A VLSI architecture for VGA 30 fps video segmentation with affine motion model estimation

メタデータ	言語: eng 出版者: 公開日: 2017-10-03 キーワード (Ja): キーワード (En): 作成者: メールアドレス: 所属:
URL	https://doi.org/10.24517/00008074

A VLSI Architecture for VGA 30 fps Video Segmentation with Affine Motion Model Estimation

Masayuki Miyama, Yoshiki Yunbe, Kouji Togo, Yoshio Matsuda
Graduate School of Natural Science and Technology
Kanazawa University
Kanazawa, Ishikawa, 920-1192, Japan
E-mail: miyama@t.kanazawa-u.ac.jp

***Abstract*—This paper proposes a VLSI architecture for VGA 30 fps video segmentation with affine motion model estimation. The adopted algorithm is formulated as a contextual statistical labeling problem exploiting multiscale Markov random field (MRF) models. The algorithm optimization for VLSI implementation is characterized by image division method, ICM labeling limited to region boundary, and omission of motion models estimation for new regions. The optimization reduces the computational costs by 82 %, the amount of memory by 95 %, and the amount of data traffic by 99 % without accuracy degradation. The VLSI architecture is characterized by pipeline processing of the divided images, concurrent motion models estimation for multiple regions, and a common processing element of update and detection labeling. The architecture enables VGA 30 fps video segmentation with 167 MHz frequency. The estimated core area using 0.18µm technology is 30 mm^2. This processor is applicable to the video recognition applications such as vehicle safety, robot, and surveillance systems under the restriction of energy consumption.**

***Index Terms*—video segmentation, motion segmentation, affine motion model estimation, real-time processing, FPGA, VLSI**

I. Introduction

Video recognition is necessary for various applications such as vehicle safety systems, robot systems, and surveillance systems [1,2]. Video segmentation partitions video into spatial, temporal or spatio-temporal regions that are homogeneous in color, texture and/or motion [3,4]. Motion segmentation, which is a kind of video segmentation, labels pixels at each frame that are associated with independently moving parts of a scene. Motion segmentation is an important basis of the video recognition.

Many motion segmentation methods have been proposed. The first category is composed of top-down hierarchical schemes, which consist in the computation of successive dominant motions. Significant areas consistent with the current dominant motion are associated with a single label, while the process is iterated on the remaining data. A drawback of these methods is that they generally break down in the absence of a well defined dominant motion. Clustering methods fall into the second category. They estimate the optical flow field between two frames and then segment the image into regions by clustering pixels with similar motions. One important shortcoming of these methods is that clustering in the parameter space is usually sensitive to the number of specified clusters.

We adopted a motion segmentation algorithm proposed in [5]. The Pseudo M-estimator (PSM) algorithm is adopted for motion model estimation because it can estimate only a dominant motion of a target region without effects of outlier motions by weighting for each pixel [6]. A statistical regularization approach based on multiscale Markov Random Field (MRF) is adopted for region labeling to overcome the drawback of the first category. Thanks to the robustness of the PSM algorithm, the segmentation algorithm does not need the time-consuming alternate iterations of the motion model estimation and the update of region boundary.

However, the segmentation algorithm entails enormous computational costs. Real-time processing is impossible using software approaches. Several systems to estimate moving regions in real-time have been developed [7, 8]. In [7], motions and shapes of the regions are not estimated accurately because of a small number of feature points for background estimation and coarse sampled contour model for region detection. In [8], non-adoption of statistical regularization approach for region labeling may cause sensitivity to noise. Both of them were implemented with several general-purpose DSPs. A dedicated VLSI processor for the motion segmentation with high accuracy has not been proposed. This paper proposes the VLSI architecture for VGA 30 fps video segmentation with affine motion model estimation.

II. Video Segmentation Algorithm

A. Motion Model with Affine Parameters

In affine motion estimation, a motion is expressed using two-dimensional affine transformation. An affine motion model ***A***, estimated from two successive images, comprises six affine parameters. Motion vectors for each pixel are calculable as follows.

$$\mathrm{A}^{t} = (\alpha_1\ \alpha_2\ \alpha_3\ \alpha_4\ \alpha_5\ \alpha_6) \tag{1}$$

$$\begin{cases} u_i = a_1 + a_2 x_i + a_3 y_i \\ v_i = a_4 + a_5 x_i + a_6 y_i \end{cases} \tag{2}$$

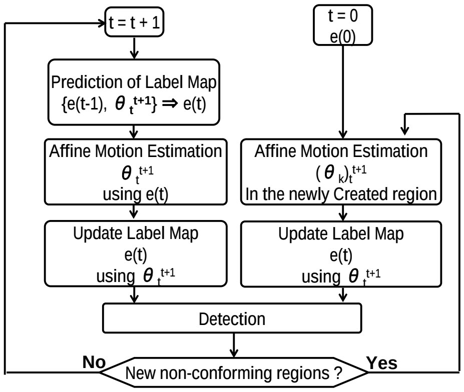


Figure 1. Flow chart of the video segementation algorithm.

B. Flowchart

Fig.1 shows a flow chart of the algorithm. *e(t)* denotes the label map at *t*, and $\boldsymbol{\theta}_t^{t+1}$ is the set of motion models associated with *e(t)* and accounting for the description of the motion field between time *t* and *t+1*. The determination of the label map at time *t*, given the label map at time *t-1*, involves four main steps.

1) Step.1 Prediction of Label Map

The label map of the partition at time *t*, denoted $\tilde{e}(t)$, is determined using the label map along with the estimated motion models obtained at time *t-1*.

2) Step.2 Affine Motion Model Estimation using PSM

The motion models $\boldsymbol{\theta}_t^{t+1}$ are estimated using the initial partition $\tilde{e}(t)$. The PSM algorithm is a robust estimation method using the M-estimator concept. To derive a motion model, an error function is defined as follows.

$$r_i = J(x_i + u_i, y_i + v_i) - I(x_i, y_i) + \xi \tag{3}$$

Therein, *I* and *J* respectively represent a current image and the subsequent image. This equation fundamentally assumes the luminance conservation row. Parameter ξ represents the global luminance change. A motion model $\boldsymbol{\theta}$ of a region *F* is defined as a vector to minimize (3) over all pixels in *F* with the weighted least squares method. It is expressed as follows.

$$\boldsymbol{\theta} = \boldsymbol{G}^{-1} \cdot \boldsymbol{G_s}$$

$$= \left(\sum_{(x_i, y_i) \in F} w_i \boldsymbol{\chi_i}^t \boldsymbol{\chi_i} \right)^{-1} \left(\sum_{(x_i, y_i) \in F} w_i \boldsymbol{\chi_i}^t \mathbf{y}_i \right)$$

$$\begin{cases} \boldsymbol{\theta} = (\boldsymbol{A}^t \quad \xi) = (a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6 \quad \xi) \\ \mathbf{y}_i = -I_t \\ \boldsymbol{\chi_i} = (I_x \quad I_x x_i \quad I_x y_i \quad I_y \quad I_y x_i \quad I_y y_i \quad 1) \end{cases} \tag{4}$$

In those equations, the luminance gradient matrices are represented as $\boldsymbol{G}$ and $\boldsymbol{G_s}$. The luminance gradients in *x*, *y*, and *t* directions are denoted respectively as I_x, I_y, and I_t. The weight for each pixel is denoted as w_i.

3) Step.3 Update of Label Map

Given the predicted map $\tilde{e}(t)$ and the estimated models $\boldsymbol{\theta}_t^{t+1}$, the estimation of the optimal partition $\hat{e}(t)$ is achieved through minimizing the energy function composed of three terms as follows.

$$U(e, o, \tilde{e}) = U_1(e, o) + U_2(e) + U_3(e, \tilde{e}) \tag{5}$$

The field of observations *o* is composed of the images at time *t* and *t+1*. The data driven term U_1 expresses the adequacy between the labels and the observations. The energy term U_2 accounts for the expected spatial properties (homogeneity) of the label field. U_3 favors the conservation of labels over time. The optimal label to minimize the local energy is determined for each pixel.

TABLE I. COMPARISON OF DATA TRAFFIC AND MEMORY AMOUNT.

		Conventional	Image division (128 × 128)
Memory bit [kbyte]	Internal	50.4	41.6
	External	787.2	0
Data traffic [Mbps]		18,855.936	82.944

4) Step.4 Detection of New Region

Within each region, sub-areas whose motion does not conform to the estimated motion model are detected. This is achieved through minimizing the energy function like as (5). Then the motion models are estimated in the newly created regions. These are repeated until no new region is detected.

III. ALGORITHM OPTIMIZATION FOR VLSI

A. Image Division Method

An internal memory to store the whole VGA image is impractical because it would require a large chip. An external memory to store the VGA image is also difficult to implement because of the huge data traffic between the processor and the memory. The image division method is proposed to solve these problems. This technique divides a large image into numerous small images; then the divided images are processed independently. Table 1 presents a comparison of the data traffic and memory amount between the original and proposed method. The proposed method reduces both the memory amount and data traffic.

B. ICM Labeling Limited to the Region Boundary

An optimal label map is obtained through minimizing the energy function both in Step.3 and Step.4 of the algorithm. The original algorithm adopts the Highest Confidence First minimization procedure, which assigns a region label to each pixel in order of confidence. The procedure improves results, but it is very complex for the hardware implementation. We adopt the standard Iterative Conditional Mode instead of the HCF. The ICM procedure assigns a label to each pixel in the raster scan order.

The original ICM procedure attempts to update all labels of a map, even though the label is surrounded with the same region labels. This feature tends to make a lot of small meaningless regions. Therefore we change the algorithm so that it does not update the label at the pixel surrounded with the eight pixels having the same region label. The algorithm only updates labels around the region boundary. This method both improves partition results and reduces the computational costs by 93 %.

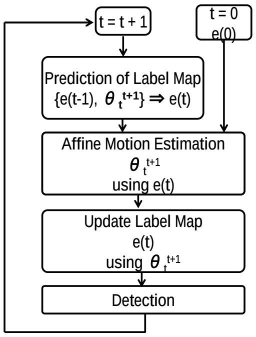


Figure 2. Flow chart of the proposed video segementation algorithm.

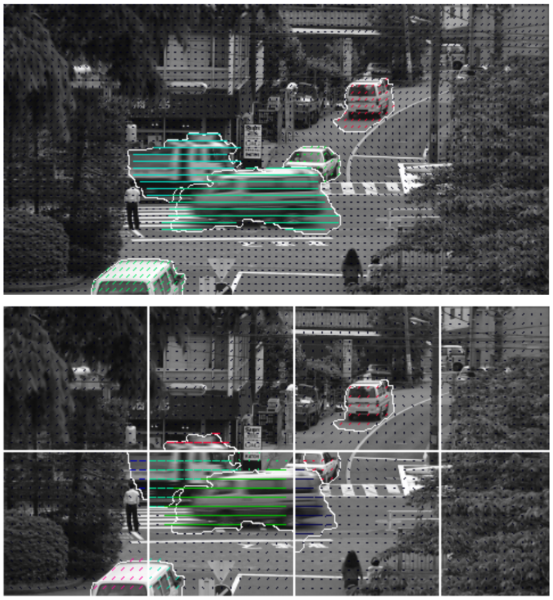


Figure 3. Simulation results (upper:original, lower:proposed).

C. *Omission of Motion Estimation for New Region*

According to the flowchart shown in Fig.1, Detection, Motion Estimation and Update steps are usually repeated many times. Real time processing of video segmentation becomes very hard because the computational costs of the motion estimation by the PSM algorithm are huge. To solve this problem we change the flowchart as shown in Fig.2 In the new flowchart, the motion models corresponding to the newly detected regions are not estimated at time *t-1*. They are estimated at the next time *t*. The Motion Estimation step is the only once, even though the many new regions are detected. The proposed method reduces the computational costs by 60 %.

D. *Simulation Results*

Fig.3 shows simulation results. The original algorithm (a) partitions cars appropriately. The proposed algorithm (b) partitions cars almost the same as the original. Note that the region boundary across the boundary of divided images continues in (b). The algorithm optimization for the VLSI implementation reduces the computational costs by 82 %, the amount of memory by 95 %, and the amount of data traffic by 99 % without accuracy degradation.

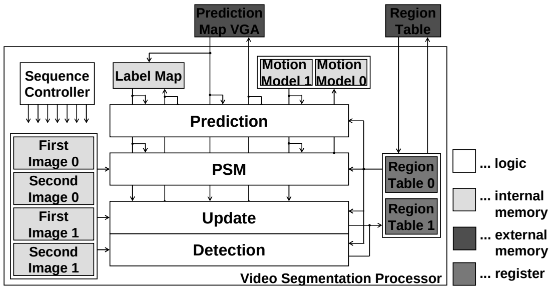


Figure 4. Block diagram of video segmentation processor.

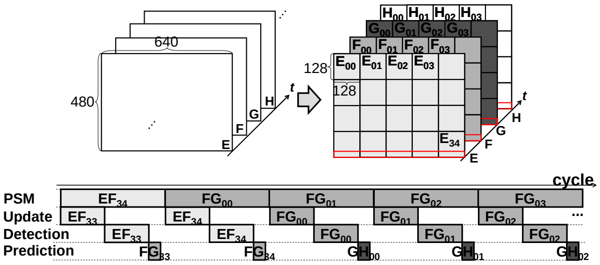


Figure 5. Timing diagram of the proposed processor.

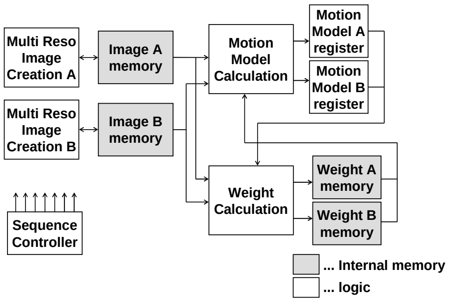


Figure 6. Block diagram of PSM processor.

IV. VLSI ARCHITECTURE

A. *Video Segmentation Processor*

Fig.4 shows the block diagram of the video segmentation processor. The processor consists of Prediction, PSM, Update, Detection, and memories.

B. *Pipeline Processing of Divided Images*

Fig.5 shows the timing diagram of the proposed processor. The pipeline consists of two stages: the PSM stage and the other stage composed of Update Detection, and Prediction. The Update, Detection, and Prediction for the divided image and the PSM for the next divided image are executed in parallel. Note that the Prediction is executed just after the Detection of the divided image at the same position of the previous frame. The proposed method doubles the throughput and enables VGA 30 fps video segmentation in real time with 167 MHz frequency.

C. *Concurrent Motion Estimation of Multiple Regions*

Fig.6 shows a block diagram of the PSM processor. Each processing block operates pixel-by-pixel in a pipeline fashion. Fig.7 shows a block diagram of the Motion Model Calculation.

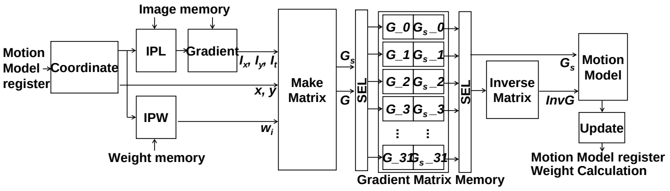


Figure 7. Block diagram of Motion Model Calculation.

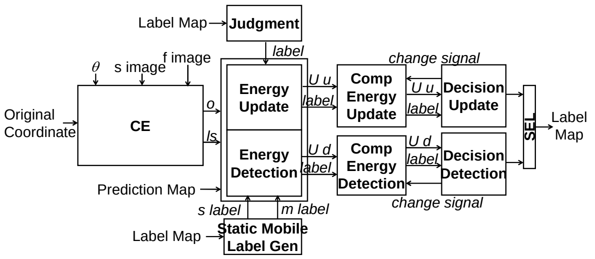


Figure 8. Block diagram of Update and Detection.

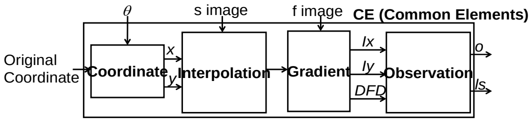
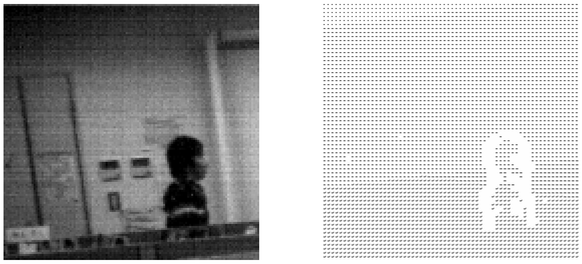


Figure 9. Block diagram of common element.



(a) Shot Image

(b) Motion Vector

Figure 10. Estimation Result of Real-Time System.

Gradient Matrix Memory is inserted to realize concurrent motion models estimation of multiple regions.

D. *Common Processing Element of Update and Detection Labeling*

Fig.8 shows a block diagram of Update and Detection. The CE is a common processing element of Update and Detection, shown in Fig.9. The share of CE results in gate size reduction.

V. VLSI IMPLEMENTATION

A. *FPGA Implementation of PSM Processor*

Table 2 shows the prototypical FPGA implementation of the PSM processor, which is a part of the proposed video segmentation processor. Fig. 10 (a) depicts an example of an image taken by the system. The image background has a motion panning to the left and the human moves to the right. Fig.10 (b) shows an estimation result. The background is estimated as a region of a dominant motion. The weight in the human part is 0, and the motion vectors in the part are not displayed. The silhouette of a human is extracted well.

TABLE II. RESULTS OF FPGA IMPLEMENTATION.

Resource	Utilization	Percentage
4 input LUT	63,779 out of 135,168	47%
18 × 18 Multiplier	95 out of 96	98%
Block RAM	128 out of 288	44%

TABLE III. CHARACTERISTIC OF THE PROPOSED PROCESSOR.

Performance		VGA 30fps
Operating Frequency		167MHz
Logic Gates(2 input NAND)		763,106 gates
Internal Memory	1port	85,380kByte
	2port	28,672kByte
Area(0.18 μ process)		29.536mm^2

B. *Area and Performance Estimation*

Table 3 shows the estimated characteristic of the proposed processor. The core area using 0.18μm technology is 30 mm^2.

VI. CONCLUSION

This paper proposed a VLSI architecture for VGA 30 fps video segmentation with affine motion model estimation. We adopted the model-based motion segmentation algorithm. The algorithm optimization reduced the computational costs by 82 %, the amount of memory by 95 %, and the amount of data traffic by 99 %. Simulation results showed the optimized algorithm partitioned moving regions without accuracy degradation. The proposed architecture enabled VGA 30 fps video segmentation with 167 MHz frequency. The estimated core area using 0.18μm technology was 30 mm^2. The ASIC implementation and its evaluation are future works.

ACKNOWLEDGMENT

This research was supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (C), 19560339, 2007-2009. This work was supported by the VLSI Design and Education Center (VDEC) and The University of Tokyo, in collaboration with Celoxica, Ltd.

REFERENCES

[1] Bruno Siciliano, and Oussama Khatib, “Springer Handbook of Robotics,” Springer-Verlag, 2008.

[2] Sergio A. Velastin, and Paolo Remagnino, “Inteligent Distributed Video Surveillance Systems,” The Institution of Electrical Engineers, 2006.

[3] Al Bovik, “Handbook of Image & Video Processing Second Edition,” Elsevier Academic Press, 2005.

[4] Yu-Jin Zhang, “Advances in Image and Video Segmentation,” IRM Press (an imprint of Idea Group Inc.), 2006.

[5] J. M. Odobez, and P. Bouthemy, “Direct Incremental Model-Based Image Motion Segmentation for Video Analysis,” Signal Processing 66, 1998, pp. 143-155.

[6] J.M. Odobez, and P. Bouthemy, “Robust multiresolution estimation of parametric motion models applied to complex scenes,” publication interne n° 1994, 788.

[7] S. Araki, T. Matsuoka, N. Yokoya, and H. Takemura, “Real-Time Tracking of Multiple Moving Object Contours in A Moving Camera Image Sequence,” IEICE Transactions on Information and Systems E83-D (7), 2000 pp. 1583-1591.

[8] S. M. Smith, and J. M. Brady, “ASEET-2: Real-Time Motion Segmentation and Shape Tracking,” IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 17, No. 8, August 1995.

[9] Y. Yunbe, M. Miyama, and Y. Matsuda, “A VGA 30fps Affine Motion Estimation Processor for Real-Time Video Segmentation,” IASTED Circuits & Systems, No. 625-010, Kailua-Kona, Hawaii, USA, August 18-20, 2008.