# An Architecture Study of Scalable Optical-Flow Processor for Real-Time Video Segmentation

Ryo YAMAMOTO, Yuki FUKUYAMA, Tadayoshi KATAGIRI †, Junichi MIYAKOSHI,
Yuki KURODA †, Noriyuki MINEGISHI †, Masayuki MIYAMA †, Hiroshi KAWAGUCHI,
Kousuke IMAMURA †, Hideo HASHIMOTO †, and Masahiko YOSHIMOTO
Kobe University and † Kanazawa University, mailto:ryo@cs28.cs.kobe-u.ac.jp

## 1. Abstract

This paper proposes optical-flow processor architecture for real-time video segmentation. The architecture embodies the hierarchical optical-flow estimation (HOE) algorithm, and achieves scalability in terms of pixel rate and accuracy. In order to reduce a hardware resource cost of the processor, we introduce a common element (CE) that carries out all calculations necessary to optical-flow derivation. For area efficiency, a 2-port DRAM compatible with a logic process is adopted. When one CE is implemented on a chip in a 90-nm process technology, the processor performs 34 GOPS at a clock frequency of 189 MHz, and can handle a CIF-30 image sequence. The core size and power are estimated at 6.02 × 5.33 $mm^2$ and 0.5 W, respectively.

## 2. Introduction

An optical flow means a motion vector of a pixel between two successive pictures, which is a basis of a computer vision. By using the optical flows, moving objects in an image sequence or movement of a camera itself can be detected. Fig. 1 is a cut of an image sequence, “Yosemite”, and its optical flows. Since, in optical-flow calculation, a set of different equations has to be solved at every pixel, the computational cost reaches a few tens GOPS even in a CIF-30 (a resolution of a CIF and a frame rate of 30 fps) image sequence. Thus, a software approach with a general processor has only focused on a small part of a picture, or has traded off accuracy [1]. For full-coverage real-time operation, dedicated hardware is required.

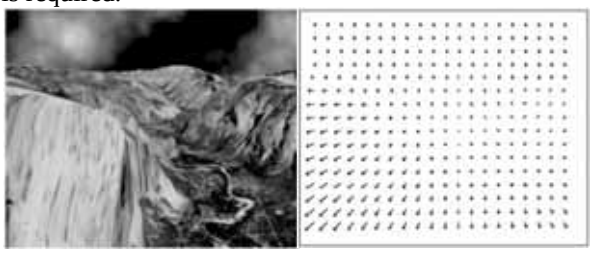


Fig. 1. “Yosemite” and its optical flows.

Fig. 2 shows an example of video segmentation for a surveillance system using the optical flow, in which contour of moving objects are extracted from the background. Other than the surveillance system, the optical flow is useful for various applications as categorized in Fig. 3. Since a pixel rate of an image sequence and required accuracy of an optical flow depend on an application, scalable architecture in terms of them is preferable.

Fig. 4 compares the proposed architecture with the conventional studies [2]-[3] in terms of accuracy (MAE: mean angle error) and pixel rate. As an algorithm, we adopt the hierarchical optical-flow estimation (HOE), which is implemented in the VLSI architecture. In fact, Fig. 2 shows a result of video segmentation with the HOE algorithm.

The proposed optical-flow processor has scalability that will be described in detail in Section 4, and thus the “4x Version” in Fig. 4 shows a scaled one by four. This version can handle a VGA-30 image sequence, which is superior to the conventional studies in terms of pixel rate. In addition, the MAE of the proposed architecture can be less than 10° by increasing the number of iteration times. Therefore, the MAE can be varied by an operation frequency, which indicates that a power can be scaled in a less accurate application.

Fig. 2. An example of video segmentation (the proposed algorithm, HOE, is used).

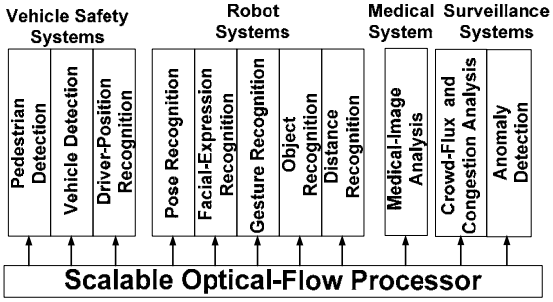



Fig. 3. Applications using optical flows.

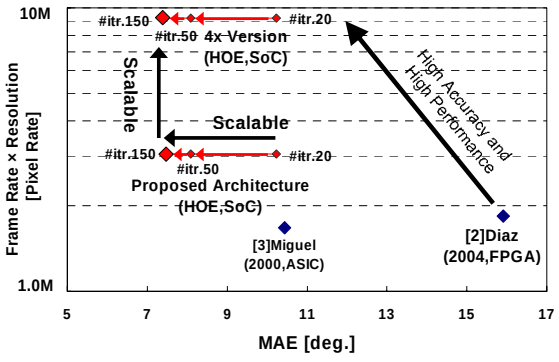



Fig. 4 Performance comparison.

## 3. Hierarchical Optical-Flow Estimation (HOE) Algorithm

Fig. 5 is a flow chart of the HOE algorithm [4] that is an improvement on the Horn and Schunck algorithm [5]. In the Horn and Schunck algorithm, a luminance gradient along a time axis is obtained using fifteen frames [6]. On the other hand, the HOE algorithm employs only three frames by means of a multi-dimensional gradient filter [7], which saves frame memory and reduces frame delay (latency). Besides, the HOE algorithm can adapt to a larger displacement of pixels thanks to the hierarchal images, while the conventional algorithm can detect just a displacement of two pixels at most [8]. By changing the number of hierarchy level ($L$ in Fig. 5), a detectable displacement is varied, which means the HOE algorithm has scalability of the displacement range in nature. For instance, when the hierarchy level is set to three, the HOE algorithm can follow a moving object within a displacement of five pixels.

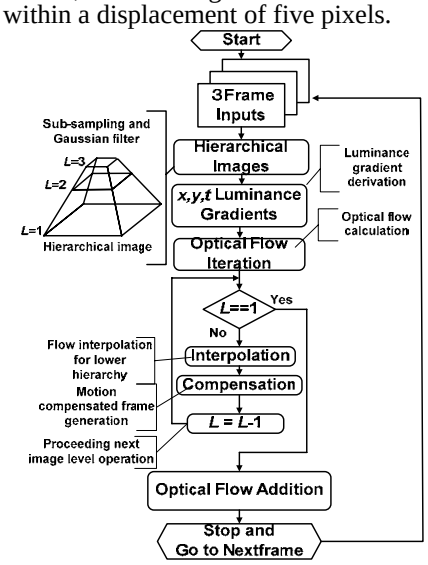



Fig. 5. A flow chart of the HOE algorithm.

In the HOE algorithm, the optical flows along $x$ and $y$ coordinates, $u_{n+1}$ and $v_{n+1}$, are given by (1), which is substantially the solution by the Gauss -Seidel method. The subscription of the optical flows, $n$+1, means the number of iteration times, and thus $\overline{u_n}$ and $\overline{v_n}$ are average optical flows obtained from eight neighboring pixels in the previous iteration. The iteration is executed until the difference between the current optical flow and previous one becomes less than a threshold (e.g. $|u_{n+1}-\overline{u_n}|<0.001$ pixel), otherwise the iteration reaches a preset value (e.g. 150 times). $I_x$, $I_y$, and $I_t$ are the luminance gradients along $x$, $y$, and $t$ coordinates, respectively, which are derived with the multi-dimensional filters mentioned previously. $\alpha$ is a weighted parameter, and avoids a local-minimum solution when $I_x$ and $I_y$ are small.

$$\begin{cases} u_{n+1} = \overline{u_n} - I_x \dfrac{I_x\overline{u_n} + I_y\overline{v_n} + I_t}{\alpha^2 + I_x^2 + I_y^2} \\ v_{n+1} = \overline{v_n} - I_y \dfrac{I_x\overline{u_n} + I_y\overline{v_n} + I_t}{\alpha^2 + I_x^2 + I_y^2} \end{cases} . \qquad (1)$$

## 4. VLSI Architecture

### 4.1. Bit-Length Optimization

Fig. 6 discusses bit lengths of an optical flow and luminance gradient in the HOE algorithm, using a several image sequences. Since the bit lengths of an optical flow and luminance gradient are expressed as decimals, shorter bit lengths are, of course, better for hardware implementation, while it results in a lower accuracy. The "Default" in the figure indicates a 32-b floating-point accuracy, where the MAE and the number of iteration times are the smallest. However, complicated hardware for the floating-point operation is necessary. In contrast, the "16-b Fixed-Point" operation is good for small-scale hardware, but the MAE and the number of iteration times worsens. The in-between "24-b Fixed-Point" operation has a comparable MAE to the "Default", but accumulation of errors caused by the less accuracy doubles the number of iteration times. Therefore, there is an optimum, and we select the "16/24-b Fixed-Point" format as the design choice. This format means that the luminance gradients have 16-b fixed-point accuracies, but the bit length of the optical flows is 24. The "16/24-b Fixed-Point" format exhibits the same accuracy and the same number of iteration times as the "Default".

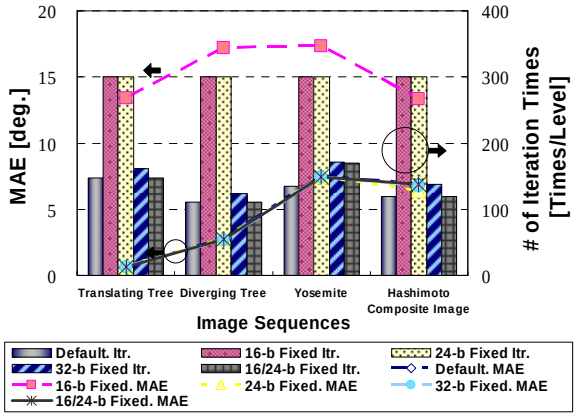



Fig. 6. MAEs and the numbers of iteration times in various bit-length formats.

### 4.2. Common Element (CE)

In the HOE algorithm, the iteration process of the optical-flow calculation using (1) is a major workload because the number of iteration times sometimes reaches at 150. Therefore, the iterative calculation of the optical flow must be executed fast by a feedback loop. In contrast, once the luminance gradients are computed, they do not need to be renewed during the iteration. This implies that dedicated circuits for the luminance gradients are useless.

In order to save the hardware resource, we propose a common element (CE) that carries out all calculations necessary to the optical-flow derivation. The CE can change its data path by a sequence controller, and cope with all kinds of calculations. Fig. 7 is a schematic of the CE including four processor elements (PE) so that a four-way SIMD architecture is implemented.

Fig. 8 illustrates a data path in a PE (compare with (1)). "AVE" averages eight optical flows in the previous iteration to obtain $\overline{u_n}$ and $\overline{v_n}$. The base element blocks ("BE1" and "BE2") calculate a numerator and denominator in (1). "DIV" is a divider. "U_V" computes an updated optical flow. "DIFF" checks the difference between the current and previous optical flows. These calculations are pipelined, and carried in a clock cycle.

With only one CE, an MAE of 7.44° is achieved in a

CIF-30 image sequence, “Yosemite”, when the maximum number of iteration times is set to 150 (see Fig. 4). This demonstrates that the proposed CE outperforms the conventional implementations.

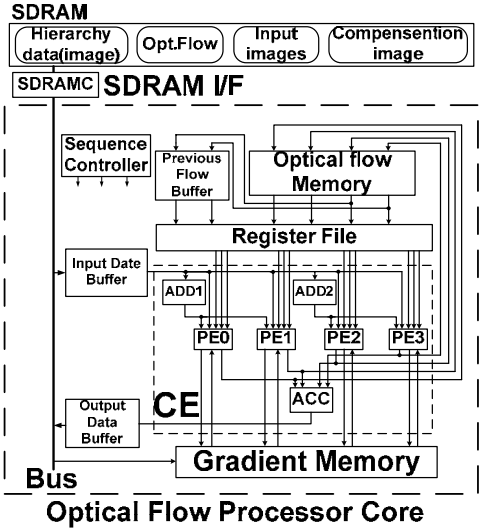



Fig. 7. A common element (CE).

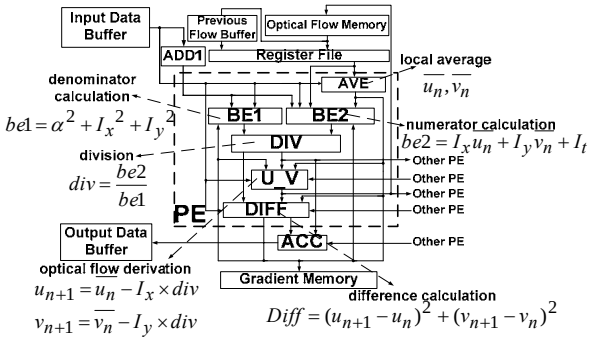



Fig. 8. A processing element (PE)

### 4.3 Scalability

The proposed processor has scalability in terms of accuracy and pixel rate. This is desirable since they depend on an application. Fig. 9 illustrates a multi-CE arrangement (two CEs), which proportionally enhances a pixel rate. When four CEs are aligned in parallel, a VGA-30 image sequence can be handled as already shown in Fig. 4, at a clock frequency of 189 MHz. The accuracy and corresponding power also can be scaled by reducing the maximum iteration times with a pixel rate kept, which is suitable to a less accurate application.

### 4.4. Memory

For one CE, each of optical-flow and luminance-gradient memories needs 4.86 Mb. Since the optical-flow memory is periodically accessed in the iteration, we adopt a 2-port DRAM compatible to a logic process in Fig. 10, for area efficiency. All the optical-flow data are read every 149 μs at least, and thus no refresh is required. The figure also shows a floor plan of the proposed optical-flow processor including one CE, where the DRAMs still occupy a half of the chip. The core size is 6.02 × 5.33 mm$^2$, estimated in a 90-nm CMOS process technology. If a memory design rule was available, the core size could be reduced to 40%, since we assumed a logic design rule here. This implies that the core size will become less than a half in an industrial design.

The total power is estimated at 500 mW in a CIF-30 image sequence. The power breakdown is 190 mW for logic circuits, 230 mW for the DRAMs, and 80 mW for SRAMs used as the luminance-gradient memory.

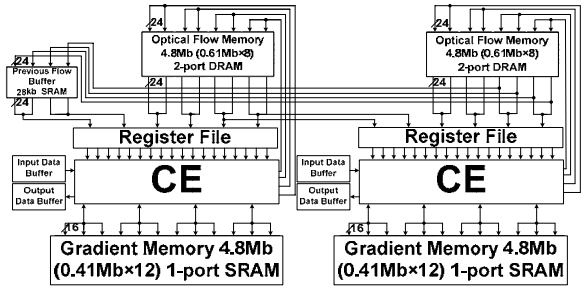



Fig. 9. Multi-CE arrangement.

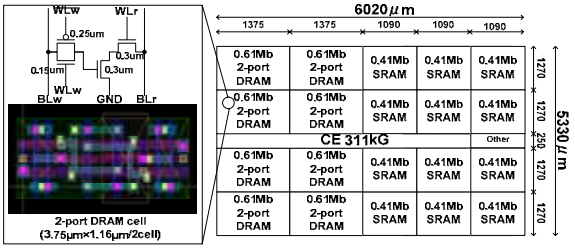



Fig. 10. A DRAM layout and floor plan.

## 5. Summary

We proposed the architecture of the optical-flow processor using the HOE algorithm, for video segmentation. The bit lengths of an optical flow, spatio-temporal-luminance gradients were optimized for the VLSI implementation. The common element (CE) carries out all calculations necessary to the optical-flow derivation, and saves hardware resources. The 2-port DRAM compatible with a logic process for optical-flow memory also reduces a core size. The proposed architecture is scalable in terms of pixel rate and accuracy, supposing various applications. The optical-flow processor can handle a pixel rate of a CIF-30 image sequence at 189 MHz and 500 mW. The core size is 6.02 × 5.33 mm$^2$ in a 90-nm process technology.


## Acknowledgments

This work has been supported by Semiconductor Technology Academic Research Center (STARC), and VLSI Design and Education Center (VDEC) of the University of Tokyo in collaboration with Cadence Design Systems, Inc and Synopsys, Inc.


## References


[1] J. Cohn, A. Zlochower, J. J. Lien, and T. Kanade, “Feature-Point Tracking by Optical Flow Discriminates Subtle Differences in Facial Expression”, Proc. IEEE Int. Conf. on Automatic FG, pp.396-401, 1998.

[2] J. Diaz, E. Ros, S. Mota, F. Pelay, and E. M. Orrigosa, “Real-Time Optical Flow Computation Using FPGAs”, Proc. ECV Workshop, 2004.

[3] M. V. Correia, and A. C. Campilho, “Real-Time Implementation of an Optical Flow Algorithm”, Proc. ICPR, vol.4, pp.247-250, 2002

[4] T. Yamamoto, K. Imamura, and H. Hashimoto, “Improvement of Optical Flow by Moving Object Detection using Temporal Correlation”, Trans. IIITE, vol.55, no.6, pp.907-911, 2001.

[5] B. K. P. Horn and B. G. Schunck, “Determining Optical Flow”, Artificial Intelligence, vol.17, pp.185-204, 1981.

[6] J. L. Barron, D. L. Fleet, and S. S. Beauchemin, “Performance of Optical Flow Techniques”, J. Comp. Vision, vol.12, no.1, pp.43-77, 1994.

[7] E. P. Simonceli, “Design of Multi-Dimensional Derivative Filters”, Proc. IEEE ICIP, vol.I, pp.790-794, 1994.

[8] B. Lucas, and T. Kanade, “An Iterative Image Registration Technique with an Application to Stereo Vision”, Proc. DARPA Image Understanding Workshop, pp.121-130, 1981.

# An Architecture Study of Scalable Optical-Flow Processor for Real-Time Video Segmentation

**Ryo YAMAMOTO, Yuki FUKUYAMA, Tadayoshi KATAGIRI †, Junichi MIYAKOSHI, Yuki KURODA †, Noriyuki MINEGISHI †, Masayuki MIYAMA †, Hiroshi KAWAGUCHI, Kousuke IMAMURA †, Hideo HASHIMOTO †, and Masahiko YOSHIMOTO**

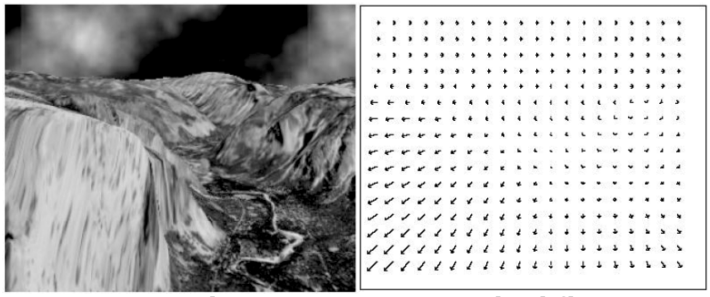
**Kobe University and † Kanazawa University**

## Outline

- **Background**
- **An optical flow algorithm**
- **Scalable optical flow processor architecture**
- **Performance estimation**
- **Summary**

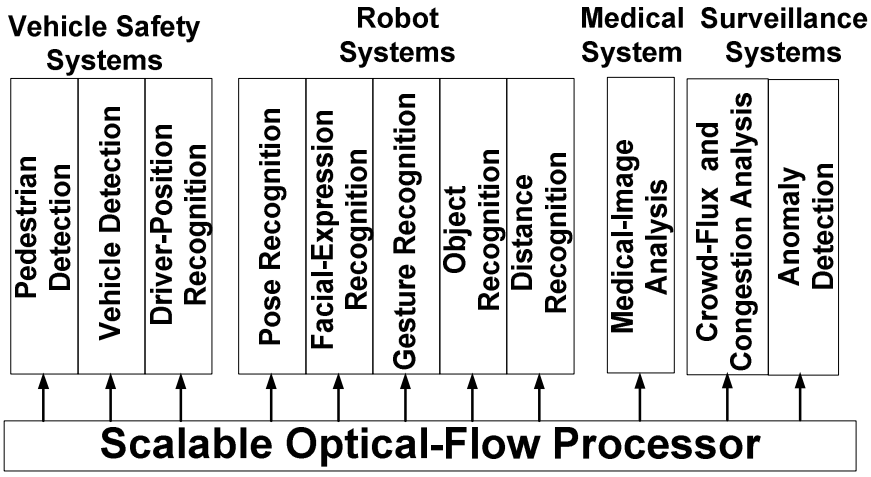
# What is optical flow?

- **A motion vector of a pixel between two successive pictures**
- **Heavy work-load for Computation**
- **Requirement for higher accuracy and higher resolution**



**“Yosemite”** **“Optical flow”**

# Applications using optical flow



# Requirements

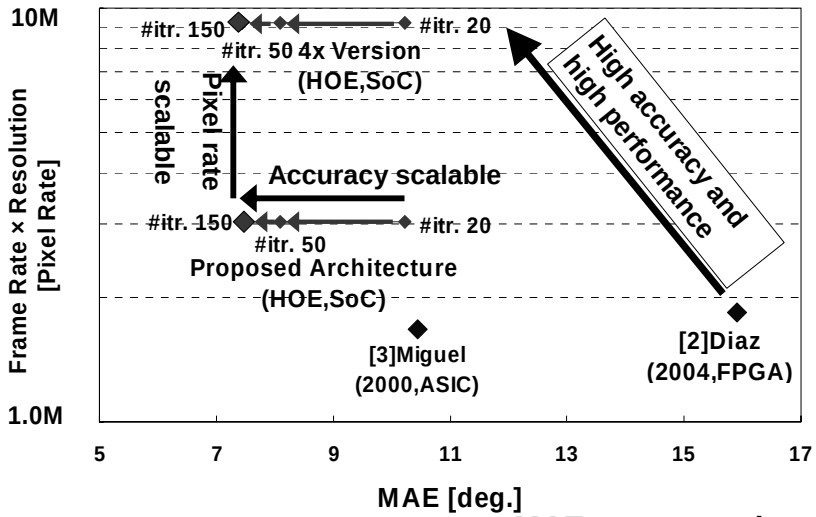
- **Processing time ⇒ Real-time**
- **Feature points ⇒ Full-coverage**

**The Dedicated hardware is required!**

- **Accuracy ⇒ Accuracy scalability**
- **Pixel rate ⇒ Pixel rate scalability**

**The scalable optical flow processor is required!**

# Position of this work

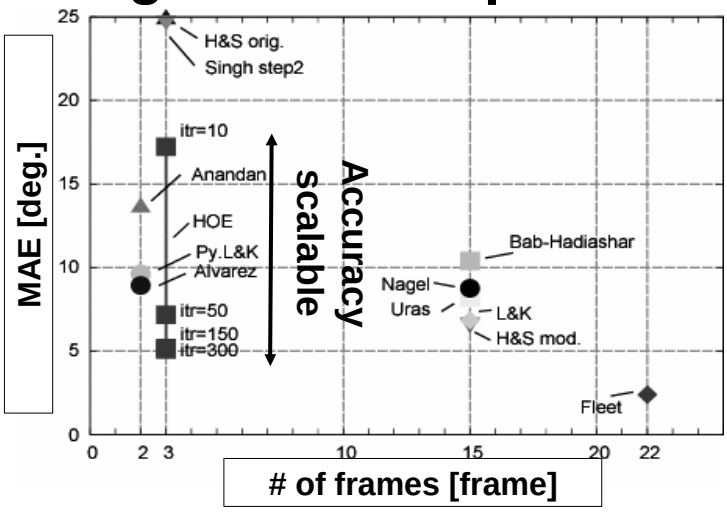



**MAE: mean angle error**

# Outline

- **Background**
- **An optical flow algorithm**
- **Scalable optical flow processor architecture**
- **Performance estimation**
- **Summary**

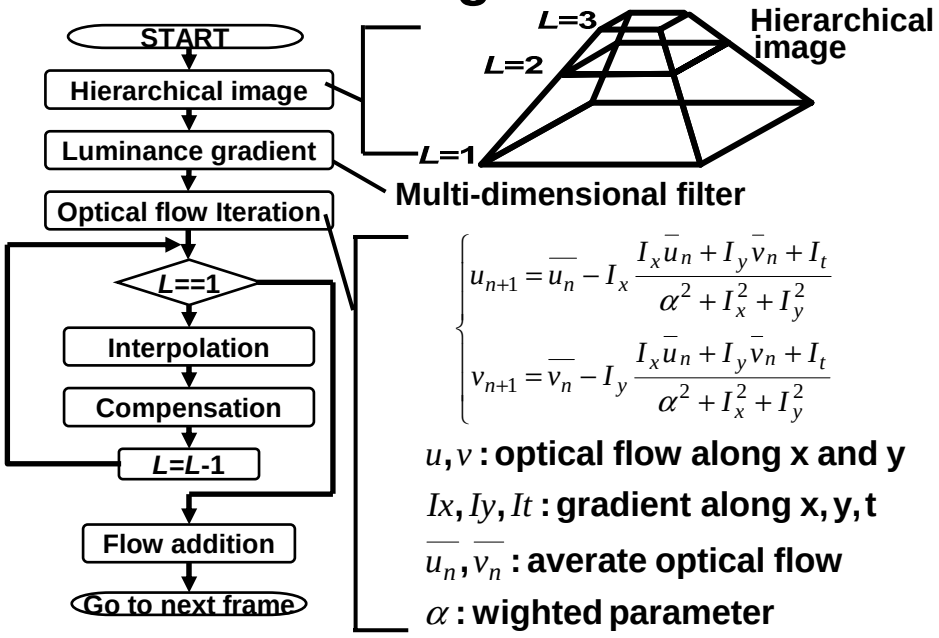
# Algorithm comparison




**Implementation efficiency and accuracy scalability**

**HOE (Hierarchical Optical flow Estimation)**

# HOE algorithm




$$u_{n+1} = \overline{u_n} - I_x \frac{I_x \bar{u}_n + I_y \bar{v}_n + I_t}{\alpha^2 + I_x^2 + I_y^2}$$

$$v_{n+1} = \overline{v_n} - I_y \frac{I_x \bar{u}_n + I_y \bar{v}_n + I_t}{\alpha^2 + I_x^2 + I_y^2}$$

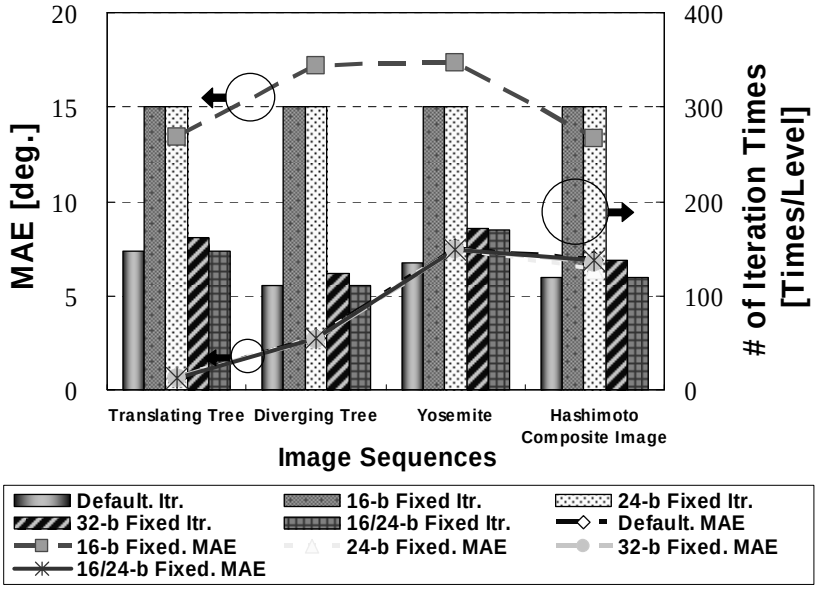
$u, v$ : optical flow along x and y

$Ix, Iy, It$ : gradient along x, y, t

$\overline{u_n}, \overline{v_n}$ : averate optical flow

$\alpha$ : wighted parameter

# Bit-Length optimization



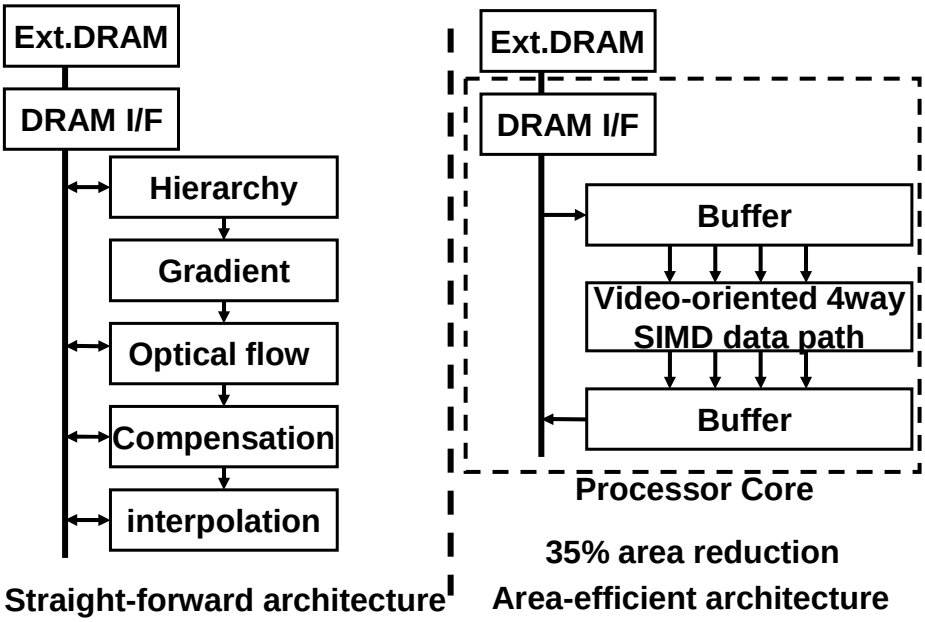
# Outline

- **Background**
- **An optical flow algorithm**
- **Scalable optical flow processor architecture**
- **Performance estimation**
- **Summary**

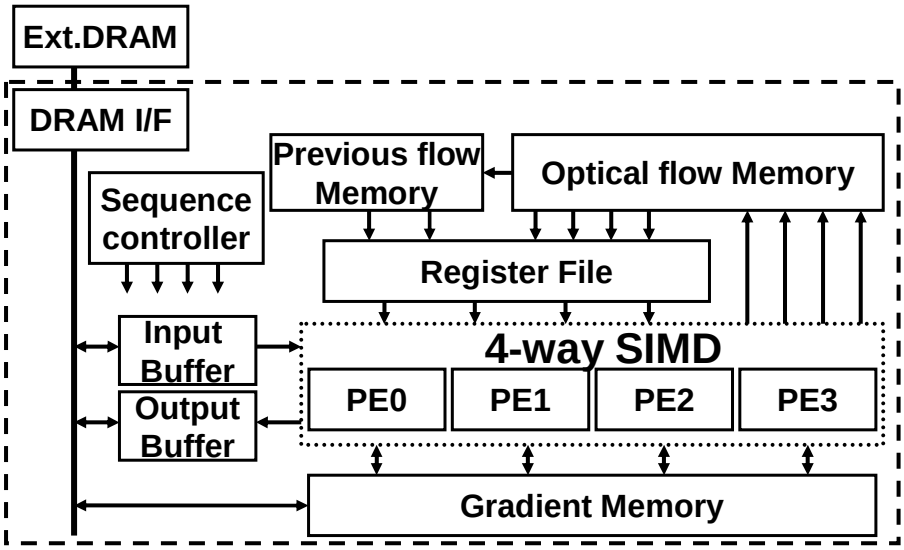
# Architecture features

- **Video-oriented 4-way SIMD data path**
- **Scalability for pixel rate and accuracy**
- **2-port 4Tr. DRAM for optical flow buffer**

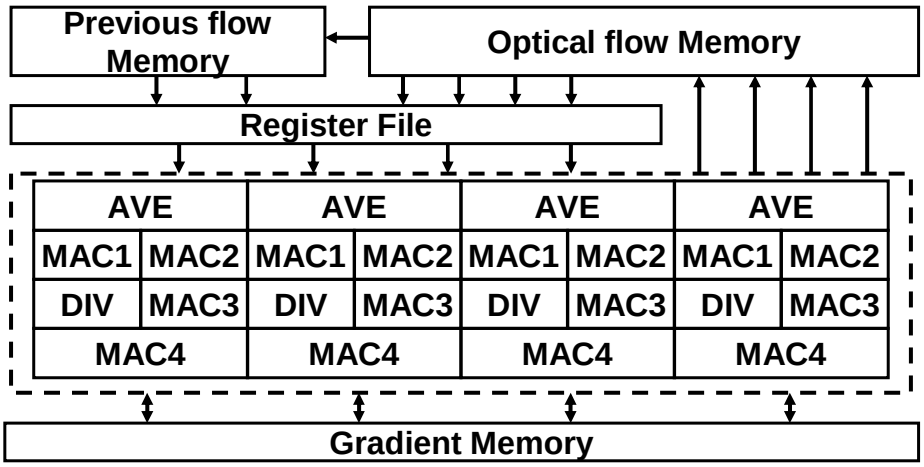
# Area efficient architecture




# Processor core



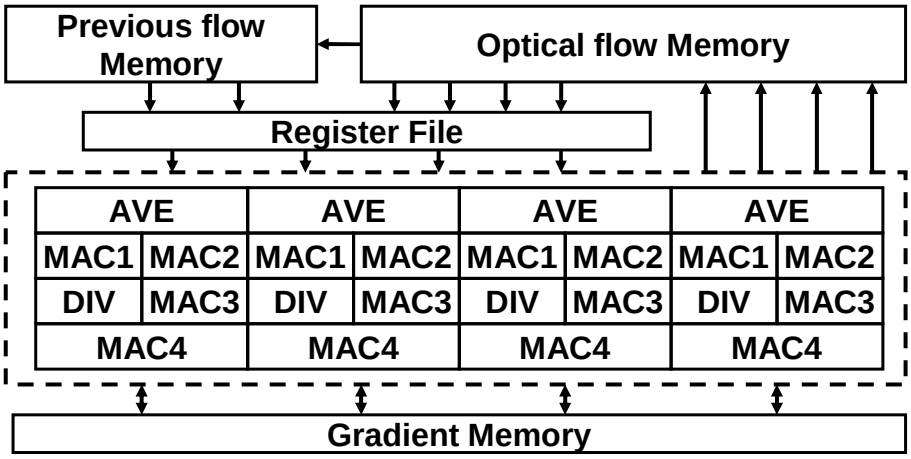
# Optical flow calculation




•AVE: average DIV: divider •MAC: multiply-accumulation

$$u_{n+1} = \overline{u_n} - I_x \frac{I_x \bar{u}_n + I_y \bar{v}_n + I_t}{\alpha^2 + I_x^2 + I_y^2}$$

# Gradient generation

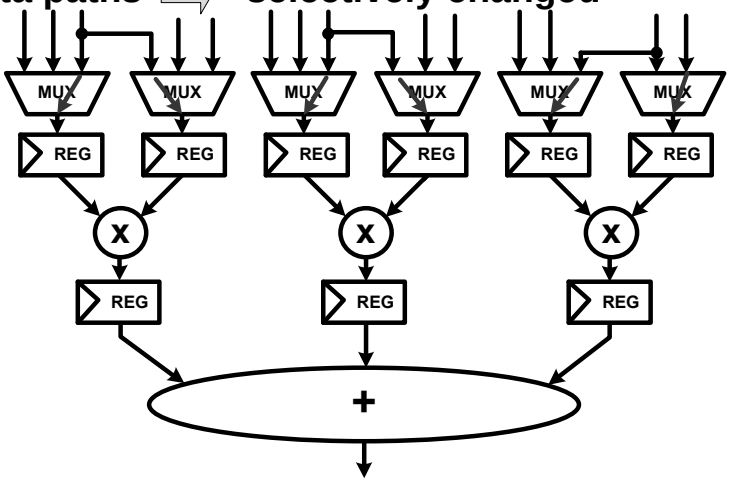



$$E\_ml \times lpf0 + E\_mm \times lpf1 + E\_mn \times lpf0$$

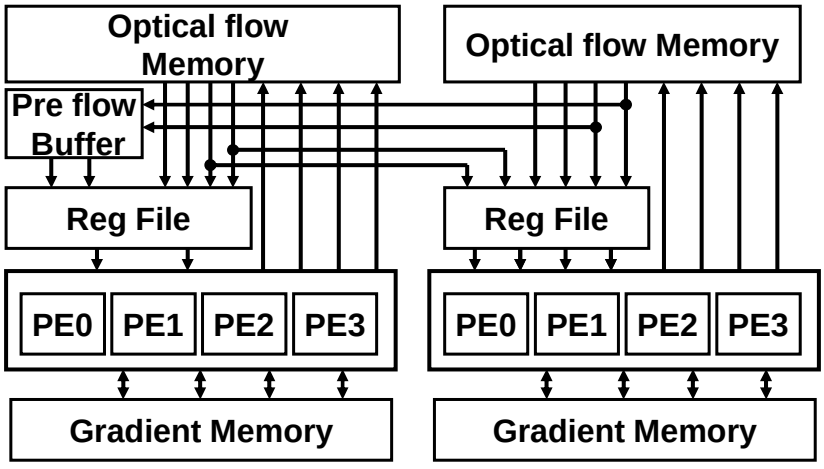
# MAC1

•Most of processes ⇨ multiply-accumulation

• Data paths ⇨ selectively changed



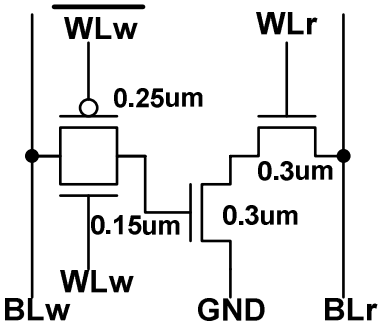

# Scalable architecture



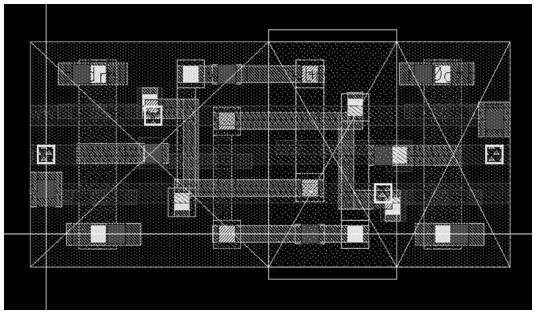

•Multiple core ⇨ High resolution applications

•Less iteration times ⇨ Low power applications

# 2-port 4Tr. DRAM




Memory cell schematic



Memory cell layout

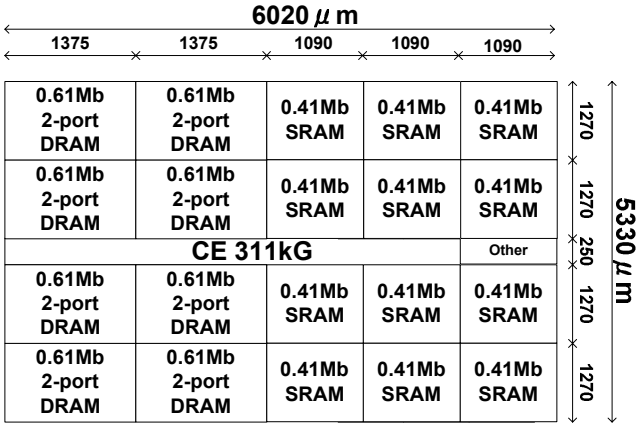
(3.75 $\mu$ m × 1.16 $\mu$ m/2cell)
90nm CMOS process

**58% area reduction**

# Outline

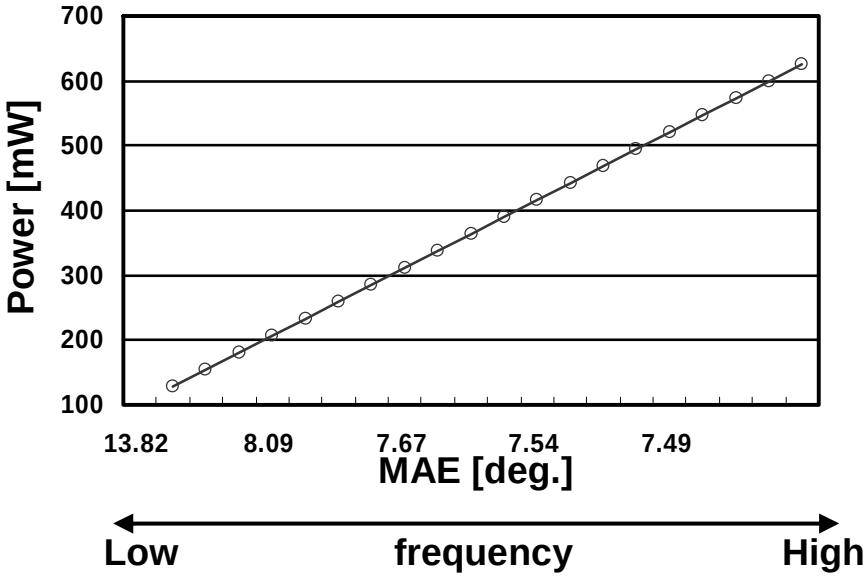
- Background
- An optical flow algorithm
- Scalable optical flow processor architecture
- Performance estimation
- Summary

# Floor plan & core characteristics

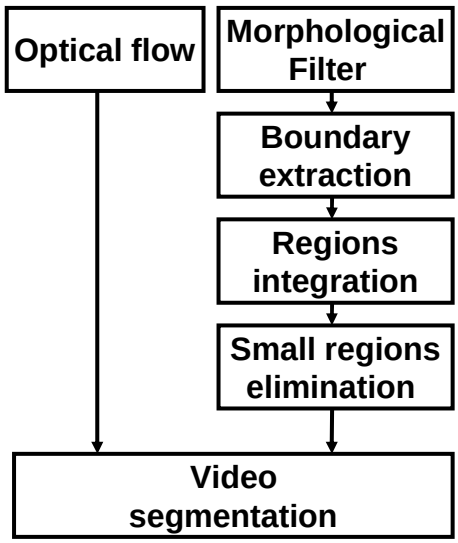



**CIF30 @90-nm**
- 4way SIMD
- 189MHz
- 9.6MbitRAM
- 32mm$^2$
- 500mW

# Power-accuracy trade-off



# Demonstration
# (Video Segmentation)




“Hall monitor”

# Outline

- Background
- An optical flow algorithm
- Scalable optical flow processor architecture
- Performance estimation
- Summary

# Summary

- **The algorithm, the architecture and the circuit are optimized for VLSI implementation**
- **The processor performs High accuracy (7.44 degree @Yosemite) and high pixel rate(CIF30fps)**
- **The power is estimated at 500mW**
- **The processor has scalability for power and pixel rate**
- **The scalable optical flow processor is suitable for practical use**

# Acknowledgements


- **This work has been supported by Semiconductor Technology Academic Research Center (STARC), and VLSI Design and Education Center (VDEC) of the University of Tokyo in collaboration with Cadence Design Systems, Inc and Synopsys, Inc.**