Single Chip Video Segmentation System with a Programmable PE Array

Shao-Yi Chien [*] , Yu-Wen Huang, Bing-Yu Hsieh, and Liang-Gee Chen
DSP/IC Design Lab, Graduate Institute of Electronics Engineering
Department of Electrical Engineering, National Taiwan University
1, Sec. 4, Roosevelt Rd., Taipei 106, Taiwan
{shoayi, yuwen, bingyu, lgchen}@video.ee.ntu.edu.tw

ABSTRACT

Video segmentation is a very important unit in content-based video encoding systems, such as MPEG-4. In this paper, a single chip video segmentation system is proposed. First, a hardware-oriented video segmentation algorithm is developed, which contains only local pixel operations and morphological operations. Simulation results show that the segmentation results of the proposed algorithm are satisfied. To achieve both high throughput and flexibility, the system is then mapped to a hardware architecture with a programmable PE array. This chip is designed with cell-based design flow and is currently under fabrication by TSMC using $0.35\mu m$ 1P4M technology. Simulation shows that the prototyping chip can achieve the processing speed of 30 QCIF frames per second and 7,680 binary morphological operations per second at 26MHz with small chip size. This chip can be integrated into any camera or real-time encoding system to support content-based coding capability.

1. INTRODUCTION

MPEG-4 standard is designed for multimedia applications [2]. The most important feature of MPEG-4 is content-based coding, where a video sequence is encoded object by object rather than frame by frame as conventional video coding standard, such as MPEG-2. Video segmentation is the technique to generate the shape information of video objects, which is required for shape coding, texture coding, motion estimation, and motion compensation in a content-based coding system. It is a very important part since without video segmentation, most of the new functionalities of MPEG-4 cannot be realized. Besides, MPEG-4 has many real-time applications; therefore, a real-time video segmentation system is urgently required for a MPEG-4 encoding system.

Many video segmentation algorithms have been proposed [1]. Some of them are based on motion information generated with motion estimation, optical flow, or change detection [6]. The segmentation results are usually not precise enough and need to be refined with complicated post-processing. Others of them are based on both motion information and spatial information. The spatial information can be derived from edge information or region information given by image segmentation algorithms, such as watershed transform [5, 7, 9]. These algorithms can give precise segmentation results; however, they are all too computationally intensive to be applied in real-time applications. Furthermore, these algorithms are not easy to be implemented in hardware since many global operations, such as watershed transform, are included, and most of the operations are not regular enough to be paralleled or pipelined.

In this paper, a single chip video segmentation system is proposed. A hardware-oriented video segmentation algorithm is first proposed, which is modified from our prior successful algorithm [3]. In this algorithm, all the operations are local pixel operations or morphological operations [8], which are very regular and suitable for hardware implementation. Besides, a programmable morphology PE array architecture is also proposed to provide both high throughput and flexibility required by a video segmentation system.

The hardware-oriented video segmentation algorithm is first described in next section. Then the hardware architecture and simulation results are shown. Finally, a short conclusion is given.

2. HARDWARE-ORIENTED VIDEO SEGMENTATION ALGORITHM

The block diagram of the proposed hardware-oriented video segmentation algorithm is shown in Fig. 1. It includes two modes: a baseline mode is designed for general situations, and a shadow cancellation mode is designed for video sequences influenced by shadow and light change.

Figure 1(a) shows the block diagram of baseline mode. It has five main parts: frame difference, background registration, background difference, object detection, and post-processing. First, the frame difference of current frame and previous frame are thresholded to form *Frame Difference Mask (FDM)*. The mask is then used in background registration to register background information into background buffer. The reliable background consists of pixels that are not parts of moving objects for consecutive *fth* frames. Next, the frame difference of current frame and background frame is also thresholded to form *Background Difference Mask (BDM)*. In object detection, if background exists, *BDM* is chosen as *initial object mask (OMi)*; otherwise, *FDM* is chosen. Finally, post-processing can refine *OMi* to *object mask (OM)*.

There are two parts in post-processing: noise region elim-

[*] Thanks SiS Education Foundation for financial support.

0-7803-7363-4/02/$17.00 ©2002 IEEE.

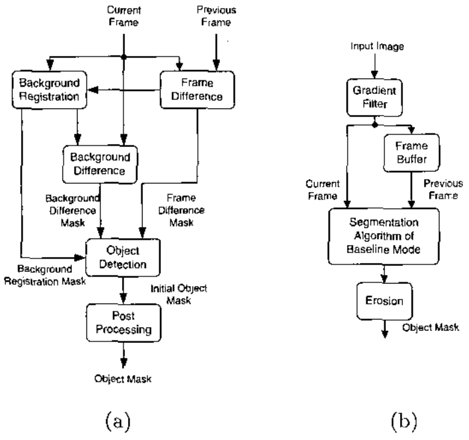

Figure 1: Block diagram of the hardware-oriented video segmentation algorithm. (a) Baseline mode; (b) shadow cancellation mode.

ination and morphological close-open operation. The noise region elimination can filter out small black regions (holes) and small white regions (noise) of *OMi*. Rather than being implemented with connected component operation, which is a global operation, it is implemented with morphological dilation, conditional erosion, and opening operations, which can be shown in the following equation:

$$(\dots((I\oplus B_m)\underbrace{\ominus B_3; I)\dots\ominus B_3; I)}_{l}\circ B_3, \tag{1}$$

where $l > (n-1)/2$, $\oplus$ is dilation, $\ominus$ is erosion, B_n is an nxn structuring element, $\circ$ is opening, which can be described by the following equation:

$$I\circ B = (I\ominus B)\oplus B, \tag{2}$$

and the conditional erosion (geodesic erosion) is:

$$X\ominus B; Y = (X\ominus B)\cup Y. \tag{3}$$

In shadow cancellation mode, which is shown in Fig. 1(b), morphological gradient filter is first applied to depress the influence of light change and shadow. An extra erosion operation is added in post-processing to eliminate the edge-thickening effect of gradient filter.

Some segmentation results are shown in Fig. 2. Several standard sequences and other sequences captured in our lab with a general camera are tested. It is shown that the proposed algorithm performs well for various situations and can successfully segment out the object masks.

No global operations and complicated branch operations are included in this algorithm. All operations are either pixel based local operations or morphological operations, which can be mapped to efficient hardware architecture. Therefore, the proposed video segmentation algorithm is very suitable for hardware implementation.

3. CHIP ARCHITECTURE

The architecture of the video segmentation chip is shown in Fig. 3. It has two parts: gray-scale part and binary part.

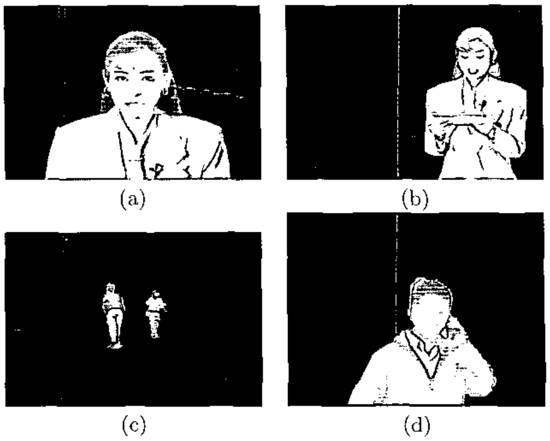

Figure 2: Segmentation results of the proposed algorithm. (a) Akiyo #50; (b) Weather #50; (c) Hall Monitor #50;(d) Frank #50.

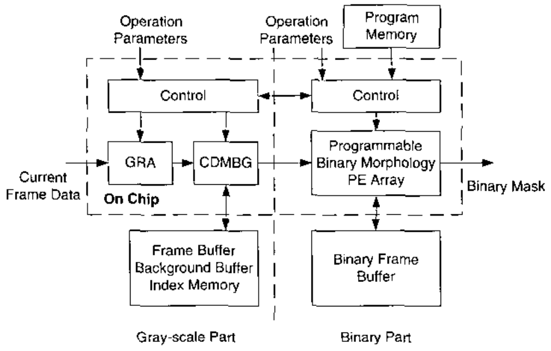

Figure 3: Architecture of the video segmentation chip.

The data width in gray-scale part is 8-bit, and the data width in binary part is 1-bit.

In gray-scale part, there are three units: control unit, gradient filter (*GRA*), and change detection mask and background generator (*CDMBG*). *GRA* applies morphological gradient filter on current frame data to eliminate shadow effect and light change effect. *CDMBG* can generate both *initial object mask (OMi)* and background information. The background information (B), previous frame (P), background indicator (BI), and stationary index (SI) are stored in an off-chip memory. Note that the background indicator can indicate where the background information exists, and the stationary index records the probability a pixel belongs to background. *OMi*, whose data width is 1-bit, is then sent to binary part.

The main unit in binary part is a programmable binary morphology PE array. The PE array can be programmed by instructions stored in a program memory to perform different kinds of binary morphological operations in post-processing of our algorithm. For complex operations, a fold-

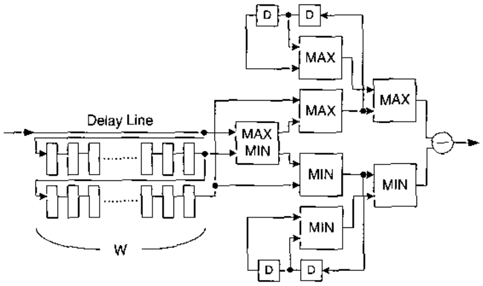

Figure 4: Architecture of morphological gradient filter using partial-result-reuse architecture.

ing technique is applied, and an off-chip memory is used to store the partial results.

The detailed architectures of *GRA*, *CDMBG*, and *Programmable PE Array* are described in the following subsections.

3.1 Gradient Filter

The morphological gradient filter can be described by the following equation:

$$G = (I \oplus B) - (I \ominus B). \qquad (4)$$

The hardware architecture is shown in Fig. 4. In Fig. 4, w is the width of the frame, and *MAXMIN* can output the maximum and the minimum of its two input values. The gradient filter is implemented with a hardware-cost efficient architecture named partial-result-reuse architecture, in which the partial results during operations are kept and reused and is proved to be superior to other morphology architectures [4]. It needs only seven 8-bit comparators in the gradient filter. Note that, for size and power consideration, the delay lines in this chip are implemented with two on-chip SRAMs rather than registers.

3.2 CDMBG

Frame difference, background registration, background difference, and object detection in Fig. 1 are processed in *CDMBG* module. All these operations are pixel operations, that is, the operation is applied pixel by pixel independently; therefore, it can be easily mapped to hardware architecture with high throughput. *CDMBG* can be implemented as shown in Fig. 5, where *DIFF Th* is the frame difference and thresholding unit, and *Decision Logic* decides if the current input pixel is a part of reliable background. If it is, it will be written into background, or the background will be unchanged.

3.3 Programmable PE Array

The post-processing including noise region elimination and close-open operation is the most computationally intensive part in our algorithm. The post-processing includes three basic binary morphological operations: dilation, erosion, and conditional erosion.

To achieve real-time requirement, the throughput should be high, and a PE array architecture is suitable; however, the operations in post-processing are different for different situations, and a flexible architecture is required. To achieve both high throughput and flexibility, a programmable binary morphology PE array architecture is proposed as shown in Fig. 6.

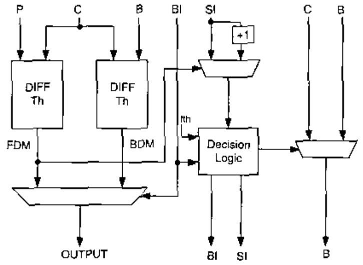

Figure 5: Architecture of CDMBG (change detection mask and background generator).

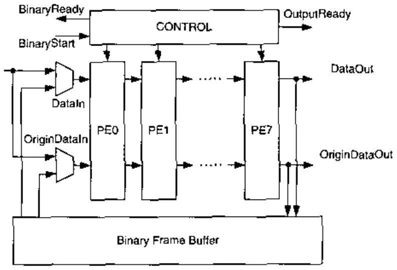

Figure 6: Programmable binary morphology PE array.

In Fig. 6, each PE can be programmed to perform one of the four operations: 3x3 dilation, 3x3 erosion, 3x3 conditional erosion, and no operation. For larger structuring elements, the chain rule of morphological operations is applied as shown below:

$$A \oplus (B \oplus C) = (A \oplus B) \oplus C. \qquad (5)$$

It means cascading two 3x3 dilation is equivalent to a 5x5 dilation. To achieve real-time requirement, eight PEs are required at 26MHz, that is, it has $4^8 = 65,536$ different configurations. If a more complex operation is mapped into the array, a folding technique is applied to decompose the operation into many simple operations, and each simple operation is mapped into the PE array in different time slot. An off-chip memory is used to store the partial results of the PE array.

The detailed architecture of a single PE is shown in Fig. 7. The max operation is replaced with OR operation in the binary morphology PE, and the hardware cost is also minimized with partial-result-reuse concept. The hardware cost of each PE can be further reduced by means of the duality property:

$$(A \ominus B)^c = A^c \oplus \check{B}. \qquad (6)$$

Similar to *GRA*, the delay lines of each PE are implemented with an on-chip SRAM. The vertical boundary control and horizontal boundary control signal is given from the control unit to deal with the boundary condition.

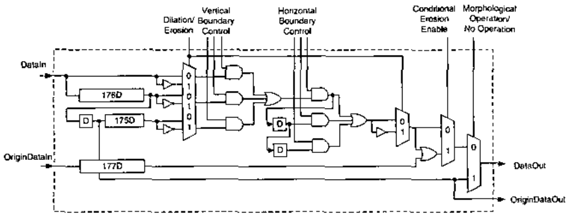

Figure 7: Detailed architecture of a single PE.

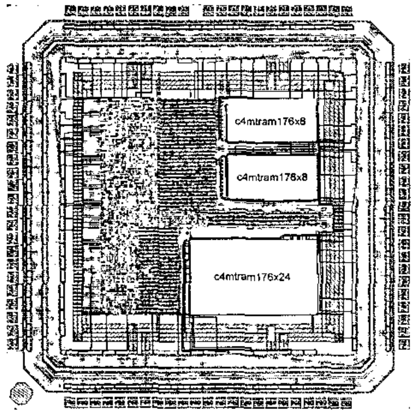

Figure 8: Chip layout of the proposed system.

4. SIMULATION RESULTS

The segmentation chip is designed in cell-based design flow with Avant! standard cell library and Avant! RAM compiler. Two memory built-in-self-test modules are also integrated into the chip to test the 8-bit SRAM and 24-bit SRAM separately.

The video segmentation chip is currently under fabrication by TSMC. The chip layout of the proposed system is shown in Fig. 8. There are three on-chip two-ports SRAM on the chip. Two of them are 176x8, which are used as delay lines in *GRA*. The other one, which is 176x24, is used as delay lines of the programmable PE array. The technology is TSMC 0.35μm 1P4M. The chip size is 2.77x2.77 mm^2 where the transistor count is 143,122. It shows the chip size and transistor count of this system is quite small and can be easily integrated into a single chip MPEG-4 encoding system. The detailed features of the video segmentation chip is shown in Table 1.

Simulation results show that this chip can achieve real-time requirement for QCIF (176x144) video at 26MHz, and the programmable PE array can perform 7,680 binary morphological operations per second, which is sufficient for most situations. Note that although the target frame size of this prototyping chip is only QCIF, it is easy to scale the design into CIF format or even larger video format.

5. CONCLUSIONS

Table 1: Features of the Video Segmentation Chip.

Technology	TSMC 0.35μm 1P4M
Package	100 CQFP (91 Pads)
Chip size	2.77mm×2.77mm
Power supply	3.3V
Power consumption	48.35mW@33MHz
Transistor count	143,122
Processing speed	30 QCIF frames/s
(at 26 MHz)	7,680 morphological operations/s
On-chip memory	2 176×8 two-port RAM
	1 176×24 two-port RAM

A single chip video segmentation system is proposed in this paper. The hardware-oriented video segmentation algorithm can give good segmentation results with only pixel-based local operations and morphological operations, which can be efficiently implemented with partial-result-reuse architecture. To provide both high throughput and flexibility, a programmable PE array architecture is proposed to process the most complex operation of the algorithm. This chip is currently under fabrication with 0.35μm 1P4M technology by TSMC. It shows the chip size is small and can be easily integrated into any content-based coding systems.

6. REFERENCES

[1] *Annex F: preprocessing and postprocessing.* ISO/IEC JTC 1/SC 29/WG11 N4350, 2001.

[2] *The MPEG-4 Video Standard Verification Model version 18.0.* ISO/IEC JTC 1/SC 29/WG11 N3908, 2001.

[3] S.-Y. Chien, S.-Y. Ma, and L.-G. Chen. An efficient video segmentation algorithm for real-time mpeg-4 camera system. In *Proc. of Visual Communication and Image Processing 2000*, pages 1087–1098. SPIE, June 2000.

[4] S.-Y. Chien, S.-Y. Ma, and L.-G. Chen. A partial-result-reuse architec-ture and its design technique for morphological operations. In *Proc. of International Conference on Acoustics, Speech, and Signal Processing 2001*. IEEE, May 2001.

[5] M. Kim, J. G. Choi, D. Kim, H. Lee, M. H. Lee, C. Ahn, and Y.-S. Ho. A vop generation tool: Automatic segmentation of moving objects in image sequences based on spatio-temporal information. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(8):1216–1226, December 1999.

[6] R. Mech and M. Wollborn. A noise robust method for 2d shape estima-tion of moving objects in video sequences considering a moving camera. *Signal Processing*, 66, 1998.

[7] T. Meier and K. N. Ngan. Video segmentation for content-based coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(8):1190–1203, December 1999.

[8] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, London, 1982.

[9] D. Wang. Unsupervised video segmentation based on watersheds and temporal tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):539–546, September 1998.