Automatic extraction of moving objects for head–shoulder video sequence

Chung Ming Kuo,* Chaur Heh Hsieh, and Yong Ren Huang

Department of Information Engineering, I-Shou University, Tashu, 840, Kaohsiung, Taiwan, ROC

Received 30 October 2003; accepted 8 January 2004
Available online 27 February 2004

Abstract

Recently, video object extraction has received great attention because it is a critical technique in object-based video processing. This paper presents a temporal-to-spatial segmentation technique to extract object from a video sequence. The temporal phase employs a simple blockwise temporal-activity measure to approximately locate the object boundary. And then a block-based maximum a posteriori (MAP) scheme, which exploits spatial features of image blocks around the approximated boundary, is adopted to refine the temporal segmentation result. The proposed technique achieves good segmentation quality with very low computational cost for head-and-shoulder sequences with static background.

Keywords: Object-based; Video object; Segmentation; Temporal-activity; MAP

1. Introduction

In the past few years, the idea of object-based video coding has been proposed to achieve high compression efficiency and to allow for more multimedia functionalities. Recently, a new object-based coding standard, MPEG-4 (Chiariglione, 1997; ISO/IEC, 1998; Koenen et al., 1997; MPEG Video Group, 2001), has been released for multimedia applications. MPEG-4 standard provides users a new level of interaction with visual contents. It offers a framework to view, access and manipulate objects rather than pixels, with great error robustness at a large range of bit rates. Obviously, to achieve the object-based video coding, the video sequences are needed

*Corresponding author. Fax: 886-7-6578944.
E-mail address: kuocm@isu.edu.tw (C.M. Kuo).
to decompose into individual objects or so called video object plane (VOP), and each VOP is then coded independently. The VOP extraction is a crucial issue for the object-based video coding (Alatan et al., 1998; Salembier and Marquès, 1999).

The extraction of video objects may be done by the existing video segmentation techniques, which are categorized into temporal-to-spatial (Diehl, 1991; Hotter, 1990; Musmann et al., 1989) and spatial-to-temporal approaches (Moscheni et al., 1998; Paragios and Triritas, 1999; Park and Lee, 1996; Salembier et al., 1995; Wang and Adelson, 1994; Won, 1998; Won and Park, 1997). The former sequentially extracts objects by iteratively determining the successive dominant motion parameters. The regions or pixels, which conform to the dominant motion parameters, are assumed to comprise one object. The other regions or pixels are regarded as undetermined ones. The process continues to estimate the subsequent dominant motion parameters for the undetermined objects. Finally, the spatial feature is used to refine the segmentation results. For an alternative approach, spatial-to-temporal, an over-segmented image is first obtained using spatial features of regions (Cortez et al., 1995; Salembier et al., 1996; Vincent and Soille, 1991), and then a region-merging procedure to identify meaningful objects is adopted using temporal information such as motion parameters. The algorithms proposed in the literatures (Moscheni et al., 1998; Paragios and Triritas, 1999; Park and Lee, 1996; Salembier et al., 1995; Wang and Adelson, 1994) usually contain three steps: (a) an initial region is generated by using the spatial characteristics, (b) temporal motion information is estimated and used for calculating a similarity measure, and (c) an object is extracted by merging similar regions according to a spatial–temporal similarity measure.

The above two approaches belong to automatic segmentation approach. However, the segmented regions are often not meaningful; e.g., a human head may be partitioned into more than two regions. Thus, in fact they cannot be regarded as VOP extraction techniques. The other common drawback of the two approaches is that many manually tuned thresholds are defined in the algorithms. The tuning of threshold values is difficult and inefficient because threshold values are always image sensitive. Furthermore, significant computation is required for both approaches.

A video object represents a meaningful entity in the world, such as a ball, a building, a human body, an aircraft, etc. (Gu and Lee, 1998). It might not be visually homogeneous in any physical sense. Video segmentation is to get a partition consisting of homogeneous spatial/temporal or spatial–temporal regions according to any given visual homogeneity criterion. Therefore, to achieve object extraction successfully, complex a priori knowledge should be included in the video segmentation. This may increase the system complexity and computational load significantly. Even so, to extract VOP accurately and reliably in an automatic manner for general video sequences is still a big challenge (Meier and Ngan, 1998). Semiautomatic methods that get some input from humans (Choi et al., 1998), e.g., by drawing initial boundary of object (Gu and Lee, 1998; Herrmann et al., 1999) were presented to alleviate the problem. However, these approaches may not be suitable for object-based video coding due to the need of user interaction.

This paper aims to develop a computationally efficient technique that can automatically extract a VOP without user interaction. To reduce the problem to a more
manageable size, a head-and-shoulder sequence with either static background or moving background after global motion compensation is considered. In addition, a moving object is defined as a *spatially-connected moving area*, which may contain several visually non-homogeneous regions. For example, a salesman takes a product in his hand (e.g., the Salesman sequence in our experiment). Although the product and the hand are different from human vision, we treat them as a single object because the product is spatially connected with the hand and they have similar motion characteristics. The definition is suitable for object-based coding, especially in video-phone or videoconferencing applications.

In this work, we first define a temporal-activity measure based on the variance of the difference of two successive frames. Using the temporal-activity measure, we separate the background and non-background regions, and bound non-background regions as close to object as possible. And then a spatial segmentation scheme using a block-based maximum a posteriori (MAP) criterion is developed to further refine the temporal segmentation result. The main idea behind the proposed technique is that it locates the rough position of silhouette of a moving object based on the temporal-activity measure, and then refines the silhouette accuracy using the MAP scheme. It is noted that the MAP scheme operates on the near-boundary region of an object. This is very different from the conventional video segmentation techniques such as (Won, 1998) that operate on the entire region of an object. In other words, the scheme focuses on the identification of the *silhouette* of objects rather than the analysis of the contents of objects. It has the advantages of simplicity, reasonable computation and high accuracy.

The paper is organized as follows. In Section II, we first describe the proposed temporal segmentation procedure. Then, a block-based MAP spatial segmentation method is described in detail. In Section III, the simulation results are given. Finally, the conclusion is drawn in Section IV.

2. Block-based temporal-spatial silhouette segmentation

The proposed technique for object extraction consists of temporal and spatial segmentation phases, in which the temporal phase performs coarse segmentation and then spatial phase refines the result. The temporal phase aims at locating the exterior silhouette (boundary) of a moving object. The exterior boundary should contain strictly all moving object pixels, but may contain some background pixels. The block-based MAP classification that exploits spatial characteristics is then used to pare off the background pixels (blocks) from the exterior boundary region. The proposed technique is described in the following.

2.1. Temporal segmentation

We first generate a regular non-uniform (RN) mesh in the previous frame using the method presented in Kuo et al. (1999). The RN mesh partitions an image frame into many non-overlapped rectangular blocks with different sizes of
32 × 32, 32 × 16, . . . , 8 × 8, as shown in Fig. 1. The partition is achieved from the local (block) variance of the difference of two successive frames (referred to as frame difference $FD$). Thus, the resulting block size is small in moving object region and large in static background. The RN mesh is simpler and more regular than the partition schemes using local motion activity, such as quadtree. The details can be found in Kuo et al. (1999). The frame difference $FD$ is also used in the proposed temporal segmentation.

The temporal segmentation we proposed consists of three steps as follows:
1. Moving blocks detection
2. Hole elimination
3. Refinement by increasing resolution on boundary.

We first define the temporal-activity measure of the block $i$ in a frame as

$$TA_i = \frac{(k \times V_i + V_{FD})}{(V_i + k \times V_{FD})},$$

where $V_{FD}$ is the variance of the frame difference $FD$, $V_i$ is the block variance calculated from $FD$, and $k$ is an integer greater than one. The above equation indicates that the larger the variance of a block, as compared to the entire variance $V_{FD}$, the greater its temporal-activity can be computed. The parameter $k$ is a weighting factor used to adjust the sensitivity of the block variance $V_i$.

Fig. 1. Variable block-size structure in RN mesh.
After temporal-activity of all blocks in FD are computed, we then calculate their average as

\[ TA_{\text{avg}} = \frac{1}{N_B} \sum_i TA_i, \]

where \( N_B \) is the total number of blocks in a frame. For each block, we compare its motion activity \( TA_i \) with the threshold \( TA_{\text{avg}} \). If \( TA_i > TA_{\text{avg}} \), the block is marked as moving. Fig. 2A shows the result for “Claire” sequence.

The above process detects the moving blocks with significant temporal-activity. Our experience indicates that it identifies most of moving blocks, but often misses the blocks that cover both static part and moving part. Therefore, further detection is necessary for the remaining blocks (the blocks unmarked in the first pass). Using the same concept, we calculate a block temporal-activity for the remaining region by

\[ TA'_i = \frac{(k \times V'_i + V_{\text{res}})}{(V'_i + k \times V_{\text{res}})}, \]

where \( V_{\text{res}} \) is the variance of the remaining regions, and \( V'_i \) is the variance of the block \( i \) belonging to the remaining regions. Again, calculating the average of temporal-activity over the remaining blocks and then applying the same procedure as in the first pass, we can detect the moving blocks from the remaining region. Fig. 2B demonstrates the results of the first two passes.

Our work aims to extract a complete object, e.g., a human body. Even if an object is moving globally, some portions of the object cannot be identified because of its uniform gray level. Therefore, after the above detection, there may exist some unmarked blocks in the object. Such a region covering one or more unmarked blocks surrounded by marked blocks is defined to be a hole. The exception is on the image border, in which the hole can be the region that is surrounded by marked blocks and borders. If a hole is found, it is marked as a moving region. Fig. 2C shows the result after the hole-elimination process.

From Fig. 2C, we can find that some portions along the boundary contain many blocks that are too coarse to definitely represent the boundary of real objects. Therefore, further refinement with higher resolution for the coarse portions is necessary. Along the segmented boundary in the temporal phase, as shown in Fig. 2C, we cal-
culate the variance of the boundary region, denoted as $V_{BR}$. Then, we subdivide the boundary block into $8 \times 8$ sub-blocks, as shown in Fig. 3, and calculate the variance of each sub-block, denoted as $V_S$. If $V_S < (V_{BR} + W_{\text{bias}})$, the sub-block $i$ is regarded as non-moving, and eliminated from the boundary of the temporal segmentation. $W_{\text{bias}}$ is a bias used for the compensation of different sub-block sizes, which is determined experimentally as

$$W_{\text{bias}} = \begin{cases} 1.5 \text{ Sub-block size } = 16 \\ 1.0 \text{ Sub-block size } = 8 \\ 0.5 \text{ Sub-block size } = 4 \\ 0.1 \text{ Sub-block size } = 2 \end{cases}$$

Repeating the procedure for all sub-block’s along the boundary until the non-moving sub-blocks are all removed. We can further divide the sub-block into $4 \times 4$, and then $2 \times 2$, then more compact boundary can be obtained consequently, see Fig. 3. Fig. 4 shows the temporal segmentation result for the subblock size of $2 \times 2$.

The temporal segmentation method presented here is significantly different from the conventional techniques with three advantageous features:

1. The temporal segmentation locates the rough boundary of a VOP. It obtains the exterior boundary of an object that covers all pixels of the VOP, and may contain some background pixels. The temporal phase separates the pure background region that contains no object pixels from the remaining non-background region that contains all object pixels and possibly some background pixels. The background pixels (actually blocks in our work) in the non-background region are further removed with spatial segmentation scheme, which will be described later.

Fig. 3. The sub-division of boundary blocks. (A) Sub-block size: $16 \times 16$, (B) sub-block size: $8 \times 8$, (C) sub-block size: $4 \times 4$, (D) sub-block size: $2 \times 2$. 
2. It introduces a simple temporal-activity measure, which is far more computationally efficient than the motion parameters used in the conventional techniques.
3. Most of comparison thresholds are calculated from image data rather than manually setting. Thus, it is more practical for real application.

2.2. Silhouette spatial segmentation using MAP estimation

The temporal segmentation phase obtains only an approximation of a VOP boundary. To improve the accuracy of the boundary, a spatial segmentation scheme using block-based MAP is developed. This scheme is basically an adaptation of Won (1998), which was originally designed for segmentation of still images. More specifically, in Won (1998), the MAP operates on all blocks of an entire image; i.e., it addresses the contents of an image. However, the spatial segmentation in this work deals with only blocks adjacent to the boundary of the temporal segmentation result. In other words, it addresses only the pixels around this boundary, and thus is referred to as silhouette MAP. Because the characteristics of contents and silhouette are different, the concept and designed patterns in our work are somewhat different from those in Won (1998).

As mentioned above, the temporal phase separates background and non-background regions. The non-background region is basically composed of rectangular blocks with different sizes. However, as mentioned in the previous subsection, the boundary region has been partitioned into many square blocks with equal size $M \times M = L$, say $2 \times 2 = 4$. The set of the square blocks of pixels is denoted as $\Omega$, and the set of the corresponding block indexes is $\Omega_l = \{1, 2, \ldots, N\}$, where $N$ is

![Fig. 4. Temporal segmentation result.](image-url)
the total number of square blocks in the boundary region. The block-based MAP scheme described in the following operates on all the blocks in $\Omega$. Since the MAP operation requires neighboring blocks as the support of the current processing block, we define two auxiliary supports $\Omega^+$ and $\Omega^-$ as the exterior and interior neighborhoods of $\Omega$, respectively, as shown in Fig. 5. The two neighborhood regions are composed of blocks with the same size of the blocks in $\Omega$. Fig. 5 also shows that $\Omega^+$ resides strictly inside the background area, a fact that will be exploited later.

Here, we let a random field $Y$ denotes the set of pixel values of all blocks in $\Omega$. Obviously, the random field can be represented as the set of blocks in $\Omega$; i.e., $Y = \{y_k | k \in \Omega\}$, where the block $y_k = \{y_k(m,n) | m, n = 1, 2, \ldots, M\}$ is a set of all random variables (pixels) in the $M \times M$ block.

We design five feature patterns: smooth and four types of edge patterns, as shown in Fig. 6A. Fig. 6B shows the corresponding complements of the patterns. We denote the patterns as a set of $S = \{m, h, v, l, r\}$, where $m$ represents the smooth pattern, and $h, v, l, r$ represent horizontal, vertical, left-diagonal, and right-diagonal edge patterns, respectively. We aim to classify every block in $\Omega$ into one of the five designed patterns by using MAP criterion. That is, we would like to obtain the pattern-label mapping of each block in $\Omega$. The pattern-label mapping of all blocks in $\Omega$ is a random field $X = \{X_k = x_k | k \in \Omega; x_k \in \{m, h, v, l, r\}\}$.

The block classification using MAP is further expressed as follows. Given the set of boundary blocks in $\Omega$, $Y$, find the optimal label assignments $X^*$ that maximizes the a posteriori probability $P(X|Y)$. By using the Bayesian formulation, we can obtain $X^*$ by solving the following equation.

\[ P(X^*|Y) = \max_{X} P(Y|X)P(X) \]

Fig. 5. Definition of MAP operation region $\Omega$ and its supports $\Omega^+$ and $\Omega^-$. 
\[
X^* = \arg \max_x P(X = x | Y = y) = \arg \max_x \frac{P(Y = y | X = x) P(X = x)}{P(Y = y)} \propto \arg \max_x [\ln P(Y = y | X = x) + \ln P(X = x)].
\] (4)

Now the problem is how to calculate the conditional probability (the first term of the sum in Eq. (4)) and the a priori probability (the second term).

For the likelihood function \( P(Y|X) \), we assume that all the blocks in \( \Omega \) are independent of each other, given the block label configuration. In such case, \( P(Y|X) \) can be expressed as

\[
P(Y = y_k | X = x_k) = \prod_{k \in \Omega} P(Y = y_k | X = x_k).
\] (5)

When a block is classified into one of five patterns, as shown in Fig. 6, the block can be considered as two non-overlapped regions, defined as the sets \( R_1 \) and \( R_2 \), respectively. Although the smooth pattern (pattern 1) actually contains one part, for consistency but without introducing error, it is also regarded as the composition of two parts. For diagonal patterns, we let the pixels in the diagonal belong to lower part. The intensity in a part is assumed to be independent identical distributed (i.i.d) Gaussian (Derin and Cole, 1986; Derin and Elliott, 1987; Won, 1998). As a consequence, the conditional probability of a block can be modeled as

\[
P(Y = y_k | X = x_k) = P(Y = y_k \in R_1 | X = x_k) \cdot P(Y = y_k \in R_2 | X = x_k)
\]

\[
= \prod_{(m,n) \in R_1} \frac{1}{\sqrt{2\pi \sigma_{k,1}^2}} \exp \left[ -\frac{1}{2\sigma_{k,1}^2} \left( y_k(m,n) - \mu_{k,1} \right)^2 \right] \cdot \prod_{(m,n) \in R_2} \frac{1}{\sqrt{2\pi \sigma_{k,2}^2}}
\]

\[
\times \exp \left[ -\frac{1}{2\sigma_{k,2}^2} \left( y_k(m,n) - \mu_{k,2} \right)^2 \right],
\] (6)
where \((\mu_{k,1}, \sigma^2_{k,1})\) and \((\mu_{k,2}, \sigma^2_{k,2})\) represent the (mean, variance) of region 1 and region 2 of a block, respectively.

Substituting Eq. (6) into Eq. (5) and taking the logarithm, we have

\[
\ln P(Y|X) = \sum_{k \in \Omega} \left[ \sum_{R_1} () + \sum_{R_2} () \right] = \sum_{k \in \Omega} f(\bullet). \tag{7}
\]

The two terms in the bracket have the same form with different operation regions. The first term can be expressed as

\[
-\frac{1}{2} \sum_{(m,n) \in R_1} \ln(2\pi\sigma^2_{k,1}) - \sum_{(m,n) \in \Omega} \left[ \frac{1}{2\sigma^2_{k,1}} (y_k(m,n) - \mu_{k,1})^2 \right]. \tag{8}
\]

The maximal likelihood (ML) estimates of model parameters \((\mu_{k,1}, \sigma^2_{k,1})\) are (Won and Derin, 1992)

\[
\hat{\mu}_{k,1} = \frac{1}{N_1} \sum_{(m,n) \in R_1} y_k(m,n), \quad \hat{\sigma}^2_{k,1} = \frac{1}{N_1} \sum_{(m,n) \in \Omega} \left( y_k(m,n) - \hat{\mu}_{k,1} \right)^2, \tag{9}
\]

where \(N_1\) denotes the number of pixels in \(R_1\). The \(\mu\) and \(\sigma\) in Eq. (8) are substituted by their ML estimates in Eq. (9). Then we obtain the first term in the bracket of Eq. (7) as

\[
\sum_{R_1} () = -\frac{N_1}{2} \left[ \ln 2\pi + 1 + \ln \hat{\sigma}^2_{k,1} \right]. \tag{10}
\]

Similarly, we can obtain \(\sum_{R_2} ()\), and then \(f(\bullet)\) as

\[
f(\bullet) = -\left[ \frac{N_1}{2} (\ln 2\pi + 1 + \ln \hat{\sigma}^2_{k,1}) \right] - \left[ \frac{N_2}{2} (\ln 2\pi + 1 + \ln \hat{\sigma}^2_{k,2}) \right]. \tag{11}
\]

Neglecting the constant terms of Eq. (11) and substituting into Eq. (7), we have

\[
\ln P(Y|X) = \sum_{k \in \Omega} \left[ \frac{N_1}{2} \ln \hat{\sigma}^2_{k,1} + \frac{N_2}{2} \ln \hat{\sigma}^2_{k,2} \right]. \tag{12}
\]

Now we consider a priori probability \(P(X)\), which is used to impose the spatial smoothness on the block patterns. The underlying pattern of the block random field is assumed to be a Gibbs random field (GRF), which is a special class of GRF called multilevel logistic (MLL) Gibbs distribution (Won and Derin, 1992). Then

\[
P(X = x_k) = \frac{1}{Z} \exp \left( \sum_{c \in C} V_c(x_k) \right), \tag{13}
\]

where \(Z\) is a normalizing constant, \(V_c(x_k)\) is a clique potential for a clique \(c\), and \(C\) is the set of all cliques in \(\Omega\) associated with the specified neighborhood system. The detailed descriptions of GRF and the related topics can be found in Derin and Cole (1986), Geman and Geman (1984), Won (1998).

In this work, the first order neighborhood system \(\eta^1\) is adopted because it is simple and sufficient to define smoothness and connectivity. \(\eta^1\) consists of the closest four
neighbors of a block, hence it is well known as the nearest-neighbor model. The neighborhood system and the corresponding clique type which contains single-block and pair-block clique potentials are shown in Fig. 7.

For the single-block clique potential at \( k \in \Omega_t \), we define

\[
V_c(x_k) = \alpha_x,
\]

where \( x_k \in \{m, h, v, l, r\} \). In this work, the value of the parameter \( \alpha_x \) is determined as follows. We assume that the input test block belongs to one of edge patterns, and then check whether it satisfies the corresponding edge pattern condition. If the answer is yes, the value of the parameter for the specified edge is set to a constant \( \alpha \); otherwise it is assigned to zero. Repeat the procedure for all edge patterns, we can obtain the values of \( \alpha_x \) for \( x_k \in \{h, v, l, r\} \). The edge pattern condition is defined as

\[
\left| \bar{\mu}_{k,1} - \bar{\mu}_{k,2} \right| \geq TH,
\]

where \( \bar{\mu}_{k,1} \) and \( \bar{\mu}_{k,2} \) are the sample means of the two half of the input block corresponding to a specific edge pattern \( x_k \), defined in Fig. 6A. The \( TH \) is a threshold value calculated in a manner described later. We take an example to further explain how to determine the value of \( \alpha_x \). For an input block, we assume that it belongs to the edge pattern \( h \), and check whether it satisfies Eq. (15), if the answer is yes, \( \alpha_x = \alpha \); otherwise \( \alpha_x = 0 \). Next we assume that the input block belongs to the edge pattern \( l \). Then check whether it satisfies Eq. (15), If yes, let \( \alpha_x = \alpha \); otherwise \( \alpha_x = 0 \). Repeat the above procedure for all of the four possible edge patterns. Finally, if the input block does not belong to any edge pattern, then it belongs to the smooth pattern \( m \), so we set \( \alpha_m = \alpha \); otherwise \( \alpha_m = 0 \).

The threshold \( TH \) is an important parameter, which is calculated automatically according to the image characteristic by

\[
TH = \frac{1}{N_b} \sum_{k \in \Omega^+} (y_{\text{max}}^k - y_{\text{min}}^k),
\]

where \( y_{\text{max}}^k \) and \( y_{\text{min}}^k \) are, respectively, the maximum and minimum intensity values of the block \( k \) in \( \Omega^+ \), and \( N_b \) is the total number of blocks in \( \Omega^+ \). The \( TH \) defined above

![Fig. 7. The \( \eta^+ \) neighborhood system and the clique types selected. (A) Neighborhood system, (B) single-clique, (C) pair-clique.](image-url)
is the average dynamic range of intensity over the background blocks along the boundary of the temporal segmentation.

As pointed out in Won (1998), the pair-block clique potentials $V_c(x_k)$ in Eq. (13) represent the continuity and smoothness with the neighboring block configurations. The number of block labels is five. Thus the total number of pair-clique configurations is $5^2$ (horizontal-pair) + $5^2$ (vertical-pair) = 50. However, some of the configurations will occur with significant possibility. So they should be given a reward. On the contrary, the configurations that hardly exist should receive a penalty. The reward or penalty is defined by offering positive or negative clique potential as

\[
V_c(x) = \begin{cases} 
\beta, & \text{if the block pattern configuration in } c \text{ belong to } S_r \\
0, & \text{if the block pattern configuration in } c \text{ belong to } S_d \\
-\beta, & \text{if the block pattern configuration in } c \text{ belong to } S_p
\end{cases}
\]

(17)

Now we define the three categories: reward $S_r$, penalty $S_p$, and don’t-care $S_d$. In normal case, the pair-block configurations possessing the property of block-continuity or background-object connectivity make more sense physically. Thus, they are candidates deserving reward. On the other hand, the pair-block configurations that do not have these properties should be punished. Fig. 8 shows the possible pair-block configurations, in which Figs. 8A and B corresponds to horizontal-type pair and vertical-type pair, respectively, as shown in Fig. 7. In Fig. 8, the pair clique in dash-line is defined as object-background connectivity, the pair clique in solid-line is defined as continuity. The remaining pairs do not make sense physically; hence they should receive penalty.

The pair-blocks that do not belong to penalty category are further classified into reward and don’t-care types according to the condition

\[
\left| \left( y_{\text{max},R_1}^c - y_{\text{min},R_2}^c \right) \right| \geq TH.
\]

(18)

Fig. 8. The possible configuration of pair clique. (A) Horizontal pair clique, (B) vertical pair clique. Note. (1) The pair clique in dash-line is defined as object-background connectivity. (2) The pair clique in solid-line is defined as continuity. (3) The else do not make sense physically.
If the above condition is satisfied, the block is categorized into $S_r$; otherwise, it is $S_d$. As shown in Fig. 6, two regions construct an edge block, $R_1$ and $R_2$. The above equation implies that the two regions have significant intensity change. More specifically, the proposed technique aims at locating the object boundary that is constructed by edge blocks. Therefore, the blocks having significant edge characteristics will be given larger potential value.

Now taking logarithm of Eq. (13), combining the result with Eq. (12), and discarding the constant terms, the objective function of Eq. (4), which is to be maximized, becomes

![Fig. 8. (continued)](image)
\[ X^* = \arg \max_{X} \left[ \ln P(Y|X) + \ln P(X) \right] \]
\[ = \arg \max_{X} \left\{ \sum_{k \in \Omega} \left[ -\frac{N_1}{2} \ln \hat{\sigma}_{k,1}^2 - \frac{N_2}{2} \ln \hat{\sigma}_{k,2}^2 + \sum_{c \in C} \psi_c(x_k) \right] \right\} \]  \hspace{1cm} (19)

Finding the optimum block-label mapping \( X^* \) which satisfies Eq. (19) for all possible configurations of \( X \) is computational prohibitive. In our work, there are five possible pattern types, and about 100 blocks needed to be processed for the temporal segmentation result. In such case, there are \( 5^{100} \) possible configurations for \( X \); hence it is difficult to search all of the configuration space. In this work, we used a deterministic relaxation scheme in Won (1998) to update block pattern type iteratively. That is, we determine the label of each block in \( \Omega \) one-by-one, and then update the labels in an iterative manner. As in Won (1998), we employ the criterion \( \arg \max_{X} \left[ \ln P(Y=y|X=x) \right] \) to determine the pattern types in \( \Omega^+, \Omega, \) and \( \Omega^- \). The results are used as initial conditions for the subsequent iteration process. Then, traveling every block \( k \in \Omega \), we update the block type \( x_k \) depending on its neighboring block types. By doing the procedure iteratively until there is no more change of the block type or the allowed maximum number of iterations is reached, we can obtain the final block label configuration.

After all blocks in \( \Omega \) have been classified, we further categorize the results into two types of blocks: edge block and background block. In our case, since the neighboring stripe is located at background, the edge block indicates the transition from background to object. The edge block definitely covers the real contour of objects. The background blocks are pared off directly. However, the edge block consists of two parts. One belongs to background, whereas the other belongs to object. Thus, we pare off the background part that is near to the real background stripe \( \Omega^- \). After
cutting away the background blocks (or parts), we obtain a new boundary $\Omega$. Repeat
the process until all blocks in the new $\Omega$ has been categorized into edge block. In such
case, the silhouette of object is located.

The edge block is defined as the block in $\Omega$ that satisfies background-object con-
nectivity. In other words, the edge block and its neighboring block form a block-pair

![Flow-chart of proposed algorithm](image-url)

Fig. 10. The flow-chart of proposed algorithm.
that has the property of background-object connectivity. The blocks of the first row and first column in Fig. 8 satisfy background-object connectivity, and thus are regarded as edge blocks. The blocks in Ω that satisfies the continuity condition are defined as background blocks. The blocks located at the diagonal direction of Fig. 8 are background blocks.

The remaining blocks in Fig. 8 cannot be classified into edge block or background block directly. To categorize these ambiguous blocks, we define a pattern that is formed by three spatially connected blocks located at $Ω^−, Ω, $ and $Ω^+, $ respectively. Two pairs of blocks, called inside pair ($Ω^−, Ω$) and outside pair ($Ω^+, Ω$), are defined, as illustrated in Fig. 9. We measure the similarity of each pair by calculating the mean square error of the two blocks of a pair. If the similarity of outside pair is greater than that of inside pair, it indicates that the current block is more similar to background; thus it is classified as background block. Otherwise, it is regarded as edge block.

In our work, the parameters related to both temporal and spatial measure are calculated from the image data. In order to avoid the influence of noisy data on the calculation of parameters, we apply a low pass filter, a simple $3 \times 3$ average filter, to smooth out the input data before temporal segmentation. Putting all of the steps de-

Fig. 11. Original test sequences. (A) Claire, (B) Trevor, (C) Salesman, (D) Table Tennis.
scribed in Section II-A and II-B, we summarize the proposed algorithm as shown in Fig. 10.

3. Simulation results and discussions

In this section, we have applied the proposed segmentation algorithm to the test image sequences including Claire, Trevor, Salesman, and Table tennis, as illustrated in Fig. 11. The sequences sizes are: Claire and Salesman with CIF resolution (352×288), Trevor with 256×256 and Table tennis with 352×256, but with different frame rates: 10 Hz for Claire sequence, and 15 Hz for all other used test sequences. The block size for MAP operation is 2×2. The parameter values for

Fig. 12. Segmentation results of the test sequences. (A) Segmented boundary, (B) extracted video object, (C) background segment.
MAP are chosen as $a = 1$ and $\beta = 1$, which are determined experimentally. Since the block size is small, the parameter values should be small. Moreover, our experiments indicate that the segmentation performance is not very sensitive to the parameter values. Therefore, we used a cut-and-try procedure to determine the parameter values, and the results are satisfactory for our applications.

To investigate the segmentation performance, we evaluate the error between the original boundary and the segmented boundary. The original boundary is actually unknown. In this work, it is obtained by carefully drawing the image contour using a mouse. The segmentation results are shown in Fig. 12, which demonstrates the segmented boundary, extracted video object, and background segment, respectively. We describe the results briefly as follows.

The sequence Claire is with uncluttered background. Also its background and object has significantly different characteristics. It can be seen that the extracted silhouette is very close to the real contour of the image object, except the hair tips on the border. The background of Trevor has some kinds of texture, which is a little more complicated than Claire. But the determined contour is still very close to real contour of the object. Salesman sequence is with very complicated background, and portions of its object boundary have similar gray level as the background. In addition, there is a shadow effect on the table. Therefore, the extracted contour has become somewhat smoothed less. In general, shadow effects, illumination change, and noise often affect the segmentation performance. Because we used image contents to calculate the necessary parameters, these effects can be reduced significantly. For example, the shadow of Table Tennis is removed in the segmented object, as seen from Fig. 12.

To further evaluate the segmentation performance, the error per contour point (error area divided by total number of contour points) for each sequence is calculated. The results are listed in Table 1. The average error is less than 2 pixels per contour point, except “Salesman,” which is a very complicated sequence. The visual distortion between the original boundary and segmented boundary for the two sequences are illustrated in Fig. 13. By carefully checking this figure, we found that the segmented object boundaries tend to penetrate into the background. This may be caused by the low-pass filtering before segmentation, as mentioned in the previous

<table>
<thead>
<tr>
<th>Sequences</th>
<th>Size of image</th>
<th>Execution time of temporal segmentation (ms)</th>
<th>Execution time of spatial segmentation (ms)</th>
<th>The length of original contour</th>
<th>Total error area (pixels)</th>
<th>Average error (pixel/contour)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Claire</td>
<td>352 × 288</td>
<td>471</td>
<td>5528</td>
<td>851</td>
<td>678</td>
<td>0.79</td>
</tr>
<tr>
<td>Trevor</td>
<td>256 × 256</td>
<td>320</td>
<td>6019</td>
<td>857</td>
<td>1039</td>
<td>1.21</td>
</tr>
<tr>
<td>Salesman</td>
<td>352 × 288</td>
<td>551</td>
<td>1582</td>
<td>962</td>
<td>2611</td>
<td>2.71</td>
</tr>
<tr>
<td>Table Tennis</td>
<td>352 × 256</td>
<td>660</td>
<td>6539</td>
<td>614</td>
<td>742</td>
<td>1.20</td>
</tr>
<tr>
<td>Average</td>
<td>500.5</td>
<td>4917</td>
<td>821</td>
<td>1267.5</td>
<td>1.478</td>
<td></td>
</tr>
</tbody>
</table>

Table 1
The performance of the proposed segmentation method
The penetration effect can be reduced by post processing, which is, however, not the main issue of this paper.

Object extraction is strongly dependent on backgrounds. When objects exist in the complicated background, it is difficult to exactly distinguish object boundary from

Fig. 13. The comparisons of original and segmented boundary for (A) Claire, (B) Salesman, (C) Trevor, and (D) Table Tennis.
background even with human eyes. Our method performs very well for scenes with uncluttered background. For complicated background, it achieves acceptable segmentation performance. Of course, because the real object boundary exist some am-

Fig. 13. (continued)
biguity and the proposed method is block based, the jagged-edges are usually unavoidable. For examples, the hair tips in Clair sequences and head of “Trevor” and “Salesman” sequences, as seen in Figs. 12 and 13. Again, we can smooth out the contour with simple post processing. In order to further demonstrate the performance, we show the segmented boundary over several frames, as shown in Fig. 14. The results indicate that segmentation performance is acceptable.

Fig. 14. Segmentation results of several frames for Claire and Salesman (A) Claire (1) 10th, (2) 50th, (3) 87th, and (4) 130th frame, (B) Salesman, (1) 7th, (2) 10th, (3) 20th, and (4) 61th frame.
Table 1 also lists the execution time of spatial segmentation with MAP and the total execution time for each frame. The results are obtained by using Intel Pentium II processor with 350 MHz of clock rate. Because MAP operates on a very small near-boundary region, the computational cost is relatively small. In addition, the average computation time for the segmentation of a CIF sequence is about 0.8 s per frame. Obviously, the proposed technique is computationally very efficient.
In this work, we assume scenarios with static background/camera and moving foreground objects. If moving background exists, e.g., Foreman sequence, global motion estimation and compensation (Mpeg Video Group, 2001) can be used to simplify the image and to obtain a static background video sequence.

4. Conclusions

In this paper, we have presented an automatic temporal-to-spatial video segmentation technique to extract semantic video objects from static background. The temporal phase first removes pure background region and obtains the exterior boundary of an object using a simple temporal-activity measure. Then a MAP scheme using spatial information is employed to refine the temporal segmentation by paring off the remaining background blocks iteratively. The proposed method is computationally efficient because (a) temporal-activity rather than motion parameters is used in temporal segmentation phase, and (b) the computationally expensive MAP is applied only in a near-boundary region, which is about 6% of an average frame. Furthermore, the most part of threshold values in the algorithm are calculated automatically, which avoids the disadvantages of user interaction.

When the objects and background have distinct spatial characteristics, the proposed technique achieves excellent performance. For complex background, it also works well with little degradation of performance. The temporal phase in our method detects moving blocks with frame-difference information. This produces unsatisfactory results for the sequence with moving background. In addition, shadow effects, illumination change, and noises may affect the segmentation performance. These will be investigated in the future.

Acknowledgments

The authors would like to express their sincere thanks to anonymous reviewers for their invaluable comments, which truly helped toward an effective presentation of this work. This work was supported by the National Science Council, Taiwan, Republic of China, under Grant NSC-89-2213-E-214-098.

References


Chung-Ming Kuo received the B.S. degree from the Chinese Naval Academy, Kaohsiung, Taiwan, in 1982, the M.S. and Ph.D. degrees all in electrical engineering from Chung Cheng Institute of Technology, Taiwan, in 1988 and 1994, respectively. From 1988 to 1991, he was an instructor in the Department of
Electrical Engineering at Chinese Naval Academy. Since January 1995, he has been an associate professor at the same institution. In August 2000, he joined the I-Shou University, where he is currently an associate professor with the Department of Information Engineering. His research interests include object-based video compression and retrieving, image processing, multimedia signal processing, and optimal estimation.

Chaur-Heh Hsieh received his BS and MS degrees in electrical engineering in 1976 and 1981, respectively, and his PhD degree in system engineering in 1990, all from Chung Cheng Institute of Technology, Taiwan, China. In 1981, he joined the faculty of the Department of Electrical Engineering at Chung Cheng Institute of Technology, and became a professor in 1993. In August 1996, he joined the I-Shou University, where he is currently Professor of the Department of Information Engineering. His research interests include image/video coding, image/video watermarking, digital filter design, VLSI architecture for real-time image processing, and wireless multimedia communications. He has published more than 100 papers in these subjects.

Yong-Ren Huang received the B.S. degree in computer science from Tang-Kang University Taiwan in 1990 and the M.S. degree in information engineering from I-Shou University Taiwan in 2000. Now, he is studying Ph.D. degree in information engineering at I-Shou University. From 1994 to 1996, he was a R&D engineer in the department of Acer TWP Corporation. Then, he ever joined Philip Origin Corporation, where he was a system engineer between 1996 and 1997. His research interests include object-based video coding, image processing, retrieving and multimedia communications.