# Multiple video object tracking in complex scenes

Proceedings of ACM International Conference on Multimedia, pp. 523-532, Juan Les Pins, France, December 1-6, 2002.

Andrea Cavallaro
Signal Processing Institute
Swiss Federal Institute of Technology (EPFL)
CH-1015 Lausanne, Switzerland
andrea.cavallaro@epfl.ch

Olivier Steiger
Signal Processing Institute
Swiss Federal Institute of Technology (EPFL)
CH-1015 Lausanne, Switzerland
olivier.steiger@epfl.ch

Touradj Ebrahimi
Signal Processing Institute
Swiss Federal Institute of Technology (EPFL)
CH-1015 Lausanne, Switzerland
touradj.ebrahimi@epfl.ch

## ABSTRACT

We present an automatic video object tracking algorithm capable of dealing with multiple simultaneous objects. The tracking is based on interactions between high-level and low-level image analysis results. The high-level result is a partition defining video objects, and the low-level result is a partition formed by homogeneous regions. For each region, a set of characteristic descriptors is produced. These region descriptors, and not regions themselves, are used to track the regions (and thus the objects) along time. Track management issues such as appearance and disappearance of objects, splitting and partial occlusions are resolved through interactions between 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. Experimental results show that the proposed method is able to track multiple rigid and deformable objects in indoor and outdoor scenes.

## 1. INTRODUCTION

Object-based coding and description offer a new range of capabilities, where the objects are separately coded and described. Industry standards, such as MPEG-4 and MPEG-7, provide the user with flexibility in content-based access and manipulation of multimedia data. To maximize the benefits of object-based representation, these standards need to be complemented with automatic techniques for extracting the video objects from video data, a problem that still remains largely unsolved.

An object-based representation requires therefore prior decomposition of sequences into semantically meaningful, physical objects. In a way, this corresponds to retracing the steps in the video-creation process. This complex problem can be formulated as one of identifying objects in the scene and separating them from the background. Tracking is a fundamental step in video object extraction. In this framework, the goal of tracking is to follow video objects in the scene and to update 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.

A tracking algorithm 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. A survey of different methods for video object tracking is presented in Section 2. Here, the capabilities and the limits of the different approaches are discussed. Based on the above mentioned discussion, Section 3 introduces the proposed tracking algorithm. The algorithm exploits an image representation as partition hierarchy and tracks video objects based on interactions between different levels of the hierarchy. The hierarchy is composed of a semantic level and a region level. The semantic level defines the topology of the video objects. The region level defines the topology of homogeneous areas constituting the objects. The technique for tracking the region partition is described in Section 4. Region tracking is then used in the interaction with the semantic partition in the joint region-semantic tracking algorithm, as described in Section 5. Finally, tracking results are discussed in Section 6, and Section 7 concludes the paper.

## 2. STATE OF THE ART

In this section we review the techniques proposed in the literature for tracking video objects. The different tracking methods can be classified into four groups: *region-based*, *contour-* and *mesh-based*, *model-based*, and *feature-based*.

### 2.1 Region-based tracking

Region-based methods track connected regions that roughly correspond to the 2D shapes of video objects. The tracking strategy relies on information provided by the entire region [8, 9, 12]. Examples of such information are motion, color, and texture. In [12] temporal tracking is realized in

four steps: motion projection, marker extraction, clustering, and region merging. First, moving objects in one frame are projected into the next frame based on a simplified linear motion model. This step establishes the correspondence between moving objects in the two frames. Since motion projection alone does not produce an accurate and complete segmentation, reliable parts of each projected object are extracted as markers of the corresponding moving objects. This extraction is based on assumptions about the relationship between the projected and the real object. Morphological erosion and pixel difference thresholding are used to this end. A modified watershed transformation, followed by region merging, yields a complete segmentation of the next frame. The spatial segmentation produces regions of homogeneous intensity. The method is able to track fast moving objects and to deal with the appearance of new objects and the disappearance of existing objects. On the other hand, the technique cannot cope with complex motion or deformation. In addition, the tracking strategy is based on several thresholds which need to be tuned manually.

A projection of the texture partition obtained through a spatial homogeneity criterion is proposed in [8]. An over-segmented partition is first obtained through color clustering. The texture partition is then adapted to the information of the next frame. The adaptation is performed by means of a fitting process between the motion compensated markers and the regions of the over-segmented partition.

In [9] an object tracking technique based on a generalized Hausdorff distance is proposed. A binary model for the video object is first derived from the edge image. The method then matches the model to objects in subsequent frames in the sequence. The goal is to update the model at every frame to follow changes in the shape of the object. This results in a sequence of binary models that guide the extraction of the video object. The method assumes that the background is stationary. In addition, this technique assumes that only one moving object appears in the scene. This is a severe limitation of this technique, as it is not able to cope with complex scenes, cluttered backgrounds and multiple moving objects.

## 2.2 Contour-based and mesh-based tracking

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. The tracking strategy proposed in [5] first estimates the parameters of a perspective motion model and then predicts the position of the contour in the next frame based on these parameters. The obtained prediction accounts for near planar rigid body motion. To deal with non-rigid body motion, the method adjusts the approximated boundary by means of a morphological watershed. The method has the advantage of being able to adapt the tracking to the video object extraction framework (boundaries with pixel-wise accuracy), but it suffers from some drawbacks. 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) [7, 10, 11], or meshes [4]. Active contour models rely on the information provided by the object boundaries [7, 10]. 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 [11]. 2D meshes have also been used to track video objects [4]. Objects are represented by 2D triangular mesh models, which are computed from the content of the image sequence. The 2D mesh forms a tessellation of the 2D shape of the video object into triangular patches. The vertices of these patches are called nodes, and are non-uniformly distributed. The initial mesh is designed so that the nodes along the boundary approximate the true object boundary. The remaining topology of the nodes is determined so that the mesh structure coincides with the spatial structure in the scene. To this end, the nodes are located along salient intensity features. Next, node motion is estimated, and motion compensation is used to warp each triangular patch according to an affine transformation. Finally, the mesh is adaptively refined to account for appearance or occlusion of object parts. 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.

## 2.3 Feature-based tracking

Instead of tracking the entire object, features of a video object can also be used 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 [1]. 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 determine which of them belong to the same object is a major drawback of these approaches.

## 2.4 Model-based tracking

The definition of parameterized object models makes it possible to exploit the a *priori* knowledge of the shape of typical objects in a given scene. 3D object models can be used 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. For example, the method in [6] is aimed at highway traffic surveillance, thus the models in the database correspond to vehicles, and only these can be detected. The detection of people and animals could be important in identifying dangerous situations, but they will be overlooked with such a method. In general, model-based tracking methods are not suitable for generic video object segmentation.

## 3. PROPOSED METHOD

By considering the video object tracking methods reviewed in Section 2, the proposed approach is designed as a hybrid between the region-based and the feature-based techniques. It exploits the advantages of the two by considering first the object as an entity and then by tracking its parts. For each frame $n$, objects are defined by a semantic partition, $\Pi_s^n$, whereas objects' parts are defined by a region partition, $\Pi_r^n$. The tracking mechanism is based on feedbacks between the semantic and the region partitions. These interactions allows the tracking to cope with multiple simultaneous objects, motion of non-rigid objects, partial occlusions, and appearance and disappearance of objects. The block diagram of the proposed approach is depicted in Figure 1. 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. Once the semantic partition is available for an image, it is automatically extended to the following image. No restriction on the way they are extracted is imposed. In the practical implementation we use the video object extraction method presented in [2], which produces the semantic partition. The semantic partition identifies the objects from the background and provides a mask defining the areas of the image containing the moving objects. Only the areas of belonging to the semantic partition are considered by the following step, which takes into account the spatio-temporal properties of the pixels in the changed areas and extracts spatio-temporal homogeneous regions. Each object is processed separately and is decomposed into a set of non-overlapping regions. The regions are defined by the clustering described in [3], thus producing the region partition as shown in Figure 2.

Given the semantic partition in the new frame and the region partition in the current frame, the proposed tracking procedure performs two different tasks. First, it defines a correspondence between the semantic partition in the current frame $n$ and the semantic partition in the new frame $n+1$. Second, 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 details of the method are presented in the following sections. First the computation and the tracking of the region partition is described, and then the interactions between region and semantic partitions to obtain video object tracking are commented.

## 4. TRACKING THE REGION PARTITION

In a dynamic scene, the topologies of the homogeneous regions vary over time. Corresponding regions at different time instances have therefore to be linked together. This temporal linkage is achieved through 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.

### 4.1 Region descriptor projection

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, $\Phi_i(n)$. 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

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

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

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

Region description projection can be represented in compact form as

$$\widetilde{\Phi}_i(n+1) = \mathbf{A}\Phi_i(n) \quad (4)$$

where

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & & & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \quad (5)$$

is the $K_i(n) \times K_i(n)$ transformation matrix. The result of region descriptor projection is a prediction (estimate) of the region partition $\widetilde{\Pi}_r^{n+1}$ in the next frame.

### 4.2 Refinement of the predicted region partition

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, $\widetilde{\Phi}_i(n+1)$, provide an effective initialization

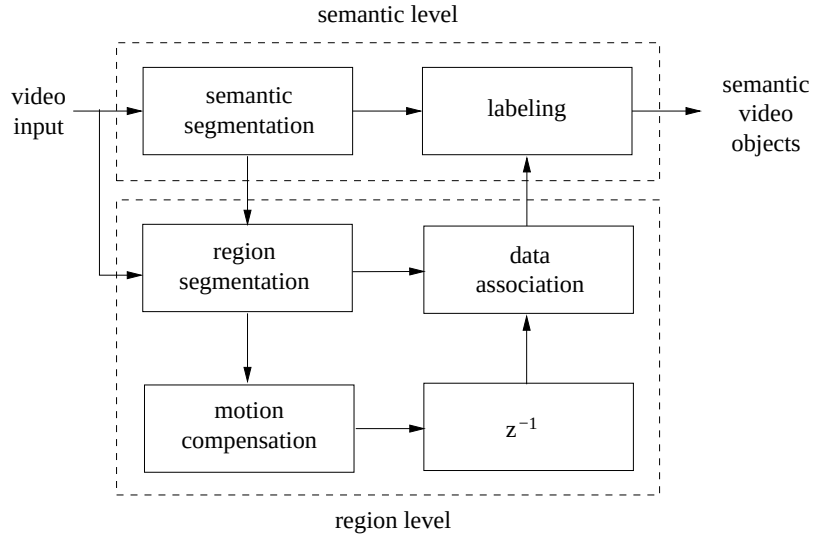


**Figure 1: Flow diagram of the proposed tracking mechanism based on interactions between the semantic and the region partitions. These interactions help the tracking process to cope with multiple simultaneous objects, partial occlusions, as well as appearance and disappearance of objects**

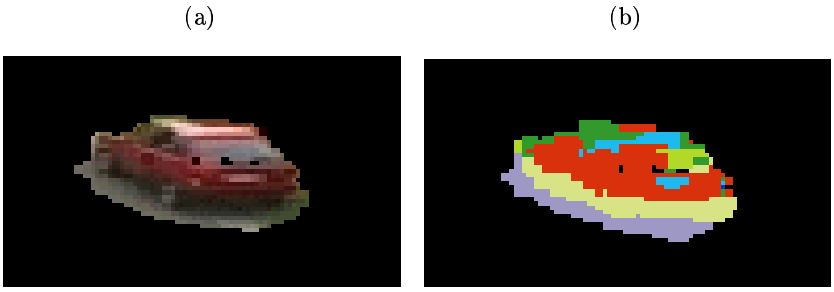


**Figure 2: Example of region segmentation driven by the results of semantic segmentation: (a) area of interest defined by the semantic segmentation and (b) regions defined by the feature-based segmentation**

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 $\Pi_r^{n+1}$ is obtained through the clustering process described in [3]. An updated region descriptor, $\Phi_i(n+1)$, defined as

$$\Phi_i(n+1) = \left(\phi_i^1(n+1), \phi_i^2(n+1), \phi_i^3(n+1), \phi_i^4(n+1), \ldots \right. \quad (6)$$
$$\left. \ldots, \phi_i^{K_i(n+1)}(n+1)\right)^T$$

is finally associated to each region. The block diagram describing the two-step region tracking strategy is shown in Figure 3.

## 5. JOINT REGION-SEMANTIC PARTITION TRACKING

Tracking the semantic partition is a difficult task in the case of multiple simultaneous objects, particularly when projections of the physical objects overlap in the image plane. To overcome this problem, the temporal evolution of the semantic 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 semantic partitions, thus resulting in a multi-level tracking algorithm.

The joint region-semantic tracking mechanism is organized in two major steps: semantic partition validation, and data association. The *semantic partition validation* step is a feedback from the region partition level to the semantic 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.

### 5.1 Semantic partition validation

The semantic partition validation step initializes the tracking process and improves the accuracy of the semantic partition in case the projections of 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, operating as follows.

#### *5.1.1 Top-down*

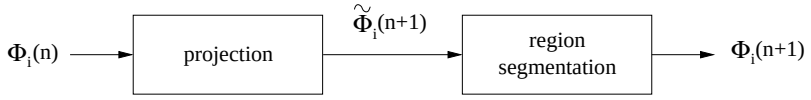


**Figure 3: The two steps of the region tracking strategy. First the region descriptors are projected into the new frame. Then their values are updated through region segmentation**

Before initializing the tracking procedure, each video object is decomposed into a set of non-overlapping regions. Each region $j$ is characterized by its region descriptor $\Phi_j(n)$. To initialize the tracking procedure, each region descriptor $\Phi_j(n)$ is associated to the corresponding semantic object, $i$. After this association, the region descriptor is denoted with $\Phi_{i,j}(n)$. This operation, referred to as *track initiation*, can be expressed as

$$\forall\, O_i(n) \quad i = 1, \ldots, N_F^n \quad \exists\, \Phi_{i,j}(n) \quad j = 1, \ldots, N_{R_i}^n, \quad (7)$$

with $N_F^n$ number of video objects in frame $n$, and $N_{R_i}^n$ number of regions for object $i$. This initialization takes place at the beginning of the tracking process and every time a new video object appears.

### 5.1.2 Bottom-up

After the initialization, the region descriptors are projected into the next frame using the procedure described in Section 4.1. This operation implicitly corresponds to motion-compensating all the pixels in each region. Let $\Phi_{i,j}(n)$ be the region descriptor for region $R_{i,j}(n)$. Region descriptor projection provides the predicted descriptor $\widetilde{\Phi}_{i,j}(n+1)$ to which the predicted region $\widetilde{R}_{i,j}(n+1)$ implicitly corresponds. The predicted region is defined as

$$\widetilde{R}_{i,j}(n+1) = \{(x', y', n+1) : (x, y, n) \in R_{i,j}(n), \quad (8)$$
$$x' = x + \phi_{i,j}^3(n), y' = y + \phi_{i,j}^4(n)\},$$

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

$$\widetilde{O_i}(n+1) = \bigcup_{j=1}^{N_{R_i}^n} \widetilde{R}_{i,j}(n+1), \quad (9)$$

that, alternatively, can be represented as

$$\widetilde{O}_i(n+1) = \{(x + \phi_{i,j}^3(n), y + \phi_{i,j}^4(n), n+1) : \quad (10)$$
$$\forall j \in O_i(n), (x, y, n) \in R_{i,j}(n)\}.$$

This procedure is straightforward in case each set of connected pixels in $\Pi_f^{n+1}$ receives projected region descriptors, and receives them from one object only. In such a case, the foregoing procedure suffices to guarantee the tracking. 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 semantic partition 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*. In the following we discuss how these events are detected, and what actions are taken to cope with them.

- *New object.* A new object is detected when a connected set of pixels $S(n+1)$ in $\Pi_f^{n+1}$ does not get any region descriptor from the projection mechanism. The detection of a new object triggers a *track initiation* (Eq.(7)).
- *Occlusion.* 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 $\Pi_f^{n+1}$ receives projected region descriptors from several objects. The semantic 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.
- *Splitting.* A splitting corresponds to the separation of a connected set of pixels in the semantic 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 $\Pi_f^{n+1}$ get region descriptors projected from the same video object.

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

## 5.2 Data association

The *tentative correspondence* obtained with the semantic partition validation step is verified through data association in order to define the *final correspondence*. Data association validates the track of each region descriptor, and as a consequence updates the track of the semantic partition. This step is particularly important when video objects change their size, and when faced with the track management issues discussed in Section 5.1.

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 $\Pi_r^{n+1}$. Then the region descriptors corresponding to $\Pi_r^{n+1}$ are compared with those of $\Pi_r^n$.

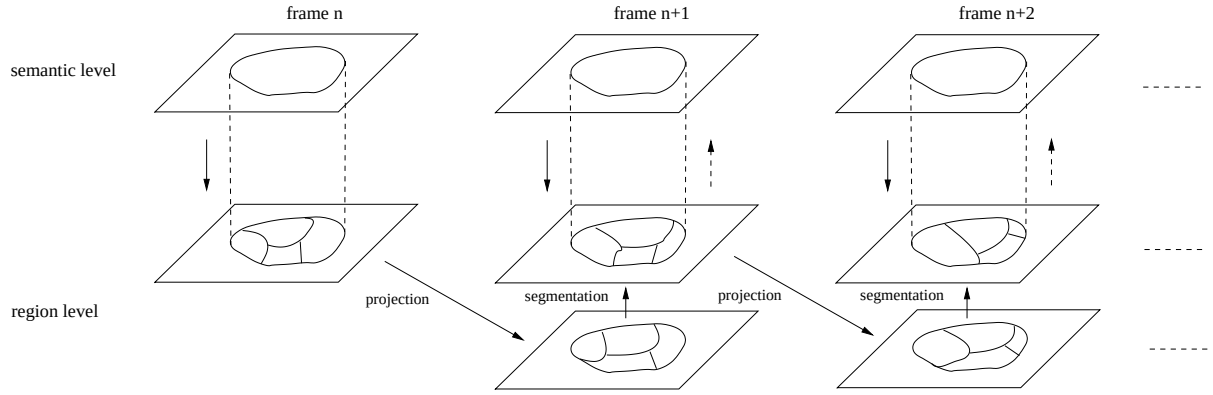


**Figure 4: Joint semantic-region tracking in the case of one video object. The semantic level $\Pi_s$ provides the focus of attention and it is improved by the feedback from the region level $\Pi_r$**

Each video object in the predicted partition is separately segmented into homogeneous regions, thus resulting in $\Pi_r^{n+1}$ (Figure 4). To verify the correctness of the *tentative correspondence* obtained with region descriptors projection, we consider the proximity between region descriptors in $\Pi_r^{n+1}$ and in $\Pi_r^n$. The proximity is computed by measuring the distance in the feature space between the region descriptors in frame $n+1$ and those in frame $n$. 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. This 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. Some features are given a higher importance than others in this distance measure, according to their reliability. The reliability is lowered for those features that have similar values in adjacent regions in frame $n$, so as to facilitate the paring. 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

$$d_{\bar{p},\bar{q}} = \min_q(d_{p,q}) = \min_p(d_{p,q}) \qquad (11)$$

If the condition in Eq.(11) is respected, the track is updated. Otherwise, region descriptors are iteratively paired based on a combination of the results of the *tentative correspondence* and the distance matrix $\mathbf{D}$. The region descriptors paired by the *tentative correspondence* are confirmed based on their distance, that is, the best point-to-point pairs are selected first, and the remaining ones are iteratively paired. This iterative process leads to the final correspondence between region descriptors. The final correspondence is finally exploited in the bottom-up feedback to update the semantic partition, thus providing the tracking results presented in the next section.

## 6. RESULTS

In this section, the results of proposed algorithm for multiple simulateneous object tracking are assessed. The tracking algorithm receives as input the results of the semantic and region segmentation and separates each single video object over time. The results are organized as follows: first the results on sequences with multiple simultaneous objects and appearance and disappearance of objects are shown. Second, the trajectories of the tracked objects are analyzed.

Figure 5 shows tracking results from sample frames of the sequence *Hall Monitor*. The goal of tracking is to extract the two moving persons separately. In this sequence, track management issues are appearance and disappearance of objects, and splitting. In the presentation of the results, the first row shows the original frames, whereas the second and the third row show the tracked video objects. The appearance of a new object, the man on the right hand side (column (a)), does not alter the tracking of the man in the left hand side. Since no descriptors are projected in the semantic partition corresponding to the man of the second row, a new track is initiated. A splitting occurrs between column (b) and (c) when the man on the left hand side leaves its case. In column (b), second row, it is possible to notice that the man and its suitcase still belong to the same semantic partition. In column (c), second row, the man and its suitcase are identified by two unconnected semantic partitions, but thanks to tracking they are interpreted as the same. The splitting has been detected and therefore the suitcase has not been interpreted as a new object. Figure 6 shows tracking results from sample frames of the test sequence *Highway*. This traf-

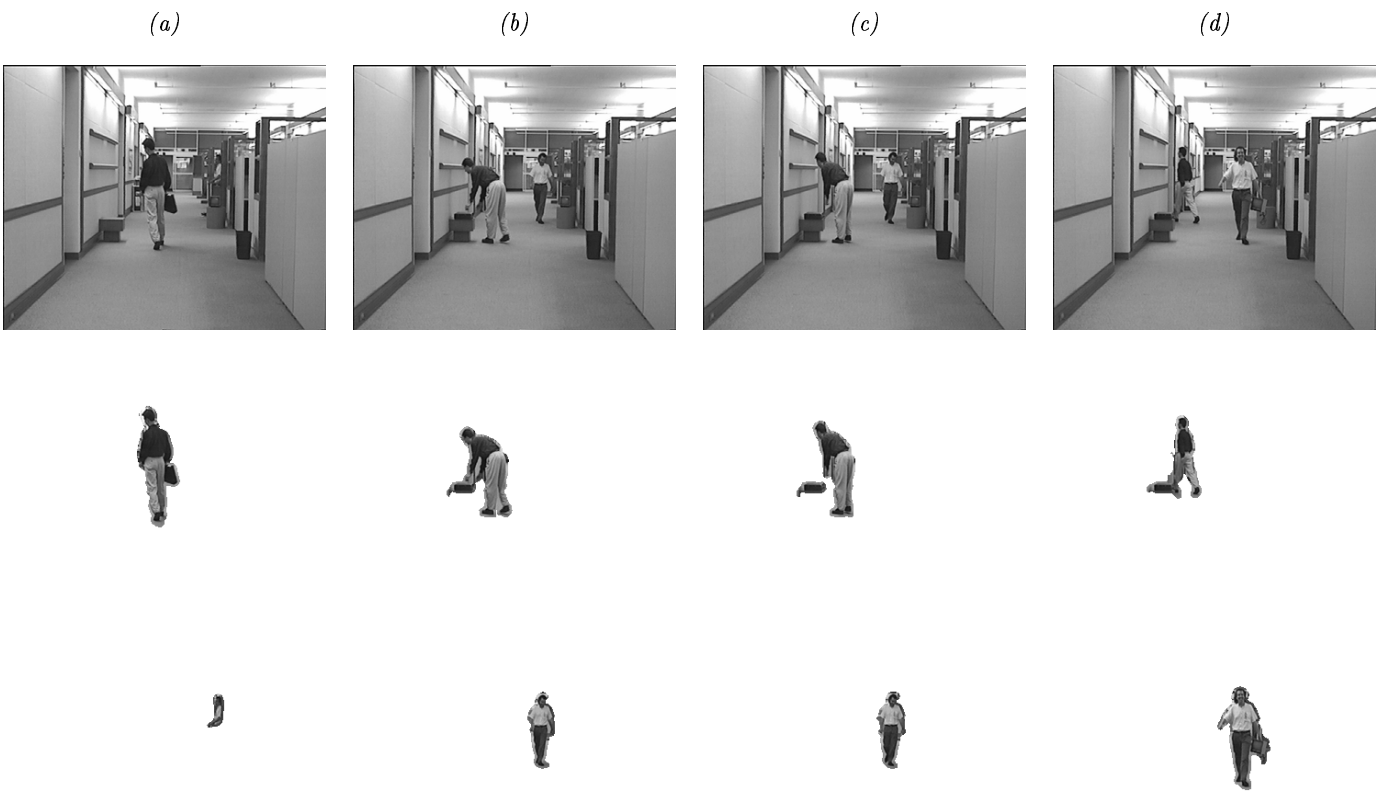


**Figure 5: Video object tracking results from sample frames of the sequence *Hall Monitor*. First row: Original frames. Second and third row show the separate video objects tracked over time**

fic surveillance sequence represents a highway with vehicles of different sizes driving on four lanes. The second and the third row show example of video object tracking. As for the previous example, the goal of tracking is to manage multiple simultaneous objects and their appearance and disappearance from the scene. The second row shows the tracking of a single vehicle until it exits the scene in the frame corresponding to column (c). The disappearance of the objects in the sequence does not alter the tracking of the objects. An example is reported with the object in the third row, which is correctly tracked even when the object shown is the second row leaves the scene. In the same way, all the other objects in the scene are separately tracked along the frames (Figure 8, second row). Figure 7 shows sample frames of the test sequence *Volley*. Two people are playing volleyball with a small object. The difficulties of this sequence are the presence of multiple simultanoues objects, non-rigid object motion and merging. The corresponding video object tracking results are presented in row 2-4. In particular, in column (d) a merging between the small object and the man on the right hand side occurs. The interaction between the region partition and the semantic partition helps in overcoming this problem and the objects are correctly tracked.

Video object tracking consists in determining the path of a known object within a video sequence. Figure 8 show the trajectories of the objects of the sequences described in the previous analysis. Column (a) displays a sample frame from the sequence and column (b) displays the corresponding trajectories of the video objects. The trajectories of the two video objects in the full sequence *Hall Monitor* (300 frames) are showed in the first row. The curve in the left show the path of the man entering first in the field of the camera. In case of camera calibration, these results could be complemented so as to provide the trajectories in the 3D scene. The second row shows the trajectories of the video objects of the sequence *Highway* in the frames from 110 to 160. 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 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. The third row shows the trajectories of the video objects of the sequence *Volley*. In the plot of tracking it is possible to notice the position of the two players represented by the curves on the left hand side and on the right hand side. The trajectories of the small object that passes from one player to the other is showed as well. The upper curve represents the track of the small object from the man of the left hand side to that

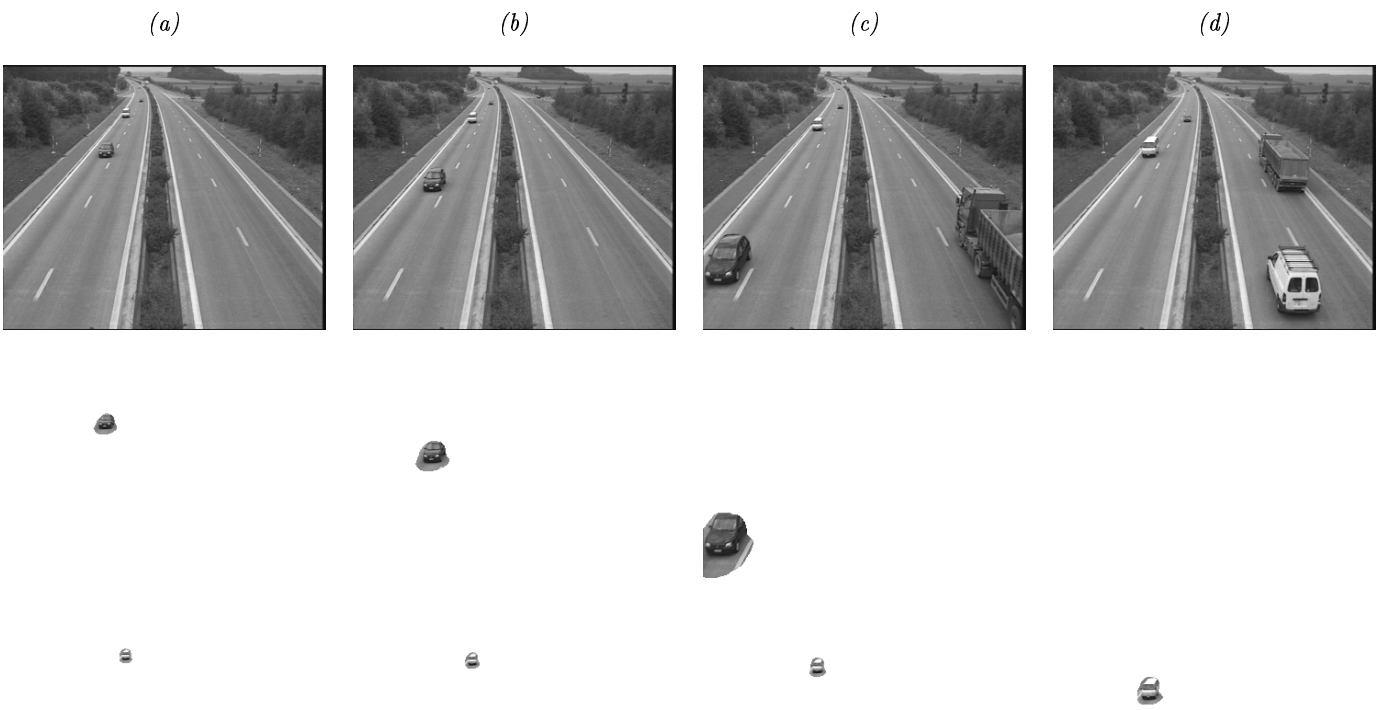


**Figure 6: Video object tracking results from sample frames of the sequence *Highway*. First row: original frames. Second and third row show the separate video objects tracked over time**

on the right. The trajectory is correctly detected until the partial occlusion corresponding to Figure 7(c). Next, the small object is completely covered by the hands of the man on the right hand side (total occlusion). When the small object is finally thrown back, a new trajectory is therefore initiated. To relate the two trajectories of the small object, the data association step should operate not only between subsequent frames, but also on a longer temporal window. This issue is part of our further research.

## 7. CONCLUSIONS

An automatic video object tracking algorithm has been presented. The tracking is based on interactions between high-level and low-level image analysis results. The the high-level result is a partition defining the video objects, and the low-level result is a partition formed by homogeneous regions. The regions have been represented by their region descriptors. Region descriptors are tracked from frame to frame as representative of video objects.

Experimental results have demonstrated that this approach has the ability to deal with multiple deformable objects, whose shape varies in time. Furthermore, it is very simple, because the tracking is based on the region descriptors, which represent a very compact piece of information about regions, and they are easy to define and track automatically. Also, the algorithm can handle different type of objects simultaneously. Finally, it can handle partial occlusions, and appearing and disappearing of objects from the scene.

Future work will be undertaken to extend the method in order to handle *total occlusions* and to track objects leaving and reentering the scene.

## 8. REFERENCES

[1] 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.

[2] 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.

[3] 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.

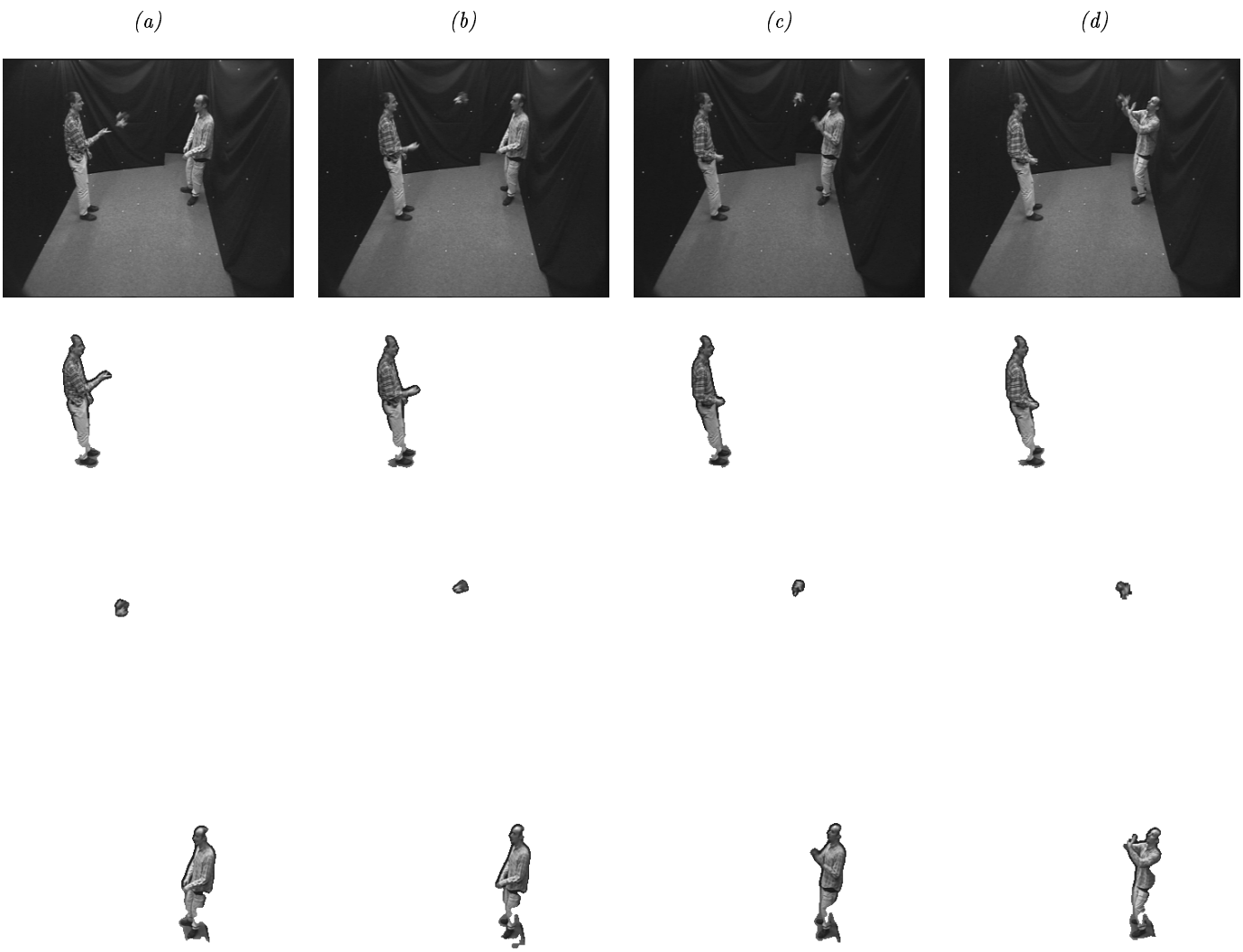


**Figure 7: Video object tracking results from sample frames of the sequence *Volley*. First row: original frames. Second to fourth row show the separate video objects tracked over time**

[4] 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.

[5] 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.

[6] 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.

[7] D. Koller, J. Weber, and J. Malik. Robust multiple car tracking with occlusion reasoning. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 189–196, 1994.

[8] 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.

[9] 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.

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

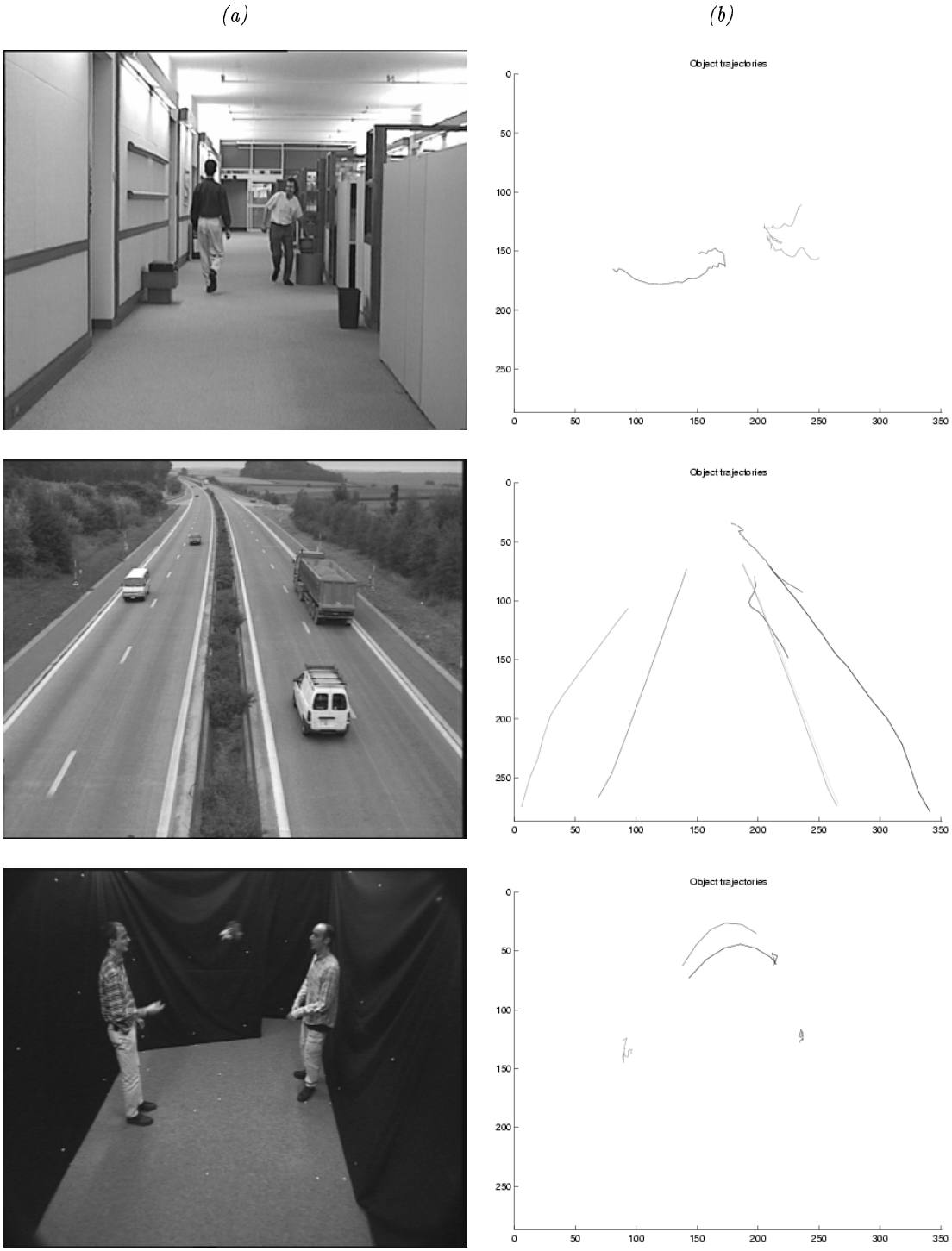


**Figure 8: Trajectories of video objects. (a) Sample frame from test sequences *Hall Monitor*, *Highway*, and *Volley*. (b) Trajectories of video objects for the sequence in the corresponding row. The horizontal and vertical axes of the graphs represent the width and the height of the frame, respectively**

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

[12] 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.