CONTENT-BASED RETRIEVAL OF VIDEO SEQUENCES UNDER PARTIAL OCCLUSION

Shahram Shirani[†], *Ali Jerbi*[‡], *Faouzi Kossentini*[†], *Rabab Ward*[†] *and Jonathan Wu*[‡]

[†]Dept. of Electrical and Computer Eng.
University of British Columbia
Vancouver BC, V6T 1Z4 Canada

[‡]Vision and Scanning Group
National Research Council
Vancouver BC, V6T 1W5 Canada

ABSTRACT

In this paper, we present a spatio-temporal segmentation method for content-based video retrieval. Our method is effective in the presence of partial occlusion or when new areas become exposed. In addition, periodic spatial segmentation is not performed. Rather, it is invoked as needed depending on whether or not new regions have been detected. A discussion of how this method can be used for content-based retrieval is also provided. Finally, experimental results that demonstrate the performance of the proposed spatio-temporal segmentation method are presented.

1. INTRODUCTION

In order to make use of the vast amount of multimedia information such as text, speech, audio, image and video, effective and efficient methods to access and manipulate this information based on its content are needed. In the case of video, this includes direct access to video frames and sequences of interest, new modes of viewing that give the viewer control over how the video is viewed, annotation and manipulation of objects, description of scenes in the video and merging of text and graphics with the video data. To support such types of interaction, object-based functionalities, such as object-based editing and content-based indexing, are emerging as the basis of the new generation of video coding methods. These functionalities are addressed directly or indirectly in the ongoing MPEG-4 and MPEG-7 standardization activities.

Current video retrieval algorithms parse a video clip into shots with consistent visual content. Using low-level segmentation techniques, the frames are partitioned into arbitrarily shaped regions according to some criterion of coherence such as luminance, color or texture. Then, the regions are combined into meaningful objects by user intervention [3]. These regions are tracked in the subsequent frames, leading to a spatio-temporal segmented video sequence. A content description of the video sequence is then obtained from the segmented video sequence using low-level features. The low-level content description provides a way of finding information in the video sequence during video retrieval [1, 2].

This work was supported by NRC and NSERC.

Most object-based video tracking algorithms have a great difficulty handling occlusion, newly exposed parts, and large object movements. In this paper, we propose a spatio-temporal segmentation method that addresses the above problems. The rest of the paper is organized as follows. In Section 2, we present the proposed method. In Section 3, object-based video retrieval based on the proposed method is discussed. Finally, in Sections 4 and 5, we present some experimental results and conclusions, respectively.

2. PROPOSED METHOD

For the spatial segmentation stage of the proposed spatio-temporal segmentation method, we employ a block-based maximum likelihood (ML) algorithm, followed by a region connecting technique. Here, image smoothness constraints as well as the statistical distribution of the gray levels are exploited. Using the user's assistance, semantic objects which are normally composed of regions can be determined. Tracking is then used to segment the subsequent frames, as described in [1, 3]. Most tracking algorithms have difficulties dealing with occlusion, newly exposed regions, and with motion of large regions. In order to address these difficulties, we propose to partition the regions belonging to a semantic object into windows, select windows that are suitable for tracking and then track these windows in the following frames.

An important part of any tracking algorithm is the motion detection stage. For detecting the motion of windows, we assume a pure translational model and employ a block-matching algorithm. For the specific windows where this model fails, an affine motion model is assumed. This will resolve the problems associated with region-based motion estimation methods, in particular, finding a single set of motion parameters for a region and coping with partially occluded and newly exposed regions. Our tracking algorithm eliminates the need for periodic spatial segmentation. Rather, spatial segmentation is invoked as needed depending on whether or not new uncovered regions have been detected. The idea of using windows for the purpose of improved robustness with respect to occlusion has been proposed in [4], albeit in the context of object recognition. In this work, windowing is exploited for tracking purposes.

As a result of spatio-temporal segmentation, each frame of the video sequence is decomposed into regions. For each region, a set of features including color, location, size, and

0-7803-5467-2/99/ $ 10.00 © 1999 IEEE

motion are obtained. Such features can be used in a retrieval application. A detailed description of the various stages of the proposed method is presented below.

2.1. Block-based Spatial Segmentation

The segmentation of an image involves partitioning the image into fragments. Each fragment is homogeneous in some sense. For segmentation, one can adopt various homogeneity criteria such as spatial homogeneity, temporal homogeneity, and semantic meaning (i.e., the segmentation result should be a meaningful entity). Among all homogeneity criteria, the spatial homogeneity criterion yields the most accurate segmentation results [5]. Therefore, it is important to adopt a good spatial homogeneity criterion. After spatial segmentation, other homogeneity criteria are exploited to finally determine the object in the video sequence.

The spatial segmentation algorithm proposed here is based on an iterative approximate Maximum Likelihood (ML) method. The image is partitioned into 8×8 non-overlapping blocks. Similar to the method proposed in [5], it is assumed that the block corresponds to either a homogeneous region (monotone or texture) or an edge region. As a criterion for homogeneity, the block's variance is compared to an experimentally determined threshold. If the variance is less than the threshold, the block is considered homogeneous. Otherwise, it is declared an edge block. However, in contrast to the method followed in [5], we do not assume any predetermined edge orientation in the block. Instead, it is assumed that the pixel values in the block are the output of a mixture of two independent identically distributed Gaussian random variables. That is, given a set of N mutually independent intensity values $I(x,y)$ within the processing window W, we assume that they are composed of two classes characterized by their Gaussian probability density functions, namely $p_1(I(x,y) \mid m_1, \sigma_1)$ and $p_2(I(x,y) \mid m_2, \sigma_2)$. Our objective is to partition the data set W of size N into two mutually exclusive sets W_1 of size N_1 and W_2 of size N_2 according to the two distributions [1].

In this work, we adopt a fast approximate ML method to solve the above problem [6]. First, the data in W is partitioned into two groups such that the sum of the variances of the groups is minimized, which in turn leads to the K-means clustering of the two groups. In other words, two disjoint subsets of W, named W_1 and W_2, are constructed in such a way that the following cost function

$$J = d_1 + d_2$$

is minimized, where

$$\begin{aligned} d_1 &= \sum_{I \in W_1} I^2 - \frac{1}{N_1}(\sum_{I \in W_1} I)^2, \text{and} \\ d_2 &= \sum_{I \in W_2} I^2 - \frac{1}{N_2}(\sum_{I \in W_2} I)^2. \end{aligned}$$

It can be shown that W_1 and W_2 are necessarily contiguous subsets of W. Therefore, the partitioning phase is done as follows. The data in the window W is initially sorted and the cost function J is computed for $N_1 = 1, 2, 3, .., N$ and $N_2 = N - N_1$, to find the value of $N_{1,min}$ which minimizes J, i.e.,

$$N_{1,min} = argmin(J).$$

The first $N_{1,min}$ elements in the rank ordered data in the window W belongs to W_1, and the rest of the data belongs to W_2. The computation of J for various values of N_1 can be done recursively, since at each step, a single data element is removed from one of the subsets of the sorted W and the same data is added to the other subset. The following recursive equation can be used for computing the cost function d_1 when N_1 increases from L to $L+1$ with $1 \leq L < N$:

$$\begin{aligned} & m_{L+1} = \frac{Lm_L}{L+1} + \frac{I_{(L+1)}}{L+1}, \\ & d_{L+1} = d_L - (L+1)m_{L+1}^2 + Lm_L^2 + I_{(L+1)}^2, \text{and} \\ & m_1 = I_{(1)}, d_1 = 0, \end{aligned}$$

where $I_{(L)}$ is the Lth data in the sorted W. Similar equations can be easily obtained for d_2. After the partitioning is performed, pixels in each of the sets W_1 and W_2 are replaced with their corresponding estimated average values. The ML estimation of the parameters of the density functions of the classes is obtained using the following equation

$$\begin{aligned} \hat{\phi} &= argmax \prod_{I(x,y) \in W} p(I(x,y) \mid \phi), \text{ or} \\ \hat{\phi} &= argmax \prod_{I(x,y) \in W} \sum_{i=1,2} p(W_i) p(I(x,y) \mid \phi, I \in W_i), \end{aligned}$$

where

$$p(I(x,y) \in W_i \mid W_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} exp(-\frac{(I(x,y) - m_i)^2}{2\sigma_i^2}), \quad i = 1, 2.$$

The regions with similar means in adjacent blocks are then combined. That is, a candidate region is combined with its neighboring regions if their mean difference is below a certain experimentally determined threshold. Similar connected regions are combined using a watershed type algorithm [7, 8]. Moreover, a post-processing stage removes all regions with a small size. In other words, if the total number of pixels in a connected homogeneous region is less than another threshold, the region is merged into the adjacent regions with the closest mean value.

Normally, a single homogeneous color or motion segmentation does not lead to a satisfactory semantic object extraction. To overcome this problem, the segmentation procedure requires assistance of the user in determining semantic objects. This can be done, for example, by selecting all the regions that correspond to an object or drawing a line around the object in the segmented image. In the next section, we propose a method to track the obtained regions and/or semantic object in a video sequence.

[1] Obviously, $N = N_1 + N_2$.

2.2. Block-based Region Tracking

After the first frame of a video sequence has been segmented, tracking is performed. The objective of tracking is to segment the subsequent frames and to establish a correspondence of regions between frames. Several proposed tracking algorithms share the following essential elements: a motion estimation procedure, a projection step and an adjustment step. The motion estimation procedure is usually based on a model for the motion (e.g., linear [9], affine [1], planar perspective [3]) and an associated method for estimating the model parameters (e.g. hierarchical block matching [9], iterative nonlinear methods [3], Newton method [1]). In all these motion estimation methods, the goal is to find the motion vectors that map a region in the present frame to a region in the next frame such that the mapping error, or a function of it, is minimized. Motion estimation methods that include mapping of the whole region often have some drawbacks. Almost all these methods have difficulties dealing with uncovered and overlapped (occluded) regions and tracking of large objects. In addition, the ones that use conventional block matching techniques are very sensitive to noise.

In order to be able to track regions under partial occlusion situations, we divide the region into small windows and track each window in the following frame. Hence, local features are used for tracking instead of global features. Given that the number of windows may be large and some of the windows may not be suitable for tracking, we use a pruning technique to select the optimal set of windows to be tracked. For example, a window containing the corners of an object is much easier to track than those containing uniform gray-levels. It is expected that the boundary of a region will be contained in the windows that are to be tracked. In order to identify the windows that should be tracked, we adopt the approach proposed in [10] and [4], where the tractability matrix G is given by

$$G = \sum (\frac{\partial I}{\partial X})(\frac{\partial I}{\partial X})^T,$$

where I is the intensity, $X = [x, y]^T$ and x and y denote the pixel coordinates. This matrix G has two eigenvalues λ_1 and λ_2. The window is accepted as a tractable one if

$$min(\lambda_1, \lambda_2) > \lambda,$$

where λ is an experimentally determined threshold.

Once the optimal set of windows are identified, the best match for each window in the next frame is sought. Each window is typically small as compared to the whole region. Therefore, a simple pure translation motion model is first assumed and a block matching algorithm is applied. If the error between the best match and the window exceeds a threshold, an affine motion model, which is more computationally demanding, is then employed. As a result, the possibilities of rotation and linear deformation are also included in the tracking stage. This is in contrast to [1], where only affine motion is employed. In the case of the affine model, the parameters of the following model are estimated using the pixels in the window, i.e.,

$$x' = a * x + b * y + c, \quad (1)$$

$$y' = d * x + e * y + f, \quad (2)$$

where (x,y) are the coordinates of a pixel in the current window, (x', y') are the pixel's map in the next frame and a, b, c, d, e, and f are the parameters of the affine model. The parameters of the model are determined by minimizing the prediction error within the window, or

$$e = \sum_{(x,y) \in B} (I_{k-1}(x, y) - I_k(x', y'))^2,$$

where I_{k-1} and I_k are the current frame and the next frame respectively and B is the window being considered. The minimization is done using an iterative gradient descent method. If the prediction error remains above the threshold even after using the affine model, the window is considered as part of a newly exposed region. As a consequence of the windowing approach, our tracking technique is able to deal with new regions appearing within a new frame, unlike the methods described in [1, 3].

All the pixels in I_k that are not matched are considered new and are spatially segmented using the method proposed in Section 2.1. To do this, a flag matrix whose size is equal to that of the frame I_k is defined. If a specific pixel in I_k is matched during the tracking stage, its corresponding flag in the flag matrix is set. At the end of tracking, the flag matrix is AND'ed with I_k, so that all the pixels of I_k that are not matched during tracking are placed in a separate image. The obtained image, which is mostly zeros except for uncovered parts and/or new objects appearing in the frame, is spatially segmented using the proposed method.

An important characteristic of our proposed spatio-temporal segmentation method is that the spatial and temporal segmentation operations are carried out simultaneously. Spatial segmentation of the new parts of the image is invoked during the processing of any frame whenever these parts appear. Consequently, to keep track of the changes in the video sequence, we do not need to perform spatial segmentation once every several frames unlike what is proposed in [1].

3. OBJECT-BASED VIDEO RETRIEVAL

After spatio-temporal segmentation, each frame in the video sequence is decomposed into regions. For each region, a set of features which are used in retrieval application are computed. The features that are presently considered in this work are:

- Color: the color histogram of the region,
- Location: the region's centroid,
- Size: the total number of pixels in the region, and
- Motion: It is very difficult to obtain a single number for motion of a region. Even affine motion models are difficult to handle because it is hard to define the distance between two sets of affine motion model parameters for a similarity test. Moreover, occlusion might affect the values of the parameters obtained for a region's motion. Thus, we consider the displacement of the region's centroid as the motion feature.

In addition to the above features, the proposed block-based approach establishes a correspondence between all

the windows belonging to a certain object in all the frames. Therefore, if the object becomes occluded in a specific frame and the above mentioned features fail to detect the object, the correspondence of windows between the present frame and the previous frames (where the object is not occluded) may indicate the object's presence.

We next describe how our proposed spatio-temporal segmentation method, along with the features discussed above, can be used for retrieval of video content. During the query, the system is presented with an image of an object or a question. For instance, the query can be: "find all the frames in the video sequence which contain this object" or "find all frames in the video sequence in which an object moves from left to right". The first example refers to an image query where the image of the object is specified and the second one is an example of a question type of query.

In the case of an image query, spatial segmentation is done on the query. As a result of this step, windows that describe each region of the query are obtained. In addition, global features associated with each region such as motion of the centroid of the region, size and shape of the region are acquired. During the retrieval process, the global features of each region of the first frame are matched with those of the query. In the case where the distance metric corresponding to the difference between the global feature of a candidate and the query's is above a certain threshold, the windows associated with regions of the query are then used in matching. This is done to predict the possibility of occlusion of parts of the object.

If the query is a question on one of the global features, all the frames with the appropriate feature(s) are selected. For example, if motion is one of the features selected in the query phase, then the displacement vectors of the regions' centroids are used for matching. Clearly, the user can indicate whatever is (are) the most important feature(s) that would be considered in the retrieval process.

4. EXPERIMENTAL RESULTS

The proposed segmentation and tracking method has been tested using several video sequences. All the video sequences used are in QCIF format at the original frame rate of 30 frame per second. Figure 1 shows three frames of the image sequence AKIYO occluded with an object (a hand) which moves in front of the video frames. Figure 2 shows the spatial segmentation of the first frame and separation of the semantic object. Figure 3(a) shows the windows selected for tracking. Figure 3(b) shows the tracking result. Those windows where good matches cannot be found are shown in white. The algorithm uses the result of the segmentation of previous frames for those blocks where a good match exists as shown in Figure 4(a). For the other blocks (white blocks), a spatial segmentation is performed and the result is shown in Figure 4(b). The figure shows that, although the occlusion is severe, the algorithm is able to track the object. The same procedure was repeated for frames number 40 and 80, and the result of spatio-temporal segmentation are shown in Figures 5(a) and 5(b), respectively. Figure 5(a) shows that parts in frame number 80 that could be matched to some parts in frame number 40. For these parts the segmentation results of frame number 40 can be used to segment frame number 80 as shown in Figure 5(b). Again, although the occlusion is severe, our spatio-temporal segmentation method performs quite well.

5. CONCLUSIONS

In this paper, we presented a spatio-temporal segmentation method for content-based retrieval based on a windowing approach whereby local features are obtained. Unlike previous methods, where it is assumed that all objects remain in the scene within the whole group of pictures (i.e., between two consecutive spatially segmented frames), our method can effectively deal with occluded and new regions that get exposed anywhere within the video sequence. Moreover, the content-based retrieval algorithm that we have discussed not only relies on low-level global features for regions, but also uses windows associated with regions to make the retrieval process more robust. Several experimental results are provided to demonstrate the performance of our spatio-temporal segmentation method.

Acknowledgment

The authors would like to thank Dr. Nancy Heckman from the Statistics department, University of British Columbia, for her constructive suggestions and guidance, and Mr. Dewey Liew from the National Research Council of Canada for preparing the occluded video sequences.

6. REFERENCES

[1] Y. Deng and B. S. Manjunath, "NeTra-V: Towards an object-basd video representation," *IEEE Trans. on Circuits and Systems for Video Technology*, pp. 616–627, Sep 1998.

[2] S. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong, "A fully automated content-based video search engine supporting spatiotemporal queries," *IEEE Trans. on Circuits and Systems for Video Technology*, pp. 602–615, Sep 1998.

[3] C. Gu and M. C. Lee, "Semiautomatic segmentation and tracking of semantic video objects," *IEEE Trans. on Circuits and Systems for Video Technology*, pp. 572–584, Sep 1998.

[4] K. Ohba and K. Ikeuchi, "Detectability, uniqueness, and reliability of eigen windows for stable verification of partially occluded objects," *IEEE Trans. Pattern Anal. and Mach. Int.*, pp. 1043–1048, Sept 1997.

[5] C. S. Won, "A block-based MAP segmentation for image compression," *IEEE Trans. on Circuits and Systems for Video Technology*, pp. 592–601, Sep 1998.

[6] J. Zhang and J. W. Modestino, "A model-fitting approach to cluster validation with application to statistical model-based image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 1009–1017, October 1990.

[7] P. Salembier and M. Pardas, "Hierarchical morphological segmentation for image sequence coding," *IEEE*

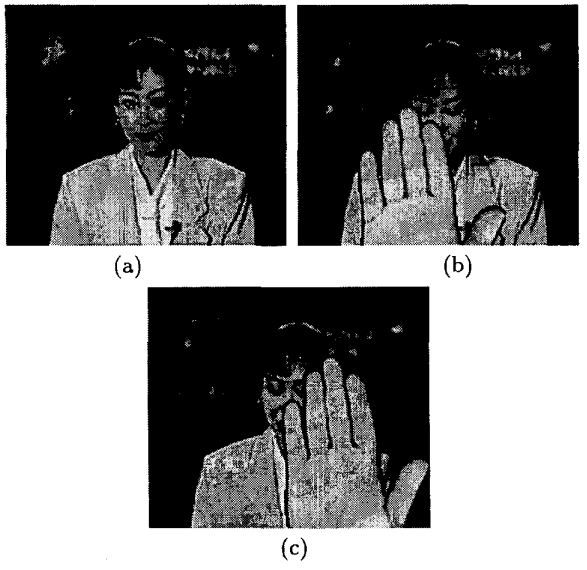

Figure 1: The video sequence AKIYO occluded using an extra object. The object moves across the video sequence and occludes various parts. Figure (a), (b), and (c) show frame number 1, 40, and 80, respectively.

Figure 2: The spatial segmentation of the first frame and separation of the semantic object.

Trans. on Image Processing, pp. 639–651, September 1994.

[8] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 583–598, June 1991.

[9] D. Wang, "Unsupervised video segmentation based on watersheds and temporal tracking," *IEEE Trans. on Circuits and Systems for Video Technology*, pp. 539–546, Sep 1998.

[10] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp. 593–600, 1994.

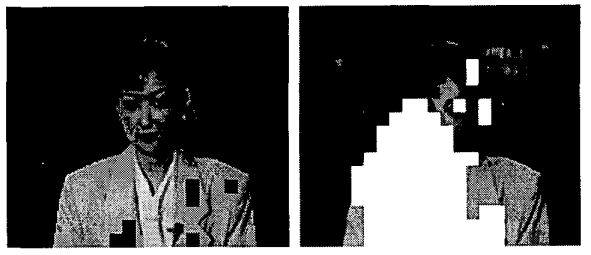

Figure 3: (a) The windowing and pruning of the object in the first frame. All the 8×8 windows, except the black ones, are used for tracking. (b) The result of tracking of the semantic object of frame number 1 in frame number 40.

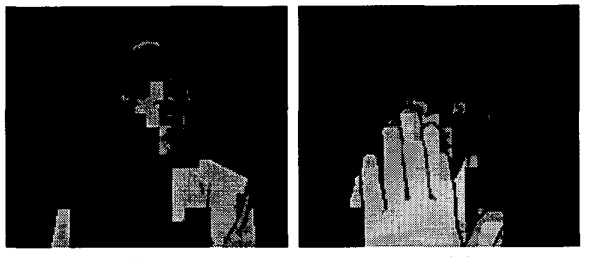

Figure 4: (a) Temporal segmentation of the object in frame number 40 based on frame number 1. (b) Spatial segmentation of the new parts in frame number 40.

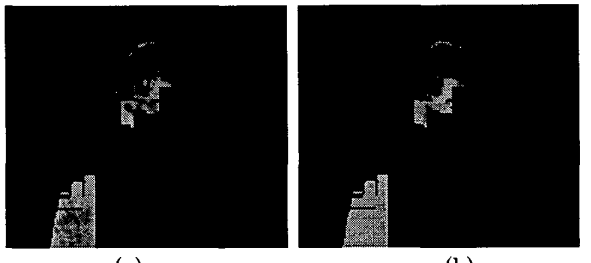

Figure 5: (a) Tracking of the semantic object between frames number 40 and 80 in the video sequence AKIYO. (b) Temporal segmentation of frame number 80 using segmentation results of frame 40 in the video sequence AKIYO.