VIDEO REGION SEGMENTATION BY SPATIO-TEMPORAL WATERSHEDS

M. A. El Saban and B. S. Manjunath

Electrical and Computer Engineering Department, University of California, Santa Barbara
{msaban, manj}@ece.ucsb.edu

Abstract

We propose a video region segmentation scheme combining spatio-temporal edges and watershed techniques. We consider the video sequence as a 3-D volume and compute color edges within this volume. These color edges form a vector field that is in turn used to obtain an edge function. This edge function is used as a topological surface for a watershed grouping stage. Considering the video as a 3-D volume results in a batch segmentation instead of the traditional frame-by-frame segmentation. The main advantages of this approach are: 1) exploiting the time continuity in the frame sequence , 2) avoiding problems in tracking regions from frame to frame, 3) using a fast watershed-based method in forming the final video regions. Preliminary experimental results are very promising.

1. INTRODUCTION

The task of segmenting a video sequence into its constituting objects/regions, and computing their motion profiles is an important first step in many computer vision systems. We propose a new technique for video segmentation based on treating the video sequence as a spatio-temporal (ST) volume (image + time dimensions). ST edge vectors are computed from the image features, namely RGB color. In this paper, we propose a novel method to achieve ST segmentation, by using ST watersheds. The ST edge vectors are used to compute a gradient function forming the topological surface of the watershed method [1]. Interesting features of the proposed segmentation method are:

- It avoids the use of heuristics boundary linking methods. Besides, it eliminates the need of frame-to-frame region tracking.
- Watershed techniques are implemented at a low computational cost.

This research was supported in part by the following grants and awards: ONR # N00014-01-0391, NSF Instrumentation # EIA-9986057 and NSF Infrastructure # EIA-0080134. We also like to thank Charless Fowlkes for providing the source code for the NCut segmentation technique.

Several researchers have addressed the spatio-temporal segmentation problem. These approaches can be broadly classified into two main categories: a) techniques that compute the segmentation through frame-by-frame analysis, and then link regions between frames, and b) techniques that consider the sequence of frames as being an ST volume (xyt volume), and solve a 3D segmentation problem. Example of segmentation using frame-by-frame analysis include Moscheni et al [2], where they merge regions from two adjacent frames using motion and spatial similarity, starting with over segmented frames. Patras et al. [3] perform video segmentation by first computing frame-by-frame watershed segmentation. This provides a number of initial segments which are subsequently labelled, according to motion information and previous frame labelling.

Tao et al. [4] use the concept of dynamic motion layers, which produces several independent motion layers in the sequence. Layer parameters and layer membership are estimated simultaneously using a generalized EM algorithm and incorporate prior knowledge from the previous frame, hence the name dynamic layers. Khan and Shah [5] use multiple cues, such as motion and color, in segmenting the video frames. They find class labels for video pixels using a maximum likelihood (ML) approach, and adjust a set of weights for the ML function using confidence measure on the extracted optical flow.

Examples of the second approach, treating the video as a spatio-temporal volume, include extensions of the normalized cut framework to video sequences by Shi and Malik [6] and Fowlkes et al. [7]. In these works, the problem is formulated as a graph-partitioning problem with voxels as nodes and weights reflecting similarities between voxels. Kompatsirias and Strintzis [8] modified the k-means clustering technique to impose spatio-temporal connectivity, and performed spatio-temporal segmentation on the whole volume again using intensity and motion cues. Korimilli and Sarkar [9] computed motion segmentation by first detecting 3D edges in the xyt volume, then fitting spatio-temporal envelope planes using Hough transform, and finally grouping similar planes using Gestalt principles. In [10], Hung et al. achieve video segmentation by merging 3D watersheds in the spatio-temporal domain using a Markov random field

(MRF) framework. Although the merging is done directly in the spatio-temporal domain, the 3-D watersheds are computed through frame-by-frame analysis.

Section 2 presents the details of the ST edge vector computation. Section 3 describes the segmentation method using watershed-based algorithms. Section 4 gives experimental results of the proposed technique. Finally, conclusions and future work are presented in section 5.

2. SPATIO-TEMPORAL EDGE VECTORS COMPUTATION

Consider a voxel $s = (x, y, t)$ in the ST volume. We are seeking to find the edge energy and direction at each voxel in this volume. In computing the edge information, we use ST color cue. A set of 3-D kernels is used to filter the ST volume to generate the edges. Consider a Gaussian derivative (GD) kernel with scale parameter $\sigma = (\sigma_s, \sigma_t)$:

$$GD_\sigma(x, y, t) = -\frac{x}{\sigma_s^2} \frac{1}{(2\pi)^{\frac{3}{2}}\sigma_s^2\sigma_t} e^{-(\frac{x^2+y^2}{2\sigma_s^2}+\frac{t^2}{2\sigma_t^2})} \quad (1)$$

where σ_s and σ_t are the GD spatial and temporal kernel scales respectively. These user controlled parameters determine the edge detail level. If the ST volume is mostly homogeneous, then the scales should be high and vice-versa. The filtering is performed by rotating this kernel by different ST angles, namely θ and β. This gives a family of rotated kernels:

$$GD_{\sigma,\theta,\beta}(x, y, t) = GD_\sigma(x', y', t') \quad (2)$$

where (x', y, t') is the rotated coordinated system by (θ,β) ST angle. In computing the edge direction, we use the difference of offset Gaussian (DOOG) along the x-direction, defined as follows:

$$DOOG_\sigma(x, y, t) = G_\sigma(x, y, t) - G_\sigma(x + d, y, t) \quad (3)$$

where d is the offset between the center of the two Gaussian kernels, taken proportional to σ. The DOOG kernel is rotated at different θ, and β angles for different edge orientations.

2.1. Color edges

The intensity edge energy (E(s,θ,β)) at a specific orientation (θ, β) is computed by convolving the voxel intensities with the GD kernel:

$$E(s, \theta, \beta) = |I(x, y, t) * GD_{\sigma,\theta,\beta}(x, y, t)| \quad (4)$$

Defining $Error(s, \theta, \beta)$ as the prediction error between two pixels (x, y, t) and (x', y', t') where (x', y', t') is at a distance d from (x, y, t) in the direction defined by (θ,β). This error can be computed using the $DOOG$ kernel:

$$Error(s, \theta, \beta) = |I(x, y, t) * DOOG_{\sigma,\theta,\beta}(x, y, t)| \quad (5)$$

From which, we compute the probability of finding an edge at orientation (θ, β) as:

$$P(s, \theta, \beta) = \frac{Error(s, \theta, \beta)}{Error(s, \theta, \beta) + Error(s, \theta_\pi, \beta_\pi)} \quad (6)$$

where (θ_π, β_π) is a 3-D orientation in the opposite direction of the vector (θ, β) . All the above is applicable for color components also, i.e. the intensity I in the previous equations could be replaced by the R, G, B values.

2.2. Aggregate ST Edge Vectors

We proceed to combine R, G, B color edge information into the edge computation by assigning relative weights:

$$E(s, \theta, \beta) = \sum_f E_f(s, \theta, \beta).w(f) \quad (7)$$

where $f = \{R, G, B\}$ is the set of features used, with the relative weights w_f.

The ST Edge vector is defined as being the vector pointing at the gradient direction, with its magnitude denoting the strength of the gradient. The direction of the local ST edge vector is computed as:

$$A(s, \theta, \beta) = \sum_{(\theta',\beta')\epsilon N(\theta,\beta)} P(s, \theta', \beta')$$
$$(\theta_s, \beta_s)_{opt} = \arg\max_{\theta',\beta'} A(s, \theta', \beta') \quad (8)$$

Where $N(\theta, \beta)$ is the set of surrounding angles used in the averaging process. Hence the edge vector will be as follows:

$$\vec{F}_{ST}(x, y, t) = \sum_{N(\theta,\beta)} |E(s, \theta, \beta)| \, (\cos\theta \cos\beta \, \vec{x} + \sin\theta \, \vec{y} + \cos\theta \sin\beta \, \vec{t}\,) \quad (9)$$

3. COMBINING GEODESIC MORPHOLOGICAL TRANSFORMATIONS WITH THE ST EDGE VECTORS

After computing the ST edge vectors, we proceed to form volume regions from the gradient magnitude g in the volume V. Morphological watershed-based techniques have been used for image segmentation with promising results. There exist fast algorithms to implement the watershed simulation. For these reasons, we use the watershed method, in grouping the ST edge vectors. The watershed transform ($WTS_M(g)$) of a function g associates with every minimum of g a *catchment basin* (*influence zone*) and create *watersheds* between neighboring basins. However, one of the main drawbacks of watershed techniques is *over-segmenta-*

tion. A successful way to overcome over-segmentation is to using flooding from selected *seeds* or *markers*.

We begin by placing a set of initial markers on the first frame of the ST-volume. We start the watershed simulation by means of geodesic skeletons of influence (SKIZ) [1]. Geodesic transformations are very efficient in implementing morphological operations. Starting with the notion of geodesic distance, one may define geodesic dilations, erosions, and most morphological operations. Let $X \subset Z^2$ be a set, r and p two points of X. We define the geodesic distance $d_X(r,p)$ between r and p as the length of the shortest path (if any) contained in X and linking r and p. Let Y be any set included in X, composed of n connected components Y_i. The geodesic zone of influence $Z_X(Y_i)$ of Y_i is the set of points of X at a finite geodesic distance from Y_i and closer to Y_i than to any other Y_j.

$$Z_X(Y_i) = \left\{ r \in X : \begin{array}{c} d_X(r,p)\, finite \\ \forall j \neq i, d_X(r,Y_i) < d_X(r,Y_j) \end{array} \right\}$$

The boundaries between the various zones of influence give the geodesic skeleton of influence of Y in X, $SKIZ(Y;X)$. Based on the geodesic $SKIZ$ definition above, we implement an ST watershed simulation by processing successively the gray levels of the gradient volume. Define $B_i(g)$ as:

$$B_i(g) = \{r \in V : g(r) \leq i\}$$

We then compute the watershed transform $WTS_M(g)$ of a function g controlled by a marker set M by iterating as follows:

$$W_{i+1}(g) = SKIZ(W_i(g); B_{i+1}(g) \cup M) \qquad (10)$$

with the initial condition:

$$W_{-1}(g) = M$$

At the end of the iterations, when the highest gray level n is processed, we have the final watersheds:

$$WTS_M(g) = W_n(g)$$

4. EXPERIMENTAL RESULTS

We present preliminary experimental results of the proposed watershed-based segmentation method, and their results are also made available on the web [11]. The segmentation procedure starts by computing the ST edge vectors as previously discussed. The flooding starts in the ST volume from initial marker locations manually placed on the first frame. The flooding simulation is implemented via an order queue, leading to the final watersheds. In the experimental results, we use 64 ST rotation angles of the derivative of Gaussian (GD) kernels for an 124x92x20 ST volume.

In Fig. 1-9, we present the results of the proposed segmentation method, and compare it subjectively with the normalized cut (NCut) for video sequences. We use the same number of segmentation regions for both techniques. From these results, we clearly see that the results of NCut tend to miss some salient spatio-temporal regions, like the player's head, and short in the *tennis* sequence, and most of the left part of the face in the *lab* sequence. Our explanation for that effect is that in the NCut framework, there is no special care taken about ST edges, thus two neighboring regions may be grouped together if similar in color. However, in our proposed method, we initially compute ST edges, and then perform grouping so background-foreground distinction is better. It is worth noting that we only used color and location cues for the implementation of the NCut algorithm, and discarded the use of any *frame-by-frame* motion cues.

On the other hand, the initial marker locations affect the subsequent grouping in the whole ST volume in our method. Ideally, we should place small markers, one per region. As an example, in Fig. 4, two markers markers were placed, one on the background, and one on the face. If two markers are placed on the background, the background will be split into two regions. A similar problem also exists for the NCut method, since the implementation requires to sample randomly some voxels from the ST volume. The computations are carried on these samples to reduce complexity and the sample locations affect the final segmentation result.

5. DISCUSSIONS

In this paper we have presented a new technique for video sequence segmentation based on treating the video sequence as a volume. The method uses a spatio-temporal edge vector computed using video attributes to form the topological surface of a watershed grouping stage. Experimental results have shown that the proposed method is promising in extracting useful regions without the need of region frame-to-frame tracking. The work presented is still in its preliminary stages, and future work is needed for reducing: 1) the computational cost of the filtering part required for gradient calculations, 2) the sensitivity of results to initial marker locations.

6. REFERENCES

[1] E. Dougherty, *Mathematical Morphology in Image Processing*, 1993.

[2] F. Moscheni, S. Bhattacharjee, and M. Kunt, "Spatiotemporal segmentation based on region merging," *IEEE Trans. on Patern Analysis and Machine Intelligence*, vol. 20, no. 9, pp. 897–915, September 1998.

[3] E. Patras, E. Hendriks, and R. Lagendijk, "Video segmentation by map labeling of watershed segments," *IEEE Trans.*

on Patern Analysis and Machine Intelligence, vol. 23, no. 3, pp. 326–332, March 2001.

[4] H. Tao, H. Sawhney, and R. Kumar, "Object tracking with bayesian estimation of dynamic layer representations," *IEEE Trans. on Patern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 75–89, January 2002.

[5] S. Khan and M. Shah, "Object based segmentation of video using color, motion and spatial information," in *Proc. of CVPR*, 2001, vol. 2, pp. 746–751.

[6] J. Shi and J. Malik, "Motion segmentation and tracking using normalized cuts," in *Proc. of ICCV*, 1998, pp. 1154–1160.

[7] C. Fowlkes, S. Belongie, and J. Malik, "Efficient spatiotemporal grouping using the nystrom method," in *Proc. of CVPR*, 2001, vol. 1, pp. 231–238.

[8] I. Kompatsiaris and M. Strintzis, "Spatiotemporal segmentation and tracking of objects in image sequences," in *Proc. of ICIP*, 1999, vol. 2, pp. 155–158.

[9] K. Korimilli and S. Sarkar, "Motion segmentation based on perceptual organization of spatio-temporal volumes," in *Proc. of ICPR*, 2000, vol. 3, pp. 852–857.

[10] C. Lai, Y. Hung, and Y. Tsai, "A bayesian approach to video object segmentation via merging 3d watershed volumes," in *Proc. of ICPR*, 2002, pp. 496–499.

[11] `http://vision.ece.ucsb.edu/~msaban/watershed.html`.

(a) (b) (c)

Fig. 1. (a) -(c) Example frames from the *foreman* sequence.

(a) (b) (c)

Fig. 2. (a) -(c) Corresponding segmentation results for the *foreman* sequence.

(a) (b) (c)

Fig. 3. (a) -(c) Corresponding NCut segmentation results for the *foreman* sequence.

(a) (b) (c)

Fig. 4. (a) -(c) Example frames taken from the *Lab* sequence.

(a) (b) (c)

Fig. 5. (a) -(c) Corresponding segmentation results for the *Lab* sequence.

(a) (b) (c)

Fig. 6. (a) -(c) Corresponding NCut segmentation results for the *Lab* sequence.

(a) (b) (c)

Fig. 7. (a) -(c) Example frames of the *Table Tennis* sequence.

(a) (b) (c)

Fig. 8. (a) -(c) Corresponding segmentation results for the *Table Tennis* sequence.

(a) (b) (c)

Fig. 9. (a) -(c) Corresponding NCut segmentation results for the *Table Tennis* sequence.