Automatic Spatio-Temporal Video Sequence Segmentation

Jozsef Vass Kanappan Palaniappan Xinhua Zhuang

Multimedia Communications and Visualization Laboratory
Department of Computer Engineering & Computer Science
University of Missouri-Columbia
Columbia, MO, 65211
E-mail: {vass,palani,zhuang}@cecs.missouri.edu

Abstract

In the paper, an automatic spatio-temporal video sequence segmentation algorithm is proposed. To address this very difficult computer vision problem, several novel algorithms have been developed which use both spatial and temporal information. First, a novel temporal segmentation algorithm is developed based on our previous work in motion estimation. Second, an iterative split-and-merge spatial segmentation scheme is proposed with an initial segmentation being provided by recursive conditioned dilation operation merging pixels into homogeneous regions, followed by an iterative refinement algorithm to obtain the final spatial segmentation. Third, temporal and spatial segmentation are linked to form spatio-temporal segmentation with the shape of each object being simplified by a morphological close-opening operation to obtain the final segmentation. Performance evaluation shows that the developed algorithm can successfully segment moving object without any human intervention.

1 Introduction

In the past decade, we have witnessed an impressive progress of video coding algorithms being developed in both academia and industry. The importance of visual communications is also evidenced by several video coding standards being adopted. The most innovative and important feature of the forthcoming MPEG-4 standard, perhaps, is its object-based functionality such as content-based manipulation and bitstream editing, content-based scalability, etc. [1]. Thus the scene is composed from independently encoded video objects [2], which requires encoder-side segmentation of the input frames. Similar to MPEG-1/2, MPEG-4 is generic, i.e., no segmentation algorithm is defined at the encoder, it is assumed that the segmentation information is available at the decoder. For synthetic video sequences, the segmentation information (the so-called alpha channel) is available, but for natural video sequences, real-time, semantically meaningful segmentation is required. Hence, automatic video segmentation is a very active research area in both the computer vision and coding community evidenced by the large number of algorithms being developed [3, 4, 5, 6].

In the paper, a generic, automatic spatio-temporal video sequence segmentation algorithm is developed. The algorithm is fully automatic, i.e., no human intervention is required. The algorithm is also generic, i.e., no a priori knowledge of the scene or camera is assumed. Performance evaluation of the algorithm shows its efficiency in moving object segmentation.

The rest of the paper is organized as follows. Section 2 describes each building block of the developed algorithm. Performance evaluation is given in Section 3. The last section concludes the paper and gives further research directions.

2 Spatio-Temporal Segmentation Algorithm

The proposed automatic spatio-temporal segmentation algorithm consists of four main building blocks:

- temporal segmentation;
- spatial segmentation;
- linking of temporal and spatial segmentation; and
- shape processing.

In the followings, the 180th frame of the "Hall Monitor" sequence will be used to demonstrate each step of the algorithm. The original frame is shown in Fig. 1.

2.1 Temporal Segmentation

Temporal segmentation is the key part of the proposed spatio-temporal segmentation algorithm. Due

0-8186-8821-1/98 $10.00 © 1998 IEEE

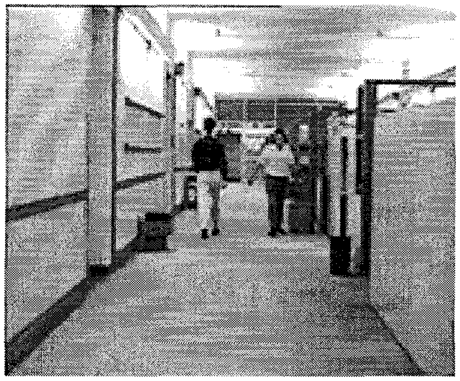

Figure 1: Original 180th frame of the "Hall Monitor" sequence.

to its simplicity, the interframe difference is traditionally utilized to detect moving components in the frame [3] with a sacrifice of the accuracy of the moving object boundaries. In the proposed algorithm, temporal segmentation is performed by calculating the so-called motion-prediction (MP) index for each pixel defined as follows: It takes binary one when pixel (x, y) is not matchable in the previous frame, otherwise it takes binary zero. It is calculated (following the spirit of [7]) for each pixel (x, y) as follows: First, to avoid unreliable matches, an initial motion vector field is estimated by using full search block matching algorithm (FSBMA) with block size of 16×16 pixels and ± 15 pixels search range. Large block size is desirable in order for FSBMA to capture the true motion and avoid trapping in local minima. Then, this initial estimate is refined for each pixel (x, y). Let f_t and f_{t-1} denote the current and previous frames, respectively. A local patch $\mathcal{G}(x, y)$ of 5×5 pixels is established around each pixel (x, y). Then, the mean-squared prediction error is determined by minimizing

$$\mathrm{RMSE}(x,y) = \left(\min_{k,l} \frac{1}{\#\mathcal{G}(x,y)} \sum_{(m,n)\in\mathcal{G}(x,y)} (f_t(m,n) - f_{t-1}(m - d_x - k, n - d_y - l))^2 \right)^{\frac{1}{2}}$$

where (d_x, d_y) is the initial motion vector of pixel (x, y), and (k, l) is the motion vector refinement determined by full search algorithm with a much reduced search range. The MP index is then obtained by thresholding $\mathrm{RMSE}(x, y)$ with a motion threshold T_M,

$$\mathrm{MP}(x,y) = \begin{cases} 1 & \text{if } |\mathrm{RMSE}(x,y)| > T_M \\ 0 & \text{otherwise.} \end{cases}$$

Finally, connected component analysis [8] is applied to the binary MP index map and small components (corresponding to noise) are removed.

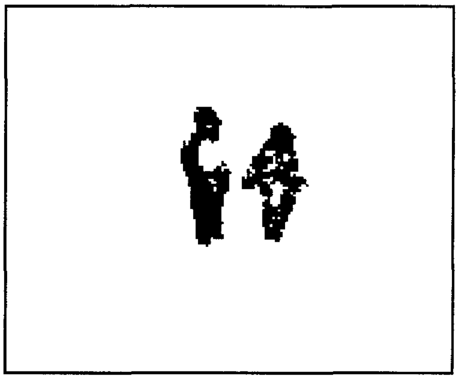

Figure 2: Result of temporal segmentation.

Lack of a good match from the previous frame results in $\mathrm{MP}(x, y) = 1$. This includes 1) new object is coming into the scene, i.e., pixels belonging to the new object do not have a good match in the previous frame; 2) uncovered background areas, i.e., pixels belonging to these areas do not have a good match in the previous frame where they were covered by the object; and 3) moving texture areas since texture areas have a high intensity variance and even a small deviation from the true motion can yield a large mean-squared error. Although $\mathrm{MP}(x, y) = 1$ means that pixel (x, y) belongs to a moving object, which is present in the current frame or in the previous frame or in both frames, $\mathrm{MP}(x, y) = 0$ does *not* necessarily imply that pixel (x, y) belongs to a still object, e.g., for interior pixels of moving homogeneous regions good match from the previous frame is readily available. The proposed temporal segmentation is illustrated in Fig. 2. Both persons have been correctly detected. Also note that moving homogeneous areas (e.g., back of the left person) have been incorrectly classified.

2.2 Spatial Segmentation

The next building block is spatial segmentation. As was pointed out before, temporal segmentation fails for moving homogeneous regions. Thus the goal of spatial segmentation is to delineate homogeneous regions for which an iterative variant of split-and-merge scheme [8] is developed. In the proposed algorithm,

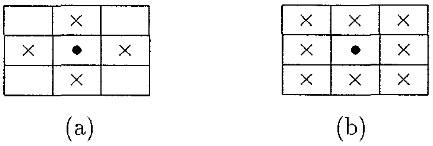

Figure 3: (a) Cross structuring element used in conditioned dilation for initial cluster detection. (b) 3×3 structuring element used in shape simplification. • and × denote the origin (included) and the support of the structuring element, respectively.

each pixel is labeled by a non-negative integer. Label 0 means that pixel (x, y) is unlabeled, i.e., it does not belong to any region. Initial segmentation is achieved by conditioned dilation operation [9] by using star (4-connected) structuring element B shown in Fig. 3a. An *irregular-shaped* region is recursively formed by appending unlabeled boundary pixels to the region if the given homogeneity criterion, i.e., the absolute difference between the region mean (mean value of pixels belonging to the same region) and the value of the pixel to be added is smaller than T_I, is satisfied:

```
BEGIN Initial-Segment()
Step 1. Given the homogeneity threshold $T_I$. Set the
    region counter $k \leftarrow 0$. Each entry of the labeling
    matrix is initialized to zero.
Step 2. Scan the image from top to bottom, left to
    right.
Step 3. If pixel $(x, y)$ is unlabeled (not assigned to
    any region) then
    $k \leftarrow k+1$. Let $(x, y)$ be the seed of region $R_k$, and
    region mean $m(R_k) = f(x, y)$. Region growing by
    conditioned dilation. Expand-Region$(x, y)$
END Initial-Segment()

BEGIN Expand-Region$(x, y)$
Step 1. For each $(i, j) \in B$ do
    If $f(x+i, y+j)$ is unlabeled and $|m(R_k) - f(x+i, y+j)| < T_I$ then
        Append pixel $(x+i, y+j)$ to region $R_k$ and
        recursively update the region mean $m(R_k)$.
        Expand-Region$(x+i, y+j)$
BEGIN Expand-Region()
```

The above initial segmentation algorithm has very low computational complexity but tends to result in inhomogeneous regions since the mean value of the region is recursively updated. (A region is said to be inhomogeneous if the region variance, i.e., variance of pixels belonging to the same region, is larger than a given homogeneity threshold.) The following iterative algorithm is developed to improve the performance of the initial segmentation algorithm:

Step 1. *Inhomogeneous region elimination.* Inhomogeneous regions resulting from initial segmentation or from previous iteration of this algorithm are eliminated by setting the label of the corresponding pixels to zero.

Step 2. *Resegmentation.* Unlabeled pixels resulting from the previous step are resegmented by using the initial segmentation algorithm. The homogeneity threshold is chosen so that the homogeneity of the new regions is guaranteed. Thus after resegmentation, all regions are homogeneous.

Step 3. *Region merging.* Resegmentation tends to oversegment the image, i.e., one homogeneous region might be segmented into two or more regions. Two neighboring regions are merged if the absolute difference between the mean value of the two corresponding regions is inferior to a given threshold.

Step 4. *Small region elimination.* To further reduce the number of regions, regions having area smaller than a given area threshold are eliminated by setting the labels of the corresponding pixels to zero.

Step 5. *Region appending.* The purpose of region appending is to assign every unlabeled pixel resulting from the previous step to an already *existing* region. The procedure starts with a small homogeneity threshold. An unlabeled pixel is appended to the most similar *neighboring* region if the difference between the region mean and the value of the unlabeled pixel is smaller than the given homogeneity threshold. After all the pixels for the given homogeneity threshold have been appended, the threshold is incremented and the procedure is repeated until all the pixels are labeled. Region appending will not increase the number of regions, but might result in some homogeneous regions become inhomogeneous.

Step 6. If the specified number of iterations is reached terminate, otherwise go to **Step 1.**

Computer experiments show that two iterations of the above algorithm gives good performance. The obtained spatial segmentation is shown in Fig. 4, where the image has been segmented into 392 regions and each region is replaced by the corresponding region mean value.

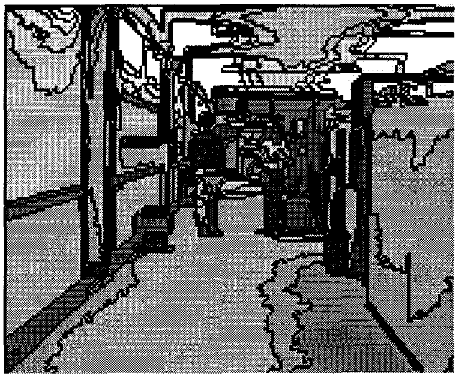

Figure 4: Result of spatial segmentation.

2.3 Linking of Spatial and Temporal Segmentation

Spatial and temporal segmentation are linked to form spatio-temporal segmentation. First, a translational motion vector is determined for each region R_i by using full search polygon matching algorithm [10] with ±15 pixels search range in both vertical and horizontal directions. To avoid false matches only regions having area larger than a given threshold are considered. Then, region R_i is classified as *moving region* if 1) it has non-zero motion vector and 2) part of region R_i belongs to moving object signaled by MP index, i.e., $\text{MP}(x, y) = 1$ for some $(x, y) \in R_i$. The result of spatio-temporal segmentation is shown in Fig. 5. Homogeneous regions missed by temporal segmentation are correctly identified. Since regions with small area are excluded from the spatio-temporal linking, objects might contain small holes.

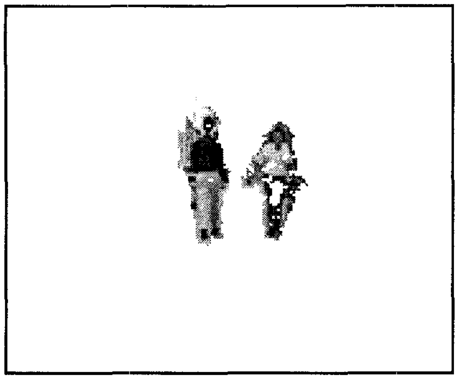

Figure 5: Result of spatio-temporal linking.

2.4 Shape Processing

The final shape processing contains two operations, hole filling and shape simplification. The shape of each object is simplified by morphological close-opening operation [11] by using a flat structuring element of size 3 × 3 shown in Fig. 3b. The finally obtained spatio-temporal segmentation is shown in Fig. 6. Note that both moving objects have been correctly identified.

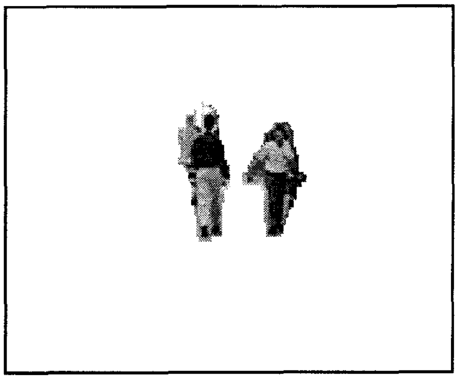

Figure 6: The final spatio-temporal segmentation after shape processing.

3 Performance Evaluation

The performance of the proposed automatic spatio-temporal segmentation algorithm is demonstrated on the "Hall Monitor" sequence in QCIF resolution starting at the 138th frame with 5 frame-per-second. As shown in Fig. 7, both moving objects are correctly segmented.

4 Conclusions

In the paper, a very efficient, generic, spatio-temporal segmentation algorithm is developed. We have also demonstrated that the proposed algorithm can succesfully segment moving objects. Further research directions include integration of temporal object tracking and development of an object-scalable video compression scheme based on the proposed automatic spatio-temporal segmentation algorithm.

References

[1] T. Sikora, "The MPEG-4 video standard verification model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 19–31, Feb. 1997.

[2] MPEG Video Group, "MPEG-4 video verification model," July 1997, Doc. ISO/IEC/JTV1/ SC29/WG11 N1796.

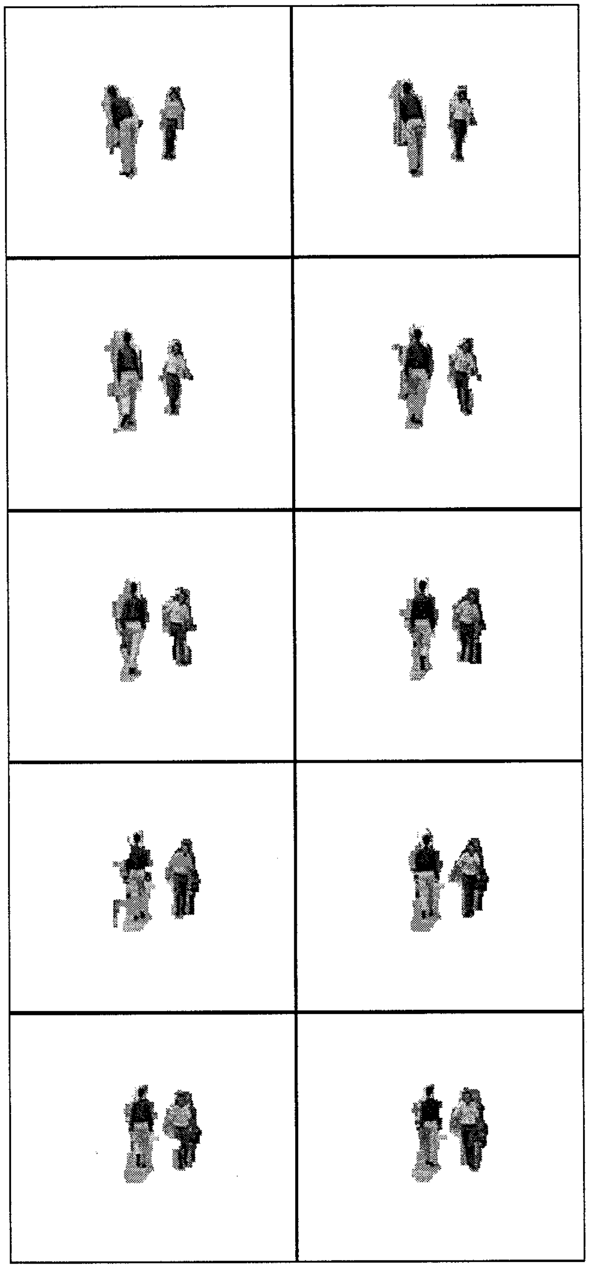

Figure 7: Result of the proposed automatic spatio-temporal segmentation algorithm of the "Hall Monitor" sequence starting at the 138th frame with 5 frame-per-second.

[3] D. Wang, C. Labit, and J. Ronsin, "Segmentation-based motion compensated video coding using morphological filters," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 3, pp. 549–555, June 1997.

[4] L. Torres and M.Kunt, Ed., *Video Coding. The Second Generation Approach*, Kluwer Academic Publishers, Norwell, MA, 1996.

[5] P. Salembier, F. Marqués, M. Paradàs, J.R. Morros, I. Corset, S. Jeannin, L. Bouchard, F. Meyer, and B. Marcotegui, "Segmentation-based video coding system allowing the manipulation of objects," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 60–74, Feb. 1997.

[6] C. Gu, T. Ebrahimi, and M. Kunt, "Morphological moving object segmentation and tracking for content-based video coding," in *Multimedia Communications and Video Coding.* 1996, Plenum, New York, NY.

[7] Y. Huang and X. Zhuang, "Two block-based motion compensation methods for video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 1, pp. 123–126, Feb. 1996.

[8] R.M. Haralick and L.G. Shapiro, *Computer and Robot Vision*, Addison-Wesley, 1992.

[9] L. Vincent, "Morphological grayscale reconstruction in image analysis: Applications and effective algorithms," *IEEE Transactions on Image Processing*, vol. 2, no. 2, pp. 176–201, Apr. 1993.

[10] M.-C. Lee, W.-G. Chen, C.-L.B. Lin, C. Gu, T. Markoc, S.I. Zabinsky, and R. Szeliski, "A layered video object coding system using sprite and affine motion model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 130–145, Feb. 1997.

[11] R.M. Haralick, S.R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 4, pp. 532–550, July 1987.