Tracking video objects in cluttered background

Andrea Cavallaro, Olivier Steiger, Touradj Ebrahimi

Abstract— **We present an algorithm for tracking video object which is based on a hybrid strategy. This strategy uses both object and region information to solve the correspondence problem. Low level descriptors are exploited to track object's regions and to cope with track management issues. Appearance and disappearance of objects, splitting and partial occlusions are resolved through interactions between regions and objects. Experimental results demonstrate that this approach has the ability to deal with multiple deformable objects, whose shape varies over time. Furthermore, it is very simple, because the tracking is based on the descriptors, which represent a very compact piece of information about regions, and they are easy to define and track automatically. Finally, this procedure implicitly provides one with a description of the objects and their track, thus enabling indexing and manipulation of the video content.**

Keywords— **Object tracking, low level descriptors, object segmentation, indexing.**

I. Introduction

Object-based representations of multimedia content provide the user with flexibility in content-based access and manipulation. International standards such as MPEG-4 and MPEG-7 allow for interoperability between different multimedia systems based on the object-based representation. However, the automatic isolation of video objects from video data is still an open problem. This problem is also referred to as *video object extraction*. Video object extraction can be decomposed into two sub problems, namely video object segmentation and video object tracking. Video object segmentation aims at identifying objects in the scene and separating them from the background. Video object tracking aims at following video objects in the scene and at updating their 2D shape from frame to frame. After a frame of the image sequence has been segmented into objects, the objects are tracked in the subsequent frames. The aim of temporal tracking is to establish a correspondence between instances of video objects over frames.

Video object tracking algorithms should be able to deal with the various dynamics in the scene. The goal is to establish a stable track for each object. A stable track results from an effective track management. The main problems to be solved in track management are track initiation, track update, and track termination. The main obstacles to effective track management are the temporal variations of the 2D shape of video objects due to perspective and motion of non-rigid objects, occlusions and other interactions between objects, splitting of one object, appearance and disappearance of objects [1].

A. Cavallaro is with the Multimedia and Vision Laboratory, Queen Mary, University of London (QMUL), London, United Kingdom. E-mail: andrea.cavallaro@elec.qmul.ac.uk.

O. Steiger and T. Ebrahimi are with the Signal Processing Institute, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. E-mail: {olivier.steiger,touradj.ebrahimi}@epfl.ch.

Object tracking methods can be classified into five groups: *model-based*, *appearance-based*, *contour-* and *mesh-based*, *feature-based*, and *hybrid* methods. *Model-based* tracking methods exploit the a *priori* knowledge of the shape of typical objects in a given scene [2]. The definition of parameterized object models makes it possible to solve the problem of tracking partially occluded objects. However, this approach is computationally expensive and presents two major drawbacks. One is the need for object models with detailed geometry for all objects that could be found in the scene, the other is the lack of generality. This last drawback prevents the system from detecting objects that are not in the database.

Appearance-based methods track connected regions that roughly correspond to the 2D shapes of video objects based on their dynamic model. The tracking strategy relies on information provided by the entire region [3], [4], [5], [6]. Examples of such information are motion, color, and texture. These methods cannot usually cope with complex deformation.

Instead of tracking the whole set of pixels comprising an object, *contour-based* methods track only the contour of the object. Tracking methods based on contours rely on motion information to first project the contour, and then adapt it to the object detected in the next frame [7]. The computational complexity is high, and large non-rigid movements cannot be handled by the method. This difficulty is due to the rigid body motion projection followed by adjustment. One improvement of the previous method is to use a deformable object motion model, such as active contour models (snakes) [8], [9], [10], or meshes [11], [12]. A contour-based representation can reduce the computational complexity. Furthermore, it allows tracking of both rigid and non-rigid objects. However, it is unable to track objects that are partially occluded. To overcome the problem of partial occlusions, a Kalman filtering approach and optical flow measurements have been introduced in the active contour model [9]. 2D meshes have also been used to track video objects [11]. This representation of motion and shape related features of video objects is based on the assumption that the initial appearance of the object can be specified and the object motion can be modeled by a piecewise affine transformation.

The fourth group of tracking methods uses features of a video object to track parts of the object. Several *feature-based* tracking techniques have been proposed, but they are not specifically designed for video object tracking. An adaptation to object tracking is presented in [13]. Here, the parts to be tracked are the corners of the objects. Tracking parts of objects results in stable tracks for the features under analysis even in case of partial occlusion of the object. However, the problem of grouping the features to deter-

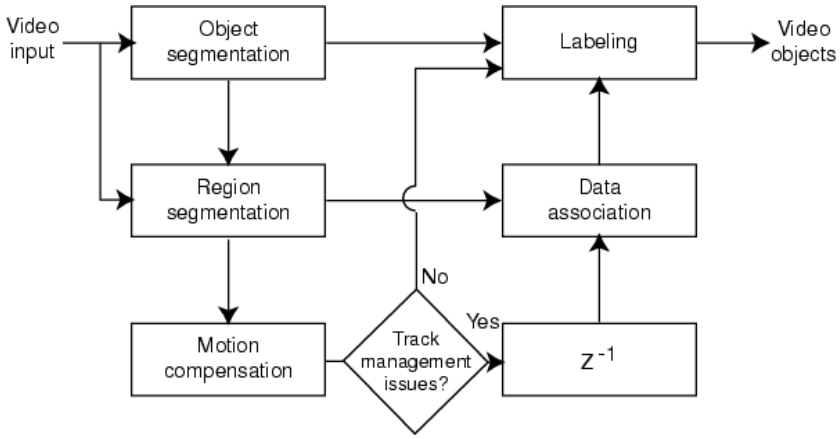


Fig. 1. Block diagram of the proposed hybrid tracking algorithm

mine which of them belong to the same object is a major drawback of these approaches.

The last group of tracking approaches is designed as a hybrid between a region-based and a feature-based technique [14], [15]. They exploits the advantages of the two by considering first the object as an entity and then by tracking its parts. These algorithms exploit an image representation as partition hierarchy and track video objects based on interactions between different levels of the hierarchy. The hierarchy is composed of a object level and a region level. The object level defines the topology of the video objects. The region level defines the topology of homogeneous areas constituting the objects. This characteristic allows the tracking system to deal with the deformation of objects. This flexibility is obtained at the cost of a higher computational complexity. Such complexity is due to the use of complex motion models to project and adapt the regions from one frame to another.

In this paper, we propose a hybrid object tracking algorithm that overcomes such limitations. The temporal evolution of the object partition is computed through interactions with the region partition. These interactions exploit the tracking of the region partition to associate the data from two successive object partitions, thus resulting in a multi-level tracking algorithm. A distinctive feature of the proposed algorithm is to operate on region descriptors instead of regions themselves. Projecting a region descriptor instead of the entire region is a simple and effective strategy. The simplicity comes from the fact that instead of projecting the entire region into the next frame, only the region descriptor needs to be processed. Therefore, there is no need for computationally expensive motion models. In addition, region descriptor projection is effective, since it can cope with deformation and complex motion, when updating the feature values in the region descriptor by refining the predicted region partition. The details of the method are presented in Section II. First, the computation and the tracking of the region partition is described, and then the interactions between region and object partitions to obtain video object tracking are commented. Next, tracking results are discussed in Section III. Here the capabilities and the limits of the proposed approach are discussed. Finally, Section IV concludes the paper.

II. Hybrid video object tracking

For each frame n, objects are defined by an object partition, Π_o^n, whereas objects' parts are defined by a region partition, Π_r^n. The tracking mechanism is based on feedbacks between the object and the region partitions. These interactions are the core of the proposed tracking algorithm and allow us to cope with multiple simultaneous objects, motion of non-rigid objects, partial occlusions, and appearance and disappearance of objects. No restriction on the way the objects are extracted is imposed. The block diagram of the proposed approach is depicted in Figure 1. The *object segmentation* module receives the video input and produces an object partition. In the practical implementation the object partition is generated by the method presented in [16]. This method is based on change detection. The change detection strategy is designed to be immune to sensor noise. To this end, image differentiation is followed by a probability-based test that adapts the change detection threshold locally. The resulting object partition identifies the objects from the background and provides a mask defining the areas of the image containing the moving objects (Figure 2). Since the result of change detection is the classification of the pixels into two classes, namely foreground and background, a change detection algorithm provides no information about different objects in the scene. For this reason, further processing is required to track the video object.

Only the areas belonging to the object partition are considered by the *region segmentation* step. This step takes into account the spatio-temporal properties of the pixels in the computed object partition and extracts homogeneous regions. Each object is processed separately and is decomposed into a set of non-overlapping regions. Homogeneous regions are detected using a multi-feature clustering approach [17]. The feature space used here is composed of spatial and temporal features. Spatial features are color features from the perceptually uniform color space CIE *Lab*, and a measure of local texturedness based on vari-

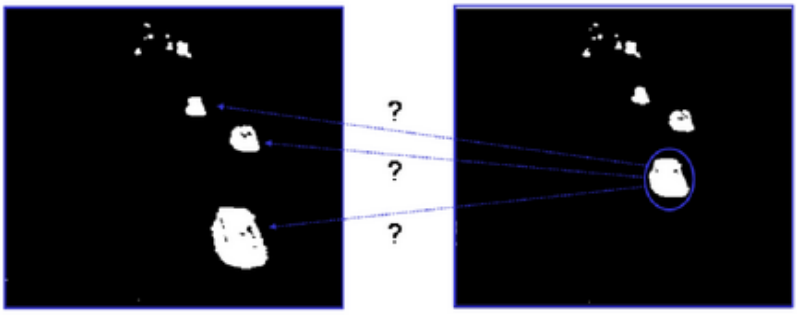


Fig. 2. Example of object partition in two successive frames. The tracking algorithm is responsible for solving the correspondance problem between two temporal instances (masks) of the same object

ance. The temporal features used here are the displacement vectors from the optical flow computed via block matching. The selected clustering approach is based on spatially unconstrained fuzzy-C-means, where a specific functional is minimized based on local and global feature reliability. Local reliability of both spatial and temporal features is estimated using the local spatial gradient [18]. The estimation is based on the observation that the considered spatial features are more uncertain near edges, whereas the considered temporal features are more uncertain on uniform areas. Global reliability is estimated by considering the variance of the features in the entire image compared to the variance of the features in a region. An example of region segmentation is given in Fig. 3. Being the clustering algorithm spatially unconstrained, homogeneous regions in the feature space may result in disconnected regions in the image space (e.g., Figure 3, frame $n+4$). This favors the flexibility of the tracking mechanism in case of deformable objects. Each region is then represented by a region descriptor. The region descriptor summarizes the value of the features in the corresponding region. Next, the tracking mechanism operates on region descriptors. Such mechanism is composed of a *motion compensation*, a *data association*, and a *labeling* stage and as described in details in the following sections.

A. Region tracking

Region tracking is based on a flexible procedure, that exploits the region descriptors in two steps. The first step projects the region descriptors from the current frame onto the next frame, and implicitly provides a predicted region partition. The second step refines the region partition, as to naturally create the updated 2D topology.

The first step for tracking the region partition is the projection of the information at the current frame n into the next frame $n+1$. Each region, $R_i(n)$, is projected by applying motion compensation to its region descriptor, $\mathbf{\Phi}_i(n)$. This operation is referred to as region descriptor projection. Region descriptor projection updates the position values of a region descriptor by means of its estimated displacement. The region descriptor is defined as

$$\mathbf{\Phi}_i(n) = \left(\phi_i^1(n), \phi_i^2(n), \phi_i^3(n), \phi_i^4(n), \ldots, \phi_i^{K_i(n)}(n)\right)^T \quad (1)$$

where $K_i(n)$ is the number of features in frame n. Let $\left(\phi_i^1(n), \phi_i^2(n)\right)$ represent the position of the region descriptor, and $\left(\phi_i^3(n), \phi_i^4(n)\right)$ its motion vector. The position and the motion vector of the region descriptor are given by the center of mass and by the mean displacement of the pixels belonging to the corresponding region.

In the specific implementation, $K_i(n) = 8$. In particular, $\left(\phi_i^5(n), \phi_i^6(n), \phi_i^7(n)\right)$ represents the mean value of the three color components in the corresponding region, and $\phi_i^8(n)$ the mean value of the texture feature [17]. The number and the type of features can change according to the application at hand.

The position predicted through motion compensation is given by

$$\begin{cases} \widetilde{\phi}_i^1(n+1) = \phi_i^1(n) + \phi_i^3(n) \\ \widetilde{\phi}_i^2(n+1) = \phi_i^2(n) + \phi_i^4(n) \end{cases} \quad (2)$$

The predicted region descriptor, $\widetilde{\mathbf{\Phi}}_i(n+1)$, retains the value of the other features unchanged from frame n to frame $n+1$, so that

$$\widetilde{\mathbf{\Phi}}_i(n+1) = \left(\widetilde{\phi}_i^1(n+1), \widetilde{\phi}_i^2(n+1), \phi_i^3(n), \phi_i^4(n), \ldots \right. \quad (3)$$
$$\left. \ldots, \phi_i^{K_i(n)}(n)\right)^T$$

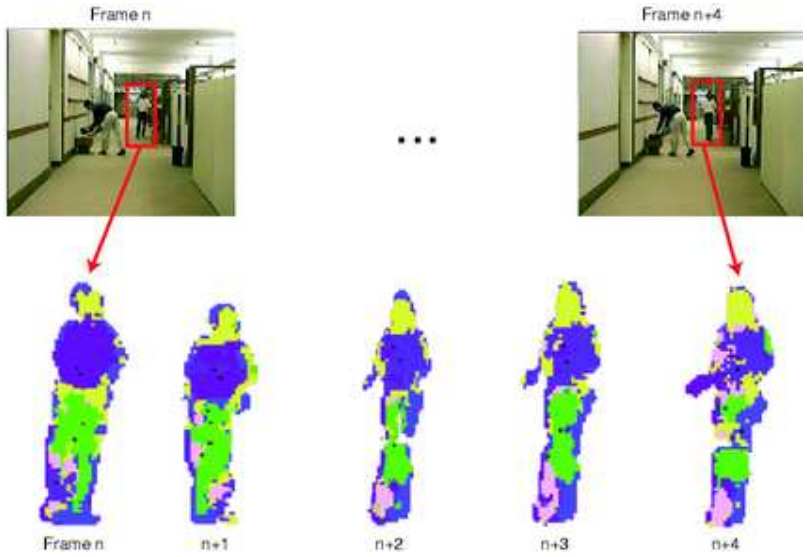


Fig. 3. Example of region segmentation. Homogeneous regions are computed in each object based on motion, color, and texture information

The result of region descriptor projection is a prediction of the region partition $\widetilde{\Pi}_r^{n+1}$ in the next frame.

The estimated feature values of the projected region descriptors should be refined to adapt the representation to the changes in the scene, to correct the inaccuracies of the projection, and to compensate for changes in viewing conditions. In fact, besides the changes related to the dynamics of the scene, the visual attributes of region descriptors are modified over time due to noise from many sources. Examples of such sources are motion estimation errors, local illumination variations, and sensor noise.

The refinement of the predicted region partition takes place naturally through region segmentation. The projected region descriptors, $\widetilde{\mathbf{\Phi}}_i(n+1)$, provide an effective initialization for the clustering process in the next frame. In addition, this initialization implicitly defines a correspondence between regions in frame n and $n+1$. The updated region partition Π_r^{n+1} is obtained through the clustering process described in [17]. An updated region descriptor, $\mathbf{\Phi}_i(n+1)$, defined as

$$\mathbf{\Phi}_i(n+1) = \Big(\phi_i^1(n+1), \phi_i^2(n+1), \phi_i^3(n+1), \ldots \quad (4)$$
$$\ldots, \phi_i^{K_i(n+1)}(n+1)\Big)^T$$

is finally associated to each region.

B. Multi-level region-object tracking

The correspondence of video objects in successive frames is achieved through the correspondence of objects' regions. Defining the tracking based on the parts of objects, leads to a flexible technique that exploits the characteristics of the video object tracking problem.

Given the object partition in the new frame and the region partition in the current frame, the proposed tracking procedure performs two different tasks.

1. It defines a correspondence between the object partition in the current frame n and the object partition in the new frame $n+1$.
2. It provides an effective initialization for the clustering procedure of each object in the new frame $n+1$. This initialization implicitly defines a preliminary correspondence between the regions in frame n and the regions in frame $n+1$.

The joint region-object tracking mechanism is organized in two major steps: object partition validation, and data association. The object partition validation step is a feedback from the region partition level to the object partition level, and results in a tentative correspondence. The data association step operates at low-level, and validates the track through region descriptor correspondence. This second step generates the final correspondence.

B.1 Object partition validation

The object partition validation step initializes the tracking process and improves the accuracy of the object partition in case the physical objects in the scene are connected in the image plane. This is achieved through a top-down and a bottom-up interaction with the region partition (Figure 4). Before initializing the tracking procedure, each video object is decomposed into a set of non-overlapping regions (Figure 4, frame n). Each region $R_j(n)$ is characterized by its region descriptor $\mathbf{\Phi}_j(n)$. To initialize the tracking procedure, each region descriptor $\mathbf{\Phi}_j(n)$ is associated to the corresponding object, $O_i(n)$. After this association, the region descriptor is denoted with $\mathbf{\Phi}_{i,j}(n)$. This operation, referred to as track initiation, can be expressed as

$$\forall\, O_i(n) \quad i = 1, \ldots, N_F^n \quad \exists\, \mathbf{\Phi}_{i,j}(n) \quad j = 1, \ldots, N_{R_i}^n, \quad (5)$$

with N_F^n number of video objects in frame n, and $N_{R_i}^n$ number of regions for object $O_i(n)$. This initialization takes

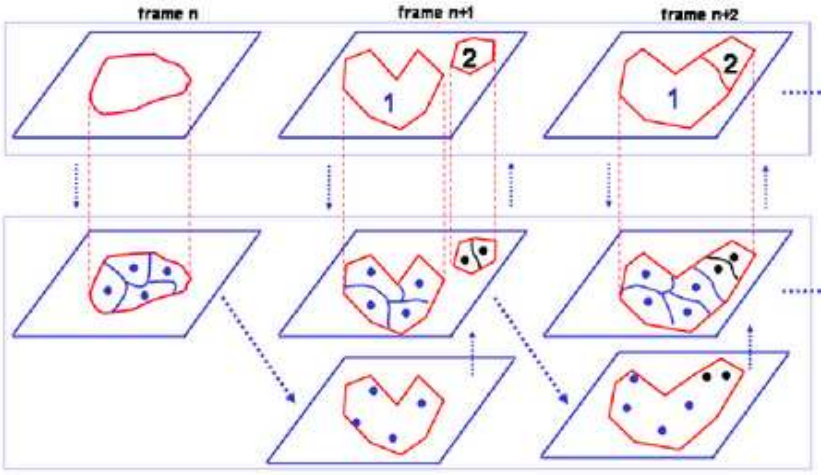


Fig. 4. Multi-level region-object tracking. The temporal evolution of the object partition is computed through interactions with the region partition. These interactions exploit the tracking of the region partition (bottom) to associate the data from two successive object partitions (top)

place at the beginning of the tracking process and every time a new video object appears. In this context, a new video object is defined as a set of connected pixels in the object partition which is not associated to another tracked object.

After the initialization, the region descriptors are projected into the next frame. This operation *implicitly* corresponds to motion-compensating all the pixels in each region. Let $\mathbf{\Phi}_{i,j}(n)$ be the region descriptor for region $R_{i,j}(n)$. Region descriptor projection provides the predicted descriptor $\widetilde{\mathbf{\Phi}}_{i,j}(n+1)$ to which the predicted region $\widetilde{R}_{i,j}(n+1)$ implicitly corresponds. The predicted region is defined as

$$\widetilde{R}_{i,j}(n+1) = \{(x', y', n+1) : (x, y, n) \in R_{i,j}(n), \quad x' = x + \phi^3_{i,j}(n), y' = y + \phi^4_{i,j}(n)\}, \tag{6}$$

where $\left(\phi^3_{i,j}(n), \phi^4_{i,j}(n)\right)$ is the motion vector of $\mathbf{\Phi}_{i,j}(n)$. After the projection, a bottom-up feedback from the region partition refines the topology of the object partition. This feedback generates a tentative correspondence by labeling the object partition Π_o^{n+1} according to the predicted region partition $\widetilde{\Pi}_r^{n+1}$. Once all the pixels in the next object partition are associated to the projected regions, we have a prediction as follows:

$$\widetilde{O}_i(n+1) = \{(x + \phi^3_{i,j}(n), y + \phi^4_{i,j}(n), n+1) : \forall j \in O_i(n), (x, y, n) \in R_{i,j}(n)\}. \tag{7}$$

This procedure is straightforward in case each set of connected pixels in Π_o^{n+1} receives projected region descriptors, and receives them from one object only. In such a case, the foregoing procedure suffices to guarantee the tracking (Figure 1). In reality, multiple simultaneous objects may occlude each other and therefore be included in the same set of connected pixels. The bottom-up interaction is used to improve the object labelling in these cases. The interaction helps to tackle some of the track management issues, such as appearance of new objects in the scene, partial occlusions, and splitting.

A *new object* is detected when a connected set of pixels $S(n+1)$ in Π_o^{n+1} does not get any region descriptor from the projection mechanism. The detection of a new object triggers a track initiation (Eq.(5)).

An *occlusion* takes place when two or more objects interact, either by getting close one to each other, or passing one in front of the other. An occlusion is detected when a connected set of pixels $S(n+1)$ in Π_o^{n+1} receives projected region descriptors from several objects. The object partition validation step separates the objects, that is, provides separate contours for each different object. This refinement is made possible by using the knowledge of the track at the region level, as shown in Fig. 4 for frame $n+2$.

A *splitting* corresponds to the separation of a connected set of pixels in the object partition into two or more subsets. This event is detected when two different disconnected sets of pixels $S_1(n+1)$ and $S_2(n+1)$ in Π_o^{n+1} get region descriptors projected from the same video object.

The predicted partition may not cover all the pixels of Π_o^{n+1}. For the object partition validation step to be complete, each pixel in Π_o^{n+1} has to be classified. If a connected component of Π_o^{n+1} receives region descriptors from one object only, all the unclassified pixels are assigned to that object. If a connected set of Π_o^{n+1} receives region descriptors from several objects, then the unclassified pixels are assigned to the closest projected region. The tentative

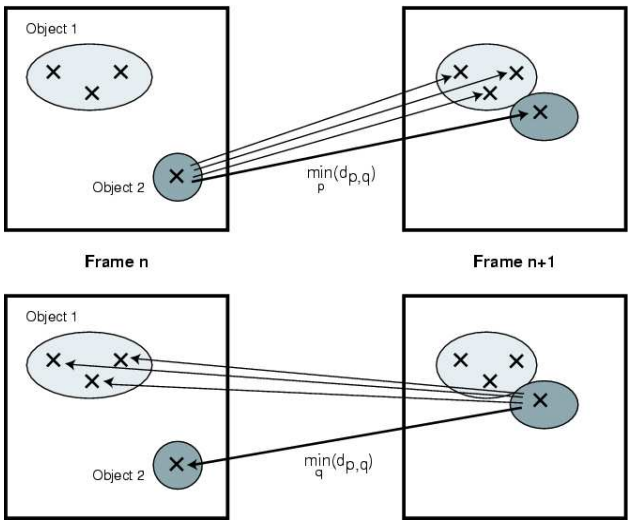


Fig. 5. Data association. In the data association stage, the region decriptors are put in correspondence over time in order to validate their track

correspondence obtained with the above-mentioned procedure is verified through data association in order to define the final correspondence.

B.2 Data association

Data association validates the track of each region descriptor, and as a consequence updates the track of the object partition. This step is particularly important when faced with track management issues.

In the data association stage, the region descriptors are put in correspondence over time. First the predicted region partition $\widetilde{\Pi}_r^{n+1}$ is updated so as to obtain Π_r^{n+1}. Then the region descriptors corresponding to Π_r^{n+1} are compared with those of Π_r^n. Each video object in the predicted partition is separately segmented into homogeneous regions, thus resulting in Π_r^{n+1}.

To verify the correctness of the tentative correspondence obtained with region descriptors projection, we consider the proximity between region descriptors in Π_r^{n+1} and in Π_r^n. The proximity is computed by measuring the distance in the feature space between the region descriptors in frame $n+1$ and those in frame n (Figure 5). These distances are then compared with the results of the projection and a decision step establishes the final correspondence.

To reduce the dimensionality of the problem, a gating process is introduced prior to the distance computation. The gating process allows us to preselect the candidate for data association by eliminating the couples of region descriptors that are highly unlikely to be temporally related. This preselection is based on a distance criterion that considers the maximum allowable displacement of a region descriptor between two frames. The value of the gating parameter can be set according to the application at hand. In the specific implementaton, the value of the gating parameter is half the vertical and horizontal image dimension for the vertical and horizontal displacement, respectively. The use of the gating process results in a lower complexity and favors stability.

After the gating process, a pair-wise distance metric is applied to all the remaining region descriptors. The region descriptors include information from different sources that are encoded with varying number of features. For example, three features are used for color, and two for motion. We refer to such groups of similar features as feature categories. To avoid masking important information when computing the distance, we use separate distance measures, $\mathcal{D}_f(\cdot)$, for each feature category. Since the results of the separate proximity measures will be fused together, it is desirable that $\mathcal{D}_f(\cdot)$ returns a normalized result, especially in the case of poorly scaled or highly correlated features. For this reason we choose the Mahalanobis metric. To compute the proximity of two region descriptors, the Mahalanobis distance can be expressed as

$$\mathcal{D}_f\big(\mathbf{\Phi}_{i,j}(n), \mathbf{\Phi}_{k,l}(n+1)\big) = \sqrt{\sum_{s=1}^{K} \frac{\big(\Phi_{i,j}(n)^s - \Phi_{k,l}(n+1)^s\big)^2}{\sigma_s^2}} \quad (8)$$

where σ_s^2 is the variance of the s^{th} feature over the entire feature space and K is the number of features. The complete point-to-point similarity measure between $\mathbf{\Phi}_{i,j}(n)$ and $\mathbf{\Phi}_{k,l}(n+1)$ is obtained by fusing the distances computed within each category

$$\mathcal{D}\big(\mathbf{\Phi}_{i,j}(n), \mathbf{\Phi}_{k,l}(n+1)\big) = \frac{1}{F}\sum_{f=1}^{F} w_f \mathcal{D}_f\big(\mathbf{\Phi}_{i,j}(n)^s, \mathbf{\Phi}_{k,l}(n+1)^s\big), \quad (9)$$

where F is the number of feature categories and w_f the weight which accounts for the reliability of each feature category. The value of F may change from frame to frame and from cluster to cluster. The value of the reliability is $w_f = 0$ for those features that have similar values in adjacent regions and $w_f = 1$ otherwise. The use of the reliability parameter facilitates the data association process by eliminating undiscriminant features from the computation of the distance.

The result of the distance computation can be represented as a matrix $\mathbf{D} = \{d_{p,q}\}$, where each row, p, corresponds to a region descriptor in frame $n+1$, and each column, q, corresponds to a region descriptor in frame n. We refer to this matrix as distance matrix. Each element of the distance matrix represents the distance between two region descriptors. The smallest element for each row and for each column identifies a possible correspondence between two region descriptors. This result is compared with that of the tentative correspondence to check if there is a conflict. A tentative correspondence between the $\bar{p}^{\text{th}}$ region descriptor in frame $n+1$ and the $\bar{q}^{\text{th}}$ region descriptor in frame n is confirmed if

$$d_{\bar{p},\bar{q}} = \min_q(d_{p,q}) = \min_p(d_{p,q}) \quad (10)$$

If the condition in Eq.(10) is respected, the track is updated. Otherwise, the final correspondence between region descriptors that do not satisfy Eq.(10) is obtained by means

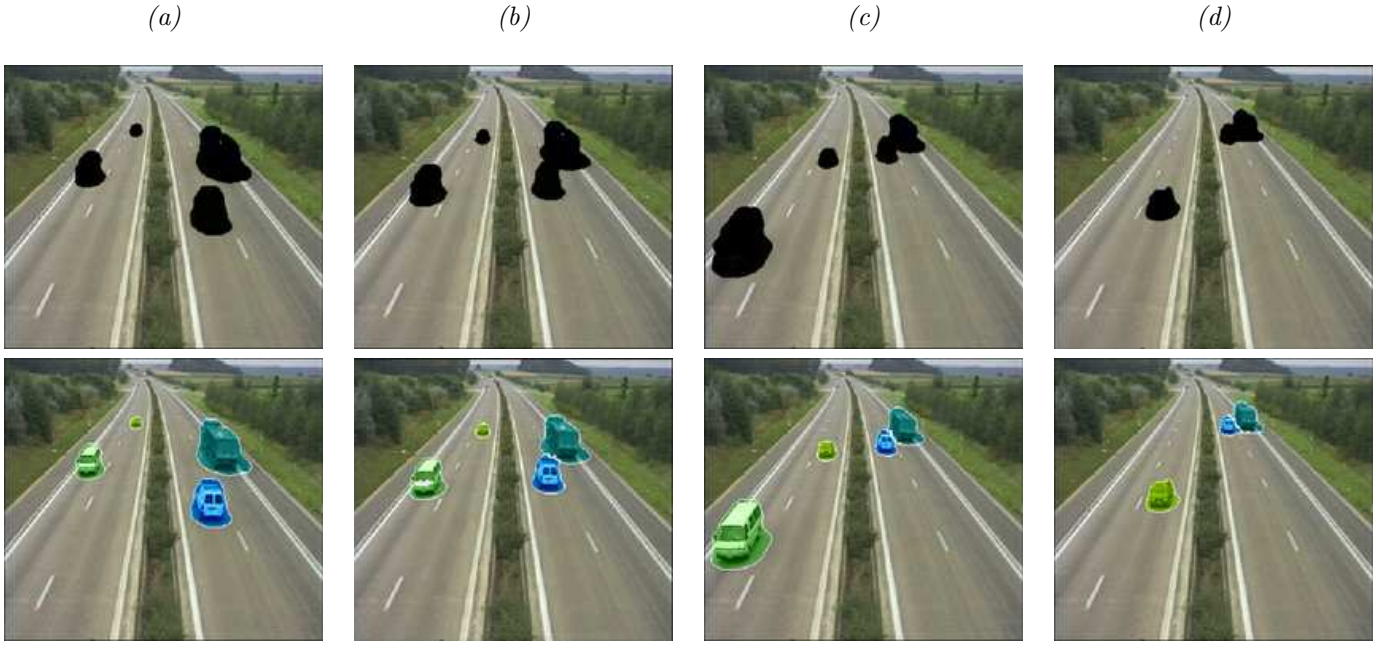


Fig. 6. Tracking results for the sequence *Highway*. Top: object segmentation results. Bottom: video objects tracked over time

of an iterative process. During this process the best point-to-point pairs are selected first. Then, the remaining ones are iteratively paired to obtain the final correspondence. This final correspondence is then exploited in the bottom-up feedback to update the object partition.

B.3 Discussion

To conclude this section, it is interesting to enumerate the advantages of the proposed approach with respect to the alternative approach of computing two separate region partitions in the current and next frames, and then pairing the region descriptors, without considering any projection. The proposed method based on projection has the following advantages:

- it is computed with data that are already available
- it is a simple operation
- it provides an additional element to the final decision for the correspondence
- it provides an educated initialization for the region partition algorithm in the next frame.

Region descriptor projection provides an estimate of the region position in the next frame, given the actual region partition and the motion information. The refinement of the predicted region partition adapts the projected partition to the current frame through spatio-temporal segmentation.

III. Experimental results

This section discusses the results of the proposed algorithm applied on real image sequences in order to track multiple objects. With reference to Figure 1, the input of the algorithm is a video sequence and the output is a set of video objects that are coherently labeled over time.

We would like to highlight here that the same set of parameters was used to generate all the results presented in this section.

The display of the results is organized as follows. First, the results of object segmentation are shown. Object segmentation defines the shape of the moving objects. The computed shape is represented as a mask (color coded in black) superimposed on the original background. Then the results of object tracking are displayed. Each object is given a label by the tracking algorithm. Each label is coded with a different color for displaying purposes. Figure 6 shows object segmentation and object tracking results from sample frames of the test sequence *Highway*, from the MPEG-7 Video Content Set. This traffic surveillance sequence represents a highway with vehicles of different sizes driving on four lanes. Here, the goal of tracking is to manage multiple simultaneous objects, their mergings, and their appearance and disappearance from the scene. Column (b) shows that the two objects on the right hand side are merged together in the object segmentation mask (top). The tracking algorithm is capable of separating the two objects (bottom) and of providing them with a coherent label over time. Column (d) shows the status of the tracked vehicles after the van on the left hand side has left the scene. The disappearance of the object does not alter the tracking performance. In the same way, all the other objects in the scene are separately tracked along the frames as shown in Figure 7. On the left a sample frame from the sequence is displayed, and on the right, the corresponding trajectories of the video objects in the frames from 110 to 160 are shown. The information of the track of each object can be exploited in the framework of advanced video surveillance. The results of image analysis (segmented video objects and their associated trajectories) can be used by a content understanding step that monitors

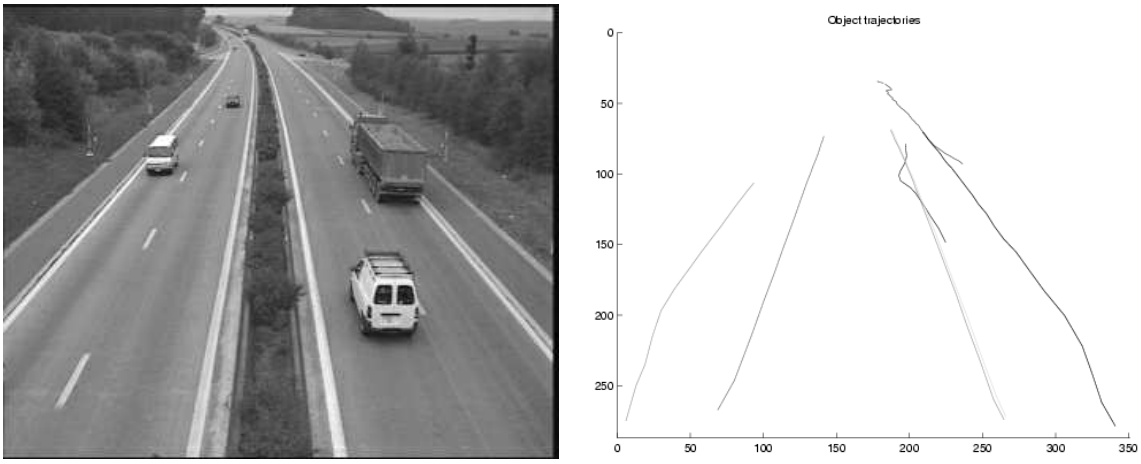


Fig. 7. Trajectories of video objects. Left: Sample frame from test sequence *Highway* (frame 110) with 4 video objects (vehicles). Right: Trajectories of the 6 video objects that appear in the sequence from frame 110 to frame 160. The horizontal and vertical axes of the graphs represent the width and the height of the frame, respectively

the behavior of objects in the scene. This information helps the content understanding module in describing events in the scene and in generating alarms in case of dangerous situations. By using a calibrated camera, these results could be complemented so as to provide the trajectories in the 3D scene. Furthermore, the knowledge of the path of each object within a video sequence enables interactive applications such as video-based hyperlinking, video editing, and object-based indexing.

Tracking results from sample frames of the sequence *Hall Monitor*, from the MPEG-4 Video Content Set, are shown in Figure 8. As opposed to the previous sequence, *Hall Monitor* represents an indoor scene with deformable objects. The goal of tracking is to follow the two moving people separately. In this sequence we want to highlight the behaviour of the tracking algorithm in case of errors in object segmentation and in case of track management issues such as splitting. It is possible to notice in column (b) that the man is casting his shadow on the wall. Since no descriptors are projected in the object partition corresponding to the shadow, a new track is initiated. The shadow is therefore correctly identified as a new object by the tracking algorithm. The appearance of a new object, the shadow, does not alter the tracking of the man on the left hand side. When the shadow and the man merge in the object segmentation mask (column (c), top), the two objects are kept separated thanks to tracking (column (c), bottom). This allows to overcome the problem introduced by the object segmentation module which wrongly detected the shadow as an object. A further analysis module could be added to the system in order to identify the shadows. Finally, column (d) shows the splitting of the man on the left hand side and his suitcase. The suitcase correctly keeps the same label as the man and it is not interpreted as a new object even if they are identified by two unconnected object partitions.

Figure 9 shows sample frames of the test sequence *Surveillance*, from the MPEG-7 Video Content Set. The difficulties of this sequence are the presence of simultaneous non-rigid objects and merging. It is important to notice that even if the segmentation mask does not separate the two people (columns (b) to (d), top), the tracking algorithm keeps their identity (columns (b) to (d), bottom). However, when the man on the right hand side is completely covered by that on the left hand side, his trajectory is lost (total occlusion). To relate the man reappearing after total occlusion to a new trajectory, the data association step should operate not only between subsequent frames, but also on a longer temporal window.

Finally, we would like to further analyse the behaviour of the proposed tracking algorithm in case of errors in the object segmentation results. Figure 10 shows a zoom from the sequence *Surveillance*. The segmentation mask (top) does not define the shape of the person correctly. In particular (column (b) and (c)), a leg of the man is identified by a set of pixels which is not connected to the rest of the body. Instead of initiating a new track for the unconnected part, the projection of the region description allows one to keep the track of the full object, thus recovering the identity when the segmentation is correct (column (d)). The interaction between the region partition and the object partition helps in overcoming this problem and the objects are correctly tracked.

IV. Conclusions

We presented an automatic tracking algorithm based on interactions between video objects and their regions. Regions are objects' areas that are homogeneous with respect to a set of features such as motion, color, and texture. Regions have been represented by their region descriptors. Each region descriptor is tracked over time as representative of the corresponding video object.

The proposed algorithm is capable of dealing with multiple simultaneous objects. Track management issues such as appearance and disappearance of objects, splitting and partial occlusions are resolved through interactions between

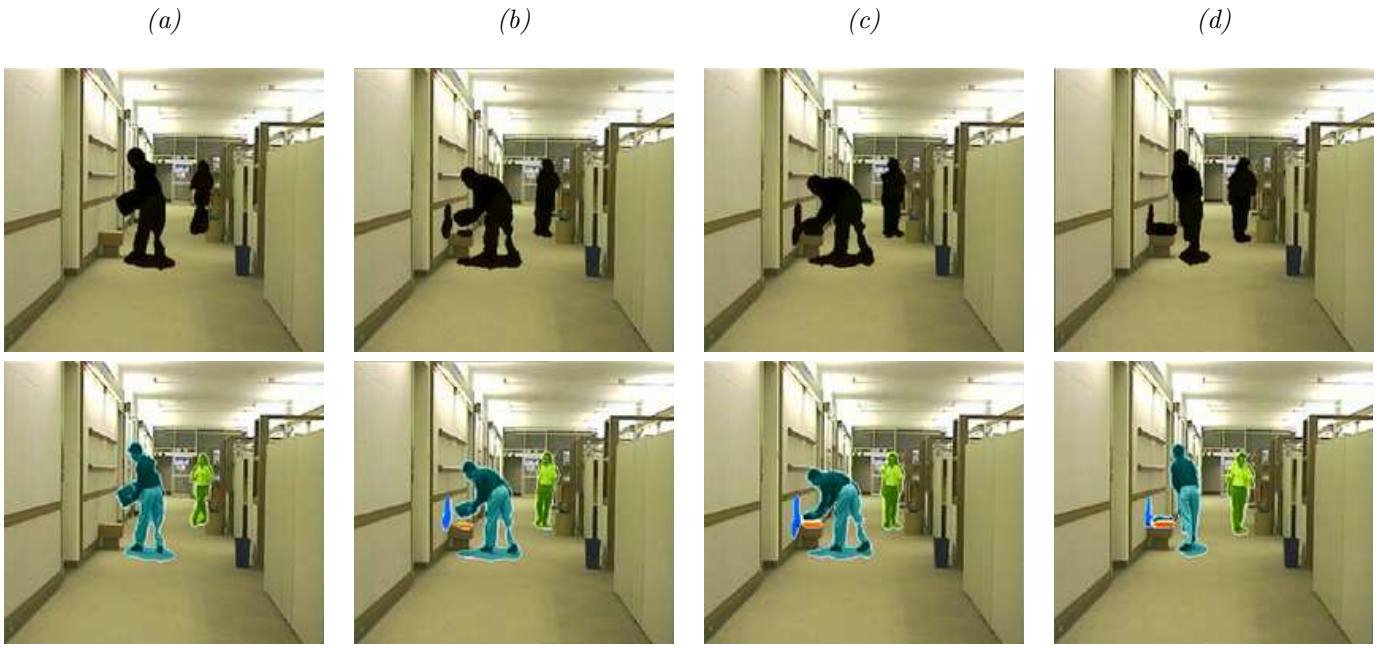


Fig. 8. Tracking results for the sequence *Hall Monitor*. Top: object segmentation results. Bottom: video objects tracked over time

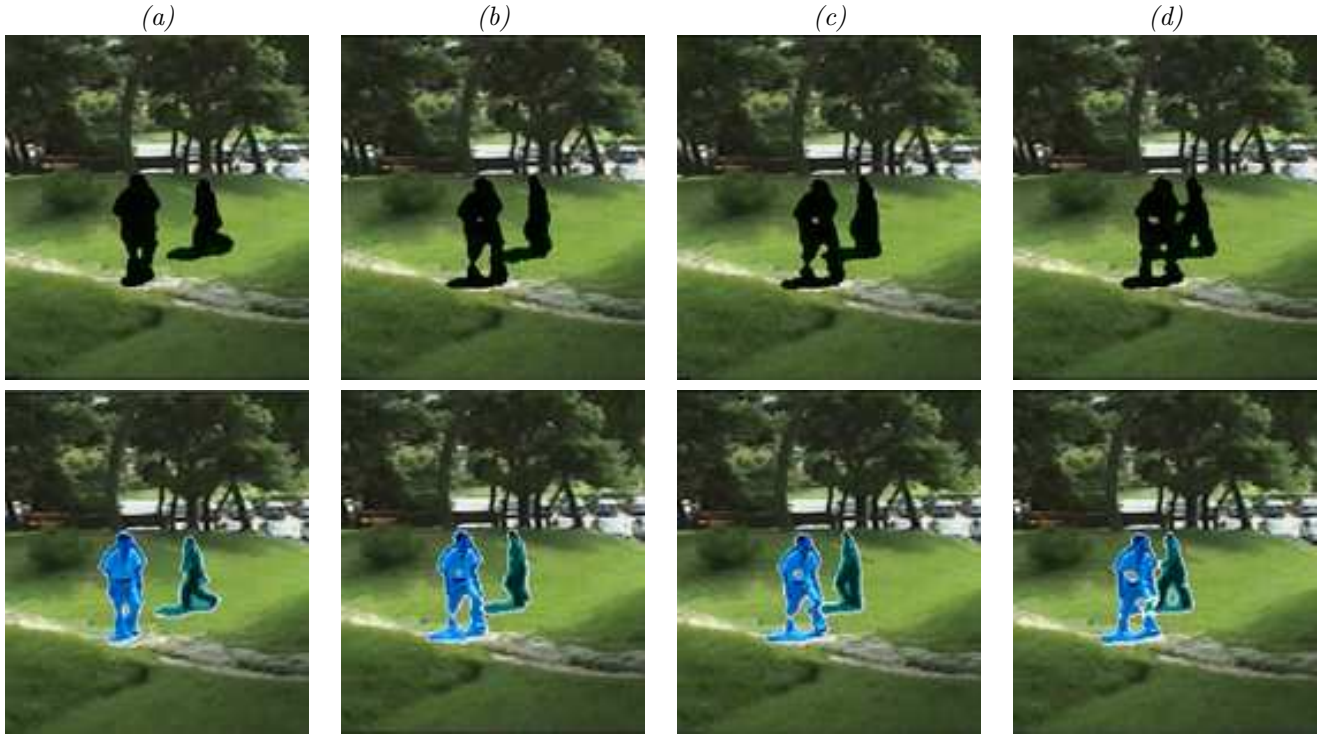


Fig. 9. Tracking results for the sequence *Surveillance*. Top: object segmentation results. Bottom: video objects tracked over time

regions and objects. Defining the tracking based on the parts of objects, identified by region segmentation, has led to a flexible technique that exploits the nature of the video object tracking problem.

The proposed technique has been demonstrated on indoor and outdoor scenes with both rigid and deformable objects without changing the parameter setting. The versatility of the tracking mechanism makes the proposed algorithm a suitable component technology for multimedia systems aiming at object-based indexing, video-based hyperlinking, visual surveillance, and video manipulation. The mechanism can also be exploited by including a priori information from a specific application.

All the components of the algorithm can run in real-time on a standard PC with the exception of the region segmentation stage. The region segmentation stage is based on

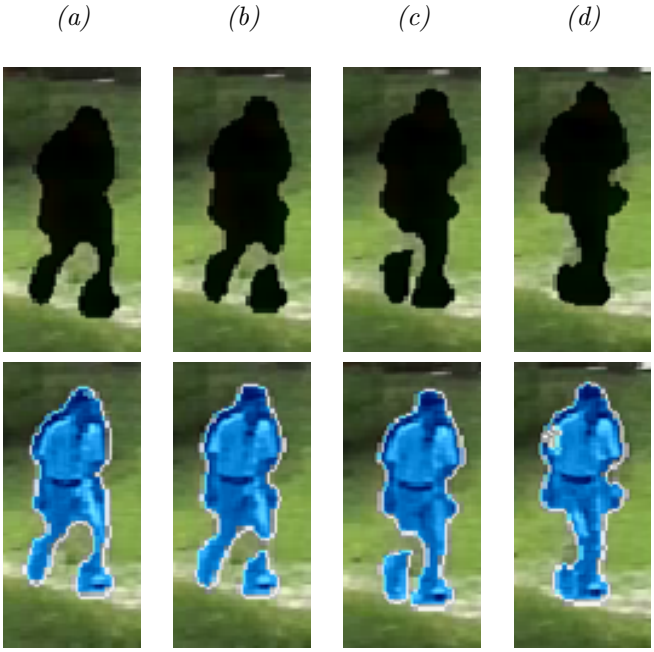


Fig. 10. Example of robustness of the proposed tracking algorithm in case of errors in the object segmentation module

an iterative process in order to produce an accurate region partition. We are currently investigating simpler region segmentation algorithms in order to find the trade-off between computational complexity and quality of the region partition. Our current work is also concentrating on overcoming two limitations of the tracking mechanism, namely initialisation of a track when groups of objects enter the scene and total occlusions. When a group of objects enters the scene, then the initialisation process assigns to them a unique label. To cope with this problem, future work should concentrate on how to associate additional semantic concepts with object segementation. This might involve the use of domain knowledge and image understanding. For instance, if the algorithm learns from the video the typical size of an object and the layout of the scene (or this information is provided as a priori knowledge), then it could separate the single objects from the group already during the initialisation process. Finally, we aim at including in the proposed scheme a ghost state mechanism to cope with total occlusions. To this end, the data association step should operate not only between subsequent frames, but also on a longer temporal window. This solution would also allow one to track objects that leave and reenter the scene.

Acknowledgments

The authors would like to thank the anonymous reviewers for their comments, which improved the content of the paper.

References

[1] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003.

[2] D. Koller, K. Danilidis, and H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*, 10(3):257–281, 1993.

[3] T. Meier and K. Ngan. Automatic segmentation of moving objects for video object plane generation. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):525–538, 1998.

[4] D. Wang. Unsupervised video segmentation based on watersheds and temporal tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):539–546, 1998.

[5] B. Marcotegui, F. Zanoguera, P. Correia, R. Rosa, F. Marques, R. Mech, and M. Wollborn. A video object generation tool allowing friendly user interaction. In *Proceedings of International Conference on Image Processing*, pages 391–395, 1999.

[6] H. Tao, H.S. Sawhney, and R. Kumar. Object tracking with Bayesian Estimation of dynamic layer representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):75–89, 2002.

[7] C. Gu and M.-C. Lee. Semiautomatic segmentation and tracking of semantic video objects. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):572–584, 1998.

[8] N. Paragios and R. Deriche. Geodesic active regions for motion estimation and tracking. In *Proceedings of 7th International Conference on Computer Vision (ICCV)*, 1999.

[9] N. Peterfreund. Robust tracking of position and velocity with Kalman snakes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(6):564–569, 1998.

[10] S. Sun, D.R. Haynor, and Y. Kim. Semiautomatic video object segmentation using VSnakes. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(1):75–82, 2003.

[11] B. Gnsel, A. M. Tekalp, and P. J. van Beek. Content-based access to video objects: Temporal segmentation, visual summarization, and feature extraction. *Signal Processing*, 66(2):261–280, 1998.

[12] J.W. Zhao, P. Wang, and C.Q Liu. An object tracking algorithm based on occlusion mesh model. In *Proceedings of International Conference on Machine Learning and Cybernetics*, pages 288–292, 2002.

[13] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik. A real-time computer vision system for measuring traffic parameters. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 495–501, 1997.

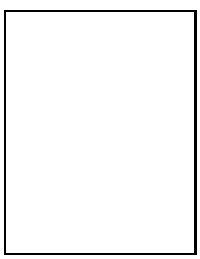
[14] F. Marques and J. Llach. Tracking of generic objects for video object generation. In *Proceedings of International Conference on Image Processing*, pages 628–632, 1998

[15] Y. Tsaig and A. Averbuch. Automatic segmentation of moving objects in video sequences: a region labeling approach. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(7):597–612, 2002.

[16] A. Cavallaro and T. Ebrahimi. Video object extraction based on adaptive background and statistical change detection. In *Proceedings of SPIE Electronic Imaging - Visual Communications and Image Processing*, pages 465–475, San Jose, California, USA, 2001.

[17] A. Cavallaro, F. Ziliani, R. Castagno, and T. Ebrahimi. Vehicle extraction based on focus of attention, multi feature segmentation and tracking. In *Proceedings of X European Signal Processing Conference (EUSIPCO)*, pages 2161–2164, Tampere, Finland, 2000.

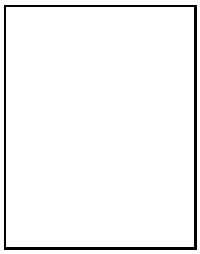
[18] R. Castagno, T. Ebrahimi, and M. Kunt, "Video segmentation based on multiple features for interactive multimedia applications," *IEEE Transactions on Circuits and System for Video Technology*, vol. 8, pp. 562–571, September 1998.



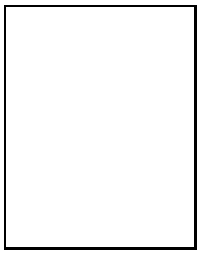
Andrea Cavallaro received his M.Sc. (Honors) in Electrical Engineering from the University of Trieste, Italy, in 1996, and his Ph.D. in Electrical Engineering from the Swiss Federal Institute of Technology, Lausanne, Switzerland, in 2002. In 1996 and 1998, he served as a research consultant at the Image Processing Laboratory, University of Trieste, Italy, working on compression algorithms for very low bitrate video coding and on digital image sequence de-interlacing. In 1997 he served the Italian Army as lieutenant at the 33rd Electronic Warfare Battalion in Treviso, Italy. From 1998 to 2003 he was a research assistant at

the Signal Processing Laboratory of the Swiss Federal Institute of Technology (EPFL). He was workpackage leader for the EU projects ACTS Modest and IST art.live. Since 2003, he is lecturer at the Department of Electronic Engineering, Queen Mary, University of London (QMUL).

His main research interests are image and video analysis, video compression and visual information description. Dr. Cavallaro was a member of the organizing committee of the 2002 IEEE International Conference on Multimedia and Expo, member of the Scientific Committee of the 2003 SPIE VCIP conference, and is member of the Technical Committee of the 2004 IEEE ICME. He served as session chair at several conferences and organized the special session on object-based video at the 2003 Visual Communication and Image Processing Conference. He acts as reviewer for several leading international conferences and journals, and he is author of more than 25 papers, including 3 book chapters.



Olivier Steiger received the M.S. degree in electrical engineering from the Swiss Federal Institute of Technology (EPFL) in Lausanne, Switzerland, in 2001. In 2001, he became a research assistant and Ph.D. student at the Signal Processing Institute of EPFL. Currently his research interests are focused on video description, video object tracking and semantic video transcoding for applications such as Universal Multimedia Access and Mixed Realities. Olivier Steiger has been an academic visitor at Queen Mary, University of London (UK), in 2003.



Touradj Ebrahimi received his M.Sc. and Ph.D., both in Electrical Engineering, from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, in 1989 and 1992 respectively. In 1993, he was a research engineer at the Corporate Research Laboratories of Sony Corporation in Tokyo, where he conducted research on advanced video compression techniques for storage applications. In 1994, he served as a research consultant at AT&T Bell Laboratories working on very low bitrate video coding. He is currently Professor at EPFL, where besides teaching, he is involved in various aspects of visual information processing and coding for multimedia applications. He has been the recipient of various distinctions such as the IEEE and Swiss national ASE award, the SNF-PROFILE grant for advanced researchers, Two ISO-Certificates for key contributions to MPEG-4 and JPEG 2000, and the best paper award of IEEE Trans. on Consumer Electronics . He became a Fellow of the international society for optical engineering (SPIE) in 2003, for outstanding contrubutions in the field of visual information processing and coding. Prof. Ebrahimi has initiated more than two dozen National, European and International cooperation projects with leading companies and research institutes around the world. He is also the head of the Swiss delegation to MPEG, JPEG and SC29, and acts as the Chairman of Advisory Group on Management in SC29. In 2002, he founded Emitall SA, an R&D and consulting company in the area of Electronic Media Innovations. He is or has been associate Editor with various IEEE, SPIE, and EURASIP Journals. Prof. Ebrahimi is a member of Scientific Advisory Board of various start-up and established companies in the general field of Information Technology. He has served as Scientific Expert and Evaluator in various Research Funding Agencies such as those of European Commission, The Greek Ministry of Development, The Austrian National Foundation for Scientific Research, and a number of Venture Capital Companies active in the field of Information Technologies and Communication Systems. His research interests include still, moving, and 3D image processing and coding, visual information security (rights protection, watermarking, authentication, data integrity, steganography), new media, and human computer interfaces (smart vision, brain computer interface). He is the author or the co-author of more than 100 research publications, and holds 10 patents. Prof. Ebrahimi is a member of IEEE, SPIE, ACM and IS&T.