AN UNSUPERVISED MULTI-RESOLUTION OBJECT EXTRACTION ALGORITHM USING VIDEO-CUBE

Fatih Murat Porikli

Mitsubishi Electric Research Labs
Murray Hill, NJ 07974

Yao Wang

Polytechnic University
Brooklyn, NY 11201

ABSTRACT

We propose a fast video object segmentation method that detects object boundaries accurately, and does not require any user assistance. Video streams are considered as 3D data, called video-cubes, to take advantage of 3D signal processing techniques. After a video sequence is filtered, marker nodes are selected from the color gradient. A volume around each marker is grown by using color/texture distance criteria. Then volumes that have similar characteristics are merged. Self-descriptors for each volume, mutual-descriptors for each pair of volumes are computed. These descriptors capture motion and spatial information of volumes. In the clustering stage, volumes are classified into objects in a fine-to-coarse hierarchy. While applying and relaxing descriptor based adaptive, similarity scores are estimated for each possible pair-wise combination of volumes. The pair that gives the maximum score is clustered iteratively. Finally, an object-based multi-resolution representation tree is assembled.

1. INTRODUCTION

Unsupervised object segmentation techniques can be categorized into three classes: region based methods using a homogeneous color criterion [1], object based approaches utilizing homogeneous motion criteria [2], and object tracking [4]. Although color oriented techniques work well in some situations where the input data set is relatively simple, clean, and fits the model well, they lack generality and robustness. The main problem arises from the fact that a video object can contain totally different colors. On the other hand, works in the motion oriented segmentation domain start with an assumption that a semantic video object has homogeneous motion. These motion segmentation works can be simply separated into two broad classes: boundary placement schemes and region extraction schemes [3]. Most of the motion-based methods are based on rough optical flow estimation or unreliable spatiotemporal segmentation. Hence, they suffer from the inaccuracy of motion boundaries. The last class of methods is object-tracking. However, the tracking algorithms need user interface, and the performance of the tracking algorithms highly depends on the initial segmentation. Most object extraction algorithms treat object segmentation as an inter-or-intra frame processing problem with some additional parametric motion model assumptions or smoothing constraints, and disregard 3D aspect of the video data. To develop an algorithm that blends motion and color, we treat video sequences as 3D volumetric data, generate 3D color consistent initial segments, and merge the initial segments by using their trajectory information. This enables us to propagate segmentation information both forward and backward in time. Because no separate motion estimation is involved, segmentation is fast. Moreover, it does not depend on either the initial region segmentation, homogeneous motion constraints, or inaccuracy of motion boundaries as it happens in the motion-oriented methods. Object boundaries are precisely found. Unlike statistical approaches, the number of objects does not have to be specified before segmentation, no user-interface is needed.

In the next section, the video-cube concept is introduced. Section III describes the stages of filtering, marker selection, volume growing, merging, and clustering volumes into objects. The experimental results and discussion are included in the last section.

2. VIDEO-CUBE

Arrangement of image frames along the time axis as shown in Fig. 1 converts the input video sequence into 3D volume of data. A video-cube $V(p)$ is formed by assigning a feature vector $\omega(p)$ that consists color values ω_Y, ω_V, ω_U and also processed scores, i.e. edges, texture, frame difference, etc., to each element $p(x,y,t)$ of 3D data V by indexing video between two scene cuts. To accelerate segmentation, we used the color components. In case of moderate object motion, an object has continuous color silhouettes in time axis. Thus, a 3D volume growing method by using color features can identify the smallest color consistent parts of the video sequence. These parts, called as volumes, are then grouped using trajectory information to determine the objects. For

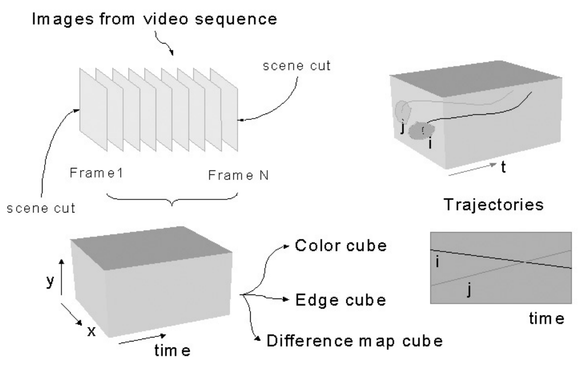

Fig. 1. Video-Cube indexing.

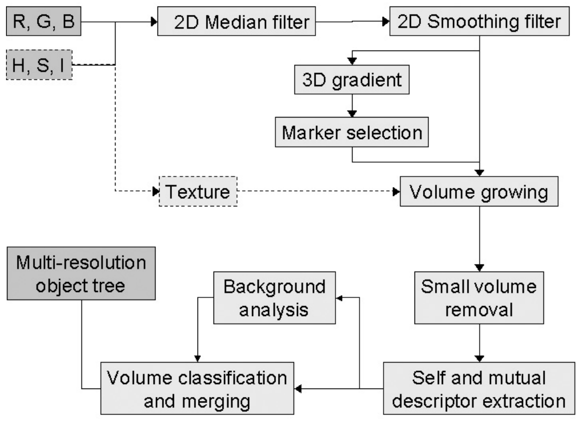

Fig. 2. Video-Cube indexing.

a streaming video, the video-cube can be generated for a certain number of frames permitting overlaps to ensure the object number consistency within.

3. OBJECT SEGMENTATION

The flow diagram of the algorithm is given in Fig. 2. We used YUV color features because it performs in accordance with human reception and more importantly, inter-color distances can be computed using the L^2-norm. The input sequence is first 3×3 median filtered to remove out intensity singularities without disturbing the edge formation.

3.1. Marker selection

After the video-cube is filtered, initial volumes are obtained by enlarging portions of video-cube such that the color distribution is uniform within. Such portions are expanded from seed points, called markers. Let S be the set of all possible spatiotemporal points, i.e., all the points of V. The gradient magnitude is computed from the color channels, and the minimum gradient magnitude point is chosen as a marker m_i. A volume W_i is grown as explained in the next section, and all the points of the volume W_i is removed from the set S

$$m_i = \arg_p \min_S \nabla V(p) \quad ; \quad S = V - \bigcup_{j=1}^{i} W_j . \qquad (1)$$

The next minimum in the remaining set is chosen, and selection process repeated until no more point remains.

3.2. Volume growing

The volumes W_i, $i = 1, .., M$ are enlarged iteratively from the markers m_i by using color/texture similarity of the feature vectors $\omega(m_i)$'s. Two distance criteria d_g,d_l are designed. The first criterion d_g measures the distance between the feature vector of the current volume and the candidate point. The second criterion d_l determines the distance between the feature vectors of the current volume and another point that is already included in the current volume and also adjoint to the candidate. Two thresholds ϵ_g, ϵ_l are set with respect to the variance and dynamic range of the color features as

$$\epsilon_g = \frac{\mu_Y + \mu_V + \mu_U}{\sigma_Y + \sigma_V + \sigma_U + 1} \qquad (2)$$

where μ is the dynamic range, η is the mean, and σ is the standard deviation

$$\mu_k = \max \omega_k - \min \omega_k \qquad k : Y, U, V \qquad (3)$$

$$\sigma_k^2 = \frac{1}{N} \sum_p (\omega_k(p) - \eta_k)^2 \qquad (4)$$

The local threshold ϵ_l is the average of the discontinuity between the neighboring points' features scaled with the points relative edgeness,

$$\begin{aligned}
\epsilon_l(x, y, t) &= \frac{\tilde{\eta}_V + \tilde{\eta}_U + \tilde{\eta}_V}{3} \sum_k \tilde{\mu}_k(p) \\
\tilde{\eta}_k &= \frac{1}{K} \sum_p |\omega_k(p) - \omega_k(q)| \\
\tilde{\mu}_k(p) &= \max \omega_k(p) - \min \omega_k(p)
\end{aligned}$$

where q_i, $i = 1, .., K$ are neighboring points of p. Let x^- be an unmarked candidate point that is adjoint to the current volume. Let x^+ be a point adjoint to x^- but already included in the current volume W_i. Then, the first global distance d_g is calculated from

$$d_g(\omega(m_i), \omega(x^-)) = \sum_k |\omega_k(m_i) - \omega_k(x^-)| \qquad (5)$$

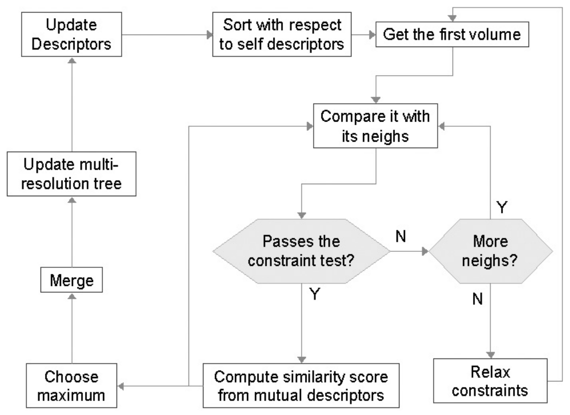
Update Descriptors
Sort with respect to self descriptors
Get the first volume
Compare it with its neighs
Update multi-resolution tree
Passes the constraint test?
N
Y
More neighs?
Y
N
Merge
Choose maximum
Compute similarity score from mutual descriptors
Relax constraints

Fig. 3. Clustering loop.

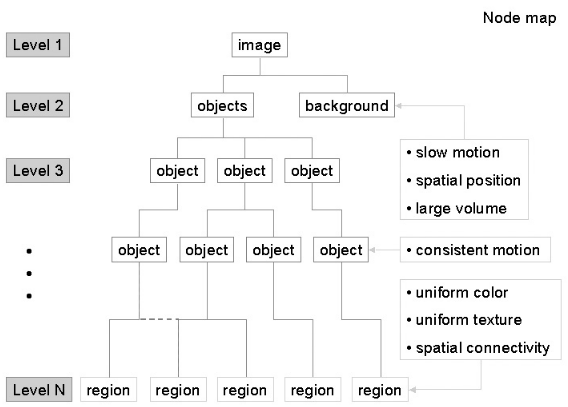

Fig. 4. Multi-resolution object tree

Similarly, the second local distance d_n is

$$d_l(\omega(x^+), \omega(x^-)) \quad = \quad \sum_k |\omega_k(x^+) - \omega_k(x^-)|. \quad (6)$$

If the distances d_g and d_l are smaller then ϵ_g and ϵ_l, the point x^- is included in the volume W_i. The neighboring point x^- is set as an active surface point for W_i, and the feature vector for the marker is updated accordingly. In the next iteration, the neighboring pixels of the active surface points are examined. Volume growing is repeated until no point remains in the video-cube.

Some of the volumes are negligible in size, however, they effect the computational load if the clustering stage. Small volumes are blended into the bordering most similar volumes that gives the best combination of the greatest mutual surface, the smallest color distance, the smallest mutual volume, and the highest compactness ratio as defined in the next section.

3.3. Descriptors

For each volume W_i, a trajectory $T_i(t) = [X_i(t), Y_i(t)]^T$ is extracted by computing the frame-wise averages of volume's points coordinates

$$T_i(t) = \begin{bmatrix} X_i(t) \\ Y_i(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{N_t}\sum x_t \\ \frac{1}{N_t}\sum y_t \end{bmatrix} \quad (x_t, y_t, t) \in W_i$$

Trajectories are the center of masses of regions in an image frame, hence they approximate the translational motion of the region. The Cartesian distance $\Delta d_{ij}(t)$ between the trajectories $T_i(t)$ and $T_j(t)$ at time t is calculated as

$$\Delta d_{ij}(t) = \sqrt{(X_i(t) - X_j(t))^2 + (Y_i(t) - Y_j(t))^2} \quad (7)$$

Some of the descriptors that interpret the mutual relations of the grown volumes W_i and W_j are defined as

$$\begin{aligned}
color &: \mu_k = \frac{1}{N_i}\sum \omega_k(p) \;\; k: Y, U, V \\
volume &: v = \cup p \\
surface &: s = \sum p_i \cap p_j, \;\; i \neq j \\
compactness &: c = v/s^2 \\
existence &: e = \sum_t (T_i \wedge T_j) \\
dist.mean &: \mu_d = \frac{1}{e(ij)} \sum_t \Delta d_{ij}(t) \\
dist.variance &: \sigma_d^2 = \frac{1}{e(ij)} \sum_t (\Delta d_{ij}(t) - \mu_d(ij))^2 \\
color\ dist. &: \Delta\mu = \sum_k |\mu_k(i) - \mu_k(j)| \\
direction\ dist. &: \Delta\phi = \sum_t |\Delta\phi_i(t) - \Delta\phi_j(t)| \\
compact.\ ratio &: r_c = c(W_i \cup W_j)/(c(i) + c(j)) \\
boundary\ ratio &: r_b = b(W_i \cap W_j)/b(i)
\end{aligned}$$

where $p_i \in W_i$, $p_j \in W_j$, and $\Delta\phi_j(t) = \angle(T_i(t) - T(t^-))$. Each descriptor is normalized to $[0, 1]$. These descriptors identify volumes motion ($\sigma_d, \mu_d, \Delta\phi$), shape ($v, s, c, r_c, r_b$), and color ($\mu_k, \Delta\mu$) characteristics. Motion characteristics such as vertical and horizontal motion, route length, mean and variance of distance, direction difference, and average change in the distance are derived from the trajectories.

3.4. Clustering

In a fine-to-coarse clustering hierarchy, the most similar volume pairs are merged to decrease the number of the volumes at each iteration. The volumes such that their motion trajectories are consistent and their combination builds a relatively compact shape are clustered. Color aspects are omitted; partly because it was already included in volume growing, and also portions of a semantically meaningful object do not have to possess the same color aspects, i.e., a human face made up from different color regions, mouth, skin, hair, etc. Volumes are sorted with respect to their sizes, compactness and existence values. Starting from the first volume in the sorted list (likely the least compact, smallest, and least visible volume), each volume is compared sequentially to its neighbors. If their mutual descriptors satisfy a set of thresholds $\lambda(\sigma_d)$, $\lambda(\mu_d)$, $\lambda(\Delta\phi)$, $\lambda(r_c)$, $\lambda(r_b)$, a similarity score S_{ij} for that pair W_i,W_j is computed. Initial values of λ's are assigned as the mean of the corresponding descriptor before the clustering stage. The constraint set is used to prevent from degenerate cases that still give high similarity scores. The most suitable descriptors used to form the similarity score are the variance of trajectory distance for motion, and the compactness ratio and mutual boundary ratio for shape relevance. The similarity score is a function of

$$S_{ij} = f(\sigma_d^{-1}, r_c, r_b) \qquad (8)$$

After similarity scores are computed for the possible volume pairs, the volume pair that gives the maximum similarity score are merged together. Then, the descriptors are updated and normalized. If no volumes are merged in the iteration, the thresholds are relaxed using the medians of the previous n^{th} object level descriptors

$$\lambda^{n+1}(g) = median(g^n) \qquad g : \sigma_d,\ \mu_d,\ \Delta\phi,\ r_c,\ r_b. \qquad (9)$$

Clustering runs until the gradient of the best similarity score become large compared to the initial gradients. An object-wise multi-resolution representation is generated Fig. 4.

4. TEST RESULTS AND CONCLUSION

Fig. 5 presents the sample results. The first row shows the original frames. The second row images are the volume growing results. The following rows are the levels from the clustering algorithm. In the figures, the volumes are color coded for illustration purposes. The results confirmed that the object extraction is robust even when the motion is large. The backgrounds and object boundaries were detected accurately. Because no separate motion computation is involved in segmentation, our algorithm is fast unlike the dense optical flow computing methods. An object-wise multi-resolution representation is obtained; therefore the extraction is not repeated when the number of segmented objects is changed as in the estimation-based or tracking-based methods.

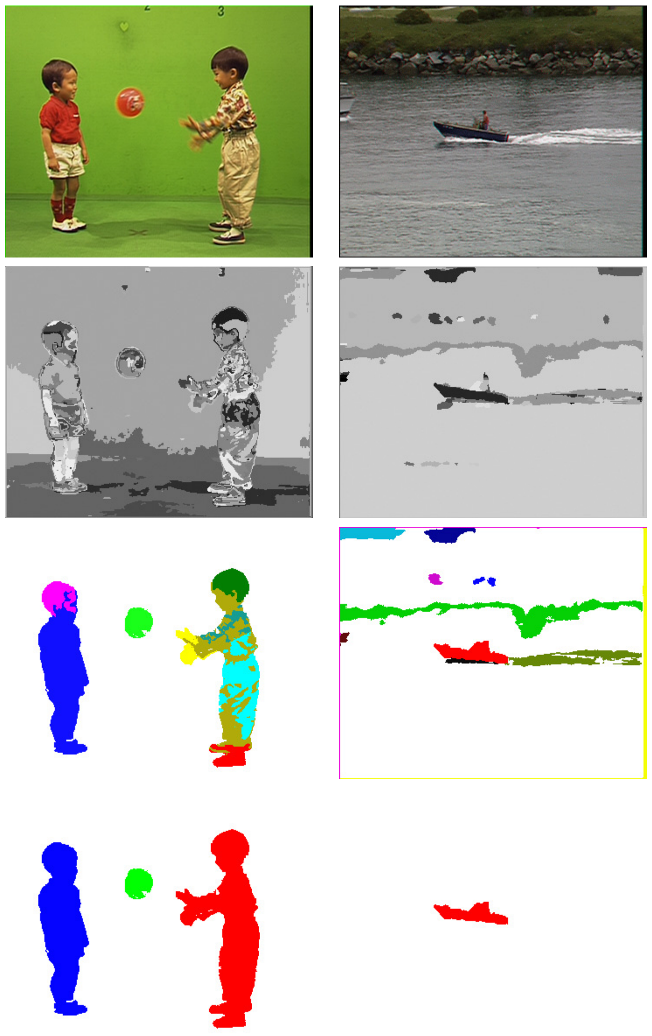

Fig. 5. Sample test results

5. REFERENCES

[1] M. Kunt, A. Ikonomopoulos, and M. Kocher, "Second generation image coding", Proceedings of IEEE, no.73, 549–574, 1985

[2] B. Duc, P. Schtoeter, and J. Bigun, "Spatio-temporal robust motion estimation and segmentation", Proc. Comput. Anall. Images and Patterns, 238–245, 1995

[3] J. Wang and E. Adelson, "Representing moving images with layers", IEEE Trans. Image Process., no.3, 1994

[4] J. K. Aggarwal, L. S. Davis, and W. N. Martin, "Corresponding processes in dynamic scene analysis", Proceedings of IEEE, no.69, 562–572, 1981