VIDEO OBJECT SEGMENTATION BY VOLUME GROWING USING FEATURE-BASED MOTION ESTIMATOR

Fatih Murat Porikli
Mitsubishi Electric Research Labs,
Murray Hill, NJ 07974, USA,
fatih@merl.com

Abstract

We present a video segmentation algorithm that blends color based region segmentation into feature-based motion estimation in an iterative clustering framework. The algorithm does not require any user assistance to accurately extract object boundaries from color video. After filtering and simplification, marker points are selected to grow 3D volumes around them by evaluating local color attributes. The volumes are refined and motion trajectories are obtained. Relational properties of volume pairs are captured in the form of mutual descriptors that are computed from motion trajectories. These descriptors designed to characterize shape as well as spatial properties of volumes. The trajectories are then used to initialize feature-based motion estimation. In the clustering stage, volumes are merged into objects iteratively until a motion similarity score of merged objects becomes negligible. Finally, a multi-resolution object tree that gives the video object planes for every possible number of objects is generated.

1 Introduction

One of the most significant achievements of the digital video technology is its unstoppable intrusion into our daily life. More and more visual information is available, hence more effort should be invested to store, transmit retrieve, and understand it. If video can be stored in the form of individual objects, retrieval of multimedia information is as simple as that of textual information. Having an object-based representation scheme that identifies the important parts of image frames, video sequences can be encoded efficiently to satisfy transmission requirements.

The purpose of video segmentation is to enable an object-based description of the scene by extracting objects of interest from a series of consecutive video frames. Video object segmentation is an initial step for many vision and multimedia applications. With a good segmentation, it is possible to access and manipulate objects in the video. It allows high-level image analysis such as object recognition and scene interpretation. Efficient object segmentation is a prerequisite for object-based video coding standard MPEG-4, and content-based multimedia database retrieval standard MPEG-7. Scene interpretation in a video sequence is key to video understanding, and is required in many applications, such as information indexing and retrieval in multimedia database, spotting and tracking of special events in surveillance video, video editing, and movie categorization, etc. Such a facility would allow recovery of desired video segments or objects from a very large database of video sequences. The efficient use of stock film archives and identification of specific activities in surveillance videos are usually cited as potential applications.

Methodically, object segmentation techniques can be grouped into three classes: region-based methods using a homogeneous color criterion [1], object-based approaches utilizing a homogeneous

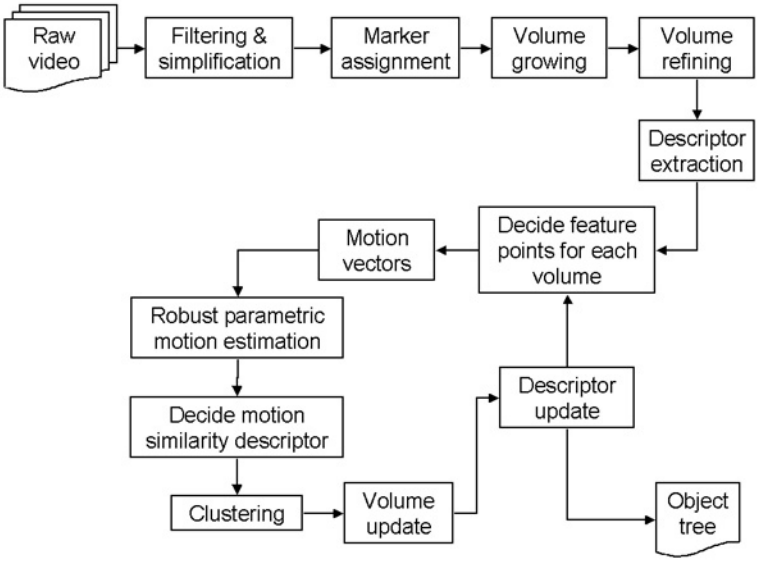

Figure 1: Flow diagram of the video segmentation algorithm.

motion criterion [2], and object tracking [3]. 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 [4]. These motion segmentation works can be simply separated into two broad classes: boundary placement schemes and region extraction schemes [5]. Most of them are based on rough optical flow estimation or unreliable spatiotemporal segmentation. As a result, they suffer from the inaccuracy of motion boundaries. The last class of methods that is related to semantic video object extraction is tracking [6]. However, the tracking algorithms need user interface, and the performance of the tracking algorithms depends extensively on the initial segmentation. In general, most of the object extraction algorithms treat segmentation as a 2-D inter-or-intra frame processing problem with some additional motion model assumptions or smoothing constraints by ignoring the 3-D nature of the video data.

We consider video sequence as a 3D volumetric data instead of separate 2D frames to develop an algorithm that blends intra-frame color segmentation schemes with inter-frame motion estimation techniques. Having 3D representation, the extracted object information can be propagated forward and as well as backward in time without saddling into initial segmentation inaccuracies or tracking limitations.

A general framework of the algorithm is shown in Fig. 1. In Section II, the volume growing concept is introduced. Section III describes the stages of filtering, marker selection, volume growing, and clustering volumes into objects. The test results and discussion are included in the conclusions.

2 Formation of Spatiotemporal Data

Instead of treating video frame by frame basis, we assembled video shot into a 3D data. By registering the color channels and frame differences along the time axis as shown in Fig. 2, a spatiotemporal structure $V(x,y,t)$ $1 \leq x \leq x_M$, $1 \leq y \leq y_M$, and $1 \leq t \leq t_M$ is constructed from t_M frames sized $x_M \times y_M$ pixels. Each element of V corresponds to a vector $\mathbf{v}(x,y,t) = [y,u,v,d]$ that consists of color and frame difference features of the spatiotemporal point (x,y,t). Here, y, u, and v stand for

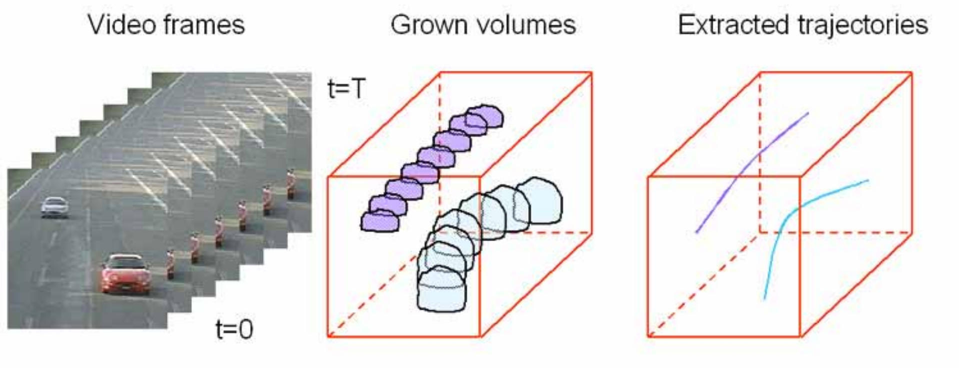

Figure 2: Constructing spatiotemporal data structure V.

the luminance and chrominance features, respectively. For simplicity, we will denote each component as a subscript, e.g., $\mathbf{v}_y$ instead of $\mathbf{v}(x, y, t)_y$.

We favored the YUV color space to the RGB space. Most of the existing color segmentation approaches have utilized the RGB color space although the RGB space has machine oriented chromatics rather than human oriented chromatics. One other disadvantage of the RGB space is the dependency of all three parameters from the light intensity. The YUV space performs in accordance with human reception, chrominance values are lighting more lighting independent, and inter-color distances can be computed using the L^2-norm as the RGB.

3 Object Segmentation

3.1 Filtering

Color channels of the input video $\mathbf{v}_y, \mathbf{v}_u, \mathbf{v}_v$ are first filtered to remove out noise. Elimination of the image flickers prevents from excessive segmentation, and decreases computation load significantly. A fast 3×3 median filter implementation [7] that exploits 2D coherence is utilized. Median filter sorts a list with respect to the magnitudes of the list elements, then finds the magnitude of the element at the middle of the list, e.g., magnitude of the third element in a five elements list. We can exploit the 2D nature of median image filter by dividing the process into several 1D sorting problem to take the advantage of working in smaller lists. Note that sorting algorithms are logarithmically complex. Two horizontal adjacent medians are computed in one step as illustrated n Fig. 3. Let a, b, c, d be 3×1 columns of a 3×4 image window. Let b_{3x3}, c_{3x3} represent the 3×3 windows around the center points of b and c . First, slices c and d are sorted, slices a and b were already sorted in previous step (6 comparison). Slice b and c are merged to thick slice bc (5 comparison). Slice a and bc are merged to compute median for b_{3x3} (4 comparison). Slice d and bc are merged to compute median for c_{3x3} (4 comparison). Therefore, the number of necessary comparisons is reduced to from 30 to 9.5. The comparisons are implemented as nested if conditions.

The Gaussian filter is basically a low-pass filter that is used to blur image, remove detail and noise. Its has bell-shaped kernels in both frequency and spatial domains

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}. \qquad (1)$$

The Gaussian outputs a weighted average of each pixel's neighborhood, with the average weighted more towards the value of the central pixels. Thus, provides gentler smoothing and preserves edges better than a mean filter. However, the Gaussian filter assumes that noise is normal distributed. For

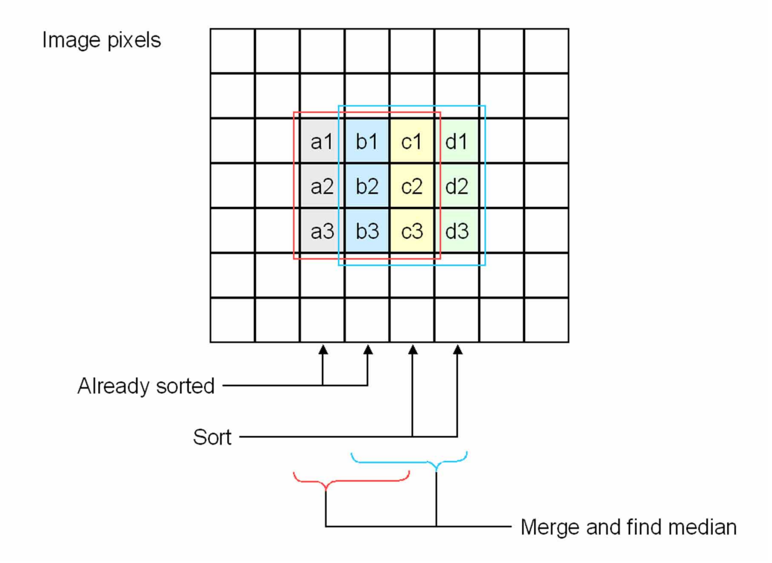

Figure 3: Fast 3 × 3 median filtering that exploits 2D coherence.

salt-and-pepper noise, it reduces the intensity of the noise, but disturbs high frequency detail and smears noise over a larger spatial region.

3.2 Marker Selection

An object is defined as a collection of image regions which have been grouped together under some criteria across several frames. A region is defined to be a contiguous set of pixels that is homogeneous in the features i.e., texture, color, motion, and shape. A video object is assumed to be made of smaller parts. Such parts are the group of points that are spatially consistent, i.e., color and texture distributions are uniform within. The grouped points, called as volumes, are expanded from seed points, called markers. For each marker, a volume is assigned and volume's attributes are initialized with the marker points attributes $\mathbf{v}(x, y, t)$. A marker point should be selected such that it can characterize its enclosing volume as relevant as possible. The points have small color gradient magnitude are good candidates to represent their local neighborhood. Let S be the set of all possible spatiotemporal points, i.e., it is all the points of V initially. 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 following section, and all the points of the volume is removed from the set S

$$m_i = \arg\min_S \nabla V(x, y, t) \quad ; \quad S = V - \bigcup_{j=1}^{i} W_j . \tag{2}$$

The next minimum in the remaining set is chosen, and selection process repeated until no point remains in the V. Finding minimum is a computationally expensive process. Rather than searching the full-resolution V, a down-converted version is used. More computational reduction is achieved by dividing down-converted V into slices. Minimum is found for the first slice, and a volume is grown, then the next minimum is searched in the next slice.

3.3 Volume Growing

Volumes are enlarged by diffusion from markers by evaluating distance metrics. For each volume W_i, a feature vector ω^i that equals to the marker m_i's feature vector is initiated. Two distance criteria

d_g,d_l are implemented. The first criterion d_g measures the distance between the feature vector of the current volume and the unmarked candidate point. This criterion serves as "global" measure. The second criterion d_l determines the distance between the feature vectors of the unmarked candidate and another marked point that is already included in the current volume and also adjoint to the candidate. Thus, the second criterion can be viewed as a "local" measure. A global threshold ϵ_g helps limiting the range of feature distance, i.e., color variation in the volume. Local threshold ϵ_l prevents overstepping edges even the global threshold permits. The global and local thresholds are determined separately for each input video. 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 defined as

$$d_g(\omega^i, \mathbf{v}^-) \quad = \quad \sum_k |\omega^i_k - \mathbf{v}^-_k| \qquad k : y, u, v \tag{3}$$

where $\mathbf{v}^-$ and $\mathbf{v}^+$ are the feature vectors of x^- and x^+. Similarly, the second local distance d_l is

$$d_l(\omega^i, \mathbf{v}^-) \quad = \quad \sum_k |\mathbf{v}^+_k - \mathbf{v}^-_k| \qquad k : y, u, v. \tag{4}$$

The magnitude operator, $|\cdot|$, is chosen as the euclidian distance of three color elements for the YUV color space. 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 V. The thresholds are made adaptable to input video by using the variance and dynamic range of the features that are the color values of the points. Variance of a feature gives information about the distribution of that feature in the spatiotemporal image. A small feature variance indicates smooth feature distribution, and high variance shows texture and noise. The global threshold is assigned high if the variance is high and it is scaled with respect to the dynamic range of the feature. Let I represent a feature, i.e, I is the luminance value. Then, the global variance σ^2 and mean η are defined

$$\sigma^2 = \frac{1}{M} \sum_{x,y,t \in V} (I(x,y,t) - \eta)^2, \qquad \eta = \frac{1}{M} \sum_{x,y,t \in V} I(x,y,t). \tag{5}$$

where M is the total number of points in the V. The dynamic range μ for is

$$\mu \quad = \quad c_2 - c_1 \tag{6}$$

$$\int_0^{c_1} h_I(z)\, dz \quad = \quad \int_{c_2}^{1} h_I(z)\, dz = 0.05 \tag{7}$$

where h_I is the normalized histogram function of the feature I. Then the global threshold ϵ_g is then assigned by scaling the dynamic range as

$$\epsilon_g = \kappa \frac{\mu}{\sigma + 1} \tag{8}$$

where $\kappa > 1$ is the sensitivity parameter that determines how fine the final segmentation should be. It is observed that a good choice is $\kappa \approx 2$. The local threshold $\epsilon_l(x,y,t)$ is the average of the discontinuity between the neighboring point features scaled with a relative edgeness score:

$$\epsilon_l(x,y,t) \quad = \quad \tilde{\eta} \cdot \tilde{\mu}(x,y,t) \tag{9}$$

$$\tilde{\eta} \quad = \quad \frac{1}{2M} \sum_{x,y,t \in V} |I(x,y,t) - I(x-1,y,t)| + |I(x,y,t) - I(x,y-1,t)| \tag{10}$$

$$\tilde{\mu}(x,y,t) \quad = \quad \max I(i,j,t) - \min I(i,j,t) \qquad x-k, y-k \leq i,j \leq x+k, y+k \tag{11}$$

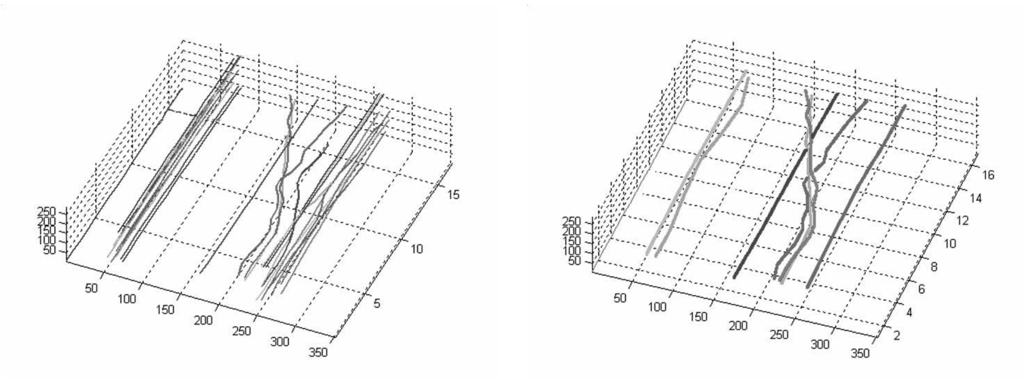

Figure 4: Sample trajectories; (a) before clustering, (b) at a lower object level

in a $2k+1 \times 2k+1$ window. To accelerate extraction of these statistics, only the first image of the video can be used instead of the whole spatiotemporal data V.

For a marker point$m_i(x,y,t)$, the growing is accomplished by evaluating the adjoint pixels distances $d_g(\omega^i, \mathbf{v}^-)$ and $d_l(\omega^i, \mathbf{v}^-)$ in a 3D 6^{th}-neighborhood. The adjoint points are $\mathbf{v}^- = (x+1,y,t),(x-1,y,t),(x,y+1,t),(x,y-1,t),(x,y,t+1),(x,y,t-1)$ where the current point is (x,y,t). If the distances are smaller than their corresponding thresholds, the new point is marked and included in the volume W_i. The feature vector w_i of the volume is updated by the new average values, the current point is changed accordingly, and growing is carried on until no more adjoining point satisfy the distance criteria. A new marker is selected, and growing is repeated.

After volume growing, the V is divided into multiple smaller parts. Some of these parts are negligible in size, however, they effect the computational load of the clustering stage due to clustering is $O(N^2)$. Also, some points, such as edges, may not be grouped into any of the volumes after growing. To assign ungrouped points into a volume, the volume growing thresholds are relaxed iteratively, and at each iteration, volumes are inflated towards the unmarked points until no more point remains. 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 that is presented next.

3.4 Descriptor Extraction

3D volume growing assembles color consistent points into homogeneous images regions at each frame. However, most of the video objects, i.e. human head, cars, contain several color regions. The next step of the object detection should be to extract and combine the motion information to built up motion consistent objects.

Descriptors are used to capture various aspects of the volumes. The volumes W_i are represented by a set of self descriptors $f(i)$ and mutual descriptors $g(i,j)$ as summarized in Table 1. These descriptors identify motion, spatial, and color characteristics, as well as the mutual correlation. The volumes will be grouped with respect to their descriptors at the clustering stage in order to assemble the objects. 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

$$T_i(t) = \begin{bmatrix} X_i(t) \\ Y_i(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{N_t}\sum x_t \\ \frac{1}{N_t}\sum y_t \end{bmatrix} \quad ; \quad (x_t, y_t, t) \in W_i. \tag{12}$$

Trajectories are the center of masses of regions in an image frame, hence they approximate the translational motion of the region. They are also used to initialize motion parameters in the model fitting

color mean	$f_1(i)$	$\frac{1}{N_i}\sum \mathbf{v}_y(x_k)\ ,\ x_k \in W_i$
volume	$f_2(i)$	$\bigcup x_k\ ,\ x_k \in W_i$
surface	$f_3(i)$	$\sum x_k \cap x_l\ ,\ x_k \in W_i\ x_l \in W_j\ i \neq j$
compactness	$f_4(i)$	$f_2(i)/(f_3(i))^2$
existence	$f_5(i)$	$\sum i_t\ ;\ i_t = 1 \leftarrow T_i(t) \neq 0$
motion	$\mathbf{f_6(i)}$	motion parameters
mean of distance	$g_1(i,j)$	$\frac{1}{N_i \cap N_j}\sum \Delta d_{ij}(t)$
variance of distance	$g_2(i,j)$	$\frac{1}{N_i \cap N_j}\sum(\Delta d_{ij}(t) - g_1(i,j))^2$
directional difference	$g_3(i,j)$	$\sum \lvert T_i(t) - T_i(t-1) - T_j(t) + T_j(t-1)\rvert$
compactness ratio	$g_4(i,j)$	$\frac{f_4(W_i \cup W_j)}{f_4(W_i)+f_4(W_j)}$
mutual boundary ratio	$g_5(i,j)$	$f_3(W_i \cap W_j)/f_3(i)$
color difference	$g_6(i,j)$	$\lvert f_1(i) - f_1(j)\rvert$
motion similarity	$g_7(i,j)$	$\lvert\lvert \mathbf{f_6(i)} - \mathbf{f_6(j)}\rvert\rvert$

Table 1: Self and Mutual descriptors

stage. Sample trajectories are shown in Fig. 4. The difference $\Delta d_{ij}(t)$ between the trajectories $T_i(t)$ and $T_j(t)$ at time t is calculated by

$$\Delta d_{ij}(t) = \sqrt{(X_i(t) - X_j(t))^2 + (Y_i(t) - Y_j(t))^2}. \tag{13}$$

This difference is utilized to obtain the set of mutual descriptors $g_1, g_{,}g_3$. The variance of the differences g_2 indicates the degree of agreement of the motions of two volumes. The mean of distance g_1 usually shows how far away a volume from the other. Generally, a small variance of distance represents two volumes have almost same translational motion i.e. have constant in frame distances along time, and a big value mostly purpose different objects. The directional difference g_3 distinguishes volumes that have constant distances but opposite directions, e.g., two volumes turning around a center point.

A second set of mutual descriptors g_4, g_5 is shape related. We define compactness f_4 for a W_i as the ratio of its volume to its squared surface size. For a spherical volume, the compactness descriptor has large value, however, more a volume becomes elongated less the compactness descriptor becomes. The compactness ratio g_4 of a pair of volumes W_i and W_j is then the amount of the change on the total compactness before and after these two volumes are merged. A small g_4 means the merging generates a less compact object which is not favorable. By clustering, we also aim to obtain more compact objects in a way. The second shape related descriptor g_5 corresponds the ratio of mutual surface of W_i and W_j to the surface of W_i. In clustering, a volume pair with high g_5 is preferred. The small values of mutual surface ratio usually refers a less relevant object except the background object. The color difference descriptor and distance descriptor g_6, g_1 is defined to refine the clustering results.

Motion dissimilarity can be characterized by the motion model parameters. After motion parameters computed for each volume at each frame, a mutual dissimilarity descriptor g_7 is utilized to evaluate frame-wise similarity of these motion models. Unlike previously mentioned motion descriptors, the dissimilarity descriptor uses an affine motion model instead of translational

$$\mathbf{f}_7 = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_5 \\ a_6 \end{bmatrix} \tag{14}$$

If the affine motion models of volumes W_i and W_j are $\mathbf{f}_7(i)$ and $\mathbf{f}_7(j)$ respectively, the dissimilarity descriptor is found by

$$g_7 = < A_i - A_j, A_i - A_j > \tag{15}$$

where A is the vector of motion parameters

$$A = [a_1 \; a_2 \; a_3 \; a_4 \; a_5 \; a_6]. \tag{16}$$

3.5 Feature-based Motion Estimation

Using 3D volume growing solves the most crucial egg-or-chicken problem of motion based segmentation; should the region of support be obtained first by color segmentation then motion field is estimated, or first motion field is obtained then region of support is determined? After volume growing, we have all the region of supports, i.e. projections of volumes on each frame. Therefore, a motion estimation can be successfully applied using these regions for each volume. To extract motion parameters, motion vectors are computed. An optical flow method would be restricted to the small motions, intensity sensitive, moreover it is computationally expensive. A dense block-matching scheme is an expensive process too. Thus, we preferred to use feature-points based motion estimation. In comparison, using selected feature points has advantage of preventing from the aperture problem rather than using all points. Only a small number of such points are enough to compute motion parameters.

First, feature points that motion vectors will be obtained are determined for each volume. A feature point p has higher spatial energy than the other points in the region. Spatial energy S^t at frame t is a variant of local texture

$$S^t(x, y) = \max I(x, y) - \min I(x, y), \qquad (x, y) \in W_i^t \tag{17}$$

A certain number of such p points are determined for each region at each frame. Then corresponding motion vectors are estimated by the cross block-matching method. Cross block-matching is a suboptimal application of exhaustive search. At the beginning, the displaced block difference error of the four motion vectors for the center point (x, y) is computed. In case of $-7, 7$ pixels motion range, the four motion vectors are $(x - 4, y - 4), (x + 4, y - 4), (x + 4, y + 4), (x - 4, y + 4)$ that are located on a diagonal cross. The minimum error point is chosen and the next four points around it is searched. Lets assume the minimum at the first stage is the vector (x', y'). Then the new four points are $(x' + 2, y'), (x' - 2, y'), (x', y' + 2), (x', y' - 2)$ which are now on the turned and halved cross. The matching is done until one pixel resolution. The above limits $-8, 8$ requires only $4 + 4 + 4 = 12$ matching to find the suboptimal solution instead of $15 \times 15 = 225$ matching to find the best solution. Although a few of the motion vectors may not accurate, the speed is multiplied almost 20 times.

Then, the affine motion parameters $a_1, ..., a_6$ are fitted by minimizing least-squares. The least-squares fitting is a maximum likelihood estimation of the fitted parameters if the measurement errors are independent and normally distributed with constant standard deviation.

3.6 Autoregressive Clustering Volumes into Objects

A fine-to-coarse hierarchical clustering is applied to merge the volumes W_i's obtained after volume growing stage. The most similar volume pairs are put together to decrease the number of the volumes at each iteration. Two volumes are similar if their motion descriptors are consistent and they built a compact shape when combined together. Since an object do not have to be built up from the same color regions, the color descriptors are omitted.

A likeness score L_{ij} is calculated for each possible adjoint volume pair W_i, W_j from the descriptors. The volume pairs with small g_1 and g_6 descriptors are privileged by giving them a rank after ordering g_1 and g_6 descriptors and combining them into a mutual rank r. The pairs having very low ranks are neglected. For the remaining the likeness score is computed. We made the following assumptions on the likeness score:

- inversely proportional to the distance variance g_2,

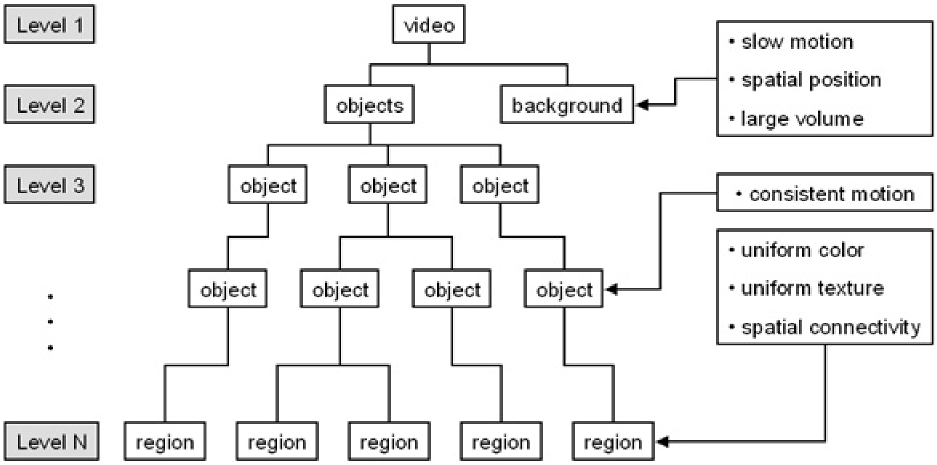

Figure 5: Multi-resolution object tree.

- inversely proportional to the direction difference g_3,
- proportional to the compactness ratio g_4,
- proportional to the mutual surface ratio g_5,
- not affected by the color difference g_6
- inversely proportional to the motion dissimilarity g_7

To combine the above assumptions into a score we developed the following function

$$L_{ij} = R(g_2^{-1}) + R(g_3^{-1}) + R(g_4) + R(g_5) + R(g_7^{-1}) \tag{18}$$

where $R(g)$ gives the rank for the pair from the ordered list of the descriptor g. After likeness scores are computed for the possible volume pairs, the pair having the maximum likeness score are merged together, the descriptors are updated accordingly.

Clustering is performed until the candidate volumes to be merged become inconsistent in motion. This is measured with the merged pairs variance descriptor g_2. Until the g_2 becomes larger than a threshold, clustering is executed. After clustering, an object-wise multi-resolution tree structure as illustrated in Fig. 5 is generated. Multi-resolution structure enables analyzing object properties by graph theory methods, and segmentation is not repeated in case the number of objects changes.

4 Conclusions

The proposed algorithm has been tested with standard MPEG-7 sequences. Fig. 6 shows sample test results for the *Children* video sequence. The first image (Fig. 6-a) is the 16^{th} frame from the original sequence. At the implementation, we selected markers with volume growing. The texture features are omitted. After the volume growing stage, 32 initial volumes are obtained as labeled with gray levels in Fig. 6-b. The trajectories of the initial volumes and object level $m = 7$ are given in Fig. 6 a-b, respectively. Figures 6 c-d-e-f are the clustering of volumes down to 20, 10, 7, 4 objects including the background. The backgrounds and object boundaries were detected accurately.

The test results confirmed that the object extraction is robust even when the motion is large. Because no computationally expensive motion computation is involved in segmentation, our algorithm

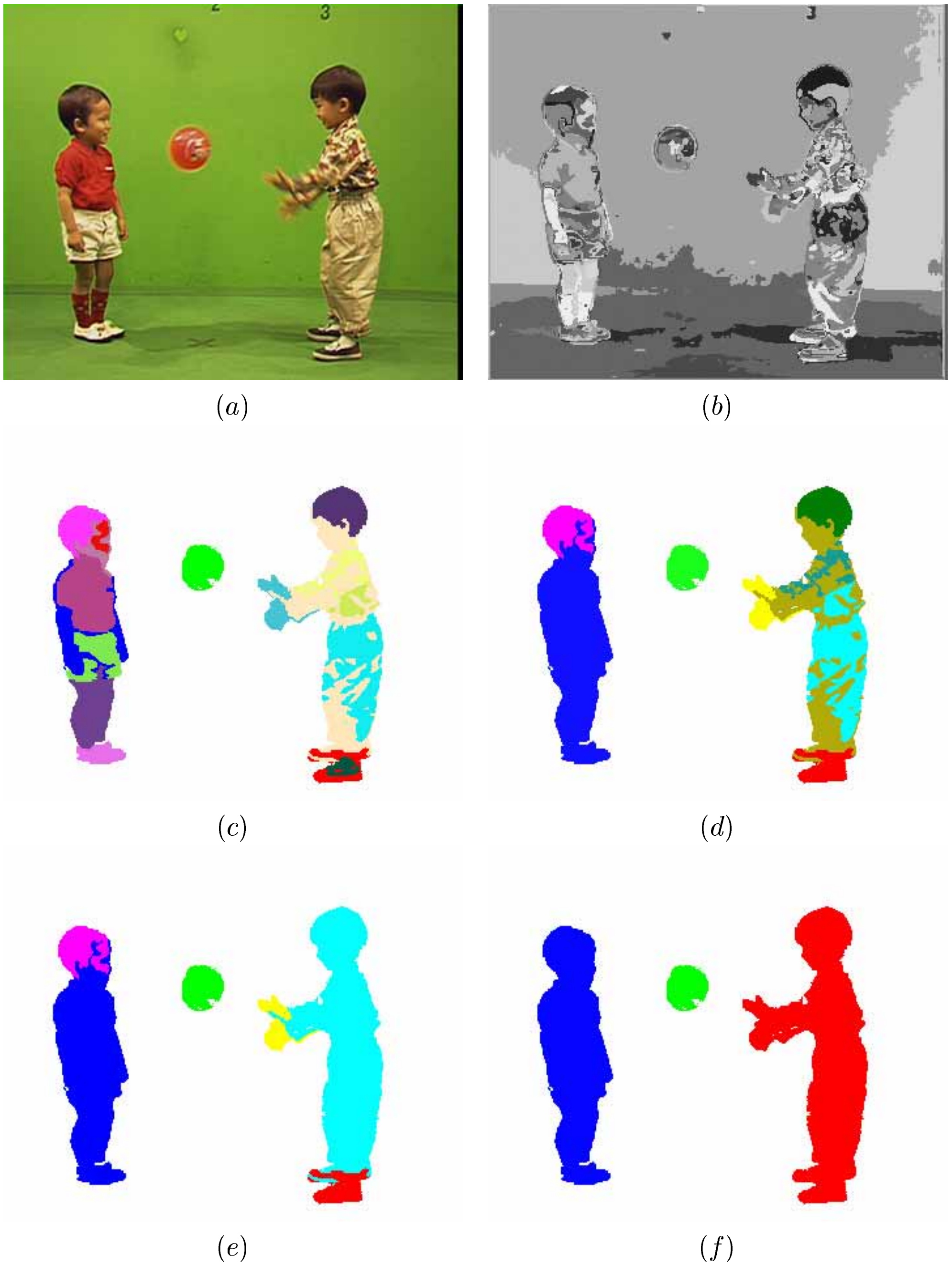

Figure 6: (a) A frame of input sequence, (b) after volume growing, (c-d-e-f) clustering results at object levels $m = 20, 10, 7, 4$, respectively. Background is counted as a separate object.

is faster than optical flow methods. Having an obvious advantage over the stochastic segmentation techniques, an object-wise multi-resolution representation is generated after clustering; therefore the extraction objects is not repeated in case the number of objects is changed. No user segmentation of object regions is involved, which is mostly required by object tracking based methods.

References

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

[2] P. Bouthemy and E. Francois: Motion segmentation and qualitative dynamic scene analysis from an image sequence. Int. J. Comput. Vision, no 10, 157–187, (1993)

[3] F. Meyer and P. Bouthemy: Region-based tracking using affine motion models in long image sequences. CVGIP-Image Understanding, 119–140, no 60 (1994)

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

[5] J. Wang and E. Adelson: Representing moving images with layers. IEEE Transaction on Image Processing, no.3 (1994)

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

[7] M. Kopp and W. Purgathofer: Efficient 3x3 median filter computations. Technical University, Vienna (1994)

[8] J. Serra and P. Soille: Watershed, hierarchical segmentation and waterfall algorithm. Proc. of Mathematical Morphology Appl. Image Process., 69–76 (1994)