in which the user intuition is employed as the evaluation function, is proposed. The self-organizing relationship(SOR) network, which is proposed by the authors and can construct the desired input/output relationship using the arbitrary evaluation function such as preference of users, is employed in order to realize the transformation corresponding to the user intuition.

The proposed method is applied to enhance the contrast of the images in accordance with the user intuition and evaluated. When this image enhancement method is implemented by hardware, it should be very useful system for applying to sensor devices.

2 SOR Network

The structure of the self-organizing relationship (SOR) network proposed by the authors is shown in Fig.1. The SOR network possesses the input layer, the output layer and the com-

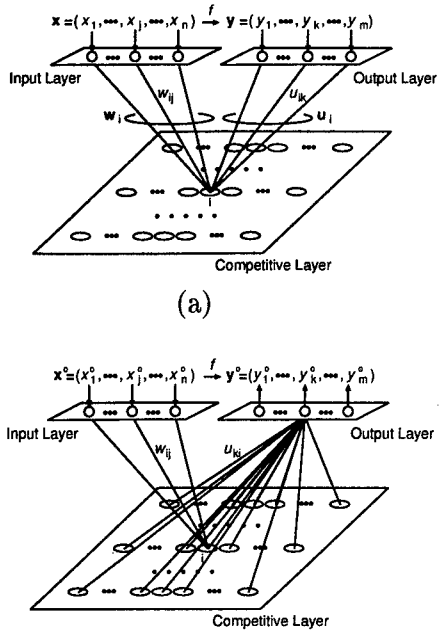


Fig. 1. The structure of the self-organizing relationship (SOR) network. (a)The learning mode. (b)The execution mode.

petitive layer containing n, m and N units, respectively. The i-th unit in the competitive layer connects to all units in the input layer and the output layer through the weight vector $\mathbf{w}_i$ and $\mathbf{v}_i$, respectively. There are two processes in the algorithm of the SOR network, the one is the learning mode, the other is the execution mode.

In the learning mode, random input/output pair $(\mathbf{x}, \mathbf{y})$ is applied, as the learning vector, to the input and the output layer together with the evaluation E for the learning vector. The evaluation E may be assigned by the network designer, given by the intuition of the user or obtained by examining the system under test. The positive E or negative E mean the good or bad relationship between the input vector and the output vector. The c-th unit in the competitive layer, which has the closest weight vector $\mathbf{v}_c = (\mathbf{w}_c, \mathbf{u}_c)$ to the learning vector $\mathbf{I} = (\mathbf{x}, \mathbf{y})$, is defined as the winner unit. The units that are located within the neighborhood of the winner unit are defined as the neighboring units. $\Delta\mathbf{v}_i$ calculated by Eq.1 is added to the old weight vectors $\mathbf{v}_i$ of the winner unit and neighboring units in order to obtain the new weight:

$$\Delta\mathbf{v}_i = \begin{cases} \alpha(t) \cdot E \cdot (\mathbf{I} - \mathbf{v}_i) & E \geq 0 \\ \frac{\beta(t)}{1-\beta(t)} \cdot E \cdot (\mathbf{I} - \mathbf{v}_i) & E < 0, \end{cases} \quad (1)$$

where $\alpha(t)$ and $\beta(t)$ are learning rate which decreases with time. In other words, when the evaluation E is positive or negative, the weight vectors of the winner unit and the neighboring units are attracted to or repulsed from the learning vector $\mathbf{I}$, respectively. The evaluation E is given by the user with intuition, the SOR network can construct the relationship between input vector and output vector based on the user intuition.

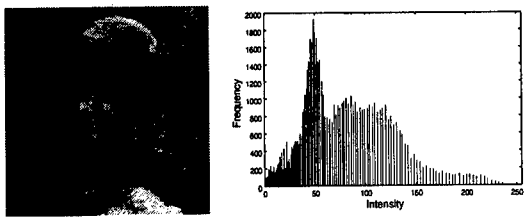
After the learning, the SOR network is ready to use as the I/O relationship generator. This operation is referred to as the execution mode and it is illustrated in Fig. 1(b). The actual input vector $\mathbf{x}^o$ is applied to the input layer, and the output z_i of the i-th unit in the competitive layer is calculated by:

$$z_i = exp(-\frac{\| \mathbf{x}^o - \mathbf{w}_i \|}{\beta}), \quad (2)$$

where β is a constant representing fuzziness of similarity. z_i represents the similarity measure between the weight vector $\mathbf{w}_i$ and the actual input vector $\mathbf{x}^o$. The output y_k^o of the k-th unit in the output layer is calculated by:

$$y_k^o = \sum_{i=1}^{N} z_i u_{ki} \; / \; \sum_{i=1}^{N} z_i, \quad (3)$$

where u_{ki} is a weight from the i-th unit in the competitive layer to k-th unit in the output layer and it is equal to u_{ik} obtained in the learning mode. The output of the network $\mathbf{y}^o = (y_1^o, \cdots, y_k^o, \cdots, y_m^o)$ represents the weighted average of $\mathbf{u}_i$ by similarity measure z_i. The relationship between actual input vector $\mathbf{x}^o$ and the output vector of the network $\mathbf{y}^o$ accords with the user intuition.



(a) (b)

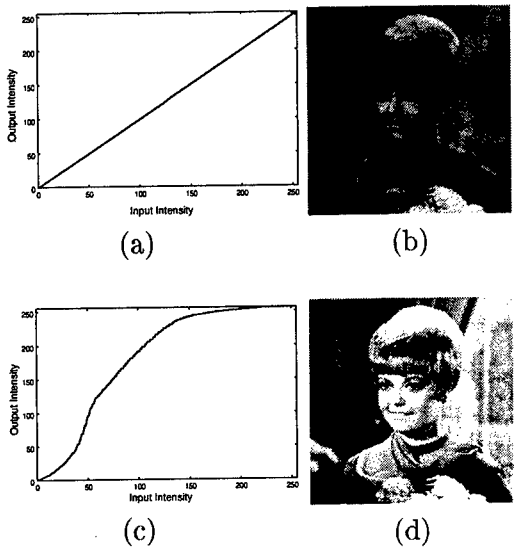
Fig. 2. Original Image (girl). (a)Image. (b)Intensity Histogram of the Image

3 Image Enhancement Using SOR Network

The new contrast enhancement method based on the user intuition is proposed. The method is realized employing the SOR network.

3.1 Conventional Method

In the image processing, the contrast enhancement is used to enhance or restrain the information of the original image in order to let the image easy to see for the user. As the conventional method of the contrast enhancement, there is the linear transformation (LT) and histogram equalization (HE). Both methods are known as the methods which are easy and powerful to enhance an image. Consider that the image shown in Fig.2(a) is enhanced. The intensity histogram of the image is shown in Fig.2(b). The levels of intensity of all images in this paper are 256. In the LT, the intensity mapping curve which extend the range of the intensity histogram of the original image from $[G_{min},G_{max}]$ to [0,255], which G_{min} and G_{max} are the minimum and the maximum intensity in the image, respectively. The intensity of the original image is transformed by using the intensity mapping curve. Fig.2(a),(b) show the intensity mapping curve and the image enhanced by the LT, respectively. The original image is enhanced naturally by this method, but if the range of the histogram of the original image is very wide, the method has no effectiveness. In the HE, the integrated function of the intensity histogram is employed as the mapping curve as shown in Fig.3(c). Fig.3(d) shows the image enhanced by the HE. The contrast enhanced image is obtained by this method. But the enhanced images sometimes have so strong contrast that the images are unnatural for users.



(a) (b)

(c) (d)

Fig. 3. The mapping curve for the image shown Fig.2 by each method and the enhanced image (a)Linear transformation (b)Histogram Equalization

In order to obtain natural images which have strong contrast, the methods using the local information of original images or the methods based on the if-then rules have been proposed[2]-[4]. In these methods, the contrast of an image is represented as an evaluation function, and an original image is enhanced to satisfy the evaluation function. Thus the decision of the evaluation function is very important. However it is very difficult to design the evaluation function corresponding to the user intuition. The enhancement method which reflects the user intuition is very useful.

The new image enhancement method, which generates the intensity mapping curve corresponding to the user intuition as shown in Fig.4, is proposed in this paper. In this method, the relationship, which is based on the intuitive evaluation, between the intensity his-

togram of the original image and the mapping curve is obtained by the learning.

3.2 Proposed Method

In the proposed method, the relationship between the histogram of the original image and the intensity mapping curve is approximated by the SOR network. The input vector is the intensity histogram of the original image. It is represented by the 256-dimensional vector $\mathbf{x} = (x_1, x_2, \cdots, x_{256})$, where x_i is the number of pixels whose intensity is i. The output vector is the intensity mapping curve which transforms the original image. It is represented by the 256-dimensional vector $\mathbf{y} = (y_1, y_2, \cdots, y_{256})$, where y_k is the output intensity for the input intensity k. $\mathbf{x}$ and $\mathbf{y}$ are employed as the input vector and the output vector of the SOR network, respectively. The evaluation of the relationship between $\mathbf{x}$ and $\mathbf{y}$ is given by the user who watches the image obtained by the intensity mapping curve $\mathbf{y}$. The learning of SOR network is achieved using these learning vectors and their evaluations. After the learning, the SOR network exhibits the desired relationship between intensity histogram and intensity mapping curve based on the user intuition. The intensity histogram of the image which should be enhanced is applied to the SOR network, and the desired intensity mapping curve is generated by execution mode of SOR network.

4 Experimental Results

The learning vectors $(\mathbf{x}, \mathbf{y})$ and their evaluations E for the learning of the SOR network should be obtained from subject at first. Fifteen images (Image 1 to Image 15) are prepared, and each image is transformed by fifteen mapping curves generated randomly, as shown in Fig.5. 225 transformed images (Image 1-1 to Image 15-15) are obtained and intuitively evaluated by the subject. In Fig.5, the evaluation of the Image p-q is 0.2, i.e., the relationship between H_p and MC_{p-q} is given the score 0.2 by the subject. The learning of the SOR network

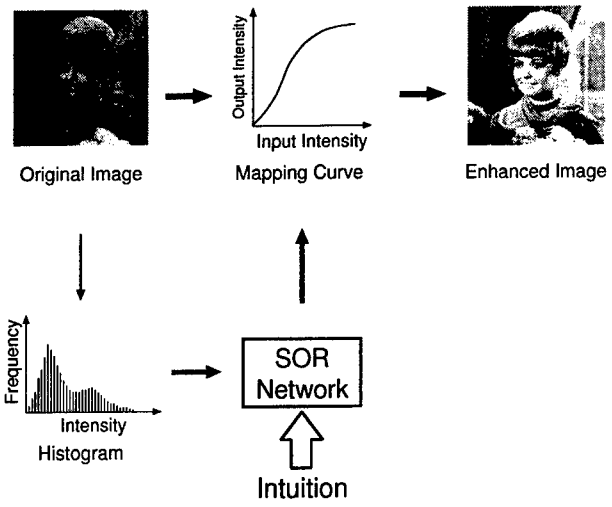


Fig. 4. Proposed image enhancement method. The intensity mapping curve for the intensity histogram of the original image is generated in accordance with the user intuition.

is achieved by using these 225 learning vectors and their evaluations.

In the learning, one learning vector is applied to the SOR network, and the weight vectors are updated in accordance with its evaluation. Applying all the learning vectors to the SOR network is defined as one iteration. In this experiments, the number of iteration is 300, the number of units in the competitive layer is 100(10×10), the initial value of the learning rate $\alpha(0)$ is 0.5, and the initial values of the weight vectors are random.

Consider that the test image 1 shown in Fig. 6(1-a) should be transformed appropriately. The histogram of the test image 1 is applied to the input layer of the SOR network after the learning, and the SOR network provides the intensity mapping curve in its execution mode. Here, the fuzziness parameter β shown in Eq.(2) is 1.0. Fig. 6(1-d) shows the image transformed by the intensity mapping curve which is generated by the SOR network. Fig. 6(1-b) and (1-c) indicate the images transformed by the LT and the HE, respectively. When these four images shown in Fig. 6(1-a),(1-b),(1-c) and (1-d) are presented to the

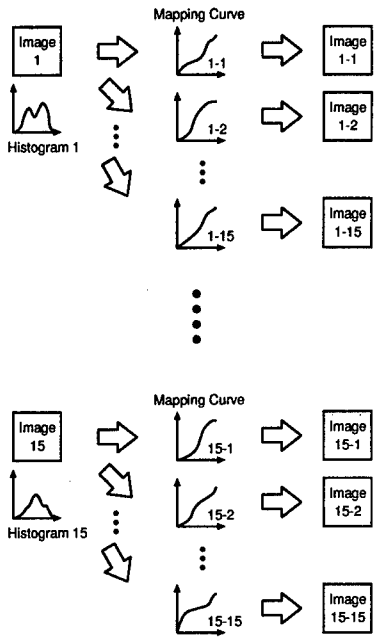


Fig. 5. How to obtain the learning vector $(\mathbf{x}, \mathbf{y})$ and to decide the evaluation E.

subject, it answers that it prefers the image transformed by the proposed method to other three images, because the image transformed by the LT has too poor contrast, and the image transformed by the HE has too strong contrast. Other four test images are transformed by the LT, the HE and the proposed method, and the transformed images are presented to the subject. It answers that it prefers the images transformed by the proposed method to other three images for all four test images.

The experiment above is achieved for seven subjects. For each test image, the original image and ones enhanced by three methods are ordered by the subjects according to their intuition. Table 1 shows the average of ranking for each test image. It is known that many subjects prefer the images by the proposed method to ones by other methods, and that the SOR network can construct the relationship between intensity histogram and intensity mapping curve based on the intuition of the subject.

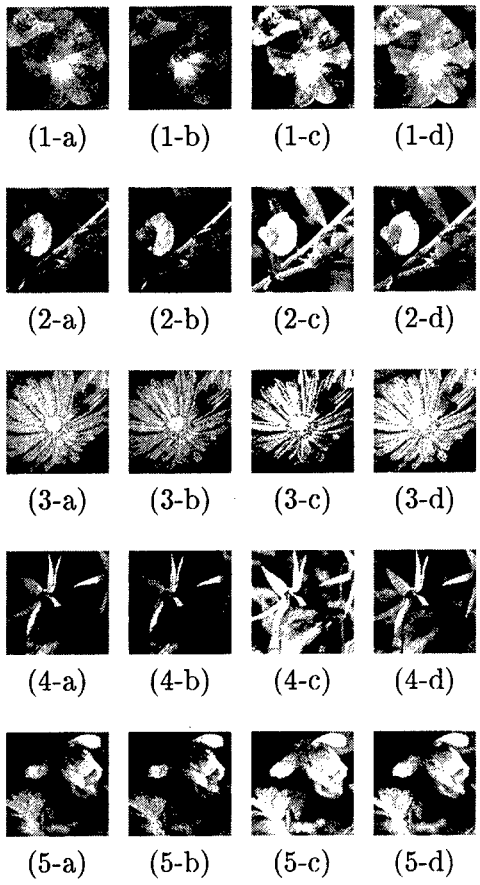


Fig. 6. Five test images and enhanced images. (-a)Original image. (-b)The image transformed by the LT. (-c)The image transformed by the HE. (-d)The image transformed by the proposed method.

5 Conclusions

In this paper, the new image enhancement method, which is based on the intuitive evaluation, is proposed. It is very important to consider the user intuition when images should be enhanced. Employing the user intuition as the evaluation function of the SOR network, the input/output relationship which is constructed by the SOR network accords with the intuitive evaluation of user.

It is applied to the image enhancement. The experimental results show that images enhanced by the proposed method accord with user intuition more than the images enhanced

Table 1. The average of the ranking for each image.

	original image	LT	HE	proposed method
image 1	2.71	3.29	2.29	1.71
image 2	2.29	3.14	2.71	1.86
image 3	2.42	2.57	3.86	1.43
image 4	3.00	3.43	2.43	1.43
image 5	2.43	3.71	2.57	1.29

by the other method.

References

[1] A. Yoshikawa and T. Nishimura, "Relationship between Subjective Dgree of Similarity and Some Similarity Indices of Fuzzy Sets," Proc. of IIZUKA'96, pp.818-821, 1996.

[2] E.L. Hall, R.P. Kruger, S.J. Dwyer, R.W. McLaren and G.S. Lodwck, "A survey of preprocessing and feature extraction techniques for radiographic images," IEEE Trans. computer, Vol.C-20, pp.1032-1044, 1971.

[3] S.M. Pizer, J.D. Austin, J.R. Perry, H.D. Safrit and J.B. Zimmerman, "Adaptive histogram equalization for automatic contrast enhancement of medical images," Proc. of SPIE, Vol.626, Medicine XIV/PACS IV, pp.242-250, 1986.

[4] A. Vanzo, "A simple cubic operator for sharpening as image," Proc. of 1995 IEEE Workshop on Nonlinear Signal and Image Processing, pp.963-966, 1995.

[5] T. Kohonen, "Self-organized formation of topologically correct feature maps," Biol. Cyber, Vol.43, pp.59-69, 1982.

[6] T. Kohonen, *Self-organization and associative memory*, Second edition, Berlin: Springer-Verlag, 1988.

[7] T. Yamakawa and K. Horio, "New design method of fuzzy logic controller using self-organizing relationship," Methodologies for the Conception, Design and Application of Soft Computing Proceedings of IIZUKA'98, pp.155-158, 1998.

3-D VLSI Architecture Implementation for Data Fusion Problems Using Neural Networks

Tuan A. Duong, David Weldon, and Tyson Thomas
Center for Space Microelectronics Technology
Jet Propulsion Laboratory, California Institute of Technology
Pasadena, CA 91109

ABSTRACT

This paper gives an overview of hardware implementation techniques employed in solving real-time classification problems using Neural Network, Principle Component Analysis (PCA), and Independent Component Analysis (ICA) techniques. The first part of the paper reviews digital, analog, and hybrid strategies for hardware implementation, outlining their advantages and disadvantages. The second part focuses on dedicated VLSI chips developed at the Jet Propulsion Laboratory (JPL).

A flexible neural network chip with 64 neurons and a 64x64 synaptic weight array with 8-bit resolution is first presented. This chip can be theoretically cascaded to form a larger network, connected in parallel to improve dynamic range or resolution, or connected in a loop to create a feedback neural network. A second neural network chip is presented that was fabricated using Silicon-On-Insulator (SOI) technology. This second chip operates at 1.5V, has neurons with variable transfer functions, and has completely compatible inputs and outputs, allowing simple and direct cascading and feedback. A 64x64 synaptic weight array chip is then introduced that has 8-bit resolution and a time response of less than 250ns. This chip was stacked to obtain a cube of 64 chips with an estimated data processing speed of 10^{12} operations per second.

A data input chip called the Column Loading Input Chip (CLIC) was designed, fabricated in 1.0μm CMOS technology, and tested. The chip can take 64x64 digital bytes and convert them into 64x64 analog inputs to a 3-D parallel processing cube. The CLIC was designed to raster through a large image window, taking a new 64-byte column or row of data from the main image every 250ns. The cube processes this data using PCA or ICA techniques and passes its output to a neural network classifier.

In the cube architecture, power consumption is one of the most important concerns and has, so far, inhibited designs of larger arrays. However, recent SOI technology seems capable of improving major aspects of performance by providing power consumption reduction, latch-up avoidance, and mixed signal noise reduction. A new 3-D architecture is proposed which is similar to the original cube but is more robust for stacking and easier to test, and its application to a hyperspectral sub-pixel classification problem is discussed.

I. INTRODUCTION

At JPL, we have developed a variety of chips that can be used as building blocks for hardware computation of general-purpose algorithms germane to sensor fusion. Our building block chips are cascadable to create larger networks that were necessary for some of our recent applications [1,2]. In addition, many of the chips are stackable in a third dimension to achieve increased parallelism, providing the computational power necessary to solve problems such as real-time spatio-temporal target recognition and Hyperspectral sub-pixel classification. Our latest 3-D chip stacks have been designed to provide computational power on the order of 10^{12} operations per second [3-5].

Section II discusses the hardware implementation strategy used in most of our chips, and explains why our approach is superior to the alternatives. Section III is an overview of the latest building block chips that we are currently using to create powerful prototype 3-D architectures. Section IV will show how the 3-D computational architectures created using our building block chips might be used to solve hyperspectral sub-pixel classification problems. The architecture presented uses Principal Component Analysis (PCA) [6] or Independent Component Analysis (ICA) [7-10] techniques to estimate end members, and then classifies these estimated end members using an artificial neural network.

II. IMPLEMENTATION STRATEGY

In order to accomplish real-time sensor fusion, fundamental operations such as addition, subtraction, and multiplication must be implemented in hardware. If artificial neural networks are to be used, the neuron transfer function must also be realized in hardware to achieve adequate speed. These operations have traditionally been implemented in primarily digital or primarily analog hardware [11,16-18], but we have developed hybrid implementations that retain the advantages of each approach while eliminating or minimizing their weaknesses [1,3].

Fully digital implementations such as the CNAPS board by Adaptive Solutions [11] are attractive for a number of reasons. First of all, digital memory allows for very robust long-term storage of synaptic weights, while digital computation has extremely high noise immunity. In addition, because of the binary nature of digital signals, very fast devices can be used without consideration for their linearity or accuracy. There is also a large amount of flexibility inherent in digital processing, allowing the implementation of nearly any desired architecture with as much precision as is required. This flexibility, however, does not usually include massively parallel implementations, especially those that are scalable. Digital implementations typically occupy a large amount of active die area as well, and have fairly high dynamic power consumption. The architectural limitations coupled with increased power consumption at high clock rates actually limit most digital implementations to relatively slow overall throughput, in spite of the high operational speed of the individual devices.

In contrast to digital implementations, analog techniques can be used to implement fully parallel architectures that are easily scalable. They are also capable of achieving higher throughput with lower power consumption and less die area than digital implementations. Unfortunately, they suffer from low noise immunity and their weight storage mechanism often requires refresh circuitry to maintain accurate values over long periods of time [12]. Alternative

approaches to analog memory, such as floating gate technology [19], eliminate the need for refresh circuitry, but they do not have arbitrary precision and cannot be updated with sufficient speed [13]. After learning, however, neural networks can tolerate relatively poor accuracy [14], so the noise and precision limits of analog computation may not be critical. In general, analog circuitry appears to be much more suitable than digital circuitry for high-density 3-D applications, but the difficulty of realizing refresh circuits across a 3-D chip stack is significant enough to warrant the use of an alternative approach.

In order to capitalize on the suitability of analog circuitry for 3-D architectures while maintaining the stability and accuracy of digital weight storage, JPL has adopted a hybrid approach. Synaptic weights are stored digitally, thereby eliminating the need for refresh circuitry while ensuring adequate time response during learning. Synaptic outputs are represented as analog current signals that can be easily combined with any number of other outputs using only a common wire. This leads to an architecture in which multiplication is performed by Multiplying Digital to Analog Converters (MDACs); addition/subtraction is the result of KCL along the output wire; and neurons are implemented as non-linear I-to-V converters. The overall result is more compact and faster than digital circuitry, but without the noise sensitivity and long-term instability of analog weight storage.

III. JPL HARDWARE

This section outlines the integrated circuit building blocks developed at JPL for hardware artificial neural networks and 3-D parallel data processing architectures. It also outlines some specific 3-D architectures designed to solve real-time spatio-temporal problems.

Neural Network Building Blocks

NN64 Chip:

In our early work, we fabricated a flexible neural network chip in 0.8μm CMOS called the NN64, whose architecture is depicted in Fig. 1. This chip contains:

- 64 voltage inputs ranging from 2.0 V to 3.0 V
- a 64x64 array of 8-bit bipolar synapses (+/-127)
- 64 variable gain neurons
- programmable bypass switches to select the summed current or neuron voltage output

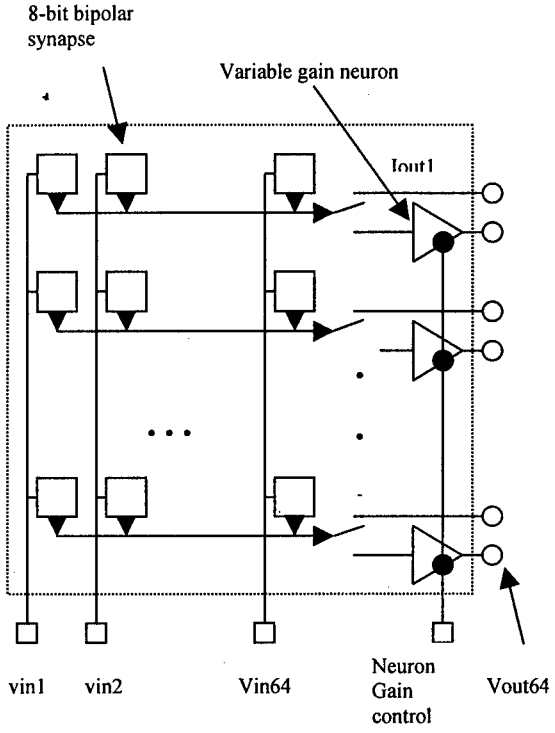


Fig. 1: Block diagram of the NN64 architecture.

The NN64 chip can be used as a basic neural building block in either a feedforward or feedback configuration. It is

potentially expandable horizontally and vertically, allowing for a much larger network to be created if necessary. It can also be connected as if it were stacked in a third dimension, which effectively increases the weight resolution and dynamic range of the network's synapses. Cascading in the third dimension also allows for multiple sub-networks to process the same input data. 3-D architectures are discussed later in the section.

SOICANN Chip:

We recently fabricated a Silicon-on-Insulator Cascadable Artificial Neural Network (SOICANN) using MIT Lincoln Labs' 0.25μm CMOS process, under sponsorship from DARPA's Low Power Electronics Program. Although this chip is not as large as NN64, it was designed to be immediately cascadable without the need for interface circuitry. This allows multiple chips to implement an arbitrarily large feedforward or feedback network. Each chip accepts 8 inputs, has 8 hidden units, has 8 output neurons, and implements a constructive network architecture based on Cascade Error Projection [21-24]. Each hidden unit can be viewed as a single neuron hidden layer with complete connection to all previous hidden layers as well as to all inputs. All neurons are programmable so as to exhibit a logistic transfer function, a gaussian transfer function, or to be bypassed completely. In addition, the output of each neuron can be either voltage or current, making the chip completely cascadable without limitation. SOICANN uses a 1.5V power supply and simulations show an input step response of less than 200 nS through a single chip. As of this writing, the SOICANN die are being shipped back to JPL and have not yet been tested.

3-D Building Blocks

Syn64 Chip:

In [2] and [3], we reported a 64x64 synaptic weight array with 8-bit resolution that was fabricated in 1.0μm AMI CMOS technology. This chip was intended to be a stackable building block for a 3-D architecture. It uses a 5V power supply and requires 64 analog voltage inputs that range from 2.0 to 3.0 volts. These inputs are then multiplied fully in parallel with 64 weight vectors that are stored digitally using an 8-bit bipolar format (+/- 127). The result of each multiplication is a current signal that is summed along one of 64 different lines. The details of this chip can be found in [2].

Column Loading Input Chip:

3-D architectures require large arrays of parallel data as input. To achieve this, the "Column Loading Input Chip" (CLIC) was designed. The CLIC receives a 64x64 array of 8-bit digital data and converts it into a 64x64 analog voltage array in 250ns [5] using a large array of compact digital to analog converters (DACs). The digital input array usually corresponds to an input sub-image of a larger main digital image that is being processed. Inside the CLIC, the sub-image can be shifted up, down, or right one position while a new column or row is loaded from the main image. This allows the sub-window to be moved around inside the main image without having to reload the entire CLIC. The CLIC was fabricated in a 0.8μm HP CMOS process. Its voltage output array is available on 4,096 metal3 pads, each of which measures 66x66μm^2. Each DAC cell in the CLIC array is 101.6x101.6μm^2.

3-D Architectures

Our first cube was created using a vertical stack of sixty-four Syn64 chips

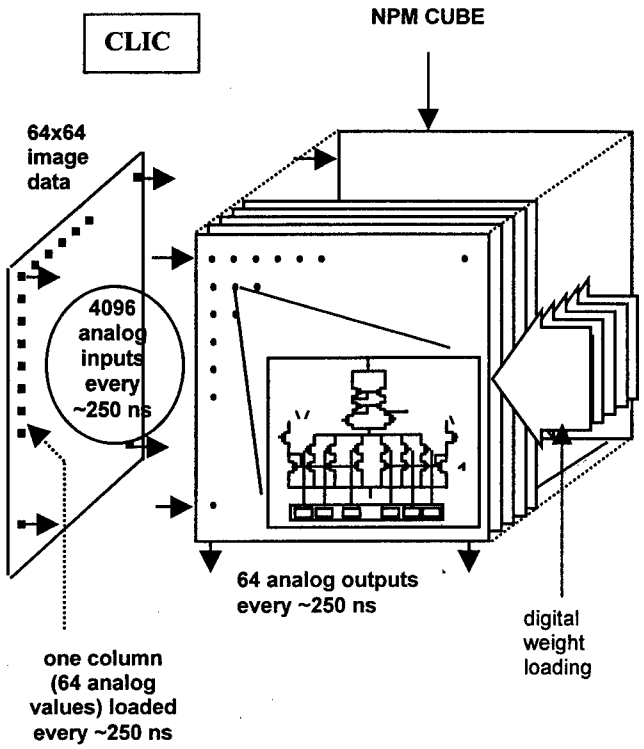


Fig. 2: 3-Dimensional Artificial Neural Network-M (3-DANN-M). In this figure, CLIC provides 64x64 fully parallel analog inputs with a new column (64-bytes) in every 250ns while the NPM performs parallel template matching.

forming a 3-D Neural Processing Module (NPM) intended for massively parallel real-time template matching for spatio-temporal problems[3]. At first an IR focal plane array, which required operation at 77K[4], was mated to the top of the NPM to provide direct parallel analog input. Later the IR focal plane array was replaced with the CLIC in order to exploit the full computational power of the NPM cube with more versatility. Fig. 2 shows a particular implementation called 3-DANN-M where the CLIC obtains a 64x64 sub-window from a 256x256 digital image and sends this sub-image to the NPM cube in a fully parallel fashion. The sub-image is multiplied with sixty-four templates in the cube where each template is a 64x64 array of 8-bit bipolar weights. All multiplications are performed in parallel every 250ns making the cube theoretically capable of 10^{12} operations per second. Fig. 3 shows a photo of the 3-DANN-M.

Current work is focused on combining the Syn64 and the CLIC functionality into a new stackable building block for the next generation 3-DANN-R. This will eliminate the difficult task of bonding the CLIC to the top of the NPM, which greatly simplifies the cube production process while enhancing testability and observability.

Several challenging problems surfaced during the design of the NPM and the CLIC. Specifically, power consumption

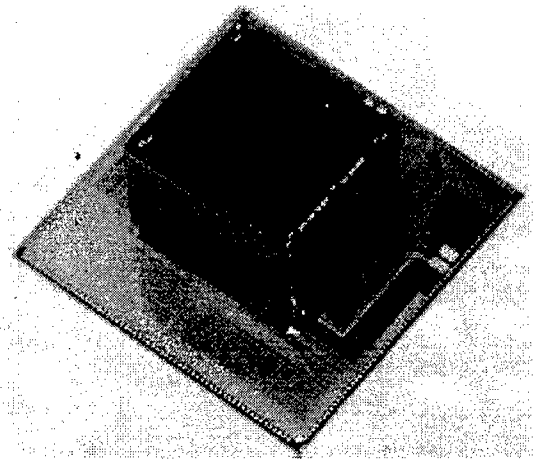


Fig. 3: 3-DANN-M. This photo shows the CLIC on top of the 3-DNPM cube sitting on the motherboard.

and mixed signal noise are so critical that they may prevent us from thinking ahead to larger arrays and bigger chip stacks. Fortunately, Silicon-On-Insulator (SOI) technology is an attractive option that has the potential to neutralize both issues. SOI technology allows us to reduce power consumption drastically by reducing the supply voltage from 5 V to 1.5 V. It also reduces mixed signal noise by eliminating the substrate coupling of digital switching noise to analog components. Since SiO_2 is a good heat conductor it should also

ameliorate thermal management within a 3-D chip structure. The SOICANN chip was designed in SOI in part to evaluate these potential advantages. We have also fabricated Winner-Take-All (WTA) circuits using the same SOI process as SOICANN and the test results are very encouraging [15].

IV. APPLICATION

A lot of interest has recently been generated by research on Hyperspectral Sensor Imaging (HSI), which can be considered as a special case data fusion problem. Real-time classification of hyperspectral data can be extremely useful for certain types of target recognition and terrain or composition identification. In addition, NASA has recently expressed interest in a space-based, low power, miniature system that is capable of classifying hyperspectral data.

The majority of current research on HSI focuses on sub-pixel detection. Unfortunately, the raw sensor data tends to be very noisy and inconsistent which makes the classification problem more difficult. PCA combined with neural networks has already demonstrated some success in sub-pixel detection [20]. Since each pixel contains data from multiple bands, all of which is available in parallel, there is a big advantage to massively parallel processing.

In our application, each pixel contains data from 224 bands of differing wavelengths. In the 3-DANN architecture it takes 4 columns, each containing 64 bands, to process a single pixel. Since neighboring pixels may have relevant information for detecting a particular sub-pixel, a 3x3 window of pixels (see Fig. 4) can be analyzed in parallel, requiring 36 columns of input data. Let the number of desired end members be N, and let $W_1, W_2, \ldots, W_N$ be the orthogonal vectors for PCA or independent vectors for ICA that are to be used for separating the end members. After processing by the 3-DANN cube, the results can be described as follows:

$$Y = \begin{bmatrix} W_1^T \\ \ldots \\ W_N^T \end{bmatrix} X$$

X is an input vector representing one pixel (224x1). This input vector can be physically stored in 4 columns of the CLIC. W_j is a weight vector stored in the columns of 3-DANN. The output vector Y, which is an estimated decoding of the end members, is then sent to the NN64, which can be used as a neural network classifier. This procedure improves detection rates by exploiting the neural network's ability to learn and generalize. Finally a WTA can select the best classification match. Fig. 5 shows the system architecture.

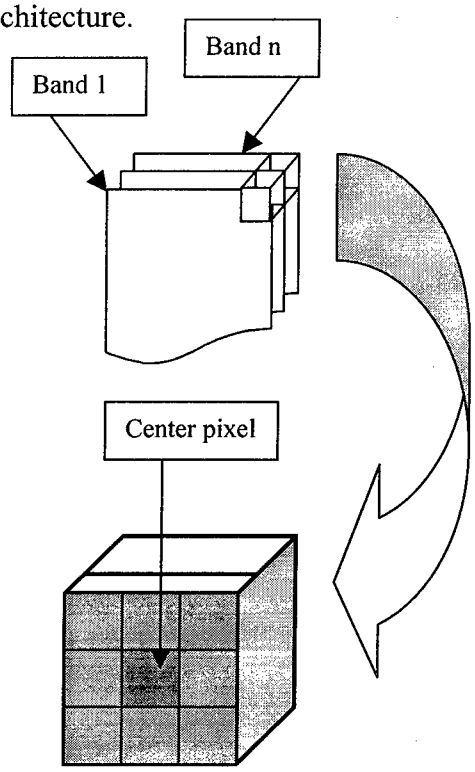


Fig. 4: Structure of hyperspectral data. In this figure, hyperspectral data consists of n=224 bands per pixel. A 3x3 sub-window is analyzed to classify the center pixel.

From hardware designed at JPL, we are able to construct a discrete system for HSI analysis. Even though it is a discrete system, it is still extremely compact and low power in comparison to other state of the art systems capable of performing hyperspectral analysis; *e.g.* banks of Super Harvard Architecture RISC Computer (SHARC) DSP processors [25].

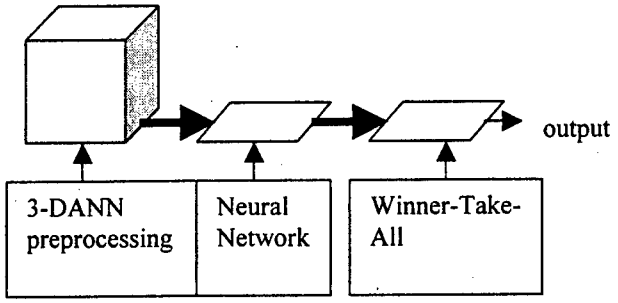


Fig. 5: Full 3-D architecture for real-time HSI sub-pixel classification problem. 3-DANN operates as a linear pre-processor to separate end members, NN64 is the neural network processor to enhance classification, and WTA selects the best match.

V. CONCLUSION

A number of powerful chips developed at JPL for use as building blocks in 3-D systems were presented briefly, along with a description of the 3-D architectures themselves. We also discussed the potential of using SOI technology to overcome two of the most difficult challenges inherent in 3-D chip stacks. Finally, we showed how our 3-D architecture might be applied to solve a hyperspectral sub-pixel classification problem.

Our proposed 3-D architecture is extremely compact and features very high-speed operation with a power consumption of less than 5 Watts. Such a system should satisfy NASA's requirements for high-density, low power, space-based systems capable of synthesizing large amounts of varied sensor data.

Acknowledgments:

The research described herein was performed by the Center for Space Microelectronics Technology, Jet Propulsion Laboratory, California Institute of Technology and was jointly sponsored by the Ballistic Missile Defense Organization/Innovative Science and Technology Office (BMDO/IST), and the National Aeronautics and Space Administration (NASA). The authors would like to thank Drs T. Daud, A. Thakoor and S. Udomkesmalee for useful discussions.

References:

1. T.A. Duong, et al., "*Learning in neural networks: VLSI implementation strategies*," In: *Fuzzy logic and Neural Network Handbook*, Chap. 27, Ed: C.H. Chen, McGraw-Hill, 1996
2. T.A. Duong, et al., "Cascaded VLSI neural network building-block chips for map classification," *Government Microcircuit Applications conference* 92, Las Vegas, NV, Nov 9-12, 1992, pp 45-46.
3. T.A. Duong, S. Kemeny, T. Daud, A. Thakoor, C. Saunders, and J. Carson, "Analog 3-D Neuroprocessor for Fast Frame Focal Plane Image Processing," *The Industrial Electronics Handbook*, Chap. 73, Ed.-In-Chief J. David Irwin, CRC PRESS, 1997.
4. T.A.Duong, et al. "Room and Low temperature performance of high speed neural network circuits," *Electrochemical Society Proceedings*, pp. 369-377, vol. 97-2, May 2-4, 1997, Montreal, Canada.
5. T.A.Duong, et al., "the results of 64x64 Analog Input Array for 3-Dimensional Neural Network Processor," *IJCNN'98* in Anchorage, Alaske, pp. 49-53, 98.
6. R.O. Duda, "Pattern Classification and Scene Analysis," John Wiley & Sons Inc., 1973.
7. A.J. Bell and T.J. Sejnowski, 1995, "An Information maximization approach to blind separation and blind deconvolution," Neural Computation, 7, 1129-1159.

8. Pham, D.,T., 1996, "Blind separation of instantaneous mixture of source via an independent component analysis.", IEEE trans. On Signal Processing, vol 44, pp2768-2779.
9. Attias and Schreiner, 1998, "Blind source seperation and deconvolution.", Neural Computing, vol 10, number 6
10. Hyvarinen and Oja, 1997, "A fast fixed point algorithm for independent component analysis.", Neural Computation, vol 9, pp 1483-1492.
11. D. Hammerstrom, "A Massive Parallel Architecture for Cost-Effective Neural Network Pattern Recognition, Image Processing, and Signal Processing Applications," Digest paper in GOMAC Conf. , pp 281-284, 1992, Las Vegas, Neveda.
12. Eberhardt S.P., T.A. Duong, and A.P. Thakoor, "A VLSI analog synapse 'building-block' chip for hardware neural network implementations," *Proc. 3rd Annual Symposium on Parallel Processing*, Fullerton, CA, March 1989, Vol. 2 pp. 257-267.
13. D.A. Kern, Experiments In Very Large-Scale Analog Computation, Ph.D. Thesis California Institute of Technology, 1992
14. J. Hopfield, "The effectiveness of analogue 'neural network' hardware," *Network* 1:27-40 (1990) (IOP Publ. Ltd., U.K.).
15. T.A.Duong, et al., "Winner/Loser-Take-All On SOI Technology for Neural Networks", SPIE, Orlando Florida, May 1998.
16. W.-C. Fang, B.J. Sheu, O.T.-C. Chen, and J. Choi, "A VLSI Neural Processor for Image Data Compression Using Self-Organizing Netorks," *IEEE Transactions on Neural Networks*, vol. 3, no. 3, pp.506-518, 1992.
17. A. Jayakumar and J. Alspector, "On-Chip Learning in Analog VLSI Using Simulated Anealing," *Digest of Papers in Government Microcircuit Applications Conference*, Las Vegas, NV, 1992, pp.277-280.
18. S. P. Eberhardt, T.A. Duong, R. Tawel, F.J. Pineda, and A.P. Thakoor, "A Robotic Inverse Kinematics Problem Implemented on Neural Network Hardware with Gradient-Descent Learning," *Proceedings of the Second ISTED International Symposium on Expert Systems and Neural Networks*, M.H. Hamza, ed., Hawaii, August 15-17, 1990, pp.70-73.
19. M. Holler, S. Tam, H. Castro, and R. Benson, "An Electrically Trainable Artificial Neural Network (ETANN) with 10,240 "Floating Gate" Synapses," *Proceedings IEEE International Joint Conference on Neural Networks*, vol. 2, June 18-22, Washington, 1989, pp.191-196.
20. A. Howard, C. Padgett, and K. Brown "Intelligent Target Detection in Hyperspectral Imagery," *Thirteenth International Conference on Applied Geologic Remote Sensing*, Vancouver, British Columbia, Canada, 1-3 March 1999.
21. T.A. Duong, A. Stubberud, T. Daud, and A. Thakoor, "Cascade Error Projection-A New Learning Algorithm," Proceedings Int'l IEEE/ICNN in Washington D.C., vol. 1, pp. 229-234, Jun. 3-Jun 7, 1996.
22. T.A. Duong, "Cascade Error Projection-An efficient hardware learning algorithm," Proceeding Int'l IEEE/ICNN in Perth, Western Australia, vol. 1, pp. 175-178, Oct. 27-Dec 1, 1995(Invited Paper).
23. T.A. Duong and T. Daud, "Cascade Error Projection With Low Bit Weight Quantization For High Order Correlation Data", Accepted to IJCNN'99 in washington D.C., 1999.
24. T.A. Duong and T. Daud, "Cascade Error Projection_a Learning algorithm for hardware implementation", Accepted to IWANN'99 in Alicante, Spain.
25. Analog Promotion SHARC2 Overview, http://www/analog.com/new/ads/html/SHARC2/started.html.

Author Index