WHOI-99-17

Woods Hole Oceanographic Institution

Electronic Still Camera Processing and Mosaicking

by

Jonathan C. Howland
Steven Lerner

December 23, 1999

Technical Report

Funding was provided by the Office of Naval Research under Contract No. N00014-95-1-1237 and by the National Science Foundation under Grant No. OCE-9627160.

WHOI-99-17

Electronic Still Camera Processing and Mosaicking

by

Jonathan C. Howland
Steven Lerner

Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543

December 23, 1999

Technical Report

Funding was provided by the Office of Naval Research under Contract No. N00014-95-1-1237 and by the National Science Foundation under Grant No. OCE-9627160.

Approved for Distribution:

[illegible]

Timothy K. Stanton, Chair

Department of Applied Ocean Physics and Engineering

TABLE OF CONTENTS

1. Introduction 4

2. Electronic Still Camera Collection and Processing 5

2.1 The Camera System 5

2.2 Collection 6

2.3 Processing 7

2.3.1. Tape Format *8*

2.3.2. Extraction of the Data from the Tape *8*

2.3.3. Image Normalization *9*

2.3.4. Histogram Specification *11*

2.3.5. Image Geolocation *12*

3. Image Archiving and Presentation 14

3.1 Image Archiving 14

4. Digital Mosaicking 15

4.1 Image Selection 15

4.2 Image Registration 16

4.3 Cut-Line selection 17

4.4 Mosaic Development 18

4.5 Mosaicking Strategies 19

4.5.1. Semi-Controlled Mosaics *19*

4.5.2. Strip And Patch Based Mosaicking *20*

4.6 The Uses of Mosaics 20

5. The Future of the Imaging Pipeline 23

6. References 24

7. Appendix A: MIS Data Formats 26

TABLE OF FIGURES

Figure 1: ESC mounted on Jason 5

Figure 2 Front of ESC Deck Box 6

Figure 3 Back of ESC Deck Box 6

Figure 4 ESC Processing System 7

Figure 5: ESC Bias Image 10

Figure 6 Normalized ESC image 11

Figure 7 Image After Adaptive Histogram Specification 12

Figure 8 Portion of Image Coverage Chart in the Visual Program. 13

Figure 9: Thumbs Plus Screen 16

Figure 10 I_RAS C Registration Screen 17

Figure 11: I_RAS C Cut-Line Selection 18

Figure 12 Mosaic of Wing Tank, Derbyshire Survey. 19

Figure 13 Archaeological Site Before and After Artifact Recovery ... 21

Figure 14: Geological Mosaic from Lucky Strike Area 21

Figure 15: Use of Visual with mosaics and original data. 22

ELECTRONIC STILL CAMERA PROCESSING AND MOSAICKING

THE DEEP SUBMERGENCE LABORATORY/DEEP SUBMERGENCE GROUP PIPELINE

1. INTRODUCTION

Since 1990, The Deep Submergence Laboratory and Deep Submergence Group of the Woods Hole Oceanographic Institution have been collecting large quantities of digital imagery and creating digital photomosaics of the sea floor. Initially, the digital image collection, processing, and mosaicking processes were all highly specialized "one of a kind" efforts. Over the past decade, this process has been refined, standardized, and made into a robust "pipeline." The collection, processing, and mosaicking can be reliably carried out on a routine basis. Deep Submergence Group personnel perform collection, processing and archiving, while science party users can be trained in mosaicking. This report will describe the pipeline, yielding insight into the evolution and purposes of each step.

Development of imaging and mosaicking efforts continues at the Deep Submergence Laboratory. The current mosaicking research thrust includes two main components, automation and quantification. The automation effort is attempting to make the process of mosaicking less consuming of manpower. The quantification effort is attempting to turn our mosaic products from being non-scaled non-metric pictures to accurate quantifiable representations of the sea floor (Singh, et al, 1998). This report will not describe those research efforts; instead it will concentrate on describing the results of the past ten years of development of the processing and manual mosaicking systems.

It is expected that the cameras, processing systems, and mosaicking tools will all continue to evolve as new imaging and computing systems become available. This report presents the state of the processing pipeline in late-1999.

The processing and mosaicking pipeline that will be described has been effectively used with data from both the Argo II and Jason vehicles, as well as with data from the U.S. Navy's NR-1 research submarine. With minor modification, it has also been used to mosaic video snapshots from the Autonomous Benthic Explorer, or ABE (Yoerger, et al, 1999).

2. ELECTRONIC STILL CAMERA COLLECTION AND PROCESSING

The mosaicking process can utilize a wide variety of formats of digital imagery from virtually any source. However, the primary sensor used during most Deep Submergence Group operations has been the Marine Imaging Systems (now Imetrix) Model 9100 Electronic Still Camera (ESC). It has been mounted on Argo II, a towed high altitude imaging platform, and on Jason, a Remotely Operated Vehicle (ROV). (Ballard, 1993). The ESC has also been used on the Alvin manned submersible. A similar camera system is used on NR-1, and has been used on other Navy deep submergence assets.

2.1 THE CAMERA SYSTEM

The first electronic still camera for underwater use was developed at the Deep Submergence Laboratory in the late 1980's (Harris, et al). The camera system was updated and commercialized by Marine Imaging Systems, Inc., and has been used extensively by the Deep Submergence Laboratory since 1989.

Figure 1 shows the ESC mounted on Jason in a down looking configuration. The camera has been mounted on both Argo and Jason in a variety of ways, including forward, down, and obliquely, in successful efforts to optimize imaging geometry. Lighting is provided by 2 300 watt second strobes on Jason and 4 600 watt second strobes on Argo.

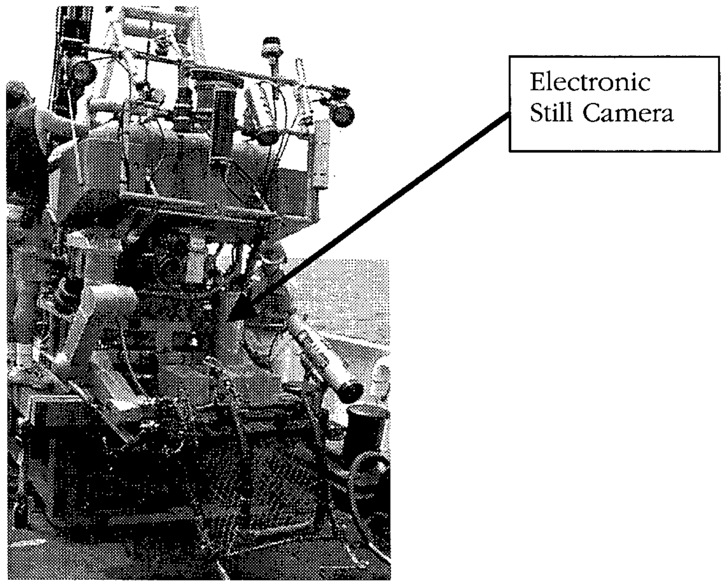


Figure 1: ESC mounted on Jason

The camera has a dynamic range of 14 bits, which allows recording of 16384 separate gray shades. In practice, this dynamic range is rarely seen in individual images, but allows the camera to function well under a wide variety of lighting conditions. The camera spatial resolution is 584x376 pixels.

The ESC is a fully digital device, which uses a high speed RS-422 telemetry system (run over fiber optics) to send camera data to a topside deck box. Figures 2 and 3 show the front and back of the deck box, which is used for camera control and data display and recording. Upon receipt of the image data, the deckbox writes the raw data to exabyte tape, performs a global histogram equalization and conversion to eight bits, and produces a video resolution representation of the image. This video output is an input to the video routing system in the Argo/Jason control van, and can be displayed in multiple locations in the van and elsewhere on the ship.

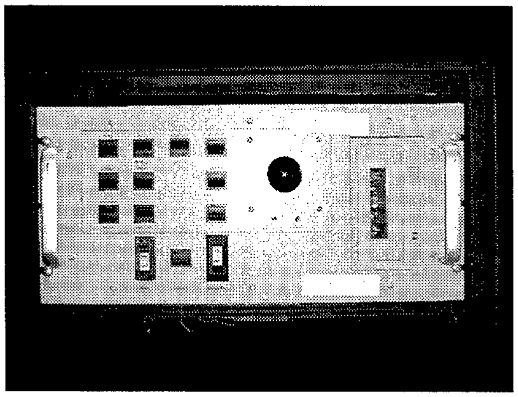


Figure 2 Front of ESC Deck Box

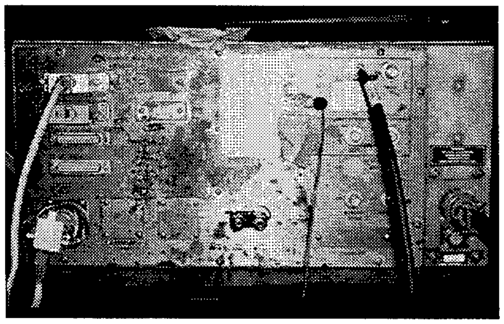


Figure 3 Back of ESC Deck Box

2.2 COLLECTION

Before beginning an electronic still camera survey, the system clock on the deck box is manually set to the time base being used in the control van. The clock is allowed to free-run after that point. During extended operations (more than a week) the clock is frequently reset to minimize drift.

In normal practice, the camera is run in fully automatic mode, and is set to expose and record imagery as quickly as possible. This is approximately one image every 13 seconds if data is being recorded to exabyte 8mm tape. Each image is approximately 0.5 megabytes in size. Exabyte tapes are normally replaced every four hours (approximately 1000 images), at the normal watch change. (Changing a tape forces a several minute gap in data recording while the tape rewinds and retensions.) Note that at full recording rates, each exabyte tape could theoretically hold sixteen hours worth of image data. However, prudence dictates minimizing the amount of data recorded to a single piece of fragile media, and the watch change makes an easy reminder to change the ESC tape. Naturally, if a key section of the sea floor is being imaged at the normal tape change time, the tape is not changed until the vehicle moves out of the area.

Data has been collected in a variety of operational patterns. The most success has been obtained when performing a dedicated ESC survey, optimizing vehicle motion to obtain high quality imagery suitable for mosaicking. See (Kelley, et al, 1999) for an example. However, the camera is usually run in automatic mode, and recording is rarely turned off while the sea floor is visible. "Incidentally" collected imagery has proven useful on many occasions.

2.3 PROCESSING

The current ESC processing system is based upon a pair of Sun workstations. However, the processing system has also been ported to Linux, and should work under any Unix-based operating system equipped with an exabyte tape drive and sufficient disk space. Figure 4 shows the processing system that has been placed in a flight case for easy deployment.

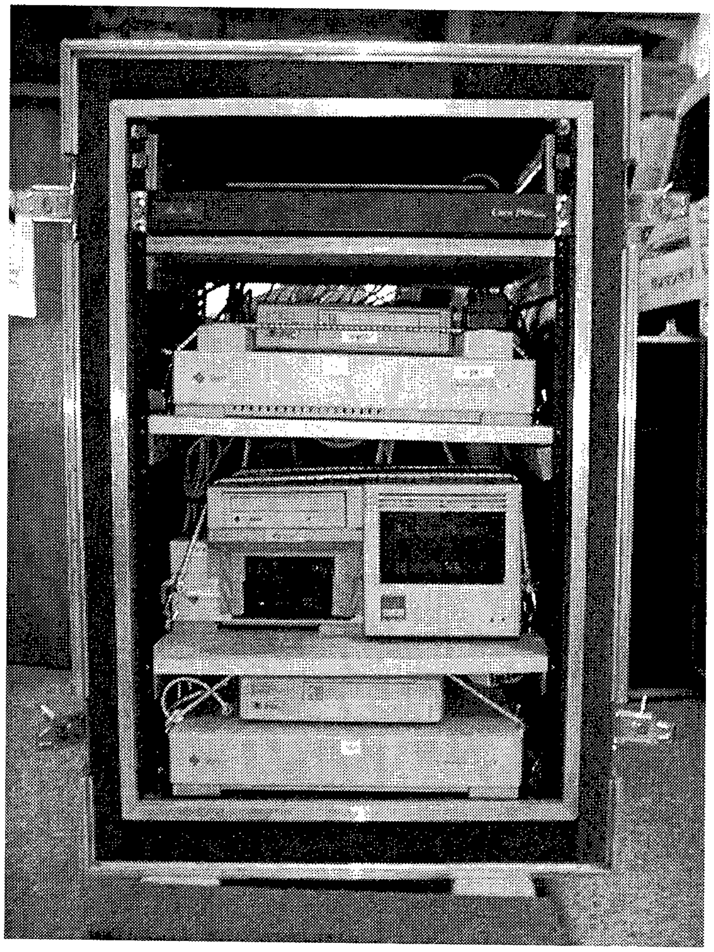
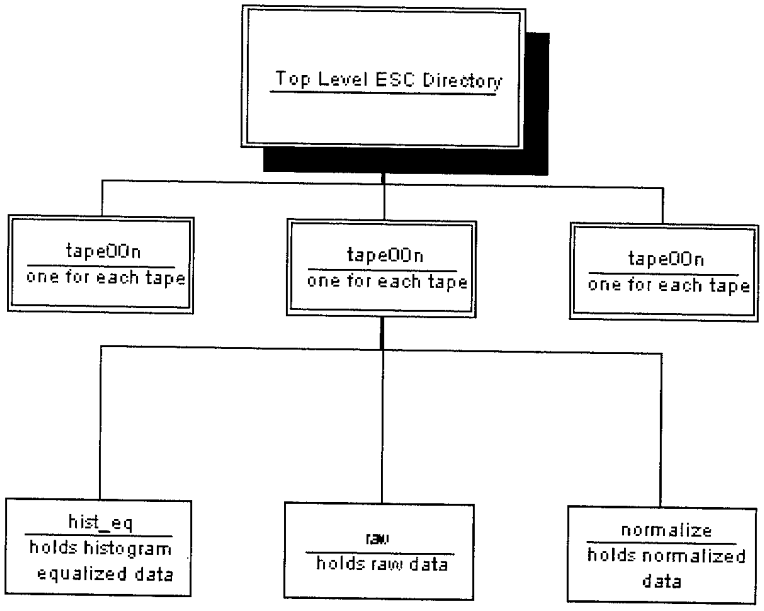


Figure 4 ESC Processing System

Before cruise data processing begins, a system of data directories is set up, matching the following structure:



The data processor begins the processing stream by inserting the new tape into an exabyte drive and typing "process_esc *tape_name*", where *tape_name* is an arbitrary unique designator for the tape. It usually follows a "one-up" increasing format, such as *tape001*, *tape002*, etc, but can be any ASCII string. A new directory named after the tape is set up in the previously described structure. It is then time to read data off of the tape, which requires an understanding of its format.

2.3.1. TAPE FORMAT

The deck box writes the image data onto the tape in a series of files. Each file contains fifty images, or less if it is the last file on the tape. A file marker is written onto the tape between each block of fifty images.

Each image is prepared in a custom ".mis" format, which is documented in Appendix A. 512 binary zeroes are written to tape, and then the image file is written in a Unix tar format. This structure requires following a relatively unwieldy procedure when reading the tape. Note that code (Unix scripts and C code) for the processing steps described are available from the authors upon request.

2.3.2. EXTRACTION OF THE DATA FROM THE TAPE

First, the files of fifty images are read from the tape and concatenated into a single large file, named *tapename*.raw. This file is read by successively stepping through it, one image at a time, skipping the blocks of binary zeroes. The resulting blocks of data are passed through the "tar" command and the resulting file (which has a name of the structure *imxxxxx*, where *xxxxx* is a successively increasing number starting at zero) extracted. Since the successive file names are identical on each tape, they are changed to reflect the time of imaging. An ASCII date-time string is extracted from each image, and a new file name created. The file name includes all of the information necessary to keep each image separate from any other, and matches the following pattern:

ESC.YYMMDD_HHMMSS.IMNUM.mis

where YY is the last two digits of the year, MM is the month, DD is the day of the month, HH is the hour, following a 24 hour system, MM is the minute, and SS is the second. IMNUM is the sequential order of the image on the tape. mis indicates that this is a Marine Imaging Systems format file. This same file name pattern is followed throughout the processing, with different format type suffixes indicating different stages of processing and image formats.

A new image time file is also created, matching the image name and its time. This file will be used later to produce files of image name merged with position and attitude. Note that if the time in the deckbox is erroneous, it is *not* written into the image header and the times file (and all subsequent products) will be incorrectly created.

When this processing step is completed normally, the result is a directory called "raw" which is filled with all of the images which were found on the tape. *tapename*.raw is removed. If a tape error occurs while reading a block of fifty images, it is possible to rewind the tape, skip forward the number of files necessary to bypass the error, and restart the extraction and processing with a new tape number. After the completion of the processing scheme, the two directories (one before and one after the error) are put together manually.

2.3.3. IMAGE NORMALIZATION

The raw images are in the .mis format file described in Reference A. Very few commercially available and supported desktop applications are able to read these format files in any simple way. This is due to both their custom format and to the fact that they contain sixteen bit data. (It was stated earlier that the MIS ESC produces fourteen bit data. However, the data is padded with zeroes to sixteen bits before storage, making the storage format slightly less compact but far easier to read on a computer.). Furthermore, the raw format files are still somewhat "contaminated" with artifacts caused by chip manufacturing techniques and by other flaws in the chip. A single piece of software is used at this step in the pipeline to solve several problems at once:

- Conversion of 16 to 8 bits, making use of the data easy in off the shelf software
- Correction for systematic patterns in image exposure
- Conversion to a standard image format.

The straightforward way to convert sixteen bits to eight is to re-map the histogram of the sixteen bit data so that it falls into 256 (eight-bit) values. However, two potentially important steps are missed if this practice is followed.

All CCD chips have flaws that show up as artifacts on the image. In particular, the chip used in the MIS camera was fabricated in a two-stage process, and the chip appears to have two distinct left/right halves. Figure 5 shows what is called a bias image. It represents what the chip reports as data with no exposure to light, and is frequently called a dark-current image (Newberry, 1995). The two-half effect can be clearly seen. Also seen is a distinct vertical striping, which is related to the manner in which the charge is read out of the CCD. The majority of these effects are removed by subtracting a bias image from each raw image before further processing. (Note that the gray scale ranges of the bias image shown if Figure 5 have been stretched so as to make them visible for this report. The actual image, if shown unchanged, would appear almost uniformly black due to the very low pixel values recorded in bias images.)

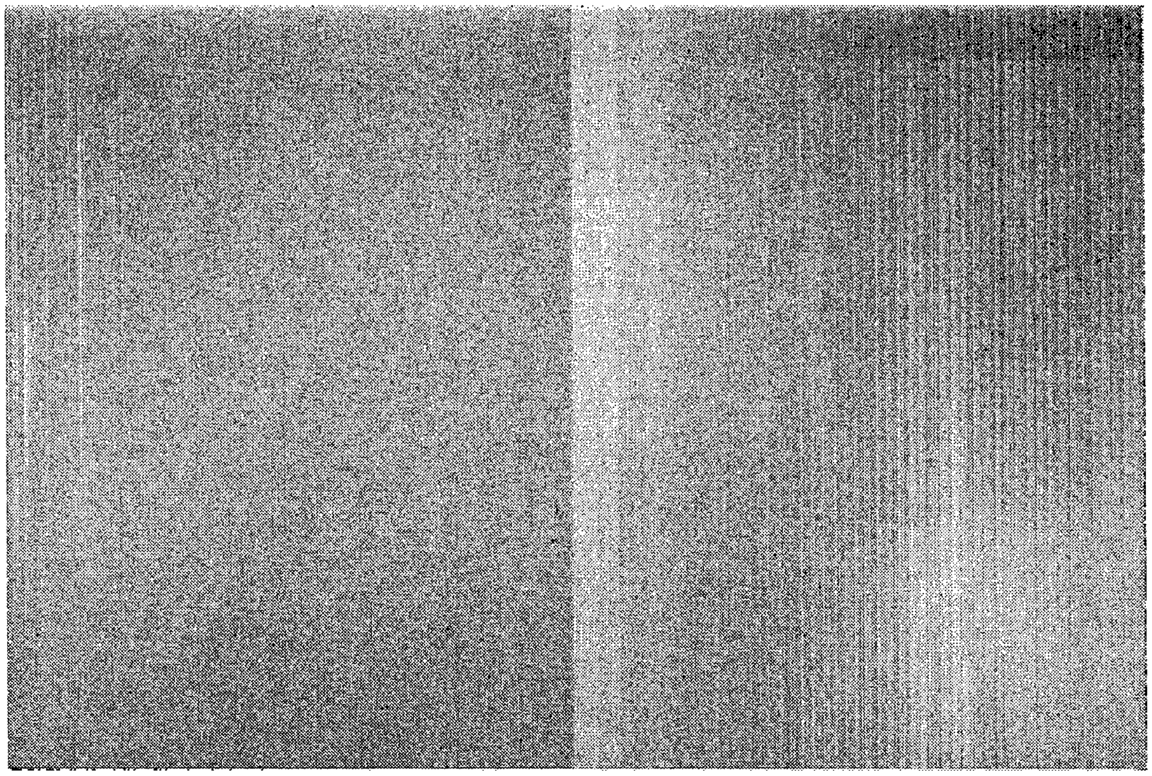


Figure 5: ESC Bias Image

Flat fielding is an essential step in calibrating a raw CCD image. It is necessary because a given intensity of light does not product an identical response in every pixel of a CCD array (Chromey, 1996). Variations occur due to sensitivity differences among pixels and the unique characteristics of the optical path (among other causes). The unwanted variations are removed by dividing a raw image by the flat field frame. The flat field is obtained by exposing the CCD chip to a range of gray light fields, and picking the one that most closely matches the exposures obtained by the images being adjusted.

These two processes are combined in a step called normalization. The output is an eight-bit Sun raster file that accurately represents the scene imaged by the camera, having very little artifacts induced by the camera. The normalized images are distinguished by a suffix of ".rf" (a customary file suffix for a Sun raster file). They are maintained in a directory called "normalize.

After the normalized files are created, the raw .mis files are compressed using the standard Unix compress command. Compressed files are indicated by a ".Z" suffix.

Figure 6 shows a normalized ESC image from the 1997 Derbyshire Survey (Howland, 1999a).

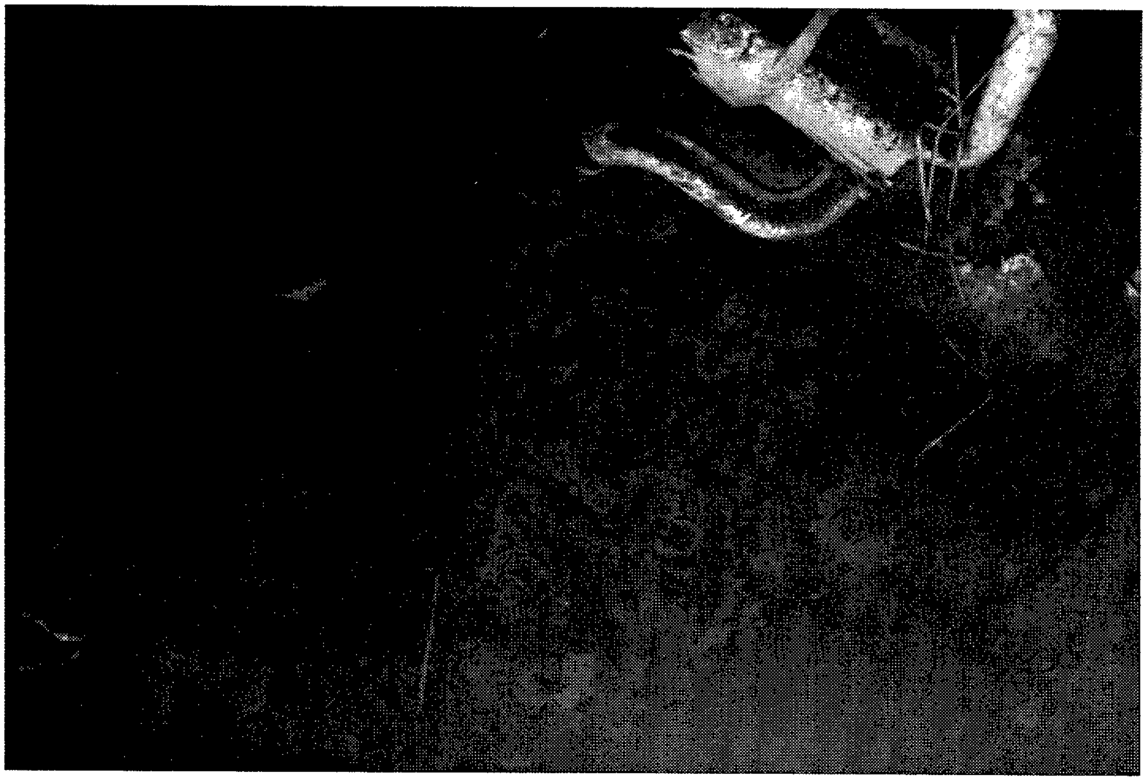


Figure 6 Normalized ESC image

2.3.4. HISTOGRAM SPECIFICATION

The next step in ESC processing is adaptive histogram specification. Underwater imagery is usually characterized by uneven illumination. Shadows can be quite useful in interpretation, since they reveal differences in height and aspect. Illumination falloff, however, shows only the physics of light propagation in the water: the farther light travels, the more it is attenuated and lost. Furthermore, the uneven illumination causes mottled mosaics, and detracts greatly from a finished mosaic product. The eyes tend to be drawn to the imperfections and apparent edges caused by illumination differences, rather than freely viewing the entire mosaic. A processing step that manipulates the gray scale values in an image to produce apparently even illumination is invaluable in adding both interpretation and mosaicking.

WHOI has been using adaptive histogram equalization, and its cousin, adaptive histogram specification for many years to successfully meet this need. Basically, each image is divided into contiguous small blocks of pixels. The histogram of each block is then calculated and either equalized or passed through the transfer function of the desired distribution. Then for each pixel of the input image, a weighted average of the transferred histograms of the surrounding blocks is calculated, and a new pixel value computed (Pizer et al, 1987).

Underwater surveys typically collect many thousands of images. It is entirely impractical to individually process each image, so a batch method of processing is used during histogram specification. Parameters are based upon DSL experience and upon imagery collected during the first several thousand ESC images; these parameters are then used to process the entire data set.

Custom software written at the Deep Submergence Laboratory was used for this processing step. The code is an adaptation of software that appeared in Graphics Gems IV (Heckbert, et al, 1994), a compendium of algorithms and techniques. The adaptations deal primarily with use of histogram specification vice equalization, and with various I/O parameters. The output of the processing is either an eight-bit sun raster file or an eight-bit tiff file, depending upon whether the new or old version of the code is being used. There is little difference in the new versus the old code other than output format.

Figure 7 shows the previously normalized image after adaptive histogram specification.

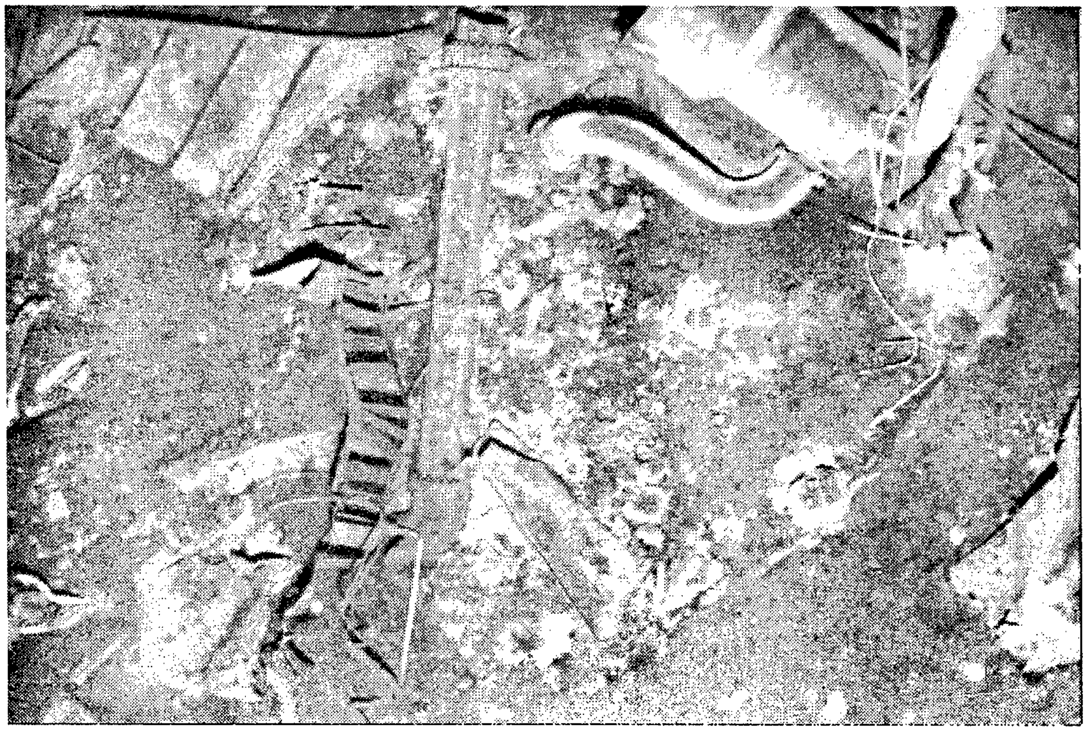


Figure 7 Image After Adaptive Histogram Specification

2.3.5. IMAGE GEOLOCATION

Imager geolocation is possible if time synchronization of image collection and vehicle navigation and attitude data is maintained. In the earliest stages of data extraction from the tape, it was mentioned that a file of image times was created. Once navigation and attitude are processed for a particular day or lowering, these data are interpolated into the image times file to produce records of vehicle position and attitude at the time of imaging. These records can be used to produce image coverage maps like Figure 8, and are exported into geo-analysis software such as Visual and the GeoBrowser (see Section 3 and (Lerner and Maffei, 2000)).

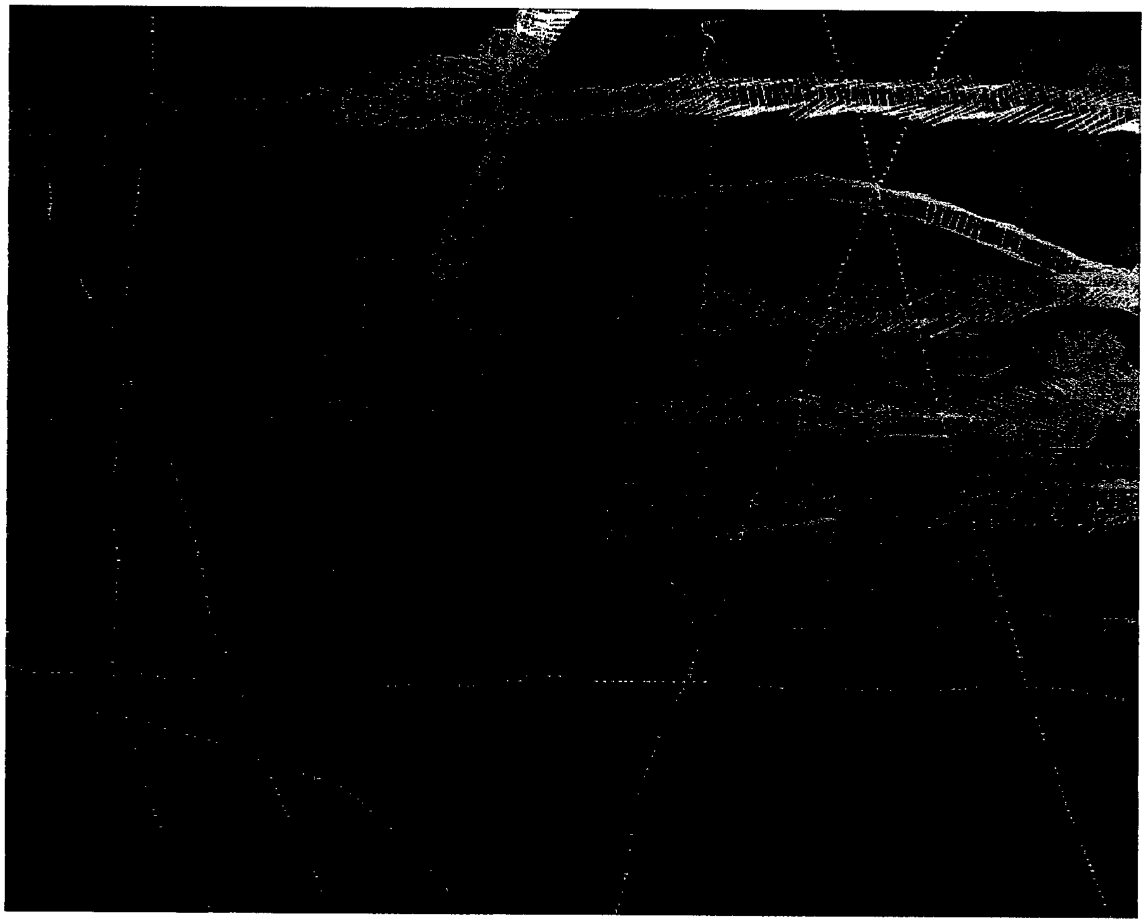


Figure 8 Portion of Image Coverage Chart in the Visual Program.

3. IMAGE ARCHIVING AND PRESENTATION

The results of each of the processing steps just described are useful in different tasks. The raw .mis images would be useful in doing any photometric studies, such as those detailed in (VanDover, et al, 1994). Additionally, they should be saved in any case since they are an unprocessed backup of the original data.

The normalized images are a quite useful representation of the sea floor, since they retain all of the shadowing and lighting falloff effects that many observers have grown accustomed to in underwater image analysis. Any efforts that attempt to derive shape from shading or from shadowing should use the normalized images. They have also been successfully used for mosaicking when lighting conditions were satisfactory and altitudes were quite low.

The histogram-specified imagery is probably best for detailed examination of the entire image, for scientific study, and for most mosaicking. Geometry specific intensity variations and lighting falloff are minimized, and some observability into shadows is apparent.

Since all of the image types are potentially useful, all are archived.

3.1 IMAGE ARCHIVING

As an ESC survey progresses, tapes are processed and data is written to a user-accessible Network File System (NFS) Unix directory. As the number of tapes processed increases and disk space becomes limited, collections of raw imagery tapes (usually five at a time) are written (using a tar format) to exabyte tapes. These archive tar tapes contain the raw, the normalized, and the histogram-specified imagery. Two copies are made of each tape, one for the Science party and one for the WHOI archive. The raw and normalized images are then removed. The histogram specified imagery is typically copied to a Windows NT system for use in mosaicking, and eventually removed altogether. The imagery on the Windows NT system is usually backed up to either Jaz disks or to CD-RW disks.

A variety of means are used to make the data available to the Science Party during a cruise. Typically, the scientists are given free read access to the imagery once it is processed. On certain cruises, more elaborate methods of data access, typically using WWW browsers have been implemented. For examples, see (Fornari, 1996) and (Lerner, 1998).

4. DIGITAL MOSAICKING

The technical challenges, geometric and otherwise, to successful mosaicking of underwater imagery are detailed in (Howland et al, 1999), which also describes the effort to develop mosaicking capabilities at the Deep Submergence Laboratory. A variety of commercial off the shelf software packages can be used for digital mosaicking on standard PC hardware once the images to be mosaicked have been processed and selected. Several of these packages have been evaluated at DSL, and we are currently using IRAS C from Intergraph (www.intergraph.com). Use of IRAS C requires a Microstation license from Bentley, Inc (www.bentley.com). GCP-Works from PCI (www.pci.on.ca) has also been evaluated and used in successful mosaicking efforts. We have experienced some success at combining a commercial automated mosaicking package (VideoBrush Photographer, www.videobrush.com) with IRAS-C.

Mosaicking using IRAS-C requires a Windows NT workstation with access to the ESC imagery. Typically, the image data is transferred to the NT workstation using an FTP client. Experiments with Windows NT Network File System (NFS) implementations have been made, but have not been found to be robust to the idiosyncrasies of shipboard networks. As the Deep Submergence Group processing system moves to Linux, it is expected that Samba will be evaluated in an attempt to minimize manual file transfers.

In typical DSG operations, the science party performs mosaicking, with assistance and training from DSG personnel.

4.1 IMAGE SELECTION

A variety of techniques have been used for image selection. On major mosaicking efforts, such as that performed during the 1994 TAG cruise (Sulanowska, et al, 1996) or the 1997 Derbyshire survey (Howland, 1999), the Visual system (Lerner, 1999) was used to provide spatial access to image information. (The images had first to be geolocated, as described in Section 3.2.5) A search capability in the 4D GeoBrowser (Lerner and Maffei, 2000) was also used in the Derbyshire effort. However, for most mosaicking efforts, a shareware thumbnail browser (Thumbs Plus, www.cerious.com) is used for access to the imagery. It allows rapid playthrough of the image data, as well as customizable pagesize views of images in a Windows file system. Figure 9 shows a screen shot of a Thumbs window, in this case being used with data from the M.V. Derbyshire Survey (Howland, 1999a).

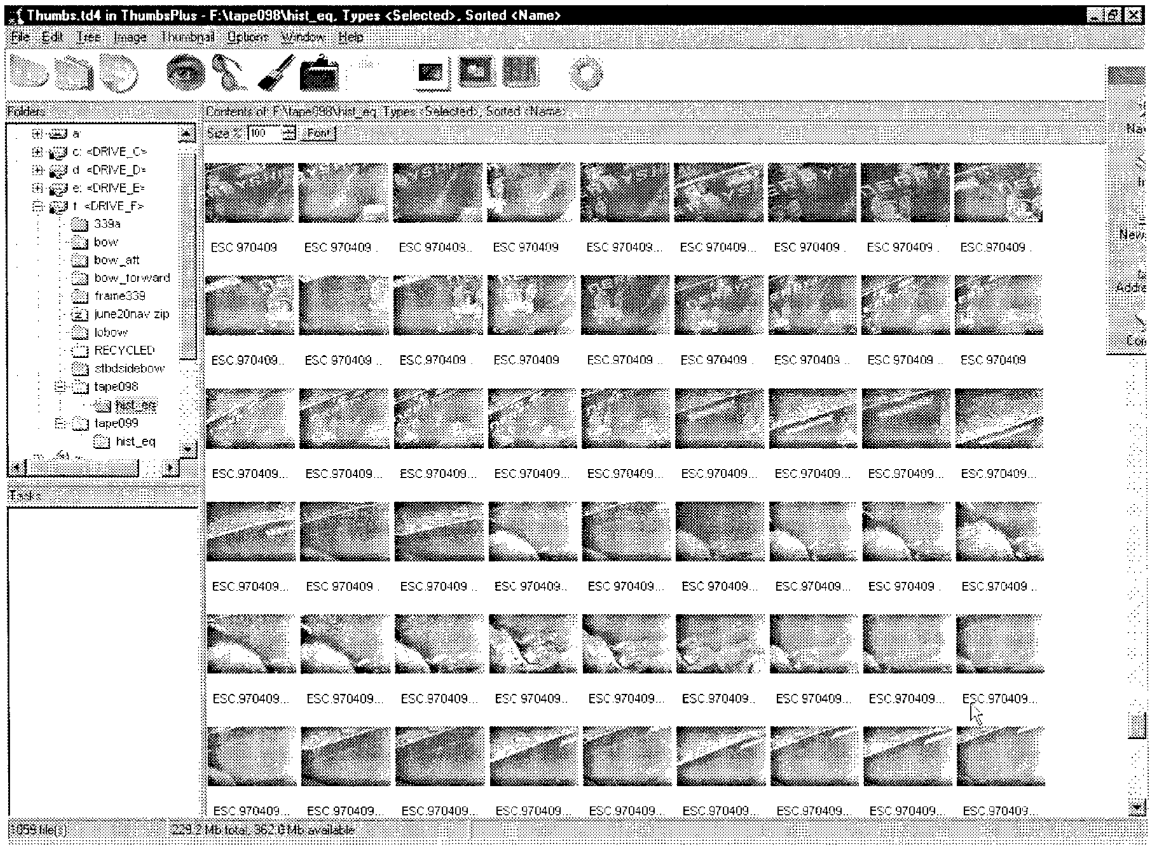

Thumbs.td4 in ThumbsPlus - F:\tape098\hist_eq, Types <Selected>, Sorted <Name>
File Edit Tree Image Thumbnail Options Window Help
Folders
Contents of F:\tape098\hist_eq, Types <Selected>, Sorted <Name>
a:
c: <DRIVE_C>
d: <DRIVE_D>
e: <DRIVE_E>
f: <DRIVE_F>
339a
bow
bow_aft
bow_forward
frame339
june20nav.zip
lobow
RECYCLED
stbdsidebow
tape098
hist_eq
tape099
hist_eq
Tasks
ESC 970409...
1059 file(s)
229.2 Mb total, 362.0 Mb available

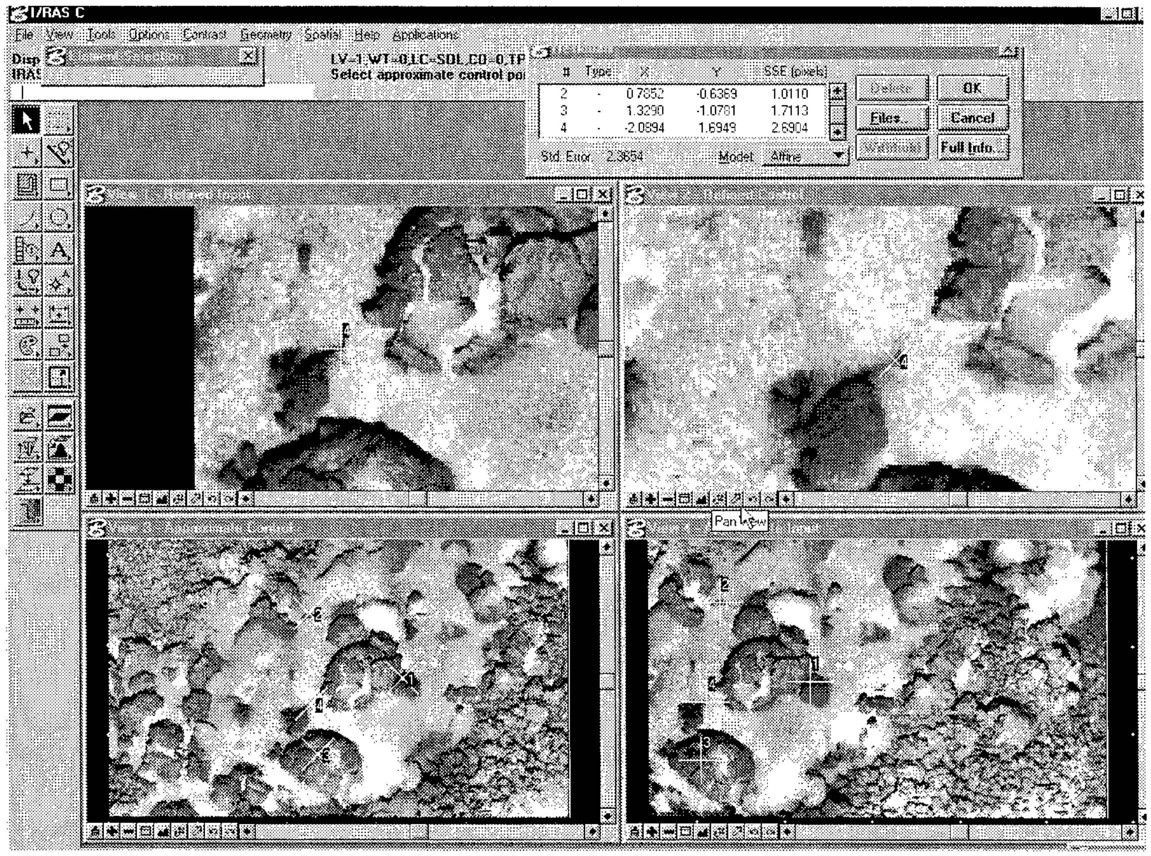
Figure 9: Thumbs Plus Screen

4.2 IMAGE REGISTRATION

In the IRAS software system, images are registered to each other and to an existing mosaic) using a variety of warps, including helmert, affine, projective, finite element, and high-order polynomials. Helmert (a single scale change and a rotation) and affine (scale changes in two directions, plus a rotation) are most often used. Figure 10 shows an image registration screen layout in use with data from the 1994 TAG survey (Kleinrock, et al, 1996).

Note that *no* image warp is inherently superior to any other. They are all physically unrealistic; all that can be hoped for using this methodology of mosaicking is a decent approximation and minimal distortion. In practice, several warps are tested, and the one that yields the best visual fit and the minimum RMS residual after a least squares fit is most frequently used.

After the new image is registered, it is placed into the target image or mosaic using the IRAS mosaic tool. Choices are made as to which image is placed "on top" of the other and a cut-line is selected.


I/RAS C
File View Tools Options Contrast Geometry Spatial Help Applications
Disp
IRAS
Delete
OK
Files...
Cancel
Full Info...
Pan View

LV=1,WT=0,LC=SDL,CO=0,TP
Select approximate control poi

#	Type	X	Y	SSE (pixels)
2	-	0.7852	-0.6369	1.0110
3	-	1.3290	-1.0781	1.7113
4	-	-2.0894	1.6949	2.6904

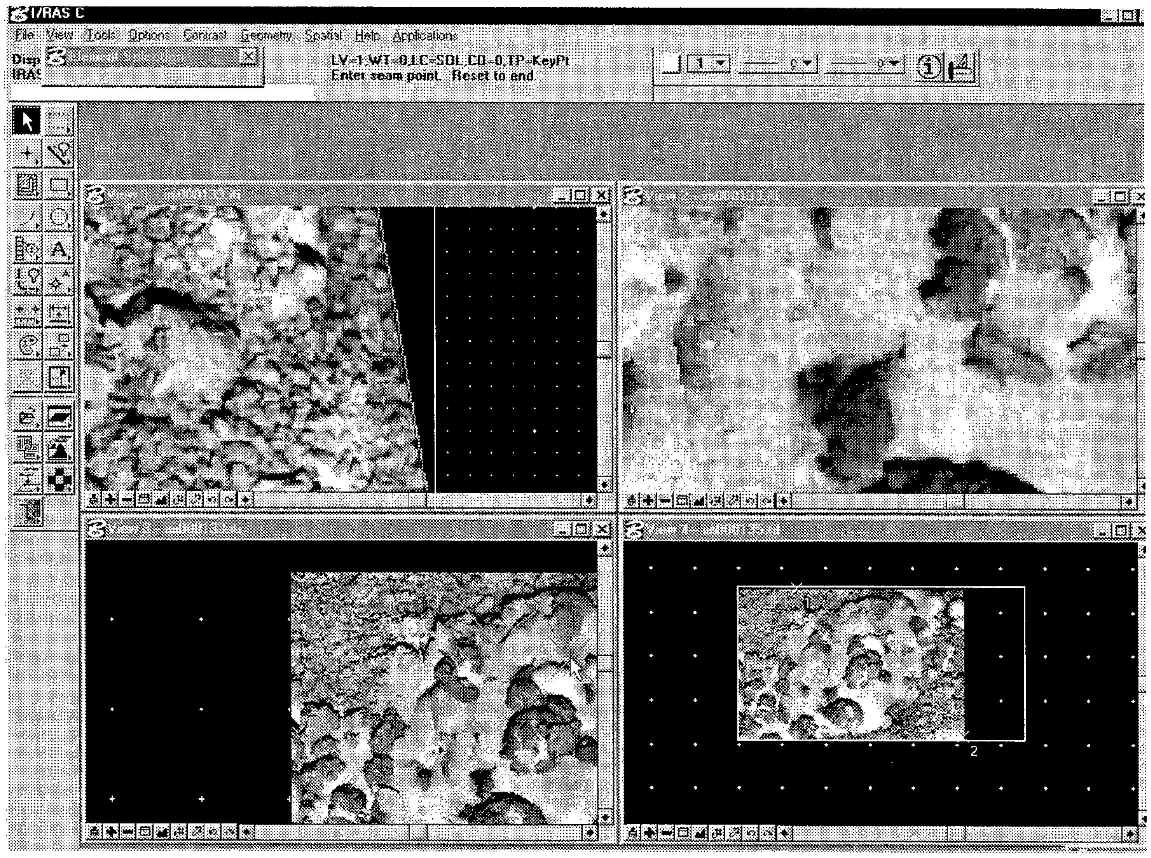
Std. Error: 2.3654 Model: Affine

Figure 10 I_RAS C Registration Screen

4.3 CUT-LINE SELECTION

Image cutlines, the digital equivalent of the film edge in manual hardcopy mosaicking, are traced using manual point input. Blending can be performed across these cutlines to minimize visible edges. In practice, the most success at avoiding edges is found by tracing cutlines along the natural edges in the image scene, such as rock outcrops or bottom texture changes. Figure 11 shows a cutline being traced in this way.

Earlier versions of DSL mosaicking software used an alpha matte "air-brush" instead of a cut-line. This approach is probably superior, producing more seamless mosaics. Numerous automated systems exist for automated seam generation in mosaics (*e.g.*, Shiren et al, 1989).


I/RAS C
File View Tools Options Contrast Geometry Spatial Help Applications
Disp
IRAS
2

LV=1,WT=0,LC=SOL,CD=0,TP=KeyPt
Enter seam point. Reset to end.

Figure 11: I_RAS C Cut-Line Selection

4.4 MOSAIC DEVELOPMENT

After each image is added to the new mosaic, a variety of partial products are removed from the workspace and the growing mosaic is saved. It is usually saved in a TIFF format, and given a name which reflects both the subject matter and the level of the work completed to data—for example, a mosaic of a hydrothermal vent made of images 1027 through 1034 might be called vent1027-1034.tif. Successive saving of partial mosaics allows retracing of steps. It is frequently necessary to abandon mosaics in progress, regressing to earlier steps and choosing new candidate images for input since distortions can grow quite rapidly.

In this manner, it is possible to create large mosaics, in almost a "production-line" method. For example, during the 1997 Derbyshire survey, over 60 mosaics were made on-board and over 130 delivered eventually (Howland, 1999a, 1999b). Figure 12 is an example from the Derbyshire survey (DETR, 1998).

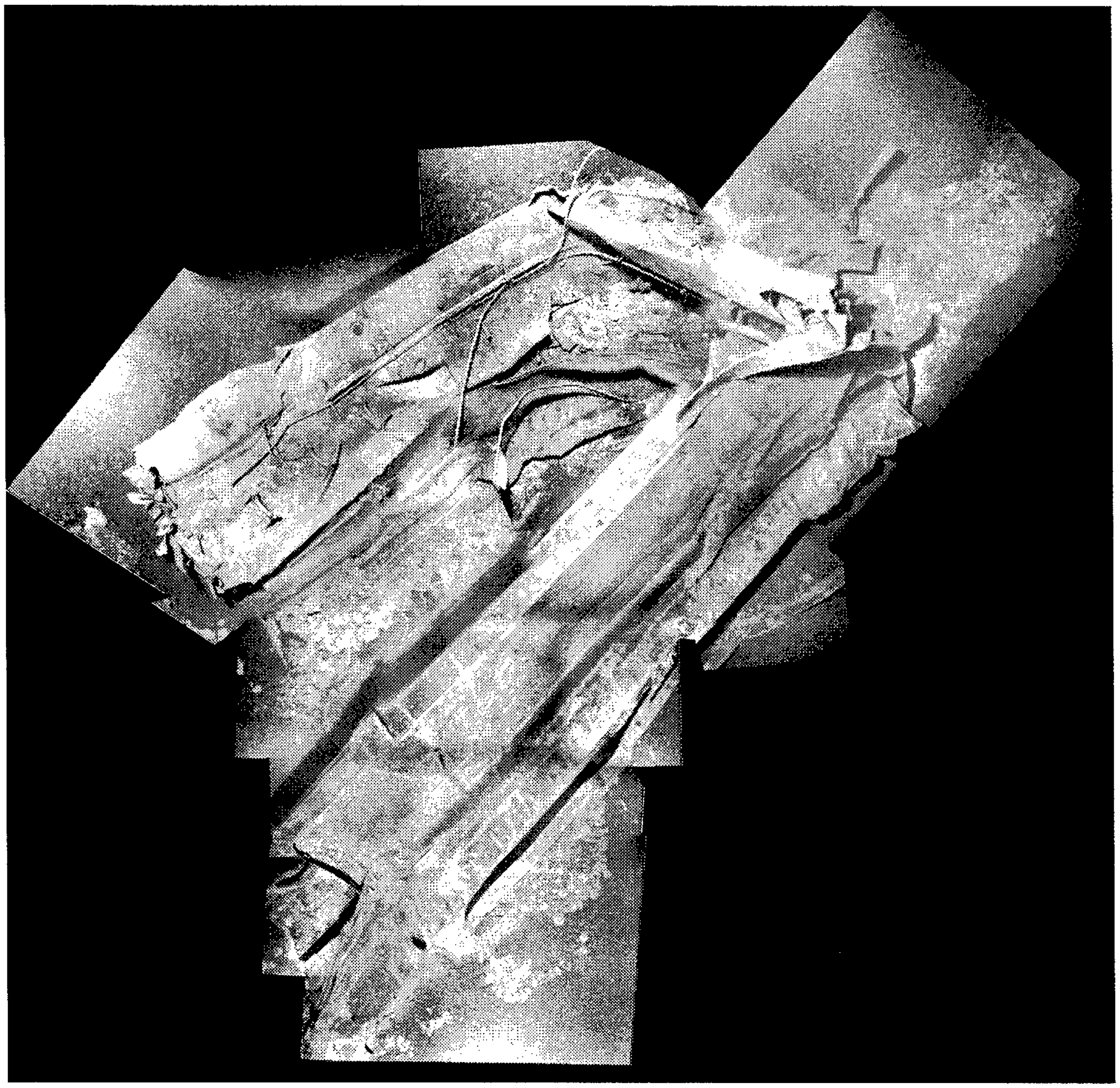


Figure 12 Mosaic of Wing Tank, Derbyshire Survey.

4.5 MOSAICKING STRATEGIES

As (Howland, et al, 1999) describes, unavoidable geometric distortions can severely limit the development of large mosaics, particularly over complex terrain or objects. However, several strategies have proven useful in minimizing the distortion and producing large relatively undistorted mosaics.

4.5.1. SEMI-CONTROLLED MOSAICS

In aerial photogrammetry, the production of controlled mosaics, in which imagery is actually registered to terrain or to base maps using ground control points is common. In oceanography, of course, ground control points are virtually unknown. However, navigation data can be used to partially "control" mosaics which would otherwise be grossly distorted by the inaccuracies in tie-point warped mosaics. The principle method used to semi-control a mosaic is to pre-place images into a base map based upon their navigation and attitude (principally heading) data. Other images are then warped to these "control" images. Error growth throughout the basemap is thus controlled and approximate distances can be scaled linearly from the mosaic.

This capability was fully implemented in early versions of custom mosaic software developed at DSL. However, when the decision was made to shift to off-the shelf software for the tie-point warping and blending step of the mosaicking process, semi-controlling mosaics, although possible, became much more difficult, and has rarely been used.

4.5.2. STRIP AND PATCH BASED MOSAICKING

Small areas can generally be mosaicked without excessive distortion using the tie point warping methodology described. They have also been created using the commercial automated mosaicking software. If the areas covered using these techniques grow too large, however (where the meaning of "too large" is entirely dependent upon the scale of the imagery, the controllability of the vehicle carrying the camera, the nature of the terrain, and the skill of the mosaicker), development of a single mosaic must stop, and another must be started. If small patches or strips developed in this way overlap, they can be mosaicked together, producing a satisfactory result.

4.6 THE USES OF MOSAICS

Among the greatest difficulties faced by the developers of the mosaicking techniques at DSL has been limiting the expectations and applications of over-zealous mosaic users. Our current mosaicking techniques do not produce a scaleable map. We cannot produce mosaics of unlimited size, not even given unlimited time. The geometric distortions inherent in making a two dimensional projection of a three-dimensional world using a multitude of small, virtually independent two-dimensional projections of that world are a fundamental limitation of our current approach. No degree of automation will allow tie-point based image warping to produce geometrically accurate maps.

Mosaics are useful for obtaining a *gestalt* view of an underwater scene. They have been useful in marine geology, archaeology, biology, and forensics. Figure 13 shows the use of a mosaic of an archaeological site in change monitoring. The two mosaics show the site before and after recovery of artifacts by the Jason vehicle (reference here). Figure 14 is a geological mosaic from the 1996 Lucky Strike survey (Fornari, 1996). In (Scheirer et al, in press), this mosaic was used to ground-truth side-scan sonar data.

Errata: The reference for the Artifact recovery figure is: "In Press, Ballard, R.D., A.M. McCann, D.R. Yoerger, L.W. Whitcomb, D.A. Mindell, J. Oleson, H. Singh, B. Foley, J. Adams, D. Piechota, and C. Giangrande; The Discovery of Ancient History in the Deep Sea Using Advanced Deep Submergence Technology; Deep Sea Research I."

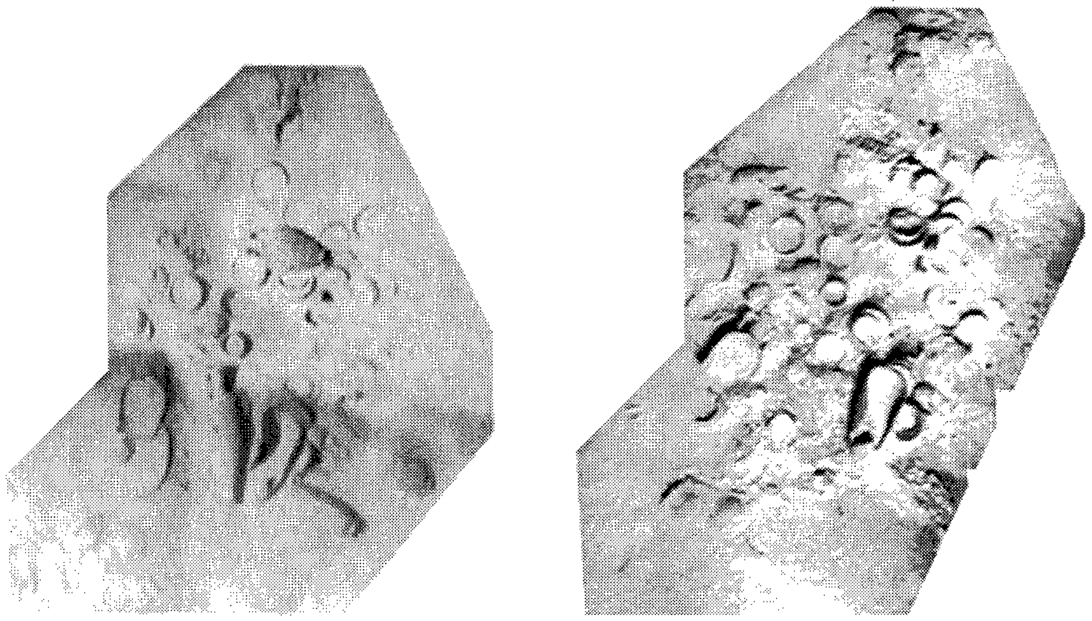


Figure 13 Archaeological Site Before and After Artifact Recovery

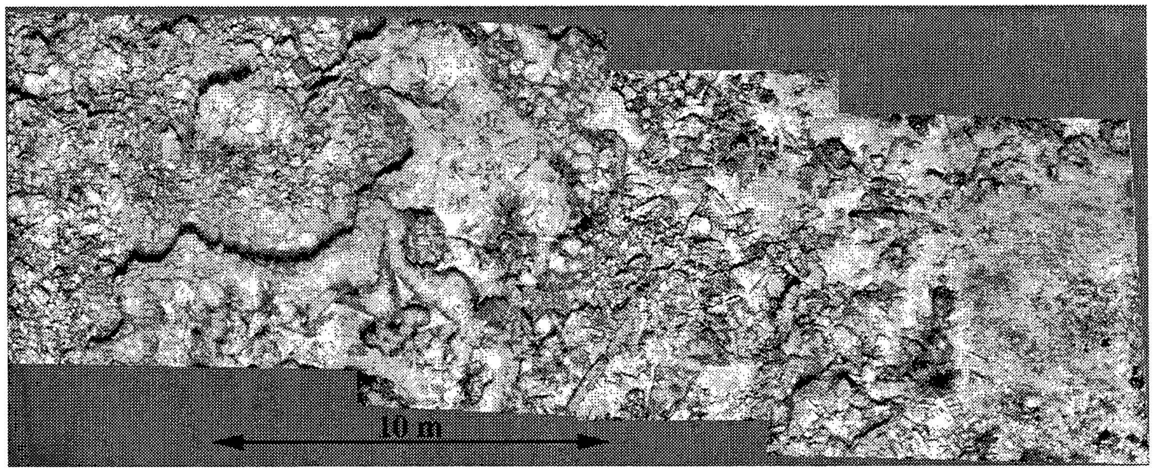


Figure 14: Geological Mosaic from Lucky Strike Area

Mosaics must, however, be used with caution by users who understand their fundamentally physics based limitations. They are best used in combination with the original image data, which can be analyzed using stereo photogrammetric techniques or in conjunction with vehicle navigation data and other sensor parameters to produce precise mensuration. Figure 15 shows a screen from the *Visual* software package (Lerner, 1999) which allows access to the original data from within a controlled mosaic.

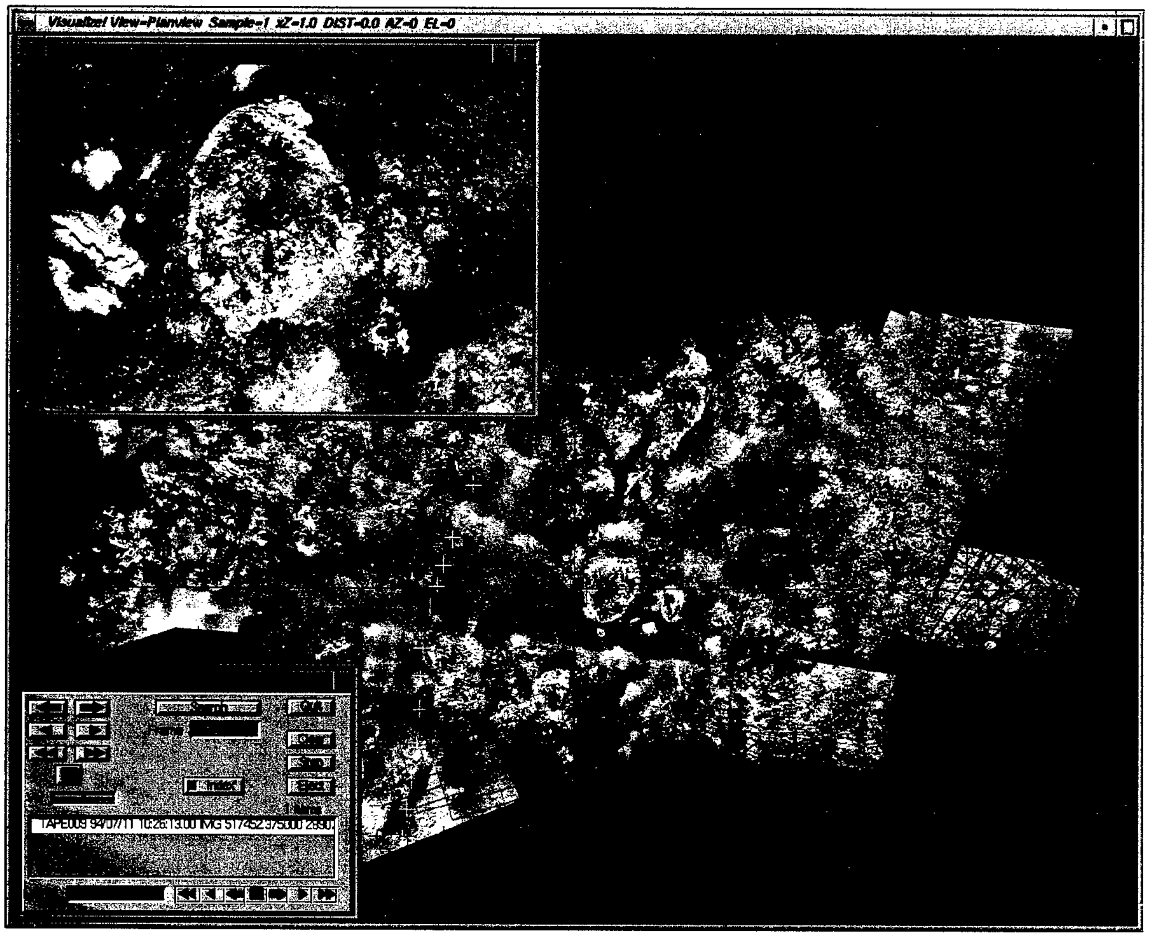


Figure 15: Use of Visual with mosaics and original data.

Ongoing efforts at the Deep Submergence Laboratory and elsewhere may eventually reduce or eliminate these strictures on the use of mosaics by allowing full use of sea-floor shape information in the mosaic. This shape information will come either from acoustic surveys or from the imagery itself, using overlapping data in stereo-based photogrammetric techniques.

5. THE FUTURE OF THE IMAGING PIPELINE

We have described a pipeline that excels at handling images produced in a particular way: collected and recorded using a particular camera and tape media, in a particular format. We fully anticipate that changes in our image collection and storage mechanisms will change the pipeline. In particular, we hope to develop an acquisition system that allows real-time access to digital image data over a shipboard network. This will allow implementation of parallelism in processing and distribution, greatly streamlining the pipeline and real-time access to the data. However, the basic approach of batch processing of imagery to support mosaicking will probably remain part of our system, albeit with more ready access to raw or normalized data for individually adaptive processing in unusual circumstances. We will continue to maintain a pipeline straight through to archiving, since although ready access to partial products will be important during oceanographic cruises, safeguarding and bringing home the original data will always remain a priority.

We intend to implement network-based access similar to that described in Section 3 on a routine basis in DSG operations.

6. REFERENCES

1. Ballard, Robert, 1993. "The Jason Remotely Operated Vehicle System," WHOI Technical Report 93-34, February 1993.

2. Chromey, Frederick, 1996. "Special Considerations for Flat Fielding," *CCD Astronomy*, Fall, p. 18-23.

3. Department of the Environment, Transport, and the Regions, (DETR), 1998. M.V> Derbyshire Surveys, UK/EC Assessors Report.

4. Fornari, Daniel, 1996. Final Cruise Report, LUSTRE '96, R.V. Knorr Cruise 145-19, August 15, 1996

5. Harris, S.E., R.J. Squires, and E.M. Bergeron, 1987. Underwater Imagery Using an Electronic Still Camera, *Proceedings, MTS/IEEE Oceans '87*, pp. 1242-1245.

6. Heckbert, Paul, 1994. *Graphics Gems IV*, Academic Press.

7. Howland, 1999a. Digital Data Logging and Processing, Derbyshire Survey, 1997. WHOI Technical Report WHOI-99-08.

8. Howland, 1999b. Additional Mosaics and Stereo Pairs of the M.V. Derbyshire, WHOI Technical Memorandum WHOI-04-99.

9. Howland, J., H. Singh, M. Marra, and D. Potter, 1999, Digital Mosaicking of Underwater Imagery, *Sea Technology*, June1999, pp 65-68.

10. Kelley, D.S., Delaney, J.R. and D. R. Yoerger, High Resolution Imaging and Sampling of Sulfide Structures from the Mothra Hydrothermal Field, Endeavor Segment, Juan de Fuca Ridge, in prep.

11. Kleinrock, M.C., Humphris, S.E., and the Deep-TAG Team (Shaw, P., Bowen, A., Crook, T., Davis, C., Elder, R., Gleason, D., Goff, J., Goldstein, L., Handley, W., Howland, J., Hussenoeder, S., Koga, K., Lerner, S., Nakamura, K., Rashid, M., Reiser Wetzel, L., Sellers, W., Sulanowska, M., Van Dover, C. and Whitcomb, L.) A2. Detailed Structure and Morphology of the Tag Active Hydrothermal Mound and Its Geotectonic Environment, Proceedings of the Ocean Drilling Program, Initial Reports, Vol. 158, 15-21, 1996.

12. Lerner, S., 1998. Data Monitoring, Access, and Analysis Systems for the M.V. Derbyshire Survey, 1997, Proceedings Volume 2 MTS/Ocean Community Conference '98, pp. 1109-1113.

13. Lerner, S., 1999. Visual: A Visualization system for Accessing and Analyzing Multi-Sensor Data, WHOI Technical Report WHOI-99-013.

14. Lerner, S., and A. Maffei, 4D GeoBrowser: A Web-Based Data Browser for Accessing and Analyzing Multi-Disciplinary Data, WHOI Technical Report, to be published 2000.

15. Newberry, Michael, 1995. "Recovering the Signal," *CCD Astronomy*, Spring, pp. 18-21

16. Pizer, S.M., E.P. Amburn, J.D. Austin, R. Cromartie, A. Gesolowitz, B. ter Haar Romeny, J. B. Zimmerman and K. Zuidervald, 1987. Adaptive Histogram Equalization and its Variations. *Computer Vision, Graphics, and Image Processing*, 39:355-368.

17. Scheirer, D. S., Daniel J. Fornari, Susan E. Humphris, and Steve Lerner, "High-Resolution Seafloor Mapping Using the DSL-120 Sonar System: Assessment of Sidescan and Phase-Bathymetry Data from the Lucky Strike Segment of the Mid-Atlantic Ridge," Submitted to *Marine Geophysical Researches*, 1999.

18. Shiren, Yang, Li Li, and Gao Peng, 1989. Two Dimensional Seam-Pont Searching in Digital Image Mosaicking, *Photogrammetric Engineering and Remote Sensing,* LV: 49-54.

19. Singh, H., J. Howland, L. Whitcomb, and D. Yoerger, 1998. "Quantitative Photomosaicking of Underwater Imagery," Proceedings of Oceans '98, IEEE/OES Conference, Nice France, October 1998.

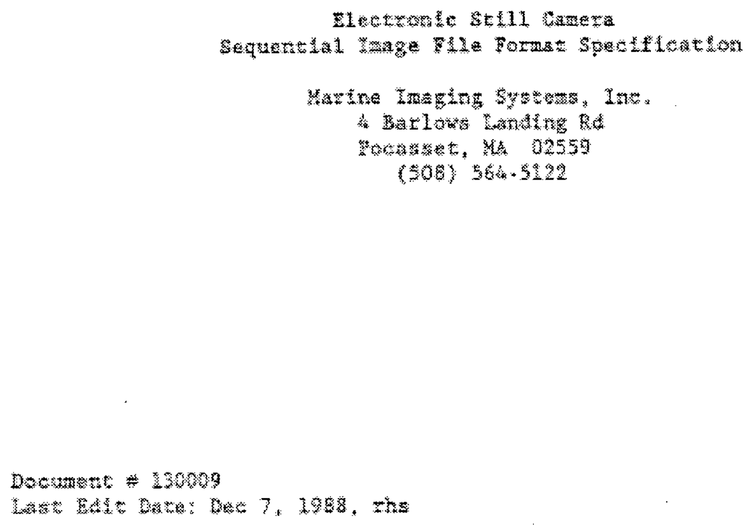
20. Sulanowska, M.M., Humphris, S.E. and Howland, J.C., 1996. "Detailed Analysis of the Surface Morphology of the Active TAG Hydrothermal Mound by Mosaicking of Digital Images," *EOS, Transactions of the American Geophysical Union*, vol. 77, p. 768.

21. Van Dover, C. L., J.R. Cann, C. Cavanaugh, S. Chamberlain, J.R. Delaney, D. Janecky, J. Imhoff, J.A. Tyson, and the LITE Workshop Participants, 1994. Light at Deep-Sea Hydrothermal Vents, EOS, 75:44-45.

22. Yoerger, D.R., Bradley, A.M., Cormier, M-H., Ryan, W.B.F. and Walden, B.B., 1999. "High Resolution Mapping of a Fast Spreading Mid Ocean Ridge with the Autonomous Benthic Explorer," 11th International Symposium on Unmanned Untethered Submersible Technology (UUST99), Durham, New Hampshire, August 1999.

7. APPENDIX A: MIS DATA FORMATS

The following scans of the Manual pages describe the .mis format used for the Marquest Model 8100/9100 camera system:



Electronic Still Camera
Sequential Image File Format Specification

Marine Imaging Systems, Inc.
4 Barlows Landing Rd
Pocasset, MA 02559
(508) 564-5122

Document # 130009
Last Edit Date: Dec 7, 1988, rhs

1. Introduction

This document will describe the image storage format of the Model 8100 Electronic Still Camera (ESC) for all sequential type media. Devices directly supported are the Archive 5945S streaming tape drive (about 60 Mbytes/tape) and the Exabyte EXB-8200 tape drive (about 2.2 Gbytes/tape).

It is assumed that the user of the ESC images has access to a Unix type computer, since all images are read and written in the Unix standard "tar" (for tape archiver) format. Many Unix computers have a tape drive compatible with the Archive tape unit.

2. Extraction of Images

For the standard 8100 ESC equipped with an Archive streaming tape drive, tapes are written in QIC-24 format with 9 tracks on a 0.25 inch tape. Tapes are typically DC-600A or DC-600XL tapes (600 foot) for about 60 Mbytes of image storage.

For extraction, the Unix command on a SUN 3 type workstation is:

```
tar -xvtf /dev/rst8
```

The "/dev/rst8" is required, since the "normal" SUN tape device is "/dev/rst", which is a QIC-11 format device.

The first image on the tape is named "im0000", and image numbers simply increment upwards. Note that for the smaller CCD arrays (less that 600x400), each image is roughly 1/2 Megabyte.

All sequential devices that MIS supports will adhere to this image numbering convention, and all sequential devices will be in "tar" format.

3. Image Format

This section will describe the image formats after they are extracted from tape. Details of "tar" formats can be found in most Unix documentation.

The image has a 1024 byte header describing the contents of the image, followed by the image itself, followed by zero padding as described in the next section.

3.1 File Sizes

Due to restrictions in the ESC Display and Recording unit (specifically, the requirement that the ESC always keep the tape streaming), the extracted file sizes are limited to the solutions of the following equation:

(file_size_in_bytes + 512) % 4096 = 0

The "%" operator is a "modulus" operator. Thus, the extracted file sizes will always be, in bytes:

(4096 - 512 = 3584), (8192 - 512 = 7680), ...

Note that when writing images to tape to plug into the ESC Display and Recording Unit, the file sizes MUST follow this convention: simply zero pad the images to the correct size. Note that since file sizes are limited to the above number of bytes, the image size will be, for example, 3584 bytes minus the 1024 byte header.

3.2 Image Header Format

The attached "C" header file shows the format of the 1024 byte header. The structure is zero padded for a total length of 1024 bytes. All integers are 4 bytes long, the most significant byte appearing first in the file.

The "time.h" include file that appears in the header include file is a Unix standard include file that describes the structure of the time stored in the image header, and is attached also. See any standard Unix documentation for an exact description of the time structure, and the routines for converting this structure into strings, etc.

The "wx,wy wh,ww" describe the origin and size of the area on the CCD chip to read out. These parameters are in display coordinates. See the section below on coordinate systems.

The "decim" is the decimation factor, which, in general, is relevant only on cameras that have limited bandwidth between the camera head and the surface Display and Recording Unit (DRU), and as of the date of this document, has not been used. "decim" is essentially the (x and y) subsampling factor used to subsample the image before transmission from the camera head to the surface DRU. Note that this is different than "binning", since binning actually combines pixels on the CCD chip. If "decim" is greater than 1, the image size is reduced by (decim*decim).

The "bin" parameter describes how the CCD chip binning factor is set before exposing and reading out the chip. Binning, which is always the same in the x and y directions, is a function of the CCD chip that combines pixels into larger super-pixels, thus reducing the image size by (bin*bin), but increasing sensitivity by the same factor. Not all CCD chips support binning. This parameter is valid only if it is greater than 1.

The "mode" parameter indicates how the image was generated. In "auto" mode, the image was a taken in the normal "reprate" second repetition rate. In the "manual" mode, the image was taken on command. As of the date of this writing, this parameter is always set to zero.

The "ccdtype" parameter is how or where the image was generated from. It can be one of the chip types shown in the image header include file, and others will be added as different chips become available. Note that test images that are created during power up sequences, etc. also are a "ccdtype". This parameter can be used to index a table yielding the CCD array size, and the order in which the CCD scans the image out.

"pixbits" will be 12 or 14, which is the number of significant bits in each pixel.

"flipx" and "flipy", which are derived from the type of CCD chip tell whether the image must be mirrored about x and/or y to display the image. See the section below on coordinate systems.

If "hmin" and "hmax" are both zero, no histogram information is included in the header. Otherwise, hmax is always greater than hmin.

The "order" parameter is discussed below in the coordinate system section.

All other parameters are self explainatory. Strings are always terminated with a null (0) byte.

3.3 Image Data

Image data is always organized as 16 bit binary pixels, with the most significant bits set to zero if they are not used. The most significant byte of a pixel always appears before the least significant byte.

3.4 Coordinate Systems

There are two basic coordinate systems that the user of the image data must understand before using the image data. The display coordinates are how the image will appear on the video (or other display device). The origin (0,0) of the display coordinate system is always in the upper left hand corner of the screen.

The CCD chip coordinate system most likely will not match the the display coordinate system, due to the physical structure and orientation of the CCD. For example, the Thompson TH7882 CCD chip has it's origin at the upper right hand corner, and reads out the pixels from top to bottom (column order), rather than raster order. To further complicate matters, the ESC DRU can rotate or mirror the image before storing it in some applications. In any case, the "order", "flipx" and "flipy" parameters always indicate how the image actually appears in the file.

The following table shows how the image data must be manipulated to load into a display device to insure that the image is not mirrored or rotated for display, based on the information in the image header.

TABLE 1. Image Storage Row, Column Ordering

	Parameter		
order	flipx	flipy	Image Storage Order
0	0	0	Col by Col, top to bottom, left to right
0	0	1	Col by Col, bottom to top, left to right
0	1	0	Col by Col, top to bottom, right to left
0	1	1	Col by Col, bottom to top, right to left
1	0	0	Row by Row, left to right, top to bottom
1	0	1	Row by Row, left to right, bottom to top
1	1	0	Row by Row, right to left, top to bottom
1	1	1	Row by Row, right to left, bottom to top

Note that a CCD chip that requires a mirror about the Y axis has not yet be delivered in any ESC, since this can be accomplished by turning the camera head 180 degrees.

```
/* Header file for storage of images on tape and disk              */
/* @(#)ihead.h        1.3  6/29/88  */
/* Robert H. Squires                                                */
/* Marine Imaging Systems, Inc.                                     */

#ifndef _TIME
#include "time.h"
#endif
#define COL_ORDER     0    /* for filling in "order" part of header */
#define ROW_ORDER     1
#define REVISION      1    /* current revision of header stuff */
#define TIMELEN          32   /* length of human readable time */
#define COMMENTLEN       256  /* length of comment field  */
#define IHEADSIZE        1024 /* size of image header */

#define TI4849        0          /* chip types, 384 x 391, Texas Instruments */
#define TH7882  1            /* 576 x 384, Thompson  */
#define TC215   2            /* 1024x1024, Texas Instruments  */
#define SYNTHETIC_IMAGE 3  /* A computer made it up */

/* image header       */
struct T_ih {
     int wx;          /* window origin x&y */
     int wy;
     int wh;          /* height and widths */
     int ww;
     int decim;       /* decimation factor */
     int bin;         /* binning factor */
     int reprate;     /* rep rate, in seconds */
     int expose;      /* exposure length, in msec */
     int mode;        /* surface mode (auto, pix on command, etc) */
     int ccdtype;     /* type of ccd chip */
     int count;       /* image count  */
     int pixbits;     /* number of bits/pixel (12 or 14) */
     int flipx;       /* 1 - must mirror around x to display */
     int flipy;       /* 1 - must mirror around y to display */
     int hmin;        /* histogram min */
     int hmax;        /* histogram max */
     int hnpix;       /* number of pixels histogrammed */
     int altitude;    /* altitude  */
     int revision;    /* revision code of image headers */
     int order;       /* see defines above for values  */
     int reserved[9];/* reserved area */
     struct tm time; /* time struct                              */
     char hrtime[TIMELEN];  /* human readable time */
     char comment[COMMENTLEN];  /* comment */
     };
```

Page 5

```
/* Include file for time stuff                                          */
/* @(#)time.h        1.1 2/8/88                                         */
/* Robert H. Squires                                                    */
/* Marine Imaging Systems, Inc.                                         */

#ifndef _TIME
#define _TIME

struct       tm {
     int     tm_sec;
     int     tm_min;
     int     tm_hour;
     int     tm_mday;
     int     tm_mon;
     int     tm_year;
     int     tm_wday;
     int     tm_yday;
     int     tm_isdst;
     };

extern struct tm *gmtime(), *localtime();
extern char *ctime(), *asctime();
extern void tzset();
extern long timezone;
extern int daylight;
extern char *tzname[];

#endif
```