PAPER *Special Section on Solid-State Circuit Design—Architecture, Circuit, Device and Design Methodology*

# A Sub-100 mW Dual-Core HOG Accelerator VLSI for Parallel Feature Extraction Processing for HDTV Resolution Video

**Kosuke MIZUNO**[†a)], ***Student Member***, **Kenta TAKAGI**[†], **Yosuke TERACHI**[†], ***Nonmembers***, **Shintaro IZUMI**[†], **Hiroshi KAWAGUCHI**[†], ***and*** **Masahiko YOSHIMOTO**[†], ***Members***

**SUMMARY** This paper describes a Histogram of Oriented Gradients (HOG) feature extraction accelerator that features a VLSI-oriented HOG algorithm with early classification in Support Vector Machine (SVM) classification, dual core architecture for parallel feature extraction and multiple object detection, and detection-window-size scalable architecture with reconfigurable MAC array for processing objects of several shapes. To achieve low-power consumption for mobile applications, early classification reduces the amount of computations in SVM classification efficiently with no accuracy degradation. The dual core architecture enables parallel feature extraction in one frame for high-speed or low-power computing and detection of multiple objects simultaneously with low power consumption by HOG feature sharing. Objects of several shapes, a vertically long object, a horizontally long object, and a square object, can be detected because of cooperation between the two cores. The proposed methods provide processing capability for HDTV resolution video (1920 × 1080 pixels) at 30 frames per second (fps). The test chip, which has been fabricated using 65 nm CMOS technology, occupies 4.2 × 2.1 mm$^2$ containing 502 Kgates and 1.22 Mbit on-chip SRAMs. The simulated data show 99.5 mW power consumption at 42.9 MHz and 1.1 V.

***key words:*** *HOG, object detection, low-power, HDTV*

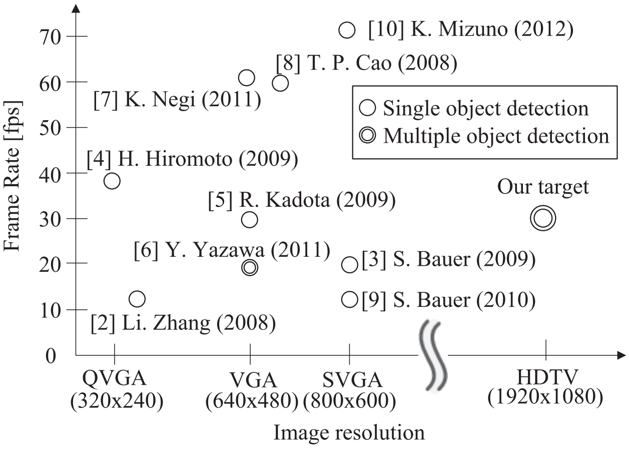


**Fig. 1** Previous works of HOG feature extraction processor.

## 1. Introduction

Object detection from visual images is an important task for many computer vision applications such as surveillance, entertainment, automotive systems, and robotics. An important algorithm used in object detection systems, Histogram of Oriented Gradients (HOG) [1], has robustness to change of illumination and attains high computational accuracy in detection of variously textured objects.

Recent high-performance general-purpose processors can achieve real-time object detection, but at a heavy computational cost. The processor requires high power consumption and is therefore unsuitable for mobile systems under limited battery conditions. Consequently, a low-power and high-performance HOG feature extraction processor is necessary to widen the range of applications.

Figure 1 presents the image resolution versus frame rate for several published descriptions of HOG hardware. In the figure, a mark for each work denotes target application. A single circle signifies single-object detection. The double circle represents multiple object detection. Zhang et al. [2] proposed efficient object detection using GPGPU. Some FPGA implementations [3]–[8] and an FPGA-GPU architecture [9] have been proposed for real-time applications. Yazawa et al. proposed a target-reconfigurable object detector for multiple object detection. However, classification data must be reloaded in other object detection so that multiple objects cannot be detected at the same time. Our previous work [10] realized an FPGA implementation with superior performance compared with that of other implementations. However, that study particularly targeted pedestrian detection. HOG features are adaptable to widely various applications. Consequently, next-generation HOG feature extraction processors must provide higher expandability and higher performance. Therefore, our goal is to develop design techniques for a real-time HOG feature extraction processor for use in multiple object detection from HDTV-resolution video.

Most conventional processors use a window-based approach, for which an amount 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 amount of computations and memory bandwidth are greatly reduced by the reuse of calculated data or by adoption of efficient computation [10]. However, the power consumption is still high for mobile applications. Figure 2 shows the power consumption rate in our previous FPGA implementation. The most dominant part is cell histogram generation. The second is SVM classification. To achieve low-power object detection,

Manuscript received August 7, 2012.
Manuscript revised November 3, 2012.

[†] 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.E96.C.433

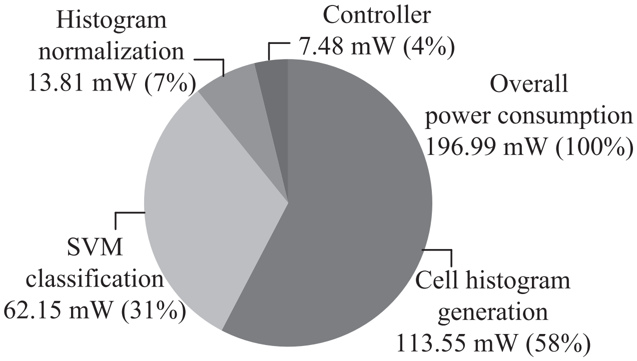


**Fig. 2** Power consumption in the previous FPGA implementation.

the dominant two parts are necessary to improve power efficiency.

To achieve real-time and low-power HOG feature extraction for HDTV resolution video, we propose the following three techniques.

- Simplified HOG algorithm with early rejection and early detection in SVM classification.
- Dual core architecture for parallel feature extraction and multiple object detection with HOG feature sharing.
- Detection-window-size scalable architecture with reconfigurable MAC array for processing objects of several shapes.

As described in this paper, details of the simplified HOG algorithm are described in Sect. 2. The proposed architecture is addressed in Sect. 3. Then, these are followed by VLSI implementation in Sect. 4. Section 5 concludes this paper.

## 2. Algorithm

### 2.1 Algorithm Overview

Figure 3 portrays a flow diagram of object detection using the original HOG algorithm [1]. Scanning on the input image is based on the detection window. The window is divided into cells, for each cell accumulating a histogram of gradient orientations over the pixels of the cell. For better invariance to illumination, histogram normalization can be accomplished by accumulating a measure of the local histogram energy over blocks and using the results to normalize all cells in the block. The normalized histograms (HOG features) are collected over the detection window. The collected features are fed to a linear SVM for object/non-object classification.

### 2.2 Simplified HOG Algorithm for Hardware Implementation

A simplified HOG algorithm for hardware implementation is introduced in this subsection. This algorithm is based on our previous work [10]. Figure 4 portrays a flow diagram of object detection using the simplified HOG algorithm. This flow is modified from the original algorithm using the following seven techniques.

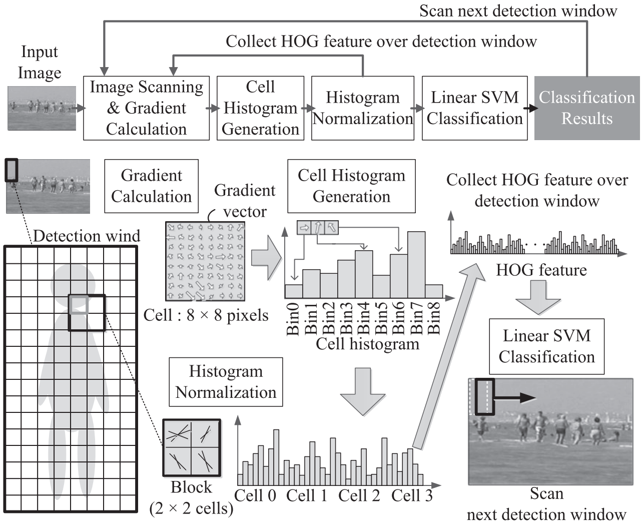

Scan next detection window
Collect HOG feature over detection window
Input Image
Image Scanning & Gradient Calculation
Cell Histogram Generation
Histogram Normalization
Linear SVM Classification
Classification Results
Gradient Calculation
Gradient vector
Cell Histogram Generation
Collect HOG feature over detection window
Detection wind
Cell : 8 × 8 pixels
Bin0 Bin1 Bin2 Bin3 Bin4 Bin5 Bin6 Bin7 Bin8
Cell histogram
HOG feature
Linear SVM Classification
Histogram Normalization
Block (2 × 2 cells)
Cell 0 Cell 1 Cell 2 Cell 3
Scan next detection window

**Fig. 3** Original HOG algorithm flow.

1. Cell-based scanning
   Figure 4 shows cell-based scanning approach. HOG features are extracted from cell-based calculations. No cell overlaps with other cells. Sharing and reuse of a cell strongly affect memory bandwidth reduction.
2. Gradient calculation using CORDIC [11]
3. Approximation of weighted voting for spatial and orientation anti-aliasing
4. Newton method with approximated initial value
5. Simultaneous SVM calculation
   Figure 4 presents simultaneous SVM calculations for cell-based processing. Partial HOG features belong to 105 windows maximally and are located at different positions in each window. Partial HOG features are multiplied and accumulated by the SVM coefficients of each window. The accumulation result is stored and reused in the subsequent SVM calculation. Simultaneous SVM calculation is suitable for parallel computing in hardware.
6. Early rejection and early detection
   This technique is newly added in this paper. A partial sum of products is compared with early classification threshold. A target object is detected or rejected in early stage if the comparison condition is true. This technique is detailed in Sect. 2.3.
7. Parameter optimization

### 2.3 SVM Classification with Early Rejection and Early Detection

In Linear SVM classification, extracted features and SVM coefficients are simply multiplied and accumulated until the operations reach window level. Consequently, this is suitable for parallel computing in hardware. Sufficient parallelism reduces the required cycle count by 99%. However,

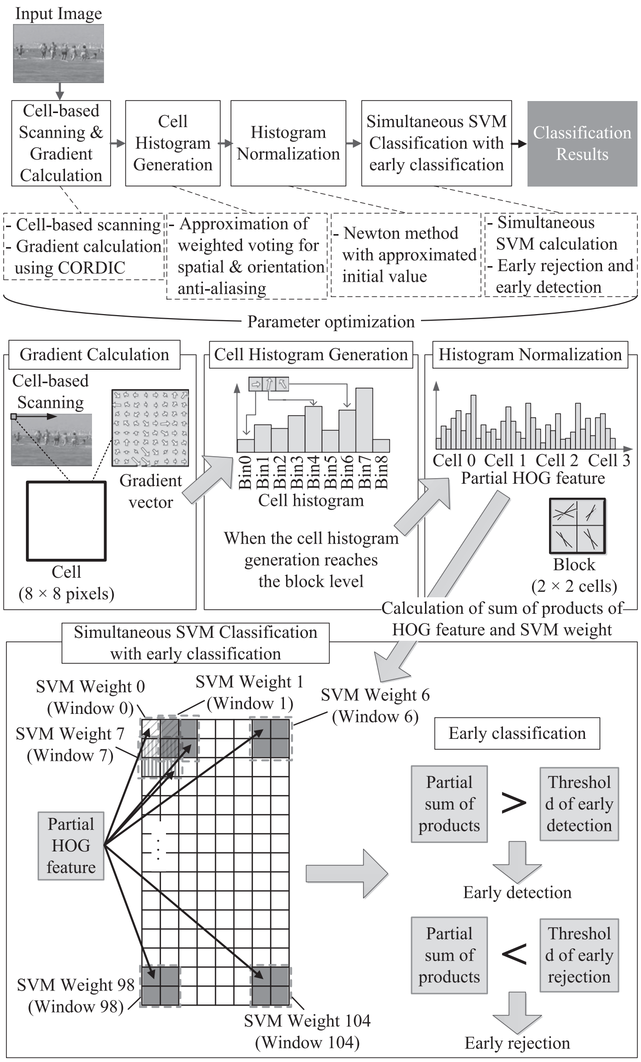

Input Image
Cell-based Scanning & Gradient Calculation
Cell Histogram Generation
Histogram Normalization
Simultaneous SVM Classification with early classification
Classification Results
- Cell-based scanning
- Gradient calculation using CORDIC
- Approximation of weighted voting for spatial & orientation anti-aliasing
- Newton method with approximated initial value
- Simultaneous SVM calculation
- Early rejection and early detection
Parameter optimization
Gradient Calculation
Cell-based Scanning
Gradient vector
Cell (8 × 8 pixels)
Cell Histogram Generation
Bin0 Bin1 Bin2 Bin3 Bin4 Bin5 Bin6 Bin7 Bin8
Cell histogram
When the cell histogram generation reaches the block level
Histogram Normalization
Cell 0 Cell 1 Cell 2 Cell 3
Partial HOG feature
Block (2 × 2 cells)
Calculation of sum of products of HOG feature and SVM weight
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
SVM Weight 98 (Window 98)
SVM Weight 104 (Window 104)
Early classification
Partial sum of products
>
Threshold of early detection
Early detection
Partial sum of products
<
Threshold of early rejection
Early rejection

**Fig. 4** Simplified HOG algorithm flow.

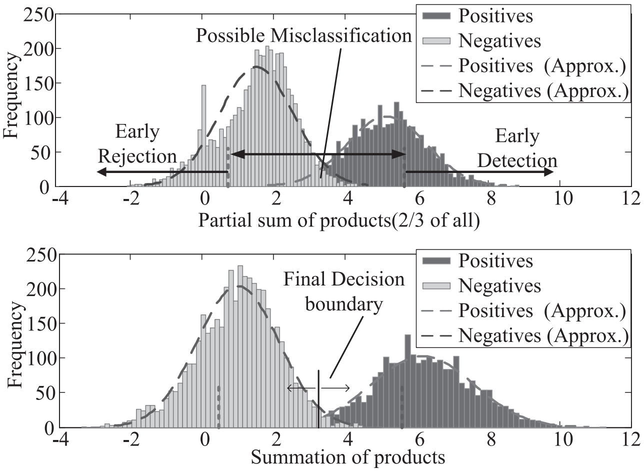


**Fig. 5** Early rejection and early detection.

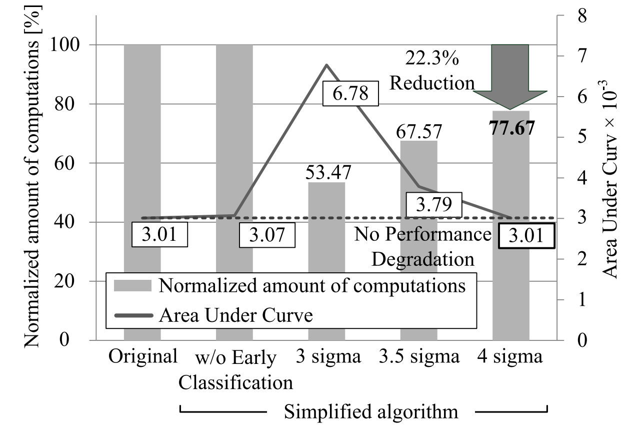


**Fig. 6** Trade-off between the amount of computations and area under curve.

the amount of computations is 6.34 GOPS and large when compared to the other calculations in an HOG processor.

As described above, a summation of products of the entire HOG feature and its paired SVM coefficients are necessary for classification. However, by classification with a partial sum of products obtained sequentially by cell-based processing, subsequent calculations can be skipped. Consequently, both the amount of computations and power consumption on classification are reduced. Figure 5 shows a distribution of summations and of partial sums (2/3 of entire feature) of positive (human) and negative (non-human) sample images in the training dataset.

Figure 5 shows that positive and negative classes are separated sufficiently on the distribution of partial sums, compared to the distribution of summations of the entire feature. This permits early classification using a partial sum instead of the entire feature and summations.

Early classification with a single threshold degrades the classification performance, because the classes on an early stage are less separated than those on a later stage. Within an overlapping area of the two classes, a misclassification occurs. To avoid performance degradation, we adopted a classification using two thresholds representing the interval causing misclassification. Estimating the interval approximating the distributions as a normal distribution, the incoming data are classified according to whether they are outside the interval. Data within the interval on the early stage are sent to the later stage and calculations are continued using subsequent partial features. Early classifications are evaluated 14 times per detection-window on every time after the partial HOG on a last row in the window is obtained.

The interval is learned preliminarily with training dataset. Then the early classification is applied to incoming image data. Classifying presumed data in the early stage permits skipping of subsequent calculations while avoiding classification performance degradation. However the amount-of-computations reductions and the performance degradation tradeoffs depend on the misclassification interval size.

Figure 6 shows a simulation result of the tradeoff between the amount of computations and area under curve

**Table 1** Algorithm parameters.

| | |
|---|---|
| Detection window size | 64 × 128 pixels |
| Cell size | 8 × 8 pixels |
| Block size | 2 × 2 cells |
| Step stride | 8 pixels, vertically and horizontally |
| # of orientation bins | 9 bins (0-180) |
| Weight for orientation anti-aliasing | 0.25 |
| Normalization method | L2-norm |
| SVM coefficients and threshold | Default value in OpenCV |
| # of HOG features | 3780 = (7 × 15 blocks) • (2 × 2 cells) • (9 bins) |

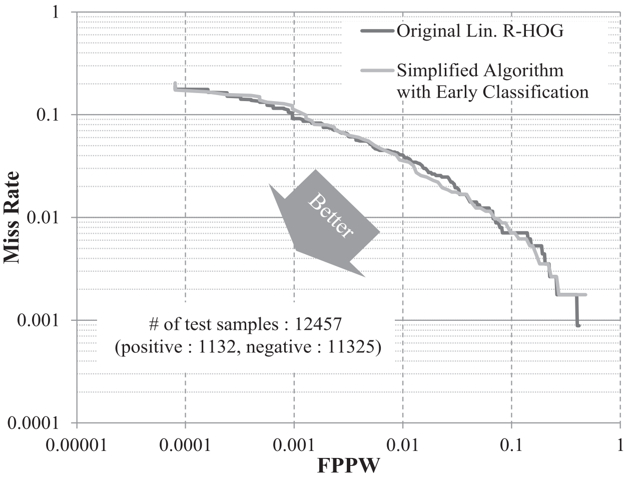


**Fig. 7** Accuracy degradation using the employed algorithm.

(AUC) of trade-off curve between miss rate and false positives per window. The lower AUC represents better classification performance. Presuming that a positive class and negative class are both distributed normally, standard deviations $\sigma$pos, $\sigma$neg and means $\mu$pos, $\mu$neg are estimated. The misclassification intervals are set from [$\mu$pos-n$\sigma$pos, $\mu$neg+n$\sigma$neg] as n varied from 3, 3.5, and 4. Simulation results show that the interval of [$\mu$pos-4$\sigma$pos, $\mu$neg+4$\sigma$neg] reduced amount of computations to 22.3% with no degradation from the original classification method. Consequently, the early classification method reduces the amount of computations in SVM classification from 6.34 GOPS to 4.92 GOPS.

### 2.4 Simulation Results

Performance and accuracy degradation were confirmed by using software implementation for the simplified algorithm with early classification. The simplified algorithm is compared with original linear rectangular HOG (Lin. R-HOG) [1]. The algorithm parameters for the simulation are shown in Table 1. The development environment is Microsoft Visual C++ 2008 Express Edition (Microsoft Corp.) with the INRIA Person Dataset [12], which includes several people in various backgrounds. Figure 7 presents a graph of false positives per window (FPPW) versus the miss rate. The original algorithm shows worse results from original paper [1] because our test condition is different from original one. The simulation results with the simplified algorithm and the optimized bit width show that no miss rate degradation occurs at 0.0001 FPPW. The algorithm that was used provides sufficient performance for general-purpose applications.

## 3. Architecture

### 3.1 Dual Core Architecture with Cell-Based Pipeline

Figure 8 depicts a block diagram of the dual core architecture and external peripherals. The proposed architecture comprises 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. The HOG feature extraction processor is controlled by an external CPU, and the input grayscale image is loaded to an internal SRAM from an external SRAM via a memory interface. The internal datapath modules process the input image and output a detection result. The CPU receives the detection result from HOG feature extraction processor; then it draws the result on a display.

Each core exchanges HOG feature and intermediate classification result with another core. This structure enables the HOG feature sharing described in Sect. 3.3 and detection-window-size scalable processing described in Sect. 3.4.

For cell histogram generation module and histogram normalization module, we used the same architecture as previously proposed in [10]. In cell histogram generation, four-way architecture is adopted because one cell is commonly used for four blocks maximally. Histogram normalization is two-stage architecture to implement L2-Hys normalization [13].

Our architecture adopts a cell-based pipeline flow, as presented in Fig. 9. The upper portion shows a relation between cells, blocks, windows, and a frame and the lower potion depicts a cell-based pipeline flow. The numbers in the left side are corresponding to the process explained below. C, B, H, and W are abbreviations of Cell, Block, HOG, and Window, respectively. Cell-based pipeline processing is conducted as follows:

1. A cell histogram is generated with cell-based scanning.
2. When the process described above reaches the block level, a block-level cell histogram is normalized; then the block-level HOG feature is extracted.
3. Block-level HOG features and SVM coefficients corresponding to each window are multiplied and accumulated.
4. A partial sum of products is compared with early classification threshold. The corresponding window is classified in early stage if the comparison condition is true.
5. An accumulation result of the window level is compared with the SVM threshold. Then the detection result is obtained.

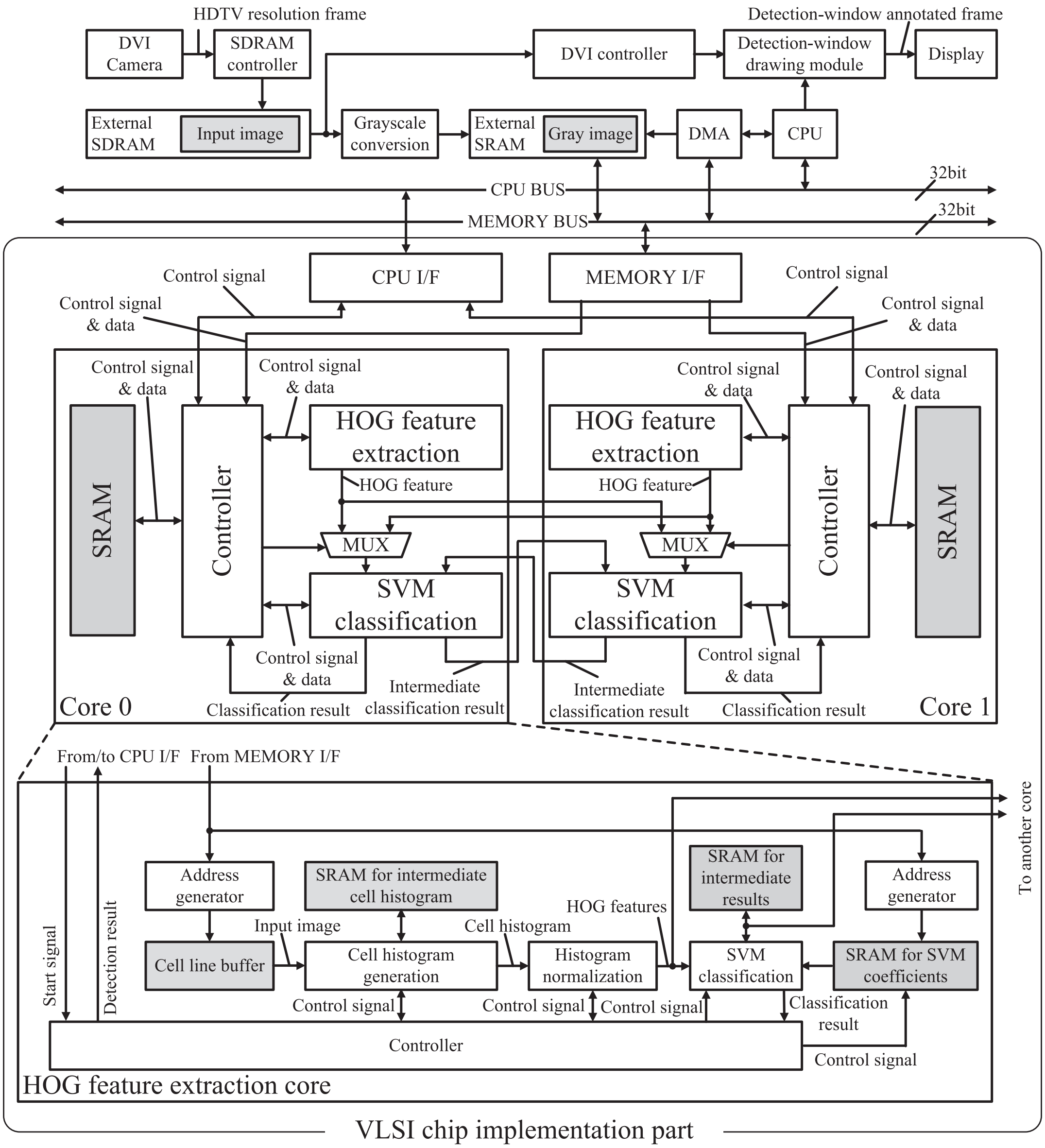


**Fig. 8** Dual core architecture for parallel feature extraction.

The cell-based pipeline architecture greatly reduces the memory bandwidth because it prevents the reloading of input pixels in different detection windows. The window-based approach for HDTV resolution requires memory bandwidth of 55 Gbps. Our architecture reduces the memory bandwidth to 0.499 Gbps with circuit area overhead for extra SRAMs to store intermediate cell histograms and classification results.

### 3.2 Parallel Processing in One Frame

This subsection describes parallel processing methods for single object detection to reduce the required cycle count. In our architecture, two methods, horizontal division and vertical division, are adopted as presented in Fig. 10.

The HDTV resolution frame is divided into two frames of 1920 × 600 pixels in the horizontal division. Each frame overlaps by 120 pixels for calculating HOG features in the border between the two frames. Horizontal division is a well-balanced method because each core manages the same number of pixels. The vertical division divides the frame into two frames of 992 × 1080 pixels and 984 × 1080 pixels. Each frame overlaps by 56 pixels. Each core can efficiently calculate HOG features because the pixel overlap between each frame is less than that of horizontal division. Therefore, the vertical division can provide more cycle reduction than the horizontal division. In the following section, two-core processing means the method using vertical division.

### 3.3 Parallel Processing for Multiple Object Detection

To detect multiple objects, conventional architecture requires independent processors corresponding to the number of target objects. However it dissipates power wastefully

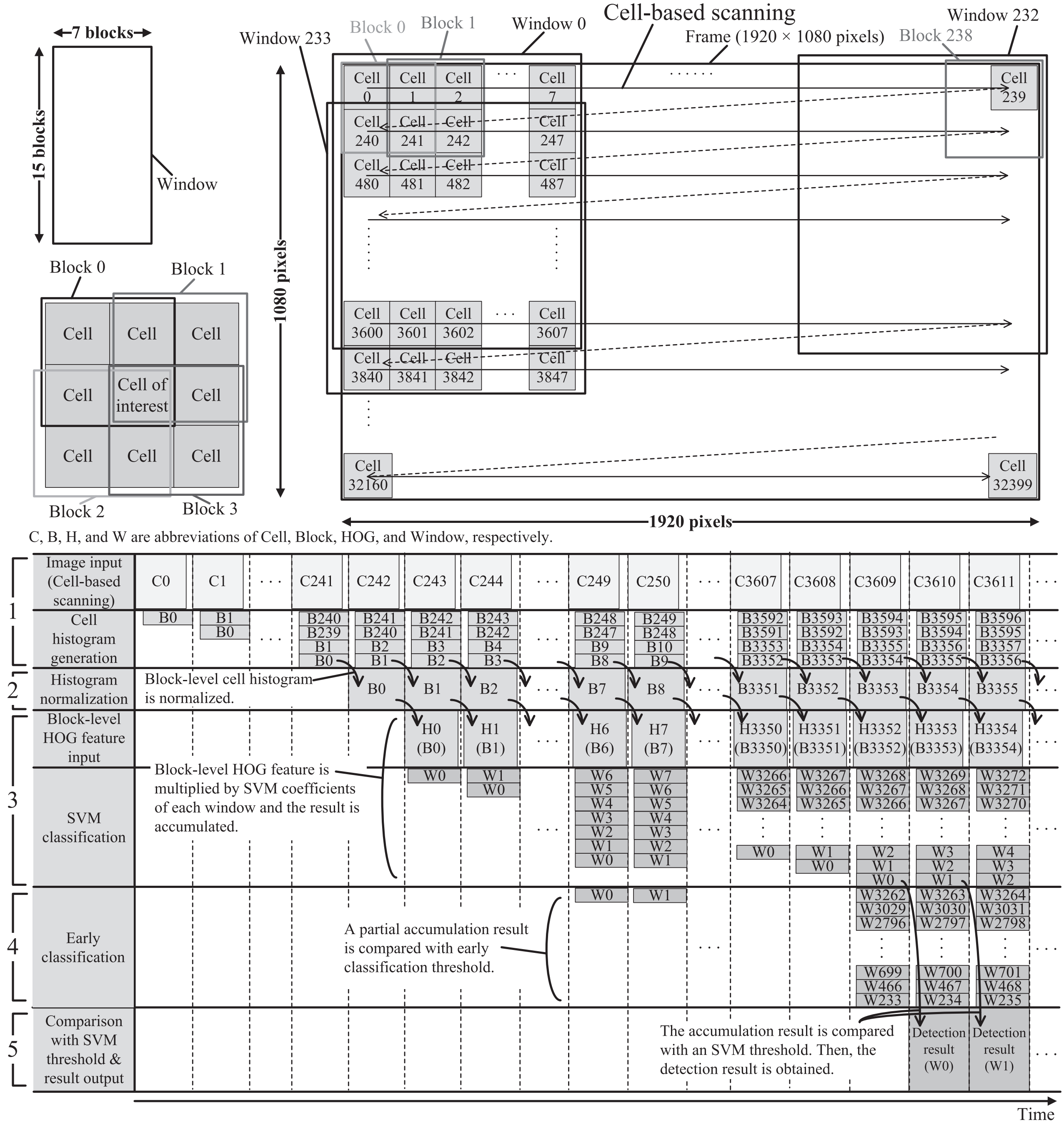


**Fig. 9** Cell-based pipeline flow.

for extraction of the same HOG feature. HOG feature extraction is the dominant part of the object detection task, as described in Sect. 1. Consequently, calculation of the same features must be avoided if low power operation is to be achieved.

Our architecture can provide low-power operation in multiple object detection. Figure 11 shows the block diagram in multiple object detection. One SVM classification module is used for processing a single object. Each core contains an independent SRAM for SVM coefficients corresponding to a single object. In case of that different SVM coefficients are loaded into the SRAM of each core, two different objects can be concurrently detected. Each core selects an internal HOG feature or external one. The HOG feature can be shared if the two cores process the same frame. Therefore, in one core, HOG feature extraction module, input image buffer, and working memory for HOG feature extraction are completely turned off to reduce power consumption.

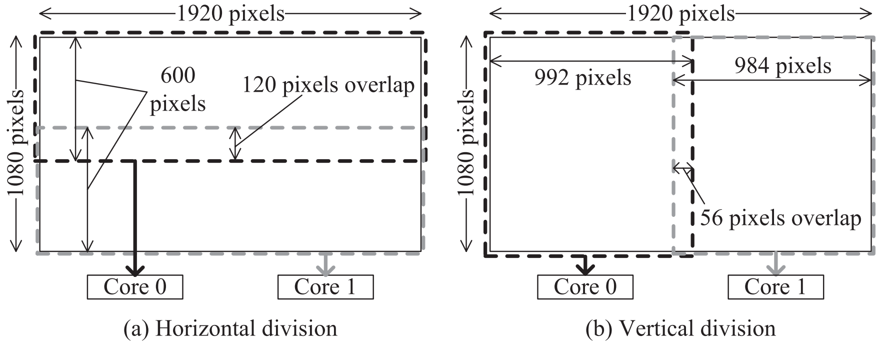


**Fig. 10** Frame division methods for parallel processing.

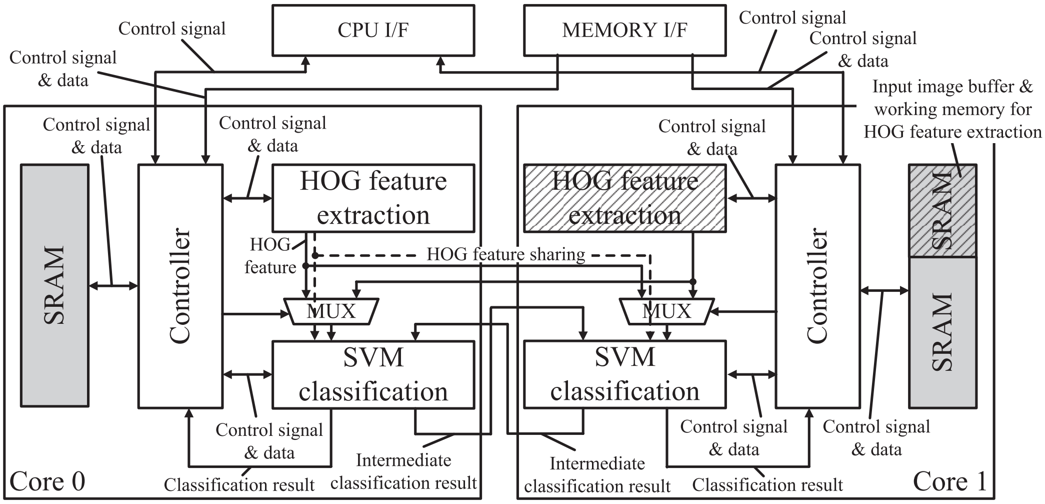


**Fig. 11** Block diagram in multiple object detection.

### 3.4 Detection-Window-Size Scalable Architecture

Object detection for a single target is insufficient for recent advanced applications. For example, on-vehicle applications target pedestrian, cars, and traffic sign. Objects have various shapes such as vertically long, horizontally long, and square. Consequently, it is demanded that an object detection processor manages several target objects.

Our architecture provides object detection for the three shapes shown above. Figure 12 shows a block diagram of a reconfigurable SVM classification module for a vertically long object, a horizontally long one, and a square one. This module comprises a reconfigurable MAC array, which is reconfigured according to the shape of target objects. A vertically long object is processed if the MAC array is connected horizontally as presented in Fig. 12(b). It corresponds to the detection window of 64 × 128 pixels. A horizontally long object is detected if the MAC array is connected vertically as displayed in Fig. 12(c). The detection of a square object is conducted with cooperation of two processor cores as presented in Fig. 12(d). The classification core 0 in core 1 loads an intermediate result in classification core 7 in core 0 as the initial value. It enables classification with a detection window of 128 × 128 pixels.

### 3.5 SVM Classification Module with Early Rejection and Early Detection

Figure 13 shows an SVM classification module with an early classification function described in Sect. 2.3. Early classification is conducted in the end of each classification core. A classification flag is enabled if an intermediate result is greater than an early detection threshold or less than an early rejection threshold. To reduce wasteful power for classification of non-target object, the classification flag controls whether each classification core runs. If a classification flag of a detection window is activated, the controller enables a clock gating of the MAC module corresponding to the detection window. The classification flag is stored to SRAM with an intermediate classification result.

### 3.6 Performance Evaluation

The number of cycle counts was estimated using a Verilog-HDL simulator. The proposed architecture was compared with our previous architecture [10] and architecture without parallelization. Estimation results are presented in Fig. 14,

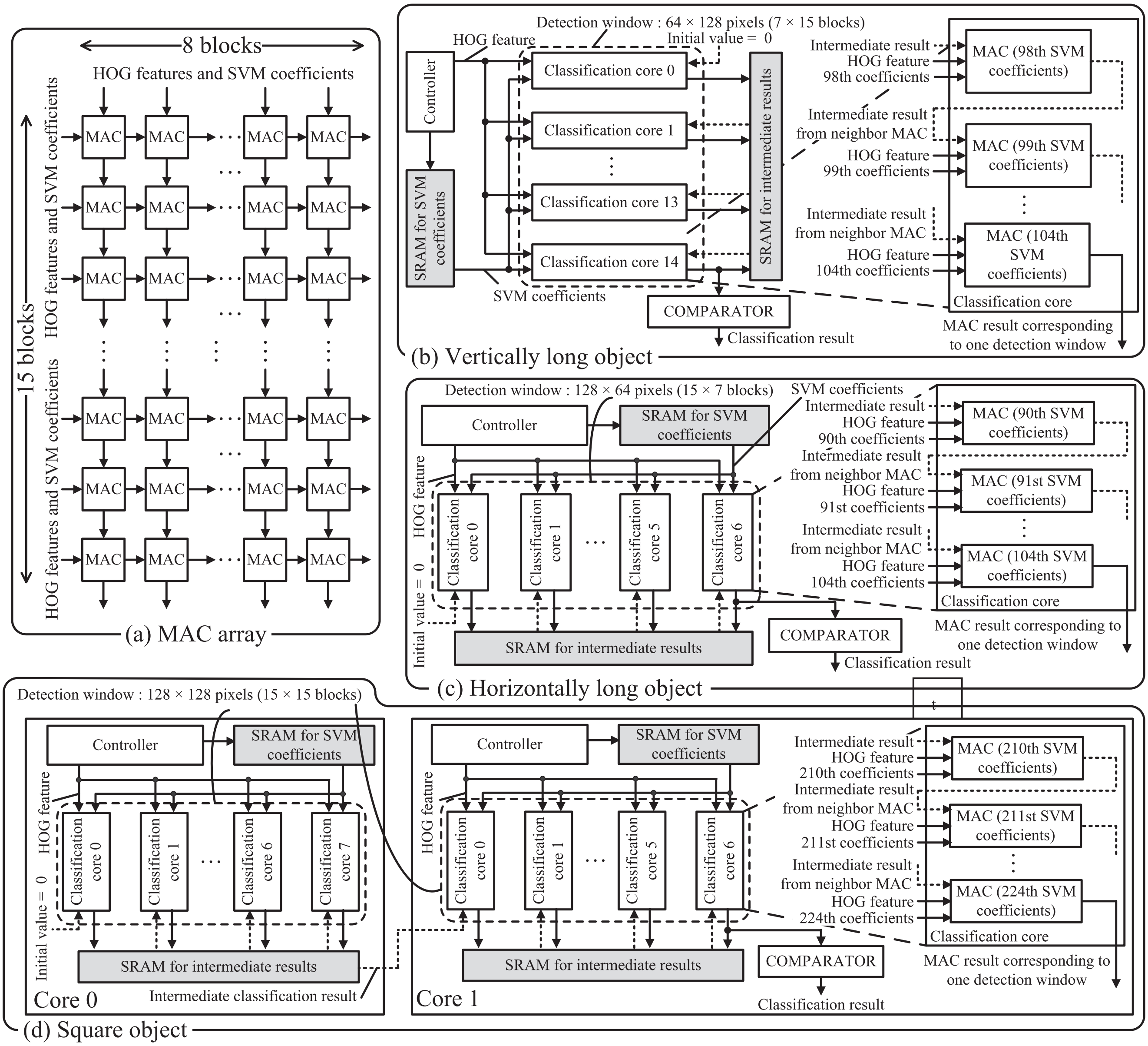

8 blocks
HOG features and SVM coefficients
15 blocks
HOG features and SVM coefficients
HOG features and SVM coefficients
MAC
(a) MAC array
Detection window : 64 × 128 pixels (7 × 15 blocks)
HOG feature
Initial value = 0
Controller
SRAM for SVM coefficients
Classification core 0
Classification core 1
Classification core 13
Classification core 14
SRAM for intermediate results
SVM coefficients
COMPARATOR
Classification result
Intermediate result
HOG feature
98th coefficients
MAC (98th SVM coefficients)
Intermediate result from neighbor MAC
HOG feature
99th coefficients
MAC (99th SVM coefficients)
Intermediate result from neighbor MAC
HOG feature
104th coefficients
MAC (104th SVM coefficients)
Classification core
MAC result corresponding to one detection window
(b) Vertically long object
Detection window : 128 × 64 pixels (15 × 7 blocks)
SVM coefficients
Controller
SRAM for SVM coefficients
HOG feature
Initial value = 0
Classification core 0
Classification core 1
Classification core 5
Classification core 6
SRAM for intermediate results
COMPARATOR
Classification result
Intermediate result
HOG feature
90th coefficients
MAC (90th SVM coefficients)
Intermediate result from neighbor MAC
HOG feature
91st coefficients
MAC (91st SVM coefficients)
Intermediate result from neighbor MAC
HOG feature
104th coefficients
MAC (104th SVM coefficients)
Classification core
MAC result corresponding to one detection window
(c) Horizontally long object
Detection window : 128 × 128 pixels (15 × 15 blocks)
Controller
SRAM for SVM coefficients
HOG feature
Initial value = 0
Classification core 0
Classification core 1
Classification core 6
Classification core 7
SRAM for intermediate results
Core 0
Intermediate classification result
Controller
SRAM for SVM coefficients
HOG feature
Classification core 0
Classification core 1
Classification core 5
Classification core 6
SRAM for intermediate results
Core 1
COMPARATOR
Classification result
Intermediate result
HOG feature
210th coefficients
MAC (210th SVM coefficients)
Intermediate result from neighbor MAC
HOG feature
211st coefficients
MAC (211st SVM coefficients)
Intermediate result from neighbor MAC
HOG feature
224th coefficients
MAC (224th SVM coefficients)
Classification core
MAC result corresponding to one detection window
(d) Square object

**Fig. 12** Block diagrams of reconfigurable MAC array (a) and SVM classification module for (b) a vertically long object, (c) a horizontally long object, and (d) a square object.

which demonstrates the superiority of the proposed architecture for HDTV resolution. Results show that the number of cycle counts with two cores in cell histogram generation, histogram normalization, and SVM classification are lower by 48.5%, 42.5%, and 50%, respectively, than the number of cycle counts of our previsou architecture. In the proposed two-core architecture, the overall process requires $1.43\times10^6$ cycles per frame. Therefore, it is inferred that the proposed architecture can accommodate HDTV resolution video at 30 fps with 42.9 MHz.

## 4. VLSI Implementation

A test chip of HOG feature extraction has been designed as presented in Fig. 15. The design includes the VLSI-oriented algorithm and the dual core architecture. This chip, which was fabricated in 65nm CMOS technology, occupies $4.2\times2.1$ mm$^2$ containing 502 Kgates and 1.22 Mbit on-chip SRAMs. The static timing analysis after back annotation shows that 110 MHz operation is attained at nominal supply voltage of 1.1 V. The chip specifications are presented in Table 2.

Gate-level power estimation was conducted to clarify effects of our implementation. Here a commercial power estimation tool was used for the whole processor core circuit with physical parameters after place and routing. Figure 16 shows estimated data of power consumption in several operation modes. The data of our earlier FPGA implementation [10] are displayed for comparison. The FPGA implementation can process SVGA resolution video (800 × 600

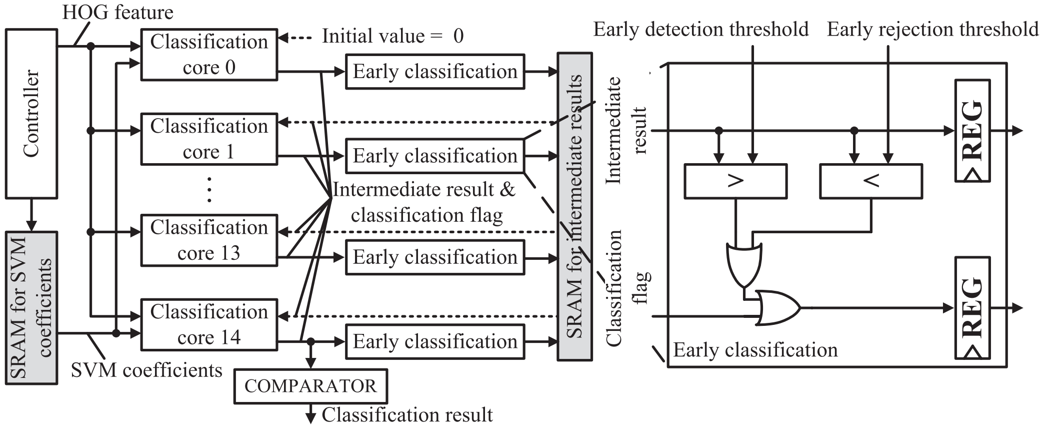


**Fig. 13** Block diagram of SVM classification module with early rejection and early detection.

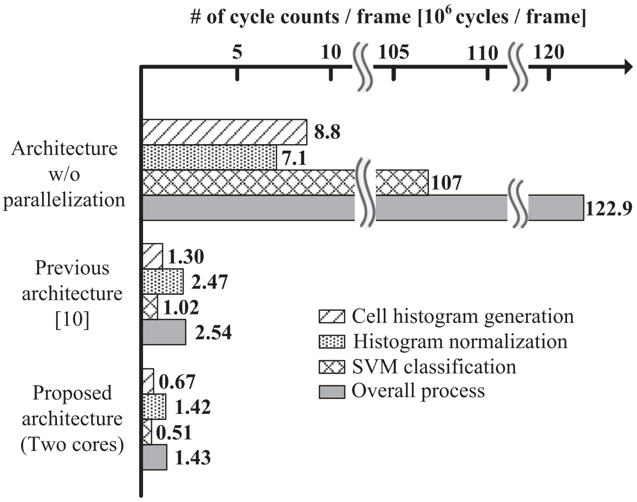


**Fig. 14** Reduction of cycle count.

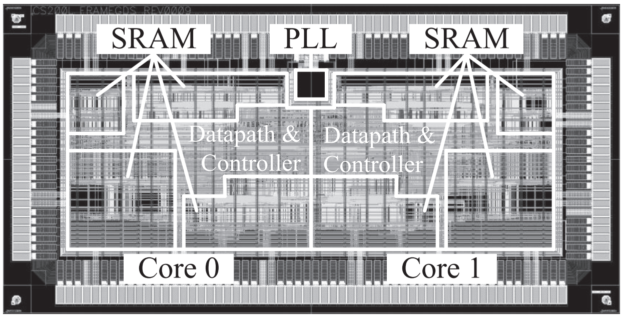


**Fig. 15** Chip layout.

**Table 2** Chip specifications.

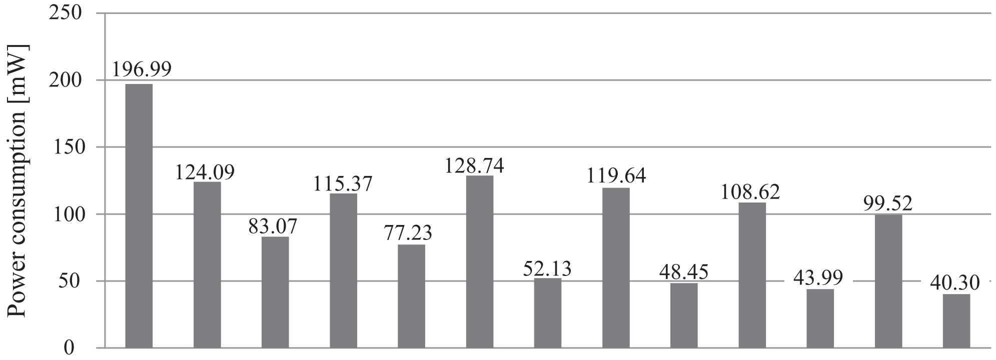
| | |
|---|---|
| Technology | 65nm CMOS |
| Chip size | 4.2 × 2.1mm$^2$ |
| Core size | 3.3 × 1.2mm$^2$ |
| Power supply | 1.1 V |
| Max frequency | 110 MHz |
| Gate count | 502 Kgates |
| Memory size | 1.22 Mbit (610 Kbit for one core) |
| Image resolution | HDTV (1920 × 1080 pixels) @ 30fps |
| Power | 99.52 mW @ 42.9 MHz 1.1 V<br>40.30 mW @ 42.9 MHz 0.7 V (min) |

pixels) at 72 fps. On the other hand, the VLSI implementation targets HDTV resolution video (1920 × 1080 pixels) at 30 fps. The power consumption of the proposed architecture is estimated in several conditions, combinations of the number of cores, early classification described in Sect. 3.5, and feature sharing described in Sect. 3.3. To prevent miss-rate degradation, $4\sigma$ is adopted for the standard deviation in early classification. In general, supply voltage at 42.9 MHz can be lowered than supply voltage at 83.4 MHz. According to past implementation result in same process technology [14], minimum supply voltages at 42.9 MHz and 83.4 MHz are 0.7 V and 0.9 V, respectively. Consequently, two-core configuration shows less power consumption than one-core configuration at minimum supply voltage. Two-core configuration with early classification and feature sharing consumes 99.52 mW at 42.9 MHz when a nominal supply voltage is 1.1 V, which attains real-time processing for HDTV resolution video with 30 fps frame rate.

Figure 17 portrays an example of real-time object detection by the proposed accelerator.

## 5. Conclusion

This paper presents a proposal of a novel architecture of real-time HOG feature extraction for parallel feature processing. The proposed method includes a simplified HOG algorithm with early classification, dual core architecture with a cell-based pipeline, and detection-window-size scalable architecture. The early classification method reduces the amount of computations in SVM classification from 6.34 GOPS to 4.92 GOPS with no accuracy degradation. The dual core architecture provides several modes of parallel processing for required cycle count reduction and power consumption reduction. In addition, the proposed architecture has high functionality for simultaneous multiple-object detection and detection-window-size scalability. Consequently, the proposed architecture is adaptable to recent ad-



| Device | FPGA | LSI | LSI | LSI | LSI | LSI | LSI | LSI | LSI | LSI | LSI | LSI | LSI |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| # of cores | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| Early classification | N/A | OFF | OFF | ON | ON | OFF | OFF | ON | ON | OFF | OFF | ON | ON |
| Feature sharing | N/A | N/A | N/A | N/A | N/A | OFF | OFF | OFF | OFF | ON | ON | ON | ON |
| Image resolution | SVGA @72fps | HDTV @30fps | HDTV @30fps | HDTV @30fps | HDTV @30fps | HDTV @30fps | HDTV @30fps | HDTV @30fps | HDTV @30fps | HDTV @30fps | HDTV @30fps | HDTV @30fps | HDTV @30fps |
| Operating frequency | 40 MHz | 83.4 MHz | 83.4 MHz | 83.4 MHz | 83.4 MHz | 42.9 MHz | 42.9 MHz | 42.9 MHz | 42.9 MHz | 42.9 MHz | 42.9 MHz | 42.9 MHz | 42.9 MHz |
| Supply voltage | 1.2 V | 1.1 V | 0.9 V (min) | 1.1 V | 0.9 V (min) | 1.1 V | 0.7 V (min) | 1.1 V | 0.7 V (min) | 1.1 V | 0.7 V (min) | 1.1 V | 0.7 V (min) |

**Fig. 16** Power consumption in several modes.

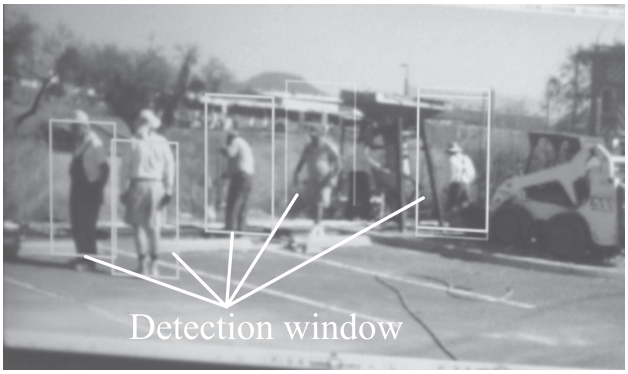


**Fig. 17** Object detection by the proposed accelerator.

vanced applications such as on-vehicle application and intelligent robots. The VLSI implementation shows 49.5% power reduction with that of the conventional processor [10]. The design techniques described herein are anticipated for application to several image recognition applications. They are anticipated specifically for their great impact on mobile applications under limited battery conditions.

## 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] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” Proc. 2005 International Conference on Computer Vision and Pattern Recognition, vol.2, pp.886–893, IEEE Computer Society, Washington, DC, USA, 2005.

[2] L. Zhang and R. Nevatia, “Efficient scan-window based object detection using GPGPU,” IEEE, CVPRW, 2008.

[3] S. Bauer, U. Brusmann, and S. Schlotterbeck-Macht, “FPGA implementation of a HOG-based pedestrian recognition system,” MPC-Workshop, July 2009.

[4] M. Hiromoto and R. Miyamoto, “Hardware architecture for high-accuracy real-time pedestrian detection with CoHOG features,” IEEE ICCVW 2009.

[5] R. Kadota, H. Sugano, M. Hiromoto, H. Ochi, R. Miyamoto, and Y. Nakamura, “Hardware architecture for HOG feature extraction,” Proc. 2009 International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp.1330–1333, IEEE Computer Society, Washington, DC, USA, 2009.

[6] Y. Yazawa, T. Yoshimi, T. Tsuzuki, T. Dohi, and H. Fujiyoshi, “FPGA hardware with target-reconfigurable object detector by joint-HOG,” Proc. SSII, Yokohama, Japan, 2011.

[7] K. Negi, K. Dohi, Y. Shibata, and K. Oguri, “Deep pipelined one-chip FPGA implementation of a real-time image-based human detection algorithm,” IEEE FPT 2011.

[8] T.P. Cao and G. Deng, “Real-time vision-based stop sign detection system on FPGA,” Proc. Digital Image Computing: Techniques and Applications, pp.465–471, IEEE Computer Society, Los Alamitos, CA, USA, 2008.

[9] S. Bauer, S. Kohler, K. Doll, and U. Brunsmann, “FPGA-GPU architecture for kernel SVM pedestrian detection,” IEEE CVPRW 2010.

[10] K. Mizuno, Y. Terachi, K. Takagi, S. Izumi, H. Kawaguchi, and M. Yoshimoto, “Architectural study of HOG feature extraction proces-

sor for real-time object detection," IEEE SiPS, Oct. 2012.

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

[12] INRIA Person Dataset. http://pascal.inrialpes.fr/data/human/

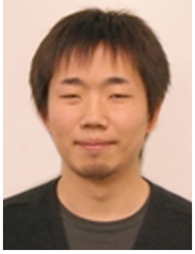
[13] D.G. Lowe, "Distinctive image features from scale invariant keypoints," Int. J. Comput. Vis., vol.60, no.2, pp.91–110, 2004.

[14] H. Noguchi, J. Tani, Y. Shimai, M. Nishino, S. Izumi, H. Kawaguchi, and M. Yoshimoto, "A 34.7-mW quad-core MIQP solver processor for robot control," Proc. IEEE Custom Integrated Circuits Conference (CICC), Sept. 2010.

**Kosuke Mizuno** received the B.S. and M.S. 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.

**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. Since 2011, he has been involved in the research and development of low-power image recognition VLSI designs.

**Yosuke Terachi** received the B.S. and M.S. degrees in Computer Science and Systems Engineering from Kobe University, Kobe, Japan in 2010 and 2012. His research interestincludes development of low-power image recognition VLSI designs.



**Shintaro Izumi** received his B.Eng. and M.Eng. degrees in Computer Science and Systems Engineering from Kobe University, Hyogo, Japan, in 2007 and 2008, respectively. 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 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, 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 2011–2012. 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.