Signal Processing: *Image Communication* ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

An attention controlled multi-core architecture for energy efficient object recognition

Joo-Young Kim *, Sejong Oh, Seungjin Lee, Minsu Kim, Jinwook Oh, Hoi-Jun Yoo

Division of Electrical Engineering, School of Electrical Engineering and Computer Science, KAIST, 335 Gwahak-ro, Yuseong-gu, Daejeon 305-701, Republic of Korea

ARTICLE INFO

Article history:
Received 2 October 2009
Accepted 8 March 2010

Keywords:
Attention controlled
Multi-core architecture
Object recognition
Visual attention
Energy efficient

ABSTRACT

In this paper, an attention controlled multi-core architecture is proposed for energy efficient object recognition. The proposed architecture employs two IP layers having different roles for energy efficient recognition processing: the attention/control IPs compute regions-of-interest (ROIs) of the entire image and control the multiple processing cores to perform local object recognition processing on selected area. To this end, a task manager is proposed to perform dynamic scheduling of various ROI tasks from the attention IP to multiple cores in a unit of small-sized grid-tile. Thanks to a number of grid-tile threads generated by the task manager, the utilization of the multiple cores amounts to 92% on average. As a result, the proposed architecture achieves 2.1× energy reduction in multi-core recognition system by indicating processing cores to focus on critical area of the image with a 0.87 mJ attention processing. Finally, the proposed architecture is implemented in 0.13 μm CMOS technology and the fabricated chip verifies 3.2× lower energy dissipation per frame than the state-of-the-art object recognition processor.

© 2010 Published by Elsevier B.V.

1. Introduction

Object recognition is the process of identifying objects out of an input image by matching their features with trained database of target objects. It has been a core technology for computer vision applications such as autonomous vehicles, autonomous mobile robots, and surveillance systems [1–5], and also can be applied to video applications such as JPEG/MPEG encoding and decoding [6]. Fig. 1 shows a simplified process of the scale invariant feature transform (SIFT) [7], a commonly used object recognition algorithm. First, various scale spaces of an input image are generated by Gaussian filtering operations with different coefficient values, and difference-of-Gaussian (DoG) images are generated by subtracting two neighboring image spaces. Then, object key-points are localized by searching local maxima/minima points among three neighboring DoG images. Each key-point is converted to a descriptor vector to describe its magnitude and orientation characteristics. After this SIFT key-point description, generated vectors are matched with the object database for the final recognition.

Since each stage of object recognition requires huge amount of computations, it is difficult to achieve a real-time operation with a conventional single general-purpose processor (GPP). Zhang et al. [8] reveals a 2.33 GHz Intel Core2 processor can only obtain 2 frame/s performance when it runs the SIFT algorithm for VGA (640 × 480) sized input video, and they show that real-time operation over 30 frame/s for the SIFT can be achieved by parallelizing it on a multi-core system containing two 2.33 GHz Intel Core2 Quad processors (8 cores). However, a multi-core system using GPPs is a costly solution in terms of silicon area, power, and energy consumption. For example, the power consumption of the

* Corresponding author. Tel.: +82 42 350 8931; fax: +82 42 350 3410.
E-mail address: jooyoung.kim@kaist.ac.kr (J.-Y. Kim).

0923-5965/$ - see front matter © 2010 Published by Elsevier B.V.
doi:10.1016/j.image.2010.03.003

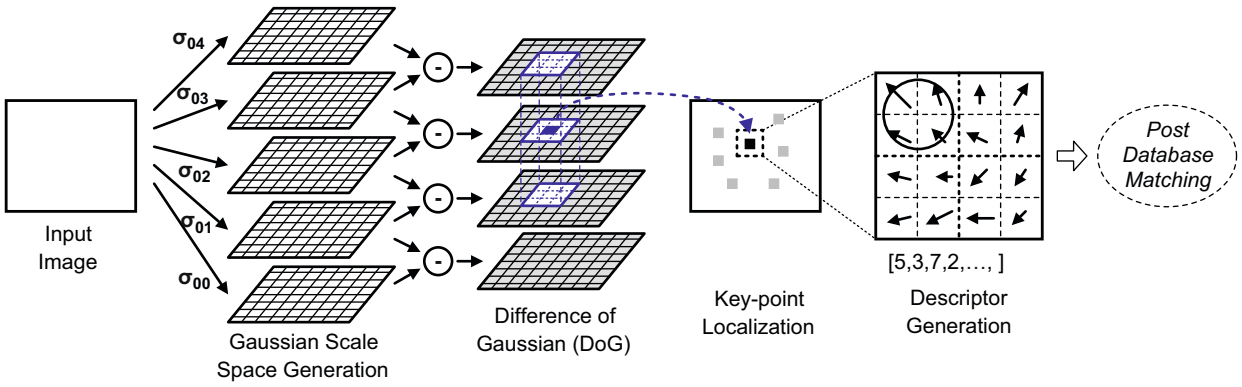


Fig. 1. SIFT based object recognition process.

multi-core system of [8] is more than 100 W, which is usually not appropriate for mobile applications.

To achieve real-time object recognition with considerably lower power, several embedded multi-core processors have been developed in recent years [1–5]. Abbo et al. [1] and Kyo et al. [2] present massively parallel single instruction multiple data (SIMD) architectures that maximize data-level parallelism in pre-processing stages of object recognition. Kim et al. [3] present a multi-core architecture including 10 processing units and 8 channel memories to exploit task-level parallelism over data-level parallelism in object recognition. Kim et al. [4] propose a special hardware IP named visual attention engine (VAE) [9] in a multi-core architecture with 8 SIMD processing cores. This processor proves that the concept of visual attention [10] can reduce the computational costs of object recognition in real system implementation. The proposed VAE extracts the saliency map out of the input image and filters the key-points extracted by the processing cores based on the map. By reducing the number of valid key-points, workloads of key-point based task such as descriptor generation and further database matching can be reduced. However, the VAE plays a role in just a pre-processing filter in the architecture because it does not have any controlling ability over multiple cores except reducing the number of key-points. The pixel based tasks such as Gaussian scale space generation, DoG generation, and local maxima/minima search do not benefit from the VAE, and are executed by multiple cores with a conventional column-wise processing for the whole area of image.

In this paper, we propose an attention controlled multi-core architecture for energy efficient object recognition. This architecture introduces two different IP layers having different roles for more energy efficient processing: the attention/control IPs that estimate regions-of-interest (ROIs) as a global workload estimator for the entire image and control the operations of multiple processing cores, and multiple processing cores that perform local object recognition processing focusing on those areas. Based on the ROI results, the introduced task manager dynamically controls the tasks and threads of multiple cores with a fine-grained grid-tile based processing model. It also manages the overall processing speed and resource allocation of multiple cores by differentiating task scheduling methods. The rest of this paper is organized as follows. In Section 2, the visual perception, an enhanced attention processing that extracts the ROIs out of an input image, is introduced and the overall object recognition algorithm based on it will be briefly covered. In Section 3, the attention controlled multi-core architecture is proposed for energy efficient object recognition. Its grid-tile based processing model, application mapping, workload distribution, and scheduling method will be explained. The overall energy saving effect of the proposed architecture will be evaluated. Section 4 describes the implementation of visual perception in energy efficient way with cellular neural networks and neuro-fuzzy classifier. Section 5 shows the silicon realization of the proposed architecture. After that, this paper will be summarized in Section 6.

2. Visual perception based object recognition algorithm

Fig. 2 shows the flow diagram of the proposed visual perception based object recognition algorithm. It combines two different algorithms, visual perception and general object feature description algorithm described in Section 1. The visual perception, an enhanced attention processing, extracts the regions-of-interest (ROIs) out of the input image to provide more complete attention information than previous one [4]. Firstly, the visual perception adopts Itti's visual attention model [10] to extract several bottom-up features such as intensity, color, orientation, and motion, and promote them to a single unified map named saliency map. Then, it selects the most salient parts of the image as the seed points and performs region growing from seed points via repeated homogeneity classifications to determine the ROIs of objects [11]. For the feature description part, we employ the SIFT algorithm [7]. Consequently, object recognition processing can be selectively performed in this algorithm with the visual perception stage that extracts the ROIs of input image prior to detailed object feature description stage.

Please cite this article as: J.-Y. Kim, et al., An attention controlled multi-core architecture for energy efficient object recognition, Signal Process. Image Commun. (2010), doi:10.1016/j.image.2010.03.003

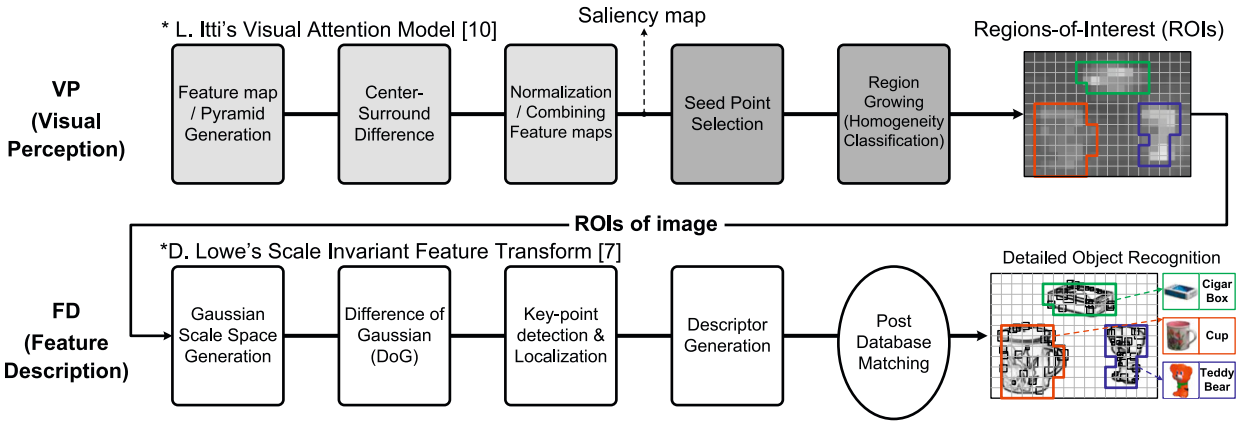


Fig. 2. Visual perception based object recognition.

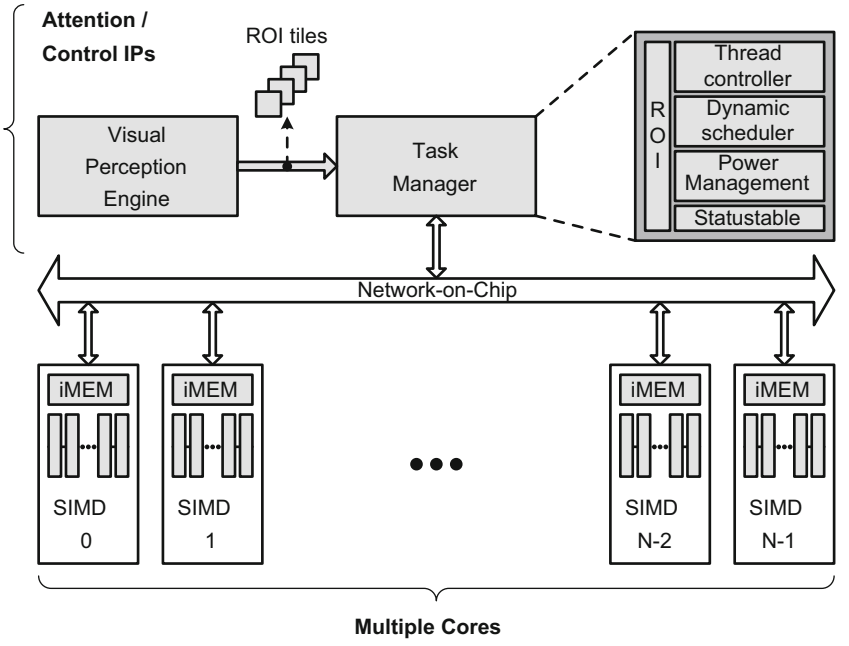


Fig. 3. Proposed attention controlled multi-core architecture.

3. Attention controlled multi-core architecture

3.1. Overall architecture

Fig. 3 shows the proposed attention controlled multi-core architecture. Unlike the conventional architectures that increase parallelism simply by exploiting more processing cores, this architecture employs two different IP layers – attention/control IP layer and parallel processing layer – having different roles for energy efficient recognition processing. The attention/control IP layer estimates global workload of overall input image, i.e., regions-of-interest, and controls multiple cores in parallel processing layer to run application program focusing on those areas, which results in more energy efficient processing. The attention/control IP layer includes a visual perception engine (VPE) and task manager (TM), and the parallel processing layer consists of N multiple cores. We call each of them an SIMD processor unit (SPU). The SPU core is a fully programmable device and each of N SPU cores can run an independent program.

The VPE is a special IP to extract attention regions out of an input image by performing the visual perception algorithm described in Section 2. As a result, it represents the ROIs of the input image in the unit of small-sized tile called grid-tile. For example, the ROIs of a VGA (640×480 pixels) sized image can be built by a number of grid-tiles whose size is 40×40 pixels. The SPU cores are employed to perform a feature description algorithm such as SIFT for the selected grid-tile area by the VPE. It exploits dual-issued VLIW instructions whose length is 51-bit and

Please cite this article as: J.-Y. Kim, et al., An attention controlled multi-core architecture for energy efficient object recognition, Signal Process. Image Commun. (2010), doi:10.1016/j.image.2010.03.003

8-way ALU datapaths for SIMD extended instructions in order to accelerate kernel image processing tasks such as Gaussian image filtering, gradient computation, and histogram operation. More detailed information about an SPU core is presented in [5]. The TM takes a role in control of multiple SPU cores. It performs dynamic scheduling of the ROI tile tasks from the VPE to SPU cores and manages multiple threads of SPU cores based on their centralized status table when they run whole application program. In addition, it can control the overall execution time of the SPU cores by its scheduling method. It can assign the grid-tile tasks as fast as possible to idle cores for real-time applications or can save the SPU core resources to meet low power requirements.

3.2. Grid-tile based ROI processing model

Fig. 4 shows a grid-tile based ROI processing model of the proposed architecture (Fig. 4b) with a conventional column-wise processing model (Fig. 4a). Most of the conventional multi-core architectures for image processing applications adopt the column-wise processing model that each processing core performs processing for the designated column region of the input image. By doing this, the workloads of overall image processing are well distributed to multiple cores and data-level parallelism is easily exploited by them in this model. However, the conventional column-wise processing cannot be applied to the proposed attention controlled architecture because attended regions will be different every time according to each input image. By assigning specific regions to specific cores like conventional model, the workloads of ROI tile tasks would not be distributed among processing cores. To handle this, we introduce a grid-tile based ROI processing model with a dynamic scheduler that assigns the ROI grid-tiles to different processing cores in a grid-tile unit every single frame. With the status table of processing cores, it can dynamically re-assign remaining grid-tile tasks to idle cores. Therefore, this grid-tile based ROI processing model is suitable in the proposed architecture to distribute the workloads of overall image, i.e., the ROIs of the input image, to multiple cores.

3.3. Application mapping

There are two strategies in application mapping in multi-core architecture. The first one is multiprocessor programming. In this scenario, multiple cores perform a one large process together while each core performs a different kind of small thread as a part of the whole process. For example, for the SIFT algorithm mapping case, cores 0–3 are mapped for Gaussian scale space generation, core 4 is for DoG computation, cores 5–7 are for key-point localizations, and so on. Therefore, intermediate data communications should be carried in this application mapping and task-level parallelism is achieved because different cores perform different kinds of tasks at the same time. However, this multiprocessor programming requires different kinds of programming for different multiple cores with program and data synchronization issues in data communications. Typically, multiprocessor programming demands much more time and efforts than single processor programming and its performance gain is limited due to the data dependency among cores. In the proposed architecture, each SPU core contains a 16-bit barrier register for program synchronization in multiprocessor programming. The TM in this case should do complex partitioning of the ROI tasks, configure each core's program, and control threads of them. The second strategy is to divide the whole process by the region while each core runs the whole application program. For example, core 0 performs SIFT for the area of grid-tile 0 while core 1 performs it for the area of grid-tile 1. In this scenario, multiple cores can achieve data and thread-level parallelism. Data communications among cores occurring in boundary condition are less than the multiprocessor programming case. Table 1 summarizes the two methods of application mapping in multi-core architecture.

Since we can avoid hard programming efforts and synchronization issues by adopting the second method, we apply it to the SIFT mapping to the proposed architecture. The SIFT kernel applicable for any tile size is mapped to each SPU core and the TM schedules the ROI grid-tile tasks to multiple SPU cores and manages the

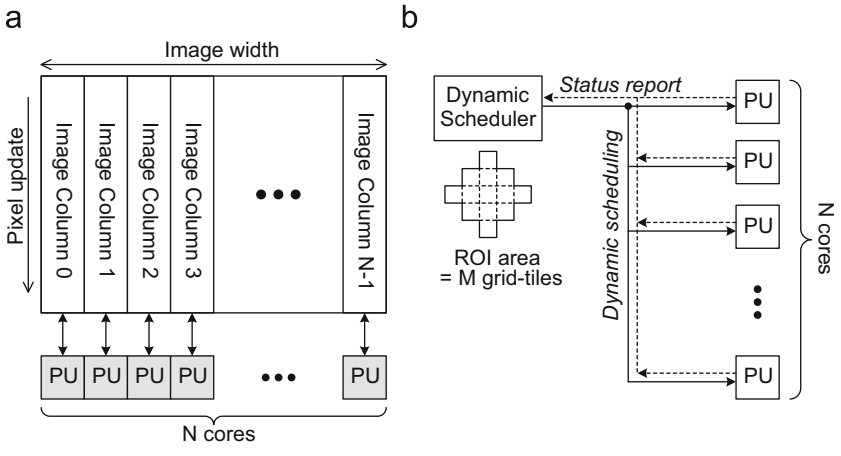


Fig. 4. (a) Column-wise processing model and (b) grid-tile based ROI processing model.

Please cite this article as: J.-Y. Kim, et al., An attention controlled multi-core architecture for energy efficient object recognition, Signal Process. Image Commun. (2010), doi:10.1016/j.image.2010.03.003

Table 1
Two methods in application mapping.

	Programming efforts	Inter-processor data transactions	Issues	Parallelism
Multiprocessor programming	High (multiple programs)	Frequently occurs	Program and data synchronization	Task-level parallelism
Divide by region	Low (use same program)	Not frequently occurs	Boundary region data sharing	Data-level parallelism Thread-level parallelism

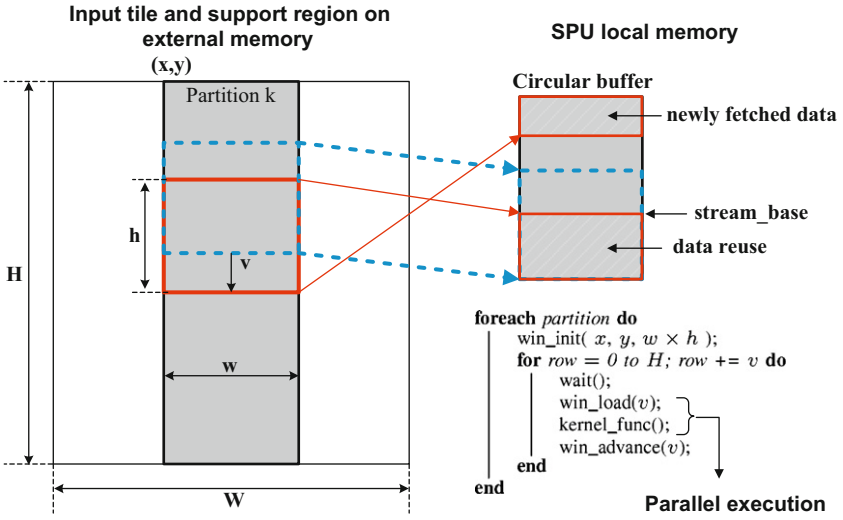


Fig. 5. Software mapping on SPU core.

threads. In the SIFT mapping to a single SPU core, a streaming approach is used to reduce the size of local embedded memory as shown in Fig. 5. A mapped task fetches input stream into the local memory by the DMA, it performs data calculations, and the output stream is written back via the DMA. The DMA is effective to fetch and store such partial streams with hiding data fetch overhead because it runs in parallel with the SIMD datapaths of SPU core. The pseudo code in the figure shows an example of processing partial 2-dimensional streams. For each partition, the *win_init* function loads the initial window as large as $w \times h$ pixels and the inner loop advances the window by $w \times v$ pixels per iteration. The wait function waits until the DMA is idle. Software pipelining technique is applied to the inner loop to hide the latency of data fetch. The *win_load* function loads the pixel data for the next iteration, and it has no flow dependence to the *kernel_func* function. Therefore, the *kernel_func* processes data fetched at the pervious loop iteration without waiting for the completion of *win_load*. The window processing creates a logical circular buffer in the local memory, so that the overlapped region across two consecutive loop iterations can be reused. The SPU's software development kit provides a runtime library including the 2-dimensional window operations and a compiler based on GNU C Compiler. While developing the applications, most of the source code is compiled by the compiler and small hotspot kernels are developed in inline assembly.

3.4. Scheduling method

In the proposed architecture, the TM can control the usage of multiple SPU cores based on the ROI results of the VPE. It can slow down the overall process with resource saving or speed it up for real-time performance by exploiting different scheduling methods. Since the number of ROI grid-tiles can be used as a workload estimator of all SPU cores, the resource allocation of SPU cores can be controlled based on it. Fig. 6 shows the flowchart of the proposed workload-aware task scheduling (WATS) that differentiates the number of operating cores based on the measured workloads. This scheduling method aims at intelligent resource allocation and power saving by performing it.

At the beginning, the TM counts the number of ROI grid-tiles from the VPE to measure the overall workload of input image. It classifies the measured workload to 1 of W workload levels to represent the amount of workloads of current frame. The W workload levels are divided by $W-1$ threshold values that can be updated in frame basis. Then, it determines the number of operating SPU cores and corresponding power domains according to the selected

Please cite this article as: J.-Y. Kim, et al., An attention controlled multi-core architecture for energy efficient object recognition, Signal Process. Image Commun. (2010), doi:10.1016/j.image.2010.03.003

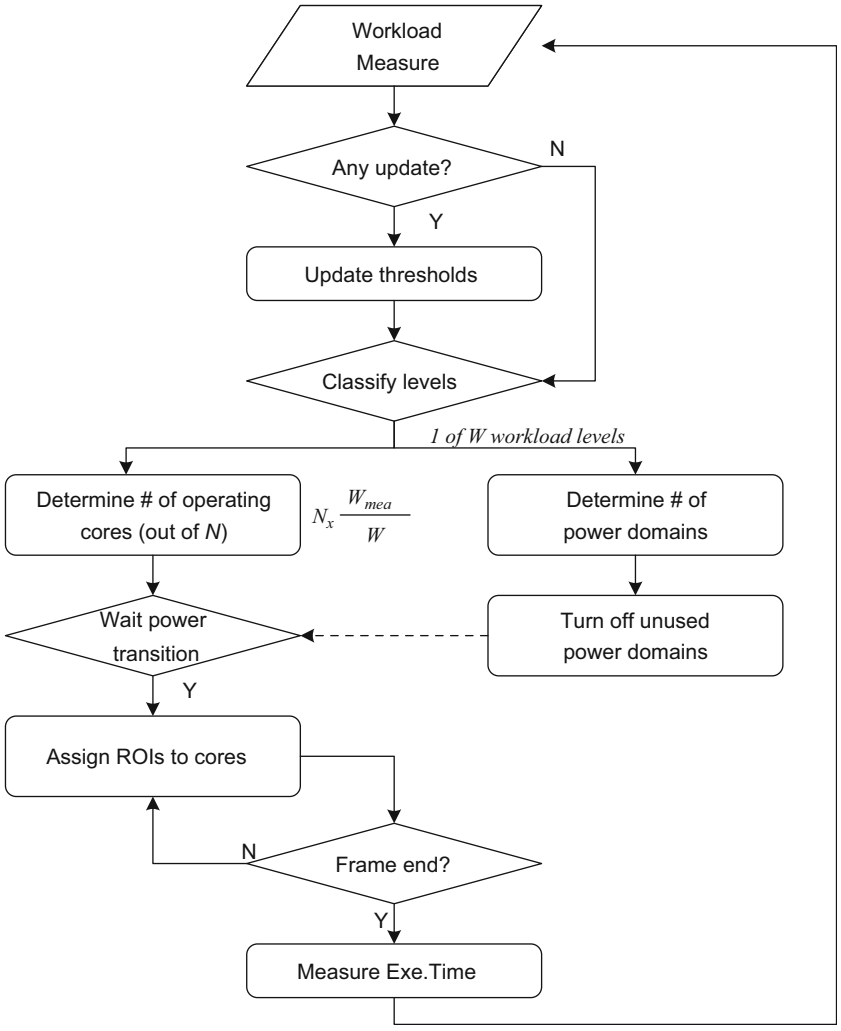


Fig. 6. Flowchart of workload-aware task scheduling.

workload level *Wmea*. Simply, we can determine the number of operating cores out of *N* cores to be a linearly proportional, *Nx* (*Wmea*/*W*). Power co-optimization is also performed by gating off the power domains of unused cores. The WATS only activates the appropriate amount of cores and power domains, and saves the rest of them. After waiting a hundred μs power transition periods of the domains, the TM starts to assign the ROI grid-tile tasks to the active SPU cores. In our implementation case, which will be described later in detail in Section 5, we choose the number of SPU cores and their power domains to 16 and 4, respectively. Since the WATS determines the number of operating SPU cores in proportion to the workload amount, the execution time of whole frame is managed around a constant value. Unless all the ROI grid-tile tasks are assigned to active SPU cores, the TM aggressively assigns remaining tasks to idle cores based on their status table. Completing all of the ROI grid-tile tasks, the TM measures the overall execution time of current frame for further information to update the threshold values. The execution time can be shortened or lengthened by decreasing and increasing the threshold values, respectively. This WATS shows the execution time of whole process and resource saving of multiple SPU cores can be managed at the same time. A scheduling method of the TM can be changed by suitably programming it for the purpose of overall system.

3.5. Experimental results

To evaluate the performance of the proposed attention controlled multi-core architecture, we perform experiments on 100 sample images from real video frames with a cycle accurate simulator in C++/System C. The size of input image is the VGA (640 × 480) and the size of a grid-tile can be configured from 24 × 24, 32 × 32, … to 72 × 72 pixels, by 8 pixels step both in width and height. In the experiments, the VPE extracts the ROIs of the input image in a grid-tile unit and the TM schedules them to *N* SPU cores with or without any scheduling method. The parameterized SIFT program is mapped to each SPU core

Please cite this article as: J.-Y. Kim, et al., An attention controlled multi-core architecture for energy efficient object recognition, Signal Process. Image Commun. (2010), doi:10.1016/j.image.2010.03.003

and the N value is set to 16. The experiments cover 4 issues of the proposed architecture. First, the utilizations of multiple SPU cores are evaluated. Second, the performance of the grid-based ROI processing architecture is analyzed according to the size of the grid-tile. Third, execution time management, and resource usage of the WATS will be covered. Last, the overall energy reduction effect by the proposed attention controlled architecture will be shown.

Fig. 7 shows the hardware utilizations of 16 SPU cores when the TM schedules the ROI grid-tile tasks to all of the 16 SPU cores without any special scheduling method. The size of grid-tile is set to 40×40 pixels. For 100 frame images, the average activation time of each SPU core is measured. With a number of ROI grid-tile tasks, the average utilization of all SPU cores is calculated as 92% while ranging from 87% to 99%. This result shows attention controlled multiple processing cores are highly utilized in the proposed architecture.

Fig. 8 shows the performance of the grid-based ROI processing according to the size of a grid-tile. We measure the average execution time per ROI grid-tile and the average number of ROI grid-tiles for the 100 sample images where the size of a grid-tile varies from 24×24 to 72×72 with a step of 8×8. As a result, the execution time of an ROI grid-tile linearly increases with the size of grid-tiles. Since the SPU core contains SIMD extended 1-dimensional vector processing, the overall execution time is linear even though the processed area increases quadratically. On the other hand, the number of ROI

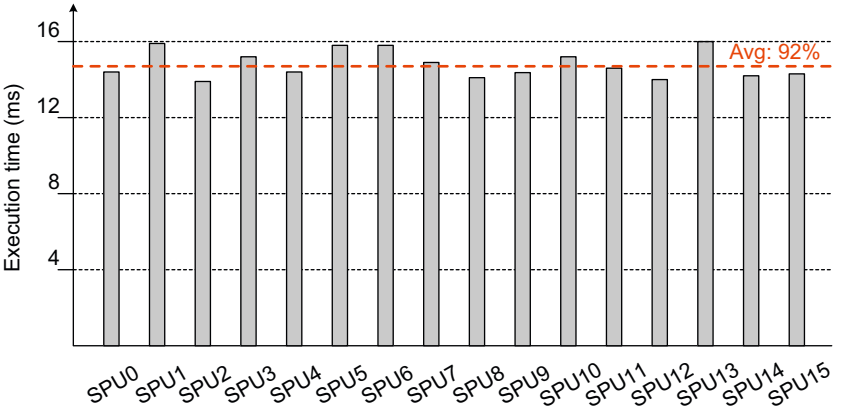


Fig. 7. Hardware utilization of SPU cores.

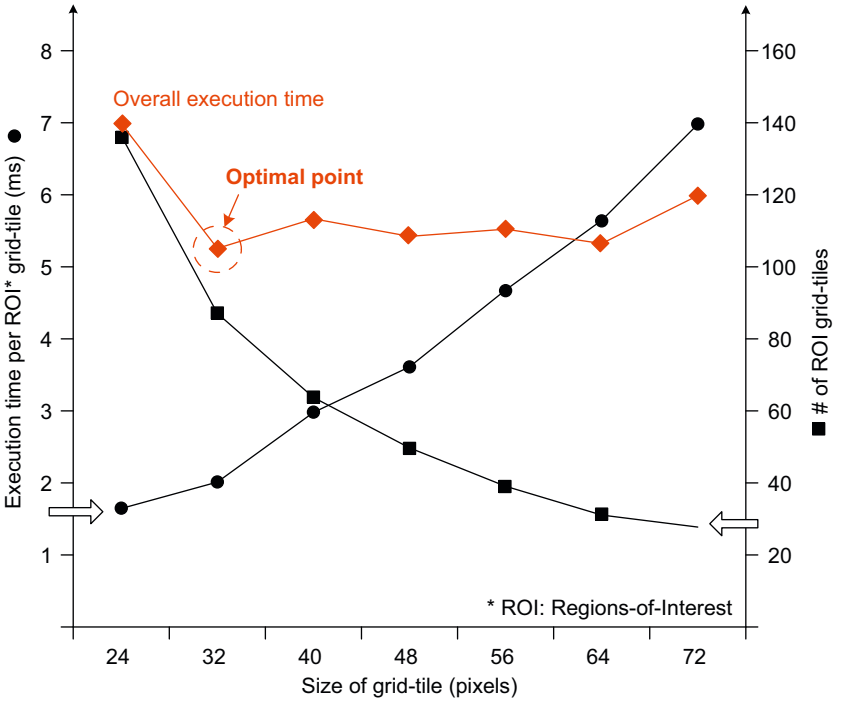


Fig. 8. Tile size vs. overall execution time.

Please cite this article as: J.-Y. Kim, et al., An attention controlled multi-core architecture for energy efficient object recognition, Signal Process. Image Commun. (2010), doi:10.1016/j.image.2010.03.003

grid-tiles decreases quadratically according to the size of grid-tile. By multiplying two factors, we can obtain overall execution time. The resulting performance is not optimal when the size of grid-tile is too small or too large, i.e., 24×24 and 72×72, however, the differences are not very large for the medium size cases from 32×32 to 64×64. The optimal point is when the tile size is 32×32. The cases of multiples of 16 such as 32×32, 48×48, and 64×64 show better performance due to their better row data mapping to SPU core. Lastly, it is remarkable that the experimental results can be affected by hardware structure and software mapping on it. In our case, we used an 8-way SIMD processor and streaming model in software mapping as described in Section 3.3. Data transaction style with external memory can be another factor in determining the optimal size of the grid-tile. Using cache or scratch pad, using DMA or not can affect the performance. In our case, we used a scratch pad with a DMA capable of 2-dimensional data access.

Fig. 9 shows the experimental results of the WATS showing its execution time management. We measured the execution times of the overall process of SPU cores when the ROI grid-tiles are scheduled to 16 SPUs by the WATS. The mean and variance value of the number of ROI grid-tiles for 100 sample images are measured as 79.33 and 22.87, respectively, where the size of input image is the VGA (640×480) and the size of a grid-tile is 40×40 pixels. As shown in the result graph, the WATS manages the overall execution time around 16.4 ms with a small variance of 2.62 under the large ROI workload variation. In this situation, the average number of operating SPU cores is measured as 13.16 and the rest SPU core resources can be used for other purposes or saved for low power consumption.

Fig. 10 shows the total energy reduction by the proposed architecture. We measured the energy consumption of conventional and attention controlled multi-core architecture under the condition that the hardware of multiple cores are the same as 16 SPU cores and the energy consumption of a controller unit is ignored in both the cases. Therefore, energy overhead of the proposed attention controlled architecture is energy dissipation by the VPE for attention processing. As a result, the attention controlled multi-core architecture reduces the total energy consumption by 2.1 times by reducing the energy consumption of energy hungry multiple cores with investing a 0.87 mJ attention processing.

4. Neuro-fuzzy visual perception implementation

In the proposed attention controlled architecture, the ROIs of an input image are used as an important clue for intelligent control of multiple cores to reduce the overall energy consumption. In this section, we describe the implementation of a special hardware named visual perception engine (VPE) that extracts the ROIs out of the input image prior to detailed object recognition. Since the visual perception is an additional process to obtain energy saving in object recognition processing, the implementa-

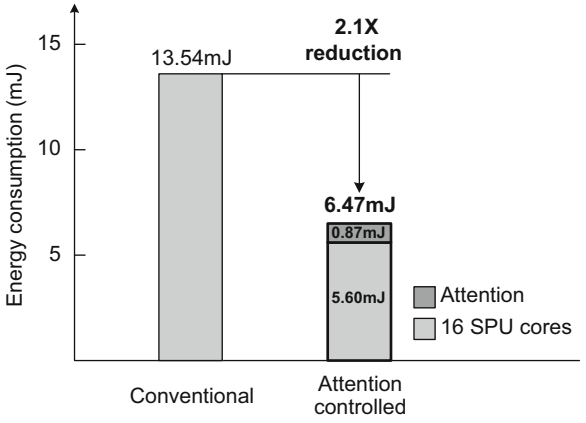
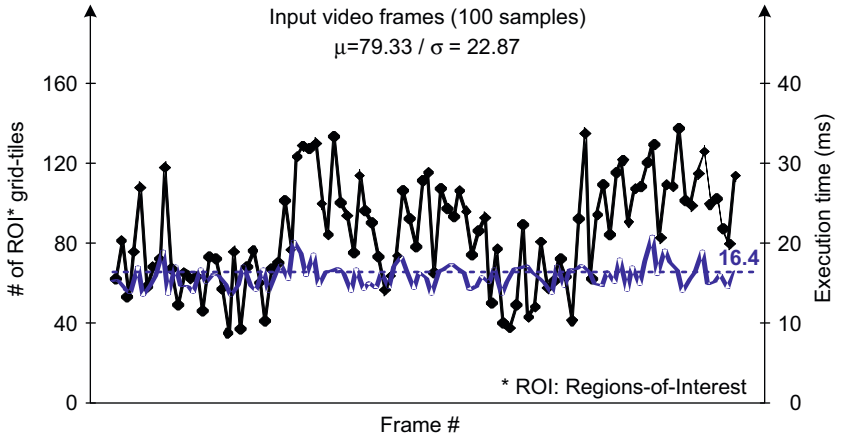


Fig. 10. Energy reduction effect.



Execution time	Variance	Resource use
16.4ms	2.62	13.16/16.00

Fig. 9. Execution time management by WATS.

Please cite this article as: J.-Y. Kim, et al., An attention controlled multi-core architecture for energy efficient object recognition, Signal Process. Image Commun. (2010), doi:10.1016/j.image.2010.03.003

tion cost of it in terms of energy dissipation should be small enough.

For energy efficient design, the VPE employs an RISC controller and 3 dedicated hardware blocks for the detailed visual perception process as shown in Fig. 11: a motion estimator (ME), a visual attention engine (VAE), and an object detection engine (ODE). The ME is employed to generate a motion vector map between two consecutive frames using a conventional block matching algorithm [12]. The VAE is employed to accelerate feature map extraction and normalization to generate saliency map. The ODE is proposed to perform final ROI classification from selected seed points that determines neighboring pixels can be joined to interest region of object or not. The RISC controller takes a role in control of the 3 dedicated hardware blocks and performing software-oriented operations. A 24 KB memory is used for storing original images and data communication among the 3 special blocks by sharing intermediate processing data. In the design of the VAE and ODE, bio-inspired 2-dimentional cellular neural networks (CNNs) and neuro-fuzzy classifier suitable for rapid feature extraction and robust classification, respectively, are employed.

4.1. Cellular neural networks based visual attention engine

Fig. 12 shows the block diagram of the VAE and its cell organizations [9]. The VAE is composed of 40×40 cell arrays and a linear array of 40 PEs. The cells perform storage and inter-cellular communication functions while the PEs are responsible for processing the cells data. Each VAE cell consists of a 6T SRAM based 4×8-bit register that stores intermediate data and result data of CNN operation, and 8-bit 4-directional shift register that shifts the register data to the adjacent cells. A shift operation on the entire chip requires only 1 cycle to complete. Based on its fully connected 2-dimentional cellular structure and fast data movement among neighbor cells, its regional and collective processing accelerates the various feature extractions that contains a lot of image filtering and convolution operations.

4.2. Object detection engine: neuro-fuzzy classifier

Fig. 13 shows the block diagram of the ODE and its transistor-level circuit implementations [11]. The ODE consists of Gaussian fuzzy membership and single-layer neural network, used for similarity measure between the seed and the target pixel and for final decision making in classification, respectively. An analog-digital mixed mode approach is used to exploit both advantages of analog and digital circuits. Firstly, data processing parts of the ODE are designed by analog circuits to obtain area and power reduction in non-linear Gaussian function and synaptic multiplier implementation. After converting 8-bit intensity, saliency, and location digital values of the target and seed pixel to analog signals, 3 analog Gaussian function circuits measure the similarities between the two pixels. Gaussian function is realized by the combination of MOS differential pair and minimum following circuit in current mode operation. The differential pair circuit generates two symmetric differential signals where each of them has exponential non-linearity characteristics, and the minimum follower circuit results Gaussian-like output by following the minimum between the two signals. Current mode neural synaptic multipliers are implemented by binary-weighted current mirrors to merge the 3 similarities with their weight values. A comparator circuit makes the final decision result 0 or 1 through thresholding operation. On the other hand, a learning part in order to update the

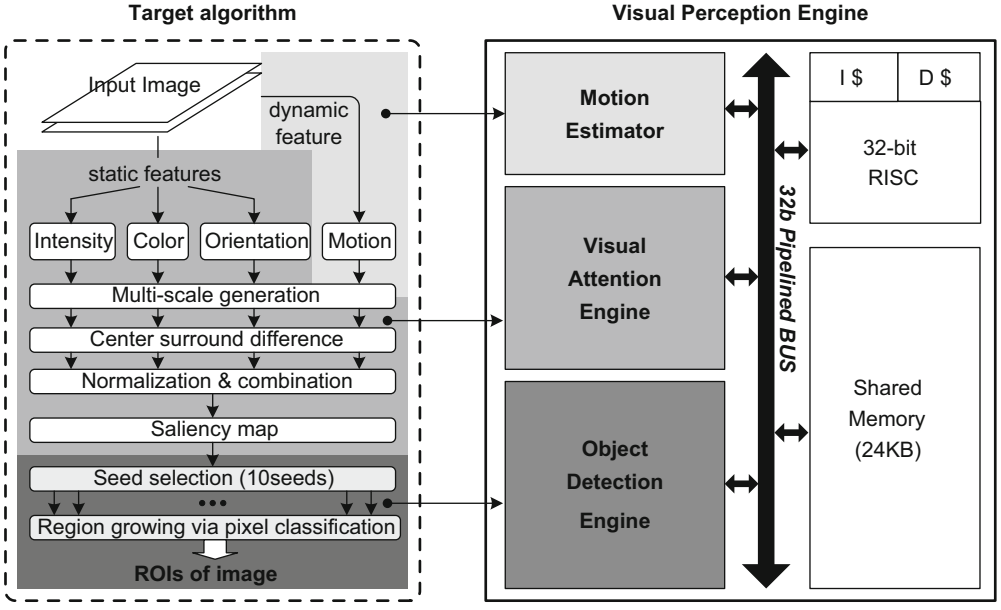


Fig. 11. Block diagram of visual perception engine.

Please cite this article as: J.-Y. Kim, et al., An attention controlled multi-core architecture for energy efficient object recognition, Signal Process. Image Commun. (2010), doi:10.1016/j.image.2010.03.003

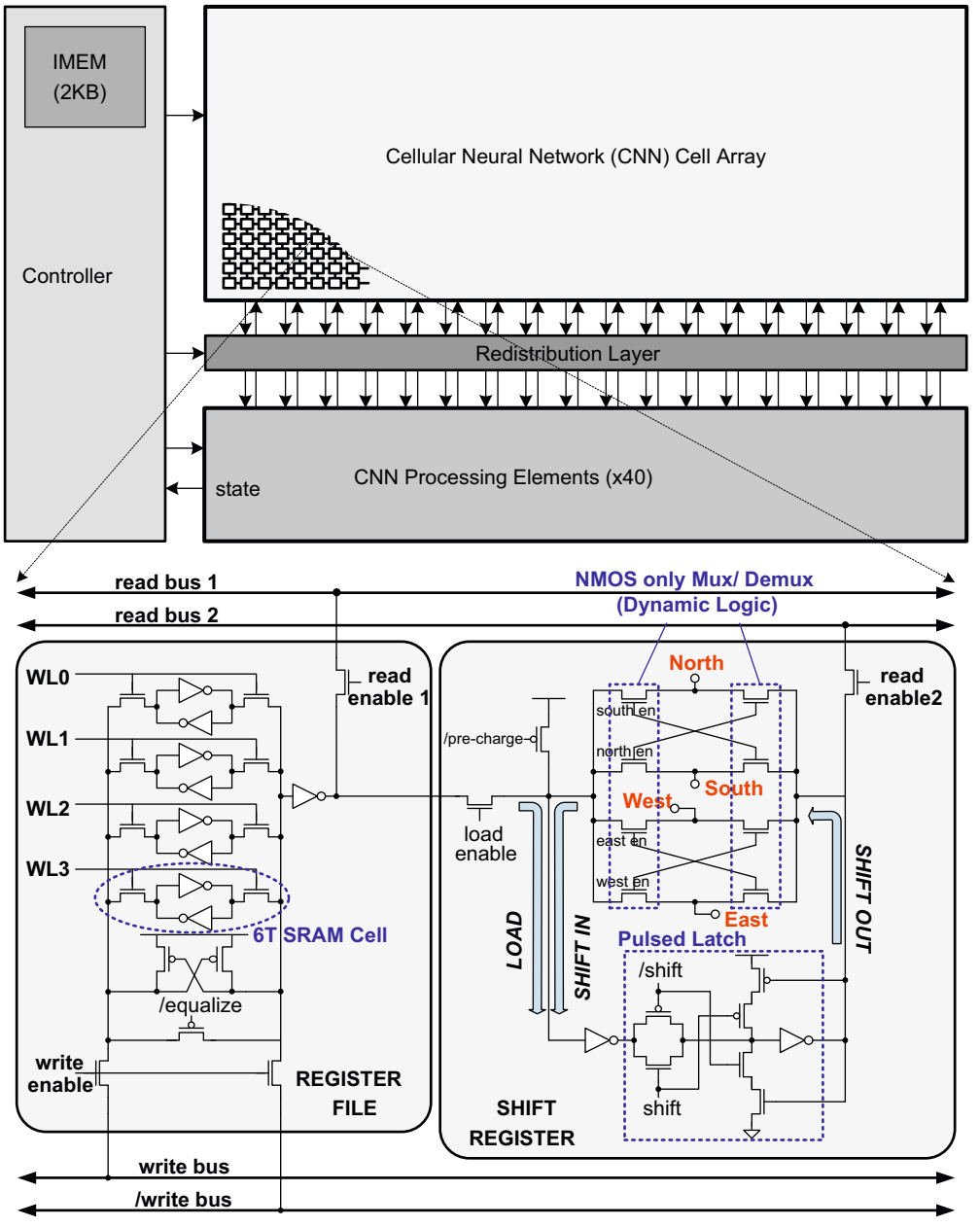


Fig. 12. Cellular neural network based visual attention engine.

weight values of neural network is implemented by digital circuits for easy management of data storage. With an unsupervised Hebbian learning that strengthens the connection weight to firing cells [13] only a multiplier and an adder are used for each weight value. As a result, analog-digital mixed mode implementation reduces the area and power consumption of the ODE by 59% and 44%, respectively, compared with those of digital-only implementations. With the 4 parallel execution units, the ODE completes the ROI detection for each region within 7 μs at 200 MHz operating frequency.

4.3. VPE evaluation

To evaluate the energy efficiency of the VPE, we compare the energy consumption of it with the other comparable processing unit when both of them run the same visual perception algorithm. We compare the VPE to an SPU core because its SIMD processing is also quite suitable for visual perception algorithm. To consider area factor, we compare the results of 2 SPU cores with the VPE. As shown in Table 2, the VPE's dedicated hardware blocks execute target algorithm more than 3.5 times faster than SPU core's SIMD acceleration. As a result, the

Please cite this article as: J.-Y. Kim, et al., An attention controlled multi-core architecture for energy efficient object recognition, Signal Process. Image Commun. (2010), doi:10.1016/j.image.2010.03.003

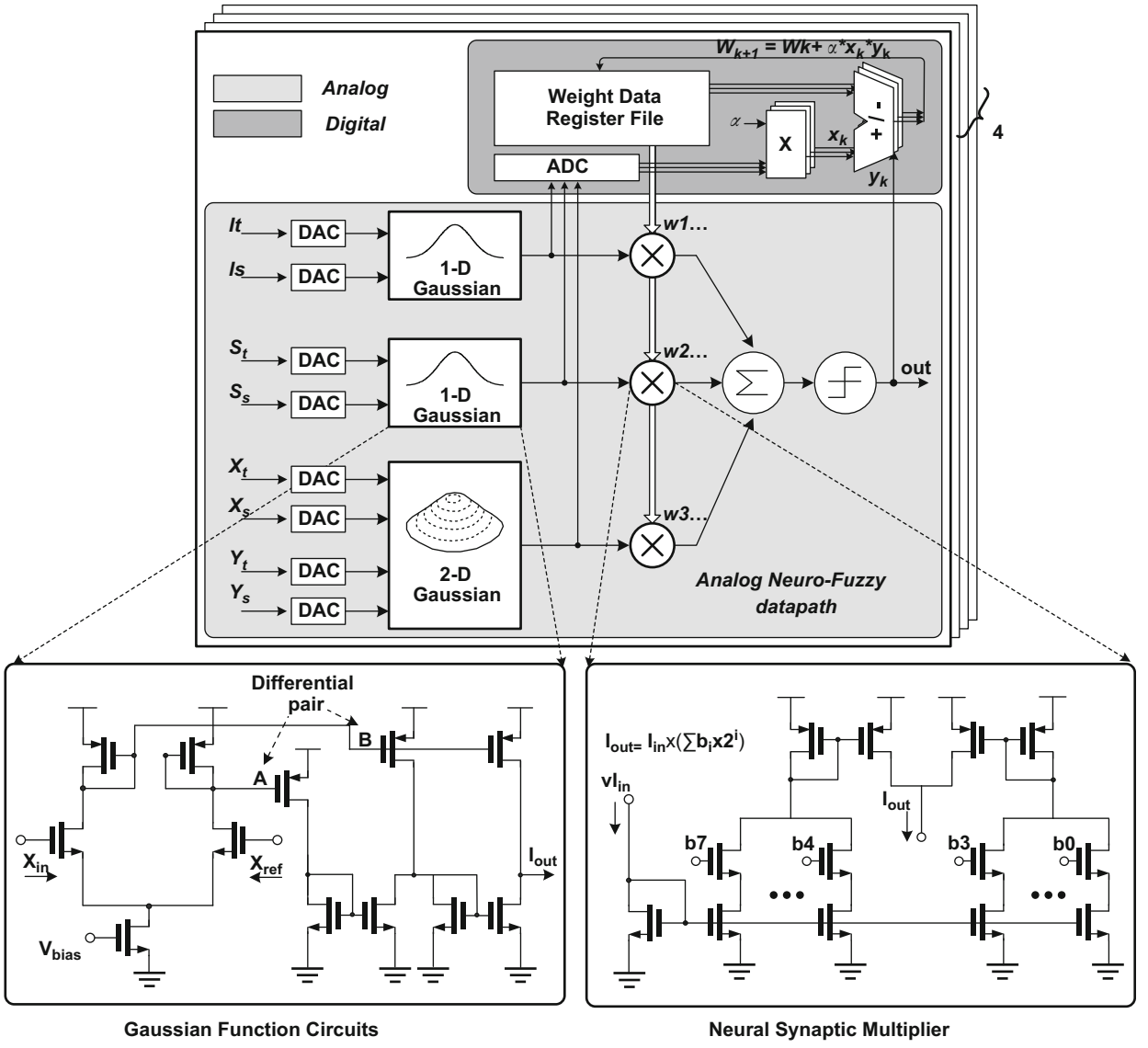


Fig. 13. Neuro-fuzzy object detection engine.

Table 2
VPE evaluation.

	Execution time	Power dissipation	Energy consumption	Energy gain
2 SPU cores	36.9 ms	67.5 mW	2.49 mJ	–
VPE	10.4 ms	83 mW	**0.87 mJ**	**2.85 ×**

VPE completes the target visual perception algorithm with 0.87 mJ energy dissipation and this is 2.85 × more energy efficient than 2 SPU cores.

5. System integration

The proposed attention controlled architecture is implemented in a real-time object recognition processor as shown in Fig. 14. Except the VPE, TM, and 16 SPU cores, a decision processor (DP) is added for post database matching task after key-point description by the 16 SPU cores. In the proposed processor, 3 recognition stages, visual perception, feature description, and database matching, operated by the VPE, 16 SPUs, and DP, respectively, are executed in the pipeline for high-recognition throughput. All 21 IP blocks including 2 external memory interfaces are fully interconnected

Please cite this article as: J.-Y. Kim, et al., An attention controlled multi-core architecture for energy efficient object recognition, Signal Process. Image Commun. (2010), doi:10.1016/j.image.2010.03.003

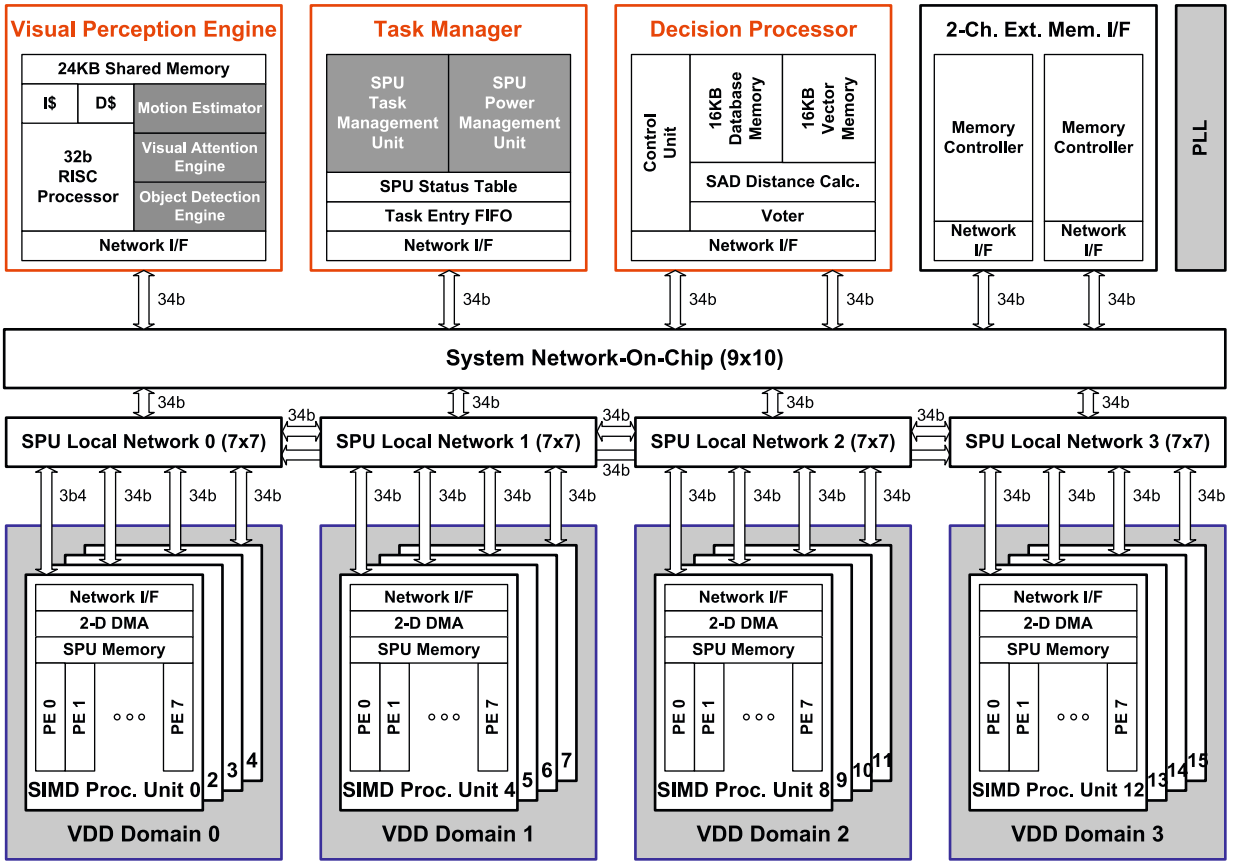


Fig. 14. System integration block diagram.

through a network-on-chip (NoC). The 16 SPU cores are divided into 4 SPU clusters where a cluster contains 4 SPU cores and has a separate power domain. Further analysis about the number of separate power domains and its power reduction effect is given in [14]. For power gating of each domain, external power gating is used. In the chip, the TM manages 4 power domains of 4 SPU clusters by requesting each of the external regulators to gate-off corresponding domain.

Fig. 15 shows the chip micrograph of the proposed processor fabricated in 0.13 μm 1-poly 8-metal CMOS technology. Its area is $7 \times 7\ mm^2$ and contains 36.4 M transistors including 3.7 M logic gates and 396 KB on chip SRAM. The operating frequency is 200 MHz for IP blocks and 400 MHz for NoC. Its peak performance amounts to 201.4 GOPS where 1 operation means 16-bit fixed-point operation. The average power consumption is 496 mW at the supply voltage of 1.2 V. Table 3 summarizes chip features.

Table 4 shows performance comparisons with previous object recognition processors [2–4,14]. While IMAP-CAR has a target application of vehicle vision, other processors share the same target application, SIFT based object recognition. In commercial CELL-BE processor [15] and multi-core object recognition processor [3], only SIFT feature description is implemented without post database matching process. On the other hand, full object recognition is implemented in dual-mode object recognition processor [4] and the proposed processor with the database size of 5632 and 16,384, respectively.

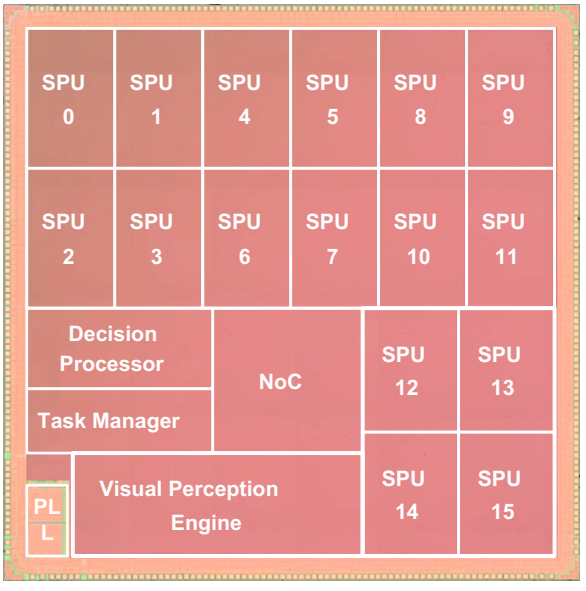


Fig. 15. Chip micrograph.

Please cite this article as: J.-Y. Kim, et al., An attention controlled multi-core architecture for energy efficient object recognition, Signal Process. Image Commun. (2010), doi:10.1016/j.image.2010.03.003

Table 3
Chip summary.

Technology	0.13 mm 1P 8 M CMOS	
Package	320 pin FPGA	
Die size	7 mm × 7 mm	
Power supply	1.2V core, 2.5V I/O	
Operating frequency	200 MHz IPs/400 MHz NoC	
Transistor counts	36.4 M transistors	
	3.73 M gates/396 KB SRAM	
Power consumption	Peak: 695 mW/ Average: 496 mW	
Peak performance	VPE	54 GOPS
	16 SPU cores	128 GOPS
	DP	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 frame/s	
Database size	16,384 vectors (50 objects)	

Table 4
Performance comparisons.

	IMAP-CAR [2]	CELL-BE [15]	CICC2007 [3]	ISSCC2008 [4]	This work
Process	130 nm	90 nm	180 nm	130 nm	130 nm
Chip size/ Transistor count	Not reported/ 26.8 M TRs	235 mm^2/241 M TRs	38.5 mm^2/0.8 M gates, 34 KB	36 mm^2/2M gates, 228 KB	49 mm^2/36 M TRs
Power supply	1.2 V	0.9–1.3 V	1.8 V	1.2 V	1.2 V
Operating frequency	100 MHz	3.2 GHz	200 MHz	200 MHz	200 MHz
Peak performance	100 GOPS	> 200 GFLOPS	81.6 GOPS	125 GOPS	201.4 GOPS
Power consumption	< 2 W	About 50 W	1.08 W (avg.)	0.6 W (avg.)	**0.5 W (avg.)**
Target application	Vehicle area detection	SIFT descriptor generation	SIFT descriptor generation	Object recognition (attention based)	Object recognition (attention based)
Hardware architecture	SIMD	Ring interconnected multi-core	Multi-core *w*/shared channel memories	Reconfigurable SIMD/ MIMD	**Attention controlled multi-core**
Frame rate	32 fps	0.48 fps	16 fps	22 fps	**60 fps**
Database size	N.A.	N.A.	N.A.	5632	16,384
Energy/frame	40 mJ	104 J	67.9 mJ	26.5 mJ	**8.2 mJ**
Screen resolution	256 × 240	UVGA (1600 × 1200)	QVGA (320 × 240)	QVGA (320 × 240)	VGA (640 × 480)
Energy/pixel	651 nJ	54 μJ	884 nJ	345 nJ	26.7 nJ

Since the attention controlled multi-core architecture reduces the overall workload of object recognition and manages multi-core resource efficiently, the proposed processor achieves 60 frame/s frame rate for the VGA-sized input video with 496 mW low power consumption. As a result, the obtained 8.2 mJ energy dissipation per frame is the lowest ever and 3.2 × less than the state-of-the-art object recognition processor.

6. Conclusions

In this article, an attention controlled multi-core architecture is presented for energy efficient object recognition. The VPE computes regions-of-interest (ROIs) out of the input image and the TM controls the multiple processing cores to perform local object recognition processing for the selected area. The TM performs dynamic scheduling of various ROI tasks from the VPE to multiple cores in a unit of small-sized grid-tile. In this architecture, the utilization of the multiple cores is measured as 92% and the 32 × 32 pixel size is the best for the grid-tile in terms of overall execution cycles. As a result, the proposed architecture achieves 2.1 × energy reduction in multi-core object recognition system with sacrificing small overhead for attention processing. Finally, the proposed architecture is realized in 0.13 μm CMOS technology and verifies 3.2 × lower energy dissipation per frame than the state-of-the-art object recognition processor.

Although this architecture is optimized for ROI based selective processing, it is also suitable for general image processing applications. With the tile based fine-grained processing, task management by the TM can achieve high utilization of multiple processing cores.

References

[1] A. Abbo, et al., XETAL-II: a 107 GOPS, 600 mW massively-parallel processor for video scene analysis, IEEE ISSCC Dig. Tech. Pap. (February 2007) 270–271.

Please cite this article as: J.-Y. Kim, et al., An attention controlled multi-core architecture for energy efficient object recognition, Signal Process. Image Commun. (2010), doi:10.1016/j.image.2010.03.003

[2] Shorin Kyo, et al., A 100 GOPS in-vehicle vision processor for pre-crash safety systems based on ring connected 128 4-way VLIW processing elements, IEEE Symp. VLSI Circuits (2008) 28–29.
[3] D. Kim, et al., An 81.6 GOPS object recognition processor based on NoC and visual image processing memory, IEEE CICC (2007) 443–446.
[4] K. Kim, et al., A 125GOPS 583 mW network-on-chip based parallel processor with bio-inspired visual attention engine, Dig. Tech. Pap. IEEE ISSCC (2008) 308–309.
[5] J.-Y. Kim, et al., A 201.4GOPS 496 mW real-time multi-object recognition processor with bio-inspired neural perception engine, Dig. Tech. Pap. IEEE ISSCC (2009) 150–151.
[6] C. Privitera, L. Stark, Algorithms for defining visual regions-of-interest: comparison with eye fixations, IEEE Trans. Pattern Anal. Mach. Intell. 22 (9) (2000) 970–981.
[7] David G. Lowe, Distinctive image features from scale-invariant keypoints, ACM Int. J. Comput. Vision 60 (2) (2004) 91–110.
[8] Q. Zhang, et al., SIFT implementation and optimization for multi-core systems, IEEE International Symposium on Parallel and Distributed Processing, IPDPS 2008.
[9] S. Lee, et al., The brain mimicking visual attention engine: an 80×60 digital cellular neural network for rapid global feature extraction, IEEE Symp. VLSI Circuits (2008) 26–27.
[10] L. Itti, et al., A model of saliency-based visual attention for rapid scene analysis, IEEE Trans. Pattern Anal Mach. Intell. 20 (11) (1998).
[11] M. Kim, et al., A 22.8GOPS 2.83 mW neuro-fuzzy object detection engine for fast multi-object recognition, IEEE Symp. VLSI Circuits (2009) 260–261.
[12] P. Pirsch, N. Demassieux, W. Gehrke, VLSI architectures for video compression—a survey, Proc. IEEE 83 (2) (1995) 220–246.
[13] D.O. Hebb, in: The Organization of Behavior, John Wiley & Sons, 1949.
[14] J.-Y. Kim, et al., A 60 fps 496 mW multi-object recognition processor with workload-aware dynamic power management, ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED), pp. 365–370, 2009.
[15] B. Kwon, et al., Parallelization of the scale-invariant keypoint detection algorithm for cell broadband engine architecture, IEEE Consum. Commun. Networking Conf. (CCNC) (2008) 1030–1034.

Please cite this article as: J.-Y. Kim, et al., An attention controlled multi-core architecture for energy efficient object recognition, Signal Process. Image Commun. (2010), doi:10.1016/j.image.2010.03.003