Real-Time Object Recognition with Neuro-Fuzzy Controlled Workload-Aware Task Pipelining

A proposed object recognition processor lightens its workload by estimating global region-of-interest features. A neuro-fuzzy controller performs intelligent ROI estimation by mimicking the human visual system, then manages the processor's overall pipeline stages using workload-aware task scheduling and applied database size control. The NFC performs workload-aware dynamic power management to reduce the proposed processor's power consumption.

Joo-Young Kim
Minsu Kim
Seungjin Lee
Jinwook Oh
Sejong Oh
Hoi-Jun Yoo
Korea Advanced Institute of Science and Technology

Object recognition, a key technology for real-time intelligent vision applications such as autonomous cruise control, mobile robot vision, and surveillance systems,[1-6] is the process of identifying objects out of an input image by matching their feature vectors with a predefined object database (see Figure 1). It broadly consists of a preprocessing stage, which extracts keypoints from the input image through a cascaded filtering process and generates their descriptor vectors, and a postprocessing stage, which matches the generated descriptor vectors with the object database for final recognition. From an operational view, object recognition involves not only low-level per-pixel processing but also high-level keypoint-related processing. The image filtering and keypoint extraction belong to low-level per-pixel processing, which applies the identical arithmetic operations to each pixel of 2D image data. For this kind of processing, we prefer to use a single instruction, multiple data (SIMD) processor architecture because it can easily exploit data-level parallelism for independent image data. On the other hand, descriptor generation and database matching are keypoint-related operations, so they can differ with respect to individual keypoints. For these types of processing, we prefer a multiple instruction, multiple data (MIMD) processor architecture with an independent instruction unit for each processing core. The execution time breakdown[7] of the most popular object recognition algorithm, scale invariant feature transform (SIFT),[8] shows that both the pixel-level image processing and keypoint-related processing are equally significant in object recognition.

Because each object recognition stage requires many computations, achieving real-time operation for object recognition using a single-threaded CPU is difficult. Zhang et al. achieved real-time operation (more than

Published by the IEEE Computer Society

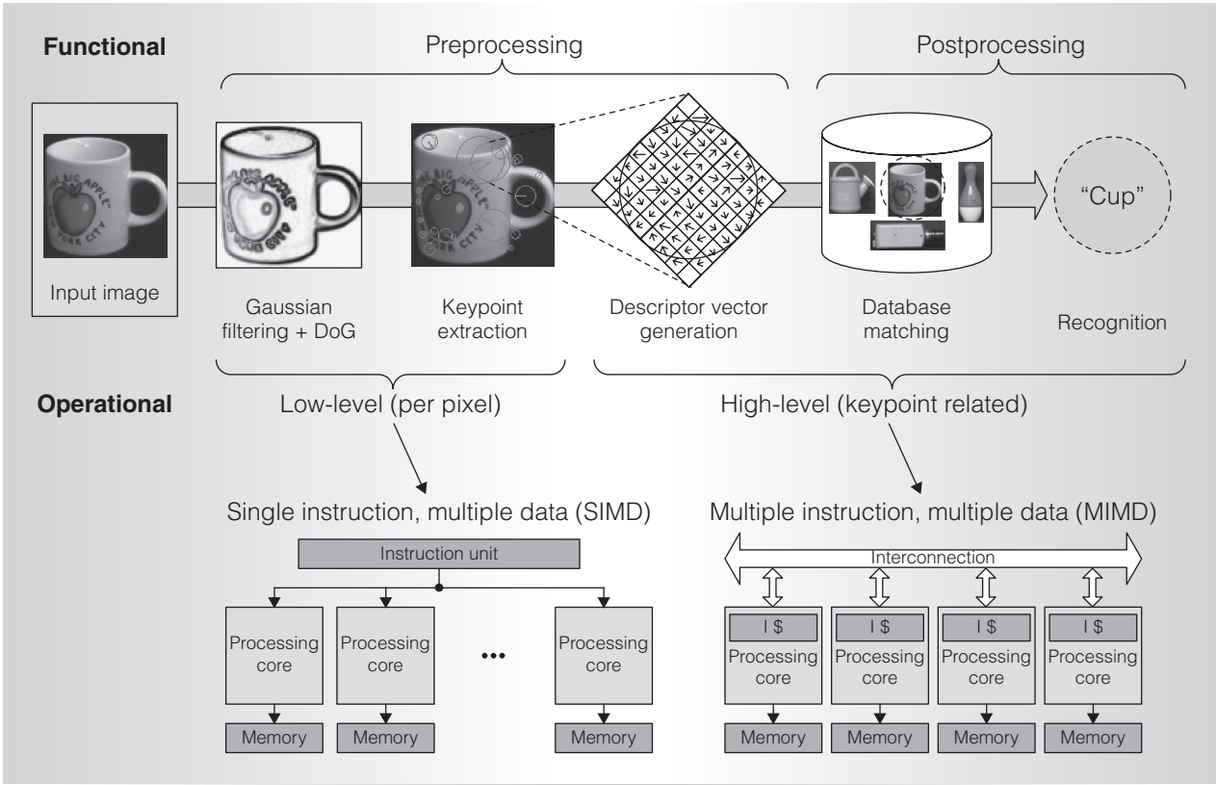


Figure 1. General object recognition process: a preprocessing stage extracts keypoints from an input image, and a postprocessing stage matches the keypoints with a trained object database. From an operational view, it requires both low-level per-pixel processing and high-level keypoint-related processing.

30 frames per second [fps]) for the SIFT algorithm at the VGA (640 × 480) video input by parallelizing it on a multicore system with two 2.33-GHz Intel Core2 Quad processors (8 cores).[7] However, a multicore system using general-purpose CPUs is extremely cost-inefficient in terms of silicon area, power, and energy consumption. For example, Zhang's multicore system consumed more than 100 W of power, which would be difficult for mobile applications. Recently, as the multicore computing paradigm has advanced, researchers have developed several embedded multicore vision processors for real-time object recognition operation with low power consumption.[1-6]

Processor architectures for object recognition

Figure 2 shows the evolution of processor architectures for object recognition.

The first architecture (Figure 2a) is a massively parallel SIMD architecture, which NEC's IMAP-CE[1] and NXP's XETAL II[2] exploit. In this architecture, hundreds of processing elements connect linearly and run the same operations via a global processor. This architecture's strength is low-level image processing, such as image filtering and enhancement, which requires the same operations for each pixel of image data to have a certain effect on the entire image. However, identical operations by hundreds of processing elements aren't suitable for object recognition's keypoint-related tasks, such as descriptor vector generation and database matching. This architecture has difficulty performing irregular and different operations with respect to each keypoint.

The second architecture (Figure 2b) is a MIMD architecture, which is generally used by multicore processors and also exploited

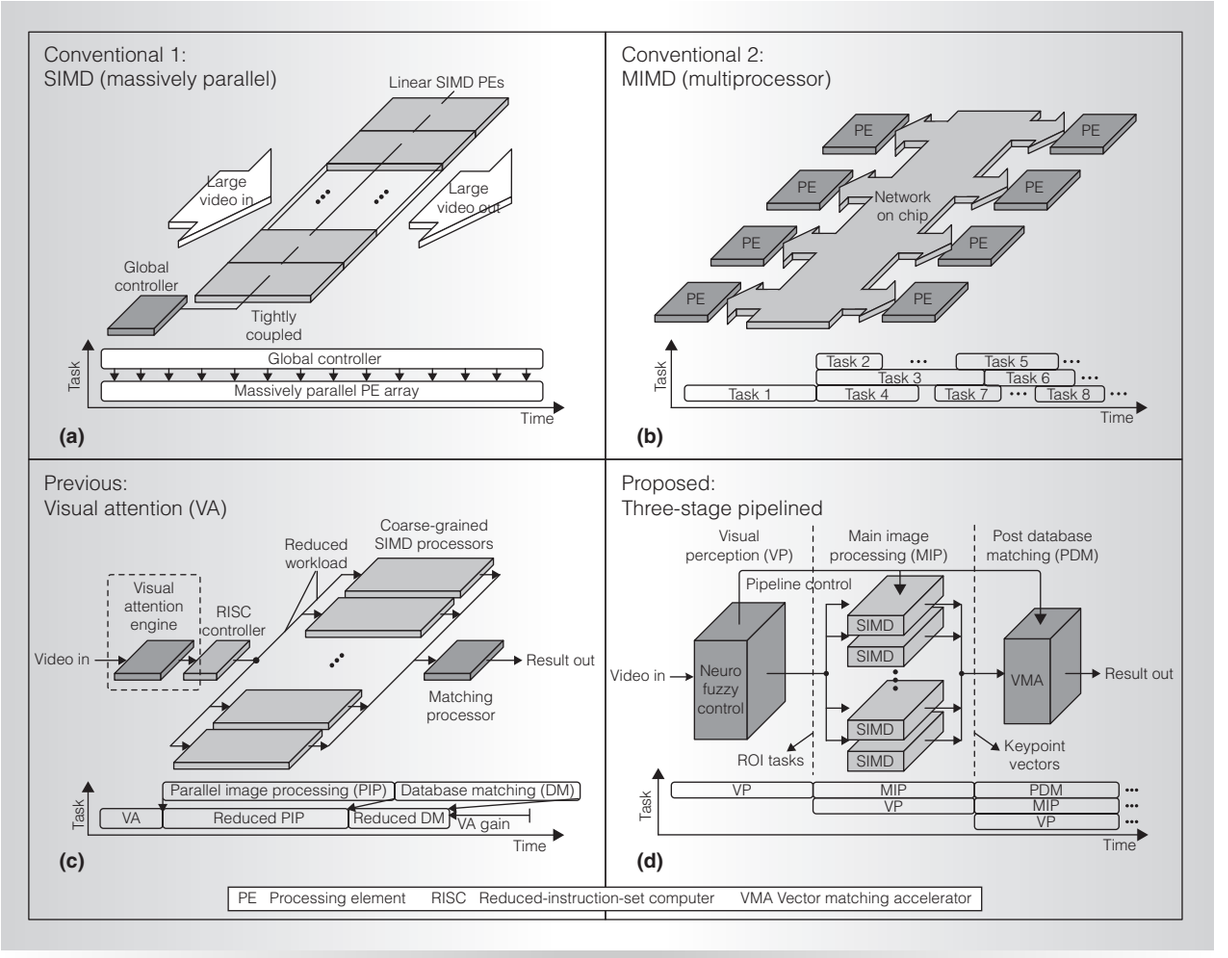


Figure 2. The evolution of processor architectures for object recognition: single instruction, multiple data, or SIMD (a); multiple instruction, multiple data, or MIMD (b); visual attention (c); and our proposed three-stage pipelined architecture (d).

by a network-on-chip (NoC)-based object recognition processor.[3] This processor employs coarse-grained processing elements and a memory-centric NoC to exploit task-level parallelism over data-level parallelism. However, it can't provide enough computing power for real-time object recognition owing to the large data-synchronization costs among independent processing elements. On the other hand, NEC's XC core[4] adopts a reconfigurable architecture that can change the SIMD and MIMD mode dynamically to obtain both processing efficiency and flexibility. Compared with its previous SIMD-only processor,[1] it significantly improves performance for several image-processing kernels. However, it's still inefficient in terms of object recognition processing because it handles the entire input image and can include unnecessary information.

The third architecture (Figure 2c) is a visual attention architecture that we proposed in other work.[5] Unlike conventional architectures, this architecture adopts a special hardware block called the *visual attention engine* (VAE)[9] with parallel processing elements. The VAE rapidly extracts an input image's saliency feature and filters the keypoints out of meaningless regions of the image. This process reduces the computational costs of keypoint-related tasks and shortens the overall object recognition time. However, this architecture's performance is limited by the VAE and parallel processing elements' sequential behavior. Because the workloads between the VAE and parallel processing

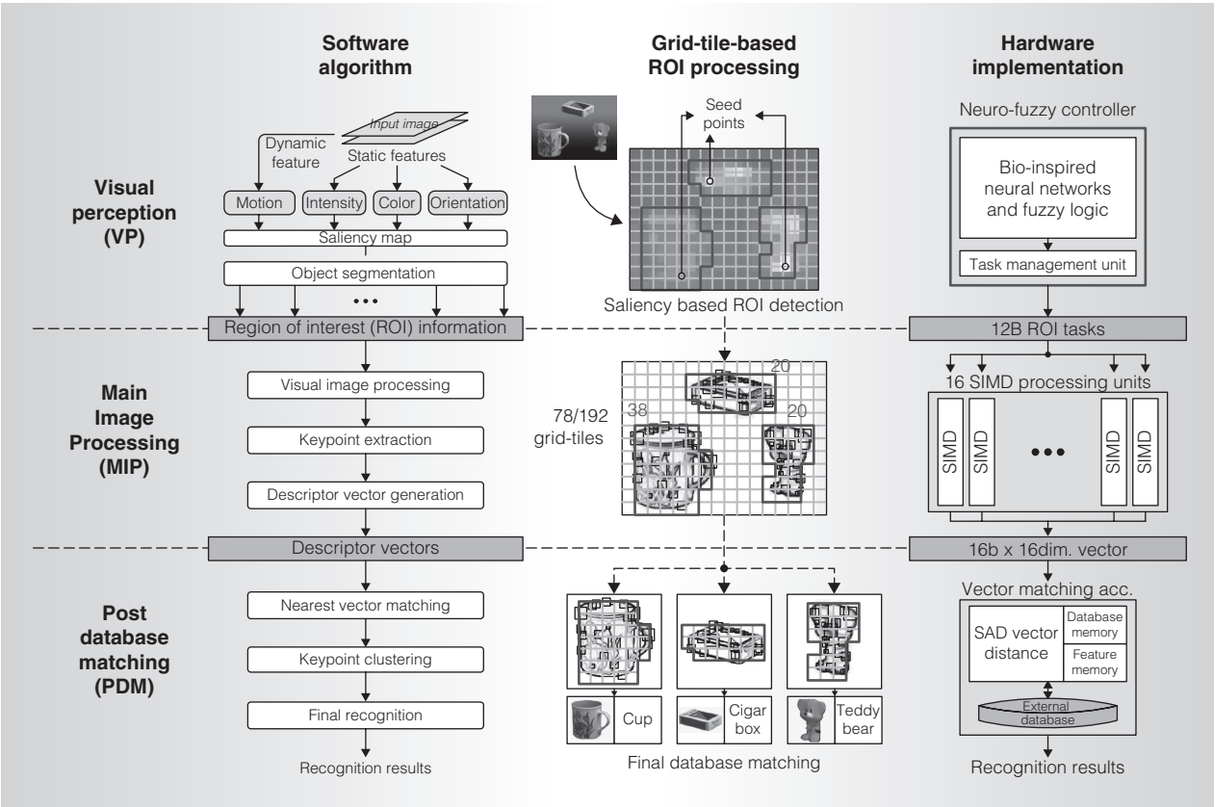


Figure 3. Our proposed three-stage object recognition processor. We insert a visual perception (VP) stage computing ROIs out of an input image to lighten and manage the workloads of the main image processing (MIP) and post database matching (PDM) stages. We employ a neuro-fuzzy controller, multiple SIMD processing units, and vector matching accelerator for hardware implementation of the three stages.

elements are extremely unbalanced, their pipelined operations are difficult to perform. Additionally, this architecture doesn't consider a pipeline balancing scheme.

We propose an energy-efficient three-stage pipelined architecture (Figure 2d) for real-time multiobject recognition.[6] In this architecture, we divide object recognition into three stages, which are executed in the pipeline for high frame-rate operation. For efficient task pipelining, we propose a neuro-fuzzy controller (NFC) for the first stage to estimate an input image's region-of-interests (ROIs) before moving to the main object recognition stages. Based on intelligent ROI estimation performed by bio-inspired neural networks and fuzzy logic, the NFC not only manages the processor's overall pipeline but also performs workload-aware dynamic power management. Thanks to neuro-fuzzy controlled task pipelining and dynamic power management, the proposed architecture achieves 60 fps object recognition while dissipating less than 500 mW at 1.2 V in a 0.13 μm technology. The obtained 8.2 mJ energy consumption per frame is 3.2 times lower than the state-of-the-art vision processors.

Three-stage pipelined object recognition processor

Figure 3 shows a software algorithm, grid tile-based processing model, along with the proposed object recognition processor's hardware implementation. The overall object recognition has three stages: visual perception (VP), main image processing (MIP), and post database matching (PDM).

In the VP stage, we estimate multiple objects' ROIs. By reducing an image's processing area to only ROIs, the VP stage lightens the workload for the following stages.

To correctly select ROIs of multiple objects even in cluttered input images, we adopted Itti's visual attention model,[10] which generates a saliency map from the input image using several bottom-up features based on the human visual system. Walther et al. demonstrate that visual attention can successfully find salient regions likely to contain objects from complex scenes.[11] And Ouerhani proves that combining several bottom-up features can effectively generate a more effective saliency map for object region detection.[12]

We based our VP software algorithm on these previous models. The VP algorithm extracts not only static features (such as intensity, color, and orientation), but also a dynamic feature (such as a motion vector) out of the 1/8 scaled-down 80 × 60 pixel input image to generate the saliency map. On the basis of the saliency map, the VP algorithm selects the objects' seed points and performs seeded region growing to detect the objects' ROIs.[13] The VP then quantizes multiple objects' ROIs in a 40 × 40 size grid tile, and the VP stage outputs the selected ROI grid tiles to the MIP stage. To implement visual attention and multiobject ROI detection, we propose using an NFC consisting of biologically inspired neural networks and fuzzy logic circuits. In the overall architecture, it also plays a role in controlling the whole processor pipeline.

The MIP stage extracts local features for object recognition from the selected ROI grid tiles from the VP stage. In local feature extraction, there are various methods, such as KLT, Harris corner detector, affine invariant features, and SIFT.[8,14] We selected the SIFT algorithm because it's invariant to image scale and rotation and provides more distinctive, robust features than the other methods. To compute SIFT object features, we extracted the keypoints from an input image using a cascaded image filtering and local extrema search, and we converted each keypoint to a descriptor that describes its magnitude and orientation characteristics in a vector format. The MIP stage outputs the descriptor vectors to the PDM stage. To implement the MIP stage, we employ 16 SIMD processing units (SIMD-PU) to accelerate parallel and complex image processing tasks. Each SIMD-PU is fully programmable and performs SIFT for the assigned grid tiles. We can change the applied algorithm for object feature extraction by programming each SIMD-PU.

The PDM stage matches the descriptor vectors from the MIP stage with a trained object database that includes tens of thousands of object vectors. For vector matching, we employ the nearest matching neighbor vector that searches the minimum distance vector in a database with respect to each input vector.[15] On the basis of the matched results, the PDM stage determines the final recognition results through keypoint clustering and voting processes.[8] To implement the PDM stage, we used a vector matching accelerator (VMA) to accelerate the nearest neighbor matching with dedicated sum of absolute differences (SAD) vector distance calculation units.

In this processor, we exploit the grid tile as a basic computational unit for selective ROI processing, unlike conventional column-based processing.[1,4,5] A grid tile's size can vary. As the size decreases, we can finely extract multiple objects' ROI. However, processing efficiency lessens within a grid tile because the MIP stage's image processing kernel requires excessive neighborhood pixel data. In this trade-off, we selected a grid tile size of 40 × 40, which is sufficient to perform the SIFT algorithm.

Model verification

We devised our three-stage object recognition processor to recognize office items in real time for autonomous mobile robot applications. To this end, we chose 50 objects from the Columbia Object Image Library (COIL-100) as the target objects.[16]

To evaluate the proposed object recognition model's performance, we needed a large set of sample images containing target objects with natural backgrounds. For this, we created 2,400 images by combining the 50 target objects with six natural background images that contained different amounts of clutter (see Figure 4). For each background image, we synthesized 400 images with one to three trained objects of random locations, sizes, and orientations. We constructed the database using images of target objects taken at 30-degree increments. It includes a total of 17,239 object keypoint vectors.

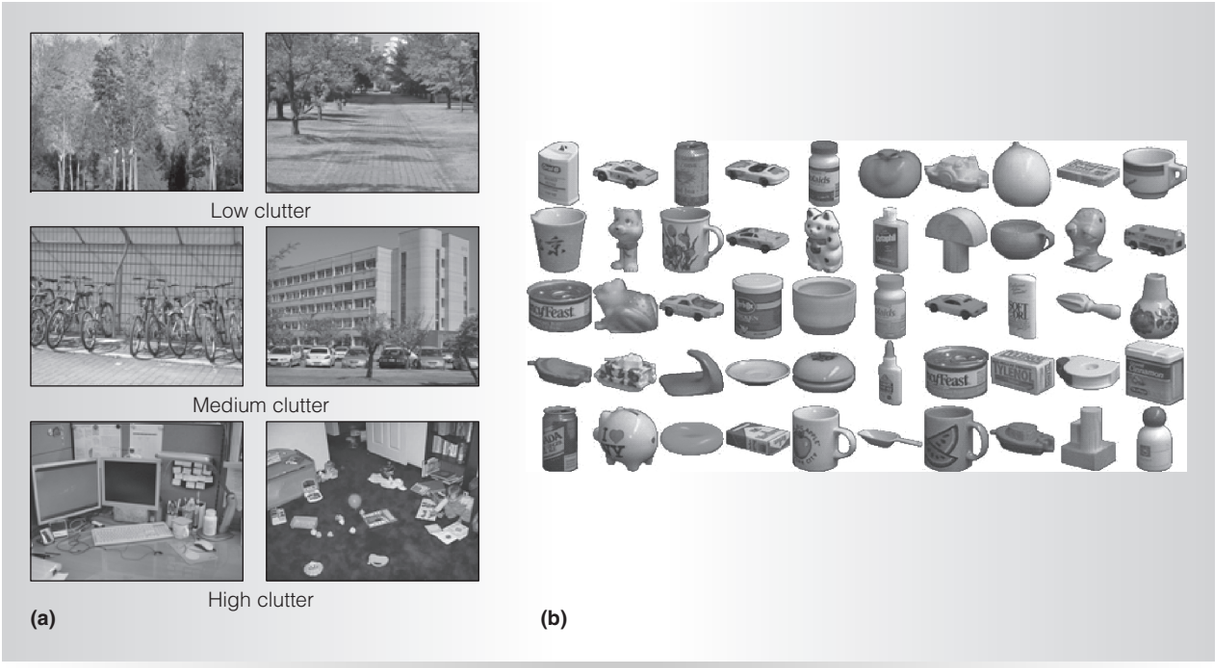


Figure 4. Simulation setup for algorithm verification. We choose 50 office items from the Columbia Object Image Library for target objects and six natural scenes containing different amounts of clutter for the image backgrounds.

For this set, the average correctly matched rate was 92 percent, without any false positives. The PDM stage also includes a filtering process that invalidates not-closely-matched keypoints with the database, as well as a keypoint clustering process that admits the final conclusive objects only when the matched keypoints are clustered together.[8] Because of this, the rate of false positive matches (where our object recognition model incorrectly recognized an object when the input image didn't include any database objects) was close to zero. By extracting ROI at the VP stage, we reduced the input image's average processing area to 32.8 percent, while the overall recognition rate was barely affected compared with the original SIFT algorithm.

Pipeline time balancing schemes

Figure 5 shows our proposed processor's overall block diagram. It consists of one NFC, 16 SIMD-PUs, one VMA, and two external memory interfaces. The power domain of the 16 SIMD-PUs separates into four power domains, and overall, 20 IP blocks are integrated by NoC interconnection.

To increase the proposed processor's throughput and use, we execute the three

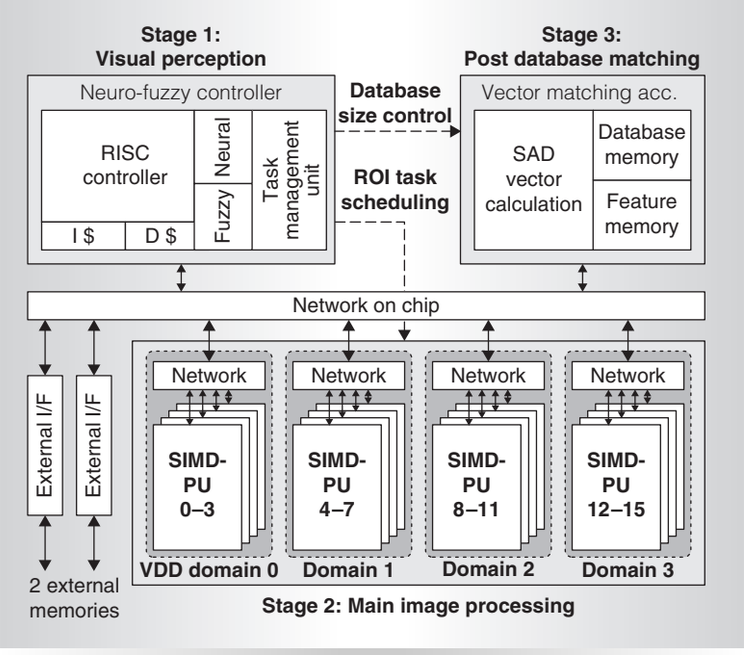
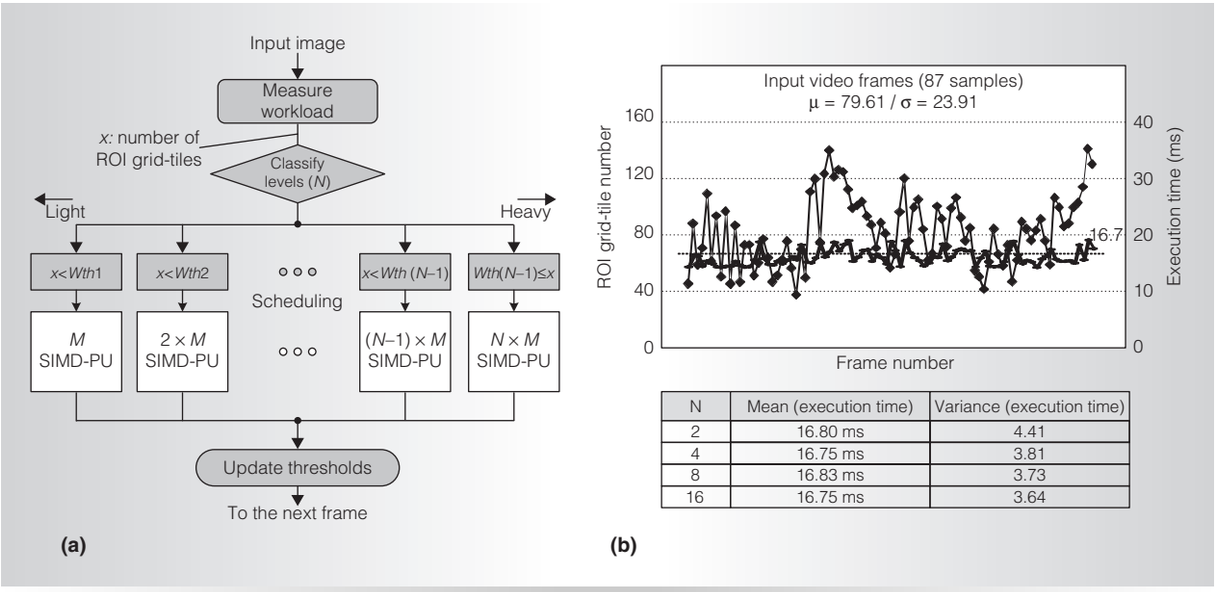


Figure 5. The proposed processor's overall block diagram. It consists of a neuro-fuzzy controller (NFC), 16 SIMD processing units (SIMD-PUs), and a vector matching accelerator (VMA), which are for visual perception, main image processing, and post database matching, respectively.



N	Mean (execution time)	Variance (execution time)
2	16.80 ms	4.41
4	16.75 ms	3.81
8	16.83 ms	3.73
16	16.75 ms	3.64

Figure 6. Workload-aware task scheduling: flow chart (a) and evaluation results (b). Workload-aware task scheduling regulates the overall execution time of main image processing to a constant level by allocating SIMD-PU resources to be proportional to their workload amount.

object recognition stages in the pipeline as shown in Figure 2d. The pipelined data are ROI grid tiles and descriptor vectors between the first and second stage and the second and third stage, respectively. In this coarse-grained task pipelining, the most important thing is to balance the pipeline stages' execution times. To this end, we first analyzed the characteristics of each stage's execution time.

Because the VP stage mainly contains repeated feature extraction processes for entire pixels of downsized input images, its execution time is constant under input image variations. On the other hand, the MIP and PDM stages have varying execution times owing to the amounts of their input workloads—that is, ROI grid tiles and descriptor vectors. The MIP stage's execution time is proportional to the number of ROI grid tiles, because this number determines the MIP stage's total processing area. Similarly, the PDM stage's execution time is proportional to the number of descriptor vectors. Although input vectors have varying matching times, the average matching time of more than 100 input vectors converts to a constant value and can be used as a parameter. Therefore, we can estimate the PDM stage's execution time with the number of descriptor vectors and average time parameter for a vector matching.

Under these conditions, the NFC, whose execution time is static, controls the execution times of the 16 SIMD-PUs and VMA to balance the three stages' execution times to target a pipeline time of 16 ms, or 60 fps. The NFC estimates the MIP and PDM stages' execution times using the number of ROI grid tiles and descriptor vectors, respectively, and controls them to conform to the target pipeline time. To this end, the NFC exploits two pipeline time-balancing schemes—workload-aware task scheduling (WATS) and applied database size control (ADSC)—for the MIP and PDM stages, respectively. As a result, the three stages' execution times are balanced to the target pipeline time even with the input images' variations, and our proposed processor achieves 60 fps for VGA (640 × 480) video input.

Workload-aware task scheduling

To control the MIP stage's execution time, the NFC performs WATS when it assigns extracted ROI grid tiles to the 16 SIMD-PUs. Figure 6a shows the WATS operating flow. First, the NFC measures the number of ROI grid tiles and classifies

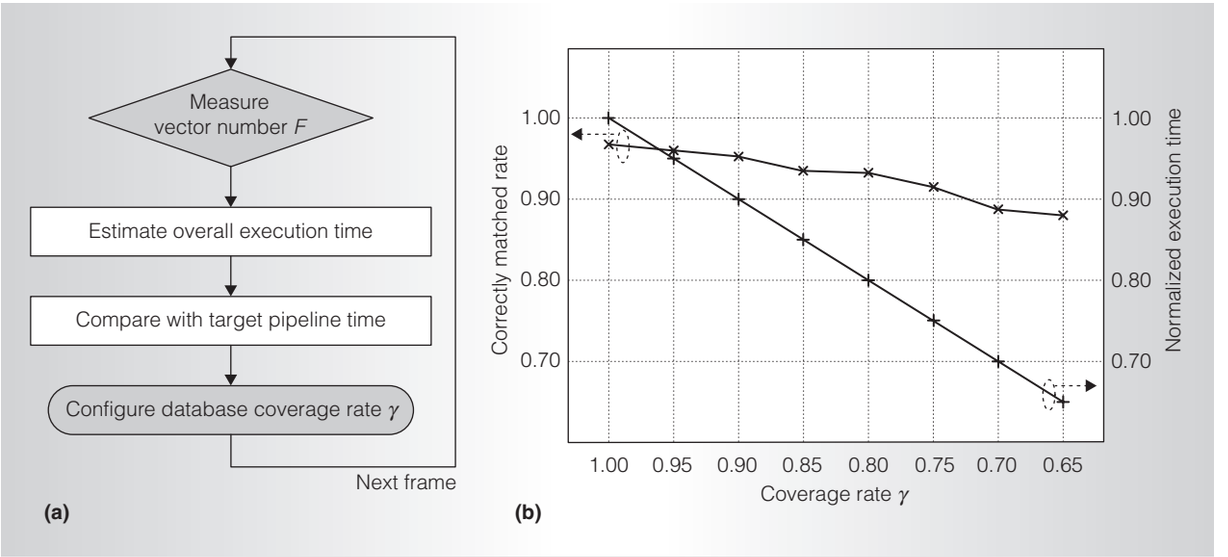


Figure 7. Applied database size control: flow chart (a) and evaluation results (b). The NFC controls the post database matching stage's execution time by adjusting the size of applied database. The recognition rate of the overall system can be decreased with a small database.

this number to one of N workload levels, divided by $N - 1$ threshold values. Then, it determines the number of operating SIMD-PUs for a frame according to the selected workload level. Because WATS determines the number of activating SIMD-PUs to be proportional to the workload, it regulates the overall SIMD-PUs' execution time to a constant level. The NFC can control the execution time by modifying the threshold values in the classification process.

To evaluate WATS' effects according to N, the number of workload levels, we measure the execution times of the MIP stage for 87 sample images from real video frames where N is 1, 2, 4, 8, and 16. The mean and variance of the number of ROI grid tiles of input images are 79.61 and 23.91, respectively, where the input image's size is VGA (640 × 480) and the grid tile's size is 40 × 40 pixels. The graph in Figure 6b shows the WATS' experimental results when N is 4. WATS manages the MIP stage's overall execution time in a constant level with a small variance even under large workload variations of input images. The table in Figure 6b summarizes the mean and variance values of evaluated execution times for various N values. As the N value increases, the variance decreases, while the mean remains constant. With large N configurations, the MIP stage's execution time is managed around a constant level more stably because we can precisely allocate resources for finely classified workload levels. We can control the mean value by adjusting threshold values in WATS. We can reduce the MIP stage's overall execution time to meet the pipeline time by lowering threshold values, which increases the number of operating SIMD-PUs for the same workloads. On the other hand, we can reduce the number of operating SIMD-PUs when the pipeline time budget is sufficient by increasing the threshold values.

Applied database size control

To control the PDM stage's execution time, the NFC performs ADSC. Using the nearest-neighbor matching algorithm[15] applied in the VMA, we can estimate the PDM stage's overall execution time using the equation $\textit{Execution time} = (\alpha + F \times \beta) \times \gamma$, where α is the database transfer time, β is a unit cycle time to match one descriptor vector with database, F is the number of total descriptor vectors, and γ is a coverage rate of database.

Figure 7a shows how the NFC adjusts the execution time using ADSC. First, it measures the number of input descriptor vectors

from the 16 SIMD-PUs. Then, it estimates the overall vector matching time using the equation, in case the coverage rate is 1. Finally, the NFC compares the estimated execution time with the target pipeline time and configures the VMA's database coverage rate to meet this target time. Figure 7b shows the ADSC's effects on the PDM's execution time and the rate of correctly matched objects. We measured the execution time and correctly matched rate for 400 synthesized images with a low clutter background under the same simulation environments we mentioned earlier, when the database's coverage rate varies from 1.0 to 0.65. As the graph shows, the PDM stage's execution time is linearly reduced as the applied database decreases in size. Meanwhile, the correctly matched rate slightly decreases when the coverage rate is linearly decreased. This is because each descriptor vector can be matched with a vector from the correct object even when the size of the applied database is reduced.

Neuro-fuzzy controller design

Figure 8 shows the NFC's overall block diagram, which broadly consists of a neuro-fuzzy visual predictor (NF-VP) and task/power management unit (TMU). The NF-VP performs ROI estimation using biologically inspired neural networks and fuzzy logic circuits, while the TMU handles control tasks such as ROI task scheduling, pipeline time balancing, and chip power management.

For human-like robust and efficient ROI estimation, the NF-VP employs three dedicated visual engines: a motion estimator, VAE, and object detection engine (ODE). The motion estimator extracts dynamic features of the motion vectors between the two sequential frames, and the VAE extracts static features such as intensity, color, and orientation and generates a saliency map by normalizing and combining the extracted feature maps. The ODE performs final ROI classification, which determines whether the target arbitrary pixel can join the object's interest region.

In designing the VAE and ODE, we employed biologically inspired cellular neural networks (CNNs) and neuro-fuzzy classifiers for fast feature extraction and robust classification, respectively. Figure 9 shows the VAE's block diagram and cell organizations.[9] The VAE consists of 40 × 40 cell arrays and a linear array of 40 processing elements. The cells perform storage and intercellular communication functions, while the processing elements process the cells' data. Each VAE cell consists of a 6T SRAM-based 4 × 8-bit register that stores intermediate data and the result data of the CNN operation, and an 8-bit 4-directional shift register that shifts the register data to adjacent cells. A shift operation on the entire chip requires only one cycle to complete. Because of the shift register's fully connected 2D cellular structure and the fast data movement among neighbor cells, the register's regional and collective processing accelerates the various feature extractions, which contain many image filtering and convolution operations.

Figure 10 shows the ODE's block diagram and transistor-level circuit implementations.[13] The ODE employs Gaussian fuzzy membership and single-layer neural networks, which measure similarity between the seed and target pixels and make the final decision in a classification, respectively. We use an analog-digital mixed-mode approach to exploit the advantages of both analog and digital circuits. We designed the ODE's data processing parts in analog circuits to reduce the area and power overhead of digital implementations of nonlinear Gaussian functions and neural synaptic multipliers. After DACs convert the 8-bit intensity, saliency, and location values of the seed and target pixels to analog signals, three Gaussian function circuits measure the two pixels' similarities for three metrics. The Gaussian function circuit is realized by combining the MOS differential pair and minimum follower circuit in the current configuration mode. The differential pair circuit outputs the symmetric differential signals, each of which has exponential nonlinearity characteristics, and the minimum follower circuit generates the Gaussian-like output by following the minimum between the two signals. After that, current-mode neural synaptic multipliers implemented by binary-weighted current mirrors merge the three measured similarities with their weight values, and the comparator circuit makes the final decision through thresholding. These analog

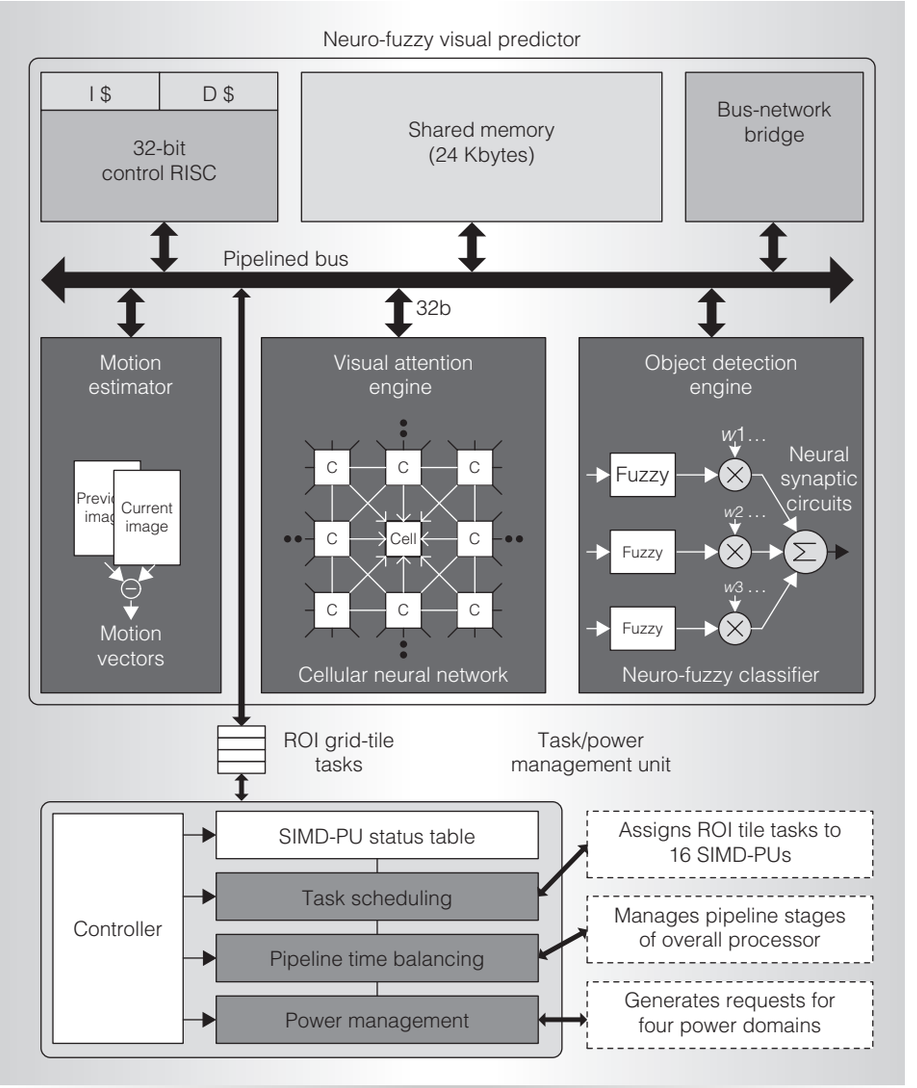


Figure 8. Block diagram of neuro-fuzzy controller. It broadly consists of a neuro-fuzzy visual predictor performing intelligent ROI estimation with neural and fuzzy processing and a task management unit controlling the rest of the IP blocks of the chip based on the estimated ROIs.

data processing parts of the ODE are robust to process, voltage, and temperature (PVT) variations. Because we can set initial connection weights and the Gaussian function's width externally, we can compensate variations by elaborately tuning them. In addition, the current operation mode benefits PVT variations because we need no additional circuits for summing signals. With tuning connection weights and Gaussian width, we can reduce a 12.91 percent fluctuation in the input signal to 2.23 percent in the final output signal. On the other hand, we implement a learning part that updates the weight values of neural networks in digital circuits for easy data storage management. With a simple unsupervised Hebbian learning,[17] only a multiplier and an adder are used for learning one weight value. As a result, analog-digital mixed-mode implementation reduces the ODE's area and power consumption by 59 percent and 44 percent, respectively, compared with those of digital-only implementations. With the four parallel execution units, the ODE completes the ROI detection for each region within 7 microseconds (μs) at 200 MHz operating frequency.

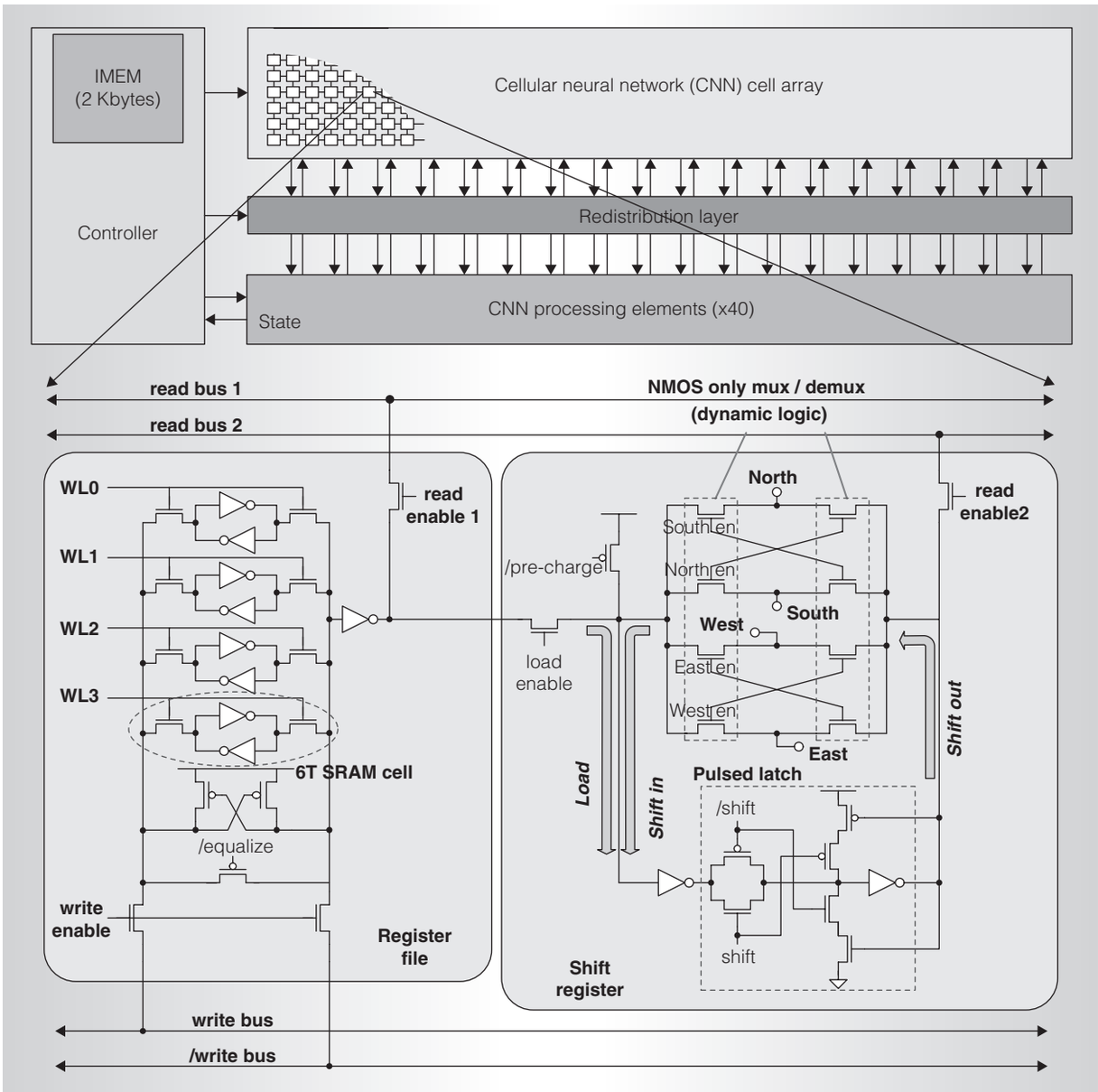


Figure 9. Cellular neural network-based visual attention engine (VAE). The VAE consists of 40 × 40 cell arrays for intercellular communication functions and a linear array of 40 processing elements for processing the cell's data. Each cell consists of four 8-bit registers and an 8-bit 4-directional shift register that shifts the register data to adjacent cells.

The remaining object recognition IPs of the proposed processor are the SIMD-PU and VMA, which we use for parallel image processing in the MIP stage and vector matching in the PDM stage, respectively. The SIMD-PU consists of eight dual-issued very long instruction word (VLIW) processing elements that can perform extended ALU operations for image data processing, such as two-way multiply/MAC, three-operanded min/max compare, and 32-bit accumulation in a single cycle.[6] The VMA employs a dedicated three-stage data path consisting of database vector read, SAD distance calculation, and distance comparison/update to accelerate nearest-neighbor vector search. We use a 512-bit wide bandwidth memory for database memory to increase the throughput in vector matching.

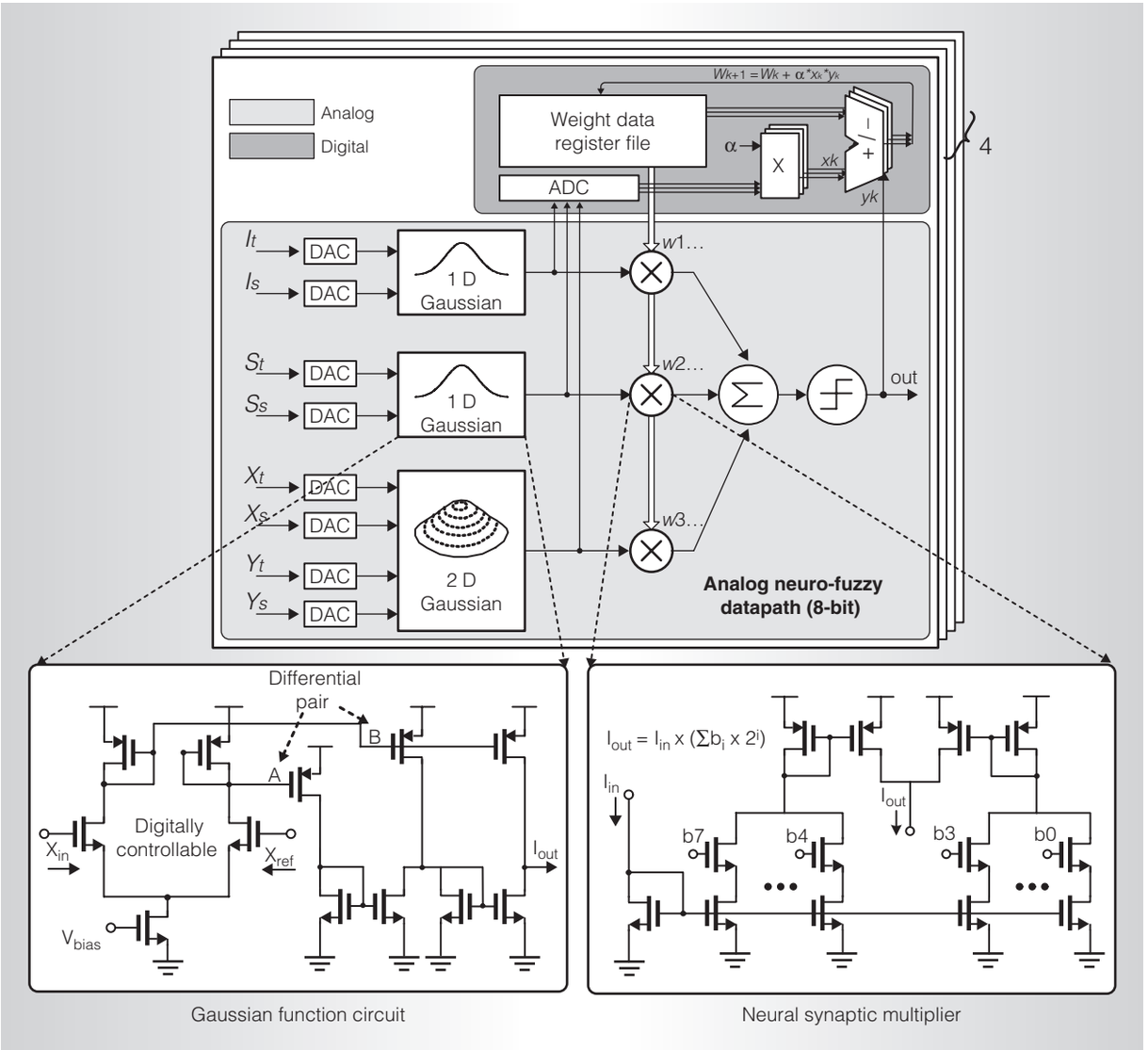


Figure 10. Neuro-fuzzy object detection engine (ODE). The ODE employs Gaussian fuzzy membership and single-layer neural networks, which measuring similarity between the seed and target pixels and final decision making in a classification, respectively. Its analog-based mixed-mode implementation alleviates the overhead of Gaussian function circuits and neural synaptic multipliers.

Workload-aware dynamic power management

To reduce power consumption in object recognition processing, the NFC performs dynamic power management for the 16 SIMD-PUs, the part of the processor that consumes the most power. Figure 11 shows the overall chip power architecture and its workload-aware dynamic power management. In the chip, the power domain of 16 SIMD-PUs is divided into four domains, and the NFC controls each of them independently. To control each power domain, we employ the off-chip power gating method[18] for low-cost implementation. The rest of the chip, including the NFC, VMA, and NoC, are in always-on domain.

For intelligent power management that reflects workload variations of input images, the NFC adopts workload-aware power

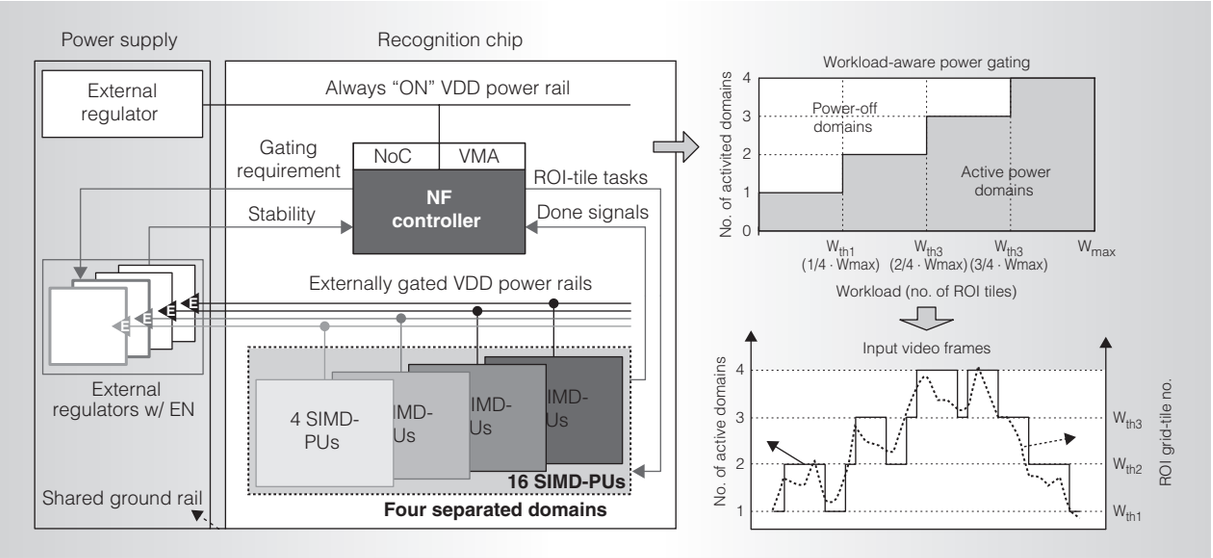


Figure 11. Chip power architecture and workload-aware dynamic power management. The NFC activates only required power domains of 16 SIMD processing units according to their input workloads with workload-aware task scheduling.

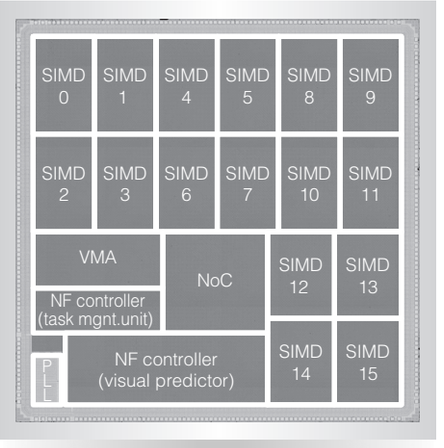


Figure 12. Chip micrograph. Most of the chip die area is spent by 16 SIMD processing units.

gating (WAPG). WAPG determines the number of activated power domains according to the workload levels based on the number of ROI grid tiles, like WATS. Simply, WAPG gates unused power domains of 16 SIMD-PUs, not assigned by WATS. When WATS determines the number of operating SIMD-PUs according to the workload level, WAPG determines the number of activating power domains. The only difference between WATS and WAPG is that we can vary the number of workload levels N in WATS but we're limited to four in WAPG. After that, the NFC sends request signals to external regulators to gate the unused power domains before it assigns the ROI grid tile tasks to the SIMD-PUs. The requests to external regulators occur only once per frame considering regulator's a few hundred microsecond (μs) settling time. By exploiting WAPG, the NFC ensures that the number of activated power domains follows the workload variations of the input frame (see the graph in Figure 11). To further reduce dynamic power in activated power domains, we apply software-controlled clock gating to each operating SIMD-PU. With dynamic power management by the WAPG and software-controlled clock gating, the 16 SIMD-PUs reduce their power consumption from 542 mW to 336 mW, a 38 percent reduction, while overall the processor consumes 496 mW of power.

Chip implementation and performance comparisons

The proposed processor is fabricated in 0.13 μm 1-poly 8-metal CMOS technology. Figure 12 shows the chip micrograph, and Table 1 summarizes its features. The 7 × 7 mm^2 chip area contains 36.4 million transistors including 3.7 million logic gates and

396 Kbytes on-chip SRAM. The operating frequency is 200 MHz for the IP blocks and 400 MHz for the NoC. Its peak performance amounts to 201.4 giga operations per second (GOPS) when 695 mW is dissipated. The average power consumption is 496 mW at the supply voltage of 1.2 V while the object recognition is running at 60 fps.

Table 2 compares the performance of our proposed processor with previous parallel and digital signal processors for vision processing.[3–5,19,20] Although XC core's target application is vehicle vision, other processors target SIFT-based object recognition. Among these, only the SIFT descriptor generation is parallelized on commercial CELL-BE[19] and Blackfin DSP,[20] and the NoC-based object recognition processor[3] without a database matching process. On the other hand, full object recognition based on the SIFT descriptor is implemented in the visual attention-based object recognition processor[5] and our proposed processor with database sizes of 5,632 and 16,384,

Table 1. Chip summary.

Chip characteristic	Description	
Technology	0.13 μm 1-poly 8-metal CMOS	
Package	320-pin FPGA	
Die size	7 mm × 7 mm	
Power supply	1.2-V core, 2.5-V I/O	
Operating frequency	200-MHz IPs, 400-MHz NoC	
Transistor counts	36.4 million transistors	
	3.73 million gates, 396-Kbyte SRAM	
Power consumption	Peak: 695 mW, Average: 496 mW	
Peak performance	NFC	54 GOPS
	16 SIMD-PUs	128 GOPS
	VMA	19.4 GOPS
	Total	201.4 GOPS
Power efficiency	290 GOPS/W	
Target application	Real-time object recognition	
Input screen	VGA (640 × 480 pixels)	
Frame rate	60 fps	
Database size	16384 vectors (50 objects)	

Table 2. Performance comparisons.

	XC[4]	CELL-BE[19]	Blackfin DSP[20]	CICC2007[3]	ISSCC-2008[5]	This work
Process	130 nm	90 nm	130 nm	180 nm	130 nm	130 nm
Chip size	Not reported	235 mm^2	24 mm × 24 mm (package)	38.5 mm^2	36 mm^2	49 mm^2
Transistor count	26.8M TRs	241M TRs		0.8M gates, 34 Kbytes	2M gates, 228 Kbytes	36M TRs
Power supply	1.2 V	0.9-1.3 V	1.2 V	1.8 V	1.2 V	1.2 V
Operating frequency	100 MHz	3.2 GHz	600 MHz	200 MHz	200 MHz	200 MHz
Peak performance	100 GOPS	>200 GFLOPS	1.2 GMACs	81.6 GOPS	125 GOPS	201.4 GOPS
Power consumption	<2 W	~50 W	280 mW (average)	1.08 W (average)	0.6 W (average)	0.5 W (average)
Target application	Vehicle area detection	SIFT descriptor generation	SIFT descriptor generation	SIFT descriptor generation	Object recognition: SIFT + single attention	Object recognition: SIFT + multiple attention
Application performance	32 fps	0.48 fps	0.88 fps	16 fps	22 fps	60 fps
Database size	N/A	N/A	N/A	N/A	5,632	16,384
Energy per frame	40 mJ	104 J	318 mJ	67.9 mJ	26.5 mJ	8.2 mJ
Screen resolution	256 × 240	UVGA (1,600 × 1,200)	CIF (352 × 288)	QVGA (320 × 240)	QVGA (320 × 240)	VGA (640 × 480)
Energy/pixel	651 nJ	54 μJ	3.1 μJ	884 nJ	345 nJ	26.7 nJ

respectively. Because of its neuro-fuzzy controlled task pipelining with two pipeline time-balancing schemes, the proposed processor achieves 60 fps, which is more than 2.8 times higher than the others. The workload-aware dynamic power management reduces the power consumption of 16 SIMD-PUs by 38 percent and achieves 496 mW total power consumption. The proposed processor's overall power consumption is less than that of previous processors, even though it sustains higher frame-rate operation. As a result, the proposed processor consumes only 8.2 mJ energy per frame, more than 3.2 times less than the other processors.

Our bio-inspired object recognition processor series are evolving to have more intelligence in silicon devices. With improved neural networks and fuzzy reasoning, the next version will provide more accurate attention and finer control of multiple processing cores to achieve more robust, energy-efficient object recognition. MICRO

References

1. S. Kyo et al., ''A 51.2 GOPS Scalable Video Recognition Processor for Intelligent Cruise Control Based on a Linear Array of 128 4-Way VLIW Processing Elements,'' *IEEE Int'l Solid-State Circuits Conf. Digest of Technical Papers,* IEEE CS Press, 2003, pp. 48-477.
2. A. Abbo et al., ''XETAL-II: A 107 GOPS, 600 mW Massively Parallel Processor for Video Scene Analysis,'' *IEEE J. Solid-State Circuits,* vol. 43, no. 1, 2008, pp. 192-201.
3. D. Kim et al., ''An 81.6 GOPS Object Recognition Processor Based on NoC and Visual Image Processing Memory,'' *IEEE Custom Integrated Circuits Conf.* (CICC 07), IEEE CS Press, 2007, pp. 443-446.
4. S. Kyo et al., ''A Low-Cost Mixed-Mode Parallel Processor Architecture for Embedded Systems,'' *Proc. 21st Ann. Int'l Conf. Supercomputing,* ACM Press, 2007, pp. 253-262.
5. K. Kim et al., ''A 125 GOPS 583 mW Network-on-Chip Based Parallel Processor with Bio-Inspired Visual Attention Engine,'' *IEEE J. Solid-State Circuits,* vol. 44, no. 1, 2009, pp. 136-147.
6. J.-Y. Kim et al., ''A 201.4 GOPS 496 mW Real-Time Multi-Object Recognition Processor with Bio-Inspired Neural Perception Engine,'' *IEEE Int'l Solid-State Circuits Conf. Digest of Technical Papers* (ISSCC 09), IEEE CS Press, 2009, pp.150-151.
7. Q. Zhang et al., ''SIFT Implementation and Optimization for Multi-Core Systems,'' *IEEE Int'l Symp. Parallel and Distributed Processing* (IPDPS 08), IEEE CS Press, 2008, pp. 1-8.
8. D.G. Lowe, ''Distinctive Image Features from Scale-Invariant Keypoints,'' *ACM Int'l J. Computer Vision,* vol. 60, no. 2, 2004, pp. 91-110.
9. S. Lee et al., ''The Brain Mimicking Visual Attention Engine: An 80x60 Digital Cellular Neural Network for Rapid Global Feature Extraction,'' *2008 IEEE Symp. VLSI Circuits,* IEEE CS Press, 2008, pp. 26-27.
10. L. Itti et al., ''A Model of Saliency-based Visual Attention for Rapid Scene Analysis,'' *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 11, 1998, pp. 1254-1259.
11. D. Walther et al., ''Selective Visual Attention Enables Learning and Recognition of Multiple Objects in Cluttered Scenes,'' *Computer Vision and Image Understanding,* vol. 100, nos. 1-2, 2005, pp. 41-63.
12. N. Ouerhani, ''Visual Attention: From Bio-Inspired Modeling to Real-Time Implementation,'' doctoral dissertation, Institute of Microtechnology, Univ. of Neuchatel, Switzerland, 2003.
13. M. Kim et al., ''A 22.8 GOPS 2.83 mW Neuro-Fuzzy Object Detection Engine for Fast Multi-Object Recognition,'' *2009 IEEE Symp. VLSI Circuits,* IEEE CS Press, 2009, pp. 260-261.
14. F. Rothganger et al., ''3D Object Modeling and Recognition Using Local Affine-Invariant Image Descriptors and Multi-View Spatial Constraints,'' *Int'l J. Computer Vision,* vol. 66, no. 3, 2006, pp. 231-259.
15. J.-Y. Kim et al, ''A 66 fps 38 mW Nearest Neighbor Matching Processor with Hierarchical VQ Algorithm for Real-Time Object Recognition,'' *IEEE Asian Solid-State Circuits Conf.* (A-SSCC 08), IEEE CS Press, 2008, pp. 177-180.
16. S.A. Nene, S.K. Nayar, and H. Murase, *Columbia Object Image Library (COIL-100),*

tech. report CUCS-006-96, Computer Science Dept., Columbia Univ., 1996.
17. D.O. Hebb, *The Organization of Behavior,* John Wiley & Sons, 1949.
18. M. Keating et al., *Low Power Methodology Manual for System-on-Chip Design,* Springer, 2007.
19. B. Kwon et al., ''Parallelization of the Scale-Invariant Keypoint Detection Algorithm for Cell Broadband Engine Architecture,'' *Proc. 5th IEEE Consumer Comm. and Networking Conf.* (CCNC 08), IEEE CS Press, 2008, pp. 1030-1034.
20. T. Ko et al., ''Exploring Tradeoffs in Accuracy, Energy, and Latency of Scale Invariant Feature Transform in Wireless Camera Networks,'' *Proc. 1st ACM/IEEE Int'l Conf. Distributed Smart Cameras* (ICDSC 07), IEEE CS Press, 2007, pp. 313-320.

Joo-Young Kim is a PhD student in electrical engineering and computer science at the Korea Advanced Institute of Science and Technology. His research interests include computer architecture and design for many-core vision processors. Kim received his MS in electrical engineering and computer science from the Korea Advanced Institute of Science and Technology. He's a student member of the IEEE.

Minsu Kim is a PhD student in electrical engineering and computer science at the Korea Advanced Institute of Science and Technology. His research interests include network-on-chip-based SoC design and bio-inspired VLSI architecture for vision processing. Kim received his MS in electrical engineering and computer science from the Korea Advanced Institute of Science and Technology. He's a student member of the IEEE.

Seungjin Lee is a PhD student in electrical engineering and computer science at the Korea Advanced Institute of Science and Technology. His research interests include investigating parallel architectures for computer vision processing. Lee received his MS in electrical engineering and computer science from the Korea Advanced Institute of Science and Technology. He's a student member of the IEEE.

Jinwook Oh is an MS student in electrical engineering and computer science at the Korea Advanced Institute of Science and Technology. His research interests include low-power digital signal processors for computer vision and, more recently, VLSI implementation of neural networks and fuzzy logics. Oh received his BS in electrical engineering and computer science from Seoul National University. He is a student member of the IEEE.

Sejong Oh is a postdoctoral research scientist at Korea Advanced Institute of Science and Technology. His interests include compiler optimization, multicore architecture, and design automation for embedded systems. Oh received his PhD in electrical engineering and computer science from the Korea Advanced Institute of Science and Technology. He's a student member of the IEEE.

Hoi-Jun Yoo is a full professor in the Department of Electrical Engineering at the Korea Advanced Institute of Science and Technology. His research interests include network on chips, 3D graphics, bio-inspired vision processors, body area networks, biomedical devices and circuits, and memory circuits and systems. Yoo received his PhD in electrical engineering from the Korea Advanced Institute of Science and Technology. He's a Fellow of the IEEE.

Direct questions and comments to Joo-Young Kim, Division of Electrical Eng., School of Electrical Eng. and Computer Science, Korea Advanced Inst. of Science and Technology, 335 Gwahak-ro, Yuseong-gu, Daejeon, 305-701, Republic of Korea; trample7@eeinfo.kaist.ac.kr.