A HYBRID MORPHOLOGY PROCESSING UNITS ARCHITECTURE FOR REAL-TIME VIDEO SEGMENTATION SYSTEMS

Shao-Yi Chien, Yu-Wen Huang, Shyh-Yih Ma, and Liang-Gee Chen*

DSP/IC Design Lab
Department of Electrical Engineering, National Taiwan University
1, Sec. 4, Roosevelt Road, Taipei 106, Taiwan
lgchen@video.ee.ntu.edu.tw

ABSTRACT

In this paper, we propose a hybrid morphology processing units architecture for real-time video segmentation systems. It contains two parts: a gray-level part and a binary part. A partial-result-reuse technique is applied to reduce the hardware cost of gray-level part. For the target of high throughput and flexibility, the binary part is implemented with a programmable PE array. Simulation shows the proposed hardware architecture is efficient in both hardware complexity and memory organization.

1. INTRODUCTION

Real-time video segmentation, which can generate shape information of video objects in real-time, is an essential preprocessing part of real-time MPEG-4 content-based coding systems [1]. We have proposed an efficient algorithm based on change detection with background registration and shadow cancellation techniques, which can give very good segmentation results and resist the influence of shadow and light change [2]. The computational intensity of this algorithm is much lower than other algorithms [3][4][5]. Besides, the proposed algorithm can be further optimized with bit-parallel and MMX techniques. It can process 25 QCIF frames per second with Pentium III 450 MHz processor when shadow cancellation mode is turned off [6]. However, if the frame size becomes large or the shadow cancellation mode is turned on, the required computation power will become too high to be afforded by general processors. Therefore, hardware implementation of video segmentation is urgently needed for real-time applications.

Mathematical morphology[7][8], which is based on set theory, is widely used for video segmentation algorithms, including the proposed one. The morphological operations are regular and very suitable for hardware implementation. For that reason, if our algorithm can be fully mapped to morphological operations, the hardware implementation will be much easier. Many hardware morphological architectures are proposed[9][10][11], but they can be further optimized.

In this paper, a hybrid morphology processing units architecture for video segmentation systems is proposed, which includes both gray-level part and binary morphological part. The gray-level part cancel shadow effect and generate change detection mask with background registration technique, and the binary morphological part is used for noise region elimination and boundary smoothing. In gray-level part, hardware cost is reduced via partial-result-reuse technique. In binary part, a programmable PE array is used to provide both high throughput and flexibility.

In Section 2, the video segmentation algorithm is first introduced. The detail architecture is shown in Section 3. Section 4 shows the implementation of this architecture and the comparison with other architectures. Finally, Section 5 gives the conclusion.

*This work is supported by SiS Education Foundation.

2. VIDEO SEGMENTATION ALGORITHM

The proposed moving object segmentation algorithm 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. The flow chart of this algorithm is shown in Fig. 1 [2].

In baseline mode, the algorithm has five steps: background registration, change detection, noise region elimination, temporal filter, and post-processing filters. First, the current frame and the previous frame are loaded to perform background registration, which registers reliable background information into background buffer. The reliable background consists of pixels which are not parts of moving objects for consecutive *fth* frames. After background registration, the current frame, previous frame, and background frame are loaded to generate a rough change detection mask *CDMi*. If the background exists, the difference of current frame and background is thresholded to form change detection mask. Otherwise, the difference of current frame and previous frame is used. Next, noise region elimination, which consists of connected component and region filtering, are applied to eliminate small noise regions. Connected component operation[7] labels each connected region a special label and records its area, and then regions with small area will be taken as noise and eliminated. Finally, morphological opening and closing operations are applied to smooth the boundary of segmentation results.

In shadow cancellation mode, morphological gradient filter is first applied to depress the influence of light change and shadow. An extra erosion operation is added in post-processing filters to eliminate the edge-thickening effect of gradient filter. Some segmentation results are shown in Fig. 2.

Operation analysis shows it needs about 700 MOPS to perform a simple morphological operation, such as dilation, in real-time with a RISC. However, lots of complex morphological operations are needed in our algorithm to segment each frame. Consequently, hardware implementation for video segmentation is necessary.

0-7803-6685-9/01/$10.00©2001 IEEE

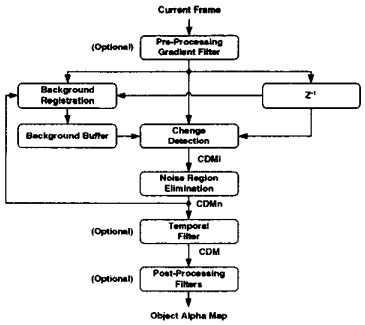

Figure 1: Flow chart of proposed video segmentation algorithm.

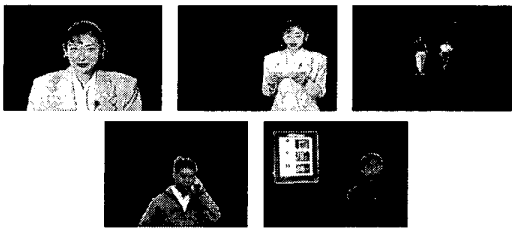

Figure 2: Segmentation results of proposed algorithm for sequence (a)*Akiyo*; (b)*Weather*; (c)*Hall*; (d)*Frank*; (e)*Shaoyi*.

3. HARDWARE ARCHITECTURE

First, all operations of our segmentation algorithm are mapped to morphological operations, which are easier for hardware implementation. Both gray-level and binary morphological operations are needed in our algorithm. They are dramatically different: the word-length of gray-level morphological operations is 8-bits so byte processing units are needed; on the other hand, binary morphological operations need only bit-level operations and can be implemented with logic gates. Therefore, it is more efficient to separate the system into gray-level part and binary part.

The block diagram of this system is shown in Fig. 3. The gray-level part contains two units: *GRA* is morphological gradient filter, and *CDMBG* is change detection and background registration unit. The binary part includes three units: *Programmable Binary Morphology PE Array* takes charge of all kinds of binary morphological operations, *Control* is the control unit of the PE array, and *Binary Frame Buffer* is used to store partial results of the PE array when dealing with complicate binary morphological operations. The detail of this system is presented in following two subsections.

3.1. Gray-level part

There are two units included in the gray-level part: *GRA* and *CDMBG*. The operation of morphological gradient filter *GRA* can be shown as following equations[8]:

Let *I* be the image, *B* be the structuring element (SE), and *GRA* be the output of gradient filter.

$$GRA = I \oplus B - I \ominus B \qquad (1)$$

where

$I \oplus B(x,y) = \max\{I(x-i, y-j) | (x,y) \in I, (i,j) \in B\}$,

$I \ominus B(x,y) = \min\{I(x+i, y+j) | (x,y) \in I, (i,j) \in B\}$,

where $\oplus$ and $\ominus$ are dilation and erosion respectively. If the SE is 3x3 as shown in Fig. 4(a), the dilation result of point (r,c) is the maximum of the nine points in Fig. 4(b), which needs eight comparators to get the result. We found that if the partial results during computation are kept and reused, the required comparators can be further reduced. The concept is shown as following equations:

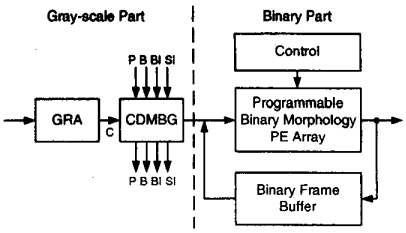

Figure 3: Block diagram of video segmentation system.

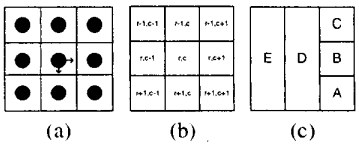

Figure 4: An example of Partial-Result-Reuse.

If the the nine pixels are divide into five groups as shown in Fig. 4(c). Let $A_{i,j} = I(i+1, j+1)$, $B_{i,j} = I(i, j+1)$, $C_{i,j} = I(i-1, j+1)$, $D_{i,j} = \max\{I(i-1,j), I(i,j), I(i+1,j)\}$, $E_{i,j} = \max\{I(i-1,j-1), I(i,j-1), I(i+1,j-1)\}$, and $F_{i,j} = \max\{A_{i,j}, B_{i,j}, C_{i,j}\}$.

$$\begin{aligned} I \oplus B(r,c) &= \max\{A_{r,c}, B_{r,c}, C_{r,c}, D_{r,c}, E_{r,c}\} \\ &= \max\{A_{r,c}, B_{r,c}, C_{r,c}, F_{r,c-1}, D_{r,c-1}\} \end{aligned} \qquad (2)$$

Equation (2) implies that only $I(r-1, c+1)$, $I(r, c+1)$, and $I(r+1, c+1)$ are required for computation, and the value of $D_{r,c}$ and $E_{r,c}$ can be reused from the former partial results $F_{r,c-1}$ and $D_{r,c-1}$ respectively. Besides, if the input signal is in raster scan, an extra delay line in needed. The Partial-Result-Reuse architecture for morphological gradient filter is shown in Fig. 5. Two serial delay lines, whose length is equal to the frame width W, is required, and they can be implemented with registers or memory. On the right side of Fig. 5, the upper part is dilation unit, and the lower part is erosion unit. The corresponding nodes of Equation (2) are marked in the dilation unit of the figure. Note that two registers are needed to store the partial results for each unit. *MAX* outputs the maximum of its two inputs, *MIN* outputs the minimum, and *MAXMIN* outputs the maximum and minimum simultaneously with only one comparator. Note that it may cause some errors at boundaries of the image; however, it won't influence final results since these errors will be eliminated in the binary part.

The *CDMBG* can be implemented directly as shown in Fig. 6. *P* is previous frame, *C* is current frame, *B* is registered background, *BI* is background indicator, which indicates if the background exists, *SI* is stationary index, which records how many consecutive frames each pixel is not in moving objects up to now, *DIFF Th* is the frame difference and thresholding unit, and *Decision Logic* decides if the current input pixel is part of reliable background. If it is, it will be written into background, or the background will be unchanged.

3.2. Binary part

The output of *CDMBG* is the initial segmentation mask, which

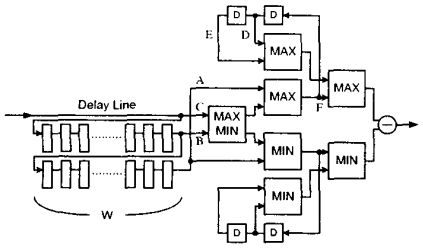

Figure 5: Architecture of morphology gradient filter.

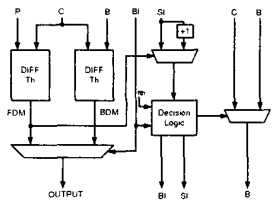

Figure 6: Change detection and background registration unit.

is binary data and can be manipulated with binary processing units. The binary part of the segmentation system contains two functions: noise region elimination and post-processing. In the proposed algorithm, the post-processing is a set of morphological operations, and the noise region elimination is based on connected component [7], which is very hard for hardware implementation. Therefore, it should be mapped to morphological operations first.

We found that the white color noise (salt noise) in *CDMi* is usually small in size, as shown in Fig. 7(a), and can be simply eliminated with morphological opening operation. However, the black color noise is usually too large to be eliminated with closing operation without degrading the precision of object boundaries. Conditional morphological operations[8] can preserve the shape information, and the combination of dilation and conditional erosion (geodesic erosion) can be used for black noise region elimination, which is a kind of reconstruction filter. The procedure can be shown as following equation:

$$(((I \oplus B_n) \underbrace{\ominus B_3; I) \ldots \ominus B_3; I)}_{l} \quad (3)$$

where B_n is nxn structuring element, and B_3 is 3x3 structuring element. The conditional erosion is

$$X \ominus B; Y = (X \ominus B) \cup Y \quad (4)$$

Note the binary dilation and erosion are simply gray-level dilation and erosion with *MAX* and *MIN* replaced by *OR* and *AND*. The frame size of example sequence *Frank* is 320x240. If we choose $n = 15$ and $l = 50$, the result of Equation (3) is shown in Fig. 7(b). After opening operation, the result is shown in Fig. 7(c), which is very similar to the effect of connected component and region filtering. Note that n is proportional to object size and l is related to the shape of object and the frame width.

The post-processing procedure contains an erosion operation to eliminate edge-thickening effect of gradient filter and a close-open operation to smooth the object boundary.

For high speed and high throughput requirements, PE array architecture is suitable for these operations. However, these operations have to be done one after another, and the parameters, such as n and l, depend on sequences and are changed in different situations. Therefore, implementation with simple PE array is very

(a) (b) (c)

Figure 7: Noise region elimination implemented with morphological operations.

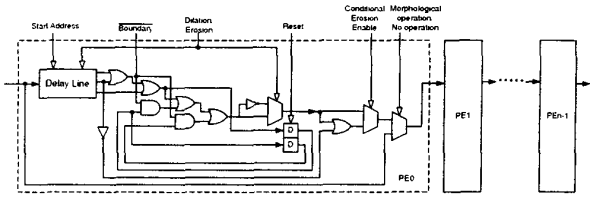

Figure 8: Programmable PE array architecture for binary morphological processing.

hardware-consuming and not flexible enough. A Programmable PE Array architecture is proposed as shown in Fig. 8. Each PE can perform one of the three functions: 3x3 dilation, 3x3 erosion, and 3x3 conditional erosion. The hardware of each PE can be further reduced by means of the duality property[7]:

$$(A \ominus B)^c = A^c \oplus \check{B} \quad (5)$$

For large size SE, the chain rule is applied as shown below:

$$A \oplus (B \oplus C) = (A \oplus B) \oplus C \quad (6)$$

It means cascading two 3x3 dilation is equivalent to 5x5 dilation. Consequently, if the PE array contains n PEs, it can perform 3^n different functions. If the length of the array is not enough, folding technique is applied; therefore, an extra feedback loop with a frame buffer is added in the system. Note that, unlike gradient filter, the errors at boundaries will be propagated in this array. Hence, extra circuits are used to avoid errors in boundary conditions.

4. HARDWARE IMPLEMENTATION

The proposed architecture was implemented and verified with Verilog HDL.The results are shown in following two subsections.

4.1. Gray-level part

A Partial-Result-Reuse architecture is used to implement gray-level morphological gradient operation. The hardware cost of this architecture and other two architectures[10][11] for gray-level gradient filter is shown in Table 1. The proposed architecture has only half hardware cost. It is quite hardware-cost-effective.

If the filter is implemented with RISC or DSP, for CIF sequences, it will require at least two frame memories, which needs about 1.6Mb, to store the partial results, where proposed architecture needs only 5.6kb as shown in Table 3. Besides, the amount of memory access if implemented with RISC is 438Mb per second where it is only 24Mb per second in this architecture. Therefore, the proposed architecture is also memory-effective.

The hardware cost of gray-level part is listed in Table 3.

4.2. Binary part

The gate count of each PE is 42.2 without memory. If 7x7 dilation is considered, 3 PEs are cascaded in the PE array. Note that

Table 1: Comparison between proposed gray-level morphology architecture and other's architecture.

Architecture	Comparator count[a]	Register number	Estimated gate count[b]
K.I. Diamantaras[10][c]	16	6	1168
M.H. Sheu[11][d]	10	16	1514
This work (Partial-Result-Reuse architecture)	7	4	599

[a]two-input comparator
[b]comparator:49gates, 8-bits register:64gates
[c]Parallel version, 1 PE
[d]Ignore the adders/subtractors

Table 2: Comparison between proposed binary morphology architecture and other's architecture when 7x7 dilation is considered.

Architecture	Gate count[a]	Required cycles per frame[b]	Number of different configuration
K.I. Diamantaras[10]	384	101376	1
E.N. Malamas[9]	5075[c]	50688	4
This work (Programmable PE array with 3 PEs)	127.2	101376	27

[a]ignore memory
[b]Assume the pipeline is full
[c]Ignore output networks of OR logic

the architecture in [10] is modified for binary situations with replacing comparators with logic gates in these comparisons. In Table 2, compared with systolic array architecture[10] and Erosion-Dilation Architecuter [9], the hardware cost of the proposed architecture is much less, and it is also more flexible than others. The proposed architecture has 27 different configuration; the EDA[9] has 4 different configuration: dilation, erosion, opening, and closing; the systolic array architecture[10] can't be programmed so it has only one configuration. The proposed hardware is more cost-effective and flexible. Note the EDA can perform two basic morphological operations at the same time so only half of cycles per frame are needed; however, the hardware cost is enormous.

If target clock rate is 30MHz, 21 PEs are required to achieve real-time requirement (30 CIF frames per second). The gate count is 932.4, and 14784b memory is in demand as shown in Table 3. It can perform about 6000 3x3 morphological operations per second. The amount of memory access will be 5.5Gb per second if RISC is used in this speed, where only 29Mb is needed in this architecture. Besides, the RISC needs at least two frame memories, 203kb, where only 15kb internal memory is needed in this architecture. Hence, the proposed architecture is memory-effective.

Note that the memory parts are not included since they are not clearly described in these architectures. The memory requirement is the minimum in our architecture if the inputted data is one channel in raster scan.

5. CONCLUSION

A hybrid morphology processing units architecture for real-time video segmentation systems is proposed in this paper. The proposed video segmentation algorithm is fully mapped to morphological operations, which are very suitable for hardware implementation. The whole system is implemented with both gray-level morphology processing unit and binary morphology processing unit. Compared with existing architectures for mathematical morphology, the proposed architectures is very efficient in both hardware cost and memory organization.

Table 3: Result of hardware implementation.

Unit	Gate count	Internal memory size
GRA	981.98	5632b
CDMBG	254.23	0b
Binary morphology PE array (21 PEs)	932.4	14784b
Total	2168.61	20416b

6. REFERENCES

[1] T. Sikora, "The MPEG-4 Video Standard Verification Model," *IEEE Trans. On Circuit and Systems for VideoTechnology*, vol. 7, no. 1,pp.19-31, February 1997.

[2] 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*, pp.1087-1098, 2000.

[3] MPEG Video Group, "Annex F: Preprocessing and Postprocessing," ISO/IEC JTC 1/SC 29/WG11 N3056.

[4] D. Wang, "Unsupervised Video Segmentation Based on Watersheds and Temporal Tracking," *IEEE Trans. On Circuit and Systems for Video Technology*, vol. 8, no. 5, pp.539-546, September 1998.

[5] R. Mech and M. Wollborn, "A Noise Robust Method for 2D Shape Estimation of Moving Objects in Video Sequences Considering a moving camera," *Signal Processing*, vol. 66, pp.203-217, 1998.

[6] S.-Y. Chien, Y.-W. Huang, S.-Y. Ma, and L.-G. Chen, "Efficient Video Segmentation on SIMD Architecture for Real-time MPEG-4 Systems," in *Proc. of Workshop on Consumer Electronics 2000*, 2000.

[7] R.M. Haralick, L.G. Shapiro, *Computer and Robot Vision*, vol. 1, 1992.

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

[9] E.N. Malamas, A.G. Malamos, and T.A. Varvarigou, "Fast implementation of binary morphological operations on hardware-efficient systolic architectures," *Journal of VLSI Signal Processing*, vol. 25, pp.79-93, 2000.

[10] K.I. Diamantaras and S.Y. Kung, "A linear systolic array for real-time morphological image processing," *Journal of VLSI Signal Processing*, vol. 17, pp.43-57, 1997.

[11] M.-H. Sheu, J.-F. Wang, J.-S. Chen, A.-N. Suen, Y.-L. Jeang, and J.-Y. Lee, "A data-reuse architecture for gray-scale morphologic operations," *IEEE Trans. on Circuits and Systems–II Analog and Digital Signal Processing*, vol. 39, no. 10, pp.753-756, October, 1992.