Distributed Computing in IoT: System-on-a-Chip for Smart Cameras as an Example

Shao-Yi Chien[1], Wei-Kai Chan[1], Yu-Hsiang Tseng[1], Chia-Han Lee[2]
V. Srinivasa Somayazulu[3], Yen-Kuang Chen[3]
[1] Graduate Institute of Electronics Engineering and Department of Electrical Engineering
National Taiwan University, Taipei, Taiwan
[2] Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan
[3] Intel Corporation, USA
Intel-NTU Connected Context Computing Center, Taipei, Taiwan

Abstract— There are four major components in application systems with internet-of-things (IoT): sensors, communications, computation and service, where large amount of data are acquired for ultra-big data analysis to discover the context information and knowledge behind signals. To support such large-scale data size and computation tasks, it is not feasible to employ centralized solutions on cloud servers. Thanks for the advances of silicon technology, the cost of computation become lower, and it is possible to distribute computation on every node in IoT. In this paper, we take video sensing network as an example to show the idea of distributed computing in IoT. Existing related works are reviewed and the architecture of a system-on-a-chip solution for distributed smart cameras is proposed with coarse-grained reconfigurable image stream processing architecture. It can accelerate various computer vision algorithms for distributed smart cameras in IoT.

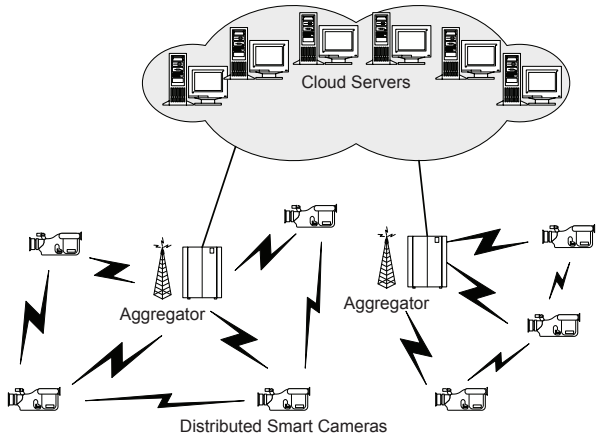

Fig. 1. Distributed smart camera as a node of an IoT.

I. Introduction

In an Internet-of-Things (IoT) or Machine-to-Machine (M2M) network, billions of smart objects, such as sensors, actuators, smart phones and smart vehicles, are connected with each other for sensing physical signals to infer context information and provide better service for human in real-time or proactively with or without users' intervention [1–3]. IoT is viewed as the next wave of information technology. It will change the way we live, play, and work.

There are four major components in IoT, including sensors, communications, computation, and service [3]. In this paper, our focus is on the context inferring process, where the first three components are involved. It is the key first step in IoT for local or cloud servers to aware the condition of the physical world.

Sensors are connected to the servers via gateways or aggregators to stream sensed data in real-time, which generates ultra-big data for further data analytics. It is infeasible and inefficient to handle all the data with cloud servers because of the limitation of computing and communication resources. To deal with such ultra-big data, distributed computing, also called as ubiquitous computing and ambient intelligence, is a more efficient approach. Rather than the centralized approach, where all the data analysis works are executed on cloud servers, with the advances of semiconductor technology, it is possible to embed computation on sensors and aggregators. System-on-a-Chip (SoC) can also play an important role in IoT that small chips are employed at the end devices for helping big data analytics at cloud servers.

In this paper, distributed video sensor network is taken as an example to elaborate the concept of distributed computing. We will start from two case studies to show the impact of distributed computing with smart cameras in Section II. Next, the second part of this paper focuses on the SoC design for smart cameras. Several previous works are reviewed in Section III, and a possible solution is then described in Section IV. Finally, we conclude this paper in Section V.

II. Distributed Computing in Video Sensor Network

Fig. 1 shows the block diagram of a video sensor network. The end devices of this network are distributed cameras. Thanks for the advances of semiconductor technology, more and more computation can be embedded into the cameras to make them become smart [4–6]. Several distributed smart cameras are connected locally to a aggregator or gateway. Next, the aggregators connected to the cloud servers. Note that it

978-1-4799-7792-5/15/$31.00 ©2015 IEEE

is possible to have more layers in the network hierarchy, and the three-layer structure is adopted in this paper for simplification. In the next two subsections, two cases are studied to discuss the performance differences between centralized and distributed approaches.

A. Video Surveillance

The first case study is a video surveillance network with 19 cameras, which is the BL-7F test bench in [7]. The setup of those 19 cameras is depicted in Fig. 2(a). Among the 19 cameras, 10 cameras are employed to monitor the hall way, and 9 cameras are employed to monitor an office. The captured videos from those 19 cameras are also shown in Fig. 2(b). All the cameras are connected to the server directly, which is a two-layer structure.

In the video surveillance application, the target is to store the video sequences when critical event occurs. Two different approaches for video data collection are conducted. The first one is the centralized approach, where all the video data are streamed back to the server. The second one is the distributed approach proposed in [7]. It is composed of an intra processing stage for a single camera and an inter processing stage between different cameras. In the intra processing stage, each camera maintains a background model respectively to identify the important input frames, which are the frames that cannot be well represented by the background model. In the inter processing stage, cameras exchange the feature information of those important input frames between each other, and the redundant frames are dropped to save the transmission bandwidth.

The transmission bandwidth analysis result is shown in Fig. 2(c). It shows that with the distributed approach, about 91.3% of the transmission bandwidth can be saved. Without the assistant from the server, the distributed computation on the sensors can reduce the bandwidth without missing video information with important event.

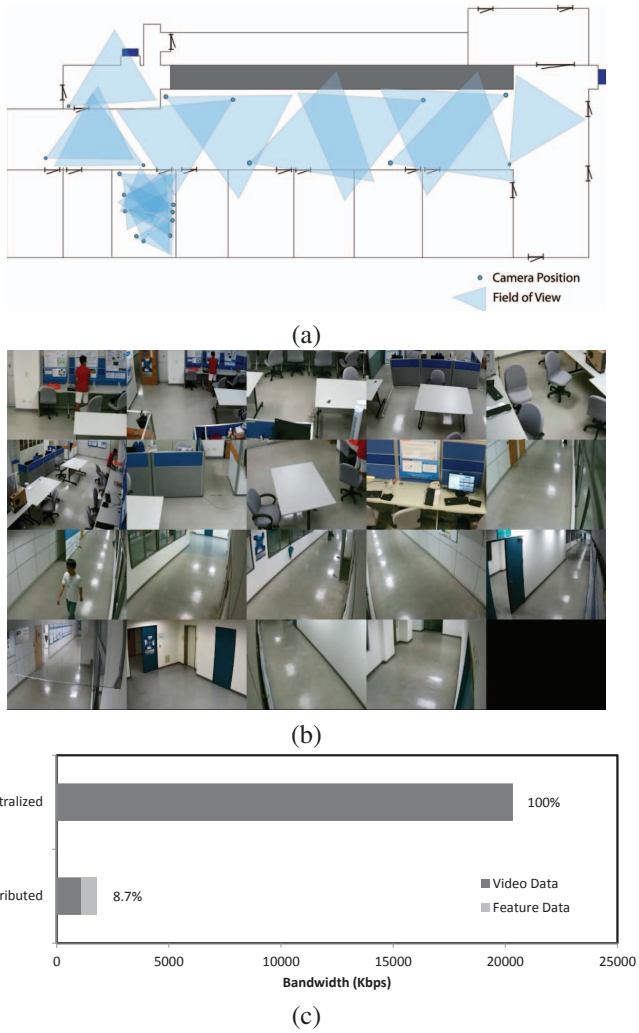

Fig. 2. (a) The floor plan and the locations of the surveillance cameras of the BL-7F dataset. (b) The captured videos from these 19 surveillance cameras. (c) The required bandwidth for video transmission with the centralized and distributed approaches.

B. Vehicle Localization

The second case study is a vehicle neighboring map generation system with video cameras in an intelligent transportation system [8]. In this system, with the sensed video data from both the roadside units (RSUs) and the vehicular on-board units (OBUs), the neighboring map describing the location information of surrounding vehicles is generated and provided to the driver assistance system of each vehicle. It can also be employed for other applications for smart cars, such as energy efficiency improvement and navigation. Experiments are conducted on a large-scale simulation system. The setting is illustrated in Fig. 3, where 40 vehicular on-board units (OBUs), 128 roadside units (RSUs), and 16 aggregators are involved in the three-layer network structure. The locations and directions of these OBUs are generated with random numbers, and each aggregator collects the video information of the RSUs and OBUs in its covering range.

Different kinds of sensors and aggregators with different computing ability are considered for the discussion of transmission bandwidth. As shown in Fig. 4, different configurations form six different test cases. The low-end sensor is the

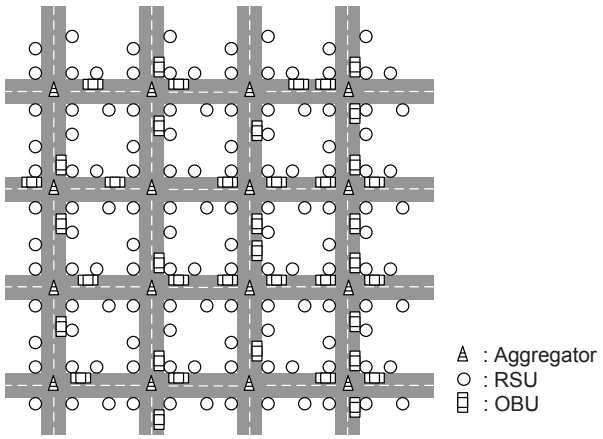

Fig. 3. Setting of the simulated vehicle neighboring map generation system with video cameras in an intelligent transportation system.

Sensor \ Aggregator	Low	Mid	High
Low	Case I	Case II	Case III
Mid		Case IV	Case V
High			Case VI

Fig. 4. Simulation cases.

video sensor with simple video capturing, coding, and transmission ability. The middle-level sensor is the video sensor with moving object detection ability. It can transmit video data to the aggregator only when it detects moving objects. The high-end sensor is the video sensor with a vehicle detection subsystem. It can transmit vehicle information instead of video data to the aggregator. On the other hand, the low-end aggregator acts like an access point that can only forward the received video data to the cloud severs. The middle-level aggregator has the ability of moving object detection, and can also employ local video summarization technique to remove redundant information from the received video data before transmitting to the cloud servers. The high-end aggregator can perform vehicle detection and generate the local neighboring maps.

The simulation result is shown in Fig. 5, where the total bandwidth of all sensors to its corresponding aggregators and all aggregators to the cloud severs are shown for different cases. It shows that with devices quipped with different computing ability, the required bandwidth could be quite different. For example, Cases I, II, and III employ low-end sensors, and hence the bandwidth from sensors to aggregators are the same in call cases; however, the bandwidth from the aggregators to the cloud server is drastically dropped when the computing power of the aggregators become higher. Similar result can also be found for Cases IV and V. On the other hand, when we compare Cases III, V, and VI, the bandwidth from aggregators to the cloud servers are the same since all of them employ high-end aggregators; however, the bandwidth from the sensors to the cloud server is also drastically dropped when the computing power of the sensors become higher. We also roughly estimate the hardware device cost, and the cost comparison result is Case III $>$ Case V $>$ Case VI $>$ Case II $>$ Case IV $>$ Case I. Note that the cost of cloud computing, communication, and deployment are not included here. With overall consideration, Cases IV and VI seem better solutions.

From these case studies, it can be observed that distributed computing can not only off-load the computation from the cloud server but also save the transmission bandwidth. The optimal way for computation distribution still requires further study, and the embedded video analysis and object recognition engines are essential for video sensors and aggregators in IoT. The related works are reviewed in the next section.

III. Review of Existing Related Works

Many vision processors for video analysis and object recognition are proposed in literatures [9–12]. In order to achieve high throughput, most of them are based on single-instruction-multiple-data (SIMD) architecture. A simplified illustration is

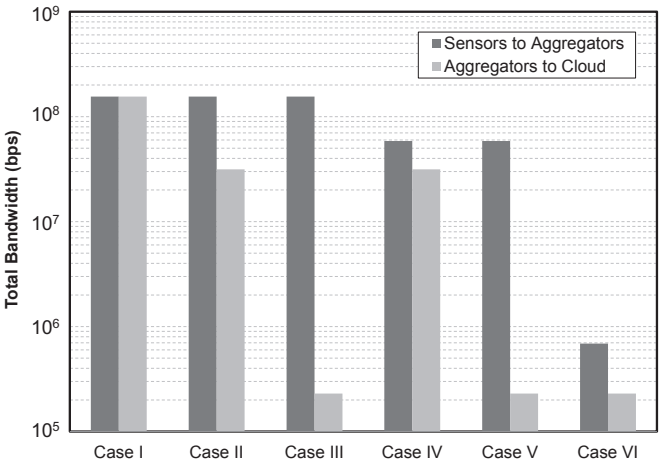

Fig. 5. Simulation results of the simulated vehicle neighboring map generation system.

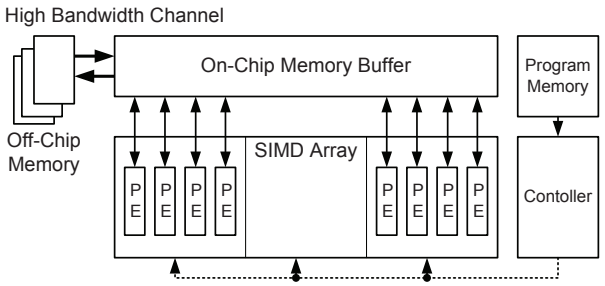

Fig. 6. Architectural concept of conventional SIMD array.

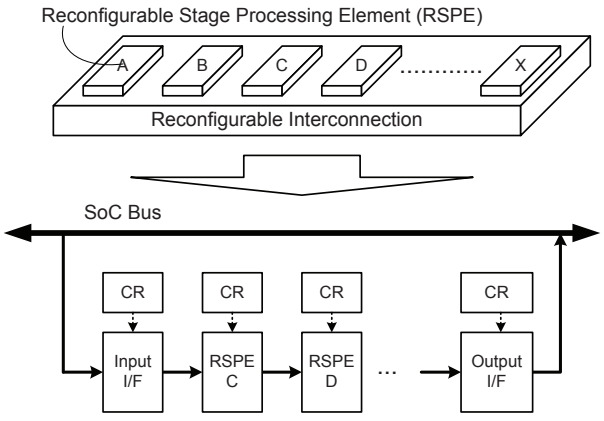

Fig. 7. Architectural concept of CRISP.

shown in Fig. 6. The high processing capability is provided by the SIMD array composed of a large amount of processing elements (PEs). To make the PE array operate with high utilization, a large multi-bank on-chip memory is usually required as well as a high bandwidth channel to the off-chip memory, which often leads to high hardware cost and high power consumption. Furthermore, some of them are even designed to work with an on-chip frame memory to achieve higher processing capability; however, because of the high cost of the on-chip frame memory, these works can only deal with low frame resolutions [10, 11].

Recently, machine learning engine with feature descriptors such as scale-invariant feature transform (SIFT) [13] is widely adopted for object recognition processors. Su *et al.* proposes an ASIC based chip for SIFT and visual vocabulary for wearable vision applications [14], and Oh *et al.* proposes an SIMD based multi-core architecture with a reconfigurable machine learning engine for SIFT based object recognition [15].

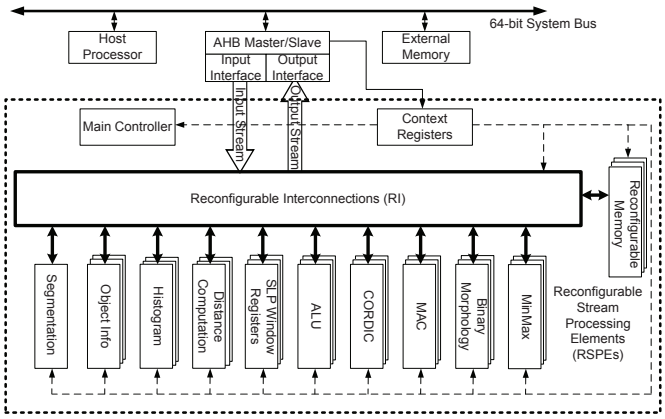

Fig. 8. Block diagram of the proposed reconfigurable smart-camera stream processor.

IV. System-on-a-Chip for Smart Cameras

A. Reconfigurable Smart-camera Stream Processor

Programmability, power efficiency, and cost efficiency are important design issues for smart camera SoC in IoT applications. Both ASIC and processor based solutions cannot address all these issues. On the other hand, reconfigurable architecture, where the function of the chip can be changed after fabrication and the data flow inside the chip can be very similar to ASIC, can often provide a balance between efficiency and programmability. Among different reconfigurable architecture, coarse-grained reconfigurable image stream processor (CRISP) architecture, which is an application-specific architecture, has been proven to be efficient for image processing applications [16].

The concept of CRISP is shown in Fig. 7. It is composed of two major components: reconfigurable stage processing element (RSPE) and reconfigurable interconnection (RI). The coarse-grained and heterogeneous RSPEs act as the reconfigurable fabric of this architecture. Each kind of RSPEs is designed for a specific class of operation for the target applications, and the minimum stream element is a pixel. For example, Multiply-and-Accumulate (MAC) RSPE, which is composed of a multiplier array and an adder tree, can deal with 2-D image filtering operations as well as matrix operations. The RI is the interconnection unit in this architecture. As Fig. 7 shows, to implement an algorithm with CRISP, the target algorithm is separated into several stages, and each stage is then mapped to the most suitable RSPE. By programming the RSPE with context registers (CRs) as well as the interconnection in RI, the input data is processed by those RSPE in a specific order in pipeline. The pipelined architecture makes the loaded data to be reused inside the chip, and the configurations are controlled by the CRs directly without the overhead of instruction fetching and decoding in the processor based architecture.

Based on this architecture, a reconfigurable smart-camera stream processor (ReSSP) is proposed, as shown in Fig. 8. ReSSP acts as a co-processor in an SoC. The 11 different RSPEs are designed to support smart camera applications, and they can be further classified into eight types as shown in Table I. Reconfigurable Memory and SLP Window Registers RSPEs are Type A. MinMax RSPE is Type B. MAC RSPE is Type C. Binary Morphology RSPE is Type D. CORDIC RSPE is Type E. ALU RSPE is Type F. Histogram is Type G. Distance Computation, Object Info, and Segmentation RSPEs are Type H. Different from conventional CRISP, in ReSSP, heterogeneous stream processing (HSP) and subword-level parallelism (SLP) architectures are proposed to support various data types in computer vision algorithms. For example, the stream elements in this system could be represented in 24-bit for color pixels, 8-bit for gray pixels, and 1-bit for object masks. The Reconfigurable Memory RSPE can support any memory usage in ReSSP with any data type and any levels of SLP. It is also be applied as a interface between heterogeneous data streams. Several RSPEs are designed to support SLP to take fully advantage of the wide system bus. Take Type D Morphology operation RSPE an example. The detailed architecture of the Binary Morphology RSPE is shown in Fig. 9. With the input bus in 64-bit, a 64-bit data is loaded into the RSPE at each cycle. It is viewed as 64 1-bit input data, and those 64 1-bit data can be processed by this RSPE at each cycle. This RSPE can support various morphology operations, which are frequently used in computer vision applications.

TABLE I
Types of RSPEs.

Type A	Data access
Type B	Sorting or minimum-maximum finding
Type C	Multiply-and-accumulate-based (MAC-based) kernel operations
Type D	Morphology operations
Type E	Fundamental mathematics function
Type F	Arithmetic and logical operations
Type G	Statistics accumulation
Type H	Algorithm specific or functional specific

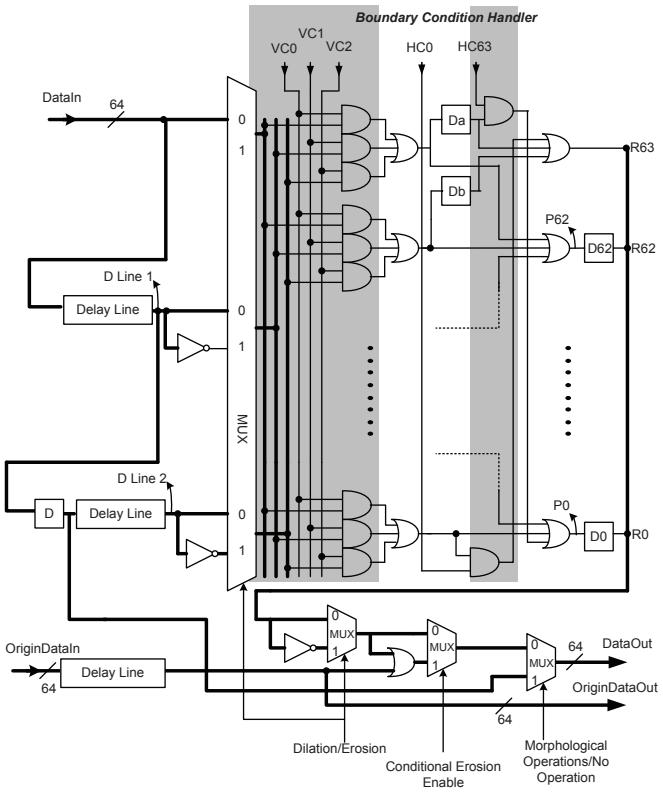

Fig. 9. Detailed architecture of the Binary Morphology RSPE.

B. Implementation Results and Comparisons

A prototype ReSSP chip is fabricated with TSMC 90nm technology. The die photo is shown in Fig. 10. Table II shows the chip specifications and comparison to related works [10–12,14,15]. Compared with [10,11], where the frame buffer is embedded into the chip, and the SIMD processor based architectures [12, 15], where large amount of on-chip memory is required, the proposed architecture outperforms previous works with smaller memory size, higher power efficiency and area efficiency, which is caused by adopting CRISP architecture. When compared with the ASIC based approach [14], the proposed architecture is still competitive in power and area efficiency.

Table III shows the performance of ReSSP for several high-level applications of smart cameras. It can execute video object segmentation and tracking for VGA-sized video in real-time. It can also support SIFT for Full-HD videos.

V. Conclusion and Future Works

Distributed computing is an essential technique for internet of things (IoT) to off-load the computation from the cloud servers as well as reduce the transmission bandwidth requirement. It is especially important for ultra-big data analysis problem in video sensor network. The two case studies show that the required bandwidth from the sensors to the aggregators highly depends on the computing power of sensors while the required bandwidth from the aggregators to the cloud servers highly depends on the computing power of aggregators. The optimization of computation distribution in IoT is a potential research topic.

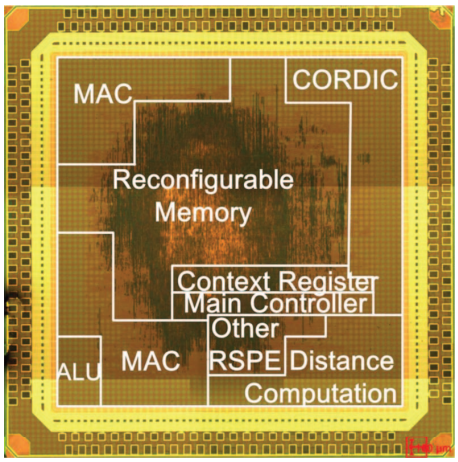

Fig. 10. Die photo.

Thanks for the advances of silicon technology, more computation can be embeded into sensors and aggregators. A reconfigurable smart-camera stream processor is proposed based on coarse-grained reconfigurable image stream processor architecture (CRISP) as well as heterogeneous stream processing (HSP) and subword-level parallelism (SLP) architectures. Experimental results show that the proposed design can achieve high area and power efficiency. In the future, we will continue designing an SoC for distributed video sensors for IoT applications. The whole network configuration will be also taken into consideration to achieve global optimization.

Acknowledgments

This work was supported by National Science Council, National Taiwan University and Intel Corporation under Grants NSC102-2911-I-002-001 and NTU103R7501.

References

[1] ITU Internet reports 2005: The Internet of Things. [Online]. Available: http://www.itu.int/internetofthings/

[2] G. Lawton, "Machine-to-machine technology gears up for growth," *IEEE Computer*, vol. 37, no. 9, pp. 12–15, Sep. 2004.

[3] Y.-K. Chen, "Challenges and opportunities of internet of things," in *Proc. 17th Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan. 2012, pp. 383–388.

[4] B. Rinner and W. Wolf, "An introduction to distributed smart cameras," *Proc. IEEE*, vol. 96, no. 10, pp. 1565 – 1575, Oct. 2008.

[5] G. Foresti, C. Micheloni, L. Snidaro, P. Remagnino, and T. Ellis, "Active video-based surveillance systems: the low-level image and video processing techniques needed for implementation," *IEEE Signal Processing Mag.*, vol. 22, no. 2, pp. 25–37, Mar. 2005.

[6] W.-K. Chan, J.-Y. Chang, T.-W. Chen, Y.-H. Tseng, and S.-Y. Chien, "Efficient content analysis engine for visual surveillance network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 5, pp. 693–703, May 2009.

TABLE II
CHIP SPECIFICATIONS AND COMPARISONS

Work	[10]	[11]	[12]	[14]	[15]	ReSSP
Process	UMC 0.18um 2P4M CMOS Image Sensor Process	TSMC 90nm IP9M CMOS	0.13μm 8 metal CMOS	TSMC 65nm	0.13μm 1P6M CMOS	TSMC 90nm 1P9M CMOS
Die Size	70.5 mm^2 (core area)	28mm^2 (die size)	50mm^2 (die size)	6.5mm^2 (die size)	32mm^2 (die size)	10.4mm^2 (3.2mmx3.2mm)
Power Supply	2.1V	Core 1.2V, I/O 2.5V	0.65V-1.2V	1.0V	1.2V	Core 1.2V, I/O 2.5V
Total Gate Count	N/A	3.00M gates	2.92M gates	0.91M	2.4M	0.9M Gates (2-Input NAND Gate, Including On-Chip Memory)
On-Chip Memory (Kb)	1,024	1,192	4,896	320	3,056	**56 (Including Context Registers)**
Working Frequency	50MHz	Max 200MHz	NoC: 400MHz Processing: Max 200MHz	200MHz	200MHz	Max 149MHz
Peak Performance (GOPS)	76.8	814	228	164.95GOPS	342	1157.82
Power Consumption	Vision Processor: 374mW	1214mW (Peak)	Peak :704mW Average: 345mW	Peak: 198.4mW Average: **52mW**	534mW	197mW (peak)
Area Efficiency (GOPS/mm^2)	4.357	29.100	9.120	25.377	10.688	**111.329**
Power Efficiency (TOPS/W)	1.283	0.671	0.469	1.18	0.640	**5.877**
Resolution and Spec for Image Analysis	Target at 128x128 image analysis	Target at 160x120 Image analysis	SIFT 640x480 30 fps and object recognition	Full HD 160-degree object viewpoint recognition	Object recognition on HD 720p video Streams	SIFT Full-HD Video Object Segmentation and Tracking 640x480 30fps and other applications with high spec

TABLE III
PERFORMANCE OF THE PROPOSED PROCESSOR.

Smart Camera Application	Supported High Level Algorithm	Specification
Video Object Segmentation and Tracking	Segmentation : Multi-Layer Background Subtraction Tracking : Particle Filter	Segmentation : 640x480 125fps Segmentation + tracking : 640x480 30fps 11 Objects (or 33000 particles per sec) with object size 80x80
Face Detection , Scoring and Ranking	Face Detection with Segmentation and Feature-based Face Scoring	150 faces per second
Object Detection and Recognition	Scale Invariant Feature Transform (SIFT)	can support 1920x1080 full HD object recognition in real-time

[7] S. Ou, C.-H. Lee, V. S. Somayazulu, Y.-K. Chen, and S.-Y. Chien, "On-line multi-view video summarization for wireless video sensor network," *IEEE J. Sel. Topics Signal Process.*, to appear.

[8] K.-W. Chen, H.-M. Tsai, C.-H. Hsieh, S.-D. Lin, C.-C. Wang, S.-W. Yang, S.-Y. Chien, C.-H. Lee, Y.-C. Su, C.-T. Chou, Y.-J. Lee, H.-K. Pao, R.-S. Guo, C.-J. Chen, M.-H. Yang, B.-Y. Chen, and Y.-P. Hung, "Connected vehicle safety science, system, and framework," in *Proc. 2014 IEEE World Forum on Internet of Things (WF-IoT)*, Mar. 2014, pp. 235–240.

[9] A. Abbo and et al., "Xetal-II: A 107GOPS, 600mW massively-parallel processor for video scene analysis," in *Dig. Tech. Papers IEEE International Solid-State Circuits Conference*, 2007, pp. 270–271.

[10] C.-C. Cheng, C.-H. Lin, C.-T. Li, S. Chang, C.-J. Hsu, and L.-G. Chen, "iVisual: An intelligent visual sensor SoC with 2790fps CMOS image sensor and 205GOPS/W vision processor," in *Dig. Tech. Papers IEEE International Solid-State Circuits Conference*, 2008, pp. 306 – 615.

[11] T.-W. Chen, Y.-L. Chen, T.-Y. Cheng, C.-S. Tang, P.-K. Tsung, T.-D. Chuang, L.-G. Chen, and S.-Y. Chien, "A multimedia semantic analysis SoC (SASoC) with machine-learning engine," in *Dig. Tech. Papers IEEE International Solid-State Circuits Conference*, 2010, pp. 338 – 339.

[12] S. Lee, J. Oh, M. Kim, J. Park, J. Kwon, and H.-J. Yoo, "A 345mW heterogeneous many-core processor with an intelligent inference engine for robust object recognition," in *Dig. Tech. Papers IEEE International Solid-State Circuits Conference*, 2010, pp. 332 – 333.

[13] D. G. Lowe, "Distinctive image features form scale-invariant keypoints," *Int. J. of Computer Vision*, 2004.

[14] Y.-C. Su, K.-Y. Huang, T.-W. Chen, Y.-M. Tsai, S.-Y. Chien, and L.-G. Chen, "A 52mW Full HD 80-degree viewpoint recognition SoC with visual vocabulary processor for wearable vision applications," *IEEE J. Solid-State Circuits*, vol. 47, no. 4, pp. 797–809, Apr. 2012.

[15] J. Oh, G. Kim, J. Park, I. Hong, S. Lee, J.-Y. Kim, J.-H. Woo, and H.-J. Yoo, "A 320 mW 342 GOPS real-time dynamic object recognition processor for HD 720p video streams," *IEEE J. Solid-State Circuits*, vol. 48, no. 1, pp. 33–45, Jan 2013.

[16] J. C. Chen and S.-Y. Chien, "CRISP: Coarse-grained reconfigurable image stream processor for digital still cameras and camcorders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 9, pp. 1223–1236, Sep. 2008.