Fast and Accurate Moving Object Extraction Technique for MPEG-4 Object-based Video Coding

Ju Guo, Jongwon Kim and C.-C. Jay Kuo
Integrated Media Systems Center and Department of Electrical Engineering-Systems
University of Southern California, Los Angeles, CA 90089-2564
E-mail:{juguo,jongwon,cckuo}@sipi.usc.edu

ABSTRACT

A fast and robust video segmentation technique is proposed to generate a coding optimized binary object mask in this work. The algorithm exploits the color information in the $L^\star u^\star v^\star$ space, and combines it with the motion information to separate moving objects from the background. A non-parametric gradient-based iterative color clustering algorithm, called the mean shift algorithm, is first employed to provide robust homogeneous color regions according to dominant colors. Next, moving regions are identified by a motion detection method, which is developed based on the frame intensity difference to circumvent the motion estimation complexity for the whole frame. Only moving regions are analyzed by a region-based affine motion model, and tracked to increase the temporal and spatial consistency of extracted objects. The final shape is optimized for MPEG-4 coding efficiency by using a variable bandwidth region boundary. The shape coding efficiency can be improved up to 30% with negligible loss of perceptual quality. The proposed system is evaluated for several typical MPEG-4 test sequences. It provides consistent and accurate object boundaries throughout the entire test sequences.

Keywords: video segmentation, color segmentation, mean shift algorithm, shape coding, affine motion, spatial segmentation, motion detection.

1 INTRODUCTION

The emerging MPEG-4 standard introduces the concept of content-based video coding and representation [1]. The object-based coding has the potential to provide a more accurate video representation at very low bit rates, and allows content-based functionalities such as object manipulation, indexing and retrieval. In the MPEG-4 standard [1], video coding is handled by the object unit, i.e. the video object plane (VOP). VOP represents one snap shot of an object in video. For each VOP, the motion, texture, and shape information is coded in separate bit streams. This allows separate modification and manipulation of each VOP, and supports the content-based functionality. For object-based video coding via MPEG-4, it is essential to have the video object in advance. However, most of the existing video clips are frame-based. Thus, video segmentation, which aims at the exact separation of moving objects from the background, becomes the foundation of content-based video coding. Even though the image/video segmentation problem has been studied for more than thirty years, it is still considered one of the most challenging image processing tasks, and demands creative solutions for major breakthrough.

The Human visual system (HVS) can effortlessly segment scenes into different semantic objects. However, most segmentation algorithms only work at the pixel level for digital images and video. Pixels are grouped into regions based on different features. The "semantic object" is usually considered as a group of homogeneous regions that are identified by low level features such as the color, motion, and texture information. Thus, the key focus of video segmentation includes region segmentation, tracking, and refinement to generate semantically meaningful regions.

Most existing video segmentation algorithms attempt to exploit the temporal and spatial coherence information in the video sequence to achieve foreground/background separation [2–9]. Temporal segmentation can identify moving objects since most moving objects have coherent motion which is distinct from the background. Spatial segmentation can determine object boundaries accurately if underlying objects have a different visual appearance (such as the color or the gray level intensity) from the background. An efficient combination of spatial-temporal segmentation modules can lead to a more promising solution to the segmentation problem. It is desirable to develop an automatic segmentation algorithm that requires no user assistance and interaction. In addition, the availability of a fast implementation is also one basic requirement, which is especially important for real time applications.

A fast and robust video segmentation technique is proposed in this work. It can be roughly described below. First, a non-parametric gradient-based iterative color clustering algorithm, called the mean shift algorithm, is employed to provide robust dominant color regions according to color similarity. With the dominant color information from previous frames as the initial guess for the next frame, the amount of computational time can be reduce to 50%. Next, moving regions are identified by a motion detection method, which is developed based on the frame intensity difference to circumvent the motion estimation complexity for the whole frame. Only moving regions are further analyzed by a region-based affine motion model, and are tracked to increase temporal and spatial consistency of extracted objects. The final shape is optimized to increase the coding efficiency as high as 30% with negligible loss of visual information.

The paper is organized as follows. A general description of the proposed segmentation algorithm is given in Section 2. A color-based spatial segmentation process is described in Section 3. A technique for motion detection and tracking to achieve the spatial-temporal information integration is examined in Section 4. Experimental results are given in Section 5 to demonstrate the performance of the spatial-temporal segmentation by using the color information. The shape optimization for coding efficiency is detailed and the performance of the resulting algorithm is provided in Section 6. Concluding remarks are given in Section 7.

2 PROPOSED VIDEO SEGMENTATION ALGORITHM

In this work, we focus on automatic video segmentation with a fast and adaptive algorithm with a reduced complexity in both spatial and temporal domains. The block diagram of the proposed automatic video segmentation algorithm is given in Fig. 1.

A fast yet robust adaptive color segmentation based on the mean shift color clustering algorithm is applied in the spatial domain. The mean shift color segmentation algorithm is used to partition an image into homogeneous regions. The mean shift algorithm has been generalized by Cheng [10] for clustering data, and used by Comaniciu and Meer for color segmentation [11]. For the k-means clustering method, it is difficult to choose the initial number of classes. By using the mean shift algorithm, the number of dominant colors can be determined automatically. Here, we develop a non-parametric gradient-based algorithm that provides a simple iterative method to determine the local density maximum. The number of color classes in the current frame can be used as the initial guess of color classes for the next frame. This helps in reducing the computational complexity of color segmentation.

For the temporal domain, a noise-robust higher-order statistic motion detection algorithm and a region-based affine motion model is employed. After dividing an image frame into homogeneous spatial regions, we determine whether each region belongs to the background or the foreground by motion detection. Only moving regions are analyzed using the region-based affine motion model. The six parameters of the affine motion model are estimated for each region. The motion information of each region is tracked to increase the consistency of extracted objects.

At the last stage, the morphological open and closure filters are used to smooth object boundaries and eliminate small regions. The final object boundaries are postprocessed to increase the shape coding efficiency with negligible loss of visual information.

The building blocks of the proposed algorithm are detailed in the next three sections.

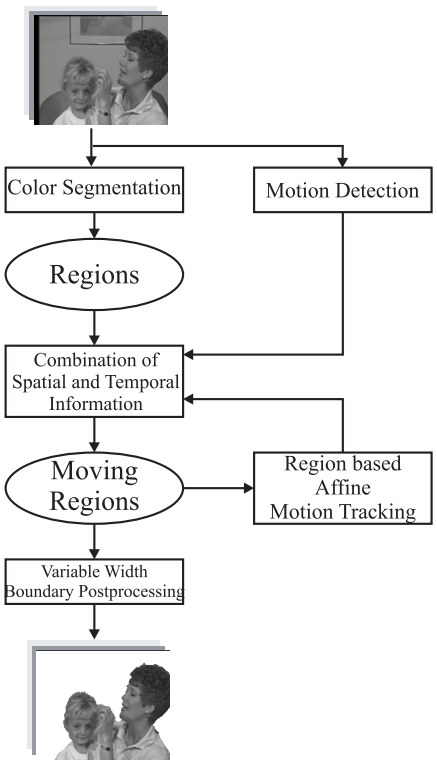


Figure 1: The block diagram of the proposed automatic video segmentation algorithm.

3 COLOR-BASED SPATIAL SEGMENTATION

The intensity distribution of each color component can be viewed as a probability density function. The mean shift vector is the difference between the mean of the probability function on a local area and the geometrical center of this region. In terms of mathematics, the mean shift vector associated with a region $S_{\vec{x}}$ centered on $\vec{x}$ can be written as

$$\vec{V}(\vec{x}) = \frac{\int_{\vec{y}\in S_{\vec{x}}} p(\vec{y})(\vec{y}-\vec{x})d\vec{y}}{\int_{\vec{y}\in S_{\vec{x}}} p(\vec{y})d\vec{y}},$$

where $p(\cdot)$ is the probability density function. The mean shift algorithm says that the mean shift vector is proportional to the gradient of the probability density $\nabla p(\vec{x})$, and reciprocal to the probability density $p(\vec{x})$, i.e.

$$\vec{V}(\vec{x}) = c\frac{\nabla p(\vec{x})}{p(\vec{x})},$$

where c is a constant. Since the mean shift vector is along the direction of the probability density maximum, we can exploit this property to find the actual location of the density maximum. In implementing the mean shift algorithm, the size of the search window can be made adaptive to an image by setting the radius proportional to the trace of the global covariance matrix of the given image. By moving search windows in the color space using the mean shift vector iteratively, one dominant color can be located. After removing all colors inside the converged search window, one can repeat the mean shift algorithm again to locate the second dominant color. This process can be repeated several times to identify a few major dominant colors.

The uniform color space $L^\star u^\star v^\star$ was used by Comaniciu et al. [11] for color segmentation due to its perceptual homogeneity. By investigating segmentation in the $L^\star u^\star v^\star$ space and the luminance space, our results shows that segmentation in the $L^\star u^\star v^\star$ space gives more robust results than that in the luminance space. Thus, the $L^\star u^\star v^\star$ color space is chosen in our scheme.

Dominant colors of the current frame are used as the initial guess of dominant colors in the next frame. Due to the similarity of adjacent frames, the mean shift algorithm often converges in one or two iterations, thus reducing the computational time significantly. For example, with the dominant color information from previous frames as the initial estimate for the next frame, the amount of computational time can be reduced up to 50%.

Color segmentation also uses the spatial relation of pixels as a constraint [11] as described below. For each frame, dominant colors are first generated by the mean shift algorithm. Then, all pixels are classified according to their distance to dominant colors. A relative small distance is used as a threshold to determine which classes the pixel belong to in the beginning. Afterwards, the threshold is doubled. Only the pixel that has a smaller distance to the dominant color and has one of its neighboring pixels assigned to the same class can be classified to this class. Finally, unassigned pixels are classified to its nearest neighboring region.

4 MOTION DETECTION AND TRACKING FOR SPATIAL-TEMPORAL INTEGRATION

Since most of semantic objects are characterized by a coherent motion pattern which is distinct from that of the background. The motion is commonly used to group regions into objects. Parametric models can be applied to describe the motion of each region by one set of parameters that is either estimated by fitting a model in the least-square sense to a motion field obtained by a non-parametric method, such as the block based matching or the optical flow method, or directly from the luminance signal $I(x, y, t)$. Affine and perspective motion models are most frequently used among parametric models.

Once the motion parameters of a region are obtained, the region can be tracked to the subsequent frame. Since motion models are derived for a rigid planar surface, there will be a tracking error for non-rigid regions or objects. Thus, the region has to be updated in the current frame. The watershed algorithm is often used for region boundary refinement and update.

Since motion estimation is computationally expensive, we use a simple and robust motion detection algorithm to locate the moving regions first. A robust motion detection method based on the frame difference calculation is used to determine whether homogeneous regions are moving or not [2]. Since the statistical behavior of inter-frame differences produced by the object movement strongly deviates from the Gaussian model, a fourth-order statistic adaptive detection of the non-Gaussian signal is performed. For each pixel at (x, y), its fourth order moments $\hat{m}_d(x, y)$ is evaluated as

$$\hat{m}_d(x, y) = \frac{1}{9} \sum_{(s,t) \in W(x,y)} (d(s, t) - \hat{\mu}(x, y))^4,$$

where $d(x, y)$ is the inter-frame difference, $W(x, y)$ is 3×3 window centered at (x, y) and $\hat{\mu}(x, y)$ is the sample mean of $d(x, y)$ inside window $W(x, y)$, i.e.

$$\hat{\mu}(x, y) = \frac{1}{9} \sum_{(s,t) \in W(x,y)} d(s, t).$$

Each pixel at (x, y) is determined to be associated with the still background or the change region according to its fourth moment $\hat{m}_d(x, y)$. The change regions obtained from higher statistic estimation include the uncovered background. The block matching algorithm is applied to the fourth order moment maps of frame differences in order to remove the uncovered background. Pixels that have null displacements are reassigned to the background. For each homogeneous region, if 85% of pixels are identified as moving pixels, the region is identified as moving.

Only for moving regions, the motion vector field is estimated by using the optical flow equation and fit with the affine motion model. The affine motion model can be written as

$$\begin{bmatrix} \mu(x, y) \\ \nu(x, y) \end{bmatrix} = \begin{bmatrix} a_1 + a_2 x + a_3 y \\ a_4 + a_5 x + a_6 y \end{bmatrix},$$

where $\mu(x, y)$ and $\nu(x, y)$ are motion vectors along horizontal and vertical directions, $a_1 \cdots a_6$ are constant parameters. The 6 parameters $(a_1 \cdots a_6)$ of the affine model are estimated by using the optical flow equation.

By assuming that intensity $I(x,y,t)$ remains constant along a motion trajectory, the optical flow equation can be written as

$$\frac{dI(x,y,t)}{dt} = I_x(x,y,t)\mu(x,y) + I_y(x,y,t)\nu(x,y) + I_t(x,y,t) = 0,$$

where I_x, I_y, and I_t are the partial derivatives with respect to x, y and t. By substitute the $\mu(x,y)$ and $\nu(x,y)$ with the affine motion model over the region R, we obtain

$$E = \sum_{(x,y)\in R} [I_x(x,y,t)\mu(x,y) + I_y(x,y,t)\nu(x,y) + I_t(x,y,t)]^2.$$

To minimize E, we differentiate E with respect to $a_1, \cdots, a_6$, and set the resulting equations to zero to obtain six linear equations with six unknowns. Parameters $a_1, \cdots, a_6$ are obtained by solving the six linear equations. Since the optical flow estimation is sensitive to noise, I_x, I_y, and I_t are obtained from the derivative maps and averaged over a median filter.

$$\begin{aligned}
I_x &= Media(I(x+1,y+1,t) - I(x,y+1,t), I(x+1,y,t) - I(x,y,t), \\
&\quad I(x+1,y+1,t+1) - I(x,y+1,t+1), I(x+1,y,t+1) - I(x,y,t+1)), \\
I_y &= Media(I(x+1,y+1,t) - I(x+1,y,t), I(x,y+1,t) - I(x,y,t), \\
&\quad I(x+1,y+1,t+1) - I(x+1,y,t+1), I(x,y+1,t+1) - I(x,y,t+1)), \\
I_t &= Media(I(x+1,y+1,t+1) - I(x+1,y+1,t), I(x+1,y,t+1) - I(x+1,y,t), \\
&\quad I(x,y+1,t+1) - I(x,y+1,t), I(x,y,t+1) - I(x,y,t)),
\end{aligned}$$

where

$$Media(t_1,t_2,t_3,t_4) = \frac{1}{2}(t_1+t_2+t_3+t_4 - Max(t_1,t_2,t_3,t_4) - Min(t_1,t_2,t_3,t_4)).$$

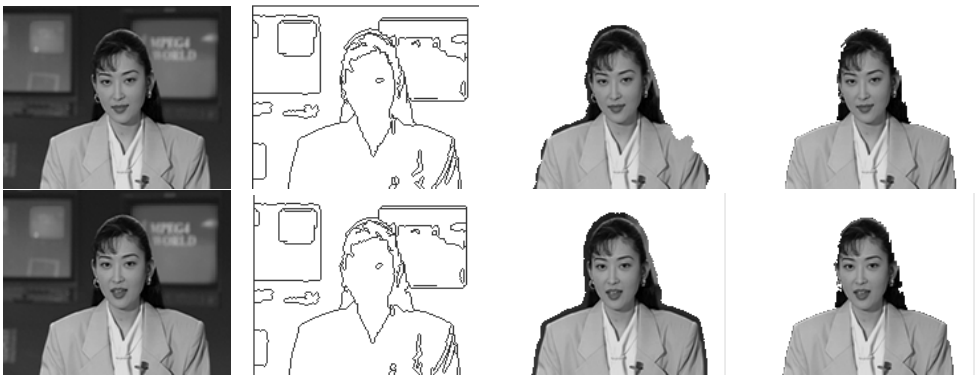
Moving objects are tracked to the next frame according to their affine motion models. The tracked region boundaries are updated by aligning to the current matched region boundaries by region matching. If over 75% pixels are the same between two regions, we say that these regions are matched. For unmatched regions, the change of detection is used to find moving regions. For each new moving region, we repeat the process of motion estimation. This process allows the detection of newly appeared objects in the scene.

5 EXPERIMENTAL RESULTS OF COLOR-BASED SPATIAL-TEMPORAL SEGMENTATION

5.1 Subjective Evaluation

Two MPEG-4 QCIF sequences, i.e. "Akiyo" and "Mother and daughter", were used to test the proposed algorithm described in Sections 3 and 4. For the "Akiyo" sequence, there is only a small motion activity in the head and shoulder regions. The original 10th and 20th image frames are shown in Fig. 2(a). The results of color segmentation are given in Fig. 2(b). We can clearly see that each image is segmented into a few regions. For example, Akiyo is segmented into the hair region, the facial region, and the shoulder region. Each region has a well-aligned boundary corresponding to the real object. The motion detection algorithm identifies the moving region, which is given in Fig. 2(c). The boundary is not well detected as compared with the real object boundary by using the motion information only. By incorporating the spatial color segmentation result, the final segmentation result is much improved as shown in Fig. 2(d).

For the "Mother and daughter" sequence, there are more head and hand motion activities than "Akiyo". The results of color segmentation is shown in Fig. 3(b), for two different frames (i.e. the 20th and 250th frames). More regions are obtained from color segmentation. All these regions are identified as belonging to either the background or the foreground. Regions, such as mother's head and shoulder, daughter's hair, shoulder and face, have contours which correspond to real objects. These objects, identified by motion detection and defined by color regions, were accurately segmented from the background as given in Fig. 3(d).

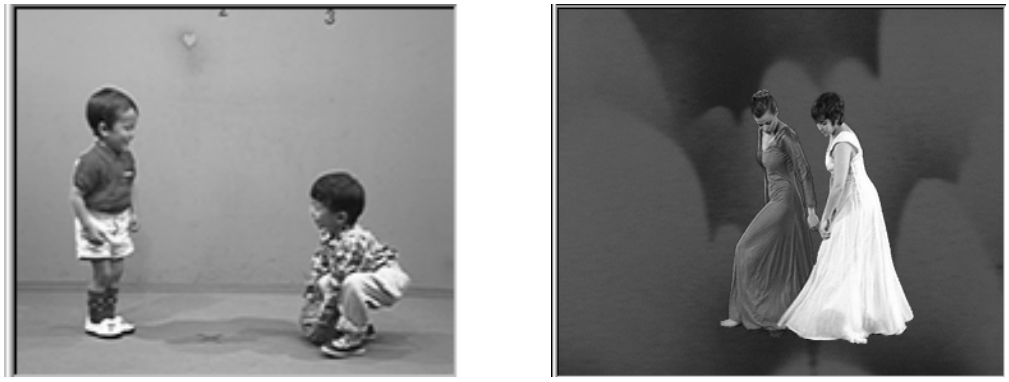


(a) (b) (c) (d)

Figure 2: The segmentation results of the "Akiyo" QCIF sequence with respect to the 20th frames: (a) the original images, (b) the color segmentation results, (c) the motion detection results and (d) the final results.

(a) (b) (c) (d)

Figure 3: Segmentation results of the "Mother and daughter" QCIF sequence with respect to the 20th and the 250th frames: (a) original images, (b) color segmentation results, (c) motion detection results and (d) final results.



(a) (b)

Figure 4: One image frame from (a) the QCIF "Children" sequence and (b) the CIF "Dancer" sequence.

To demonstrate the tracking effects, results of "children" QCIF and "dancer" CIF sequence are shown in Fig 5 and Fig 6, respectively. One image from each of the original sequences is shown in Fig 4. In the "children" sequence, there are fast motion activities, such as the non-rigid motion of bodies and the motion of the ball. Segmentation results of this sequence are shown in Fig. 5. From these results, we see that the proposed algorithm tracks both the fast moving ball and the non-rigid motion of human body pretty well. In the "dancer" sequence, there is fast motion of human bodies. Segmentation results of this sequence are shown in Fig. 6. Again, the proposed algorithm performs well for body segmentation and tracking.

Figure 5: Segmentation results of the "Children" QCIF sequence: frame no. 4, 8, 12 and 15 (the 4 images in the top row), frame no. 41, 46, 51 and 56 (the 4 images in the middle row), and frame no. 191, 193, 196 and 199 (the 4 images in the bottom row).

Figure 6: Segmentation results of the "Dancer" CIF sequence: frame no. 41, 43, 46 and 49 (the 4 images in the top row) and frame no. 51, 53, 56 and 59 (the 4 images in the bottom row).

5.2 Objective Evaluation

Although many segmentation algorithms have been proposed, it is still a very difficult problem to evaluate the quality of the generated video objects. In MPEG-4, only subjective evaluation by tape viewing was adopted to decide the quality of segmentation results. It is desirable to use to an objective measure by comparing the

segmented object with the reference object. Two criteria, i.e. spatial accuracy and temporal coherency of the video object, have been viewed as important measures to evaluate the performance of various algorithms.

Wollborn and Mech [12] proposed a simple pixel-based quality measure. The spatial distortion of an estimated binary video object mask at frame t is defined as

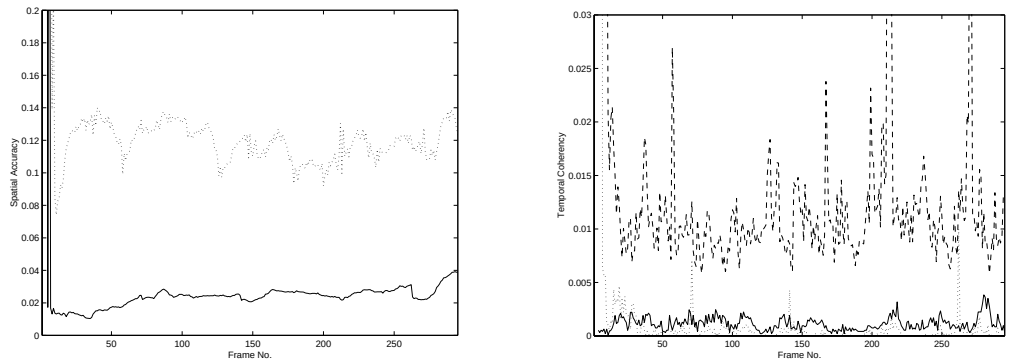
$$d(A_t^{est}, A_t^{ref}) = \frac{\sum_{(x,y)} A_t^{est}(x,y) \oplus A_t^{ref}(x,y)}{\sum_{(x,y)} A_t^{ref}(x,y)},$$

where A_t^{ref} and A_t^{est} are the reference and the estimated binary object masks at frame t, respectively, and $\oplus$ is the binary "XOR" operation. The temporal coherency is measured by

$$\eta(t) = d(A_t, A_{t-1}),$$

where A_t and A_{t-1} are binary masks at frame t and $t-1$, respectively. Temporal coherency $\eta^{est}(t)$ of the estimated binary mask A^{est} should be compared to temporal coherency $\eta^{ref}(t)$ of the reference mask. Any significant deviation from the reference indicates a bad temporal coherency.

In addition to the visual evaluation given in Section 6.1, segmentation results of the proposed algorithm are evaluated by using both criteria described above. The corresponding results of the "Akiyo" QCIF sequence are shown in Figs. 7(a) and (b). For the reference mask, the hand-segmented mask from the MPEG-4 test material distribution is utilized. In Fig. 7(a), the dot line is obtained by using higher statistic motion detection only while the solid line is the proposed scheme. We see that the spatial accuracy is improved very much by using the color segmentation algorithm. The error is less than 2% in most frames. In Fig. 7(b), the solid line denotes the reference mask, the dot line the proposed scheme, and the dash line the motion detection using the high order statistic method only. The temporal coherency curve also closely follows the one of the reference mask.



(a) (b)

Figure 7: The objective evaluation of the "Akiyo" QCIF sequence object mask for (a) spatial accuracy and (b) temporal coherency.

We see from these results that temporal segmentation can identify moving regions while spatial segmentation provides the important information of object boundaries. The proposed algorithm exploits spatial information of color similarity, and obtains the accurate region boundary automatically. Since the human visual system is very sensitive to the edge information, our segmentation results provide excellent visual quality.

6 VARIABLE WIDTH BOUNDARY POSTPROCESSING FOR OPTIMIZED SHAPE CODING

6.1 Algorithm Description

The object masks obtained from the spatial and temporal segmentation procedures described in Sections 3 and 4 sometimes have irregularity in the boundaries such as small gulfs and isthmi due to temporal and spatial signal

fluctuations. This will give visually annoying appearance and also increase the cost of shape coding. We use the morphological open and close operators to remove gulfs and isthmi, and to smooth object boundaries to increase the shape coding efficiency. A circular structuring element with 2 pixel radius is used in the morphological open and close operation. To optimize the shape coding efficiency, the following postprocessing technique is developed.

The object-based codec requires shade coding, which consumes extra bits comparing to the frame-based coding approach. The efficiency of shape coding becomes important, especially at low bit rates. Currently, MPEG-4 adopts the down-sampling of the shape mask for lossy shade coding. However, the down-sampling of the shape mask fails to exploit some visual properties of the shape. Since the human visual system (HVS) is not sensitive to edges with a low gradient, less bits can be used for the part of shape located at the low gradient area. Moreover, the extracted shape might not perfectly represent the real object boundary due to the limitation of object segmentation algorithms. If a less number of bits are used for the inaccurate part of the shape, we tend to have a more efficient shape coding scheme.

Generally speaking, the segmentation process has been treated as a preprocessing operation and completely separated from the coding process in MPEG-4. Once the video object mask is obtained through the segmentation process, only the binary formatted shape mask is conveyed to the object-based video coder for coding. The video coder does not possess the information of the confidence on the obtained object boundary and the visual significance of the object boundary. In this work, we design a segmentation algorithm that exploits the visual properties of object shapes by using a variable width object boundary to represent the object shape as an intermediate step while the final object boundary is obtained according to the coding bit rate requirement. In other words, the object boundary is quality scalable in the sense that it can preserve the visual significant information available at a certain bit rate.

Since most segmentation algorithms are based on the image spatial gradient to distinguish the object from the background. If the spatial gradient is zero or small, it will make segmentation algorithms fail or prone to error because even the human visual system (HVS) cannot segment the object without a distinct color from the background. Therefore, we use the image spatial gradient perpendicular to the object boundary as a criterion to generate the variable width boundaries. Four templates are used to match the boundary directions, horizontal, vertical, and two diagonal directions. The gradient is obtained by using the Robinson filters:

$$\underset{horizontal}{\begin{vmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{vmatrix}} \quad \underset{vertical}{\begin{vmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{vmatrix}} \quad \underset{diagonal}{\begin{vmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{vmatrix}} \quad \underset{diagonal}{\begin{vmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{vmatrix}}$$

Modified morphological dilation and erosion operations are performed at pixels along object boundaries. For dilation, four connected neighboring pixels are set to 1 if the gradient of the pixels is below a certain threshold, where 1 indicates that the pixel is belong to object mask. Four connected neighboring pixels are set to 0 for erosion, where 0 indicates that the pixel is belong to background. The modified dilation and erosion operations are defined below.

For dilation δ_s, if $(x, y) \in Boundary$ and $|Gradient(x, y)| < T$, then

$$Pixel(x, y) = Pixel(x + 1, y) = Pixel(x, y + 1) = Pixel(x, y - 1) = Pixel(x - 1, y) = 1.$$

For erosion ϵ_s, if $(x, y) \in Boundary$ and $|Gradient(x, y)| < T$, then

$$Pixel(x, y) = Pixel(x + 1, y) = Pixel(x, y + 1) = Pixel(x, y - 1) = Pixel(x - 1, y) = 0.$$

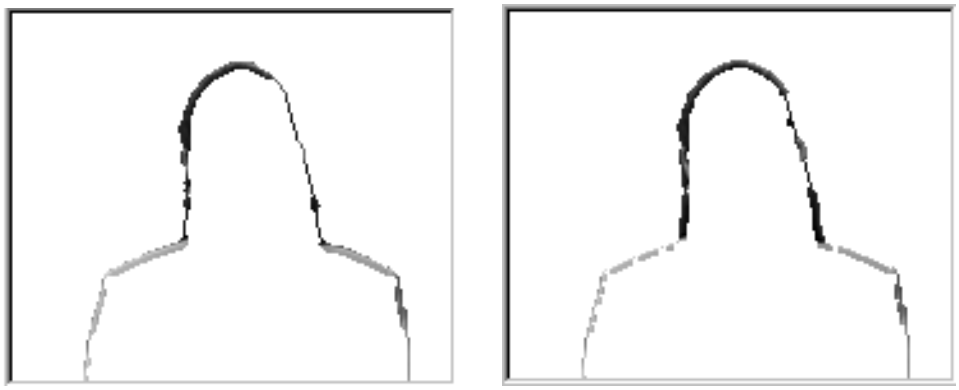
where $T = t\sigma$, σ is the intensity variance and t is a scale factor (set to 0.75 for the current application).

The difference of the dilation and erosion operations generate a band of variable width for object boundaries, i.e.

$$BB = \delta_s(MASK) - \epsilon_s(MASK)$$

The dilation and erosion operations can be repeated for a couple of times until the desired bandwidth of the object boundary is achieved.

We have analyzed the image gradient along object boundaries by using the luminance and the color. The results of the region boundary with a variable width are given in Fig. 8. As shown in the figure, the bandwidth obtained with the luminance gradients is not as uniformly distributed as the one obtained with the color gradients. Pay special attention to the hair region in the "Akiyo" sequence. It is very difficult to discern the edge by using the luminance information only, which has a large area of uncertainty. The width is much smaller by using the color gradient. It demonstrates that the object boundary can be more accurately estimated by using the color rather than the luminance.



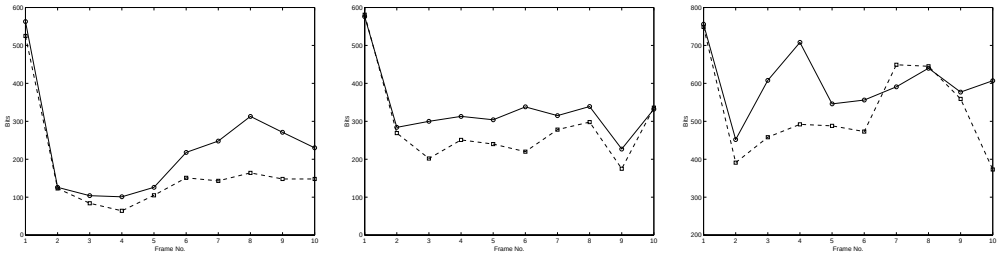
(a) (b)

Figure 8: The detected region boundary of a variable width by using (a) the color gradient and (b) the luminance gradient.

After generating the object boundary of a variable bandwidth, we develop a scheme to generate the final object boundary with a better shape coding efficiency. The current MPEG4 shape coding scheme applies the macro-block based arithmetic coding. The interframe shape coding uses motion compensation. Our scheme minimizes the interframe shape coding by adopting an "inertia principle", which let the shape of the current frame follow the shape of the previous frame as much as possible. Pixels within the boundary band are classified according to the position of the pixels in the previous frames. If the pixel lies inside the object mask in the previous frame, the pixel is classified as the object pixel in the current frame.

6.2 Experimental Results

We use the experimental results to illustrate the advantage of using the shape postprocessing by using the variable-width boundaries for several MPEG4 test video clips in this section. In comparison with the MPEG4 shape coding algorithm, we can demonstrate 10% to 30% saving for shape coding. First, we encode the shape by using the "IPPP..." coding format, where "I" indicates the intra-coded frame and "P" indicates the predictive coded frame based on the forward prediction. Three shapes were considered in our shape coding experiments. One was the reference shape of the "Akiyo" sequence provided by MPEG-4 while the other two were shapes generated from the "Akiyo" and "Mother and Daughter" sequences by using the fast segmentation algorithm developed in this paper. Results of the bit rate for each frame are shown in Fig. 9, where the circle and the solid line represent the bit rate of the shape without coding optimization. The square and the dotted line represent the bit rate of the shape after coding optimization. The reconstructed 10th frame after predictive coding for "Akiyo" and "Mother and daughter" are compared in Fig. 10. These results are perceptually identical. The total number of bits are compared in Table 1. Results were also obtained for shape coding using the "IBBPBBP..." coding format. The bit rates are shown in Table 2. The improvements in shape coding are still significant.



(a) (b) (c)

Figure 9: Comparison of shape coding results with (dotted line) and without (solid line) coding optimization: (a) the "Akiyo" reference mask, (b) the generated "AKiyo" mask and (c) the generated "Mother and Daughter" mask.

(a) (b) (c) (d)

Figure 10: The reconstructed 10th image frame after shape coding: (a) "Akiyo" without shape coding optimization, (b) "Akiyo" with shape coding optimization, (c) "Mother and Daughter" without shape coding optimization, and (d) "Mother and Daughter" with shape coding optimization.

Sequence	Average Bits / Frame		
	Original	Proposed	Saving
"Akiyo QCIF Reference Mask"	2300	1655	28%
"Akiyo QCIF Generated Mask"	3328	2850	14%
"Mother & Daughter QCIF Proposed Mask"	6041	5277	13%

Table 1: Comparison of shape coding efficiency for three binary masks with the frame coding order of "IPPPP...".

Sequence	Average Bits / Frame		
	Original	Proposed	Saving
"Akiyo QCIF Reference Mask"	2943	2217	25%
"Akiyo QCIF Generated Mask"	4052	3722	8%
"Mother & Daughter QCIF Generated Mask"	6802	5998	12%

Table 2: Comparison of shape coding efficiency for three binary masks with the frame coding order of "IBBPBBP...".

7 CONCLUSION AND FUTURE WORK

A new video segmentation algorithm for the MPEG-4 object based coding was proposed in this work. The proposed segmentation scheme can be implemented very fast. Besides, the color segmentation combined with the region-based motion detection gives very accurate video segmentation results. The final segmented shape is optimized for the coding purpose, which increases the coding efficiency up to 30%. The performance of the proposed segmentation scheme was demonstrated via several experimental results.

The object boundary of a variable width was used as an intermediate result, which served the purpose of linking the segmentation process and the coding stage together. One possible extension of this work is to design a more complicated cost algorithm to achieve a quality scalable shape coding scheme. It is under our current investigation. The variable width boundary can also be used to generate the gray scale object mask, which allows the object to be merged to the background gradually. The gray scale value in the object boundary region can be set according to the spatial gradient variation.

8 REFERENCES

[1] "Information Technology - Coding of Audio-Visual Objects:Visual," Doc.ISO/IEC 14496-2 Final Committee Draft, May. 1998

[2] J. Ohm, Ed.,"Core experiments on multifunctional and advanced layered coding aspects of MPEG-4 video," Doc. ISO/IEC JTC1/SC29/WG11 N2176, May 1998.

[3] C. Gu and M.G. Lee, "Semantic video object segmentation and tracking using mathematical morphology and perspective motion model," *IEEE International Conference on Image Processing*, Santa Barbara, CA, Oct. 1997.

[4] D. Wang, "Unsupervised video segmentation based on watersheds and temporal tracking," *IEEE. Trans. Circuits Syst. Video Technol.*, vol. 8, pp 539-546, Sept. 1998.

[5] P. Bouthemy and E. Francois, "Motion segmentation and qualitative dynamic scene analysis from an image sequence," *Int. Journal of Computer Vision*, Vol. 10, pp.157-182, 1993.

[6] F. Dufaux and F. Moscheni, "Spatio-temporal segmentation based on motion and static segmentation," *IEEE International Conference on Image Processing*, Washington, Oct. 1995.

[7] L. Torres and M. Kunt, *Video Coding (The Second Generation Approach)*, Kluwer Academic Publishers, 1996.

[8] D. Zhong and S.F. Chang, "Video object model and segmentation for content-based video indexing," in *IEEE International Symposium on Circuits and Systems*, Hong Kong, June 1997.

[9] Y. Kanai, "Image segmentation using intensity and color information," in *Visual Communications and Image Processing*, Jan. 1998.

[10] Y. Cheng, "Mean shift, mode seeking, and clustering," in *IEEE Trans. Pattern Anal. Machine Intell.*, Vol.17, pp.790-799, 1995.

[11] D. Comaniciu and P. Meer, "Robust analysis of feature space: color image segmentation," in *Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, June, 1997.

[12] M. Wollborn and R. Mech, "Refined procedure for objective evaluation of video object generation algorithms," Doc. ISO/IEC JTC1/SC29/WG11 M3448, March, 1998.