DOI 10.1007/s11265-014-0870-7

A Real-time Scalable Object Detection System using Low-power HOG Accelerator VLSI

Kenta Takagi · Kotaro Tanaka · Shintaro Izumi · Hiroshi Kawaguchi · Masahiko Yoshimoto

Received: 27 September 2013 / Revised: 1 January 2014 / Accepted: 7 January 2014 / Published online: 14 March 2014

Abstract As described in this paper, a real-time object detection system using a Histogram of Oriented Gradients (HOG) feature extraction accelerator VLSI is presented. The VLSI [1, 2] enables the system to achieve real-time performance and scalability for multiple object detection under limited power condition. The VLSI employs three techniques: a VLSI-oriented HOG algorithm with early classification in Support Vector Machine (SVM) classification, a dual-core architecture for parallel feature extraction, and a detection-window-size scalable architecture with a reconfigurable MAC array for processing objects of different shapes. The test chip was fabricated using 65 nm CMOS technology. The measurement result shows that the VLSI consumes 43 mW at 42.9 MHz and 1.1 V to process HDTV (1920 × 1080 pixels) at 30 frames per second (fps). A multiple object detection system and a multiple scale object detection system are presented to demonstrate the system flexibility and scalability realized by VLSI and applicability for versatile application of object detection. On the multiple object detection system, a real-time object detection for HDTV resolution video is achieved with 84 mW of power consumption on a task to detect 2 types of targets while keeping comparable detection accuracy as software-based system. On the multiple scale object detection system, a task to detect 5 scales of a target is accomplished using a single VLSI. The power consumption of the VLSI is estimated to 102 mW on the task.

Keywords HOG · Real-time object detection · VLSI · Multiple object detection

1 Introduction

Object detection is a crucial task for many computer vision applications such as surveillance, entertainment, automotive systems, and robotics. Application to automotive systems has entered the spotlight in recent years. The World Health Organization (WHO) predicted that traffic accident fatalities would reach 1.9 million per year worldwide by 2020 as the automobile ownership rate increases [3]. An attempt to assist drivers to be aware of dangers with computer vision technology is effective for avoiding fatal accidents. Techniques to detect various types of objects such as pedestrians, bikers, vehicles, and traffic signs have been investigated.

In any object detection system, feature extraction techniques are crucial to understand the characteristics of input images effectively. HOG [4], a widely accepted feature for object detection, is robust against changes of illumination, attaining high accuracy in the detection of variously textured objects. HOG is particularly effective and commonly used to detect humans and objects.

Recent progress in the area of high-performance general-purpose processors enables them to achieve real-time object detection. However, those processors require high power consumption, rendering them unsuitable for mobile systems, which have limited battery capacity and thermal design constraints. Moreover the detection performance is significant issue. One approach to improve system performance is to

K. Takagi (✉) · K. Tanaka · S. Izumi · H. Kawaguchi · M. Yoshimoto
Graduate School of System Informatics, Kobe University, Kobe, Japan
e-mail: takagi_k@cs28.cs.kobe-u.ac.jp

K. Tanaka
e-mail: tanaka@cs28.cs.kobe-u.ac.jp

S. Izumi
e-mail: shin@cs28.cs.kobe-u.ac.jp

H. Kawaguchi
e-mail: kawapy@godzilla.kobe-u.ac.jp

M. Yoshimoto
e-mail: yosimoto@cs.kobe-u.ac.jp

process high-resolution images. High-resolution images such as HDTV are more informative than lower-resolution images. For instance, HDTV images can cover a wider angle of view and examine finer details of objects. Similar to the problem in power consumption, even high-performance general-purpose processors can process only small amounts of high-resolution data such as those of HDTV in real time. Consequently, low-power and high-performance HOG feature extraction processors are necessary to expand the range of applications.

One example of the needs of object detection is relatively simple: to detect a single object or a fixed sized object. However, some applications require more complicated and difficult tasks. For example, drive-assist systems must detect not only pedestrians but also vehicles and bicycles simultaneously. Moreover, the relative positions between camera and detection targets change dynamically on the drive assist system. Changes in distance between the camera and the target produce changes in the appearance of the target object. Therefore, high flexibility and scalability to adapt to the changes in sizes, shapes and types of the target are necessary for object detection systems and object detection processors.

2 Rerated Works

Figure 1 presents the image resolution versus frame rates reported from several related works of HOG hardware. Zhang et al. [5] proposed object detection with GPGPU. Some FPGA implementations [6–11] and an FPGA–GPU architecture [12] have been proposed for real-time applications. A target-reconfigurable object detector for multiple object detection was proposed by Yazawa et al. [9]. However, reloading of parameters for other objects is necessary to detect other objects. Therefore, multiple objects cannot be detected simultaneously. Our previous work [13] on FPGA is superior to other works. However, it particularly targeted pedestrian detection. HOG features are adaptable to widely versatile applications. Therefore, anticipated HOG feature extraction processors must provide higher flexibility. Our goal is the development of design techniques for a real-time HOG feature extraction processor intended for use in multiple object detection from HDTV-resolution video.

The most common approach used in conventional processors is a window-based approach, for which the number of computations of 447.7 GOPS and memory bandwidth of 55 Gbps are necessary for HDTV resolution because of repetitive computations. Our previous work demonstrated that the computations and memory bandwidth are reduced considerably by the reuse of calculated data and the adoption of efficient computation methods [13]. However, power consumption remains high for mobile applications. Figure 2 portrays simulated power consumptions of the single/multiple object detection system using our previous processor [13]. The horizontal axis shows the number of the types of the target object. For example, 1 means human detection and 2 means human + vehicle detection simultaneously. The single-object detection using the previous FPGA-based processor consumes 196.99 mW on SVGA (800×600 pixels) resolution at 72 fps. The most dominant and second-most dominant activities are, respectively, cell histogram generation and SVM classification. The previous FPGA-based processor architecture does not support multiple object detection. Therefore multiple processors are necessary to detect multiple objects, consuming 200 mW times the number of objects. Extracting HOG features in every processor is redundant because the extracted features from a single image are the same. If the processors share the extracted feature, then the power consumption of the multiple object detection system is reduced as presented in Fig. 2. However, the power consumption in the classifier modules, the second most dominant part, increases linearly. Therefore, the power reduction technique should be applied to the classifier. The histogram generation module, the most dominant part, constantly consumes more than 100 mW. The module also requires a power reduction technique. Achieving low-power object detection demands

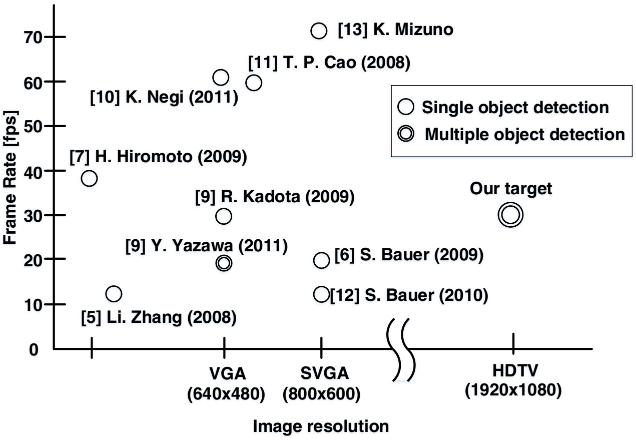


Figure 1 Previous works of HOG feature extraction processor.

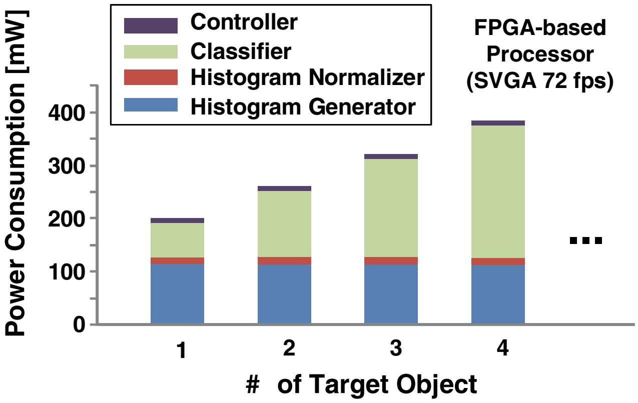


Figure 2 Estimated Power consumption of the previous FPGA-based processor [13] in the multiple object detection system.

improvement of the power efficiency of these two dominant processes.

3 Algorithm for the VLSI Processor

3.1 A Simplified HOG Algorithm

The VLSI we have developed [1] employs a simplified HOG algorithm for hardware implementation, as presented in Fig. 3. This algorithm is modified from the original algorithm [6]. The flow in Fig. 3 is based on our previous work [13]. We have confirmed that the simplified HOG algorithm reduces the implementation cost, maintaining the same detection accuracy as the original one on a Detection Error tradeoff simulation, as shown in Section 3.2. The simplified HOG algorithm employs the following seven techniques.

3.1.1 Cell-Based Scanning

Object detection systems with HOG feature commonly employ the sliding-window-based approach. The features are extracted within and classified by a detection window, while shifting the region of interest. The step size of 1 cell (8×8), which is a unit of HOG calculation, is used to scan input images densely [6].

When a processor scans and reloads the image and compute the feature based on the detection window, memory bandwidth and computational workload become high. These costs are the obstacles to implement the processor. However, most of the computations on the sliding-window-based approach are redundant, because most of the window areas overlap to the succeeding window. Once the calculations, which are described in following subsection 2–6, are executed within one window, the results are stored and re-used for the latter windows to reduce redundant workload. Moreover,

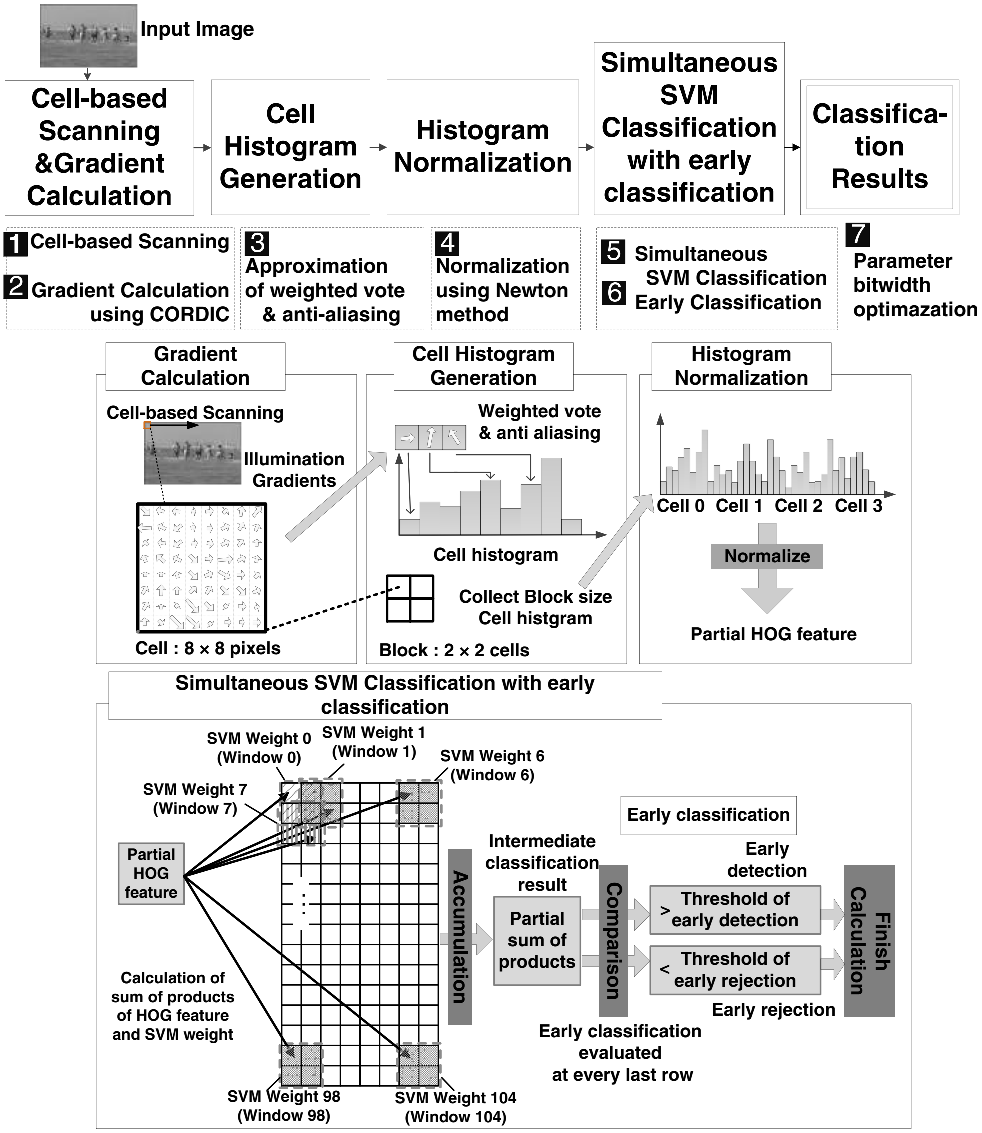

Input Image
Cell-based Scanning &Gradient Calculation
Cell Histogram Generation
Histogram Normalization
Simultaneous SVM Classification with early classification
Classifica-tion Results
1 Cell-based Scanning
2 Gradient Calculation using CORDIC
3 Approximation of weighted vote & anti-aliasing
4 Normalization using Newton method
5 Simultaneous SVM Classification
6 Early Classification
7 Parameter bitwidth optimazation
Gradient Calculation
Cell-based Scanning
Illumination Gradients
Cell : 8 × 8 pixels
Cell Histogram Generation
Weighted vote & anti aliasing
Cell histogram
Collect Block size Cell histgram
Block : 2 × 2 cells
Histogram Normalization
Cell 0 Cell 1 Cell 2 Cell 3
Normalize
Partial HOG feature
Simultaneous SVM Classification with early classification
SVM Weight 0 (Window 0)
SVM Weight 1 (Window 1)
SVM Weight 6 (Window 6)
SVM Weight 7 (Window 7)
Partial HOG feature
Calculation of sum of products of HOG feature and SVM weight
SVM Weight 98 (Window 98)
SVM Weight 104 (Window 104)
Accumulation
Intermediate classification result
Partial sum of products
Comparison
Early classification
Early detection
> Threshold of early detection
< Threshold of early rejection
Early rejection
Finish Calculation
Early classification evaluated at every last row

Figure 3 Simplified HOG algorithm flow [1].

memory bandwidth is greatly reduced by avoiding the reload of the entire data of the new window. Nevertheless, window-based reusing of the data requires large amounts of memory for buffering the input image and storing the calculation result. The memory capacity also depends on the window size. Therefore the window-based reusing is unsuitable for VLSI processor.

Our approach focuses on the cells. Scanning of the input image is based on cells. HOG features are extracted from cell-based calculations as in Fig. 3. The calculation results for one cell are stored and reused on the latter calculation stage. After scanning the data of each pixels within one cell, an adjacent cell is newly scanned in row-wise manner. No cell overlaps with other cells and it doesn't depend on the target window size. Sharing and reusing the results of the cell-based calculation greatly reduce the memory bandwidth. Moreover the cell-based approach requires 75 % less memory capacity than that of window-based-approach.

3.1.2 Gradient Calculation Using CORDIC

The CORDIC algorithm [14] is a well-known hardware-friendly method used to calculate trigonometric function because it merely requires addition, subtraction, bit shift, and table lookup. Illumination gradient orientation and magnitude is calculated using CORDIC.

3.1.3 Approximation of Weighted Voting for Histogram Generation

After calculating the illumination gradient within one cell, a weighted histogram of gradient orientation is generated. Each pixel within the cell votes a weighted gradient magnitude for a bin corresponding to its orientation. The weight for voting is calculated using the pixel's relative position in the cell, relative position among adjacent cells, and orientation. Weighted voting is necessary to avoid aliasing caused by the orientation binning. We approximated the weighting into bit shift operation.

3.1.4 Newton Method with Approximated Initial Values

Generated cell histograms are collected and normalized by much larger local regions called blocks (2×2 cells). To normalize the histogram, a reciprocal of L2-norm of the histogram is calculated using Newton's method. An initial value for Newton's method is obtained by approximation using bit shift operation. The approximation of the initial value reduces the necessary iteration count to reach convergence.

3.1.5 Simultaneous SVM Calculation

Figure 3 presents simultaneous SVM calculations for cell-based processing. Partial HOG features, which belong to 105 windows maximally, are located at different positions in each window. Partial HOG features are multiplied by the SVM coefficients of each window and are accumulated. The accumulation result is stored and reused in subsequent SVM calculations. Simultaneous SVM calculation is suitable for parallel computing in hardware.

3.1.6 Early Classification with the Accumulation Results

Before finishing all SVM calculations, data can be treated as they are already classified if the intermediate classification result is sufficiently low (or high), as portrayed in Fig. 4. Thereby, subsequent calculations can be skipped. Classifications on early stages are evaluated 14 times per detection window.

Pairs of preliminarily learned thresholds are used for the comparison. These represent possible misclassification intervals. The intervals are expressed as [μtar-4σtar, μnon+4σnon]. The means μtar, μnon and the standard deviations σtar, σnon are estimated, presuming that the intermediate classification results of a target class and a non-target class are normally distributed. Data within the two thresholds on an early stage are sent to the later stage and calculations are continued. Early classification reduces the number of computations by 22.3 % from 6.34 GOPS to 4.92 GOPS with no degradation of classification accuracy, on an Area Under the Curve simulation on the INRIA dataset [15].

3.1.7 Parameter Optimization

The precision of the approximations presented above and fixed-point operations depend on the bit width of the parameters. Optimizing the parameters reduces the required computing unit size and memory capacity.

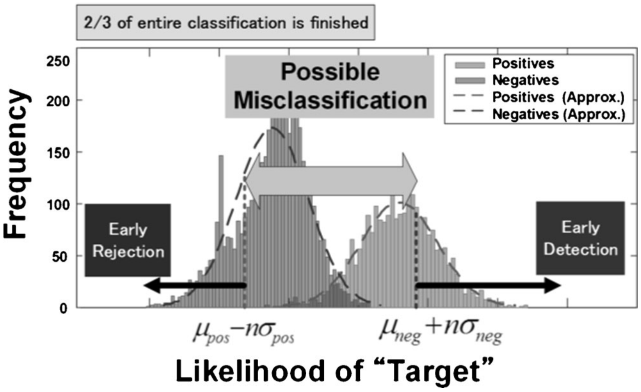


Figure 4 Distribution of intermediate classification result.

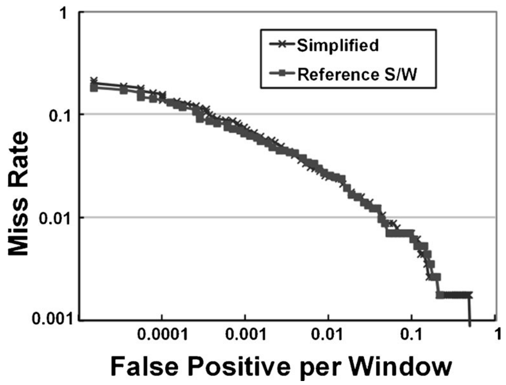


Figure 5 Detection error tradeoff evaluation on INRIA dataset.

3.2 Detection Performance Evaluation

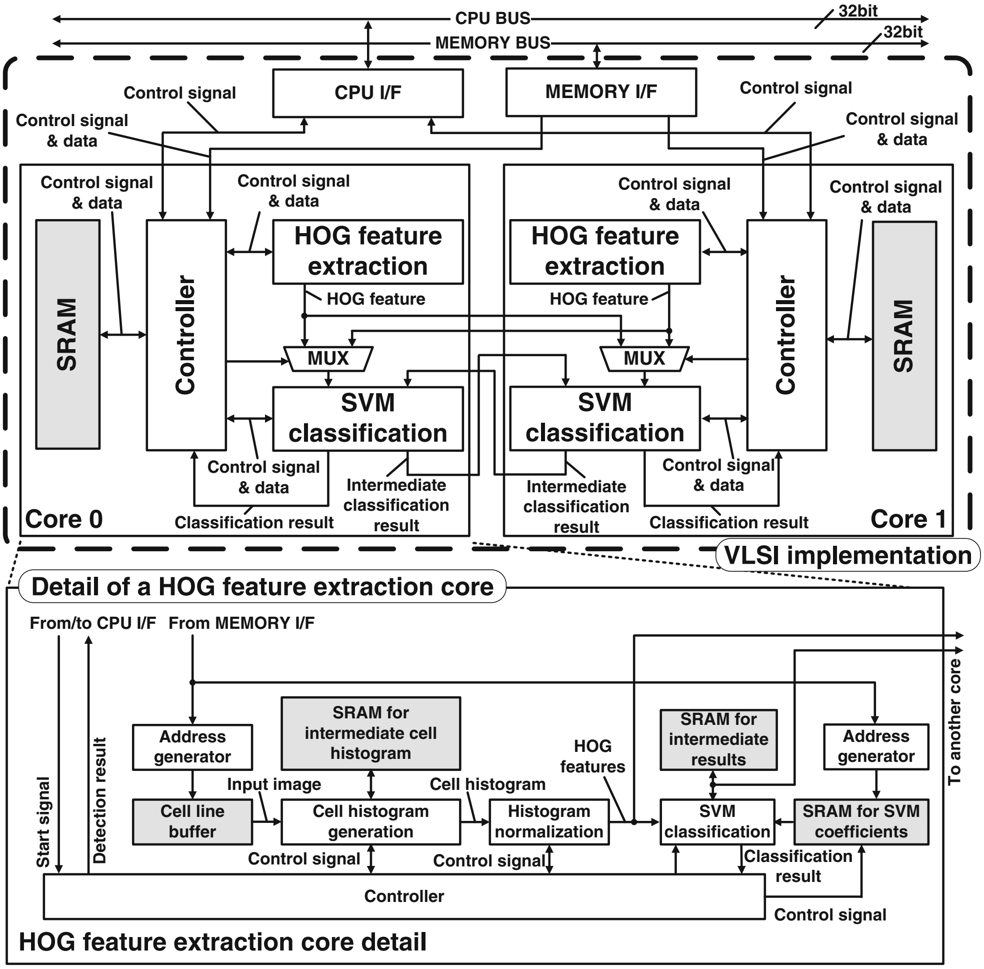
We have evaluated the detection performance of the simplified algorithm for VLSI. INRIA Person Dataset [15] is used to benchmark the algorithm. It contains non-human images and normalized 64 × 128 pixel human images. Figure 5 depicts tradeoff between False Positive per Window and Miss Rate. The simplified algorithm, which employs fixed point calculation, approximation methods, parameter optimization, and early classification, is comparable to the reference software implementation of the original HOG algorithm.

4 The VLSI Architecture

4.1 Dual Core Architecture with Cell-Based Pipeline

Figure 6 depicts a block diagram of the dual core architecture. The VLSI architecture consists of two HOG feature extraction cores, a CPU interface, and a memory interface. The HOG feature extraction core comprises a controller, address generators, cell histogram generation module, histogram normalization module, SVM classification module, and working SRAMs. An external CPU controls the HOG processor. The input grayscale image is loaded from external SRAM to internal SRAM via a memory interface.

Figure 6 Overall architecture of the HOG processor VLSI [1].



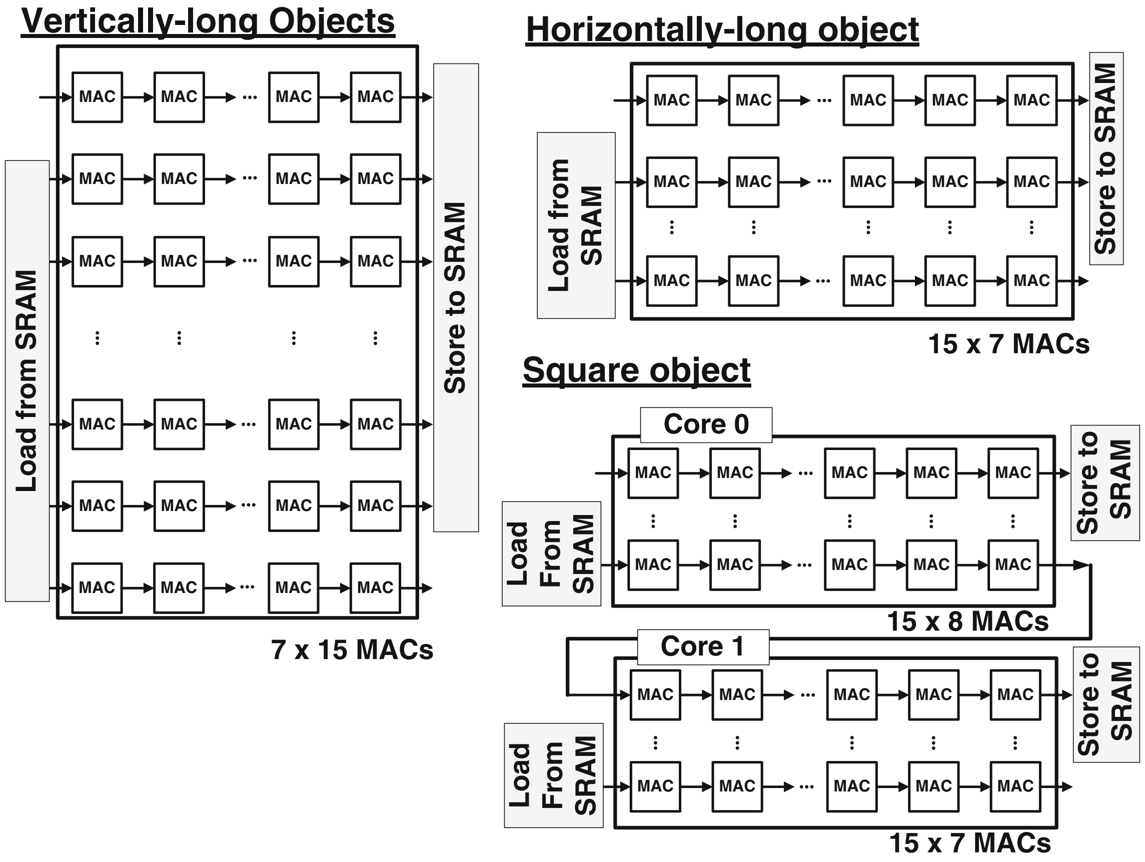


Figure 7 Reconfigurable MAC array module.

Four-way architecture is adopted to the cell histogram generation module architecture so that one cell is shared for four blocks. The cells have to be voted using different weights corresponding to their positions in the block. The cell histograms are generated in cell-based scanning manner. The intermediate histogram are stored in and loaded from a working SRAM.

The cell histogram normalization module adopts two-stage architecture to implement L2-Hys normalization [16]. Sub modules to obtain normalization coefficients adopt four-stage architecture to improve the precision using the Newton's method iteratively.

The SVM classification module adopts detection-window-size scalable architecture as described in Section 4.3. Furthermore, each core can share HOG features and intermediate classification results with another. The SVM classification module normally receives extracted feature from the cell histogram normalization module in the same core. The controller switches the multiplexer's inputs according to the flag in a configuration control register. The controller also enables the classification module to access the other classification module's output. This structure enables the processor to operate in several modes, as described in Section 4.2.

The HOG processor architecture has a cell-based pipeline flow. Cell-based pipeline processing is conducted along these five stages described below.

1. A cell histogram is generated with cell-based scanning. The generated cell histograms are stored into the working SRAM until 2×2 cell histograms are acquired.
2. When that process reaches the block level (2×2 cells) and four cell histograms are collected, a block-level cell

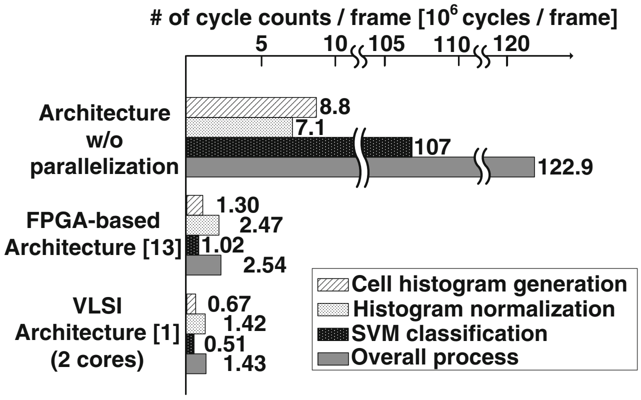


Figure 8 Reduction of cycle count for HDTV resolution [1].

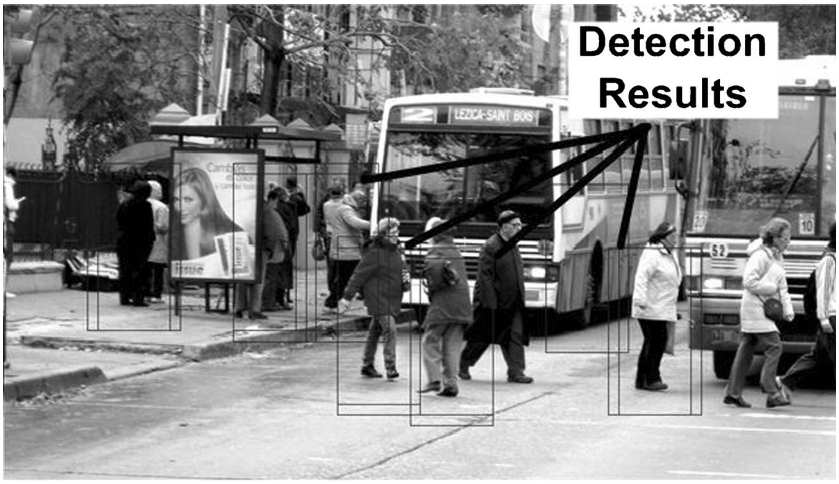


Figure 9 Pedestrian detection result with the VLSI [1].

histogram is normalized. Then the block-level HOG feature is extracted.

3. Block-level HOG features and its paired SVM coefficients corresponding to each window are multiplied and accumulated.
4. A partial sum of products is compared with early classification thresholds. A window is classified in the early stage if the comparison condition is true.
5. An accumulation result of the entire window level is compared with the SVM threshold. All remaining windows are classified based on this comparison. Then the final detection result is obtained.

The window-based approach requires memory band-width of 55 Gbps for HDTV images. The cell-based pipeline architecture reduces it markedly to 0.499 Gbps, thereby preventing

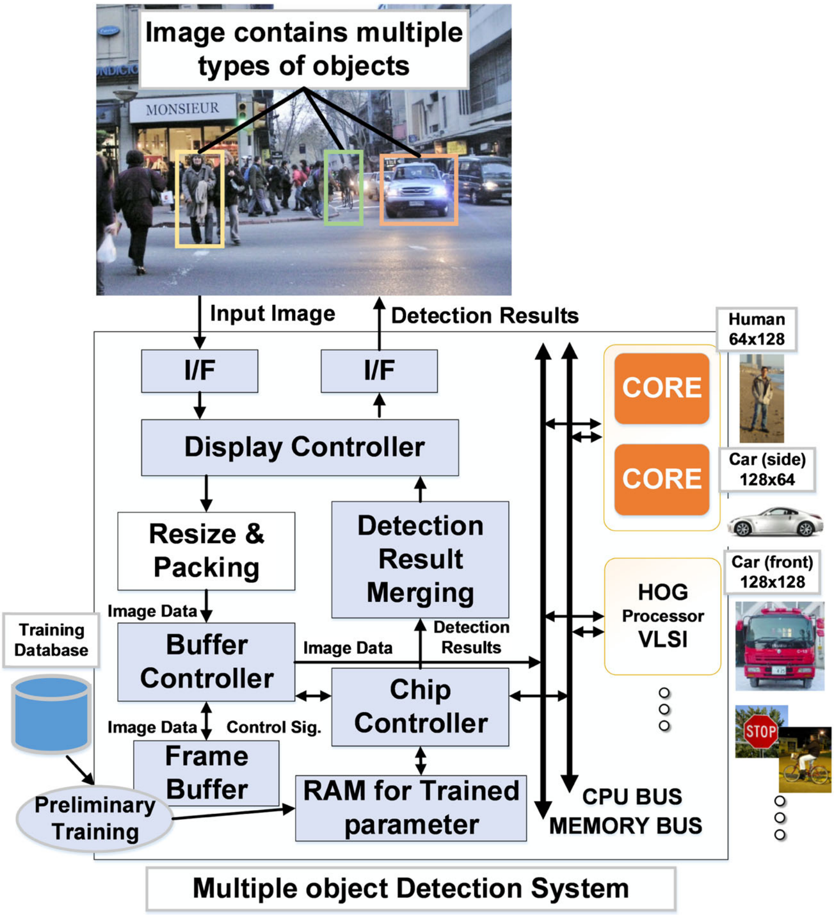


Figure 10 Multiple object detection system using the VLSI.

reloading of input pixels in different detection windows. However, the cell-based architecture requires circuit area overhead for extra SRAMs to store intermediate cell histograms and classification results.

4.2 Parallel Processing and Operating Modes

In the VLSI architecture, the required cycle count for object detection is reduced, sharing the workloads between the cores by dividing an image into two half-images for each core. Thereby, it achieves high-speed processing. In contrast, low-power processing is achieved by minimizing the operation frequency.

Detection for a single target object is insufficient for recent advanced applications. For example, on-vehicle applications must detect pedestrians, cars, and traffic signs. FPGA-based previous architecture [10] requires another processor to detect another target. Furthermore, it wastes power for extraction of the same HOG feature, although feature extraction is the dominant part of the object detection task.

The SVM classification module in the processor core contains an independent SRAM dedicated for SVM coefficients. Each core stores different SVM coefficients for different objects. Sharing HOG features to another core, feature extraction processes in one core can be turned off completely to reduce power consumption.

4.3 Detection-Window-Size Scalable Architecture

The classification module is comprised of 8 × 15 MAC arrays (a last column is only used to detect square shaped object) as illustrated in Fig. 7. These MACs are connected row-wise to classify vertically-long rectangular object in default configuration. Firstly the module receives a partial HOG feature and multiplies to correspond SVM coefficient. Then the data is shifted to neighboring MAC as the cell-based scan goes on in row-wise manner. The intermediate classification data is stored in a SRAM after passing the last MAC in a row. Then the data is loaded from SRAM to the MAC when the process comes to the next row. The partial HOG features are sent one after another due to the pipeline processing, every MAC excepting the last column are active. Therefore, 105 detection-windows are classified at once.

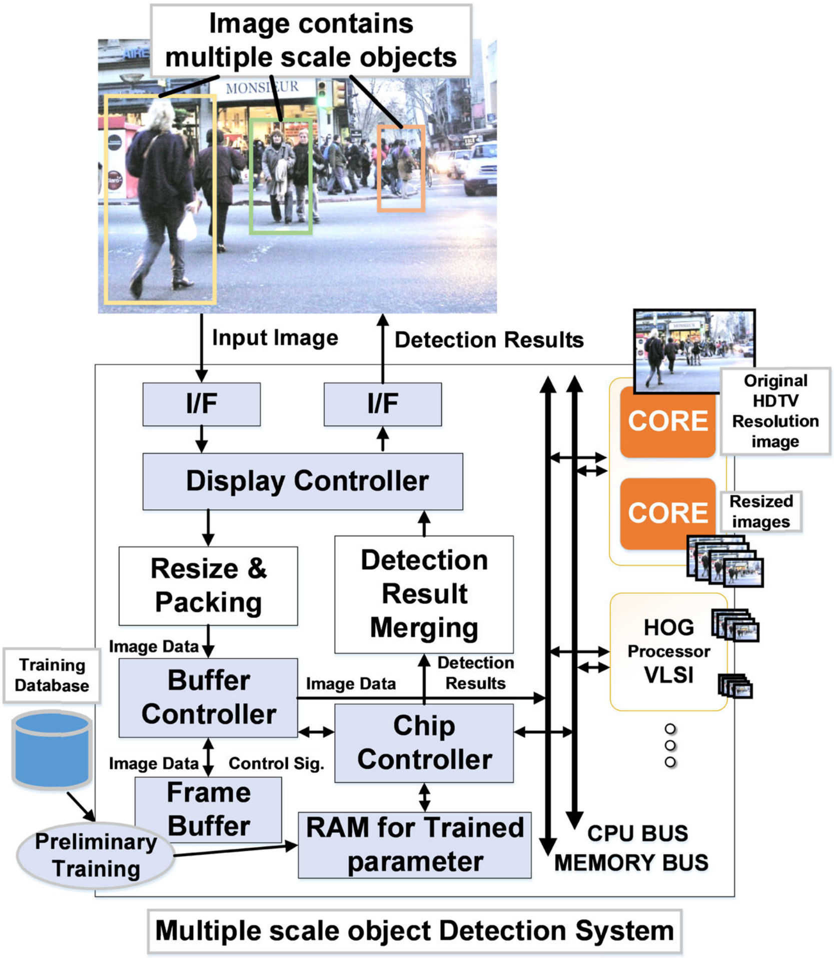


Figure 11 Multi-scale object detection using the VLSI.

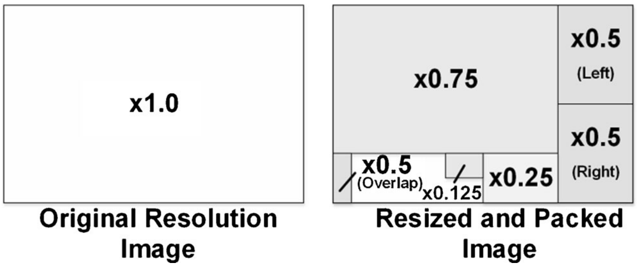


Figure 12 Example of a packed image.

The architecture also provides object detection for different shapes: square objects and vertically/horizontally long rectangular objects. The controller module automatically changes the direction of the dataflow in the classification module according to the flag in configuration register. For horizontally-long object, the MACs are reconfigured by connecting MACs in column-wise.

In order to detect square object, the MAC array modules in each core process the data of each rectangular region. An intermediate result of core0 is loaded into core1 as an initial value. This module structure allows a greater flexibility for multiple object detection.

4.4 Performance Evaluation of the Architecture

The number of cycle counts for each calculation in HOG-based object detection was estimated using a Verilog-HDL simulator. A comparison among the VLSI architecture [1], the previous FPGA-based architecture [13] and architecture without parallelization is presented in Fig. 8.

The VLSI architecture is superior to others on HDTV resolution. Results show that the processing with two cores reduces the number of cycle counts compared with previous FPGA-based architecture. The reduction rates of histogram generation, histogram normalization, and SVM classification are, respectively, 48.5 %, 42.5 %, and 50 %. In the dual core VLSI architecture, the overall process requires 1.43×10^6 cycles per frame. Therefore, the VLSI can process HDTV resolution video at 30 fps with 42.9 MHz.

5 Scalable System Architecture

As an example of the applications of the HOG processor VLSI, we developed a pedestrian detection demonstration system [1]. Figure 9 presents a sample image of the detection.

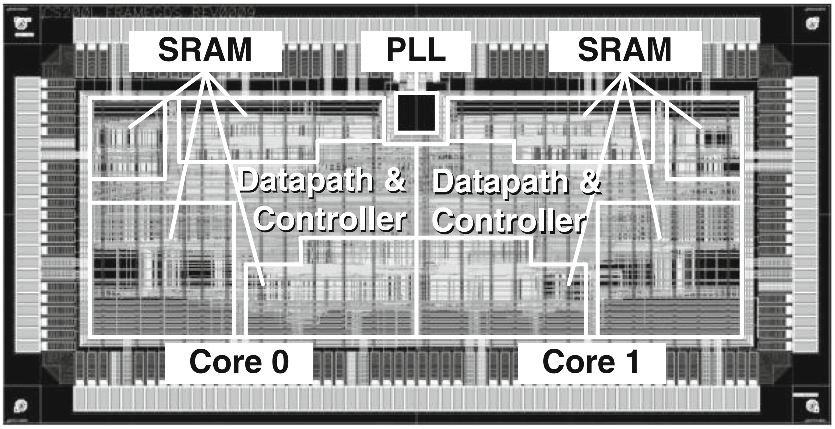
Following Sections 5.1 and 5.2 describes how to achieve more practical object detection system using the VLSI. Section 5.1 focuses on detection of multiple type objects. Section 5.2 focuses on multiple scale object detection. Both systems employ the same baseline architecture. The number of target type and the searching scale are easily scaled up in the system. The system can perform multiple object detection and multiple scale object detection at the same time. The system configurations and the role of the each core in the VLSIs are described in 5.1 and 5.2.

5.1 The Application to Multiple Object Detection

The previous FPGA processor [13] supports only single-object detection. If we use two or more processors to detect multiple objects simultaneously, then a system can process only one specific shaped object (64×128 pixel). The capability and flexibility of the system are limited. However, as presented in Fig. 10, an object detection system using the VLSI can assign different shaped objects to each core, such as humans and cars, across in front of the camera.

The system consists of the HOG VLSIs and a FPGA board that a chip controller and other peripheral modules are implemented in. The VLSIs are connected to chip controller and buffer controller via a CPU bus and a memory bus respectively. The data flow of the system is described below. Firstly the image is captured into the system through the DVI interface module. Display controller module manages the timing of the capture of input image. The captured image is converted into

Figure 13 Chip layout [1].



grayscale image and sent to the Resize and Packing module. It is not utilized for this application, but for multiple scale detection. Then the data are sent to the frame buffer controller. It controls to transfer the image data to the VLSIs. The Chip Controller module manages the communication between VLSIs and frame buffer and operating status of the VLSIs. To detect multiple types of objects, the trained parameter sets correspond to each target for SVM classification modules are required. The parameters are obtained by offline training of the image dataset. The trained parameters are loaded into each core in the VLSIs through the chip controller. The Chip controller also set the appropriate values corresponding to the detection target into the configuration registers in each core. Detection results are sent back to the controller once any processor detects the target. The results are collected and merged. Then the display controller renders the merged detection result.

On the detection system using the HOG VLSI, early classification architecture and feature sharing architecture reduce the power consumption as shown in Section 6. The detection system can easily scales up the number of the target objects by adding more VLSIs to the system. The power consumption and detection accuracy are discussed in Section 7.

5.2 Application for Multiple-scaled Object Detection

As described earlier, the FPGA-based architecture that supports fixed-scale object doesn't work well on applications for which the relative position between the camera and detection target changes dynamically. The changes in distance between the camera and the target produce changes in the target object appearance.

The system which simply assigns one resized image to one processor consumes wasted processing power. Figure 11 portrays a multiple scale object detection system using the VLSI. On this system, captured input image is sent and resized and combined into one image.

One core in the VLSI processes the original HDTV resolution image. This core uses the original image to examine fine scale objects. The other core processes the resized or subsampled images. Resized images are smaller than the original HDTV resolution image. And each core in the VLSI is design to handle the HDTV resolution. Therefore, processing only one resized image in one core makes a vacancy area in the internal buffer and the core has surplus processing ability.

Packing the multiple resized images, as presented in Fig. 12, uses the surplus memory and processing power effectively. The example presented in Fig. 11 shows images that have been resized into×0.75, ×0.5, ×0.25, and×0.125 scales are packed into it as if it were one image. The×0.5 scaled image is packed into half-sized images. To avoid misdetection, the boundary region is additionally processed. The system can process these four scaled images and an original image within a single VLSI. The Chip Controller receives the detection results from each core. The detection results include whether the target is detected or not, detected positions and its likelihood values. The detection results for any scales and positions are sent to the Detection Result Merging module. Overlapping detections at nearby scales and positions are merged to the most reasonable object by the merging module. The multiple scale object detection system using the VLSI can scale up its detection reliability according to the applications' demand using multiple VLSI processors to examine different scales. The system using the HOG processor VLSI has high scalability for the number of the object's scales, maintaining an advantage in power consumption.

Table 1 Chip specifications.

Technology	65 nm CMOS
Chip size	4.2×2.1 mm2
Core size	3.3×1.2 mm2
Power supply	1.1 V (nominal)
Max frequency	110 MHz
Gate count	502 Kgates
Memory size	1.22 Mbit (610 Kbit for one core)
Image resolution	HDTV (1920×1080 pixels) @ 30 fps
Measured power	43.0 mW @ 42.9 MHz 1.1 V
Consumption	17.4 mW @ 42.9 MHz 0.7 V (min)

6 VLSI Measurement Result

A test chip was designed as presented in Fig. 13 [1]. The design includes the VLSI-oriented algorithm and a dual-core

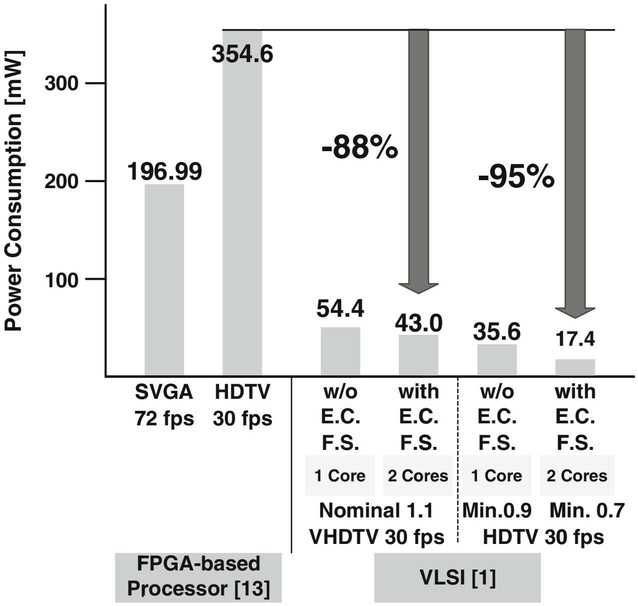


Figure 14 Power consumption of the FPGA-based processor and the VLSI.

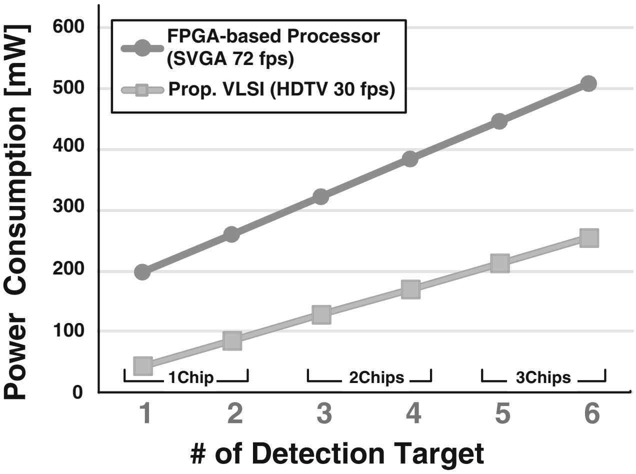


Figure 15 Power consumption of the VLSI in the multiple object detection system.

architecture. This chip, which was fabricated in 65 nm CMOS technology, occupies 4.2×2.1 mm2 containing 502 Kgates and 1.22 Mbit on-chip SRAMs. Chip specifications are presented in Table 1. We analyzed the power consumption of the VLSI using an LSI test system with a test pattern that simulates a human detection task on HDTV resolution image. Figure 14 is the power consumption comparison between our previous FPGA-based processor and the VLSI. The power consumption of the VLSI is measured at nominal operation voltage 1.1 V and minimum operation voltage. Our FPGA-based architecture consumes 196.99 mW when it processes an SVGA resolution (800×600 pixels) image at 72 fps. Although the processor cannot handle the HDTV in real time, its estimated power consumption is expected to reach 354.6 mW to process HDTV at 30 fps. The measurement result shows that the power consumption of the VLSI is reduced considerably from that of the previous FPGA-based device. The value w/o E.C. and F.S. means that the test chip was operated with neither the Early Classification function nor Feature Sharing function. Furthermore, under those circumstances, only one core in the VLSI is active. To process HDTV at 30 fps with single core, 84.3 MHz of operating frequency is necessary. The minimum voltage is 0.9 V at the single core operation. The measured values of power consumption at nominal voltage and minimum voltage were 54.4 mW and 35.6 mW, respectively. The value with E.C. and F.S. means that the test chip was operated with Early Classification function and Feature Sharing function. Under these conditions, both cores in the VLSI are active. To process HDTV at 30 fps with dual core processing, 42.9 Mhz of operating frequency is necessary. The operating frequency is not simply halved: each core must process a half-size image and small overlapped region to avoid misdetection near the boundary. The minimum voltage is lowered to 0.7 V at the dual core operation. The measured values of power consumption at nominal voltage and minimum voltage were 43.0 mW and 17.4 mW, respectively. Results of comparison show that the VLSI with power reduction techniques at nominal voltage reduces the power consumption 88 % from FPGA-based processor, with reduction of 95 % at minimum voltage.

7 System Evaluation

7.1 Power Consumption

Figure 15 presents a comparison of the power consumption of the processors in the multiple object detection systems. One is the estimated power consumption when the FPGA-based processor detects the object on SVGA resolution. The other is the estimated power consumption in the VLSI-based system with HDTV resolution video. If the system processes target objects of two types, then the power consumption of the VLSI-based system is 84 mW. It is 67 % lower than that of the FPGA-based system. The HOG VLSI employs dual core architecture. Therefore up to two target objects are assigned and detected per chip. The system can scale up the number of the target objects using multiple VLSI chips. The system power consumption increases as the number of target objects increases. However, the increase rate in the VLSI-based system is 32 % lower than that of the FPGA-based system.

Figure 16 presents a comparison of the power consumption of the processors in the multiple scale object detection systems. In Fig. 16, one is the FPGA-based system on SGVA and the other is the VLSI-based system. Both systems employ image packing described in the Section 5.2. The VLSIs can process 5 scales per chip. The power consumption of the VLSI-based system is 102 mW. It is 74 % lower than that of the FPGA-based system.

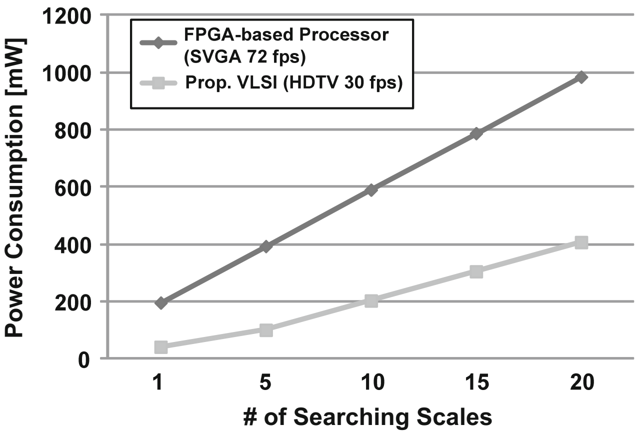


Figure 16 Power consumption of the VLSI in the multiple scale object detection system.

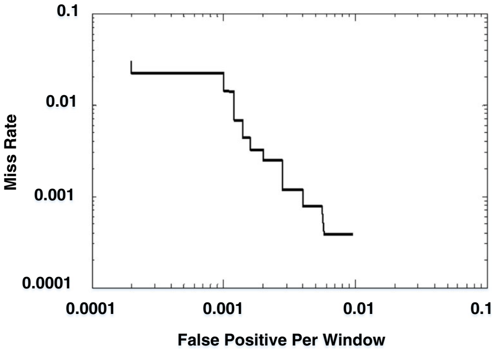


Figure 17 Detection accuracy on vehicle detection.

7.2 Detection Accuracy

The detection accuracy of the VLSI on pedestrian detection is shown in Fig. 5 in Section 3.2. The VLSI can perform pedestrian detection accurately. However, the capability of detecting objects other than humans is an important issue. The previous FPGA-based architecture supported only human detection application. Therefore, parameter optimization such as bit-width optimization is specified for human detection.

To verify the effectiveness and adaptability of the VLSI to other object detection task, we have benchmarked the VLSI with a vehicle detection dataset. Several datasets are available on the internet. A GTI's Vehicle Image Database [17], which contains 4000 vehicles' rear images and 4000 non-vehicle images. The images were acquired from video sequences that have been captured from vehicle mounted camera. The data are normalized into a 64×64 pixel format. We evaluated the VLSI architecture by average of five hold-out cross validation. Figure 17 presents the Detection Error Tradeoff Curve on the vehicle detection using the VLSI. The result shows that both the miss rate and false positives are lower than the result of the human detection task. The classification accuracy using the VLSI was about 95 %, which is compatible with other software-based detection systems benchmarked on the same dataset [18].

8 Conclusion

A real-time object detection system using the HOG feature extraction accelerator VLSI was presented in this report. The VLSI architecture, fabricated using 65 nm CMOS technology, employs a simplified HOG algorithm with early classification, a dual-core architecture with a cell-based pipeline, and a detection-window-size scalable architecture. Measurement results show that the VLSI consumes 43 mW at 42.9 MHz and 1.1 V to process HDTV resolution video at 30 fps. The dual core architecture and detection-window-size scalable architecture enables the dual cores to operate collaboratively. This architecture provides high scalability for multiple object detection. The object detection system using the VLSI can be easily adapted to different size and types of the target objects. In cases of human detection and vehicle detection, benchmarked results show that the detection accuracy obtained by the multiple object detection system using the VLSI is comparable to that by other software-based systems.

Consequently, a multiple object detection system using VLSI presents great advantages in real-time performance and available resolution. It can easily scale up the number of target objects and searching scales answering the various needs of applications without any great increase in power consumption.

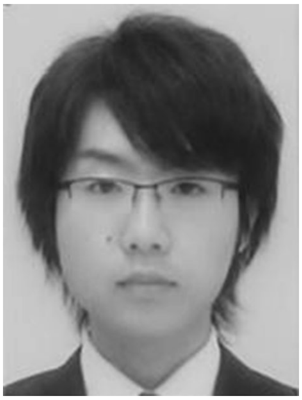
It satisfies the demands of recent advanced applications such as on-vehicle applications and intelligent robots.

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".

References

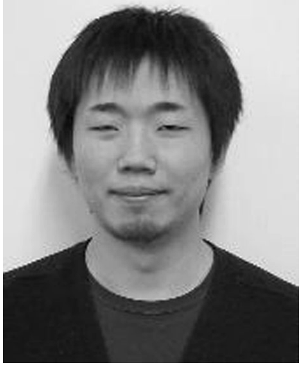
1. Takagi, K., et al. (2013) A SUB-100MW dual-core HOG accelerator VLSI for real-time multiple object detection, *IEEE International Conference on Acoustics, speech, and Signal Processing (ICASSP)*.
2. Mizuno, K., Takagi, K., et al. (2013). A sub-100mW dual-core HOG accelerator VLSI for parallel feature extraction processing for HDTV resolution video. *IEICE Transactions on Electronics, E96-C*(4).
3. World Health Organization "Decade of Action for Road Safety 2011-2020: saving millions of lives", May 2011.
4. Dalal, N., & Triggs, B. (2005) Histograms of oriented gradients for human detection, in *Proceedings of the 2005 International Conference on Computer Vision and Pattern Recognition*, vol. 2. Washington, DC, USA: IEEE Computer Society, pp. 886–893.
5. Zhang, L., & Nevatia, R. (2008) Efficient scan-window based object detection using GPGPU, *IEEE, CVPRW.*
6. Bauer, S., Brunsmann, U., Schlotterbeck-Macht, S. (2009) FPGA Implementation of a HOG-based pedestrian recognition system, *MPC-Workshop*, July 2009.
7. Hiromoto, M., & Miyamoto, R. (2009) Hardware architecture for high-accuracy real-time pedestrian detection with CoHOG Features, *IEEE ICCVW.*
8. Kadota, R., Sugano, H., Hiromoto, M., Ochi, H., Miyamoto, R., Nakamura, Y. (2009) Hardware architecture for HOG feature extraction, in Proceedings of the 2009 International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Washington, DC, USA: IEEE Computer Society, pp. 1330–1333.
9. Yazawa, Y., Yoshimi, T., Tsuzuki, T., Dohi, T., Fujiyoshi, H. (2011) FPGA Hardware with target-reconfigurable object detector by Joint-HOG, in *Proceeding of SSII.* Yokohama, Japan.
10. Negi, K., Dohi, K., Shibata, Y., Oguri, K. (2011) Deep pipelined one-chip FPGA implementation of a real-time image-based human detection algorithm, *IEEE FPT* 2011.

11. Cao T. P., & Deng, G. (2008) Real-time vision-based stop sign detection system on FPGA", in Proceeding of Digital Image Computing: Techniques and Applications. Los Alamitos, CA, USA: IEEE Computer Society, pp. 465–471, 2008.
12. Bauer, S., Kohler, S., Doll, K., Brunsmann, U. (2010) FPGA-GPU Architecture for Kernel SVM Pedestrian Detection, *IEEE CVPRW* 2010.
13. Mizuno, K., Terachi, Y., Takagi, K., Izumi, S., Kawaguchi, H. Yoshimoto, M. (2012) Architectural study of HOG feature extraction processor for real-time object detection", *IEEE SiPS*.
14. Volder, J. E. (1959). The CORDIC trigonometric computing technique. *IRE Trans. Electron. Computers., EC-8*, 330–334.
15. INRIA Person Dataset. http://pascal.inrialpes.fr/data/human/
16. Lowe, D. G. (2004). Distinctive image features from scale invariant keypoints. *International Journal of Computer Vision, 60*(2), 91–110.
17. GTI's Vehicle Image Database. http://www.gti.ssr.upm.es/data/Vehicle_database/Vehicle_database.html
18. Arróspide, J., Salgado, L., & Nieto, M. Video analysis based vehicle detection and tracking using an MCMC sampling framework. *EURASIP Journal on Advances in Signal Processing, 2*.



Kenta Takagi received the B.E. degree in Computer Science and Systems Engineering from Kobe University, Kobe, Japan in 2012. He is currently on the master course at Kobe University. His current research is a low-power image recognition VLSI designs. He was a recipient of IEEE SSCS 2013 Japan Chapter Academic Research Award. He is a student member of IEEE, and IEICE.

Kotaro Tanaka received the B.E. degree in Communication Network Engineering from Okayama University, Okayama, Japan in 2013. He is currently on the master course at Kobe University. His current research is real-time image recognition processor designs.



Shintaro Izumi respectively received his B.Eng. and M.Eng. degrees in Computer Science and Systems Engineering from Kobe University, Hyogo, Japan, in 2007 and 2008. He received his Ph.D. degree in Engineering from Kobe University in 2011. He was a JSPS research fellow at Kobe University from 2009 to 2011. Since 2011, he has been an Assistant Professor in the Organization of Advanced Science and Technology at Kobe University. His current research interests include biomedical signal processing, communication protocols, low-power VLSI design, and sensor networks. He is a member of the IEEE, IEICE, and IPSJ.

?

Hiroshi Kawaguchi received B.Eng. and M.Eng. degrees in electronic engineering from Chiba University, Chiba, Japan, in 1991 and 1993, respectively, and earned a Ph.D. degree in electronic 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 as a Research Associate in 2003. In 2005, he moved to Kobe University, Kobe, Japan. Since 2007, he has been an Associate Professor with The Department of Information Science at that university. He is also a Collaborative Researcher with The Institute of Industrial Science, The University of Tokyo. His current research interests include low-voltage SRAM, RF circuits, and ubiquitous sensor networks. 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 Design and Implementation of Signal Processing Systems (DISPS) Technical Committee Member for IEEE Signal Processing Society, as a Program Committee Member for IEEE Custom Integrated Circuits Conference (CICC) and IEEE Symposium on Low-Power and High-Speed Chips (COOL Chips), and as an Associate Editor of IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences and IPSJ Transactions on System LSI Design Methodology (TSLDM). He is a member of the IEEE, ACM, IEICE, and IPSJ.

Masahiko Yoshimoto joined the LSI Laboratory, Mitsubishi Electric Corporation, Itami, Japan, in 1977. From 1978 to1983 he had been engaged in the design of NMOS and CMOS static RAM. Since 1984 he had been involved in the research and development of multimedia ULSI systems. He earned a Ph.D. degree in Electrical Engineering from Nagoya University, Nagoya, Japan in 1998. Since 2000, he had been a professor of Dept. of Electrical & Electronic System Engineering in Kanazawa University, Japan. Since 2004, he has been a professor of Dept. of Computer and Systems Engineering in Kobe University, Japan. His current activity is focused on the research and development of an ultra low power multimedia and ubiquitous media VLSI systems and a dependable SRAM circuit. He holds on 70 registered patents. He has served on the program committee of the IEEE International Solid State Circuit Conference from 1991 to 1993. Also he served as Guest Editor for special issues on Low-Power System LSI,IP and Related Technologies of IEICE Transactions in 2004. He was a chair of IEEE SSCS (Solid State Circuits Society) Kansai Chapter from 2009 to 2010. He is also a chair of The IEICE Electronics Society Technical Committee on Integrated Circuits and Devices from 2011 to 2012. He received the R&D100 awards from the R&D magazine for the development of the DISP and the development of the real time MPEG2 video encoder chipset in 1990 and 1996, respectively. He also received 21th TELECOM System Technology Award in 2006.