

Vision Platform for Mobile Intelligent Robot Based on 81.6 GOPS Object Recognition Processor

Donghyun Kim, Kwanho Kim, Joo-Young Kim, Seungjin Lee and Hoi-Jun Yoo

Dept. of EECS, KAIST, 373-1, Gu-Seong Dong, Yu-Seong Gu, Daejeon, 305-701, KOREA

+82-42-869-8068

donghyun53@eeinfo.kaist.ac.kr

ABSTRACT

To enable power-efficient object recognition of mobile intelligent robots, 81.6GOPS object recognition processor is proposed. Based on analysis of Scale Invariant Feature Transform (SIFT) algorithm, architecture of the proposed processor is designed to support both task and data level parallelism. 10 Processing Elements (PEs) are integrated for task parallelism, and each PE is equipped with SIMD instruction for data parallelism as well. In addition, Visual Image Processing memory replaces complex local maximum pixel search operation with a single read operation for further performance gain. With the proposed processor, we also realized vision platform for real-time SIFT computation of mobile robots. The chip operation is tested up to 200MHz and consumes 540mW in the vision platform at 1.8V supply voltage and 100 MHz operation frequency.

Categories and Subject Descriptors

C.3. [Computer Systems Organization]: Special-Purpose and Application-Based Systems

General Terms: Design, Measurement, Performance, Verification

Keywords: Object Recognition, Network-on-Chip, Multi-Processor SoC.

1. INTRODUCTION

Recently, autonomous navigation has become a mandatory function of mobile intelligent robots and previous works based on Simultaneous Localization and Mapping (SLAM) algorithm [1-4] has tried to realize it. In the SLAM, Scale Invariant Feature Transform (SIFT) based object recognition is widely used to improve accuracy in localization of robots [1, 2, 5, 6, 7] because the SIFT is robust to luminance and scale variations of input image.

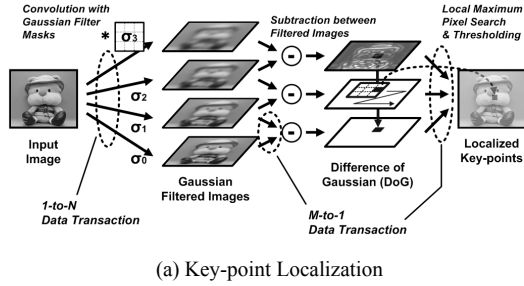
The SIFT based object recognition requires huge computing power because it involves numerous iterations of image convolutions and local maximum pixel search operation over entire input image. For mobile intelligent robots, providing high-performance computing is very challenging because they are generally equipped with limited power supply. Previous implementations of intelligent robots rely their processing on power-hungry general purpose processors suffering from significant power overhead [1, 2, 8, 9]. For example, Pioneer 3DX [9], which adopts state of the art laptop at the time of publication, consumes more power in its embedded computer than its mechanical movement. The operable time of such robots is limited to tens of minutes [1, 9].

Even though few previous object recognition processor implementations are reported in [19-21], they are all targeted for vehicular application having much relaxed power constraints. To the date, there is no dedicated processor implementation which provides sufficient performance for the SIFT computation with low-power consumption. This leads us to design a single chip application specific processor with power efficient features [13-15] and to implement robot vision system based on the implemented chip. In this work, we present high-performance vision system that consumes much less power compared to general purpose processor based vision systems.

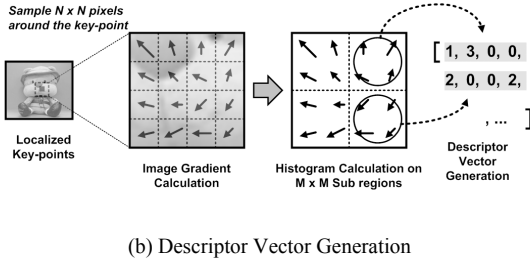
The rest of the paper is as follows. In section 2, SIFT based object recognition is analyzed as a target application. From the analysis, desirable chip architecture and features are also discussed. In section 3, details of the proposed object recognition processor implementation is described. The chip realizes the features decided in section 2 for fast and power-efficient SIFT computation. Design and verification flow and used tool chains are also explained. After that, section 4 reports fabrication and measurement results. Design of the vision platform based on the proposed processor is described in section 5. Performance evaluation results are also covered. Finally, conclusion is made in section 6.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 2008, June 8–13, 2008, Anaheim, California, USA
Copyright 2008 ACM 978-1-60558-115-6/08/0006...5.00



(a) Key-point Localization



(b) Descriptor Vector Generation

Fig. 1. Overall Flow of the SIFT Object Recognition

2. TARGET APPLICATION: SIFT

Fig. 1 shows overall flow of the SIFT computation which is divided into (a) key-point localization and (b) descriptor vector generation stages. For key-point localization, Gaussian filtering with varying coefficients is repeatedly performed on the input image. Then subtractions among the filtered images are carried out to yield Difference of Gaussian (DoG) images. By performing the DoG operation, edges of different scales are detected from input images. After that, locations of key-points are decided by finding local maximum pixels using 3x3 search window over the entire DoG images. The pixels having local maximum value greater than given threshold become key-points.

The next stage of the key-point localization is descriptor vector generation. For each key-point location, $N \times N$ pixels of input image are first sampled and then, the gradient of the sampled image is calculated. The sample size N is decided according to the DoG image where the key-point is selected. Finally, descriptor vector is generated by computing orientation and magnitude histograms over $M \times M$ sub regions of the sampled input image.

The huge amount of computation during the SIFT computation motivates us to exploit parallelism. At first, data parallelism is revealed in the Gaussian filtering task, which single instruction is repeatedly applied to the entire pixels on the input image. On the other hand, task level parallelism is apparent in the descriptor vector generation stage. At each key-point, the processing starts with fetching $N \times N$ pixels from the input image, and the amount of subsequent computation depends on the number of pixels. Because the N varies according to the key-points, every computation workload also varies at each key-point. In

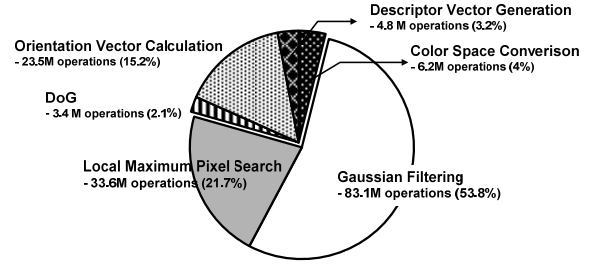


Fig. 2 Amount of Required Computation

such situation, supporting independent task executions of multiple Processing Elements (PEs) rather than utilizing all the PEs in SIMD mode execution is more efficient. For that reason, we adopted multi-processor architecture for the proposed object recognition processor to facilitate task level parallelism. To exploit data parallelism as well, we added an SIMD instruction for Gaussian filtering within the ISA of the integrated PEs.

We also investigated data transactions in the SIFT computation to discover requirements of the on-chip interconnection. In computing the SIFT, each of the tasks such as Gaussian filtering, DoG and local maximum pixel search reads data from its former stage task and delivers processed data to its subsequent task. Considering characteristics of such data transactions and multi-processor architecture, it is straightforward to organize pipelined task execution on a multi-processor architecture, with proper mapping of the tasks. Therefore, supporting efficient 1-to- N and M -to-1 data transactions, which is shown in Fig. 1 (a), between stages of task level pipeline is desirable feature.

For further improvement in performance, amount of required computation is calculated for each of the SIFT tasks as shown in Fig. 2, without regarding overhead of instruction fetching, managing data structure. Two most demanding tasks are the Gaussian Filtering and local maximum pixel search operation. Whereas the Gaussian filtering is accelerated by the SIMD instruction, large amount of cycle is consumed for loading 9 pixels and subsequent 9 comparisons and conditional branches. To reduce overhead of pixel loading and comparison, we implemented special purposes memory that reads out address of local maximum pixel in response to the center pixel address input of the search window. In summary, hardware features that are advantageous for the efficient SIFT computation is as follows

- Multi-processor architecture for task parallelism
- SIMD instruction for data parallelism
- Interconnections that support efficient 1-to- N and M -to-1 data transactions
- Special purpose memory to accelerate local maximum pixel search operation

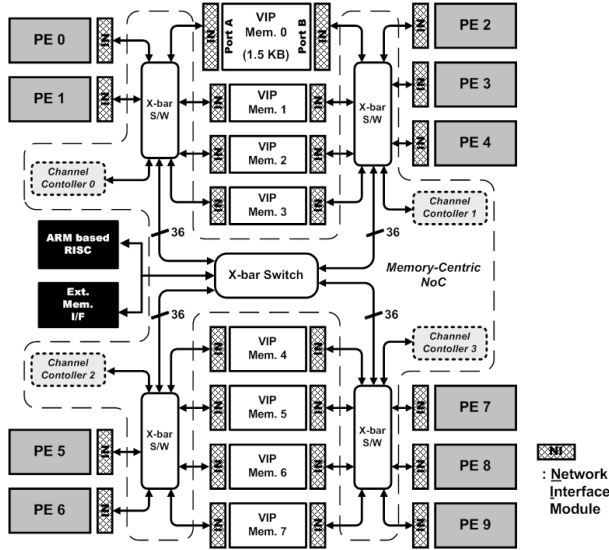


Fig. 3 Object Recognition Processor Architecture

3. Chip Implementation

3.1 Processor Architecture

Overall architecture of the proposed object recognition processor is shown in Fig. 3. The main components of the proposed processor are 10 SIMD Processing Elements (PEs), 8 Visual Image Processing (VIP) memories and an ARM based RISC processor. The 10 PEs are integrated for task level parallelism such as parallel execution of Gaussian filter operations with different coefficients. The RISC controls overall operation of the proposed processor by initiating execution of each PE. The 8 VIP memories provide inter-PE communication buffers and accelerate the local maximum pixel search task.

For efficient 1-to-N and M-to-1 data transactions, interconnection among the 10 PEs and the 8 VIPS memories are provided by the Memory-Centric NoC [13-

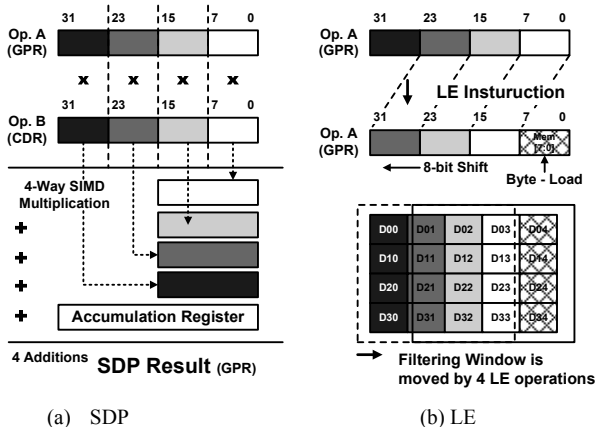


Fig. 4 SDP (SIMD) & LE Instructions

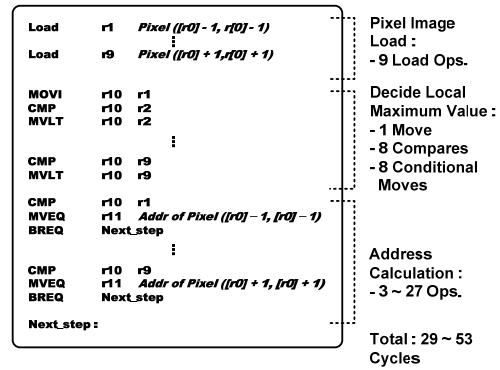


Fig. 5 Local Maximum Pixel Search Operation

15]. Contribution of the Memory-Centric NoC to the power-efficient SIFT computation is reduced external memory transactions. It is enabled by facilitating pipelined execution of the SIFT tasks. The Memory-Centric NoC is composed of 5 crossbar switches, 4 channel controllers and Network Interface (NI) modules. Since most of the data transactions locally occur among the PEs and VIP memories, hierarchical star-topology instead of regular mesh topology is adopted for power and area efficiency [16]. The Memory-Centric NoC manages dynamic utilization of the communication buffers and provides memory transaction control scheme between producer and consumer PEs.

3.2 SIMD Processing Element

To accelerate Gaussian filtering tasks, special instructions for image filtering are implemented in the SIMD PE. The instructions are SDP (Sum of Dot Product) and LE (Load Extension). For the SDP instruction, 12 Coefficient Dedicated Registers (CDR) are also added to store filter coefficients. As shown in Fig. 4, the SDP instructions calculate 8 bit 4-way SIMD multiplications and subsequent 4 additions including accumulation in a single cycle. The SDP instruction brings one operand from General Purpose Register (GPR) and the other operand from the CDR. The LE instruction is combination of 8 bit shift and byte load operation, which is designed to support seamless filter window movement over image data. Once filter coefficients and image data are stored in the CDR and GPR respectively, replacing pixels of the GPR using the LE instruction is equivalent to moving filter mask over input image when the SDP instruction follows the series of the LE instructions. By performing the SDP instruction, each PE performs 8 operations in a single cycle and its operation frequency is designed to 200 MHz. As a result, the 10 PEs contribute to 16 GOPS of the total performance.

3.3 Visual Image Processing Memory

The VIP memory is specially designed to find location of local maximum pixel inside 3x3 search window in a single cycle. As shown in Fig. 5, searching local maximum pixel

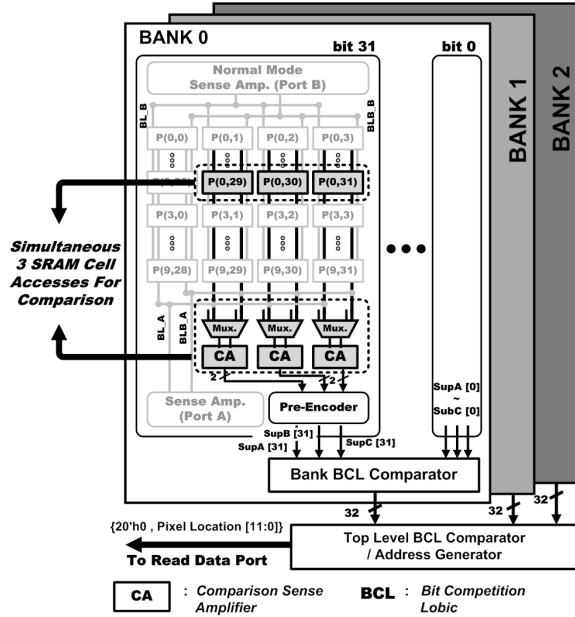


Fig. 6 Visual Image Processing Memory Architecture

location needs 29~53 cycles on ARM based RISC. By replacing this time consuming computation with single read operation, huge performance gain is obtained. In addition to the normal memory operation, the function of the VIP memory is to read out address of local maximum pixel inside 3x3 search window in response to the address input of center pixel in the window. Since local maximum pixel search operation takes 41 cycles on average, the 8 VIP

memories operating 200MHz gives 65.6 GOPS performance gain. The overall architecture of the VIP memory is shown in Fig. 6. In the VIP memory, 12 rows by 32 columns of 32 bit pixels are stored which result in total 1.5KB capacity. To compare 9 pixel values in one cycle, every row is interleaved into 3 banks so that bank number assigned for each row is decided by modulo-3 operation. Three pixels in the same row are first compared inside the bank, then results from 3 banks are compared again to find local maximum pixel among 9 pixels. Address of local maximum pixel is automatically generated according to the comparison result by address generation unit. At each bank, 3 Comparison Amplifiers (CA) are integrated into every 4 bit line pairs to read 3 pixel values simultaneously. The transistor size of the CA is smaller than normal sense amplifier because it does not drive long capacitive DB lines. To minimize area overhead of comparison logic in the memory, Bitwise Competition Logic (BCL) [17] is also devised. By adopting the BCL, transistor count of comparator is reduced from 2400 to 536 when compared to the conventional adder based comparator. More details of the VIP memory is described in [18].

3.4 Design & Verification Flow

The proposed object recognition processor is designed by semi-custom design flow. In the case of the VIP memory, full-custom design is necessary because modification to the internal cell and sense amplifier architecture is required for integration of the comparator logic. The other components of the chip are designed based on automated design flow. The design and verification flow of the proposed processor with used EDA tools is shown in Fig. 7. To verify co-operation of the modules designed by different design flows, we carried out verilog-nanosim co-simulation.

For this simulation, gate level netlist from the automated design flow and transistor level netlist obtained from the schematic design is used together. From the co-simulation, decision on additional synchronizer design is made to cope with clock skew between synthesized modules and full-custom design modules. It took 9 months from application analysis to tape-out with 7 graduate students.

4. Chip Fabrication Results

Fig. 8 shows the chip photograph and summary of implementation results. The proposed object recognition processor is fabricated using 0.18um standard CMOS process technology. The die size is 7.7 x 5 mm² and its operation frequency is 400 MHz for the Memory-Centric NoC and 200 MHz for other parts of the chip. Total gate counts excluding on-chip SRAMs and Visual Image Processing memories are 838.8K in terms of unit-NAND2 gates. The chip operation is separately verified up to 200 MHz and it operates at 100 MHz in our vision platform. The operation frequency of the chip is limited to guarantee stable synchronization with the FPGA in the vision

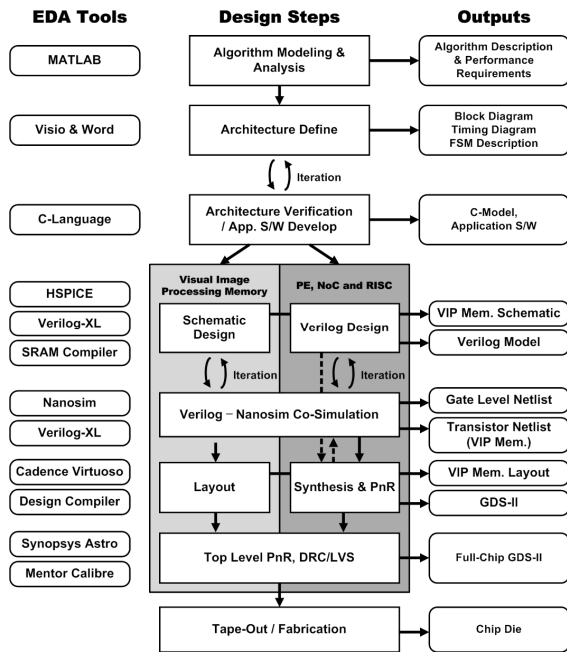
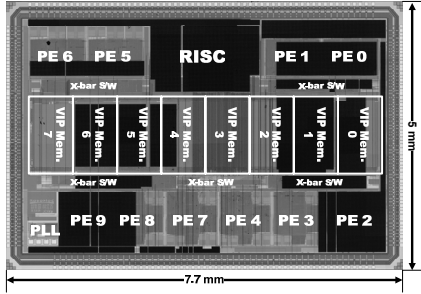


Fig. 7 Chip Design & Verification Flow



(a) Chip photograph

Technology	0.18um 1-poly 6 Metal
Chip Size	7.7mm x 5mm
Clock Freq.	400 MHz (NoC) / 200 MHz (Other Part)
Gate Counts (NAND2 Equiv.)	838.8K Gates
On-Chip Memory	Total : 30 KB VIP Memory : 1.5KB x 8 PE Local Mem. : 1KB x 10 RISC Cache : Data \$ 4KB, Inst. \$ 4KB
Peak Power Consumption	1.4W at 1.8V

(b) Implementation Summary

Fig. 8 Chip Fabrication Result

platform. Simulated peak power consumption is 1.4W at 1.8V and 200MHz operating condition. Measured power consumption is 540mW at 100MHz operating frequency at 1.8V power supply voltage.

5. Vision Platform

5.1 Vision Platform Implementation

Fig. 9 shows block diagram of the implemented vision platform based on the proposed processor. The platform incorporates 8 MB asynchronous SRAM and 8 MB flash memory to provide the proposed processor with working and code memory, respectively. For video image

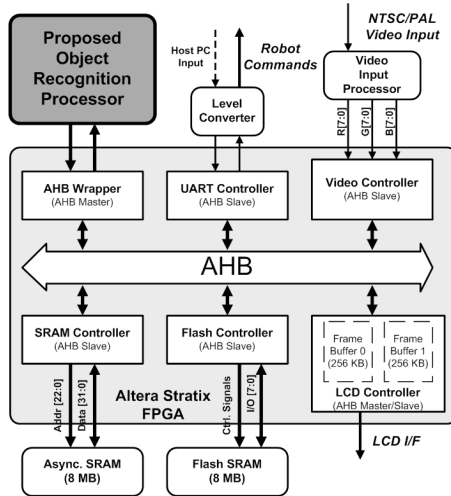


Fig. 9 Block Diagram of the Vision Platform

acquisition, it receives NTSC/PAL analog inputs and video input processor converts the input signal into digital RGB format. Memory controllers, UART and LCD controller are implemented as soft IPs in the FPGA for flexibility of the vision platform. In the FPGA, AHB is implemented as an interconnection structure for the system components. To reduce overhead of moving image data into the frame buffer, the LCD controller operates as a DMA device and autonomously fetches video data from the external SRAM. By adopting FPGA, observing bus signals and internal signals of the soft IPs are also possible. We used ALTERA Stratix series FPGA supporting SignalTap II logic analyzer which shows internal signal probing from the host PC, using JTAG interface. In the platform implementation, verification of each soft IPs and overall operation mainly carried out using the signal probing feature.

TABLE I. SIMULATION CASES AND CORRESPONDING FEATURES

Cases	Execution Mode	LE Inst.	SDP Inst.	VIP Memory
A	Data Parallel	X	X	X
B	Data Parallel	O	X	X
C	Data Parallel	O	O	X
D	Pipelined Task	O	O	X
E	Pipelined Task	O	O	O

5.2 Performance Evaluations

To evaluate performance of the proposed processor, execution time of the key-point localization stage is compared for the 5 simulation cases in Table I. The execution time is measured for 320x240 pixels image. In the cases A, B and C, the SIFT tasks are sequentially executed while the 10 PEs execute each task in parallel. In the case of D and E, the different tasks are executed in pipelined manner and input image is sequentially fetched. Execution times for the 2.3 GHz Intel Core2 Duo and 200 MHz ARM9 TDMI processors are compared together to prove superior performance of the proposed processor over the conventional state of the art processors. By tracking the reduction in execution time from the case A to E, advantages of the proposed techniques are clearly represented. At first, the SDP SIMD instruction reduces execution time of the Gaussian filtering task drastically. (B → C) After that, execution time is further reduced by switching to the pipelined task from the data parallel execution mode, because it reduces overhead of external memory transactions having large access latency. (C → D) This is the advantage of the Memory-Centric NoC which facilitates pipelined task execution. In addition, adopting the VIP memories in the task level pipeline gives additional 2 times performance gain. (D → E) The proposed object

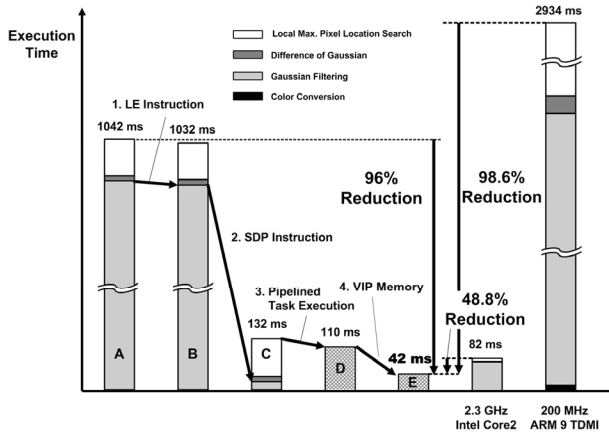


Fig. 10 Performance Evaluation Results

recognition processor achieves 48.8 % and 98.6% reduction in execution time, compared to the Core2 Duo and ARM9 processors respectively.

5.3 Vision Platform Realization

Fig. 11 shows vision platform implementation results. The size of the PCB is 10cm x 12cm and it is assembled on the human type mobile robot. In Fig. 11 key-point localization of the SIFT is demonstrated. In the screen shoots of the LCD, white rectangles represent localized key-points. The key-points are correctly detected on a point of human attention such as eyes and mouth of the doll in this case. The measured power consumption of the vision platform is about 2.6W when the proposed processor operates at 100 MHz consuming 540mW.

6. Conclusion

For power-efficient object recognition of mobile intelligent robot, 81.6 GOPS object recognition processor is

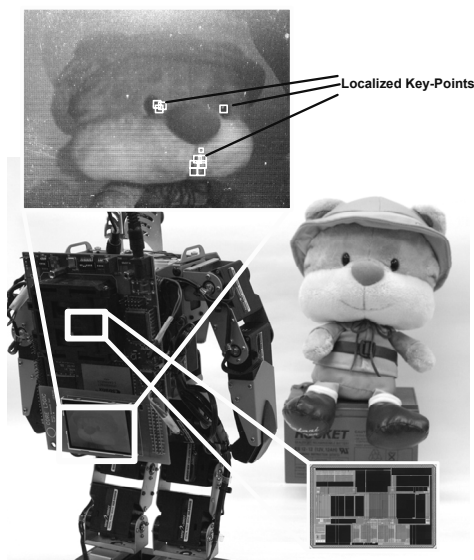


Fig. 11 Vision Platform Realization

implemented. Vision platform for mobile robots is also realized using the proposed processor as a main component. Compared to the around 14W power consumption in [9], the SIFT computation is realized with much lower power consumption in 2.6W.

7. References

- [1] Sunghwan Ahn, et. al., "Data Association Using Visual Object Recognition for EKF-SLAM in Home Environment," Proceedings of IEEE Intl. Conf. on Intelligent Robots and Systems, pp. 2760-2765, 2006.
- [2] Patric Jensfelt, et. al., "Augmenting SLAM with Object Detection in a Service Robot Framework," IEEE Intl. Symposium on Robot and Human Interactive Communication, pp. 741-746, 2006.
- [3] Bertolli F., Jensfelt P., Christensen H.I., "SLAM using Visual Scan-Matching with Distinguishable 3D Points," IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4042-4047, Oct., 2006.
- [4] Zhang Nan, Li Maohai, Hong Bingrong, "Active Mobile Robot Simultaneous Localization and Mapping," IEEE International Conference on Robotics and Biomimetics, pp.1671-1681, Dec., 2006.
- [5] David G. Lowe, "Distinctive Image Features from Scale-Invariant Key points," ACM Intl. Journal of Computer Vision, Vol. 60, Issue 2, pp. 91-110, 2004
- [6] Bertolli F., Jensfelt P., Christensen H.I., "SLAM using Visual Scan-Matching with Distinguishable 3D Points," IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4042-4047, Oct., 2006.
- [7] Eggers S.J., et al., "Simultaneous Multithreading: A Platform for Next-Generation Processors," IEEE Micro, Vol. 17, Issue 5, pp.12-19, Sep.-Oct., 1997.
- [8] Lin C.Y., Jo P.C. and Tseng C.K., "Multi-Functional Intelligent Robot DOC-2," IEEE-RAS International Conference on Humanoid Robots, pp.530-535, Dec., 2006.
- [9] Yongguo Mei, et al., "A Case Study of Mobile Robot's Energy Consumption and Conservation Techniques," Proceedings of the IEEE Intl. Conf. of Advanced Robotics, pp. 492-497, 2005.
- [10] Shorin Kyo, et. al., "A 51.2 GOPS Scalable Video Recognition Processor for Intelligent Cruise Control Based on a Linear Array of 128 4-Way VLIW Processing Elements," Digest of Technical Papers, IEEE Intl. Solid-State Circuits Conf., Vol. 1, pp. 48-477, 2003.
- [11] Wolfgang Raab, et. al., "A 100 GOPS Programmable Processor for Vehicle Vision Systems," IEEE Design & Test of Computers, Vol. 20, Issue 1, pp. 8-15, Jan.-Feb., 2003.
- [12] Jun Tanabe, et. al., "Visconti: Multi-VLIW Image Recognition Processor based on Configurable Processor," Proceedings of the IEEE Custom Integrated Circuits Conf., pp. 185-188, 2003.
- [13] Donghyun Kim, et al., "Solutions for Real Chip Implementation Issues of NoC and Their Application to Memory-Centric NoC," ACM/IEEE International Symposium on Networks-on-Chip, pp. 30-39, May, 2007.
- [14] Donghyun Kim, et al., "An 81.6 GOPS Object Recognition Processor Based on NoC and Visual Image Processing Memory," Proceedings of the IEEE Custom Integrated Circuits Conference, pp.443-446, Sep., 2007.
- [15] Donghyun Kim, et al., "Implementations of Memory-Centric NoC for 81.6 GOPS Object Recognition Processor," Proceedings of the IEEE Asian Solid States Circuits Conference, pp.47-50, Nov., 2007.
- [16] Kangmin Lee, Se-Joong Lee, and Hoi-Jun Yoo, "Low-Power Networks-on-Chip for High-Performance SoC Design," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 14, No.2, pp.148-160, February, 2006.
- [17] Joo-Young Kim and Hoi-Jun Yoo, "Bitwise Competition Logic for Compact Digital Comparator," Proceedings of the IEEE Asian Solid States Circuits Conference, pp.59-62, Nov., 2007.
- [18] Joo-Young Kim, Donghyun Kim, et al., "Visual Image Processing RAM for Fast 2-D Data Location Search," Proceedings of the IEEE European Solid-State Circuits Conference, pp.324-327, Sep., 2007.
- [19] Shorin Kyo, et. al., "A 51.2 GOPS Scalable Video Recognition Processor for Intelligent Cruise Control Based on a Linear Array of 128 4-Way VLIW Processing Elements," Digest of Technical Papers, IEEE Intl. Solid-State Circuits Conf., Vol. 1, pp. 48-477, 2003.
- [20] Wolfgang Raab, et. al., "A 100 GOPS Programmable Processor for Vehicle Vision Systems," IEEE Design & Test of Computers, Vol. 20, Issue 1, pp. 8-15, Jan.-Feb., 2003.
- [21] Jun Tanabe, et. al., "Visconti: Multi-VLIW Image Recognition Processor based on Configurable Processor," Proceedings of the IEEE Custom Integrated Circuits Conf., pp. 185-188, 2003.