PAPER *Special Section on Circuits and Design Techniques for Advanced Large Scale Integration*

A Low-Power Real-Time SIFT Descriptor Generation Engine for Full-HDTV Video Recognition

Kosuke MIZUNO[†a)], **Hiroki NOGUCHI**[†], ***Student Members***, **Guangji HE**[†], **Yosuke TERACHI**[†], **Tetsuya KAMINO**[†], **Tsuyoshi FUJINAGA**[†], ***Nonmembers***, **Shintaro IZUMI**[†], ***Student Member***, **Yasuo ARIKI**[†], ***Nonmember***, **Hiroshi KAWAGUCHI**[†], ***and*** **Masahiko YOSHIMOTO**[†], ***Members***

SUMMARY This paper describes a SIFT (Scale Invariant Feature Transform) descriptor generation engine which features a VLSI oriented SIFT algorithm, three-stage pipelined architecture and novel systolic array architectures for Gaussian filtering and key-point extraction. The ROI-based scheme has been employed for the VLSI oriented algorithm. The novel systolic array architecture drastically reduces the number of operation cycle and memory access. The cycle counts of Gaussian filtering module is reduced by 82%, compared with the SIMD architecture. The number of memory accesses of the Gaussian filtering module and the key-point extraction module are reduced by 99.8% and 66% respectively, compared with the results obtained assuming the SIMD architecture. The proposed schemes provide processing capability for HDTV resolution video (1920 × 1080 pixels) at 30 frames per second (fps). The test chip has been fabricated in 65 nm CMOS technology and occupies $4.2 \times 4.2\,\text{mm}^2$ containing 1.1M gates and 1.38 Mbit on-chip memory. The measured data demonstrates 38.2 mW power consumption at 78 MHz and 1.2 V.

key words: *SIFT, image recognition, low-power, HDTV*

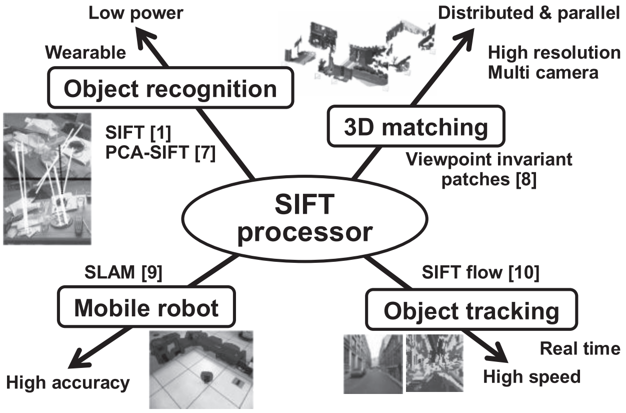


Fig. 1 Various applications and algorithms based on SIFT.

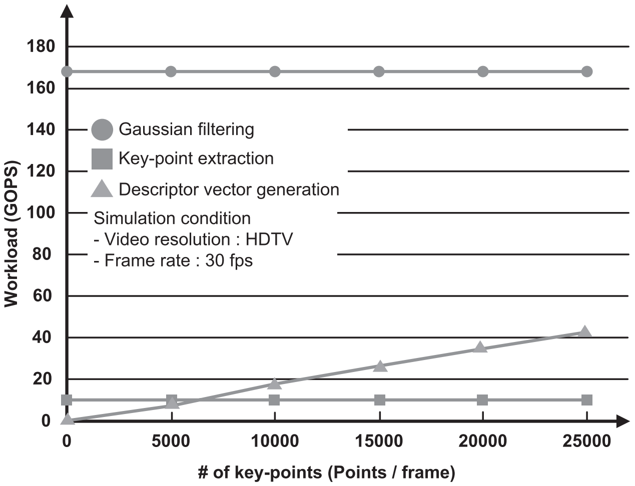


Fig. 2 Workload analysis of SIFT descriptor generation.

1. Introduction

In recent years, image recognition systems have been developed through evolution of circuit technologies and algorithms for image processing. A major algorithm used in image recognition systems, Scale Invariant Feature Transform (SIFT) [1], is invariant to changes of scale, rotation and illumination. Moreover, several algorithms, PCA-SIFT [7], Viewpoint invariant Patches [8], Simultaneous Localization and Mapping (SLAM) [9], and SIFT flow [10], are based on the SIFT algorithm. Because of its consistency and expandability, SIFT has high adaptability to object recognition, image matching, object tracking, mobile robot, and so on (Fig. 1). Most of the applications suffer from high power consumption because of heavy workload required in SIFT descriptor generation. In addition, each application has different requirements such as high video resolution, high frame-rate and high accuracy. Therefore, low-power and high-performance SIFT processor is valuable over a wide range application.

Some processors implementing SIFT have been previously developed [2]–[6]. The previous works are unsuitable especially for mobile application under limited battery condition on account of high operating frequency and high power dissipation. We previously proposed FPGA implementation for SIFT [6]. The FPGA implementation provides real-time operation at VGA resolution video. However, it cannot work in real-time for high resolution video. The proposed processor has resolved the problem of power consumption and obtained higher performance, compared with the previous processors.

Figure 2 portrays the workload analysis of the SIFT descriptor generation employing software program [11] for the original SIFT algorithm. The horizontal axis and the vertical axis indicate the number of key-points per one frame and workload, respectively. Key-points which have extreme

Manuscript received August 13, 2010.
Manuscript revised November 7, 2010.

[†]The authors are with Kobe University, Kobe-shi, 657-8501 Japan.
a) E-mail: mi-no@cs28.cs.kobe-u.ac.jp

DOI: 10.1587/transele.E94.C.448

values represent features of an image. Gaussian filtering and descriptor vector generation dominate the large part of the whole workload. At 5,000 key-points, the workload of Gaussian filtering occupies 90 percent of all. It is seen that reduction of the computation workload of Gaussian filtering is necessary for low-power and real-time SIFT descriptor generation.

Figure 3 shows the memory-bandwidth analysis of the SIFT descriptor generation using the same software [11]. Gaussian filtering for HDTV resolution video requires 5374 Gbps. Also key-point extraction requires 1254 Gbps at the same resolution. According to Fig. 2, the workload of the key-point extraction is not so heavy that the computation power is not issue. However, memory bandwidth is major design issue for real-time and low-power implementation of the Gaussian filter and the key-point extractor. Therefore, a dedicated hardware design for the key-point extraction is demanded. Our design reduces the memory bandwidth by adopting novel systolic array architectures for Gaussian filtering and key-point extraction. These architectures employ extensive reuse of pixels and intermediate results.

The above analysis shows real-time and low-power SIFT descriptor vector generation requires resolving the three problems: (1) the workload of Gaussian filtering, (2) memory bandwidth of Gaussian filtering, and (3) the memory bandwidth of key-point extraction. To achieve a real-time and low-power SIFT descriptor vector generation, we propose the following three techniques with low accuracy degradation:

- Gaussian filtering architecture with 120-way systolic array for workload and memory bandwidth reduction.
- Buffer register to provide pixels for the 120-way Gaussian filtering architecture for efficient pixel reuse.
- Key-point extraction architecture with 81-pixel parallel systolic array for memory access reduction.

This paper describes a sub-100 mW SIFT descriptor generation processor with real-time full-HDTV video recognition. The processor provides several applications with longer life time and higher performance, compared with the previous processors [2]–[6].

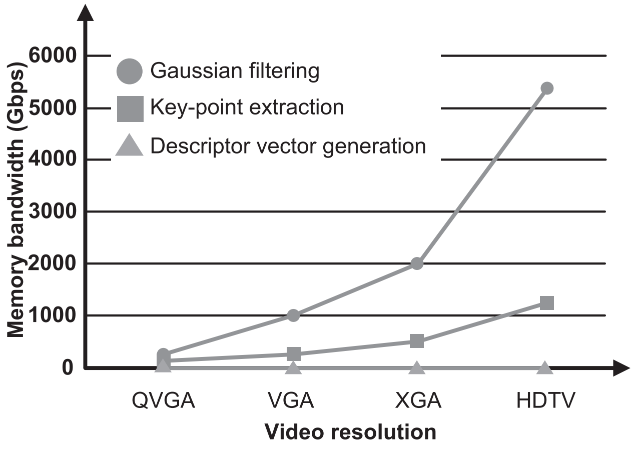


Fig. 3 Memory-bandwidth analysis of SIFT descriptor generation.

In this paper, the detail of the employed SIFT algorithm is described in Sect. 2. The proposed architecture is addressed in Sect. 3. Then, these are followed by VLSI implementation and power measurement in Sect. 4. Section 5 concludes this paper.

2. Algorithm

2.1 Algorithm Overview

Figure 4 shows a flow diagram of original SIFT algorithm. Hierarchical algorithm is adopted to obtain robustness to a scale change. The SIFT descriptor generation consists of Gaussian filtering, key-point extraction and descriptor vector generation. The input image is smoothed by Gaussian filtering for key-point extraction. Adjacent Gaussian filtered images are subtracted for difference-of-Gaussian (DOG) generation. Key-point is detected by searching on DOG. Key-point detection uses three DOG images. Maxima and minima of DOG images are detected by comparing a pixel in middle scale to its 26 neighbors in 3 × 3 regions at the current and adjacent scales. Finally, SIFT descriptor vectors are obtained by calculating a gradient histogram of luminance around the key-point. The above process continues until the highest level picture.

2.2 SIFT Algorithm for VLSI Implementation

An employed algorithm for VLSI implementation is introduced in this subsection. The major differences between the original algorithm and the employed algorithm are two parts, pre-processing for input image and Gaussian filtering.

Original scheme handles a whole image all at once. It can provide high-accuracy result, but it causes increase in hardware resources. On the other hand, the employed method processes an image divided into Region of Interests (ROI). The introduction of ROI provides a structure appropriate for clock gating and operating frequency control with little accuracy degradation. In addition, it reduces a capacity in an on-chip memory to 0.3% in the case of HDTV image.

There are two methods for Gaussian pyramid generation. One method computes a Gaussian filtered image in

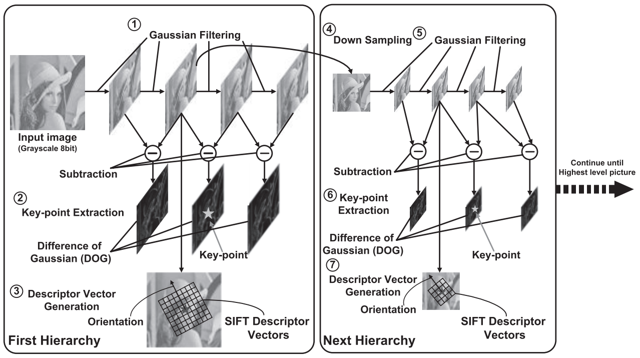


Fig. 4 Original SIFT algorithm flow.

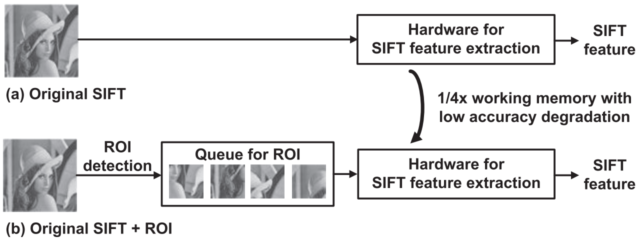
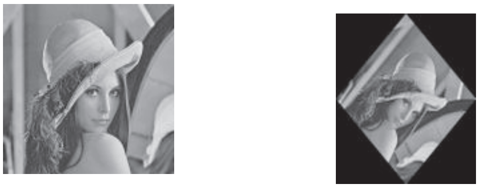


Fig. 5 Conceptual diagram of introducing ROI into original SIFT.



(a) Original image (b) Transformed image

Fig. 6 Test Image pair of (a) original image and (b) transformed image with scale changes and viewpoint changes.

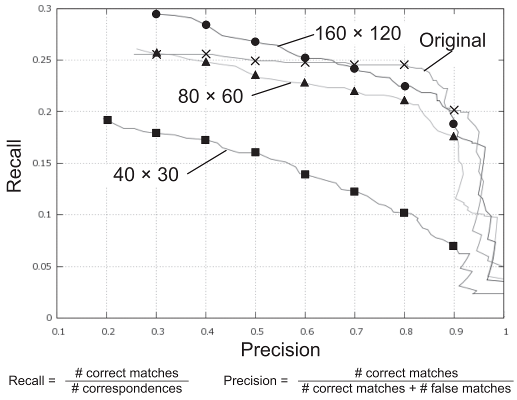


Fig. 7 Recall versus precision in image matching.

any hierarchy recursively from an input image. This method keeps down the workload, but it increases some latency and required working memory. The other one generates each Gaussian filtered image solely from an input image concurrently. It enables a low-latency operation and a parallel hardware implementation and reduces required working memory with some workload overhead. Therefore, the latter method has been employed for the VLSI design.

2.3 Region of Interests

Figure 5 depicts a conceptual diagram introducing ROI into the original SIFT algorithm: (a) is the original algorithm, and (b) is a ROI-based algorithm dividing input image into four ROIs. The ROI-based algorithm drastically reduces the capacity for working memory with low accuracy degradation. Reference [12] specified for object recognition employs 40 × 40 for ROI size. However, it is supposed that larger ROI size is required for several wide-ranging applications for SIFT.

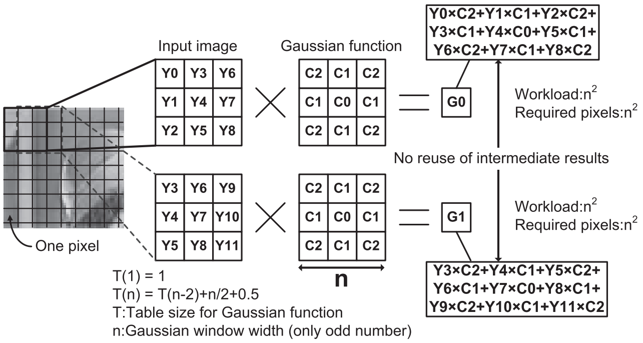


Fig. 8 Disadvantage of using two-dimensional Gaussian filtering.

A simulation has been conducted by using software for image matching to determine ROI size. The software has been built up on Microsoft Visual C++ 2008 Express Edition. Figure 6 shows a test image pair of original image and transformed image with scale changes and viewpoint changes. The recall versus precision in image matching application [13] has been evaluated for various ROI sizes as shown in Fig. 7. It is seen that the accuracy in 80 × 60 is relatively close to the original one. After the trade-off study between accuracy and memory capacity, 80 × 60 has been chosen as ROI size. Moreover, we chose 10 pixels as the overlap pixel count between neighboring ROIs in order to prevent the accuracy degradation in processing around a boundary of ROI. Hence we employed the 80 × 60 ROI-based SIFT with 10 pixel of ROI overlap. In the case of HDTV resolution, the employed algorithm requires the working memory for 80 × 60 pixels, while original SIFT does 1920 × 1080 pixels. Consequently, the employed method reduces working memory by 99.7% compared with the original scheme.

2.4 Two-Dimensional Gaussian Filtering

In this subsection, we describe a two-dimensional Gaussian filtering algorithm suitable for a hardware implementation. Figure 8 shows two-dimensional Gaussian filtering at a pixel level, where Y is the input image, C is the coefficient of Gaussian function, and G is the result of one Gaussian filtering. A result of two-dimensional Gaussian filtering is the sum of a result by multiplication of one pixel Y and one coefficient C. It is hard to reuse an intermediate result in this scheme because the same combination of one pixel and one coefficient do not exist in adjacent Gaussian filtering.

Figure 9 depicts the combination of two 1-dimensional Gaussian filterings where C is the coefficient of Gaussian function, and G′ is the intermediate result of 1-dimensional Gaussian filtering. This scheme can drastically reduce workload and the number of memory accesses thanks to a high reusability of intermediate results G′.

2.5 Simulation Results

A simulation has been conducted by using software for object recognition to confirm a performance and an accuracy

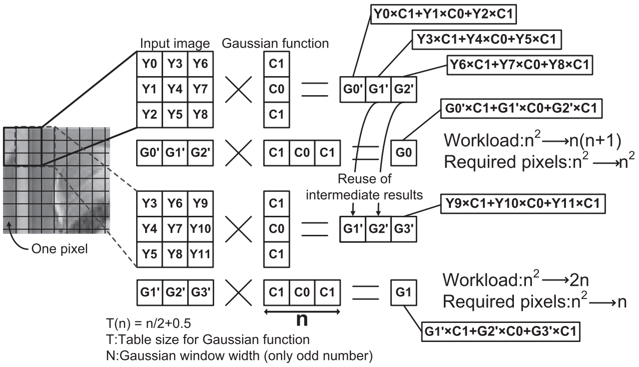


Fig. 9 Advantage of using one-dimensional Gaussian filtering.

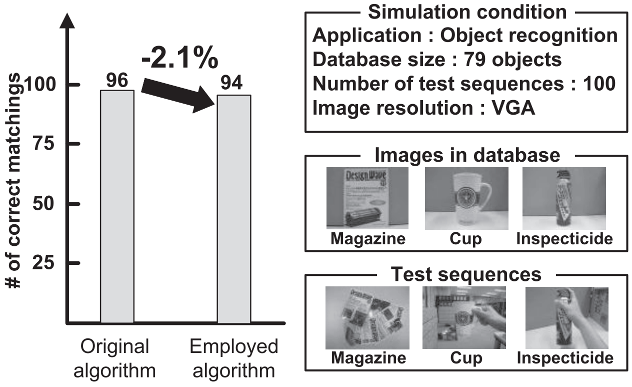


Fig. 10 Accuracy degradation by the employed algorithm.

degradation of the employed algorithm. The software has been built up on Microsoft Visual C++ 2008 Express Edition. Figure 10 shows the result of object recognition test and experimental condition. The respective numbers of objects and test images are 79 and 100. The used database was created by us. The database includes general objects in a house or an office such as a cup, a book and so on. The simulation results show that the recognition rate degradation is 2.1%. It is seen that the employed algorithm provides sufficient performance for general-purpose applications.

3. Architecture

3.1 Three-Stage Pipelined Architecture

Figure 11 shows a block diagram of the three-stage pipelined architecture. The proposed architecture is comprised of a global sequencer, a Gaussian filtering module, a key-point extraction module, a descriptor vector generation module, SRAMs for several image data, and the other peripherals. Three-stage pipeline can efficiently handle ROI and considerably reduce power dissipation. Each SRAM for Gaussian filtered image buffers images over 6 scales in one hierarchy. The SIFT descriptor generation engine is controlled by a RISC processor, and input image is loaded to an ROI buffer from an external SDRAM via a memory interface. The three data-path modules independently access an SRAM for a Gaussian filtered image. The part enclosed by dotted line has been implemented in VLSI chip described in Sect. 4.

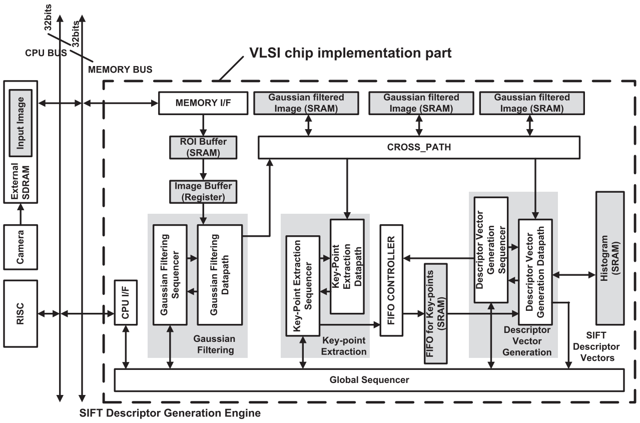

32bits
32bits
CPU BUS
MEMORY BUS
VLSI chip implementation part
Input Image
External SDRAM
Camera
RISC
MEMORY I/F
Gaussian filtered Image (SRAM)
Gaussian filtered Image (SRAM)
Gaussian filtered Image (SRAM)
ROI Buffer (SRAM)
CROSS_PATH
Image Buffer (Register)
CPU I/F
Gaussian Filtering Sequencer
Gaussian Filtering Datapath
Gaussian Filtering
Key-Point Extraction Sequencer
Key-Point Extraction Datapath
Key-point Extraction
FIFO CONTROLLER
FIFO for Key-points (SRAM)
Descriptor Vector Generation Sequencer
Descriptor Vector Generation Datapath
Descriptor Vector Generation
Histogram (SRAM)
SIFT Descriptor Vectors
Global Sequencer
SIFT Descriptor Generation Engine

Fig. 11 Three-stage pipelined architecture.

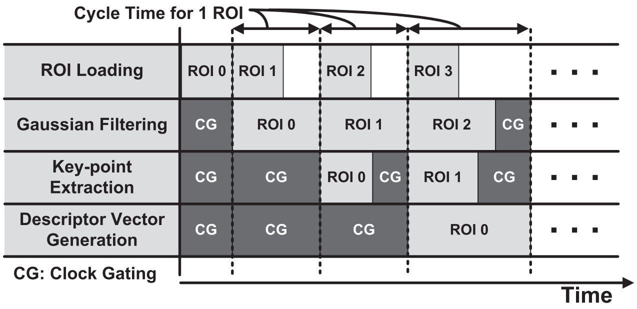


Fig. 12 Three-stage pipeline flow.

As shown in Fig. 12, the proposed pipeline consists of three stages (Gaussian filtering, Key-point extraction, and Descriptor vector generation). ROI is loaded from external SDRAM in background. The global sequencer feeds a different ROI to each stage. Cycle time for processing one ROI depends on the slowest stage. When any stage goes to idle state, the global sequencer executes clock gating for power minimization.

Each computation module contains parallel pixel-processing circuits and buffer circuits for pixel reuse to reduce required computation cycles. As a result of three-stage pipelining and highly parallel circuits, the proposed architecture can realize an energy-efficient and real-time SIFT descriptor generation with any resolution of input image up to HDTV.

3.2 Gaussian-Filtering Architecture with 120-Parallel Ring-Connected Systolic Array

First, a Gaussian filtering module is described. Gaussian filtering accounts for most of the workload in the SIFT operation. Therefore, this module is the most critical portion for a low-power and high-speed system design. We implemented 120-parallel systolic-arrays (SA) architecture to resolve the problem of workload. Figure 13 shows the block diagram

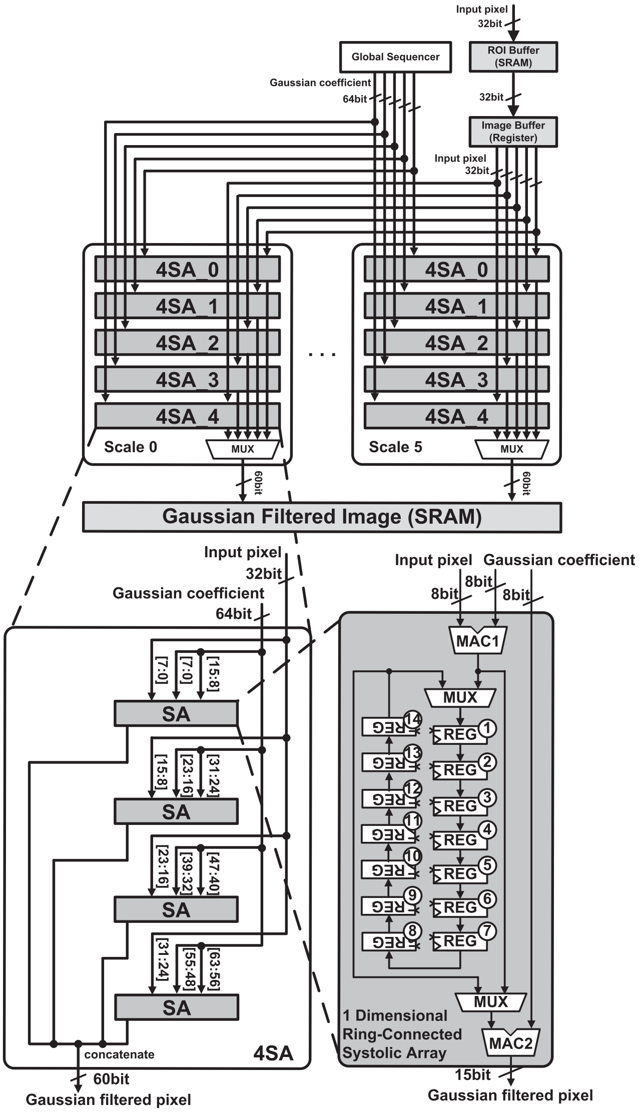


Fig. 13 Block diagram of Gaussian-filtering module.

of Gaussian filtering module comprised of 120 SAs. Twenty SAs per one scale are prepared for generating continuous 20 pixels in parallel. This scheme reduces the memory bandwidth by reusing input pixels. One SA has two multiplier accumulators (MAC), ring-connected fourteen registers and one multiplexer. One MAC takes one-dimensional Gaussian filtering operation. The ring-connected registers store an intermediate result of one-dimensional Gaussian filtering for reuse. Each SA reads one pixel per one cycle and operates with a pixel- level pipeline.

Image Buffer is introduced to efficiently provide input pixels for 120-way architecture. Figure 14 shows the block diagram of Image Buffer. Image Buffer has five shift registers buffering 18 pixels. Each shift register provides four pixels per one cycle for the corresponding 4SAs. Image Buffer reduces the overhead of memory read and the cycle count by pre-loading input pixels from ROI Buffer.

Figure 15 illustrates the operation of SA architecture.

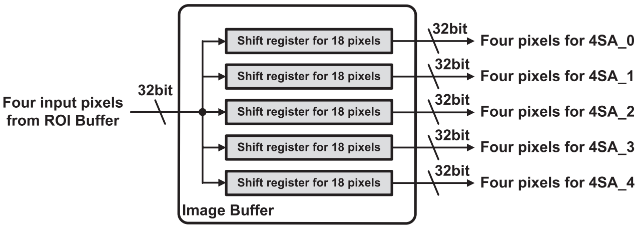
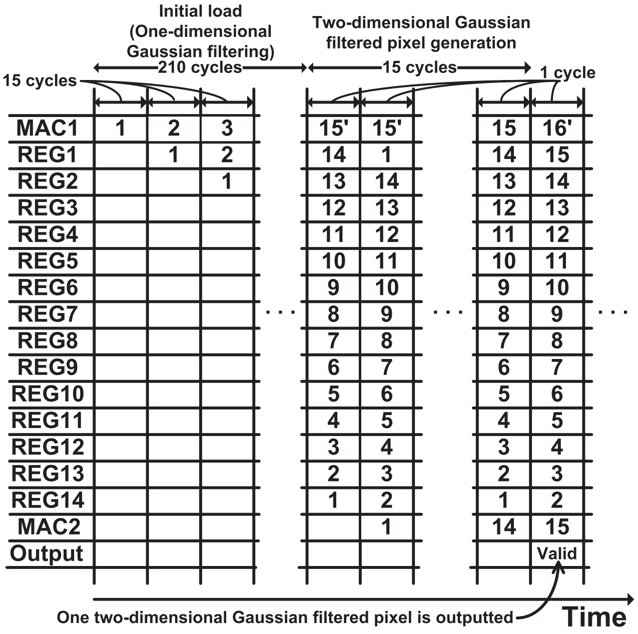


Fig. 14 Block diagram of Image Buffer module.


Initial load (One-dimensional Gaussian filtering)
210 cycles
Two-dimensional Gaussian filtered pixel generation
15 cycles
1 cycle
15 cycles
One two-dimensional Gaussian filtered pixel is outputted
Time

MAC1	1	2	3		15'	15'		15	16'	
REG1		1	2		14	1		14	15	
REG2			1		13	14		13	14	
REG3					12	13		12	13	
REG4					11	12		11	12	
REG5					10	11		10	11	
REG6					9	10		9	10	
REG7				· · ·	8	9	· · ·	8	9	· · ·
REG8					7	8		7	8	
REG9					6	7		6	7	
REG10					5	6		5	6	
REG11					4	5		4	5	
REG12					3	4		3	4	
REG13					2	3		2	3	
REG14					1	2		1	2	
MAC2						1		14	15	
Output									Valid	

Fig. 15 Flow diagram of SA architecture.

The REG's suffix number in Fig. 13 corresponds to the number in Fig. 15. Initially, all registers have no data. Then, an intermediate result of one-dimensional Gaussian filtering G′ is loaded into all registers as an initial load, in the first 210 cycles. Then the computation for two-dimensional Gaussian filtering is carried out. One two-dimensional Gaussian filtered pixel is generated every 15 cycles. In the first 14 cycles, each of the registers shifts intermediate data to the next register, and then MAC2 executes the multiply/accumulate operation of an intermediate data and Gaussian coefficient. Lastly, MAC1 provides the 15th intermediate data for MAC2 and REG1, and then one two-dimensional Gaussian filtered pixel is outputted via MAC2. REG1 stores the 15th data for the next two-dimensional Gaussian filtered pixel generation. This operation continues until reaching the end of an ROI.

3.3 Key-Point Extraction Architecture with 81-Pixel Parallel Systolic Array

Next, a key-point extraction module is described. The key-point extraction suffers from the second heaviest memory bandwidth in the SIFT operation, as shown in Fig. 3. Therefore, the parallelization and pixel reuse has been extensively

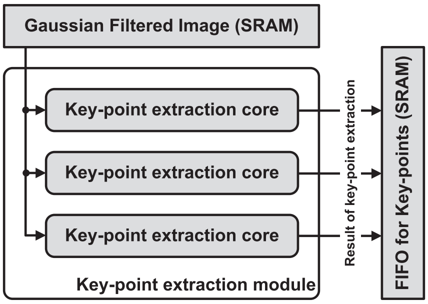

Gaussian Filtered Image (SRAM)
Key-point extraction core
Key-point extraction core
Key-point extraction core
Key-point extraction module
Result of key-point extraction
FIFO for Key-points (SRAM)

Fig. 16 Block diagram of key-point extraction module.

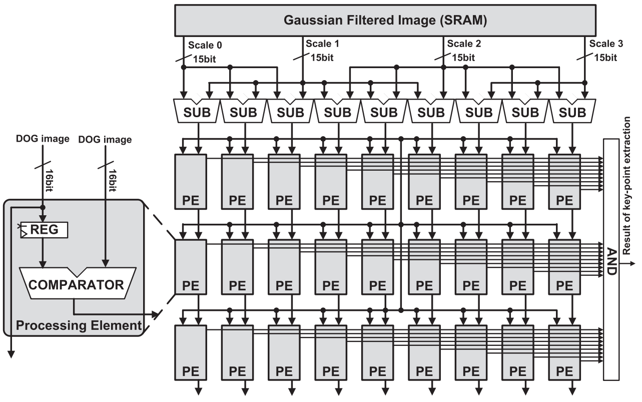


Fig. 17 Block diagram of key-point extraction core.

employed for the key-point extraction module. Figure 16 presents the whole diagram of the key-point extraction module. The datapath consists of three key-point extraction cores. The key-point extraction requires 27 pixels in three adjacent difference of Gaussian (DOG in Fig. 4). For that reason, the key-point extraction core comprises 27 processing elements (PE), as shown in Fig. 17. The input data for PE flow from top to bottom. Each PE reuses a pixel received from the upper PE. This scheme reduces the number of memory accesses to 34% of the former level.

3.4 Descriptor Vector Generation Architecture

Lastly, a descriptor vector generation module is described. Figure 18 depicts the block diagram of a descriptor vector generation module. This module comprises a sequencer module, a couple of orientation generators and a couple of descriptor vector generators.

The orientation generator is comprised of gradient generation module, a coordinate rotation digital computer (CORDIC) [14], and histogram generation module. It generates a gradient histogram of luminance. The gradient histogram is stored in a working memory for reuse in the descriptor vector generation block. Information about the orientation is derived from the gradient histogram.

The descriptor vector is calculated by using the information about orientation and the gradient histogram. First, the descriptor vector generator compensates the orientation of a key-point. Next, 16 histograms of eight directions are generated. Finally, 128-dimensional SIFT descriptor vectors for one key-point are obtained.

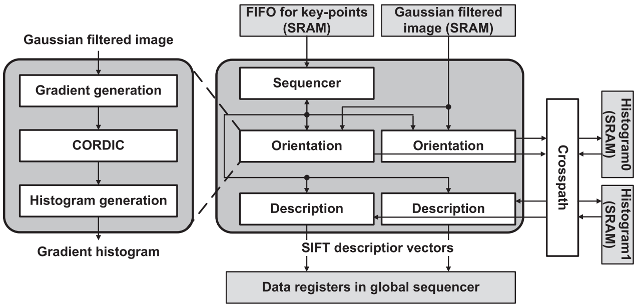


Fig. 18 Block diagram of descriptor vector generation module.

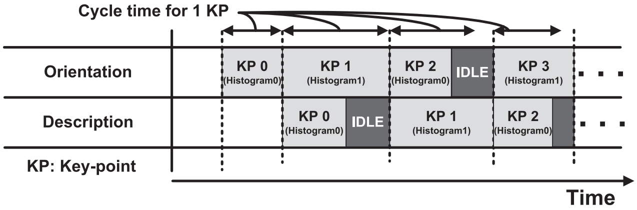


Fig. 19 Key-point level pipeline flow.

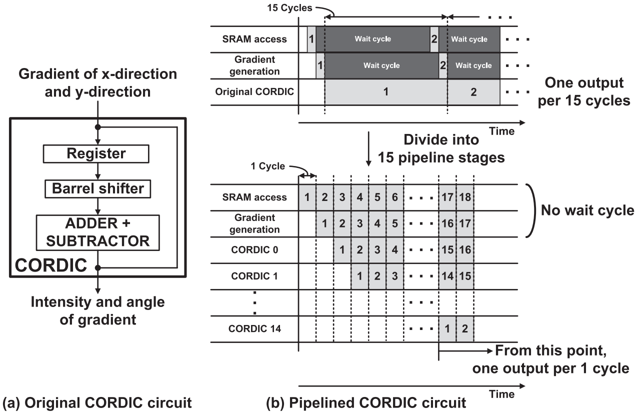


Fig. 20 (a) Original CORDIC circuit and (b) CORDIC circuit using pipeline.

The orientation generation and the descriptor vector generation are executed with key-point level pipeline operation as shown in Fig. 19.

The bottleneck of descriptor vector generation part is CORDIC function part. Figures 20(a) and (b) shows the original CORDIC circuit [14] and the operation of the pipelined CORDIC circuit, respectively. The original CORDIC circuit consists of a register, a barrel shifter, an adder and a subtractor. The original circuit results in low through-put and low memory-access rate because it takes 15 cycles to output one result and reads one gradient. Hence, the pipelined CORDIC circuit has been adopted for high-speed operation in our design. The pipelined circuit achieves one output per cycle.

3.5 Performance Evaluation

The number of cycle counts and memory accesses was estimated by using Verilog-HDL simulator. The test vectors are the images in the original database as shown in Fig. 10. The proposed architecture was compared with an assumed non-parallelized architecture and SIMD architecture. Table 1 summarizes the specifications of the non-parallelized architecture, the SIMD architecture and the proposed architecture. The estimation in the SIMD architecture was carried out on the assumption that input pixels are provided for all the processing units every one cycle.

Estimation results are summarized in Fig. 21 and Fig. 22, which demonstrates effects of the proposed architecture. The introduction of the proposed systolic array architecture into Gaussian filtering enables the reuse of intermediate results for the reduction in cycle count and memory access as shown in Fig. 9. Moreover, the parallelized architecture with image buffer reduces the overhead of memory read and enables real-time Gaussian filtering at HDTV resolution. On the other hand, the SIMD architecture has no reusability of intermediate results. In key-point extraction, the 81-pixel parallel systolic array architecture can make full use of pixel reusability in neighboring key-point extraction, allowing a drastic reduction of the memory access count.

As a result, the number of cycle counts in Gaussian filtering is reduced by 82%, compared with the SIMD architecture. The number of memory accesses in the Gaussian filtering and the key-point extraction are reduced by 99.8% and 66% respectively, compared with the results obtained using the SIMD architecture. Hence it is seen that the proposed architecture has resolved the three problems: (1) the workload of Gaussian filtering, (2) memory bandwidth of Gaussian filtering, and (3) the memory bandwidth of key-point extraction as described in Sect. 1.

Table 1 Architecture specifications.

	Non-parallelized architecture	SIMD architecture	Proposed architecture	
			w/ one-way systolic array	w/ multiple-way systolic array & image buffer
Gaussian filtering	One MAC	120-way SIMD	One-way systolic array	120-way systolic arrays with image buffer
Key-point extraction	One processing element	81-way SIMD	One-way systolic array	3-way systolic arrays
Descriptor Vector generation	One orientation block & one description block	One orientation block & one description block	One orientation block & one description block	Two orientation block & two description block

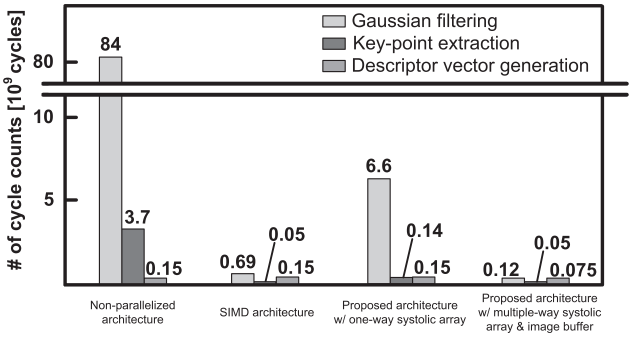


Fig. 21 Reduction of cycle count.

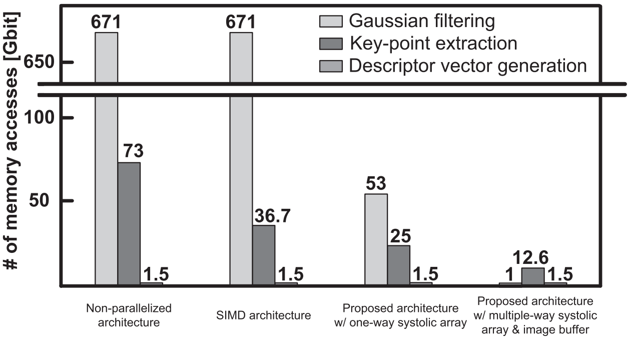


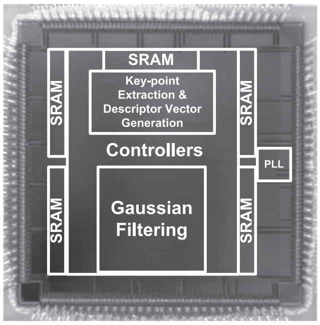
Fig. 22 Reduction of memory access.

4. VLSI Implementation

A test chip of SIFT descriptor generation has been designed as shown in Fig. 23. The design includes the VLSI oriented algorithm, the three-stage pipelined architecture and novel systolic array architectures. This chip has been fabricated in 65 nm CMOS technology and occupies $4.2 \times 4.2\,\mathrm{mm}^2$ containing 1.1M gates and 1.38 Mbit on-chip SRAM.

Figure 24 shows a measured data of power consumption versus operating frequencies. It is seen that 100 MHz operation was achieved at 1.1 V. Also the power consumption of the proposed processor is 38.2 mW at 78 MHz when a nominal supply voltage is 1.2 V, which attains real-time processing for HDTV resolution video with 30 fps of frame rate.

Figure 25 shows comparison to conventional processors. Reference [2] and Ref. [3] employed 180 nm technology and 130 nm technology, respectively. Thus, the performance of the conventional processors is scaled to 65 nm for fair comparison. This work succeeded in increasing frame rate to 53 times and reducing power consumption by 39%, compared with [2]. The simple comparison with [3] is difficult as the power consumption of [3] is comprised of three processes: pre-processing, SIFT descriptor gener-



Technology	65nm CMOS
Supply Voltage	Core 1.2V, I/O 3.3V
Chip Size	4.2mm x 4.2mm
Core Area	3.2mm x 3.4 mm
Logic Gate Count	1.1M gate
On-chip SRAM	1.38Mbit
Operating Frequency & Power Consumption	38.2mW@78MHz for 1920x1080(HDTV)/30fps
	14.5mW@27MHz for 640x480(VGA)/60fps
Target Application	SIFT descriptor generation

Fig. 23 Chip microphotograph.

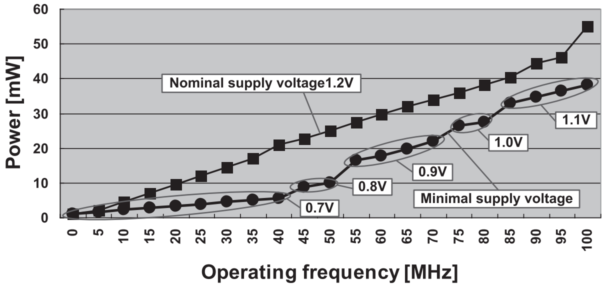
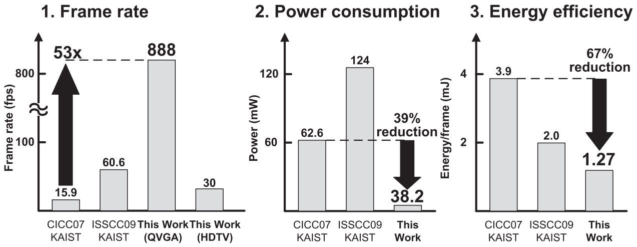


Fig. 24 Power consumption versus operating frequencies.



	KAIST, CICC2007 [2]		KAIST, ISSCC2009 [3]		This Work
	Original	Scaled to 65nm	Original	Scaled to 65nm	
Technology(nm)	180	65	130	65	65
Chip size (mm^2)	38.5	5	49	12.3	17.6
Power supply (V)	1.8	1.2	1.2	1.2	1.2
Operating frequency[MHz]	200		IP:200 NOC:400		78 (max:100)
Frame rate (fps)	15.9		60.6		30 (888@QVGA)
Power (mW)	1080	62.6	496	124	38.2
Energy/frame (mJ)	67.9	3.9	8.2	2	1.27
Video resolution	QVGA (320 x 240)		VGA (640 x 480)		HDTV (1920 x 1080)
Target Application	SIFT descriptor generation		Object recognition		SIFT descriptor generation

Fig. 25 Comparison to conventional processors with technology scaling.

ation and object recognition. However, it is seen that the proposed processor provides lower power consumption than [3] because SIFT descriptor generation dominates 63% of the whole workload in [3].

5. Conclusion

This paper proposes a novel VLSI design implementation of the SIFT descriptor generation. This implementation features a hardware-oriented algorithm, three-stage pipelined architecture, and parallel systolic array architectures. Evaluation of the fabricated VLSI demonstrates 39% power reduction of the real-time SIFT descriptor generation compared with that of the conventional processor [2]. Results show that the proposed architecture provides SIFT descriptors in real-time for image recognition system with HDTV resolution video. The design techniques described here are expected to be applied to several image recognition applications and specifically bring a big impact for mobile application under limited battery condition.

Acknowledgments

The VLSI chip in this study has been fabricated in the chip fabrication program of VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with STARC, e-Shuttle, Inc., and Fujitsu Ltd. This research has been supported by the Semiconductor Technology Academic Research Center (STARC). This development was performed by the author for STARC as part of the Japanese Ministry of Economy, Trade and Industry sponsored "Silicon Implementation Support Program for Next Generation Semiconductor Circuit Architectures." This work was supported by KAKENHI (18200003).

References

[1] D.G. Lowe, "Distinctive image features from scale invariant keypoints," Int. J. Comput. Vis., vol.60, no.2, pp.91–110, 2004.

[2] D. Kim, K. Kim, J.Y. Kim, S. Lee, and H. Yoo, "An 81.6 GOPS object recognition processor based on NoC and visual image processing memory," CICC, pp.443–446, Sept. 2007.

[3] J.Y. Kim, M. Kim, S. Lee, J. Oh, K. Kim, S. Oh, J.H. Woo, D. Kim, and H.J. Yoo, "A 201.4GOPS 496 mW real-time multi-object recognition processor with bio-inspired neural perception engine," ISSCC Dig., pp.150–151, Feb. 2009.

[4] V. Bonato, E. Marques, and G.A. Constantinides, "A parallel hardware architecture for scale and rotation invariant feature detection," IEEE Trans. Circuits Syst., vol.18, no.12, pp.1703–1712, Dec. 2008.

[5] L. Yao, H. Feng, Y. Zhu, Z. Jiang, D. Zhao, and W. Feng, "An architecture of optimised SIFT feature detection for an FPGA implementation of an image matcher," ICFPT 2009, 2009.

[6] K. Mizuno, H. Noguchi, G. He, Y. Terachi, T. Kamino, H. Kawaguchi, and M. Yoshimoto, "Fast and low-memory-bandwidth architecture of SIFT descriptor generation with scalability on speed and accuracy for VGA video," FPL, Aug. 2010.

[7] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," CVPR 2004, 2004.

[8] Changchang Wu, B. Clipp, X. Li, J.M. Frahm, and M. Pollefeys, "3D model matching with viewpoint-invariant patches (VIP)," CVPR 2008, 2008.

[9] S. Se, D. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," Int. J. Robot. Res., vol.21, no.8, pp.735–758, Aug. 2002.

[10] Ce Liu, J. Yuen, A. Torralba, J. Sivic, and W.T. Freeman, "SIFT flow: Dense correspondence across different scenes," ECCV 2008, 2008.

[11] R. Hess, "SIFT feature detector (source code)," 2007. Available: http://web.engr.oregonstate.edu/˜hess/

[12] J.Y. Kim, M. Kim, S. Lee, J. Oh, S. Oh, and H.J. Yoo, "Real-time object recognition with neuro-fuzzy controlled workload-aware task pipelining," IEEE Micro, vol.29, no.6, pp.28–43, Nov.-Dec. 2009.

[13] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," IEEE Trans. Pattern Anal. Mach. Intell., vol.27, pp.1615–1630, 2005.

[14] J.E. Volder, "The CORDIC trigonometric computing technique," IRE Trans. Electron. Comput., vol.EC-8, pp.330–334, 1959.

Kosuke Mizuno received the B.E. and M.E. degrees in Computer Science and Systems Engineering from Kobe University, Kobe, Japan in 2008 and 2010, respectively, where he is currently pursuing the Ph.D. degree in engineering. His current research interests include high-performance and low-power multimedia VLSI designs. He is a student member of IEEE.

Hiroki Noguchi respectively received B.E. and M.E. degrees in Computer and Systems Engineering in 2006 and 2008 from Kobe University, Hyogo, Japan, where he is currently pursuing the Ph.D. degree. His research interests include low-power SRAM designs for multimedia and ubiquitous media digital LSIs, mixed integer programming for the real time robots, and speech recognition algorithms and hardware implementation. He is a student member of IEEE.

Guangji He received the B.E. degree in electronic engineering from Beijing Institute of Technology, Beijing, China, in 2007. He is currently pursing the M.E. degrees in Computer Science and Systems Engineering from Kobe University, Kobe, Japan. His current research interests include high-performance and low-power multimedia VLSI designs.

Yosuke Terachi received the B.E. degree in Computer Science and Systems Engineering from Kobe University, Kobe, Japan in 2010. He is currently on the master course at Kobe University. Since 2009, he has been involved in the research and development of low-power image recognition VLSI designs.

Tetsuya Kamino received the B.E. and M.E. degrees in Computer Science and Systems Engineering from Kobe University, Kobe, Japan in 2008 and 2010, respectively. His research interests include high-performance and low-power image processing.

Tsuyoshi Fujinaga received the B.E. degree in Computer Science and Systems Engineering from Kobe University, Kobe, Japan in 2009. He is currently on the master course at Kobe University. Since 2008, he has been involved in the research and development of low-power speech recognition VLSI.

Shintaro Izumi received his B.E. and M.E. degrees in Computer Science and Systems Engineering from Kobe University, Kobe, Japan, in 2007 and 2008, respectively. Currently, he is a Ph.D. course student and a JSPS research fellow at Kobe University. His current research interests include communication protocol, low-power VLSI design, and wireless sensor network. He is a student member of the IEEE.

Yasuo Ariki received his B.E., M.E. and Ph.D. in information science from Kyoto University in 1974, 1976 and 1979, respectively. He was an assistant professor at Kyoto University from 1980 to 1990, and stayed at Edinburgh University as visiting academic from 1987 to 1990. From 1990 to 1992 he was an associate professor and from 1992 to 2003 a professor at Ryukoku University. Since 2003 he has been a professor at Kobe University. He is mainly engaged in speech and image recognition and interested in information retrieval and database. He is a member of IEEE, IPSJ, JSAI, ITE and IIEEJ.

Hiroshi Kawaguchi received the B.E. and M.E. degrees in electronic engineering from Chiba University, Chiba, Japan, in 1991 and 1993, respectively, and the Ph.D. degree in engineering from the University of Tokyo, Tokyo, Japan, in 2006. He joined Konami Corporation, Kobe, Japan, in 1993, where he developed arcade entertainment systems. He moved to the Institute of Industrial Science, the University of Tokyo, as a Technical Associate in 1996, and was appointed a Research Associate in 2003. In 2005, he move to the Department of Computer and Systems Engineering, Kobe University, Kobe, Japan, as a Research Associate. Since 2007, he has been an Associate Professor with the Department of Computer Science and Systems Engineering, Kobe University. He is also a Collaborative Researcher with the Institute of Industrial Science, the University of Tokyo. His current research interests include low-power VLSI design, hardware design for wireless sensor network, and recognition processor. Dr. Kawaguchi was a recipient of the IEEE ISSCC 2004 Takuo Sugano Outstanding Paper Award and the IEEE Kansai Section 2006 Gold Award. He has served as a Program Committee Member for IEEE Symposium on Low-Power and High-Speed Chips (COOL Chips), and as a Guest Associate Editor of IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences. He is a member of the IPSJ, IEEE and ACM.

Masahiko Yoshimoto received a B.S. degree in Electronic Engineering from the Nagoya Institute of Technology, Nagoya, Japan, in 1975, and an M.S. degree in Electronic Engineering from Nagoya University, Nagoya, Japan, in 1977. He received a Ph.D. degree in Electrical Engineering from Nagoya University, Nagoya, Japan in 1998. He joined the LSI Laboratory, Mitsubishi Electric Power Products Inc., Itami, Japan, in April 1977. During 1978–1983 he was engaged in the design of NMOS and CMOS static RAM, including a 64 K full CMOS RAM with the world's first divided-word-line structure. From 1984, he was involved in research and development of multimedia ULSI systems for digital broadcasting and digital communication systems based on MPEG2 and MPEG4 Codec LSI core technology. Since 2000, he has been a Professor of the Dept. of Electrical and Electronic Systems Engineering at Kanazawa University, Japan. Since 2004, he has been a Professor of the Dept. of Computer and Systems Engineering at Kobe University, Japan. His current activities specifically emphasize research and development of multimedia and ubiquitous media VLSI systems including an ultra-low-power image compression processor and a low-power wireless interface circuit. He holds 70 registered patents. He served on the Program Committee of the IEEE International Solid State Circuit Conference during 1991–1993. In addition, he has served as a Guest Editor for special issues on Low-Power System LSI, IP, and Related Technologies of IEICE Transactions in 2004. He received R&D100 awards from R&D Magazine in 1990 and 1996, respectively, for development of the DISP and development of a real-time MPEG2 video encoder chipset.