Unsupervised spatiotemporal segmentation algorithm for on-chip video systems

Chi-Chou Kao
National Pingtung Institute of Commerce
Department of Information Technology
51 Min-Sheng E. Road
Pingtung, Taiwan
E-mail: cckao@casdc.ee.ncku.edu.tw

Chuen-Yau Chen
National Yunlin University of Science and Technology
Department of Electrical Engineering
University Road, Section 3
Yunlin, Taiwan
E-mail: cychan@ieee.org

Yen-Tai Kao
National Cheng Kung University
Department of Electrical Engineering
No. 1, Ta-Hsueh Road
Tainan, Taiwan
E-mail: cckao@casdc.ee.ncku.edu.tw

1 Introduction

In the past, the video standards, such as MPEG-1 and MPEG-2, used frame-based coding algorithms in various multimedia applications. However, they did not have good flexibility in access and manipulation of objects. To provide efficient coding performance, the MPEG-4 multimedia communication standard enables content-based functionalities by using the video object plane (VOP), which includes the shape and texture information of a semantically meaningful object in the scene as the basic coding element. Using such functionality, plain video sequences can be decomposed into several objects for recomposing with other objects or be transmitted through a network separately to increase the efficiency of bandwidth usage.

Segmentation can be defined as the operation of partitioning a scene into regions extracted according to a given criterion. Image segmentations have included extraction of features such as edges and curves and integration of these features into continuous shapes that are spatially coherent. Video segmentations, in addition to those techniques used for image segmentation, have included temporal change detection due to motion of individual objects in a temporally coherent manner. Hence, an efficient segmentation algorithm is more necessary in video than in image processing. Moreover, many multimedia application systems have been implemented on chips to achieve high-speed processing for video segmentation.

According to the given segmentation criteria, conventional segmentation algorithms can be divided into two categories: (1) spatial segmentation algorithms, whose criteria are spatial homogeneity, and (2) temporal segmentation algorithms, whose criteria are change detection. A spatial segmentation algorithm consists of three steps. In the first step, morphological filters are used for image simplification. The second step approximates the temporal gradient by a morphological gradient operator, which is also treated as an input to a watershed algorithm for identifying the regions with homogeneous intensity. The third step merges the regions, usually oversegmented, in the watershed algorithm. The algorithms focus on finding more precise object boundaries. However, because both the watershed algorithm and the region merging are computation-intensive operations, the computational complexity is very high. For temporal segmentation, the algorithms consist of two steps. The first step consists in estimating a change detection mask; the second step consists in generating an object mask by eliminating the uncovered background from the change detection mask in the first step.

We believe that the temporal segmentation algorithms are more efficient than the spatial segmentation algorithms because they distinguish a moving object from the background. However, the quality of the segmentation will not be satisfactory if the speed of the object changes significantly in the sequence. Moreover, the motion estimation is a time-consuming operation, and the whole processing speed is significantly decreased in turn.

In this paper, we propose spatiotemporal algorithms to design an efficient segmentation system that is suitable for implementing on a chip and for any object motion. Good performance of the proposed algorithms is demonstrated by the simulation results. The rest of this paper is organized as follows. An overview of the proposed system used to ex-
tract the moving objects is given in Sec. 2. Section 3 describes in detail the spatiotemporal segmentation algorithms. The experimental results are shown in Sec. 4. Finally, Sec. 5 gives conclusions.

2 The System Overview

The basic idea of the proposed system is to use simultaneously the spatial and the temporal segmentation algorithms based on the accumulative information technique to produce the expected segmentation masks. In contrast with other segmentation systems,9–15 our judgment criterion for motion is not based on the difference between two adjacent frames. Instead, we process a sequence of frames to compute the differences among the successive frames. Using two consecutive frames to establish a correspondence of moving objects between frames is the most common method. However, the resulting motion estimations are not accurate if the object moves on a noiseless or self-colored background. It is obvious that the more the moving frames there are, the larger is the difference between the moving object and the background. Moreover, in general, noise cannot remain in the same position for a sequence of moving frames, so that any pixel that changes significantly from frame to frame can be treated as noise and removed from the object region. Therefore, we strongly believe that processing a sequence of frames can produce more accurate segmentation masks than processing just two consecutive frames. Figure 1 illustrates the effect of computation with different numbers of frames on the performance of the proposed system.

The proposed system is divided into four main steps—temporal segmentation, spatial segmentation, object detection, and contour smoothing—as shown in Fig. 2. The spatial segmentation and the temporal algorithms are first used in parallel to produce the edge tracks called the edge masks and the moving object regions called the moving block masks, respectively. Next, we combine these masks to obtain the initial object masks. Accordingly, the number of time-consuming motion estimations can be reduced, and the whole processing speed is significantly accelerated in turn. Hence, although we process more frames in this system than in other systems,9–15 good object masks can still

![Fig. 1 The effect on the computation of using sequences of (a) two frames, (b) six frames, (c) eleven frames, and (d) twenty-six frames.](image)

![Fig. 2 Diagram of the proposed video segmentation system.](image)
be acquired without increasing the computational complexity. Such a system is suitable for implementing on a chip. Finally, in the contour-smoothing step, the initial object mask is postprocessed to remove small residual background regions so that the final object masks are good.

3 A Spatiotemporal Segmentation Algorithm

Recall that the proposed system is divided into four main steps: temporal segmentation, spatial segmentation, object detection, and contour smoothing. In this section, we describe in detail the segmentation algorithms of those steps.

3.1 Temporal Segmentation Step

The objective of the temporal segmentation step is to analyze the motion relationships among the given video sequence frames and generate the moving block masks. For this purpose, we use an unsupervised segmentation method in which human assistance is unnecessary. In this method, because the initial moving block masks approximate to the final moving block masks, the number of motion estimations can be greatly reduced. It is suitable for implementing the whole system in hardware. Figure 3 illustrates an unsupervised segmentation scheme.

When we compute the difference of blocks to find the moving block masks, a large block size is chosen that will cause small noise flutters in luminosity in the selected moving block mask. Moreover, the signal-to-noise ratio will be too small to distinguish the objects from the background. However, although using small block size efficiently removes most of the random noises, it will need a large number of computations. According to the experimental results shown in Fig. 4, a 4×4 block size is the best choice for a 64×64 frame. In other words, if the block size is selected as 1/16 of the original frame size, the proposed system achieves the best video quality.

The accumulative information technique is used to extract the moving block masks. Let \( n \) be a suitable number of frames that are processed in sequence to find the moving block masks. Let \( T_1 \) and \( T_2 \) denote the threshold values for determining whether a block belongs to the moving block mask or the background. Note that \( n, T_1, \) and \( T_2 \) are treated as the inputs of the chip and can be dynamically adjusted by designers to improve the video quality. We compare every frame with the first frame. If the difference \( D_k(i,j) \) of the block at the same position \( (i,j) \) between the first frame \( f_1(i,j) \) and the \( k \)'th frame \( f_k(i,j) \) is more than \( T_1 \), the block value \( B_k(i,j) \) is accumulated into the corresponding memory \( M(i,j) \), where \( 2 \leq k \leq n \). After all frames are processed, if \( M(i,j) \) is more than \( T_2 \), the corresponding block is selected to be part of the moving block mask, \( MB(i,j) \).

The procedures can be described by the following equations:

\[
D_k(i,j) = \sum_{i=1}^{4} \sum_{j=1}^{4} |f_1(i,j) - f_k(i,j)|, \quad 2 \leq k \leq n, \tag{1}
\]

Fig. 5 Generation of the moving block masks: (a) the initial moving block masks, (b) effect of noise-region elimination, and (c) the final moving block masks.

Fig. 4 Block size comparison: (a) 2×2 block size, (b) 4×4 block size, and (c) 8×8 block size.

Fig. 6 Noise removal: (a) basic block element expression and (b) scan procedure.
The full directional Laplacian edge operator.

\[
B_k(i,j) = \begin{cases} 
1 & \text{if } D_k(i,j) > T_1, \\
0 & \text{otherwise},
\end{cases} \quad 2 \leq k \leq n, \tag{2}
\]

\[
M(i,j) = \sum_{k=2}^{n} B_k(i,j), \quad 2 \leq k \leq n, \tag{3}
\]

\[
MB(i,j) = \begin{cases} 
1 & \text{if } M(i,j) > T_2, \\
0 & \text{otherwise}.
\end{cases} \tag{4}
\]

After extracting the initial moving block masks as shown in Fig. 5(a), there is still some tiny noise due to the camera noise or irregular object motion. To refine the quality of the moving block masks, we need a method to eliminate noise. The traditional method is to use morphological filtering operations. However, the need to structure the element shape and size will lead to a high time complexity.

A 3x3 block matrix is called a basic block element if we have found the largest value in the matrix and filled it into all the nonzero blocks of the matrix. Figure 6(a) shows a diagram of a basic block element. We select the first nine blocks in the initial moving block as the first 3x3 block. Let the blocks in the third row of the preceding 3x3 block be the blocks in the first row of the next 3x3 block. Accordingly, we set all 3x3 blocks for every fourth column in the initial moving block as shown in Fig. 6(b). Next, the first cycle is performed to set all 3x3 block matrices as the basic block elements by scanning the initial moving block from top to bottom and row by row. When the first cycle is

![Fig. 9](image-url) The edge masks (a) using the traditional Laplacian operator, and (b) using the proposed spatial segmentation algorithm.
finished, the local maximal value is propagated downward. Finally, the blocks located in the middle of the central blocks of the existing $3 \times 3$ blocks are selected as the central blocks of another set of $3 \times 3$ blocks. We perform the second cycle by scanning from bottom to top to distribute the largest value to other connected points, as shown in Fig. 6.

The pixel densities of the regions that have large noise are much less than the pixel densities of the other regions in the initial moving block. For example, in Fig. 5(a), the pixel density of region $A$ is less than that of region $B$. After scanning, there is a cluster of pixels that has a larger density than elsewhere. Generally, the speckles form small groups whose values are less than 50. Therefore, we set blocks whose values are smaller than 50 to zero and the others to one for the final moving block. Figure 5(b) shows the effect of noise-region elimination on Fig. 5(a), and the final moving block masks are shown in Fig. 5(c).

### 3.2 Spatial Segmentation Step

In this subsection, an efficient spatial segmentation algorithm including color degradation and Laplacian operator steps is proposed to generate the edge masks.

The common edge-detection technique is the computation of a local derivative operator such as the Laplacian operator, which is a second derivative. However, the Laplacian operator is time-consuming because of problems in edge direction detection such as noise sensitivity and double edge generation. To reduce the computational complexity, we execute a color degradation step in which every pixel is quantized by shifting pixel intensity bits. The Laplacian operator is improved by detecting full direction changes to solve the edge direction problem. Figure 7 shows the full directional Laplacian edge operator. The method not only detects edges reliably but also simplifies the computation procedures so that it is suitable for hardware implementation.

Figure 8(a) is the original histogram of a frame of the video “Akiyo.” Although it contains a wide spread of values, as would be expected, the computation complexity will be large for detecting edges. Obviously, the more shifted bits there are, the smaller the number of regions and the computation complexity are. However, if the number of shifted bits is too large, the resolution is reduced so that the quality of the images is not good. According to experimental results on many video frames, shifting 4 to 5 bits is the best choice. Figure 8(b) and 8(d) illustrate the selection of the number of shifted pixel intensity bits.

Figure 9(b) shows the edge masks generated by the proposed spatial segmentation algorithm. By comparison with the edge masks produced by using only the traditional Laplacian operator as shown Fig. 9(a), it is seen that the proposed method is superior to the traditional one.

### 3.3 Object Detection Step

An object mask is the bound on the variation of every frame within a short period. We find the conjunction edge masks by mapping the moving block masks to the edge masks for every frame to produce the object masks. Figure 10 Generation of the initial object masks: (a) moving block masks, (b) edge masks, (c) mapping moving block masks to edge masks, and (d) initial object masks.

![Fig. 10](image1)

![Fig. 11](image2)

![Fig. 12](image3)

![Fig. 13](image4)
10 illustrates the procedure. It is obvious that the boundary of the object masks is very rough. We present a method to smooth the contours in the next subsection.

### 3.4 Contour-Smoothing Step

In this step, spurs are removed from a mask. To improve the quality, we calibrate line segments with multiple reference values and execute depth and breadth simultaneously.

The contour curves of a mask are divided into four: the left, right, upper, and lower. For each contour curve, if the distance from a nonspur point to any point \( x \) is less than a default value, then \( x \) is viewed as a nonspur point. Otherwise, we use the interpolation method to judge whether a point is a spur or not. If any point is located out of the interpolation regions that is computed from existing nonspur points, that point is viewed as a spur.

Figure 11 illustrates the procedure. Suppose the points in region \( A \) are nonspur points and the distances from region \( A \) to region \( B \) and from region \( B \) to region \( D \) are less than the default value. Hence, all points in \( B \) and \( D \) are viewed as nonspur points. The region \( C \) needs to be further judged by using the interpolation method. In this step, the interpolation method is the main determinative factor that is used to judge whether a point is a spur or not. Hence, the default value is set generally as the distance from the center of the object mask to its nearest boundary point.

The traditional interpolation method, as shown in Fig. 12, needs division and precision operations, which are not suitable for hardware implementation. To overcome this problem, we propose a method in which values do not need to be converted to floating-point numbers.

Let \((x_{\text{head}},y_{\text{head}})\) and \((x_{\text{tail}},y_{\text{tail}})\) be the locations of two known nonspur points. If \((x_{\text{head}},y_{\text{head}})\) and \((x_{\text{tail}},y_{\text{tail}})\) are both odd or both even, a new nonspur location can be located by shifting to \((x_{\text{head}}+x_{\text{tail}}+1,y_{\text{head}}+y_{\text{tail}})\); otherwise, it is located by shifting to \((x_{\text{head}},y_{\text{head}}+y_{\text{tail}})(x_{\text{tail}},y_{\text{tail}})\). This step is executed recursively by changing \((x_{\text{head}},y_{\text{head}})\) and \((x_{\text{tail}},y_{\text{tail}})\) until interpolation lines are found. Any point that is located out of the interpolation region is viewed as a spur. For example, in Fig. 13, \((x_3,y_3)\) can be determined according to \((x_1,y_1)\) and \((x_2,y_2)\), which are the initial known nonspur locations. The new nonspur locations, \((x_4,y_4)\) and \((x_5,y_5)\), can be found respectively by computing the pairs of nonspur locations \((x_1,y_1)(x_3,y_3)\) and \((x_3,y_3)(x_2,y_2)\). The nonspur locations form an interpolation region to judge whether a point is a spur or not.

The effect of contour smoothing is shown in Fig. 14. It can be seen that spurs are removed as required but that part of the region that is not required to be deleted has been. However, the algorithm removes spurs on convex objects.

For an initial object mask, we execute depth and breadth contour smoothing simultaneously to generate the final object mask. Figure 15 illustrates the procedure. Using the final object mask, the video object plane (VOP) can be extracted as shown in Fig. 16.

### 4 Experimental Results

We have simulated the proposed algorithm on the standard MPEG-4 test sequences to evaluate its objective qualities. Figure 17 shows the segmentation results for several benchmark sequences taken from “Akiyo” and “Grandma.” The “Akiyo” sequences do not have background noise, so their segmentation results are better than those of “Grandma.” In the “Grandma” sequences, shadows of the sofa appear in the background region. Compared with the previous results, our segmentation results are quite good. Fast-
moving or low-resolution benchmark sequences are more difficult to segment because the resolution of this specimen becomes low, so that several background pixels are mixed with foreground objects. However, the moving objects can still be extracted by our method, as shown in Fig. 18.

We use a personal computer with a Pentium-III 800-MHz processor as the test platform to implement the proposed algorithm, shadow cancellation (SC) mode, and global motion compensation (GMC) mode. The test sequences are in QCIF format. In this work, a processing speed of 30 QCIF frames per second, meeting the real-time requirement, can be achieved. However, the processing speed is slowed down to 21 QCIF frames per second in SC mode, and 10 QCIF frames per second in GMC mode. In GMC mode, the processing speed is 10 QCIF frames per second, meeting the real-time requirement, can be achieved. However, the processing speed is slowed down to 21 QCIF frames per second in SC mode, and 10 QCIF frames per second in GMC mode. In GMC mode, the processing speed is 10 QCIF frames per second. These experimental results reveal that the proposed algorithm is suitable for chip implementation.

5 Conclusions

In this paper, we have proposed an algorithm making use of accumulative information of multiple frames for moving object segmentation. It is divided into four main steps: temporal segmentation, spatial segmentation, object detection, and contour smoothing. In every step, we present many techniques to improve the performance of the traditional methods so that the property of the final video object planes is suitable for implementing on a chip. We believe that this algorithm can be realized easily in hardware with proper data architecture.

References


Chi-Chou Kao received a BS in engineering science from National Cheng-Kung University, Taiwan, in 1990, and MS and PhD degrees in electrical engineering from National Cheng-Kung University, Taiwan, in 1996 and 2002, respectively. Dr. Kao received the Long-Term Distinguished Paper Award, Acer Foundation, in 2003, and appears in the 23rd (2006) edition of Who’s Who in the World. His research interests include graph algorithms, combinatorial optimization, and circuit design. He is currently an assistant professor in the Department of Information Technology, National Pingtung Institute of Commerce, Taiwan.

Chuen-Yau Chen received BS and PhD degrees in electrical engineering from National Cheng Kung University (NCKU), Tainan, Taiwan, in 1995 and 2001, respectively. From 1999 to 2000, he was the system administrator of the VLSI/CAD Laboratory at the Department of Electrical Engineering, NCKU. From 2001 to 2003, he was with the Department of Electronic Engineering, I-Shou University, Kaohsiung, Taiwan. He is currently an assistant professor with the Department of Electrical Engineering, National Yunlin University of Science and Technology, Taiwan. His current research interests are SOC system integration and verification, VLSI design for signal processing systems, and current-mode/mixed-signal circuit design. Dr. Chen is a member of Phi Tau Phi Honorary Scholastic Society, Taiwan Integrated Circuit Design Society, Digital IP Consortium, and SOC Consortium—VLSI Educational Program of Ministry of Education. He served as a member of the technical program committee of the R.O.C. VLSI Design/CAD Symposium in 2002. He also served as the finance chair of the 2004 IEEE Asia-Pacific Conference on Circuits and Systems.

Yen-Tai Lai received a BS from the National Taiwan Ocean University in 1975, an MS from National Cheng-Kung University, Taiwan, in 1980, both in electrical engineering, and a PhD in electrical engineering and computer science from the University of Illinois, Chicago, in 1984. After graduation, he worked at RCA, Solid State Division, Somerville, New Jersey, from 1985 to 1988. He is currently a professor in the Department of Electrical Engineering, National Cheng-Kung University, Taiwan.