Fast Object Detection and Segmentation in MPEG Compressed Domain

Orachat Sukmarg and K. R. Rao
Department of Electrical Engineering,
University of Texas at Arlington
Arlington, Texas 76019, USA
Email: oxs3476@omega.uta.edu, krrao@exchange.uta.edu

Abstract: In this paper, we present a fast algorithm for object detection and segmentation in MPEG compressed domain using color clustering, region merging based on spatiotemporal similarities, background/foreground classification, and pixel edge extraction. The features extracted from the blocks of segmented object in compressed domain can be used for fast object tracking and indexing at low level. Moreover, these blocks can be decompressed to obtain details of a specific object in pixel domain and can be used for high level indexing. By using the proposed algorithm, we can reduce the amount of the information needed to be processed, and therefore, save the computational time, and increase the processing speed. Also we need to perform inverse DCT on only some parts of the image.

Keywords
Spatiotemporal segmentation, MPEG compressed domain, graph clustering.

I. INTRODUCTION

The large amount of visual information, handled by the image and video databases, requires effective and efficient search and indexing algorithms. This visual information is usually stored in a compressed form. Therefore, if we want to extract features from the object of interest, we not only have to segment the object and extract its features, but also need to decode the image and video in advance. Since the decoding is a relatively expensive process, segmenting the object and extracting its features directly from the compressed domain would be an effective way to achieve fast and efficient algorithm for searching a large database and indexing the object.

Various segmentation approaches have been investigated and most of them are based on examining the images in pixel domain [1-7]. Lucchese and Mitra [3] used 2D k-means clustering using color information, and then associating these clusters with appropriate luminance values, using 1D k-means algorithm. Ji and Park [4] used the artificial neural networks to merge homogeneous regions based on the information of the luminance, the chrominance difference, the region proximity, and the areas of the regions. Wu and Reed [2] used a region-growing method, a Gibbs-Markov random field (GMRF) model and contour relaxation. Torres et. al. [5] used a motion segmentation algorithm. They partitioned the image into rectangular regions and computed affine motion parameters for each region. These motion parameters are clustered using k-means algorithm to form homogeneous regions with similar motion parameters. Choi et. al. [6] considered spatial and temporal information jointly in their segmentation algorithm. Their algorithm consists of three steps, region simplification, region growing, and motion-based region fusion. Dufaux et. al. [7] used spatiotemporal segmehtation algorithm based on luminance information and motion parameters. The luminance is filtered by morphological operator, and then clustered using k-means algorithm. At the end, regions with similar motions are merged using k-medoid clustering. Moscheni et. al. [1] used the spatiotemporal similarity as their merging criteria. Their spatial similarity is obtained from the test statistic of the gradient value along the boundary of the regions. Their temporal similarity is derived from test statistic of the residual distribution and motion parameters.

Only few researchers have proposed the segmentation algorithm in compressed-domain [8-9]. De Queiroz [8] segmented JPEG documents into specific regions such as those containing halftones, text, and continuous-tone pictures using the EMC-based segmentation. Wang [9] proposed a fast algorithm to automatically detect faces in MPEG compressed video. He used skin-tone statistics, shape constraints, and energy distribution of the luminance DCT coefficients to detect and locate the face position.

In this paper, we propose a fast algorithm to detect and segment objects in MPEG compressed video. Once we locate the object regions, we need to decode only small portions of the video frame back to pixel domain to obtain the detail information of the object. Our segmentation algorithm consists of four main stages, initial segmentation using sequential leader and adaptive k-means clustering, region merging based on spatiotemporal similarities, foreground/background classification, and object detail extraction. The results of segmenting an object are useful and efficient for fast object tracking and indexing at low level. Other feature extraction techniques can be applied on those decoded blocks to obtain detailed features at high level.

0-7803-6355-8/00/$10.00©2000 IEEE

This paper is organized as follows. Section 2 details the proposed segmentation algorithm and pixel-edge extraction. Section 3 presents the experimental results. Finally, we conclude the results of our algorithm and outline future work in section 4.

II. COMPRESSED DOMAIN OBJECT SEGMENTATION

A. Initial segmentation

In our case, the initial segmented regions are generated from 3D spatial information based on dc image and ac energy information. The luminance and chrominance components are *Y, Cb, Cr,* which form the color space used in MPEG sequence. These color and ac-energy information are used to cluster the image using sequential leader clustering to form homogeneous regions without knowing the number of clusters in advance. The sequential leader clustering uses a threshold to decide whether the input data should be in the existing cluster or a new cluster should be created. After we obtain a number of clusters, we apply adaptive k-mean clustering to the image iteratively until no more changes occur in each cluster. The clustered regions with areas less than a threshold are then merged into their neighbors using luminance and ac energy distance, and boundary ratio criteria. The result of this process is the set of initial regions to be used in the spatiotemporal segmentation.

B. Spatiotemporal Segmentation

We use both spatial and temporal information as the criteria for our region-merging algorithm. Spatial information ensures the boundary of the object. Temporal information provides the temporal change characteristics. Both spatial and temporal similarities are measured by the hypothesis test statistic between two adjacent regions and combined into a single value. Regions are merged based on the hypothesis test of acceptance or rejection of the null hypothesis. The advantage of using hypothesis test is that it is less sensitive to noise and false alarms.

For spatial similarity, we calculate the entropy of the ac energy from the luminance information. The entropy value is a measure of uncertainty about the contrast of the partitioned regions. In our assumption, the regions, which are considered as parts of an object, will have high entropy values. The measurements of the entropy values e_A and e_B, of the two adjacent regions (*A* and *B*) are modeled as Gaussian distribution with mean, μ, and standard deviation σ, E_A and E_B (E_A, $E_B \sim N(\mu, \sigma)$). The values of μ and σ are estimated over all the regions in the image.

The measure of spatial similarity is the difference, ($D_E = E_A - E_B$), between these two entropy values, normalized by σ. This difference is a Gaussian random variable with zero mean and $\sqrt{2}\sigma$ ($D_E \sim N(0, \sqrt{2}\sigma)$). The decision of the hypothesis testing is based on the mean value of D_E. Those two regions are spatially similar and can be merged if the difference is zero. The realization, d_e, of the test statistic, D_E, is given by $d_e = e_A - e_B$. The spatial similarity, S_S, between two adjacent regions (A,B), can then be expressed as

$$S_S = 1.0 - \frac{1}{\sqrt{2\pi\sigma}} \int_{-d_e}^{d_e} e^{-\frac{1}{\sqrt{2\sigma^2}}x^2} dx = 1.0 - \frac{1}{\sqrt{2\pi}} \int_{-\frac{d_e}{\sigma}}^{\frac{d_e}{\sigma}} e^{-\frac{t^2}{2}} dt \qquad (1)$$

For temporal similarity, we perform 3D Sobel filter along x-, y-, and t-axes. The sobel operator [13] is shown in Fig. 1.

$$s(t, y, x) = \left(\underset{t}{\boxed{\begin{array}{|c|c|c|}\hline -1 & -3 & -1 \\ \hline -3 & -6 & -3 \\ \hline -1 & -3 & -1 \\ \hline \end{array}}} \quad \underset{t+1}{\boxed{\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}}} \quad \underset{t+2}{\boxed{\begin{array}{|c|c|c|}\hline 1 & 3 & 1 \\ \hline 3 & 6 & 3 \\ \hline 1 & 3 & 1 \\ \hline \end{array}}} \right)$$

Fig.1. 3D Sobel operator.

The resulting sequences contain gradient estimates along their associated dimensions The temporal similarity is derived based on the hypothesis test of the distribution of the temporal gradient. This hypothesis test, in our case, is the well-known Kolmogorov-Smirnov *(K-S)* test [12] which measures the overall difference between two cumulative distribution functions . The Kolmogorov-Smirnov statistic *D* is a simple measure and is defined as the maximum value of the absolute difference between two cumulative distribution functions. For comparing two different cumulative distribution function $F_1(x)$ and $F_2(x)$, the *K-S* statistic is

$$D = \max_{-\infty < x < \infty} \left| F_1(x) - F_2(x) \right| \qquad (2)$$

The temporal similarity is a function of the *K-S* test statistic value and the area of the interested region. This temporal similarity can be written as,

$$S_T = 2\sum_{j=1}^{\infty} (-1)^{j-1} e^{-2j^2\delta^2} \qquad (3)$$

which is a monotonic function with the limiting values

$$\delta = \left[\sqrt{N} + 0.12 + \frac{0.11}{\sqrt{N}} \right] D \qquad (4)$$

where N is the area of the interested region.

In our algorithm, we use two measures of spatiotemporal similarities. One is used to form regions with strong spatiotemporal similarity. The other is used to merge the regions with lower spatiotemporal similarity but high average temporal change within the region. The first spatiotemporal similarity proposed by Moscheni et. al. [1] is written as follow:

$$Sim(A, B) = S_T\,(1- f_L(Max -- S_S)) \tag{5}$$

with $0 \leq f_L \leq 1$, and

$$\begin{aligned} Max &= \max(S_{AE}, S_{EF}), \\ S_{AE} &= \max_{I \in Y_A}(S_{AI}), \\ S_{EF} &= \max_{I \in Y_E}(S_{EI}) \end{aligned}$$

The temporal hypothesis test in [1] is based on motion parameters and residue distribution. However, motion parameters extracted from the MPEG stream do not represent the regions well since they are derived on a macroblock basis. Since computing motion parameters is complex and computationally expensive, we chose to perform the hypothesis test based on the temporal change obtained from the 3D Sobel filter, which is simple and less complex.

We add the second measure of spatiotemporal similarity in order to merge the regions with lower spatiotemporal similarity but high average temporal change. The second spatiotemporal similarity, which combines the average temporal change with spatiotemporal similarities, is

$$Sim(A,B) = \alpha S_T + \beta T_{Avg} + (1-\alpha-\beta)S_S \tag{6}$$

where α and β are weighting factors

$$T_{Avg} = \begin{cases} 1 & if\ \lambda_A, \lambda_B > \tau \\ 0 & otherwise \end{cases} \tag{7}$$

where λ_A and λ_B are average temporal changes in regions A and B respectively.

The average temporal change of each region is thresholded by the overall average temporal change of the entire image, τ. This result is then used as another temporal similarity measure in the segmentation process.

C. Region merging using Graph-based cluster

The spatiotemporal similarities are calculated and used to create a similarity graph between regions. This graph is thresholded and clustered. We perform two clustering stages. First clustering stage is used to merge regions, which form cycles in graph. This process ensures that the merged regions have high spatiotemporal similarities between them. The second clustering stage is used to merge regions based on the number of graph edges connecting between an interested cluster and its neighbor cluster, and those connecting within the interested cluster itself. At each merging stage, the regions are merged and the threshold is reduced. The region similarity and its graph are updated after the second clustering stage. This clustering process is performed iteratively until we reach the minimum threshold or no more new regions are created.

D. Background/Foreground Classification

The result of spatiotemporal segmentation is then applied to foreground/background classification in order to separate background from the objects. The classification decision is based on the average temporal change of regions. The regions with high average temporal changes will be classified as objects. The classification threshold is the average of the difference between maximum and minimum values of the average temporal changes. Even though these features extracted in compressed domain are coarse, it is quite appropriate for fast video browsing and for fast object tracking. If the detail features, such as shape, are required, we can decode DCT coefficients around the boundary of the object and use edge extraction algorithm to obtain the detailed edges in pixel-domain. Since the amount of information to be processed for object segmentation is small, this algorithm can be implemented for fast indexing and object tracking.

E. Object Detail Extraction in Pixel Domain

In some applications, the detail features in pixel domain may be needed such as video indexing and object recognition. The compressed-domain features may not be sufficient. Once we locate the object target region, the actual edges of the object can be extracted by applying Canny Edge Detection only on that region. To obtain object detail, we can project those edges onto those inverse DCT blocks of segmented object. The edges obtained from edge detection algorithm may not be continuously connected. We use our edge-tracking algorithm to find the missing edge pixels. To search and obtain the missing edge pixels, the search direction is based on the direction of the edge of segmented DCT blocks. Even though using this tracking algorithm may sometimes lead to wrong edge position, the result is promising and the time used to obtain object boundaries is small.

III. RESULTS

In this section, we present the segmentation results in compressed domain and edges extracted in pixel domain using our algorithm. The experiments are performed on three different MPEG video sequences, "Akiyo", "Table Tennis", and "News" from MPEG-7 video test set. The video sequences are extracted to obtain the DCT coefficients. The DC image and AC energy are clustered and then the regions with small areas are merged to form reasonable large regions. The criteria, used to merge small-sized regions with the larger ones, are weighted sum of the difference of luminance, AC energy, and boundary ratio between two adjacent regions. The result is used as a set of initial segmented regions. Then we apply Sobel filter to obtain temporal change information, which will be used in the next merging process. The spatial and temporal similarities between the interested region and its neighbor are then computed. We use two measurements of spatiotemporal similarities. The first spatiotemporal similarity has a parameter f_L, which is used to specify how much spatial information we want to associate with temporal information. In our case, we use f_L=0.6 for all three test sequences. The first similarity is used during first stage of the merging process. There are two parameters used in the second similarity, α and β. We use the value of $\beta = 0.2$. However, the value of α starts between 0.3-0.4 and then increases at each clustering process. After computing spatiotemporal similarity, a connection graph is constructed, thresholded and clustered using two stages of the graph clustering. The threshold starts at maximum spatiotemporal similarity over all regions in the image and reduces by T_{step1} for first clustering stage and T_{step2} for second clustering stage. After the second stage, the clustered regions are updated and the clustering threshold is reduced. The clustering process continues until the threshold reaches the minimum threshold value or no more merged regions occur. This threshold used in the first spatiotemporal similarity depends on each video sequence. The minimum value of the clustering threshold used in the second spatiotemporal similarity is 0.1. The original and the segmentation results of DC images are shown in Fig. 2.

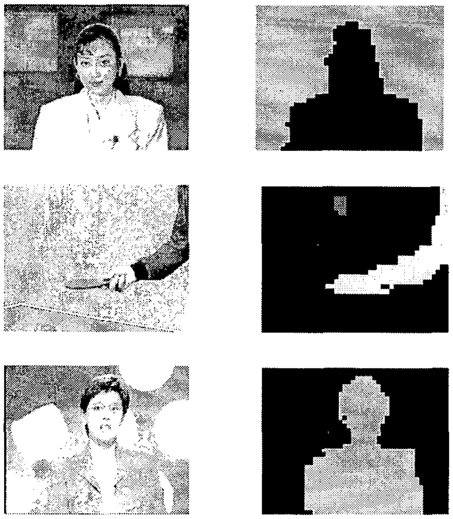

Fig.2. Original images (left) and segmented objects in compressed domain (right).

We show the results of inverse DCT blocks of the entire object area and the blocks at the boundary of the object in Fig. 3.

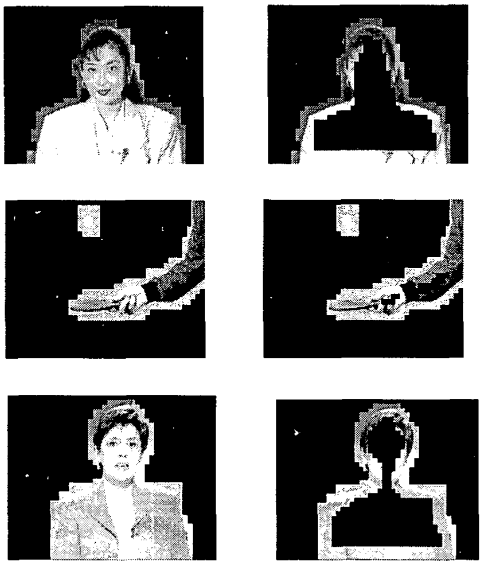

Fig.3. The results of inverse DCT blocks of the entire object area (left) and the blocks at the boundary of the object (right).

The edges obtained from canny edge detection may not be continuously connected. The edge pixels are missing at the areas with low luminance contrast such as Akiyo's hair, the top of table tennis bat, and the right arm of the woman in News clip. The missing edges are tracked according to the direction of the edge of segmented DCT blocks and used for locating the actual position of the objects in pixel domain. Fig. 4 shows the actual position of the segmented objects in pixel domain.

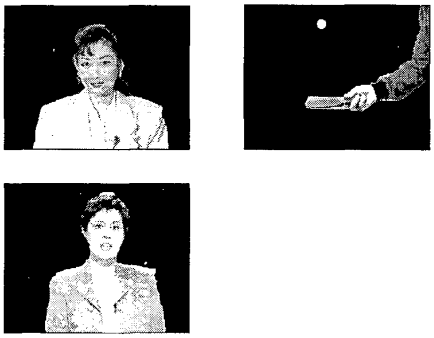

Fig.4. The results of the segmented objects in pixel domain.

From Fig.4, we can see that edges obtained from our algorithm are efficient and accurate, and therefore can be used for indexing or recognizing the object of interest.

IV. CONCLUSIONS

We proposed a segmentation algorithm to segment objects of interest in MPEG compressed domain. Our segmentation algorithm is simple and efficient. Since there is no need to decode the compressed data and only small information is needed to be processed, it requires less processing time, and therefore, increases the processing speed. Even though the features extracted from segmented objects in the compressed domain are coarse, they contain enough information for fast video browsing and object tracking at the low level. However, if the detail information of the extracted objects is required for high level indexing and recognition, only blocks at the segmented regions are decoded and other feature extraction techniques can be performed on these decoded blocks. Also, we can efficiently extract edges of the object by decoding only the blocks at the boundaries of segmented objects. In future work, we will extract useful features from the segmented objects obtained in the compressed domain for low level indexing. Also some decoding may be performed to obtain the detail features for high level indexing.

V. REFERENCES

[1] F. Moscheni, S. Bhatacharjee, and M. Kunt, “Spatiotemporal segmentation based on region merging”, IEEE Trans. Pattern Analysis and Machine intelligence, Vol. 20, pp. 897-915, Sept. 1998.

[2] G. K. Wu and T. R. Reed, “Image sequence processing using spatiotemporal segmentation”, IEEE Trans. circuits and systems for video technology, Vol. 9, pp. 798-807, Aug. 1999.

[3] L. Lucchese and S. K. Mitra, “Unsupervised segmentation of color mages based on k-means clustering in the chromaticity plane”, Proc. of Content-based access of image and video libraries, pp. 74-78, 1999.

[4] S. Ji and H. W. Park, “Image segmentation of color image based on region coherency”, Proc. of ICIP, Vol. 1, pp. 80–83, 1998.

[5] L. Torres, D. Garcia, and A. Mates, “A robust motion estimation and segmentation approach to represent moving images with layers”, Int'l Conf. on Acoustics, Speech, and Signal Processing, Vol. 4, pp. 2981-2984, 1997.

[6] J. G. Choi, S. W. Lee, and S. D. Kim, “Video segmentation based on spatial and temporal information”, Int'l Conf. on Acoustics, Speech, and Signal Processing, Vol. 4, pp. 2661-2664, 1997.

[7] F. Dufaux, F. Moscheni, and A. Lippman, “Spatio-Temporal segmentation based on motion and static segmentation”, Proc. of ICIP, pp. 306-309, 1995.

[8] R. L. de Queiroz, “Processing JPEG-Compressed Images and Documents”, IEEE Trans. Image Processing, Vol. 7, pp. 1661-1672, Dec. 1998.

[9] H. Wang and S. F. Chang, “A highly efficient system for automatic face region detection in MPEG video”, IEEE Trans. Circuit and Systems for Video Technology, Vol.7, pp. 615-628, Aug. 1998.

[10] B. L. Yeo and B. Liu, “On the exaction of DC sequence from MPEG compressed video”, Proc. of ICIP, pp. 260-263, 1995.

[11] K. R. Rao and P. Yip, Discrete Cosine Transform: Algorithms Advantages, Applications. New York: Academic Press, 1990.

[12] W. H. Press et al., Numerical Recipes in C. Cambridge University Press, 1992.

[13] C. Pudney. (Apr. 11, 1995) 3-D Sobel operator. Available: sci.image.processing newsgroup, article 12170.