Graph-Based Spatio-temporal Region Extraction

Eric Galmar and Benoit Huet

Institut Eurécom, Département multimédia, Sophia-Antipolis, France
{galmar, huet}@eurecom.fr

Abstract. Motion-based segmentation is traditionally used for video object extraction. Objects are detected as groups of significant moving regions and tracked through the sequence. However, this approach presents difficulties for video shots that contain both static and dynamic moments, and detection is prone to fail in absence of motion. In addition, retrieval of static contents is needed for high-level descriptions.

In this paper, we present a new graph-based approach to extract spatio-temporal regions. The method performs iteratively on pairs of frames through a hierarchical merging process. Spatial merging is first performed to build spatial atomic regions, based on color similarities. Then, we propose a new matching procedure for the temporal grouping of both static and moving regions. A feature point tracking stage allows to create dynamic temporal edges between frames and group strongly connected regions. Space-time constraints are then applied to merge the main static regions and a region graph matching stage completes the procedure to reach high temporal coherence. Finally, we show the potential of our method for the segmentation of real moving video sequences.

1 Introduction

Multimedia technologies are becoming important in many aspects of our nowaday lives. Processing of huge amount of raw data requires efficient methods to extract video contents. Achieving content-based functionnalities, such as search and manipulation of objects, semantic description of scenes, detection of unusual events, and recognition of objects has driven intensive research over the past years. To exploit video contents, shots must be decomposed into meaningful objects which are composed of space time regions. This process is called video *indexing*.

Unsurpervised extraction of video objects is generally based intensively on motion information. Two strategies are generally adopted. The first one searches for homogeneous colored or textured regions, and then groups the regions that undergo similar motion [1]. The second strategy performs motion estimation to yield coherent moving regions, then groups adjacent regions basing on color cues [2]. Sophisticated methods use robust motion estimation to deal with multiple objects and motion. However, tracking becomes difficult in case of non-rigid or fast motion, and the apparition and disappearance of new object models cannot be

A. Campilho and M. Kamel (Eds.): ICIAR 2006, LNCS 4141, pp. 236–247, 2006.

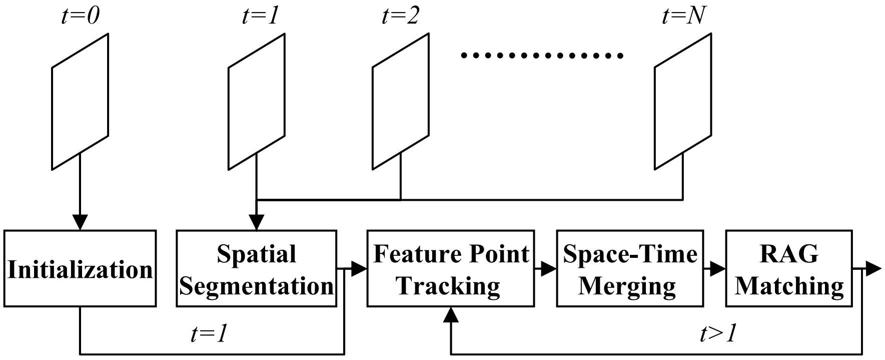


Fig. 1. Scheme of the overall segmentation process

integrated easily. To overcome these problems, two alternative approaches have been proposed, spatio-temporal segmentation and graph-based region merging.

The first category searches for meaningful volumes inside a block of frames to improve temporal coherence. A feature clustering approach is described in [3]. Elementary objects are represented as color patches with linear motion, called video strands. Space-time features describing color, position, and dynamics are extracted for each pixel. Therefore, video shot can be mapped to a 7D feature space representative of the strands. A hierarchical mean-shift technique is then employed to cluster pixels and build object hierarchy jointly. A probabilistic representation scheme is proposed in [4]. The video sequence is modeled by a succession of spatial gaussian mixture models (GMM). GMMs are initialized via EM algorithm in the first frame, then are updated on subsequent frames. Appearance of new objects is handled by thresholding a likelihood map and creating new models from unlabeled connected pixels. This allows the method to track coherent regions with complex motion patterns.

These spatio-temporal approaches are robust at the expense of memory bandwith and computational cost when the shot duration becomes important. In region merging approaches, the segmentation is first initialized on each frame from an image segmentation technique. Popular algorithms are derived from watersheds [5] or color quantization [6] and yield to segments with small color variations. Spatial and temporal merging are then achieved by labeling or matching.

Unlike pixel-based approaches, region-based graphs use more reliable region information and allow to represent various relationships between regions. In [7], a set of spatial region adjacency graphs (RAG) is built from a shot section, and then the optimal partition of the whole graph is found according to a global cut criterion. However, the method suffers from the instability of image segmentation on different frames. To make matching easier, Gomila et al. [8] reduce the difference between consecutive RAGs by a region splitting process. For each frame, a hierarchy of segmentations is generated through a multiscale image segmentation method. Closer RAGs are then built by checking if missing regions are edited in the decomposition. Then, the graphs are iteratively merged using a relaxation technique.

The proposed approach is closely related to both space-time and graph-based region-merging. It aims at decomposing video shots into spatio-temporal regions. Unlike other methods, we give particular attention to the stability of the projected spatial segmentation for both static and moving regions, in prospect of object detection and region-based shot representation. This paper is organized as follows. Section 2 provides an overview of the proposed algorithm and motivations for our approach. Section 3 introduces the efficient graph-based merging algorithm used at different stages of the process. In section 4, we describe the temporal merging procedure. Finally, experimental results illustrate the application of our algorithm to real video sequences in section 5.

2 Overview of the Proposed Approach

Extraction of space-time regions can be difficult when video objects show strong variations in color, texture or motion. Unfortunately, these features are common in real video sequences. In this work, we design an incremental scheme to reduce the complexity of region grouping and matching tasks.

A block diagram of our system is shown figure 1. The segmentation is initialized on the first frame of the shot from coherent spatial regions and defines the spatial level of details of the segmentation. A graph-based segmentation algorithm is used for this purpose. Then, the method iteratively processes frame pairs. The regions are grouped temporally in three steps. The first stage builds slightly oversegmented spatial regions in the new frame, so that these new regions corresponds to a partition of the previous segmentation. Instead of using motion compensation, we track a population of feature points to create dynamic temporal edges between regions. This allows us to group with high confidence static and moving regions that are strongly connected. We then complete the temporal linkage of static regions using local edges, under space-time merging constraints. At this stage, the segmentation maps become close and region neighborhoods can be compared. Finally, we test the validity of new regions by comparing locally RAGs between frame pairs.

With this design, we achieve incremental merging with strong rules, reaching progressively temporal coherence for various region types.

3 Spatial Merging

In this section, we present the efficient graph segmentation algorithm introduced in [9]. Then we describe how we apply it to initialize regions and how we adapt it for partial segmentation of new frames.

3.1 Efficient Graph Based Segmentation

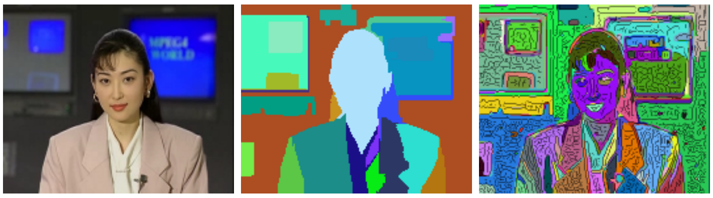
Let $G = \{V, E\}$ be a weighted undirected graph. Each vertex is a pixel. The algorithm aims to decompose G into a partition $S = \{C_1, C_2, \dots, C_k\}$ of G, where each component is a minimum spanning tree (MST). The procedure is similar to Kruskal's algorithm, with addition to a merging criterion to limit the grouping of components. At each step, two components are merged if the minimum edge connecting them is weaker than the maximum edges of the components plus a tolerance depending on the component size. Therefore, fewer merge are done when the region size increases. Thanks to the adaptive rule, the algorithm is sensitive in areas of low variability whereas it remains stable in areas of high variability preserving both local and global properties.

We apply this algorithm to segment the first image by building the graph on a pixel grid, so that the algorithm is fast and subgraphs correspond to spatially connected regions. In the experiments, the weights are built using color distance.

3.2 Edge Constrained Segmentation

Using directly the procedure described in 3.1 to initialize regions in any frame does not work, since the segmentation may differ substantially from one frame to another. To avoid resegmentation, we adapt the method so that S_t is oversegmented compared with S_{t-1}. To this aim, we use an edge detection map $\mathscr{C}_t$ to discard possible region boundaries from the merge. Thus, the propagation is done in areas of low-variability, resulting in more homogeneous components.

Edge-constrained segmentation and original image segmentation can be compared in figure 2. We can see that the constrained method (c) results in a decomposition, or oversegmentation of the unconstrained one (b). In addition, since we use Canny detection, edges with local intensity variations are also pruned so that the components are more homogeneous.



(a) (b) (c)

Fig. 2. (a) Input Image. (b) Unconstrained image segmentation used as initialisation. (c) Edge-constrained initialisation of the new regions.

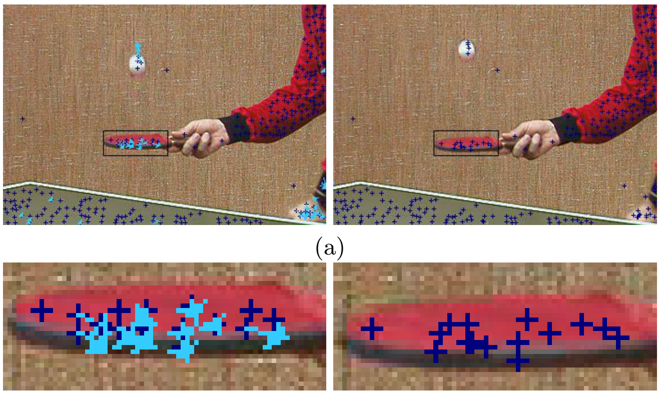
4 Temporal Grouping

In this section, we first describe the temporal grouping of regions based on dense feature points and space-time constraints. Then, we show how RAGS are employed to check efficiently the stability of the regions.

4.1 Feature Point Matching

The regions in the previous segmentation S_{t-1} have various shape, size and possibly non-rigid motion. In addition, regions might be partially occluded in the new frame, so that one region can have several matches in the next frame. In this case, traditional motion compensation cannot be used. Our solution is to group new oversegmented regions by spreading a population of feature point trackers $\mathbf{P_f}$. In this way, no hypothesis is made on motion models and we avoid optical flow computation on full regions.

Feature point trackers have been proposed by Tomasi et al. [10]. Good feature points are extracted from corners or textured regions. However, these points are likely to correspond to region borders, thus hampering the matching between regions. Therefore, we rather consider flat points that we can expect to lie reliably inside regions, at the expense of motion precision. Feature points are then tracked using a block matching algorithm. Figure 3 shows typical feature point detection and tracking. We can see that feature points are concentrated in homogeneous areas (fig. 3a). Even if some tracked points are inaccurate (fig. 3b), they can be considered as outliers in the statistical distribution of the points. We explain how we use these points for region grouping in the next section.



(a)

(b)

Fig. 3. (a) Distribution of feature point matches. (b) Feature points inside the racket. Arrows represent the estimated displacement.

4.2 Region Grouping with Feature Points

Feature points matches described in the previous section can be viewed as potential inter-frame edges between pair of regions. We construct a 3D graph $G_T = \{V_T, E_T\}$ between two consecutive frames. The node set E_T contains two subsets of regions A and B generated from S_{t-1} and S_t. The edge set contains inter-frame arcs generated from feature point pairs. Due to possible high variations (section 4.1), grouping based on single linkage will no be relevant. We consider instead robust grouping analysing statistical properties of connections between subsets A and B.

The procedure (fig.4) is based on a sequence of tests. We first simplify the graph and prune weak connections between A and B with the M_{strong} test.

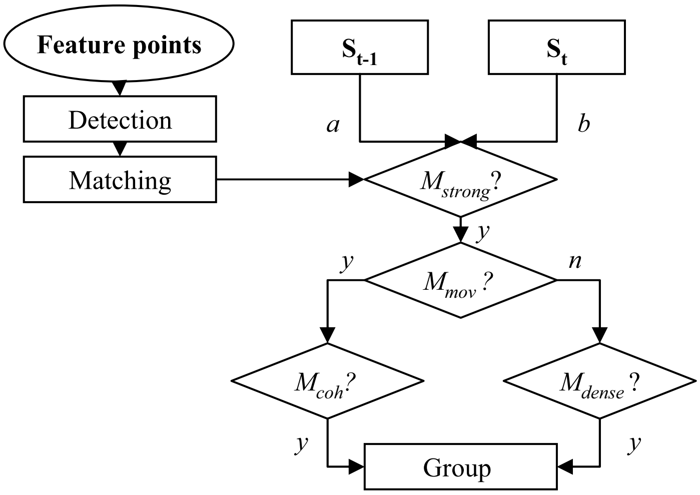


Fig. 4. The temporal region grouping scheme. Feature points are first detected and matched. Then, the temporal merging of strongly connected regions is performed using a hierarchy of tests.

Second and third tests (M_{mov} and M_{coh}) verify if region couples undergo significant and similar motion. This helps to detect potential splitting regions. Finally, we further check the homogeneity of region couples (M_{dense}) for static regions. Denote by $a \in A$ and $b \in B$ two candidate regions for grouping.

$\mathbf{M_{strong}}$: Two regions a and b are strongly connected if there is a significant proportion of arcs linking a to b. Formally, we compare the cut between a and b to the degrees of a and b. The test is accepted if :

$$cut(a,b) > \alpha \min deg(a), deg(b) \tag{1}$$

α is fixed to $\alpha = 0.5$ in our experiments. In other words, if edges are given equal weights, the test is verified when at least half edges of either a or b connects a to b. Once all regions have been tested, weak edges that do not satisfy the condition are pruned.

$\mathbf{M_{mov}}$: From the displacement of feature points, we deduce information on region motion. For this purpose, we map the points to a velocity space $\mathscr{D} = [d_n, \beta d_\theta]$ where d_n is the displacement norm and d_θ is the motion orientation. β controls the influence of orientation information with respect to motion speed. In case that there is substantial background or camera motion, the displacements are compensated with the mean velocity of the complete set of points. The test separates moving regions from static regions. The moving condition is given by

$$d_n(a) > d_{mov} \tag{2}$$

where d_{mov} is a minimum substantial displacement. Default value is $d_{mov} = 3$ in all our experiments.

$\mathbf{M_{coh}}$: If a and b are moving regions, they must undergo coherent motion to be grouped. A simple measure is to compare the variance of the velocity distributions of a, b and $a \cup b$. The test $M_{coh}(a, b)$ is given by

$$tr(C_{a \cup b}) < \gamma(tr(C_a) + tr(C_b)) \tag{3}$$

where C_a denotes the covariance matrix of the velocity points of a. The test favors the creation of new moving regions in S_t when one region in S_{t-1} is matched to ones with different motions. In this way, we handle apparition of new moving regions.

$\mathbf{M_{dense}}$: When either region has no motion, we further check if they have comparable homogeneity. We characterise this feature by the density of feature points between regions, since each point corresponds to a local maximum of homogeneity. The density η_a of one region a is estimated by

$$f_a = \frac{card(a \times V_T)}{size(a)} \tag{4}$$

As the density is variable over the regions, we use a statistical proportion test for that purpose. Let's consider two parent populations P_a and P_b representing space-time regions and their final proportion of points p_a and p_b. a and b are samples drawn from P_i and P_j. f_a and f_b are estimations of p_a and p_b.

We consider the following hypotheses

$$\begin{aligned} H_0 &: p_a = p_b \\ H_1 &: p_a \neq p_b \end{aligned} \tag{5}$$

Assuming normal laws for P_a and P_b , it is possible to check if we can accept H_0 with a significance level α [11].

At the end of the process, temporal grouping has been performed reliably on homogeneous moving regions. To group more textured areas on the sequence, the population of seed points will be increased inside regions finally created in S_t, i.e. if they have not been matched in S_{t-1}. In this way, the tracked points will focus progressively on the regions of interest.

4.3 Grid-Based Space-Time Merging

We complete the segmentation S_t by a space-time merging technique applied on the unmatched regions. The method is an adaptation of the efficient graph algorithm discussed in section 3.2 for grouping components spatially and temporally. We construct a space-time pixel grid on a 3D volume bounded by two successive frames. As in [9] each component C_i is characterized by its internal variation, which represents a p-quantile of the weight distribution of the edges inside C_i. However, this turns out to be too complex in practice and we use the mean weight μ_i of C_i as a measurement. When comparing two components C_i and C_j , a new space-time merging rule is applied to examine both local and global properties of the grouping:

$$\|\mu_i - \mu_j\| < \tau_G \text{ and } \max(W_L) < \tau_L \tag{6}$$

where

$$\tau_G = \max(T_G, p_G \min(\mu_i, \mu_j)) \tag{7}$$

$$\tau_L = \min(T_L, \mu_i) \tag{8}$$

τ_L and τ_G are local and global adaptive thresholds. Default parameters are $T_G = 10$, $p_g = 0.3$, $T_L = 5$ in all experiments. For local properties, we define a four edge neighborhood W_L (fig. 5a). The neighborhood is considered as homogeneous if the maximum weight is weak compared to the variability μ_i and T_L. Small values of T_L limit grouping in inhomogeneous areas. In this way, we do not merge component from edges with high variability. For global properties, we check if the components have similar homogeneity. For regions with strong homogeneity, we consider directly the distance between μ_i and μ_j. For more variable components, a tolerance p_g is accepted on the relative error between μ_i and μ_j. Small values of T_G and p_g limit the temporal variation of the components.

Thus, by combining these two aspects, the merging occurs in space-time areas of low local variability on globally coherent components.

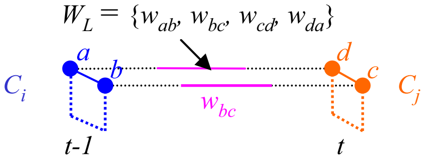


Fig. 5. Space-time grid based merging and local neighborhood W_L

4.4 Subgraph Matching

The last step in the process is to confirm the creation of new regions by analysing region neighborhoods at time $t-1$ and t. Thanks to the previous merging steps, segmentations S_t and S_{t-1} are sufficiently close to be compared. We consider, as in section 4.2, a 3D graph on a volume bounded by two successive frames. The graph contains region adjacency graphs (RAG) R_{t-1} from S_{t-1} and R_t from S_t. It also includes inter-frame edges corresponding to the temporal grouping of regions. For each node v in R_t, we define its neigborhood subgraph $G_t^N(v)$ as the smallest subgraph containing all its adjacent nodes $u \in R_t$. Let v_n be a node from a new region in R_t and $u \in G_t^N(v_n)$ connected to a node $u' \in R_{t-1}$. Let consider a distance measure $d(u, v)$ between two nodes. We denote by u'' a node in $G_t^N(u')$. u'' and v_n are matched temporally if

$$d(u'', v_n) < \min_{z \in G_{t-1}^N(u'')} d(u'', z) \tag{9}$$

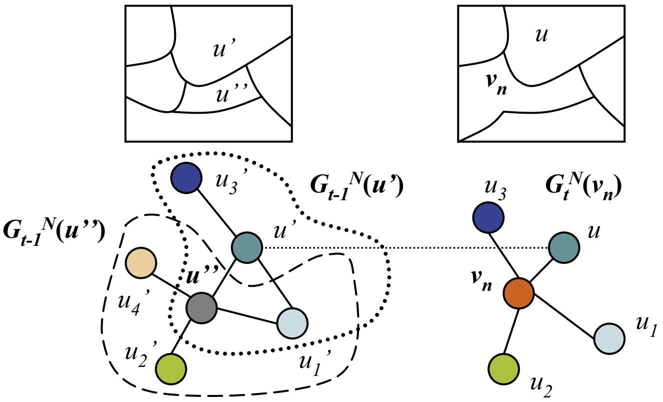
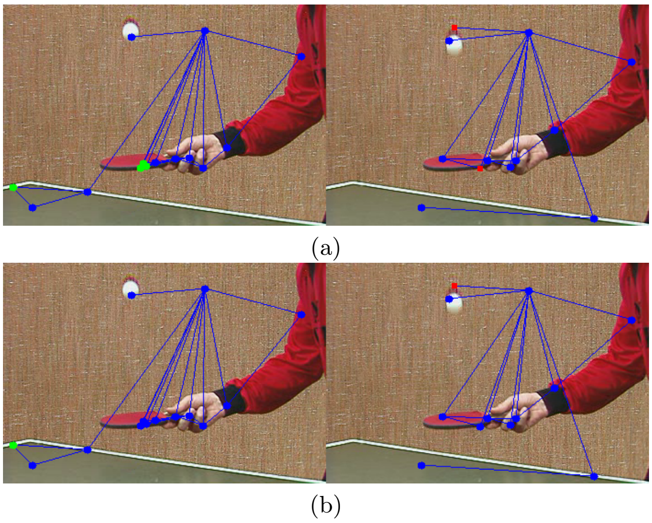


Fig. 6. Neighborhood subgraphs for matching new nodes v_n. For each node $u \in G_t^N(v_n)$, the neigborhood of u in RAG_{t-1}, $G_{t-1}^N(u')$ is examined. Lost nodes u'' are then retrieved by comparing v_n to adjacent nodes of u'' in $G_{t-1}^N(u'')$.

Equation 9 checks if an untracked node in R_{t-1} can be matched with a new node in R_t in the proximate neighborhood (fig. 6). In this way, lost objects can be recovered in case of fast motion or homogeneity changes. For the distance measure, the node attributes represent dominant color (c) and size (s) of the regions. For two nodes u and v, the distance is given by


(a)

(b)

Fig. 7. Subgraph matching. Untracked nodes are shown as green (clear) rounds, tracked nodes as dark (blue) rounds and new nodes as (red) squares. (a) RAGs before matching. (b) RAGs after matching.

$$d(u,v) = |c_u - c_v|^2 \, \frac{s_u s_v}{s_u + s_v} \tag{10}$$

Thus, we favor the grouping of smaller regions with similar attributes.

An example of matching is shown figure 7 on the *tennis* sequence. Before matching (fig. 7a), untracked regions are located in the racket and the table left corner. The new regions are located above the ball and inside the racket border. After matching (fig. 7b), the nodes at the racket border have been grouped as they have close similarity, whereas the table left corner is not linked to any new node and thus cannot be reliably tracked.

5 Experimental Results

In this section, we test the proposed method on various real video sequences. We analyse the segmentation results on the *akiyo*, *tennis*, and *walking* sequences (CIF format). The processing time is about 1s per frame on a 2.8GHz PC with unoptimized code.

Figure 8 shows the final spatio-temporal segmentation, i.e. when all the frames have been processed. In figure 8a, the video is composed of stationary background and slow head motion. We see that the main regions are the woman

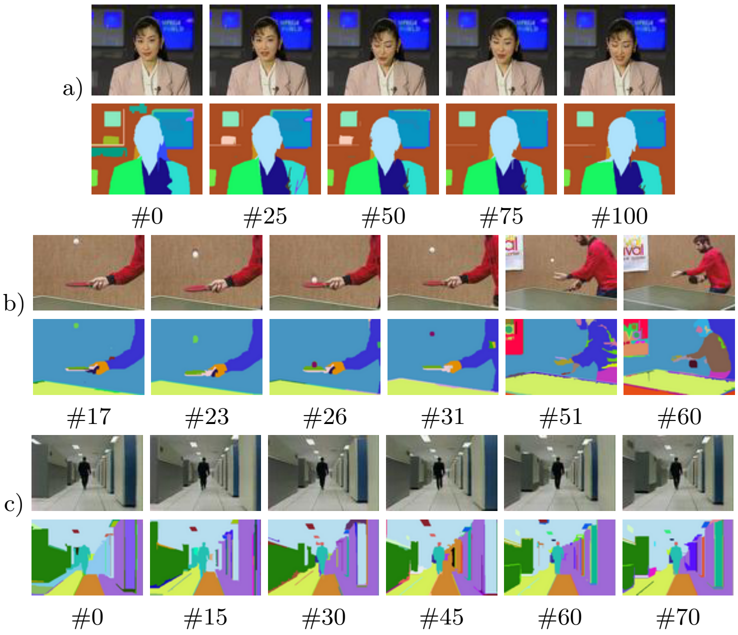


Fig. 8. Segmentation results. a) *akiyo* sequence. b) *tennis* sequence. c) *walking* sequence.

and the TV screens which have smooth spatial variations whereas tiny varying components such as face elements are not kept. In the next images, face moving elements are detected, but they are too tiny to be extracted from the sequence. In consequence, these elements are incorporated into the face.

In figure 8b, the video is composed of several motions. The ball and the racket undergo rigid motion whereas the player undergoes non rigid-motion. Besides theses motions, the camera is zooming out during the entire sequence. We see that the ball region remains until it hits the racket in frame #26. As the ball was speeding up in previous frames, the ball and its shadow were splitted into two adjacent regions. The similarity between these regions is lower than their temporal similarity with the new ball region, so that a new region is created for the ball. The ball is tracked successfully until frame #31. From this moment on, the camera quickly zooms out and the ball becomes smaller and less homogeneous. As a result, the ball sometimes does not appear after the spatial merging stage. However, the other regions, which are larger and more stable, such as the table, the racket and the hand are correctly segmented during the whole sequence. Finally, we can see that a strong scale change happens gradually between frame #31 and frame #60. While the player is appearing progressively at the left of the image, the corresponding regions are splitted until fitting the body of the player. In this way, the segmentation follows the temporal changes of the video.

In the last sequence (8c), the camera is tracking the walking man so that the walls surrounding him are moving towards the foreground and exiting the frame progressively. In the first frame #0, the regions are composed of the man, the tiled floor, the walls, the ceiling and the lamps. The man region remains consistent along the sequence, just as the different parts of the walls and the lights until they exit the frame. We can further notice that apparent static regions such as the floor and the ceiling are coherent in the entire sequence.

These results illustrate the potential of the method to extract coherent volumes from video shots. Given a level of details, both moving and static elements can be tracked thanks to our hierarchical matching stage. Besides, we handle dynamic temporal changes by favoring the creation of new regions when some regions cannot be reliably matched between frame pairs. In this way, we achieve good compromise between the span and the consistency of the regions. Therefore, the method can help higher level grouping tasks considerably.

6 Conclusion

We have proposed a new method for extracting meaningful regions from videos. Graphs appear as an efficient solution to build space-time relationships at different levels. We have used both pixel-based graphs to build low-level regions and RAGs to enforce consistency of the regions. We have proposed a temporal grouping method exploiting feature points to handle both static and dynamic regions. Finally, encouraging results show that the method is promising as a preliminary step for object-based video indexing and retrieval.

Acknowledgments

This research was supported by the European Commission under contract FP6-027026-K-SPACE.

References

1. D. Zhong and S. Chang. Long-term moving object segmentation and tracking using spatio-temporal consistency. In *ICIP'01*, volume 2, pages 57–60, Thessaloniki, Greece, Oct. 2001.
2. H. Xu, A. Younis, and M. Kabuka. Automatic moving object extraction for content-based applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(6):796–812, June 2004.
3. D. DeMenthon and D. Doermann. Video retrieval using spatio-temporal descriptors. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 508–517, Berkeley, CA, USA, Nov. 2003.
4. H. Greenspan, J. Goldberger, and A. Mayer. Probabilistic space-time video modeling via piecewise gmm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):384–396, Mar. 2004.
5. L. Vincent and P. Soille. Watersheds in digital space: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, Aug. 1991.
6. Y. Deng and B. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):800–810, Aug. 2001.
7. C. Yuan, Y. F. Ma, and H. J. Zhang. A graph theoritic approach to video object segmentation in 2d+t space. Technical report, MSR, 2003.
8. C. Gomila and F. Meyer. Graph-based object tracking. In *ICIP'03*, volume 2, pages 41–44, 2003.
9. P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, Sept. 2004.
10. J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, Seattle, WA, USA, 1994.
11. J. S. Milton and J. Arnold. *Introduction to Probability and Statistics.* McGraw-Hill, 2002.