CVGIP: IMAGE UNDERSTANDING
Vol. 60, No. 2, September, pp. 119–140, 1994

Region-Based Tracking Using Affine Motion Models in Long Image Sequences

FRANÇOIS G. MEYER[1] AND PATRICK BOUTHEMY

IRISA/INRIA, Campus Universitaire de Beaulieu, 35042 Rennes Cedex, France

Received April 20, 1993; accepted February 8, 1994

This work investigates a new approach to the tracking of regions in an image sequence. The approach relies on two successive operations: detection and discrimination of moving targets and then pursuit of the targets. A motion-based segmentation algorithm, previously developed in the laboratory, provides the detection and discrimination stage. This paper emphasizes the pursuit stage. A pursuit algorithm has been designed that directly tracks the region representing the projection of a moving object in the image, rather than relying on the set of trajectories of individual points or segments. The region tracking is based on the dense estimation of an affine model of the motion field within each region, which makes it possible to predict the position of the target in the next frame. A multiresolution scheme provides reliable estimates of the motion parameters, even in the case of large displacements. Two interacting linear dynamic systems describe the temporal evolution of the geometry and the motion of the tracked regions. Experiments conducted on real images demonstrate that the approach is robust against occlusion and can handle large interframe displacements and complex motions. © 1994 Academic Press, Inc.

1. INTRODUCTION

The ability to track a moving target is common to many biological vision systems. Movements of the eyes and the head are responsible for the fixation of a moving target on the fovea. In the human visual system, eyes and the vestibular system provide a sensory input to complex control systems that govern eye movements [39]. When a moving stimulus is detected by the visual system, first a saccadic eye movement shifts the object of interest near the fovea. The pursuit is then initiated and smooth eye movements stabilize the moving target on the fovea. The tracking is therefore based on two distinct stages: detection and discrimination of the moving target and then the pursuit of the object of interest.

We advocate that a computer-vision tracking algorithm should also rely on these two successive steps. This paper explores a new approach to the problem of tracking objects in a sequence of images. The approach relies on the two previous operations: detection and discrimination of the moving targets and then pursuit of the targets. A motion-based segmentation detects targets moving with respect to the camera. A pursuit algorithm is then initiated for each moving target. The article focuses on the pursuit stage. Regions provide robust primitives to perform the tracking at an "object level." An affine model of the 2-D motion field characterizes the motion of each tracked region. The tracking algorithm relies on two interacting dynamic systems that capture the temporal evolution of the geometry and the motion of the successive projections in the image plane of objects moving in the scene. Standard filtering techniques generate recursive estimates of the region geometry and velocity. This paper is organized as follows. In the next section, we review existing tracking methods, and we propose a general framework to classify them. Section 3 describes the tracking algorithm. The estimation of the motion parameters that describe the motion of each tracked target is presented in Section 4. Section 5 develops the geometric filter responsible for the update of the shape and the position of each tracked region. Results of experiments conducted on real images are given in Section 6. A short version of the paper was presented in [26].

2. RELATED WORK

Even though a great variety of tracking algorithms have been proposed in the recent years, it is possible to classify the numerous and different approaches in four distinct categories, according to their ability to achieve the detection and discrimination task. The first set contains methods that have not addressed the necessary detection and discrimination stage. The second set consists of methods that are not required to detect moving objects. Methods in the third set rely on a high-level interactive detection and discrimination stage. Finally, the fourth set is composed of methods that use an a priori knowledge of the shape of the object to perform the detection and discrimi-

[1] The author is now with the Departments of Mathematics and Diagnostic Radiology, Yale University, New Haven, Connecticut 06510. E-mail:meyer@noodle.med.yale.edu.

1049-9660/94 $6.00

nation stage. Let us examine each of the four sets in detail.

The first set comprises the tracking algorithms based on an active vision paradigm [1, 9]. The concept of active vision relies on the ability to control the parameters of the camera in a fairly antropomorphic way. It appears that visual tasks are facilitated when the camera motion is actively controlled [1]. Active vision tracking algorithms require the object of interest to be foveated (i.e., located at the center of the image). However none of these methods have addressed the initial problem of detecting and isolating the moving target.

Methods included in the second set eliminate the difficult problem of discrimination. An intermediate representation of each image, similar to the "primal sketch" of David Marr [23], is elaborated. Edge segments are then tracked along the sequence [13, 14, 40]. An interpretation of the image sequence in terms of moving objects is not possible at this stage. A further processing is required, which should cluster edges into consistent objects [40]. The estimation of the velocity of the target is based on the correspondence process performed by the tracking. When the tracking is achieved directly in the image plane, constant velocity models [13] or constant acceleration models [14] describe the evolution of the target in the image plane. These models of the target position are polynomials in time. They only provide a local approximation to the real projected trajectories of objects and thus are ill-suited for a long-range tracking. Alternative approaches have assumed "smoothness of motion" to establish the correspondence between features (specific points, edge corners) in multiple frames [32, 36] and formulate the problem as the minimization of a cost function.

The third class is made up of methods based on active contours [2, 20, 37]. These methods exploit a user interactive initialization of the tracking process to perform the detection and discrimination task. The shape of the contour is coarsely delineated in the first frame of the sequence. The initial contour converges iteratively toward the solution of a differential equation. This solution corresponds to a local minimum of an energy function that takes into account the spatial gradient of the intensity function as well as a smoothness constraint about the contour. If the initialization is well performed the contour is attracted toward the expected boundary of the moving target. No restriction is imposed on the shape of the target nor on the type of motion: rigid or deformable. The solution obtained in a given frame serves as the initial condition to the minimization problem in the subsequent frame. A severe limitation of these methods is their inability to track targets moving with a large velocity. Indeed no motion information is available to predict the initial position of the contour in the next frame.

Finally, the fourth set consists of model-based tracking algorithms [17, 19, 21, 34]. A 3-D polyhedral model of the object of interest is given. Detection and discrimination of the moving target thus reduce to a problem of recognition. In [21] this task is achieved with a Newton method and requires a good initial knowledge of the object location. These methods are well suited for polyhedral and manufactured objects where a wire frame model of the object exists.

It is clear from the previous analysis that no tracking method has properly addressed the detection and discrimination stage in the more general case. As a consequence, the tracking of a complete target at an object level and with no a priori information about the shape or the location of the target in the first frame has failed to be addressed. Moreover in most tracking methods the motion information necessary for the pursuit is inferred from the tracking and depends on the success of the correspondence stage. We propose a new approach to the problem of tracking a moving target from a sequence of monocular images. The two successive operations, detection and discrimination, and then pursuit are properly addressed. The detection and discrimination stage is successfully achieved thanks to a motion-based segmentation method previously developed in the laboratory and presented in [7]. Moving objects are detected and the contour of the projection of each object in the image plane is obtained. A pursuit is then initiated for each object. The pursuit directly tracks the complete projection of the target in the image. We present now the tracking method.

3. THE TRACKING ALGORITHM

3.1. Detection and Discrimination of a Moving Target

We need to detect moving objects. This task is achieved thanks to a motion-based segmentation that divides images into regions corresponding to different objects moving with distinct velocities. The motion-based segmentation method, fully described in [7], ensures stable motion-based partitions owing to a statistical regularization approach. The segmentation problem is formulated as the estimation of a label field, modeled by a Markovian random field. This approach requires neither explicit 3-D measurements nor the estimation of optical flow fields. It mainly relies on the spatiotemporal variations of the intensity function while making use of 2-D affine motion models. Each region can be interpreted as the silhouette of the projection of an object in the scene, in relative motion with respect to the camera. Each time a new object appears in the field of view a pursuit is initiated. New regions are easily detected since they are assigned new labels by the motion-based segmentation. We present now the pursuit algorithm.

3.2. Pursuit of a Moving Target

The pursuit stage handles directly the region representing the projection of the considered object in the image. Tracking the target at an object level offers at least three important advantages. First, the tracking is more robust and less sensitive to partial occlusion as shown in the experiments. Second, for each tracked region a dense estimation of the motion field within the region can be performed. We actually use a parametric model of the motion field. Finally, the interpretation of the target as an object is straightforward, and its trajectory in the image plane is directly attainable.

The motion information is not inferred from the tracking as in standard tracking algorithms, but the tracking itself is based on the target motion estimated from the sequence. The target motion is extracted from two successive frames and makes it possible to anticipate the position of the target in the next frame. It is interesting to note that, in a similar way in the human visual system, smooth pursuit eye movements utilize an estimate of the target motion, in order to compensate for the slip of the target on the retina during the pursuit of a target [39].

We propose an approach with a complete model for the prediction and update of the object geometry and motion. Figure 1 shows the overview of the algorithm. We use two filters: a *geometric filter* and a *motion filter*. The geometric filter and the motion filter estimate the shape, the position, and the motion of the region. The two filters interact: the estimation of the motion parameters enables the prediction of the geometry of the region in the next frame. Our approach has some similarities with the one proposed in [33]. The authors constraint the target displacement in the image plane to be a 2-D affine transform. An overdetermined system makes it possible to compute the motion parameters. However, the region representation and the segmentation step are quite different and less efficient. Moreover neither the motion parameters nor the region geometry were filtered over time.

We describe now the model that characterizes the evolution of the tracked region. An affine transform is used to capture the evolution of a region between two successive frames. We explain how the parameters of each affine transform, associated with a tracked region, can be estimated. It is shown that the affine transform can be obtained from a set of instantaneous motion parameters, which can be efficiently estimated, as explained in Section 4.1.

3.2.1. *The Region Evolution Model*

We assume that each region R in the image at time $t + 1$ is the result of an affine transformation $(\mathbf{\Phi}, \mathbf{u})$ of the region R, in the image at time t. Hence every point $(x(t), y(t)) \in R$ at time t will be located at $(x(t + 1), y(t + 1))$ at time $t + 1$, with

$$\begin{pmatrix}x\\ y\end{pmatrix}(t+1) = \mathbf{\Phi}(t)\begin{pmatrix}x\\ y\end{pmatrix}(t) + \mathbf{u}(t). \qquad (1)$$

The affine transform has already been used to model small transformations between two images [16, 22, 33]. We need to determine the parameters of the affine transform. To achieve this, we first note that $(\mathbf{\Phi}, \mathbf{u})(t)$ can be derived from the set of parameters of a local affine motion model. We approximate the 2-D motion field within the region R with an affine model:

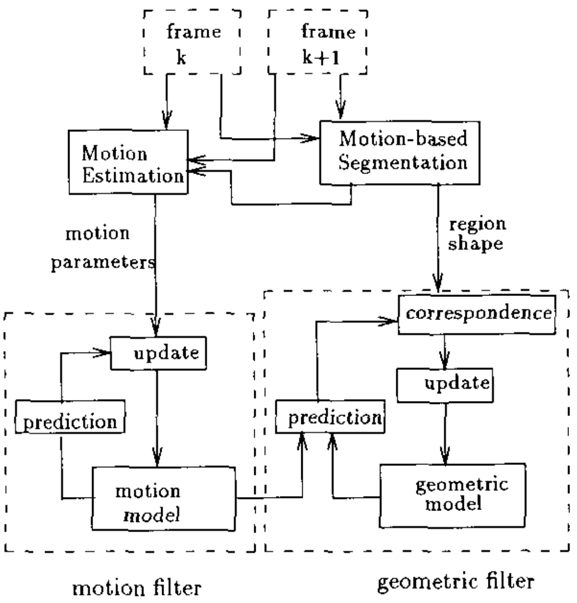


FIG. 1. The complete region-based tracking filter.

$$\forall(x, y) \in R,\ \begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix}(t) = \mathbf{A}(t)\begin{pmatrix}x\\ y\end{pmatrix}(t) + \mathbf{b}(t). \qquad (2)$$

Affine models of the motion field have also been proposed in, e.g., [6, 11, 29]. The 2×2 matrix $\mathbf{A}(t)$ stands for a large class of motion: rotation, scaling, shear, etc. We have shown in [25] that it is theoretically possible to retrieve the 3-D motion from the affine parameters and their partial derivatives with respect to time. In particular, the time to collision was accurately estimated from monocular sequences of real images [25]. We advocate therefore that an affine model of the 2-D motion field conveys valuable and robust information about the 3-D motion and structure.

If we now expand $(x, y)^T$ in Taylor series and if we drop the second-order terms

$$\begin{pmatrix}x\\ y\end{pmatrix}(t+1) = \begin{pmatrix}x\\ y\end{pmatrix}(t) + \delta t\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix}(t),$$

where δt is the time step between two successive frames. From (1), (2), and above we get

$$\mathbf{\Phi}(t) = \mathbf{I}_2 + \delta t\mathbf{A}(t) \quad \text{and} \quad \mathbf{u}(t) = \delta t\mathbf{b}(t), \qquad (3)$$

where $\mathbf{I}_2$ is the 2×2 identity matrix.

We have converted the problem of estimating the matrix

$\mathbf{\Phi}(t)$ and the vector $\mathbf{u}(t)$ into the problem of estimating the parameters of the affine model of the 2-D motion field $(\mathbf{A}, \mathbf{b})(t)$. A multiresolution scheme has been developed to solve the latter problem and is presented in the next section. The method provides accurate and robust estimates of the motion parameters, even in the case of large displacements.

4. MOTION ESTIMATION

4.1. Multiresolution Estimation of the Motion Parameters

The idea is to estimate the motion parameters with a "coarse-to-fine" strategy. We use a Gaussian low-pass pyramid of each image. A rough estimate of the parameters is obtained at the lowest resolution. The estimate is subsequently refined using the higher-resolution images. The approach is similar to [30] where the authors derive estimates of the 3-D translational and rotational velocity in the limited case where the depth of the scene is constant or known. Similar hierarchical methods have already been employed for motion measurement. For instance, a hierarchical estimation of the optical flow with a gradient-based algorithm is proposed in [15]; in [3] the author describes a hierarchical matching scheme for the determination of dense displacements fields.

We build two low-pass Gaussian pyramids for each image at time t and $t + \delta t$, as described in [10]. We build a pyramid for the segmented image, which provides the delineations of regions at each level. First, we estimate at the lowest resolution level, L, the six motion parameters of each region in the image, with a least-squares fit to normal flows. We use the well-known image flow constraint equation [18] that relates the motion field $\mathbf{v}(x, y)$ with the spatial gradient of the image intensity $\nabla\mathbf{I}(x, y)$, and with the partial derivative with respect to time of the image intensity $I_t(x, y)$:

$$\nabla\mathbf{I}(x, y) \cdot \mathbf{v}(x, y) + I_t(x, y) = 0. \tag{4}$$

If we consider an affine model of the 2-D motion field $\mathbf{v}_{(\mathbf{A},\mathbf{b})}(x, y)$ given by (2), we can measure the adequacy of the model with the real flow using the following "data-model adequacy" variable:

$$\psi_{(\mathbf{A},\mathbf{b})}(x, y) = \nabla\mathbf{I}(x, y) \cdot \mathbf{v}_{(\mathbf{A},\mathbf{b})}(x, y) + I_t(x, y). \tag{5}$$

A maximum-likelihood estimation of $(\mathbf{A}, \mathbf{b})$ is achieved, which reduces here, assuming that $\psi_{(\mathbf{A},\mathbf{b})}(x, y)$ are independent zero-mean Gaussian variables with the same variance, to a least-squares estimation. Each image in the Gaussian pyramid is a blurred and subsampled (of a factor 2 in both directions) version of its predecessor. The displacements measured in pixel, at level $l + 1$, are half the displacement at level l. As a result, the optical flow constraint equation (4), which assumes that the image motion is small, can be more easily applied at the lowest resolution level L.

Then we refine the estimate using the higher resolution levels, with an equation similar to (5). For the clarity of the presentation and to lighten the notation, we will drop the time variable t wherever it can be done without causing confusion. Let $\mathbf{p}^l$ be a point within a region at a given level l. Let $\mathbf{v}_{(\mathbf{A},\mathbf{b})^l}(\mathbf{p}^l)$ be the affine model of the velocity at location $\mathbf{p}^l$:

$$\mathbf{v}_{(\mathbf{A},\mathbf{b})^l}(\mathbf{p}^l) \triangleq \mathbf{A}^l\mathbf{p}^l + \mathbf{b}^l. \tag{6}$$

The motion field estimated at the coarser level $l + 1$ is projected on the current level and accounts for an initial displacement. Let $\delta\mathbf{p}^l$ be the displacement at location $\mathbf{p}^l$. Assuming that the brightness of a moving point remains constant all along the trajectory of the point, we get

$$I(\mathbf{p}^l + \delta\mathbf{p}^l, t + \delta t) = \mathbf{I}(\mathbf{p}^l, t).$$

In fact (4) is the differential version of the above equation. Since $\delta\mathbf{p}^l = \mathbf{v}_{(\mathbf{A},\mathbf{b})^l}(\mathbf{p}^l)\delta t$, we estimate the displacement $\delta\mathbf{p}^l$. Let us assume that $\mathbf{p}^{l+1}$ is the father of $\mathbf{p}^l$ at level $l + 1$, and let $\delta\hat{\mathbf{p}}^{l+1}$ be the displacement estimated, at level $l + 1$, at location $\mathbf{p}^{l+1}$. We can project $\delta\hat{\mathbf{p}}^{l+1}$ on the level l. The decimation used to build the pyramid reduces each image by a factor two in each direction. Hence the projection of $\delta\hat{\mathbf{p}}^{l+1}$ on the level l is equal to $2\delta\hat{\mathbf{p}}^{l+1}$. It serves as an initial estimate to compute $\delta\mathbf{p}^l$. We define $\delta^2\mathbf{p}^l$ as the incremental estimate to be computed at level l (see Fig. 2):

$$\delta^2\mathbf{p}^l \triangleq \delta\mathbf{p}^l - 2\delta\hat{\mathbf{p}}^{l+1}. \tag{7}$$

It follows from above that

$$I(\mathbf{p}^l + 2\delta\hat{\mathbf{p}}^{l+1} + \delta^2\mathbf{p}^l, t + \delta t) = I(\mathbf{p}^l, t).$$

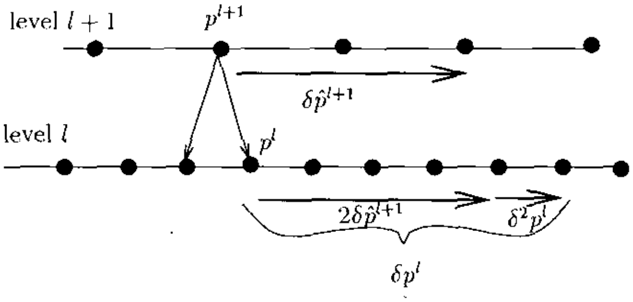


FIG. 2. Hierarchical estimation of the motion parameters.

Using a first-order expansion of I about $\mathbf{p}^l + 2\delta\mathbf{p}^{l+1}$ we get

$$I(\mathbf{p}^l + 2\delta\hat{\mathbf{p}}^{l+1}, t + \delta t) + \nabla\mathbf{I}(\mathbf{p}^l + 2\delta\hat{\mathbf{p}}^{l+1}, t + \delta t) \cdot \delta^2\mathbf{p}^l = I(\mathbf{p}^l, t). \quad (8)$$

Let $\widehat{(\mathbf{A}, \mathbf{b})}^{l+1}$ be the estimate of $(\mathbf{A}, \mathbf{b})^{l+1}$ obtained at level $l + 1$. We have

$$\delta\hat{\mathbf{p}}^{l+1} = \mathbf{v}_{\widehat{(\mathbf{A},\mathbf{b})}^{l+1}}(\mathbf{p}^{l+1})\delta t = (\hat{\mathbf{A}}^{l+1}\mathbf{p}^{l+1} + \hat{\mathbf{b}}^{l+1})\delta t. \quad (9)$$

Let us define $\Delta\mathbf{A}^l \triangleq \mathbf{A}^l - \hat{\mathbf{A}}^{l+1}$ and $\Delta\mathbf{b}^l \triangleq \mathbf{b}^l - 2\hat{\mathbf{b}}^{l+1}$ as the refinement of the motion parameters to be estimated at the current level l. Then from (6), (7), and (9) and since $\mathbf{p}^l = 2\mathbf{p}^{l+1}$ we obtain

$$\delta^2\mathbf{p}^l = (\Delta\mathbf{A}^l\mathbf{p}^l + \Delta\mathbf{b}^l)\,\delta t.$$

Consequently (8) leads to

$$\nabla\mathbf{I}(\mathbf{p}^l + 2\delta\hat{\mathbf{p}}^{l+1}, t + \delta t) \cdot (\Delta\mathbf{A}^l\mathbf{p}^l + \Delta\mathbf{b}^l)\,\delta t + I(\mathbf{p}^l + 2\delta\hat{\mathbf{p}}^{l+1}, t + \delta t) + I(\mathbf{p}^l, t) = 0. \quad (10)$$

Equation (10) is linear with respect to $\Delta\mathbf{A}^l$ and $\Delta\mathbf{b}^l$. If we consider (10) for all points $\mathbf{p}^l$ within the region we obtain an overconstrained system of linear equations. Least-squares estimates $(\Delta\hat{\mathbf{A}}^l, \Delta\hat{\mathbf{b}}^l)$ of $(\Delta\mathbf{A}^l, \Delta\mathbf{b}^l)$ can thus be obtained. At the lowest resolution level L we use (5) to derive the estimate $(\hat{\mathbf{A}}^L, \hat{\mathbf{b}}^L)$ of $(\mathbf{A}^L, \mathbf{b}^L)$:

$$\nabla\mathbf{I}(\mathbf{p}^L, t) \cdot (\mathbf{A}^L\mathbf{p}^L + \mathbf{b}^L)\,\delta t + I(\mathbf{p}^L, t + \delta t) - I(\mathbf{p}^L, t) = 0. \quad (11)$$

At each level we have

$$\hat{\mathbf{A}}^l = \sum_{k=l}^{L-1} \Delta\hat{\mathbf{A}}^k + \hat{\mathbf{A}}^L \quad \text{and} \quad \hat{\mathbf{b}}^l = \sum_{k=l}^{L-1} 2^{k-l}\Delta\hat{\mathbf{b}}^k + 2^{L-l}\hat{\mathbf{b}}^L.$$

Rather than incrementaly warping the image at time t toward the image at time $t + 1$, as proposed in [6], we use an "incremental" version of the image flow constraint equation (10) at each level. It is possible to give a geometric representation of (10). For the clarity of the figure we represent the image in one dimension only in Fig. 3. The interpretation is absolutely the same in two dimensions. If we use a discrete version of (4) to estimate the displacement $\widetilde{\Delta p^l}$, at location p^l, we get

$$\nabla I(p^l, t) \cdot \frac{\widetilde{\Delta p^l}}{\delta t} + \frac{I(p^l, t + \delta t) - I(p^l, t)}{\delta t} = 0,$$

where $\nabla I(p^l, t)$ is the slope of the intensity function at location p^l, and at time t. We obtain

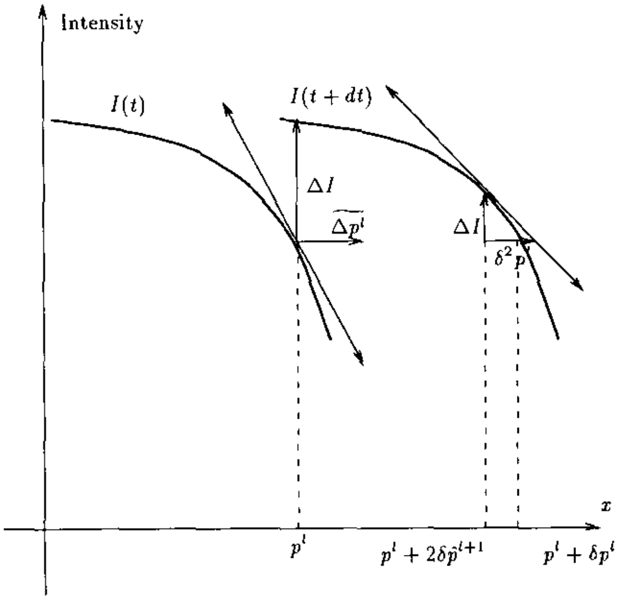


FIG. 3. Geometric interpretation in one dimension of the computation of the incremental estimate $\delta^2 p^l$.

$$\widetilde{\Delta p^l} = -\frac{I(p^l, t + \delta t) - I(p^l, t)}{\nabla I(p^l, t)}.$$

We note that the above discretization is valid if the intensity function $I(p, t)$ is linear with respect to space and time parameters. In other words, if the displacement is small, the intensity function around p^l can be locally approximated with a first-order expansion, and the discrete version of the optical flow constraint equation will be approximatively satisfied. In the example in Fig. 3 the estimate of $\widetilde{\Delta p^l}$ is not accurate.

However, if we assume that we have an initial estimate of the displacement $2\delta\hat{p}^{l+1}$ we can use (10) to obtain an incremental estimate $\delta^2 p^l$,

$$\delta^2 p^l = -\frac{I(p^l + 2\delta\hat{p}^{l+1}, t + \delta t) - I(p^l, t)}{\nabla I(p^l + 2\delta\hat{p}^{l+1}, t + \delta t)},$$

where $\nabla I(p^l + 2\delta\hat{p}^{l+1}, t + \delta t)$ is the slope of the image intensity function at location $p^l + 2\delta\hat{p}^{l+1}$ and at time $t + \delta t$. If the initial estimate is close enough to the real displacement δp^l, then the small incremental displacement will be accurate.

4.2. Iterative Refinement at the Same Resolution Level

The incremental refinement presented in the previous section is actually first performed at the same level before exploiting the next higher level. An equation very similar to (10) is derived and the incremental estimate is also obtained with a least-squares estimation. Performing a refinement at the same level enables us to capture all the motion information available at that level before exploiting the next higher resolution level. Two or three refinements are usually performed at the same level.

4.3. Computational Issues

Number of levels of the pyramid. To determine the number of levels of the pyramid, we have to take into account two criteria: the size of each region present in the segmented image and the magnitude of the parameters to be estimated. Indeed, on one hand we noted that the level at which the multiresolution estimate is correct depends only on the magnitude of the motion parameters to be estimated. If the estimate is correct for a certain level, no significant improvement can be noted using more levels. In [5] the authors pointed out that the velocity estimated with the image flow constraint equation (4) corresponds to the true velocity if the spatial discretization step is equal to the interframe displacement. The level of the pyramid should be chosen such that the discretization step at the coarsest level is just larger than the displacement to be estimated at that level. On the other hand, at each successive level the region area is divided by four. At the lowest resolution level the least-squares estimation of the motion parameters may give erroneous results if there are not enough points within the region. Our approach is the following. We estimate the motion parameters with the largest allowable pyramid (defined by the size of the region). Then we select the minimum level at which no improvement could be noted if we were using the next higher level. The problem of directly deciding at which level the estimation should be performed is an interesting one and has not been solved yet by our approach.

Selection of the points of the region kept for the least-squares fit. As pointed out in [35] least-squares estimates are quite sensitive to outliers. Therefore a procedure has been developed to discriminate between points of the region that are kept for the least-squares estimation and other points of the region that are discarded. The principle is the following. Equations (10) and (11) assume that $\mathbf{p}^l$ at time t and $\mathbf{p}^l + 2\delta\hat{\mathbf{p}}^{l+1}$ at time $t + \delta t$ are in the same region. In order to ensure this requirement when computing the least-squares system, we discard all points $\mathbf{p}^l$ in the image at time t for which $\mathbf{p}^l + 2\delta\hat{\mathbf{p}}^{l+1}$ is not in the region at time $t + \delta t$. An estimate of the location of the region in the image at time $t + \delta t$ is provided using the prediction of the motion parameters at time t, which are generated by a Kalman filter, as explained in next section. If a part of the tracked object is visible at time t and becomes occluded at time $t + \delta t$, then the procedure will discard all the points occluded at time $t + \delta t$. If the occlusion is only partial, a sufficient number of points remain to obtain a reliable least-squares estimate. Consequently an interesting advantage of our approach is that, even in the case of partial occlusion, reliable motion estimates can still be obtained.

Center of the affine transform. We discuss here the choice of the center of the affine model of the motion field. We have assumed so far that the center of the affine motion model was the origin (0, 0) of the image. It is actually more convenient to select the centroid of each region as the center of the motion model. Letting (x_g, y_g) be the centroid of a given region, we denote $(\mathbf{A}_g, \mathbf{b}_g)(t)$ the affine model of the motion field centered at (x_g, y_g):

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix}(t) = \mathbf{A}_g(t)\begin{pmatrix} x - x_g \\ y - y_g \end{pmatrix}(t) + \mathbf{b}_g(t).$$

It is easy to prove that

$$\mathbf{A}_g = \mathbf{A} \quad \text{and} \quad \mathbf{b} = \mathbf{b}_g - \mathbf{A}\begin{pmatrix} x_g \\ y_g \end{pmatrix}.$$

Consequently $\mathbf{A}$ does not depend on the choice of the center. However, we note in (10) and (11) that points that are at a large distance from the origin (with a high $\mathbf{p}_x^l$ and $\mathbf{p}_y^l$ components) may have an unreasonable influence on the least-squares fit. We prefer to have an homogeneous distribution of the weighting factors in (10) and (11), and thus we use the centroid as the center of the affine model. Moreover, if the centroid is chosen as the center of the affine motion model, then the vector $\mathbf{b}_g$ can be physically interpreted as the value of the motion model at the centroid. It is thus easier to describe the temporal evolution of $\mathbf{b}_g$ than to describe the temporale volution of $\mathbf{b}$. This aspect plays an important role if we perform a temporal filtering of the motion parameters, as presented in the next section.

Finally, bilinear interpolations of I and $\nabla \mathbf{I}$ must be computed in (10) for points which are not on the grid.

4.4. Recursive Estimation of the Motion Parameters

The multiresolution method provides us with instantaneous measurements of the motion parameters. We need to filter these measurements to generate more accurate and more stable estimates. A Taylor-series expansion of each component of $(\mathbf{A}, \mathbf{b})(t)$ yields a local approximation of the temporal evolution of the parameter. After having experimented with a third-order expansion in [26], we have observed that the resulting dynamic system tends to diverge when the filter does not receive measurements for a long time interval. Even though using the first three terms of the expansion results in a more precise model, we have noted that a filter based on a second-order expansion is usually more robust. The state transition matrix thus only takes into account the first two terms of the expansion. Consequently the dynamic system can generate predicted estimates of each of the motion parameters, based on the previous measurements, in situations of long occlusions, as shown in the experiments. Moreover, we have observed in many sequences that the correlation coefficients between the six components of $(\widehat{\mathbf{A}, \mathbf{b}})(t)$ are negligible. Assuming that the six variables are indeed independent, we have decided to decouple the six filters. Let $a_i(t)$, $i = 1, \ldots, 6$, be one of the components of $(\widehat{\mathbf{A}, \mathbf{b}})(t)$, and let $\tilde{a}_i(t)$ be the instantaneous measurement of $a_i(t)$, given by the multiresolution algorithm. The following dynamic system provides a local approximation to the temporal evolution of each motion parameter a_i,

$$\begin{cases} \begin{bmatrix} a_i \\ \dot{a}_i \end{bmatrix}(t+1) = \begin{bmatrix} 1 & \delta t \\ 0 & 1 \end{bmatrix}\begin{bmatrix} a_i \\ \dot{a}_i \end{bmatrix}(t) + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \end{bmatrix}(t) \\ \tilde{a}_i(t) = a_i(t) + \beta(t), \end{cases} \tag{12}$$

where $\beta(t)$ is a sequence of zero-mean Gaussian white noise of variance σ_β^2. $[\varepsilon_1, \varepsilon_2]^T(t)$ is a sequence of zero-mean Gaussian noise vectors of covariance matrix $\mathbf{R}$ [27]:

$$\mathbf{R} = \sigma_{\mathbf{R}}^2 \begin{bmatrix} \dfrac{\delta t^3}{3} & \dfrac{\delta t^2}{2} \\ \dfrac{\delta t^2}{2} & \delta t \end{bmatrix}.$$

A standard Kalman filter [24] generates recursive estimates of each motion parameter. The initialization of the Kalman filter is thoroughly discussed in [27].

4.5. Experiments

We present two experiments conducted on real data that illustrate the performance of the filter. The plots represent the evolution of the parameters of the affine motion model, namely $(\mathbf{A}, \mathbf{b})(t)$, with respect to time. The origin of the coordinate system is the upper left corner of the image. The first coordinate axis is horizontal and is oriented from left to right. The second coordinate axis is vertical and is oriented from top to bottom.

4.5.1. A Quantitative Experiment

The first experiment was performed on real images to validate the multiresolution estimation method. We compare here the measurements given by our method with the ground truth determined by independent means. Rather than generating a synthetic motion in the image, we calculated the exact motion field caused by the known 3-D motion of a camera in the scene.

The poster sequence. The sequence is composed of 32 images of size 256 × 352 pixels. It has been acquired with an experimental cell constituted by a camera mounted on the end effector of a 6 d.o.f. robot. The camera is undergoing a planar motion toward a poster pinned on a wall. The first and the last frames of the sequence are displayed in Fig. 4. The constant velocity of the camera, in the camera coordinate system (see Fig. 5), is $U = 125$ mm/s, $V = 0$, and $W = 250$ mm/s. Initially the camera is at 70 cm from the wall and slightly slanted (17°), i.e., the image plane is not parallel to the wall. The parameters of the affine model are calculated analytically with the expressions relating the first-order approximation of the optical flow with the 3-D kinematics parameters of a planar surface undergoing rigid motion and the intrinsic parameters of the camera [25]. For each parameter we

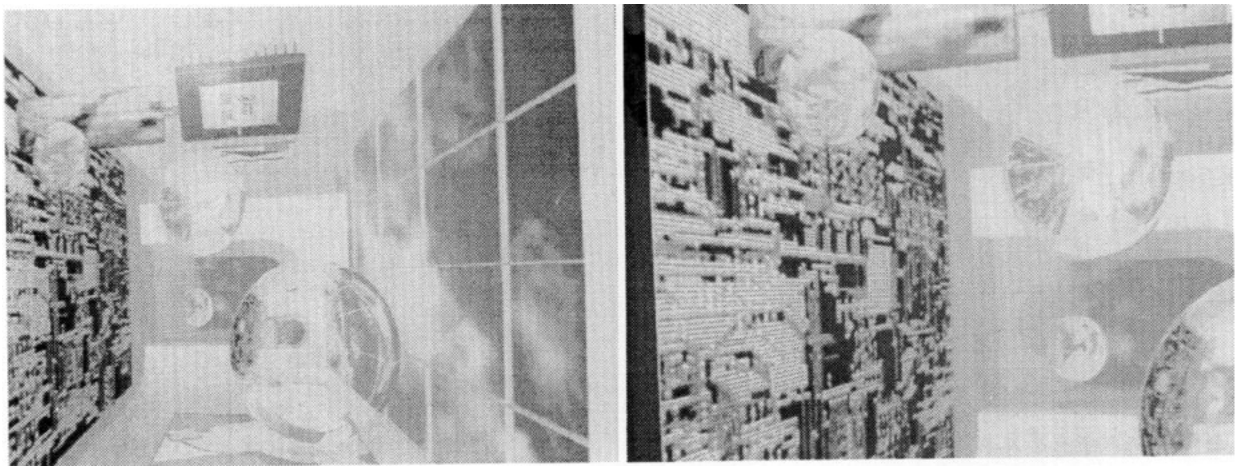


FIG. 4. Sequence poster at time t_1 (left), and at time t_{32} (right).

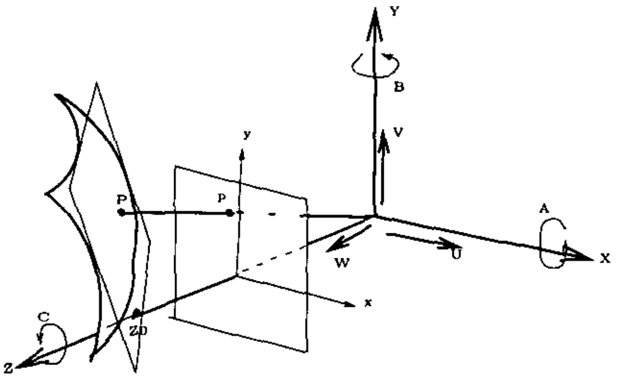


FIG. 5. Camera coordinate system.

have plotted the true value of the parameter and the estimate obtained with different pyramid levels: three levels, two levels, and one level (monoresolution). Comparative performance results from the different levels are displayed in Figs. 6 and 7, which show respectively the estimates of $b_1(t)$ and the estimates of $A_{1,1}(t)$, and $A_{2,2}(t)$. Since $A_{1,2} = 0$, $A_{2,1} = 0$, and $b_2 = 0$ for the considered motion we did not plot these coefficients. We can note the good correspondence between the three level multiresolution measurements and the true parameters. Because the magnitude of the motion is important (4 to 6 pixels per frame) we need three levels to accurately estimate the motion parameters. This experiment validates the multiresolution method for the estimation of the motion parameters of the affine model of the 2-D velocity field.

4.5.2. *A Qualitative Experiment*

The second experiment illustrates the ability of the algorithm to obtain motion estimates which are qualitatively good. The quality of the motion parameters is evaluated based on the good performance of the tracking, presented in Section 6.

The crossroad sequence. The second sequence is composed of 66 real images of size 288 × 344 pixels. The experiment addresses the problem of long occlusions. The scene takes place at a crossroad as shown in Fig. 8. A white van is coming from the left of the image and going to the right. A black car is driving behind the van at the same speed. A white car is coming from the opposite directions. While crossing the image from right to left with an oblique trajectory in the image plane, the white car is approaching the camera. The car disappears behind the van during 23 images and reappears at the end of the sequence. A new label is assigned to the reappearing car by the motion-based segmentation process. We have plotted the motion parameters: $A_{2,2}$ and b_1 of the car before and after occlusion in Fig. 9.

The moment the car disappears, we have no more motion measurements. We are nevertheless able to predict the value of each parameter using the dynamic system (12). We note that the prediction coincides with the motion parameter of the reappearing car, in Fig. 9. The dynamic system (12) is thus accurate enough to generate predictions during long time intervals. This experiment illustrates the efficiency of the motion filter to provide accurate motion parameters even in the case of complex motion

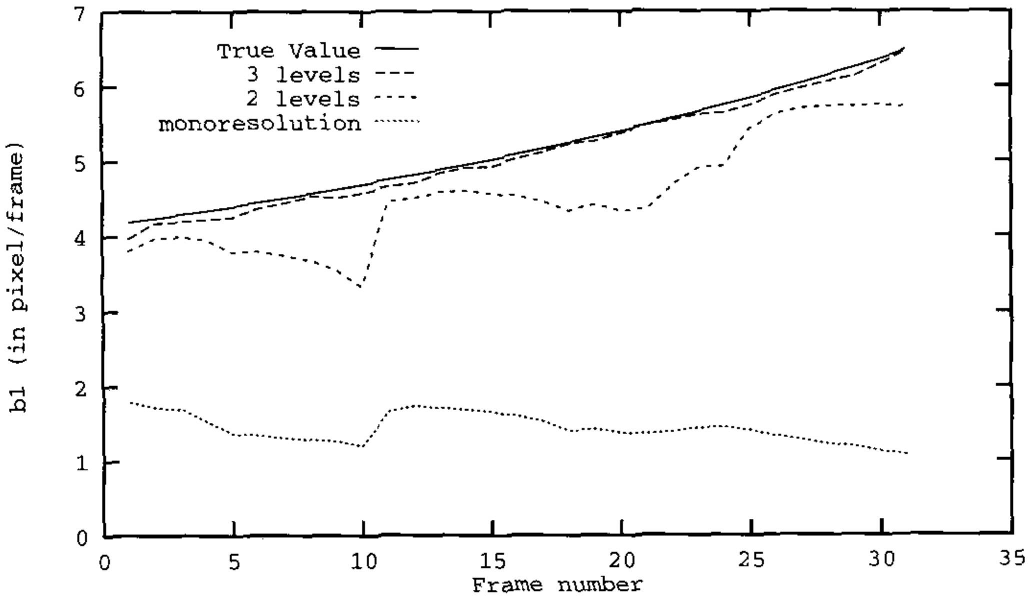


FIG. 6. Sequence poster: true value, multiresolution estimate for different pyramid levels, and monoresolution estimate of the motion parameter b_1.

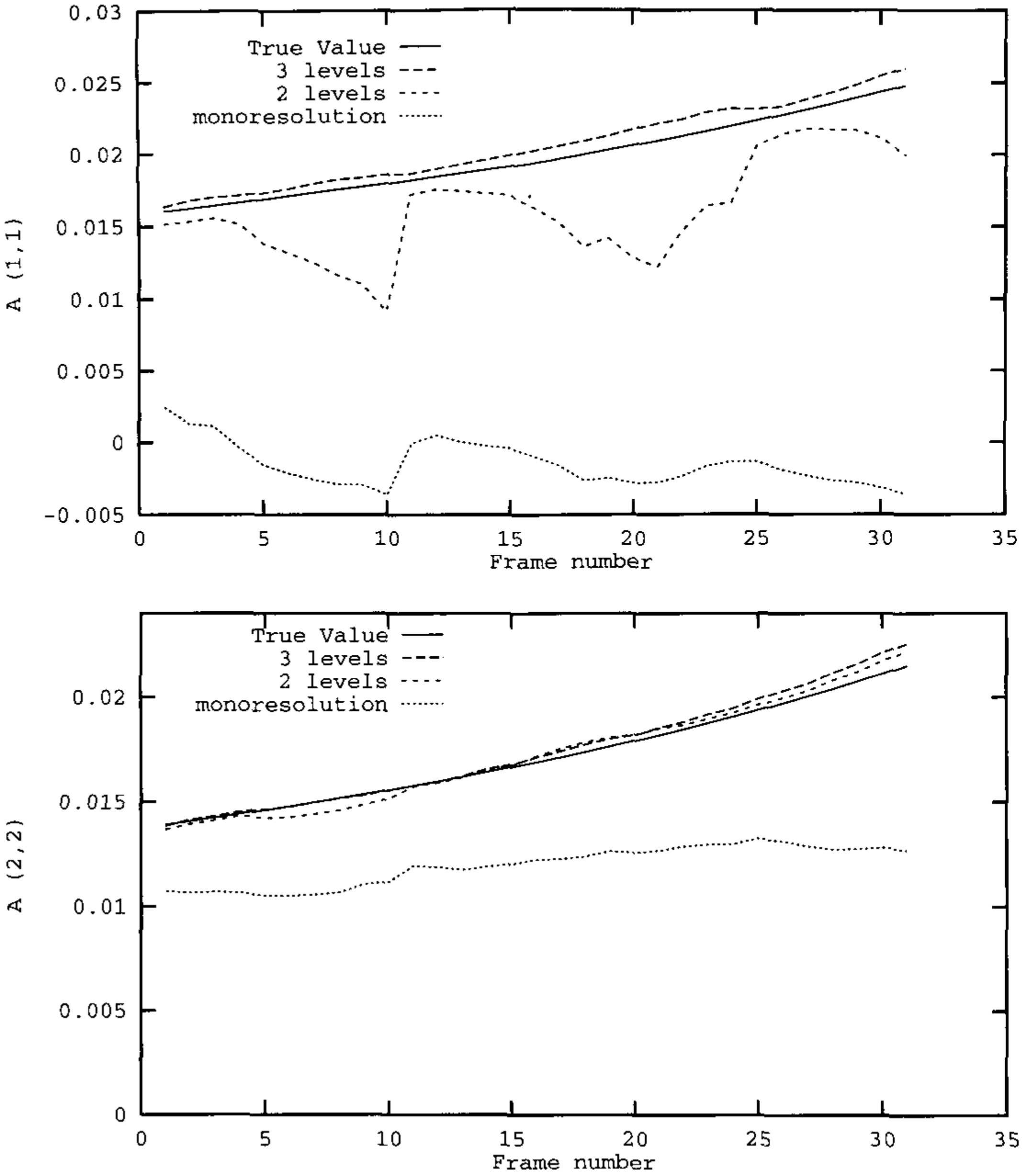
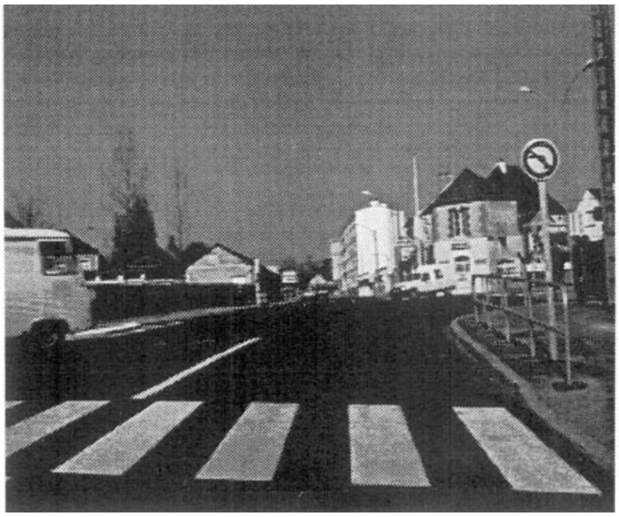


FIG. 7. Sequence poster: true values, multiresolution estimates for different pyramid levels, and monoresolution estimates, of the motion parameters $A_{1,1}$ (top) and $A_{2,2}$ (bottom).



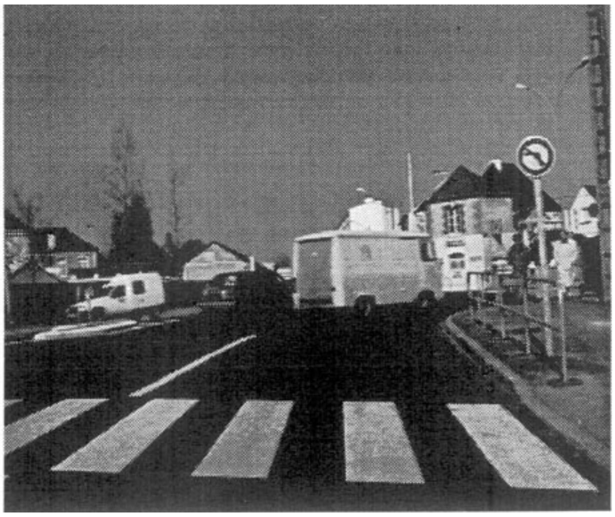


FIG. 8. Sequence crossroad at time t_3 (left) and at time t_{60} (right).

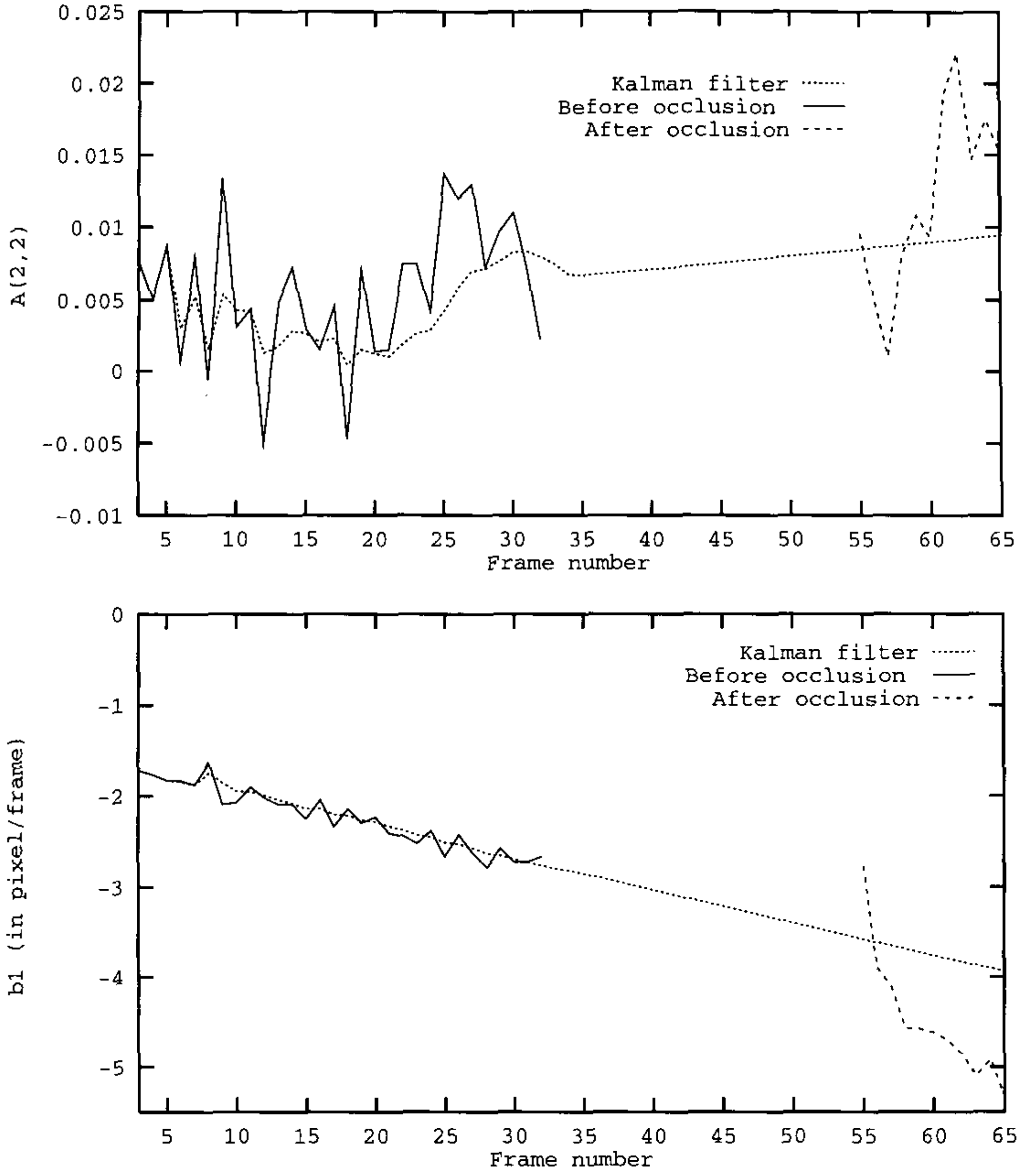


FIG. 9. Sequence crossroad: measurements before and after occlusion, and Kalman estimates of the motion parameters $A_{2,2}$ (top) and b_1 (bottom).

behavior, and in the presence of a complete and long occlusion. The idea of globally comparing the set of estimates of a given parameter, before and after occlusion, has been exploited in [28] to infer that the two pieces of trajectory (before and after occlusion) come from one and the same object in motion.

We present in the next section the geometric filter that generates recursive estimates of the shape and the position of a tracked region.

5. GEOMETRIC FILTER

First, we describe the region descriptor intended to represent the silhouette of a region all along the sequence. Then the equations of the geometric filter that characterizes the evolution of each tracked region are provided. Finally, the measurement algorithm that generates instantaneous observations of the position and shape of tracked regions is presented.

5.1. The Region Descriptor

We need a model to represent regions. The representation of a region is not intended to capture the exact boundary of the projection of the tracked target. It should give a description of the shape and location that supports the task of tracking in presence of partial occlusion. We choose to represent regions with some of its boundary

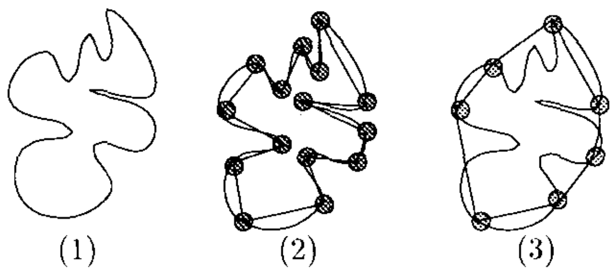


FIG. 10. The region descriptor: (1) region, (2) boundary points approximating the region, and (3) vertices of the convex hall.

points. This is achieved through a polygonal approximation of the region. The local features, corners, and points of high curvature are conserved by the approximation. A good approximation should be "close" to the original shape and have the minimum number of vertices. We use the approach developed by Wall and Danielsson in [38]. This method is simple and fast and gives good results on our data. This representation offers the property of being flexible enough to follow the deformations of the tracked silhouette. Furthermore this representation results in a compact description which decreases the amount of data required to represent the boundary, and it yields easily tractable models to describe the dynamic evolution of the region.

Our region tracking algorithm requires the matching of the prediction and an observation. The matching is achieved more easily when dealing with convex polygons. Among the boundary points approximating the silhouette of the region, we retain only those which are also the vertices of the convex hull of the considered set of points [31]. As shown in the results reported further, polygonal approximations do not restrict the type of objects nor the type of motion considered. As a matter of fact, using exactly the same approach, more complex models of shape could be used: splines, B-splines, superquadrics, etc. We can represent the region descriptor with a vector of dimension $2n$. This vector is the juxtaposition of the coordinates $[x_i, y_i]$ of the vertices of the convex hull of the polygonal approximation of the region: $[x_1, y_1, x_2, y_2, \dots, x_n, y_n]^T$. Figure 10 illustrates the definition of the region descriptor.

5.1.1. Automatic Control of the Adequacy of the Region Descriptor

We represent the tracked region with a constant number of points during successive time intervals of variable size. At the beginning of each time interval the region descriptor is initiated again with the current observation generated by the motion-based segmentation stage. Two reasons can lead to a new initiation of the region descriptor. First, if the attitude of the tracked target changes significantly, it can become necessary to update the number of vertices of the region descriptor in order to maintain a precise description of the silhouette of the target. Since the boundaries of regions generated by the motion-based segmentation are not very accurate, we only use a small number of vertices in the polygonal approximation of the region. Such vertices are usually located around the points of maximum curvature of the boundary. A large change in the attitude will thus require to update the number and the position of these vertices, and a new polygonal approximation of the region must be performed. Second, if the considered object is occluded when the region descriptor is initiated at the beginning of a time interval, and it becomes disoccluded, we need to enrich the region descriptor with the incoming information about the appearing silhouette of the object. In all other situations the number of vertices of the region descriptor remains constant over time. In particular, if the object becomes partially occluded the region descriptor still gives a complete view of the silhouette of the target, with the same number of vertices as before occlusion. We present now the testing procedure that has been developed to automatically detect occlusion and disocclusion.

5.1.2. Occlusion and Disocclusion Detection Algorithm

We explain here the principles of the occlusion and disocclusion detection algorithm. Necessary refinements of the region descriptor, after significant changes in the object attitude, can also be detected. We consider a region $R(t)$ that represents the projection of a viewed object in the image. Assuming that the divergence $div(\mathbf{v})$ of the motion field $\mathbf{v}$ is constant within the region, then the derivative of the area $\mathcal{A}(t)$ of $R(t)$, with respect to time, can be expressed as

$$\frac{\partial \mathcal{A}(t)}{\partial t} = div(\mathbf{v})\mathcal{A}(t).$$

This property results from Green's theorem and elementary calculus. Since we are using affine models of the motion field, the divergence $div(\mathbf{v}_{(\mathbf{A},\mathbf{b})})$ is constant within each tracked region, and it can be obtained from the parameters of the affine motion model. As a result, the expected variation of area of the tracked region can be compared to the measured variation of area. Let us define

$$\begin{aligned}\Delta\mathcal{A}_n = {} & \text{predicted area of } R \text{ at time } n \\ & - \text{measured area of } R \text{ at time } n.\end{aligned}$$

We assume that $(\Delta\mathcal{A}_n)$ is a sequence of zero-mean Gaussian white noise as long as the object is not occluded or disoccluded. We detect significative changes in the mean of $(\Delta\mathcal{A}_n)$. On one hand, each time a disocclusion is detected (we detect a significant decrease), the region descriptor is initiated again with the current observation obtained from the motion-based segmentation stage. On the other hand, when an important occlusion is detected (we detect a significant increase), we still keep the region descriptor, but we decide to discard the motion parameters estimated at the corresponding instant. Indeed, if a significative occlusion is detected the motion parameters will tend to be biased by the occlusion since there will not be enough points to perform a reliable least-squares estimation. For this reason, we prefer then to rely on the estimate generated by the Kalman filter, described in the previous section. However, as pointed out in the previous section, if the tracked object is only partially occluded the motion estimation scheme still delivers reliable motion estimates.

Since we are interested in both increase and decrease, we use a two-sided cumulative sum test (CUSUM) [4]. We run two CUSUM algorithms concurrently, and we compute at each instant the four quantities

$$s_n = \sum_{i=1}^{n} (\Delta\mathcal{A}_i - d) \quad \text{and} \quad m_n = \min_{1\le k\le n} s_k$$

$$S_n = \sum_{i=1}^{n} (\Delta\mathcal{A}_i + d) \quad \text{and} \quad M_n = \max_{1\le k\le n} S_k$$

and we accept the hypothesis that a change has occurred if

$$s_n - m_n > \lambda \quad \text{to detect a decrease}$$

$$M_n - S_n > \lambda \quad \text{to detect an increase,}$$

where d is the expected change magnitude, and λ is the threshold of the test. Since the decision rule integrates all the past information, this CUSUM algorithm is robust to instantaneous spurious variation of $(\Delta\mathcal{A}_n)$ due to noise, as illustrated in the following results. We present now results of experiments illustrating the behavior of the algorithm that controls the adequacy of the region descriptor.

The crossroad sequence. The sequence, presented in Section 4.5, illustrates the efficiency of the method to detect disocclusion situations and to incrementally enrich the region descriptor. The white van, coming from the left, is only partially visible in the beginning of the sequence. The black car is driving so closely behind the van that the segmentation is unable to split the two objects as illustrated in the right column of Fig. 17 at time t_{60}. Therefore the sequence illustrates a case of long disocclusion. The predicted area and the measured area of the global region, "van + black car," have been plotted in Fig. 11.

From frame 3 until frame 42 the global region is progressively appearing. The region descriptor is thus initiated every two or three frames with the measurement. The region descriptor is efficiently updated with the incoming information about the appearing region. From frame 43 until frame 66, the region van + black car is entirely in the image. Because the region is going away from the camera the apparent area of the objects reduces. We can note that there is a good correspondence between the predicted area, calculated with the affine parameters, and the measured area.

The parking sequence. We present here another experiment conducted on a sequence of real images of 75 frames (numbered 123–197) of size 256 × 256 pixels. A car is undergoing a 3-D rotation in the scene, as shown in Fig. 12. We can note that there is an important change in the attitude of the car, which is due to the large magnitude of the 3-D rotation. Because the camera is stationary,

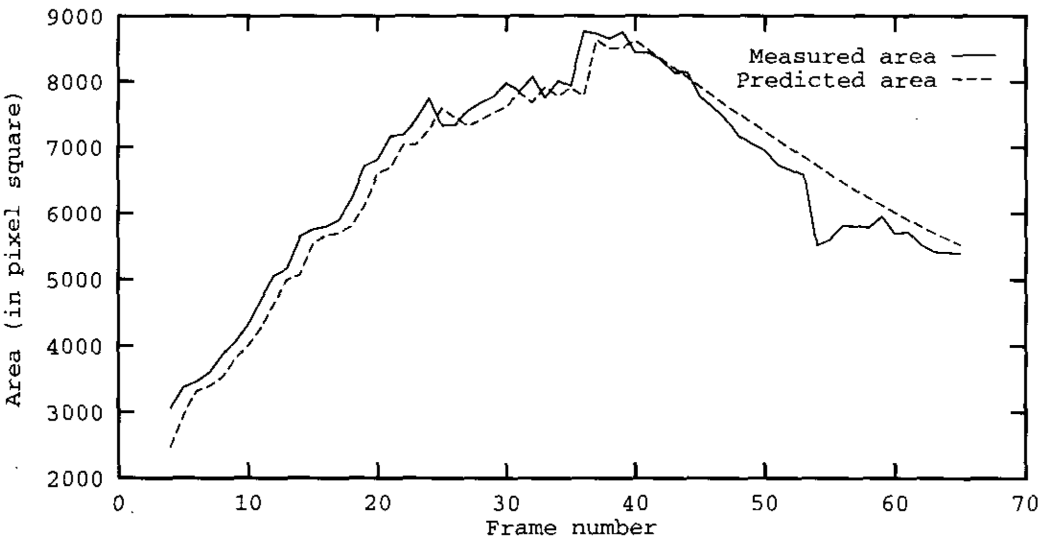


FIG. 11. Sequence crossroad: measured and predicted area of the "van + black car."

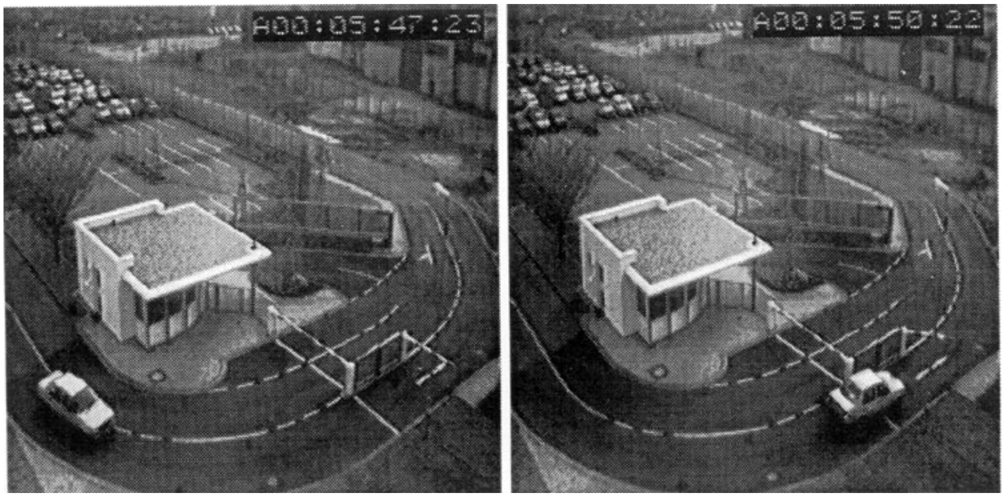


FIG. 12. Sequence parking at time t_{123} (left) and at time t_{197} (right).

the instantaneous position of the moving car was obtained with a motion detection algorithm [8]. The important change in the attitude of the car over the sequence requires that we update the number and the position of the vertices of the region descriptor. Our method achieves this task quite nicely as shown in Fig. 18. The predicted area and the measured area have been plotted with respect to frame numbers in Fig. 13.

From frame 124 until frame 161 the car is approaching and its attitude is changing significantly. The polygon that represents the region must be initiated again. The abrupt variations of area that correspond to new initiations of the region descriptor can be noted in Fig. 13. More vertices are added to take into account the important change in the attitude (the region descriptor is composed of six vertices at time 124 and of nine vertices at time 161.) In the second part of the sequence, the car is receding and the area of its projection in the image is decreasing. The number of vertices reduces (seven vertices at time 197). Furthermore we note that the test is insensible to spurious variations of area as illustrated in Fig. 13.

5.2. The Geometric Filter

We have presented the region evolution model in Section 3.2.1 and the region descriptor in the above section. We can now define the geometric filter which describes the evolution of the shape and the position of a tracked region. Assuming that the evolution of each vertex of the region descriptor is described by the dynamic system (1),

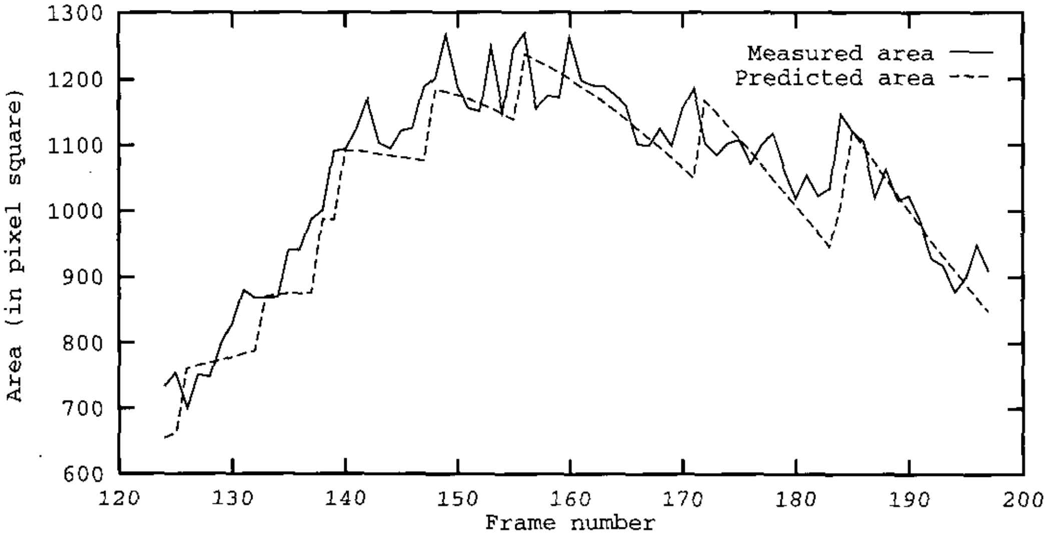


FIG. 13. Sequence parking: measured and predicted area of the car.

we obtain the following dynamic system for n vertices $(x_1, y_1), \ldots, (x_n, y_n)$ of the region descriptor,

$$\begin{bmatrix} x_1 \\ y_1 \\ \vdots \\ x_n \\ y_n \end{bmatrix} (t+1) = \begin{bmatrix} [\mathbf{\Phi}(t)] & & 0 \\ & \ddots & \\ 0 & & [\mathbf{\Phi}(t)] \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ \vdots \\ x_n \\ y_n \end{bmatrix} (t) + \begin{bmatrix} \mathbf{I}_2 \\ \vdots \\ \vdots \\ \mathbf{I}_2 \end{bmatrix} \mathbf{u}(t) + \begin{bmatrix} \zeta_1 \\ \vdots \\ \vdots \\ \zeta_n \end{bmatrix},$$

where $\mathbf{\Phi}(t)$ and $\mathbf{u}(t)$ have been defined in (1). $\zeta_i = [\zeta_i^x, \zeta_i^y]^T$ is a two-dimensional, zero-mean Gaussian noise vector. We choose a simplified model of the noise covariance matrix

$$cov(\zeta_1, \ldots, \zeta_n) = \sigma_\zeta^2 \mathbf{I}_{2n},$$

where $\mathbf{I}_{2n}$ is the $2n \times 2n$ identity matrix. This assumption enables us to break the filter of dimension $2n$ into n filters of dimension 2.

The matrix $\mathbf{\Phi}(t)$ and the vector $\mathbf{u}(t)$ account for the displacements of all the points within the region, between t and $t + 1$, and thus capture the global deformation of the region. Even though each vertex is tracked independently, they are actually coupled since the temporal evolution of each vertex is described by the same dynamic system. For each vertex, the measurement is given by the position of the vertex in the segmented image. The measurement process generates the measurement as explained in Section 5.3. Let $[x_i, y_i]^T(t)$, a vertex of the descriptor of the tracked region, be the state vector, and let $[\tilde{x}_i, \tilde{y}_i]^T(t)$ be the measurement vector, which contains the coordinates of the measured vertex. The following system describes the dynamic evolution of $[x_i, y_i]^T(t)$:

$$\begin{cases} \begin{bmatrix} x_i \\ y_i \end{bmatrix} (t+1) = \mathbf{\Phi}(t) \begin{bmatrix} x_i \\ y_i \end{bmatrix} (t) + \mathbf{u}(t) + \boldsymbol{\zeta}(t) \\ \begin{bmatrix} \tilde{x}_i \\ \tilde{y}_i \end{bmatrix} (t) = \begin{bmatrix} x_i \\ y_i \end{bmatrix} (t) + \boldsymbol{\eta}(t). \end{cases} \tag{13}$$

$\boldsymbol{\zeta}(t)$ and $\boldsymbol{\eta}(t)$ are two sequences of zero-mean Gaussian white noise vectors. $\mathbf{u}(t)$ is interpreted as a deterministic input. $\mathbf{\Phi}(t)$, the matrix of the affine transform, is the state transition matrix. Both are computed at each step with the motion filter. We estimate $[x_i, y_i]^T(t)$ with a standard Kalman filter. We use the first measurement as the initial estimate. Still, we have not defined precisely how the measurement vector is obtained. We present now the measurement vector and the algorithm used to generate this vector.

5.3. The Measurement Vector

We need a measurement of the tracked region, in each image, in order to update the prediction generated by the geometric filter. The measurement is derived from the observation (i.e., the region) obtained by the segmentation process. For a given region the measurement vector should depict this region with the same number of points as the region descriptor. As explained in Section 5.1.1, this number remains constant over a limited number of frames. The measurement algorithm consists of two stages, as shown in Fig. 14.

First, the association of observations and predictions is performed. We have to decide whether an observation given by the segmentation (a region) corresponds to a previously tracked object or to a new one. As long as there is no complete occlusion, the segmentation process assigns the same label over time to the same region. The correspondence between established trajectories and observations obtained by the segmentation is thus straightforward. If trajectories of regions cross each other, new labels corresponding to reappearing regions after occlusion will be created, while labels before occlusion will disappear. For each new appearing region a trajectory is initiated. Linking partial trajectories that comes from one and the same object in motion has been discussed in [28].

Then, for each observation-to-trajectory pairing, we globally register the predicted region and the observed region. This registration involves determining the transformation necessary to superimpose the prediction onto the observation. To perform this matching we represent the prediction and the observation with their region descriptor. Let us assume that the prediction is composed of n points, and that the observation obtained by the segmentation is represented by m points, (if the silhouette

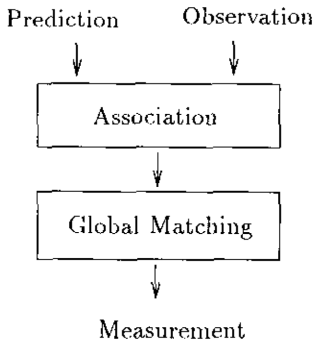


FIG. 14. Measurement algorithm.

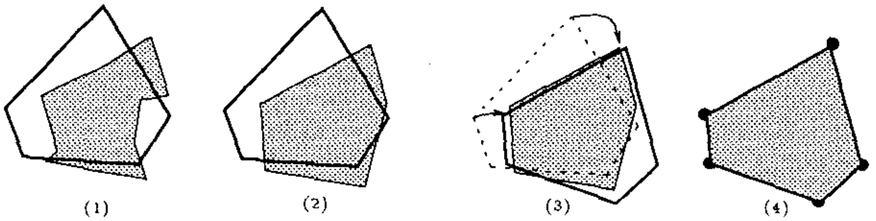


FIG. 15. The measurement algorithm: (1) observation obtained by the segmentation (gray region) and prediction (solid line), (2) convex hull of the observation, (3) matching of polygons, and (4) effective measurement: vertices of the gray region.

of the observation is occluded we have $m \leq n$). We will move the polygon corresponding to the prediction in order to globally match it with the observation composed of m points, (see Fig. 15).

After the registration is properly achieved, we are able to associate individually each point of the prediction with the corresponding point of the observation. Since the shape of the tracked region may change, the region descriptor of the prediction and the region descriptor of the observed region may have different number of points. We select the n vertices of the correctly superimposed shape of the prediction onto the observation, as the measurement vector (see Fig. 15). The measurement coincides indeed with the segmented region. Furthermore, if the object is partially occluded, the measurement still gives a complete virtual view of the silhouette of the region. This approach is global and does not require the usual matching of specific features which is often a difficult issue. The measurement is not sensitive to small uncertainties and spurious variations in the shape created by the motion-based segmentation. The measurement method is well suited for temporal tracking since it delivers steady measurements of the shape and the position of the tracked region.

It should be emphasized that the role of the region descriptor is only to provide a tractable description of the region. In our approach, the information about the trajectory of the region is mainly captured by the sequence of affine transforms $(\mathbf{\Phi}, \mathbf{u})(t)$. Consequently the changes in the number and the location of the vertices of the region descriptor, which are necessary to provide an adequate description of the region, have no effect on the smoothness of the trajectory. Rather than relying on the set of trajectories of each vertex of the region descriptor, the region tracking is mainly based on the estimation of the dense visual information captured by the affine transform $(\mathbf{\Phi}, \mathbf{u})(t)$.

We discuss now the algorithm that establishes the global matching between the observation and the prediction.

5.3.1. *Global Registration Algorithm*

If we represent the silhouette obtained by the segmentation and the predicted region with their region descriptor, the problem of superimposing the shape of the prediction onto the observation reduces to the problem of matching two convex polygons with possibly different number of vertices. Matching is achieved here by moving a polygon and finding the best translation and rotation to superimpose it on the other one. We did not include scaling in the transformation; otherwise in the case of occlusion the minimization process would scale the prediction to achieve a best matching with the partially occluded observation. Most of the works that follow this approach define a distance on the space of shapes (Haussdorf distance, Minkowski distance, Frechet distance, etc.) and seek the geometrical transformation that minimizes the distance between the two shapes. We use the measure proposed by Cox *et al.* in [12]. Let P_l and P_2 be the two polygons, and let us suppose that we are moving polygon P_2. We are looking for transform T (where T is the composition of a rotation and a translation) that minimizes the function $\mathcal{F}$,

$$\mathcal{F}(T) = \mathcal{D}(P_1, T(P_2)) = \sum_{M_1 \in P_1} d(M_1, T(P_2))^2 + \sum_{M_2 \in P_2} d(T(M_2), P_1)^2, \tag{14}$$

where $d(M, P)$ is the Euclidean distance of a point M to the polygon P. The minimum of $\mathcal{F}$ generally occurs for positions of polygons that a human observer would have intuitively imagined. The advantage of this measure is that it has interesting properties regarding the problem of minimization. The function $\mathcal{F}$ is continuous and differentiable. It is also convex with respect to the two parameters of the translation. Unfortunately the function is not convex with respect to the rotation parameter. A preconditioned quasi-Newton conjugate gradient method is used to solve the minimization problem with respect to the three parameters. We noted during the experiments that the affine transform actually generates very accurate predictions. The magnitude of the transformation T is thus always quite small.

In order to validate the tracking algorithm we have conducted numerous experiments on sequences of real images. We present in the next section three significative experiments.

6. RESULTS

The tracking algorithm has been tested on a large number of synthetic and real images. We present three experiments, performed on sequences of real images, that best illustrate the method. We first present two experiments that involve situations of long occlusion. We show that we are able to predict the position of the reappearing objects after the occlusion, even though the occlusion is quite long.

6.1. The Breakfast Sequence

The sequence *breakfast* consists of 44 real images of size 256 × 352 pixels. It has been acquired with an experimental cell constituted by a camera mounted on the end effector of a 6 d.o.f. robot. The camera undergoes small random vibrations from left to right. Observations obtained by the motion-based segmentation are thus noisy. Each segmented region includes the moving object and its reflection in the table (since the reflection is moving along with the object), as illustrated in the right column of Fig. 16.

Polygons representing each tracked object are superimposed onto the original images. Two boxes are manually moved between each frame, as shown in the left column of Fig. 16. Each box undergoes a translation with constant velocity in the scene. The interframe displacement is large, 2 pixels per frame at the beginning and 9 pixels per frame at the end of the sequence. While boxes disappear behind the cereal pack, their trajectories cross. When each box is only partially occluded the motion parameters can still be obtained, as explained in Section 4.3. Moreover, thanks to the measurement definition, the region descriptor still provides the complete view of the tracked object. The ability to maintain an accurate tracking in the case of partial occlusion is a significant advantage of the pursuit algorithm over standard tracking methods. When each box reappears it is assigned a new label by the segmentation algorithm as shown in the right of Fig. 16, at time t_{27}. We can observe the efficiency of the affine model to accurately predict the location of the reappearing object. We note indeed in the left of Fig. 16 at time t_{27} that each newly reappeared region is located within the polygon of the predicted estimate, obtained from the measurements before occlusion. The results show the efficiency of the approach to track multiple regions in an image sequence, with long occlusions.

6.2. The Crossroad Sequence

The sequence has been presented in Section 4.5. The polygons representing the tracked regions are superimposed onto the original images at time t_3, t_{20}, and t_{60} in the left column of Fig. 17. The corresponding segmented regions at the same instants are presented in the right column of Fig. 17. The algorithm accurately tracks the car, even when it is partially occluded, until it completely disappears, as shown in the left column of Fig. 17 at time t_3 and t_{20}.

When the car is completely behind the van, the motion filter generates only predictions. The geometric filter itself relies on these predictions to predict the shape and position of the region until the car reappears. Even though the car is occluded during 23 frames, the prediction is not late when the car reappears. The prediction coincides with the reappearing car, as illustrated in Fig. 17 at time t_{60}. It is important to note that the projection in the image of the velocity of the white car is not constant over the sequence, as shown in Fig. 9. We have processed the same sequence with a first version of the region-based tracking algorithm based on a constant velocity model. We noted that the prediction was very late when the car pops up. We have also tried constant acceleration models, but the resulting filter is extremely unstable. After a short period of time without measurements the filter diverges. More complex 2-D motion models were to be obtained if an efficient and reliable tracking was expected. Affine motion models provide such efficient 2-D motion models that can capture complex 3-D motion. It should be pointed out that, in the results presented here, we did not initiate a new trajectory with the reappearing region. We have only superimposed the prediction generated with the tracking filter. This association between partial trajectories has been investigated in [28]. This example illustrates the good performance of the region-based tracking in the presence of a long and total occlusion.

6.3. The Parking Sequence

Finally, we present a last experiment that illustrates the good performance of the affine motion model to track an object which is undergoing a large 3-D rotation in the scene. The affine motion model can indeed describe complex 3-D motions. The sequence has been presented in Section 5.1.2. The polygons representing the tracked region are superimposed ontó the original images at time t_{124}, t_{161}, and t_{197} in the left column of Fig. 18. Because the camera is stationary, the instantaneous position of the moving car was obtained with a motion detection algorithm [8]. Regions detected by the motion detection algorithm are displayed in the right part of Fig. 18. The shadow of the car is included in the tracked region as shown in Fig. 18 at time t_{197}. We note the important change in the attitude of the car in Fig. 18 at different instants: t_{123}, t_{161}, and t_{197}. The sequence also illustrates the efficiency of the procedure that automatically updates the region descriptor. The trajectory of the car is represented in Fig. 19. For the clarity of the figure, the tracked car has only been represented every 10 frames.

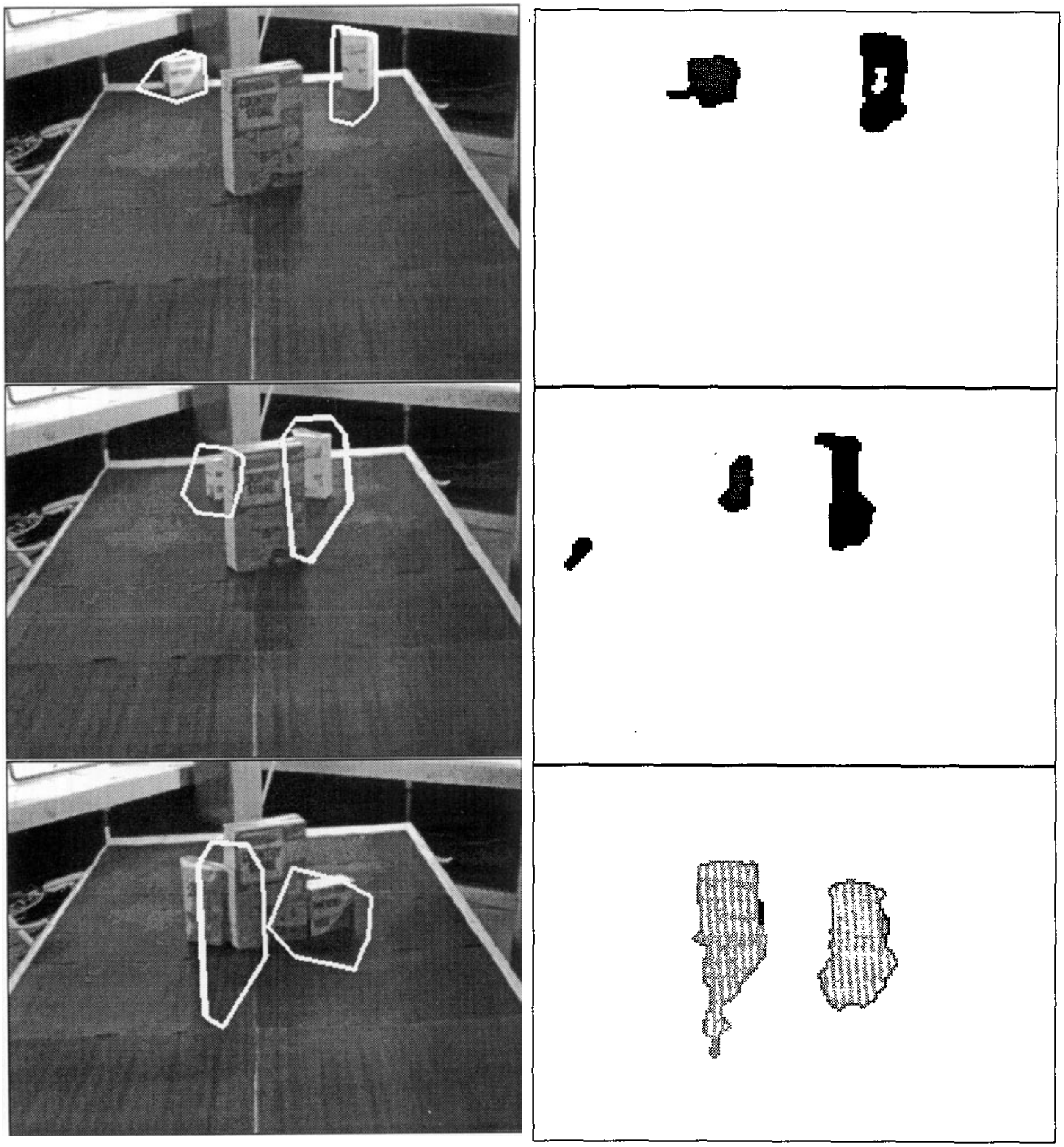


FIG. 16. Sequence breakfast, from top to bottom: tracked regions superimposed on original images (left) and segmented images (right), at time t_1, t_{11}, t_{27}.

7. CONCLUSION AND FUTURE RESEARCH

This work has investigated a new approach to the tracking of regions in an image sequence. The approach relies on two successive operations: detection and discrimination of moving targets and then pursuit of the targets. A motion-based segmentation algorithm, previously developed in the laboratory, provides the detection and discrimination stage. This paper has focused on the pursuit stage. A pursuit algorithm has been developed that directly tracks the region representing the projection in the image of a moving object. Rather than relying on a set of trajectories of individual points or segments, the region tracking is based on the dense estimation of an affine model of the motion field within the region. This parameterized motion field yields an affine transform that

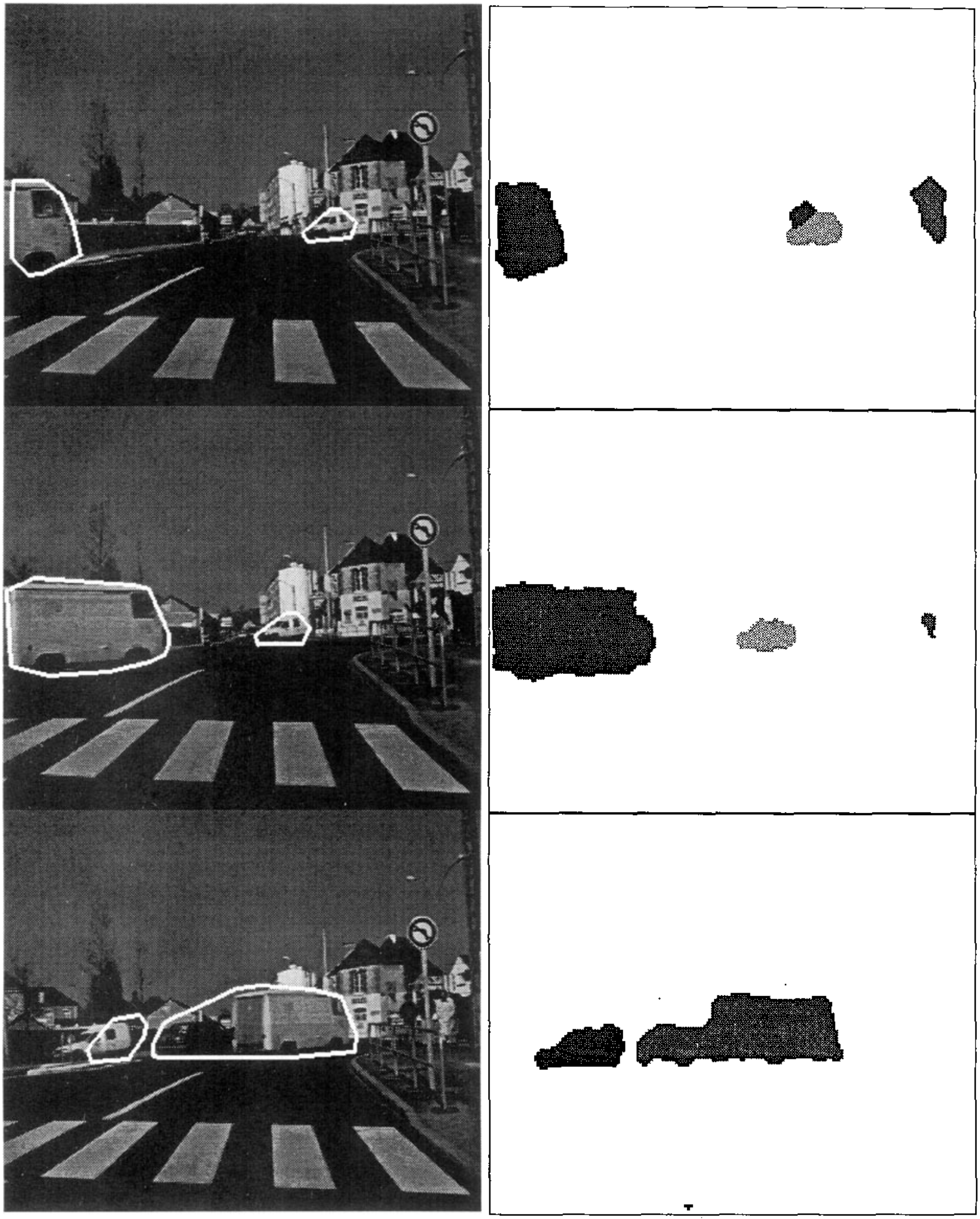


FIG. 17. Sequence crossroad, from top to bottom: tracked regions superimposed on original images (left) and segmented images (right), at time t_3, t_{20}, t_{60}.

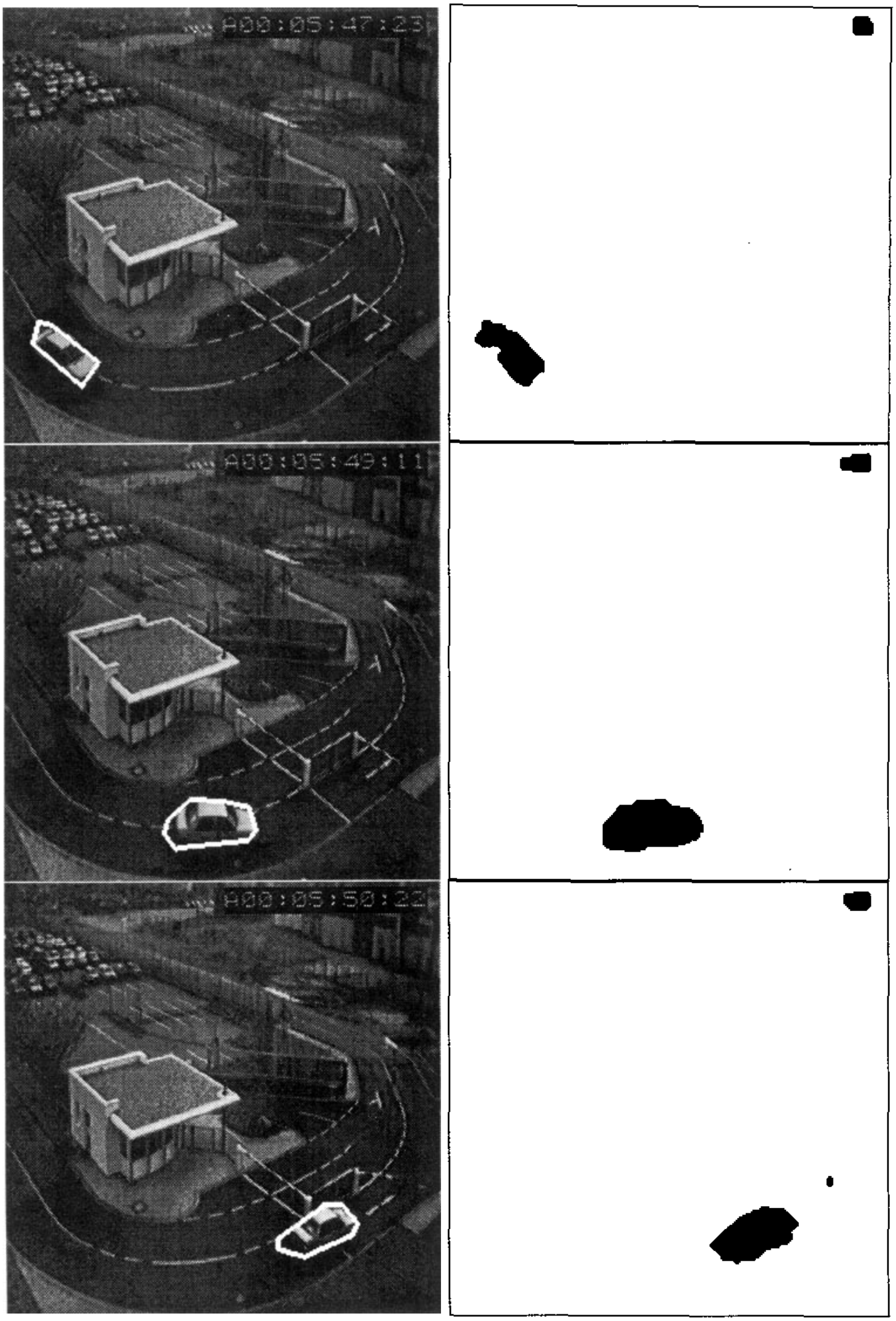


FIG. 18. Sequence parking, from top to bottom: tracked regions superimposed on original images (left) and segmented images (right), at time t_{123}, t_{161}, t_{197}.

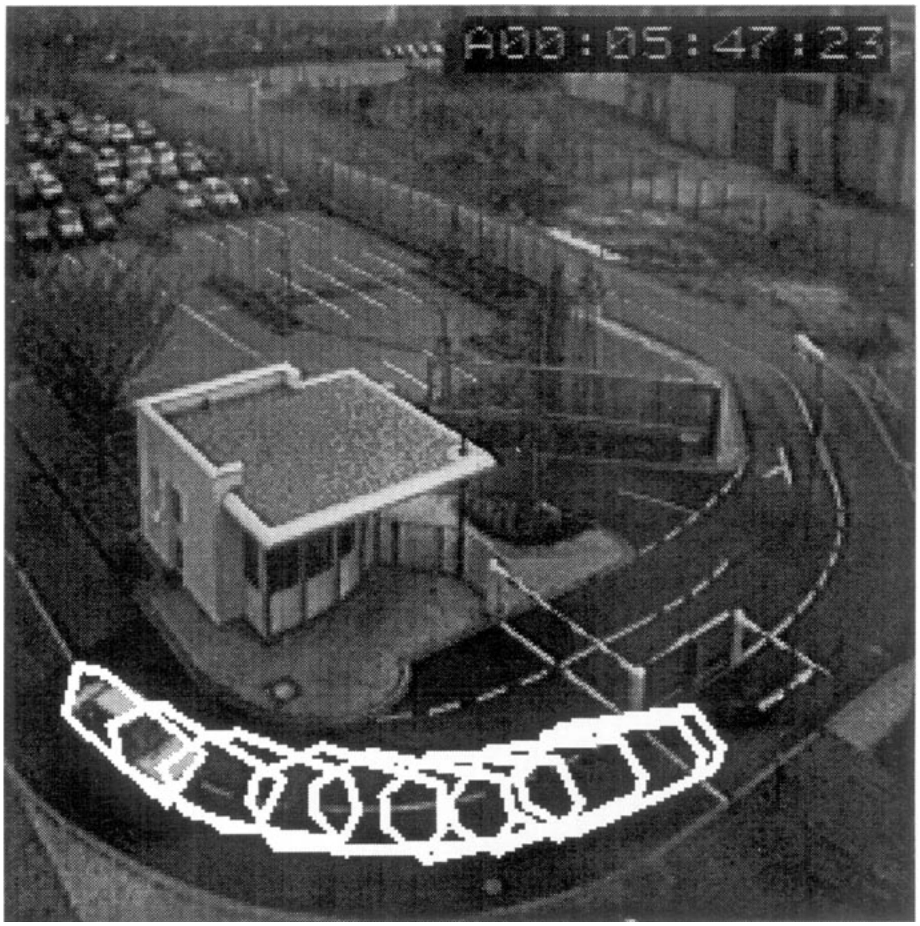


FIG. 19. Sequence parking. Tracked car superimposed on the same original image every 10 frames, from time t_{123} until time t_{197}.

makes it possible to predict the position of the region in the next frame. A linear dynamic system based on this affine transform describes the temporal evolution of the tracked target. The estimation of the parameters of the affine model of the motion field is performed directly from two successive frames of the image sequence, with a multiresolution approach. The method provides reliable estimates of the motion parameters, even in the case of large displacements. Another linear dynamic system characterizes the temporal evolution of the motion parameters. Standard Kalman filtering techniques generate recursive estimates of the geometry and the motion of the projections of the targets. Experiments conducted on real images have demonstrated that the approach is robust against partial occlusion and can handle large interframe displacements, as well as complex 3-D motion. The tracking algorithm delivers spatiotemporal trajectories in the (x, y, t) domain of objects moving in the scene.

This work has emphasized the maintenance of the tracking during the pursuit stage. However, if an object completely disappears behind another moving object, (as in the *crossroad* sequence), the tracking procedure will not interpret the two objects, before and after occlusion, as one and the same object. Such situations of complete occlusion and disocclusion or trajectories crossing are frequent in real outdoors scenes. In order to properly deal with these situations, we have investigated in [28] a new method to link partial spatiotemporal trajectories that can be generated by the tracking algorithm presented in this paper.

A necessary future work involves now the interpretation of the spatiotemporal trajectories obtained in the (x, y, t) domain, by our method, in terms of spatial trajectories in the (x, y, z) domain. All the information captured by the affine model of the motion field will be used to this end.

APPENDIX: NOMENCLATURE

- I the intensity function,
- R a region present in the image,
- $\begin{pmatrix} x \\ y \end{pmatrix}(t)$ a point in region R,

- $\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix}(t)$ the 2-D motion field vector at location $(x, y)^T$,
- $\mathbf{v}(\cdot)$ the 2-D motion field,
- $(\mathbf{\Phi}, \mathbf{u})(t)$ the affine transformation,
- $(\mathbf{A}, \mathbf{b})(t)$ the parameters of the affine model of the 2-D motion field,
- (x_g, y_g) the centroid of the tracked region,
- $(\mathbf{A}_g, \mathbf{b}_g)(t)$ the affine model of the motion field centered at (x_g, y_g),
- δt the time step between two successive frames,
- $\psi_{(\mathbf{A},\mathbf{b})}(x, y)$ a random variable that measures the adequacy of the motion model,
- $\nabla \mathbf{I}(\cdot)$ the spatial gradient of the intensity function,
- $I_t(\cdot)$ the partial derivative with respect to time of the intensity function,
- $\mathbf{v}_{(\mathbf{a},\mathbf{b})}(\cdot)$ the affine model of the 2-D motion field,
- L the maximum level of the pyramid,
- $\mathbf{p}^l$ a point within a region at given level l in the pyramid,
- $(\mathbf{A}, \mathbf{b})^l$ the parameters of the affine model of the motion field,
- $\mathbf{v}_{(\mathbf{A},\mathbf{b})^l}(\mathbf{p}^l)$ the affine model of the motion field at location $\mathbf{p}^l$,
- $\delta\hat{\mathbf{p}}^{l+1}$ the displacement estimated, at level $l + 1$, at location $\mathbf{p}^{l+1}$,
- $\delta^2\mathbf{p}^l$ *the incremental estimate to be computed at level* l,
- $(\widehat{\mathbf{A}, \mathbf{b}})^{l+1}$ the estimate of $(\mathbf{A}, \mathbf{b})^{l+1}$ obtained at level $l + 1$,
- $\acute{\mathbf{A}}^{l+1}$ the estimate of $\mathbf{A}^{l+1}$ obtained at level $l + 1$,
- $\hat{\mathbf{b}}^{l+1}$ the estimate of $\mathbf{b}^{l+1}$ obtained at level $l + 1$,
- $\Delta\mathbf{A}^l$, $\Delta\mathbf{b}^l$ the refinement of the motion parameters to be estimated at level l,
- p^l a point of level l in the pyramid, in the one-dimensional case,
- $\widetilde{\Delta p}^l$ the displacement at location p^l,
- $\nabla I(p^l)$ the slope of the intensity function at location p^l, in the one-dimensional case,
- $\delta\hat{p}^{l+1}$ the displacement estimated at location p^{l+1}, in the one-dimensional case,
- δp^l the displacement at location p^l, in the one-dimensional case,
- $a_i(t)$, $i = 1, \ldots, 6$, one of the component of $(\widehat{\mathbf{A}, \mathbf{b}})(t)$,
- $\tilde{a}_i(t)$ the instantaneous measurement of $a_i(t)$,
- $\beta(t)$ a sequence of zero-mean Gaussian white noise,
- σ_β^2 the variance of $\beta(t)$,
- $[\varepsilon_1, \varepsilon_2]^T(t)$ a two-dimensional zero-mean Gaussian noise,
- $\mathbf{R}$ the covariance matrix of $[\varepsilon_1, \varepsilon_2]^T(t)$,
- $\sigma_{\mathbf{R}}^2$ a common scalar factor in $\mathbf{R}$,
- U, V, W the 3-D velocity of the camera,
- $A_{1,1}$, $A_{2,1}$, $A_{1,2}$, $A_{2,2}$ the components of $\mathbf{A}$,
- b_1, b_2 the components of $\mathbf{b}$,
- $[x_i, y_i]$ a vertex of the convex hull of the polygonal approximation of the region,.
- $[x_1, y_1, x_2, y_2, \ldots, x_n, y_n]^T$ the region descriptor,
- $R(t)$ a region that represents the projection of a viewed object in the image,
- $div(\mathbf{v})$ the divergence of the motion field $\mathbf{v}$ in $R(t)$,
- $\mathcal{A}(t)$ the area of $R(t)$,
- $(\partial\mathcal{A}/\partial t)(t)$ the partial derivative of $\mathcal{A}(t)$ with respect to time,
- $\Delta\mathcal{A}_n$ the difference between the predicted area of R at time n and the measured area of R at time n,
- *d the expected change magnitude in the CUSUM test,*
- λ the threshold of the CUSUM test,
- s_n a sum in the CUSUM test,
- S_n a sum in the CUSUM test,
- m_n a variable in the CUSUM test,
- M_n a variable in the CUSUM test,
- $\boldsymbol{\zeta}_i = [\zeta_i^x, \zeta_i^y]^T$ a two-dimensional zero-mean Gaussian noise,
- $cov(\boldsymbol{\zeta}_1, \ldots, \boldsymbol{\zeta}_n)$ the covariance matrix of $[\boldsymbol{\zeta}_1, \ldots, \boldsymbol{\zeta}_n]^T$,
- σ_ζ^2 a common scalar factor in $cov(\boldsymbol{\zeta}_1, \ldots, \boldsymbol{\zeta}_n)$,
- $\mathbf{I}_{2n}$ the $2n \times 2n$ identity matrix,
- $\boldsymbol{\eta}(t)$ a sequence of zero-mean Gaussian white noise vectors,
- T a composition of a 2-D rotation and a 2-D translation,
- $\mathcal{F}$ the function that expresses the distance between two polygons,
- P_1, P_2 two polygons,
- M_1 a vertex of P_1,
- M_2 a vertex of P_2,
- $d(M, P)$ the Euclidean distance of a point M to the polygon P.

ACKNOWLEDGMENTS

We gratefully acknowledge Professor Hans-Helmut Nagel and Dr. Dieter Koller for providing the image sequence *parking* presented in Figs. 18 and 19, and Henner Kolnig for giving access to the sequence. This work was supported in part by MRT (French Ministry of Research and Technology) in the context of the EUREKA European project PROMETHEUS, under PSA-Contract VY/85241753/14–Z10, and in the context of the French project Orasis within the Gdr-Prc CHM (Man Machine Interface). François G. Meyer was supported by a C.N.R.S. and Région Bretagne fellowship.

REFERENCES

1. J. Aloimonos, I. Weiss, and A. Bandopadhay, Active vision, *Int. J. Comput. Vision* **1**(4), 1988, 333–356.
2. A. A. Amini and J. S. Duncan, Bending and stretching models for LV wall motion analysis from curves and surfaces, *Image Vision Comput.* **10**(6), July/Aug. 1992, 418–430.
3. P. Anandan, A computational framework and an algorithm for the measurement of visual motion, *Int. J. Comput. Vision* **2**, 1989, 283–310.
4. M. Basseville and I. V. Nikiforov, *Detection of Abrupt Changes: Theory and Application*, System Sciences Series Prentice–Hall, Englewood Cliffs, NJ, 1993.

5. R. Battiti, E. Amaldi, and C. Koch, Computing optical flow across multiple scales: An adaptative coarse-to-fine strategy, *Int. J. Comput. Vision* **6**(2), 1991, 133–145.

6. J. R. Bergen, P. J. Burt, R. Hingorani, and S. Peleg, A three-frame algorithm for estimating two-component image motion, *IEEE Trans. PAMI* **PAMI14**(9), 1992, 886–896.

7. P. Bouthemy and E. François, Motion segmentation and qualitative dynamic scene analysis from an image sequence, *Int. J. Comput. Vision* **10**(2), 1993, 157–182.

8. P. Bouthemy and P. Lalande, Recovery of moving object masks in an image sequence using local spatio-temporal contextual information, *Opt. Eng.* **June** 1993.

9. C. Brown, Gaze controls cooperating through prediction, *Image Vision Comput.* **8**(1), 1990, 10–17.

10. P. J. Burt, The pyramid as a structure for efficient computation, in *Multiresolution Image Processing and Analysis* (A. Rosenfeld, Ed.), pp. 6–35, Springer–Verlag, Berlin/New York, 1984.

11. P. J. Burt, J. R. Bergen, R. Hingorani, R. Kolczynski, W. A. Lee, A. Leung, J. Lubin, and H. Shvaytser, Object tracking with a moving camera, in *IEEE Workshop on Visual Motion, Irvine, CA, March 1989*, pp. 2–12.

12. P. Cox, H. Maitre, M. Minoux, and C. Ribeiro, Optimal matching of convex polygons, *Pattern Recognit. Lett.* **9**(5), 1989, 327–334.

13. J. L. Crowley, P. Stelmaszyk, T. Skordas, and P. Puget, Measurement and integration of 3D structures by tracking edge lines, *Int. J. Comput. Vision* **8**(1), 1992, 29–52.

14. R. Deriche and O. Faugeras, Tracking line segments, *Image Vision Comput.* **8**(4), 1990, 261–270.

15. W. Enkelmann, Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences, *Comput. Vision Graphics Image Process.* **43**, 1988, 150–177.

16. C. S. Fuh and P. Maragos, Affine models for motion and shape recovery, in *Proceedings of Visual Communications and Image Processing '92, Boston, MA*,**SPIE 1818,** pp. 120–134.

17. D. B. Gennery, Visual tracking of known three-dimensional objects, *Int. J. Comput. Vision* **7**(3), 1992, 243–270.

18. B. K. P. Horn and B. G. Schunck, Determining optical flow, *Artif. Intell.* **17**, 1981, 185–203.

19. D. Koller, K. Daniilidis, T. Thórhallson, and H. H. Nagel, Model-based object tracking in traffic scenes, In *Proceedings of Second European Conference on Computer Vision, ECCV-92, Santa Margherita, Italy* (G. Sandini, ed.) pp. 437–452, Springer–Verlag, Berlin/New York, 1992.

20. F. Leymarie and M. D. Levine, Tracking deformable objects in the plane using an active contour model, *IEEE Trans. Pattern Anal. Mach. Intell.* June 1993, 617–634.

21. D. G. Lowe, Robust model-based motion tracking through the integration of search and estimation, *Int. J. Comput. Vision* **8**(2), 1992, 113–122.

22. B. D. Lucas and T. Kanade, An iterative image registration technique with an application to stereo vision, in *Proceedings of 7th IJCAI, Vancouver, 1981*, pp. 674–679.

23. D. Marr, *Vision, A Computational Investigation into the Human Representation and Processing of Visual Information*, Freeman, San Francisco, 1982.

24. J. S. Meditch, *Stochastic Optimal Linear Estimation and Control*, McGraw–Hill, New York, 1969,.

25. F. Meyer and P. Bouthemy, Estimation of time-to-collision maps from first-order motion models and normal flows, in *Proceedings 11th International Conference on Pattern Recognition, The Hague, 1992*, pp. 78–82.

26. F. Meyer and P. Bouthemy, Region-based tracking in an image sequence, in *Proceedings of ECCV-92, Italy* (G. Sandini, ed.), pp. 476–484, Springer–Verlag, Berlin/New York, 1992.

27. F. Meyer and P. Bouthemy, Region-based tracking in an image sequence, research report, Technical report 1723, INRIA, France, 1992.

28. F. G. Meyer and P. Bouthemy, Exploiting the temporal coherence of motion for linking partial spatiotemporal trajectories, in *Proceedings of CVPR '93, New York, 1993*, pp. 746–747.

29. S. Negahdaripour and S. Lee, Motion recovery from image sequences using only first-order optical flow information, *Int. J. Comput. Vision* **9**(3), 1992, 163–184.

30. S. Peleg and H. Rom, Motion-based segmentation, in *Proceedings 10th IEEE Conference on Pattern Recognition, Atlantic City, 1990*, pp. 109–113.

31. F. P. Preparata and M. I. Shamos, *Computational Geometry An Introduction*, Springer-Verlag, Berlin/New York, 1985.

32. K. Rangarajan and M. Shah, Establishing motion correspondence, *CVGIP: Image Understand.* **54**(1), 1991, 56–73.

33. R. J. Schalkoff and E. S. McVey, A model and tracking algorithm for a class of video targets, *IEEE Trans. PAMI* **PAMI-4**(1), 1982, 2–10.

34. J. Schick and E. D. Dickmanns, Simultaneous estimation of 3D shape and motion of objects by computer vision, *Proceedings of the IEEE Workshop on Visual Motion, Princeton, NJ, October 1991*, pp. 256–261.

35. B. G. Schunck, Robust computational vision, Technical report CSE-TR-60-90, The University of Michigan, May 1990.

36. I. K. Sethi and R. Jain, Finding trajectories of feature points in a monocular image sequence, *IEEE Trans. PAMI* **PAMI-9**(1), 1987, 56–73.

37. D. Terzopoulos, A. Witkin, and M. Kass, Constraints on deformable models: Recovering 3D shape and nonrigid motion, *Artif. Intell.* **36,** 1988, 91–123.

38. K. Wall and P. E. Danielsson, A fast sequential method for polygonal approximation of digitized curves, *Comput. Vision Graphics Image Process.* **28,** 1984, 220–227.

39. R. H. Wurtz, H. Komatsu, M. R. Dürsteler, and D. S. Yamasaki, Motion to movement: Cerebral cortical visual processing for pursuit eye movements, in Signal and Sense: Local and Global Order in Perceptual Maps (G. Edelman, W. E. Gall, and W. M. Cowan, Eds.), pp. 233–260, Wiley, New York, 1990.

40. Z. Zhang and O. D. Faugeras, Three-dimensional motion computation and object segmentation in a long sequence of stereo frames, *Int. J. Comput. Vision* **7**(3), 1992, 211–241.