INRIA

UNITÉ DE RECHERCHE
INRIA-RENNES

Rapports de Recherche

1992

25ème anniversaire

N° 1723

Programme 4
Robotique, Image et Vision

REGION-BASED TRACKING IN AN IMAGE SEQUENCE

François MEYER
Patrick BOUTHEMY

Juillet 1992

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

IRISA

INSTITUT DE RECHERCHE EN INFORMATIQUE
ET SYSTEMES ALEATOIRES

Campus Universitaire de Beaulieu
35042 - RENNES CEDEX FRANCE
Tél. : 99 84 71 00 - Télex : UNIRISA 950 473 F
Télécopie : 99 38 38 32

Region-Based Tracking in an Image Sequence *

Suivi de primitives régions dans une séquence d'images

Programme 4
Publication Interne n° 664 - Juin 1992 - 50 pages

François Meyer[†], Patrick Bouthemy
IRISA/INRIA, Campus Universitaire de Beaulieu,
35042 Rennes Cedex, France

Abstract

This paper addresses the problem of motion tracking in a sequence of monocular images. We want to establish and maintain the successive positions of objects in a sequence of images. The use of regions as primitives for tracking enables us to directly handle consistent object-level entities. On one hand, a motion-based segmentation process based on normal flows and first order motion models provides us with instantaneous measurements of the geometry of each region present in the segmented images. Shape and position of each projected object are estimated with a recursive algorithm along the sequence. On the other hand, a motion filter based on a multiresolution estimation scheme and temporal filtering generates reliable estimates of the motion parameters of each region. The proposed approach relies on adequate modeling and measurement of the geometry and kinematics of object projections. In particular no 3-D information is required. It realizes a good trade-off between tractability and efficiency. Occlusion situations can be handled. We have carried out experiments on both sequences of synthetic and real images depicting complex outdoor scenes to illustrate the performance of this new method.

[*] This work is supported by MRT (French Ministry of Research and Technology) in the context of the EUREKA European project PROMETHEUS, under PSA-contract VY/85241753/14/Z10

[†] Supported by a student grant of C.N.R.S. and Brittany County Council

Résumé

Nous nous intéressons ici au problème du suivi d'objets mobiles dans une séquence d'images monoculaires. Nous voulons suivre les positions successives des objets au cours de la séquence. Le suivi s'effectue directement sur les régions, qui sont les projections des objets en mouvement dans la scène. D'une part, un algorithme de segmentation au sens du mouvement, fondé sur les dérivées spatio-temporelles de la fonction intensité et un modèle affine du champ des vitesses, nous permet d'obtenir des valeurs instantanées des paramètres géométrique de chaque région de l'image de segmentation. La forme et la position de ces régions sont estimées récursivement au cours du temps. D'autre part, un filtre cinématique utilisant une estimation multirésolution des paramètres du mouvement apparent 2-D et un filtrage temporel, fournit des valeurs fiables des paramètres cinématiques de chaque région. L'approche proposée s'appuie sur une modélisation appropriée de la géométrie et de la cinématique des projections des objets dans l'image. En particulier l'approche ne requiert pas d'information 3-D. Les situations d'occlusion peuvent être prises en compte. Nous avons mené des expériences à la fois sur des séquences synthétiques et sur des séquences réelles, représentant des scènes extérieures complexes, afin de tester l'efficacité de la méthode.

Contents

1 Introduction

Digitized time-ordered image sequences provide an actually rich support to analyze and interpret temporal events in a scene. Obviously the interpretation of dynamic scenes has to rely somehow on the analysis of displacements perceived in the image plane, which represent an important intrinsic intermediate information between input sensor data and output scene-related features. During the 80's, most of the works have focused on the two-frame problem, that is recovering the 3-D motion either from the optical flow field derived between time t and time $t+1$, [Adi85, BH83, NY86, WKS87], or from the matching of distinguished features (points, contour segments, ...) previously extracted from two successive images, [FLT87, LH86, MSA85].

Both approaches usually suffer from different shortcomings, like intrinsic ambiguities, [Adi89, Ber88], and above all numerical instability in case of noisy data, [HW88, JJ88]. Attempts to alleviate these difficulties have been undertaken by introducing for instance regularization terms, [YM86], or active vision paradigms, [AWB87, ST90]. As far as 3-D information recovery from motion is concerned, it is obvious that performance is improved by considering a more distant time interval between the two considered frames (by analogy with an appropriate stereo baseline). But matching problems become then overwhelming. Therefore, an attractive solution is to take into account more than two frames [Sha86], and to perform tracking over time using recursive temporal filtering, [BC86]. Moreover it offers advantages like formal modeling, useful prediction and smoothing. Tracking thus represents one of the central issues in dynamic scene analysis.

First investigations were concerned with tracking of points, [SJ87], and contour segments, [CSD88, DF90]. However the use of vertices or edges lead to a sparse set of trajectories and can make the procedure sensitive to occlusion. The interpretation process requires to group these features into consistent entities. This task can be more easily achieved when working with a limited class of a priori known objects [SD91]. It appears that the ability of directly tracking complete and coherent entities should enable to more efficiently solve for occlusion problems, and also should make the further scene interpretation step easier. This paper addresses this issue. Solving it requires to deal with a dense spatio-temporal information. We have developed a new tracking method which takes into account regions as features and relies on 2-D motion models.

Thi report is organized as follows. Section 2 gives a general description of the region tracking algorithm. A basic version of the algorithm based on a *geometric filter* is given in Section 3. Section 4 describes an improved version where a *motion filter* and a *geometric filter* cooperate. Results on synthetic and real data are presented to show the efficiency of the approach. Section 5 contains concluding remarks.

2 Region Modeling, Extraction and Measurement

We want to establish and maintain the successive positions of an object in a sequence of images. Regions are used as primitives for the tracking algorithm. Throughout this paper we will use the word, "regions", to refer to connected components of points issued from a motion-based segmentation step. The region can be interpreted as the silhouette of the projection of an object in the scene, in relative motion with respect to the camera.

Previous approaches, [Gam84, Gor89], to the "region-tracking" issue generally reduce to the tracking of the gravity center of regions. The problem of these methods is their inability to capture the projection in the image plane of complex 3-D motion of objects. Since the center of gravity of a region in the image may not correspond to the same physical point throughout the sequence, its trajectory does not actually characterize the evolution of the concerned region. It can even lead to erroneous interpretation.

We proceed as follows. First the segmentation of each image is performed using a motion-based segmentation algorithm previously developed in our lab. Second the correspondence between the predicted regions and the observations supplied by the segmentation process is established. At last a recursive filter refines the prediction, and its uncertainty, to obtain the estimates of the region location and shape in the image. A new prediction is then generated for the next image.

2.1 The Motion Based Segmentation Algorithm

The motion-based segmentation method, fully described in [FB91], 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 optic flow fields. It mainly relies on the spatio-temporal variations of the intensity function while making use of 2-D first-order motion models. It also manages to link those partitions in time, but of course to a short-term extent.

When the moving object is occluded for a while by another object or by the background and reappears, the motion-based segmentation process may not maintain the same label for the corresponding region over time. The same problem arises when trajectories of objects cross each other. Labels before occlusion may disappear and leave place to new labels corresponding to reappearing regions after occlusion. Consequently, tracking regions over long periods of time requires a filtering procedure to be steady. A truly trajectory representation and determination is needed. The segmentation process will provide only instantaneous measurements.

In order to work with regions, the concept of region must be defined in some mathematical sense. We describe hereafter the region descriptor used throughout this paper.

2.2 The Region Descriptor

2.2.1 The Region Representation

We need a model to represent regions. The representation of a region is not intended to capture the exact boundary. It should give a description of the shape and location that supports the task of tracking in presence of partial occlusion. Representation of shapes based on computation of moment-type values (e.g. center of gravity, principle axes, higher order moments, ...) are of little use when only a partial view is available, since they are obviously sensitive to occlusion.

We choose to represent regions with some of its boundary points. The contour is sampled in such a way that it preserves shape information of the silhouette. We must select points that best capture the global shape of the region. This is achieved through a polygonal approximation of the region. The local features, corners, points of high curvature are conserved by the approximation. Several methods have been proposed for polygonal approximation of a two dimensional shape. 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 Danielson in [WD84]. A criterion controls the closeness of the shape and the polygon. This method is simple, fast and gave good results on our data.
The region can be approximated accurately by this set of vertices. 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 hull. 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. We compute the convex hull with the technique described in [PS85]. As shown in the results reported further polygonal approximations do not restrict the type of objects nor the 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, ...

2.2.2 The Region Descriptor

This descriptor is intended to represent the silhouette of the tracked region, all along the sequence. We represent the tracked region with a constant number of points during successive time intervals of variable size. At the beginning of the interval we determine in the segmented image the number of points, n, necessary to represent the concerned region. We maintain this number fixed as long as the distance, defined in (1), between the predicted region and the observation extracted from the segmentation is not too important. The moment the distance becomes too large, the region descriptor is reset to an initial value equal to the observation. Consequently a new track is initialized. Thus the new state vector of the *geometric filter*, which describes the tracked region, is initialized with the current observation. This announces the beginning of a new interval.

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 polygonal approximation of the region : $[x_1, y_1, x_2, y_2, \ldots, x_n, y_n]^T$. Figure 1 illustrates the definition of the region descriptor.

2.3 The Measurement Vector

2.3.1 Measurement Definition

We need a measurement of the tracked region, in each image, in order to update the prediction generated by the filter. The measurement is derived from the observation

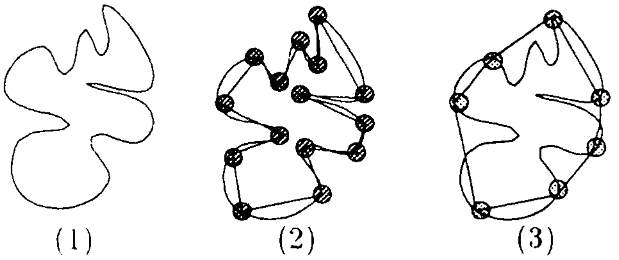

Figure 1: The region descriptor ; (1) Region ; (2) Boundary points approximating the region ; (3) Vertices of the convex hull

obtained by the segmentation process. Each observation is a region of the segmented image. For a given region we would like a measurement vector that depicts this region with the same number of points as the region descriptor. This number remains constant throughout a limited number of frames. The measurement algorithm consists of three stages, (See Fig. 2)

1. Association of observation produced by the segmentation and prediction generated by the filter ;

2. Global matching, or registration of the prediction and the observation ;

3. Local matching of each region and generation of the measurement.

First the association of observations and predictions is performed. We have to decide whether an observation given by the segmentation corresponds to a previously tracked object, or to a new one. When there is no occlusion the segmentation process assigns a same label over time to a given region ; thus the correspondence between prediction labels and observation labels is 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. In this case more complex methods must be derived to obtain the correct association. These methods are discussed later.

If the observation can be associated with a previously tracked region we will update the existing trajectory by refining the prediction with the observation. Updating the prediction requires the global matching, and the local matching of the predicted region and the observed one. The global matching involves determining the transformation necessary to globally superimpose the prediction onto the observation. Registration is achieved here at the region level. To perform this registration 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 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. 3).

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

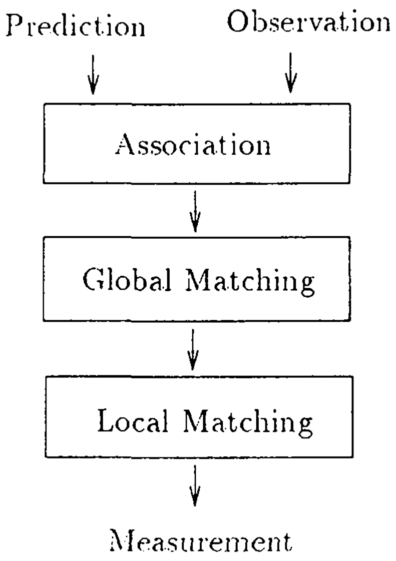

Figure 2: Measurement algorithm

of the tracked region may change, the region descriptors of the prediction and the observed region may have different number of points. Thus we do not take the vertices of the observation as the measurement vector. We prefer to select the n vertices of the correctly superimposed shape of the prediction onto the observation, as the measurement vector. The measurement coincides indeed with the segmented region. Furthermore this measurement algorithm has the following properties. If the object is partially occluded, the measurement still gives an equivalent complete view of the silhouette of the region. This approach deals with the problem of occlusion at the level of the region itself. It does not require the usual matching of specific features which is often a difficult issue. Indeed the measurement algorithm works on the region taken as a whole. Moreover the measurement is not sensible to small uncertainties and spurious variations in the shape created by the motion-based segmentation.

Thus the measurement method is well suited for temporal tracking since it delivers steady measurements of the shape and position of the tracked region.

2.3.2 Measurement Algorithm

If we represent the convex hull of the silhouette obtained by the segmentation and the prediction vector as two polygons, the problem of superimposing the shape of the prediction onto the observation reduces here to the problem of matching two convex polygons with possibly different number of vertices.

We do not use the structural approach nor the methods based on distances invariant to translation, rotation and scale, because we are interested in the transformation between the two polygons. Matching is achieved by moving a polygon and finding the best translation and rotation to superimpose it on the other one. We did not include scaling in

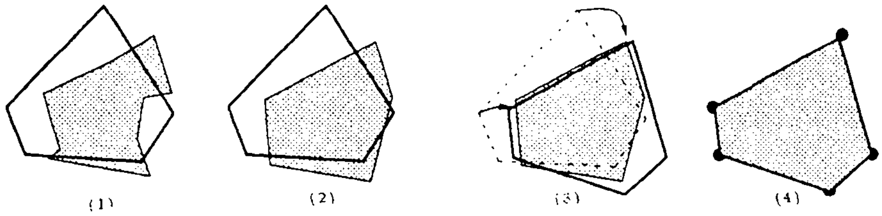

Figure 3: The measurement algorithm : (1) Observation obtained by the segmentation (grey region), and prediction (solid line) ; (2) Convex hull of the observation ; (3) Matching of polygons ; (4) Effective measurement : vertices of the grey region.

the transformation, otherwise in the case of occlusion the minimization process will 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,...) and seek the geometrical transformation that minimizes the distance between the two shapes.

We use the measure proposed by Cox et al. in [CMMR89]. If P_1 and P_2 are two polygons, we define the measure by :

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

where $d(M, P)$ is the Euclidean distance of a point M to the polygon P. This measure is not a distance because it does not obey to the triangular inequality. The minimum of $\mathcal{D}(P_1, P_2)$ 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. Let us suppose that we are moving the polygon P_2. We are looking for the transform T that minimizes the measure between P_1 and $T(P_2)$:

$$\mathcal{F}(T) = \mathcal{D}(P_1, T(P_2)) = \sum_{M_1 \in P_1} d(M_1, T(P_2))^2 + \sum_{M_2 \in P_2} d(T(M_2), P_1)^2 \qquad (1)$$

where T is the composition of a rotation and a translation. The function $\mathcal{F}$ is continuous, differentiable. It is also convex with respect to the two parameters of the translation. Thus conjugate-gradient methods can be efficiently used to solve the optimization process. 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.

3 The Basic Region-Based Tracking Algorithm

We propose hereafter a basic version of the tracking algorithm. The tracking algorithm relies on the geometric filter described hereafter. By suitable choice of the time scale, the time step between two successive frames can be chosen as unity.

3.1 The Geometric Filter

Each region is represented by the set of vertices (x_i, y_i) of the region descriptor.

We assume that trajectories of these vertices are independent. This assumption is considered for two reasons. First it gives more freedom for each vertex to evolve independently from the others. Thus the filter will be able to track complex transformation of the regions. The second reason is related to the size of the filter. The assumption allows us to decouple the state vector and to break a large filter containing all the vertices into n smaller filters.

Using another approach, a more accurate modeling of the correlation of the trajectories of the vertices is proposed in the next section.

Because the projection on the image plane is not a linear transformation, the motion of a point in the image sequence, undergoing a constant velocity motion in the scene, can generally not be represented by a motion with constant velocity not even with a constant acceleration. For this reason there is no exact transformation giving $x(t+1)$ and $y(t+1)$ as a linear function of the derivative of $x(t)$ and $y(t)$. To derive $x(t+1)$ and $y(t+1)$ from $x(t)$ and $y(t)$ we assume that the derivatives of order larger than three can be neglected, and we represent the third derivative as a sequence of zero-mean Gaussian white noise $\varepsilon(t)$ of variance σ_ε^2.

The measurement is described in Sect. 2.3.1. We are able to measure the position of each vertex.

Let

$$\mathbf{x}(t) = [x_i(t), \dot{x}_i(t), \ddot{x}_i(t)]^T \text{ and } \mathbf{y}(t) = [y_i(t), \dot{y}_i(t), \ddot{y}_i(t)]^T$$

-1z-1zbe the two state vectors ; and let $\mathbf{m} = [m_1(t), m_2(t)]$ be the measurement vector. Then the following dynamic system can be derived for each point (x_i, y_i) of the region descriptor in the image at time t, as shown in Appendix A :

$$\left\{ \begin{array}{ccccc} \mathbf{x}(t+1) & = & \mathbf{\Phi x}(t) & + & \xi_1(t) \\ \mathbf{y}(t+1) & = & \mathbf{\Phi y}(t) & + & \xi_2(t) \\ m_1(t) & = & \mathbf{Cx}(t) & + & \eta_1(t) \\ m_2(t) & = & \mathbf{Cy}(t) & + & \eta_2(t \end{array} \right. \tag{2}$$

$\xi_1(t)$ and $\xi_2(t)$ are two sequences of zero-mean Gaussian white noises of covariance matrix $\mathbf{Q}$. $\eta_1(t)$ and $\eta_2(t)$ are two sequences of zero-mean Gaussian white noise of variance σ_η^2. We have :

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 1 & \frac{1}{2} \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{C} = [1 \;\; 0 \;\; 0] \tag{3}$$

The covariance matrix of $\xi_1(t)$ or $\xi_2(t)$ is $\mathbf{Q}(t) = E\left[\xi_j(t)\xi_j{}^T(t)\right]$ $(j = 1$ or $2)$ has the following form [see Appendix A] :

$$\mathbf{Q} = \sigma_\xi^2 \begin{bmatrix} \frac{1}{20} & \frac{1}{8} & \frac{1}{6} \\ \frac{1}{8} & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{6} & \frac{1}{2} & 1 \end{bmatrix}$$

We assume that the above linear dynamic system is sufficiently accurate to model the motion of the region in the image. We want to estimate the vectors $\mathbf{x}(t)$ and $\mathbf{y}(t)$ from the measurement $v_1(t)$ and $v_2(t)$ respectively. The Kalman filter [Gel74] provides the optimal linear estimate of the unknown state vector from the measurements, in the sense that it minimizes the mean square estimation error and by choosing the optimal weight matrix gives a minimum unbiased variance estimate.

A standard Kalman filter generates recursive estimates of the two vectors $\mathbf{x}(t)$ and $\mathbf{y}(t)$.

Let t_0 be the instant of the first measurement. We calculate the first estimate at time t_0+2. First and second derivative are estimated at time t_0+2 using the first, second and third measurements. We obtain the following expression, as explained in appendix B :

$$\hat{\mathbf{x}}(t_0+2) = \begin{bmatrix} v_1(t_0+2) \\ v_1(t_0+2) - v_1(t_0+1) \\ v_1(t_0+2) - 2v_1(t_0+1) + v_1(t_0) \end{bmatrix}$$

$$\hat{\mathbf{y}}(t_0+2) = \begin{bmatrix} v_2(t_0+2) \\ v_2(t_0+2) - v_2(t_0+1) \\ v_2(t_0+2) - 2v_2(t_0+1) + v_2(t_0) \end{bmatrix}$$

The covariance matrix of $\hat{\mathbf{x}}(t_0+2)$ or $\hat{\mathbf{y}}(t_0+2)$ is given by. as shown in Appendix B :

$$E\left[\hat{\mathbf{x}}\hat{\mathbf{x}}^T\right](t_0+2) = \begin{bmatrix} \sigma_\eta^2 & \frac{3}{2T}\sigma_\eta^2 & \frac{1}{T^2}\sigma_\eta^2 \\ \frac{3}{2T}\sigma_\eta^2 & \frac{T^3}{3}\sigma_\xi^2 + \frac{13}{2T^2}\sigma_\eta^2 & \frac{9T^2}{40}\sigma_\xi^2 + \frac{6}{T^3}\sigma_\eta^2 \\ \frac{1}{T^2}\sigma_\eta^2 & \frac{9T^2}{40}\sigma_\xi^2 + \frac{6}{T^3}\sigma_\eta^2 & \frac{23T}{30}\sigma_\xi^2 + \frac{6}{T^4}\sigma_\eta^2 \end{bmatrix}$$

It should be pointed out that though the equations of the resulting filter are the same as those derived in the "token tracker" algorithms. [CSD88. DF90], our approach is quite different. Indeed we deal with a "region-level" measurement, and system models. Furthermore the segmentation is motion-based and thus the tracking process does not track spatially-segmented regions. but entities corresponding to objects moving in the scene.

3.2 Results

The airplane sequence

We present hereafter experiments performed on a sequence of real images with synthetic motion that will validate the approach. We have generated a sequence of 41 images. We superimposed on a fixed real image a patch cut out from another real image. This patch represents a plane. It undergoes a rotation motion about the upper right corner of the image. We can observe that the inter-frame motion is rather large (about 10 pixels). In the middle of the sequence, the plane disappears, as if crossing under an imaginary "cloud". It progressively reappears, the nose first, then the wings, and finally the entire plane. This sequence illustrates a case of occlusion.

Though the inter-frame displacements of the object are significantly large, the motion-based segmentation yields good results. When the nose of the plane reappears after having crossed under the "cloud", the segmentation algorithm of course considers it as a new region and assigns it a new label. As the plane is getting out of the "cloud" with the same new region label, the tail of the plane, with the initial region label disappears under the "cloud". We have here an occlusion problem : a region disappears while another reappears.

Experiment 1

Figure 18 illustrates the good performance of the method in the presence of a disappearing region. The top half of this figure shows the three images of the initial sequence, at time t_1, t_{16} and t_{23}, while bottom half offers a view of the results of the tracking at the corresponding instants. The tracked region is displayed as a polygon, where vertices are the points actually estimated by the filter. The polygon is superimposed on the segmentation map.

The algorithm accurately tracks the region, even in presence of partial occlusion, as shown in the central frame at time t_{16}. The magnitude of the variance of the measurement noise, σ_η^2 is computed as a function inversely proportional to the distance, defined by (1), between the prediction and the observation generated by the segmentation algorithm.

Hence when the region is almost completely occluded, the filter essentially ignores the measurement and relies on the system model. We observe on the rightmost frame, at time t_{23} that in such a case the estimate falls behind the observation. Nevertheless the newly appearing nose of the plane coincides with the "nose" of the region estimate. It is then possible to make the hypothesis that both regions correspond to the same object. Simple rules are sufficient to do it, based on the amount of area of the emerging region included in the estimate currently tracked. Let us point out that the motion-based segmentation can deliver the information that the emerging region is a new one, owing to the value of its label (greater than the number of regions in the last frame before this new region appears). In case of remaining conflicts, the motion measurement derived from the segmentation step can easily remove the ambiguity.

We can observe in Fig.19 the superposition of the region estimates from time t_1 to time t_{23}. We have only displayed one region over two for the clarity of the figure. It should be noted that though we choose here a simplified system model, we nevertheless have quite good results.

Experiment 2

This experiment illustrates the creation of a new region. We use the same sequence ; but we only consider the frames where the plane reappears (frames 30 to 41).

Original images are presented on top of Fig. 20 at time t_{30}, t_{35}, t_{41} ; and the tracked region is superimposed on the segmentation on the bottom of the figure. The size of the region descriptor is set to 5 because of the small size of the region in the frame where it appears for the first time.

Figure 21 shows the superposition the estimates of the region corresponding to a tracking starting from time t_{30} and ending at t_{41}. We can notice that the algorithm poorly updates the shape changement of the region. This phenomenon can be due to the lack of information concerning the region evolution.

Furthermore we noticed on another sequence (see sequence *van* in Sect. 4.2.5) that this basic version of the algorithm could not generate reliable estimates during long period of time without measurements.

On one hand, we observed that the filter based on the dynamic system (2) diverges after an extended period of time without measurements. This effect is present even though we add a fictitious process noise to take into account modelling errors (non linearity, biases, roundoff and truncation errors in the computation).

On the other hand we have tried a constant-velocity system model instead of (2). It appears that the filter is more stable but the prediction is always late. This phenomenon is more significant when measurements are not present and the filter generates only estimates. This is the case when a tracked region is being occluded.

In order to build a more robust tracking algorithm we decided to directly extract the motion information from the sequence ; and to use a more accurate model of the region evolution. The following section describes the complete algorithm based on these ideas.

4 The Improved Region-Based Tracking Algorithm

We propose an approach with a complete model for the prediction and update of the object geometry and motion. The motion information is not inferred from the tracking as before, but the tracking itself is based on motion parameters directly extracted from the sequence. Figure 4 shows the overview of the algorithm. We use two models : a *geometric model* and a *motion model*. At the higher level the geometric filter and the motion filter estimate shape, position and motion of the region from the observations produced by the segmentation. The two filters interact : the estimation of the motion parameters enables the prediction of the geometry of the region in the next frame. The shape of the region is compared with the prediction and the parameters of the region geometry are updated. A prediction of the shape and location of the region in the next frame is calculated from the estimates of the motion parameters and the estimates of the region geometry.

The motion-segmentation of each image is still performed with the algorithm described in Sect. 2.3.1. The motion parameters are given by one of the two methods described in Sect. 4.2.

Our approach has some similarities with the one proposed in [SM82]. The authors constraint the target motion in the image plane to be a 2-D affine transform. An overdetermined system allows to compute the motion parameters. However, the region representation and the segmentation step are quite different and less efficient. Furthermore the authors do not propose any temporal filtering of the motion parameters and the region geometry. Besides their approach does not take into account the problems of possible occlusion, or junction of trajectories.

4.1 The Geometric Filter

We assume that each region R, in the image at time $t+1$ is the result of an affine transformation 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 :

$$\begin{pmatrix} x \\ y \end{pmatrix}(t+1) \quad = \quad \mathbf{A}(t)\begin{pmatrix} x \\ y \end{pmatrix}(t) \quad +\mathbf{b}(t) \tag{4}$$

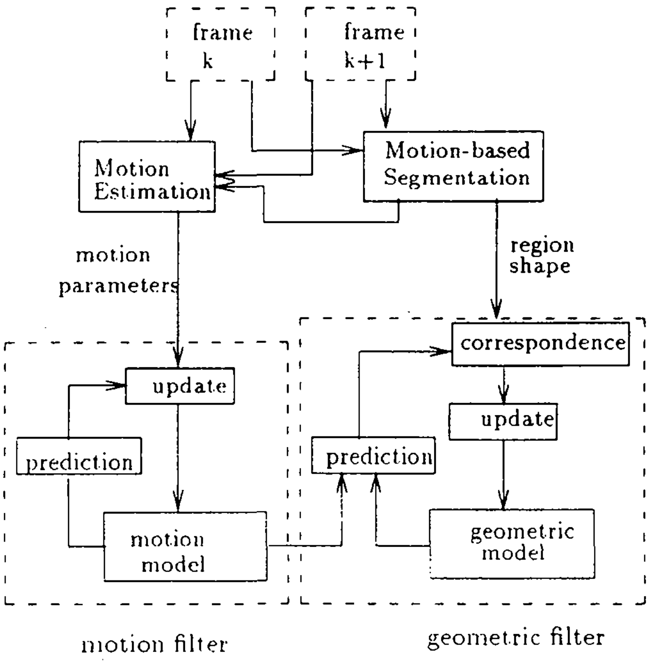

Figure 4: The complete region-based tracking filter

The affine transform has already been used to model small transformation between two images, [SM82].

As shown in (4) the region evolution model requires now the estimation of $A(t)$ and $b(t)$. These six parameters (four for $A(t)$, two for $b(t)$) are derived from another set of variables : the six parameters of an affine model of the 2-D velocity field within the region.

Indeed we can consider a first-order development of the 2-D motion field within each region. For a given region R in image at time t, the parameters of the 1st order model of the 2-D velocity field can be denoted by a matrix $\mathbf{M}(t)$ and a vector $\mathbf{w}(t)$ such that :

$$\forall (x, y) \in R, \quad \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix}(t) = \mathbf{M}(t) \begin{pmatrix} x \\ y \end{pmatrix}(t) + \mathbf{w}(t) \qquad (5)$$

Even if 2nd order terms generally result from the projection in the image of a rigid motion, they are sufficiently small to be neglected in such a context of tracking, which does not involve accurate reconstruction of 3-D motion from 2-D motion. Affine models of the velocity field have already been proposed by, e.g. [BBH*89] and [Adi85]. Furthermore the 2×2 matrix $\mathbf{M}(t)$ stands for a large class of motion: rotation, scaling, shear, ...; it indeed reflects valuable information on 3-D scene motion, as pointed out in [FB90]. Moreover this will allow us to keep linear relations for the Kalman filter.

The matrix $\mathbf{A}(t)$ and the vector $\mathbf{b}$ can be derived from the parameters of the affine model of the 2-D velocity field. Using a first-order approximation of the velocity we obtain the following relations :

$$\mathbf{A}(t) = \mathbf{I}_2 + \mathbf{M}(t) \quad \text{and } \mathbf{w}(t) = \mathbf{b}(t) \qquad (6)$$

where $\mathbf{I}_2$ is the 2×2 identity matrix. For convenience sake, the time step between two successive frames, δt, is chosen as unity.

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

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

where $\mathbf{A}(t)$ and $\mathbf{b}$ have been defined above in (7). $\zeta_i = [\zeta_i^x, \zeta_i^y]^T$ is a two dimensional, zero mean Gaussian noise vector. We choose a simplified model of the noise covariance matrix. We will assume that :

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

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 $\mathbf{A}(t)$ and the vector $\mathbf{b}(t)$ accounts for the displacements of all the points within the region, between t and $t+1$. Therefore the equation captures the global deformation of the region. Even though each vertex is tracked independently, the system model provides a "region-level" representation of the evolution of the points.

For each tracked vertex the measurement is given by the position of the vertex in the segmented image. The measurement process generates the measurement as explained in Sect. 2.3.1. The following system describes the dynamic evolution of each vertex (x_i, y_i) of region descriptor of the tracked region. Let $\mathbf{s}(t) = [x_i, y_i]^T$ be the state vector, and $\mathbf{m}(t)$ the measurement vector which contains the coordinates of the measured vertex,

$$\left\{ \begin{array}{rclllll} \mathbf{s}(t+1) & = & \mathbf{A}(t)\mathbf{s}(t) & + & \mathbf{b}(t) & + & \zeta(t) \\ \mathbf{m}(t) & = & \mathbf{s}(t) & & & + & \eta(t) \end{array} \right. \tag{7}$$

$\zeta(t)$ and $\eta(t)$ are two sequences of zero-mean Gaussian white noise. $\mathbf{b}(t)$ is interpreted as a deterministic input. $\mathbf{A}(t)$ is the matrix of the affine transform. Both are computed at each step with the motion filter. We estimate $\mathbf{s}(t)$ with a standard Kalman filter.

We use the first measurement as the initial estimate.

4.2 The Motion Filter

4.2.1 Monoresolution Estimation of the Motion Parameters

The motion-based segmentation algorithm requires the measurement of the parameters $\mathbf{M}(t)$ and $\mathbf{w}(t)$ defined in (5), for each region. A maximum-likelihood estimation is therefore performed. When the image sequence involves large displacements it appears that this method yields underestimated values.

On the other hand, the geometric filter requires estimates of the motion parameters which have to be, if not precise, at least correct in magnitude. For this reason, we have developed a multiresolution method to provide reliable motion parameters. These parameters have been used in an other context to derive accurate time-to-collision measurements based on a 2-D affine model of the velocity field, in [MB92]. We present hereafter the monoresolution scheme utilized within the segmentation algorithm. The multiresolution approach is given afterwards.

The principle of the estimation of the six parameters of the 2-D affine model of the velocity field is similar to the solution proposed in [Bro87]. Let Θ be the vector of the six parameters of the affine model : the four coefficients of $\mathbf{M}(t)$ and the two coefficients of $\mathbf{w}(t)$. The following "data-model adequacy" variable is considered :

$$\psi_\Theta(x, y) = \nabla\mathbf{I}(x, y).\mathbf{v}_\Theta(x, y) + I_t(x, y) \tag{8}$$

where $\mathbf{v}_\Theta(x, y)$ is the 2-D affine velocity field at location (x, y), given by (5). $\nabla\mathbf{I}(x, y)$ is the spatial gradient of the image intensity at pixel (x, y), and $I_t(x, y)$ is the partial derivative with respect to time of the image intensity. It relies on the well-known image flow constraint equation. [HS81]:

$$\nabla\mathbf{I}.\mathrm{v} + I_t = 0 \tag{9}$$

A maximum-likelihood estimation is achieved, which reduces here, assuming that ψ_Θ are independent zero-mean Gaussian variables of variance σ_ψ^2, to a least square estimation. Let $\mathcal{X}$ be the following $6 \times N$ array :

$$\mathcal{X} = \begin{bmatrix} I_x(p_1) & I_y(p_1) & xI_x(p_1) & xI_y(p_1) & yI_x(p_1) & yI_y(p_1) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ I_x(p_N) & I_y(p_N) & xI_x(p_N) & xI_y(p_N) & yI_x(p_N) & yI_y(p_N) \end{bmatrix}$$

Each row of $\mathcal{X}$ contains the partial derivatives of the brightness function I with respect to space and the products of these derivatives with the spatial coordinates, for each point $p_i, i = 1 \ldots N$ of the considered region. And let $\mathcal{Y}$ be the N vector which contains the partial derivatives with respect to time of the brightness function I, for the N points of the region :

$$\mathcal{Y} = [-I_t(p_1), \ldots, -I_t(p_N)]^t$$

Applying (8) to the N points of the region, we obtain :

$$\mathcal{Y} = \mathcal{X}\Theta - \Psi \text{ with } \Psi = [\psi_\Theta(p_1), \ldots, \psi_\Theta(p_N)]^t$$

The optimal estimator $\hat{\Theta}$ has the following classic expression :

$$\hat{\Theta} = \left[\mathcal{X}^t\mathcal{X}\right]^{-1}\mathcal{X}^t\mathcal{Y} \tag{10}$$

it follows :

$$\hat{\Theta} = \Theta - \left[\mathcal{X}^t\mathcal{X}\right]^{-1}\mathcal{X}^t\Psi \tag{11}$$

and $[\mathcal{X}^t\mathcal{X}]^{-1}\mathcal{X}^t\Psi$ is a zero mean Gaussian white noise of covariance matrix :

$$\sigma_\psi^2\left[\mathcal{X}^t\mathcal{X}\right]^{-1} \tag{12}$$

4.2.2 Multiresolution Estimation of the Motion Parameters

The image flow constraint equation (9) is in fact approximated by :

$$\nabla \mathbf{I}.\delta \mathbf{p} + I(p, t + \delta t) - I(p, t) = 0 \tag{13}$$

where $\delta \mathbf{p}$ is the displacement at location $\mathbf{p}$ from time t to time $t + \delta t$. The approximation of the velocity $\mathbf{v}$, at $\mathbf{p}$ is :

$$\mathbf{v} = \frac{\delta \mathbf{p}}{\delta t}$$

and δt is the time step between two successive frames. Even if δt is chosen as unity throughout all the paper, we will keep the notation δt in this section to emphasize on the temporal role of this constant. When the displacement, $\delta \mathbf{p}$, between two successive frames is not sufficiently small the approximated equation (13) does not hold anymore.

The idea is to estimate the six 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 [PR90] 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, hierarchical estimation of the optical flow with a gradient-based algorithm are proposed in [Enk88] ; in [Ana89] 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 [Bur84] ; and a pyramid for the segmented image. Each image in the Gaussian pyramid is a blurred and subsampled version of its predecessor. As a result the displacements measured in pixel, at level $l + 1$, are twice as less as the displacements at level l. Thus the approximation of the image constraint equation (13) can be more easily applied at the lowest resolution level. Delineations of regions are provided at each level by the pyramid of the segmented image.

First we estimate at the lowest resolution level, L,the six motion parameters of each region in the image, with (10). Then we refine the estimate. For a given level l, the displacement estimated at the coarser level $l + 1$ is projected on the current level and accounts for an initial displacement. Let $\mathbf{p}^l$ be a point within a region at level l. Let $\mathbf{v}_{\Theta^l}(\mathbf{p}^l)$ be the affine model of the velocity at location $\mathbf{p}^l$. Θ^l is the vector of the motion parameters to be estimated at level l. In fact we will estimate a refinement of Θ^{l+1} with an equation similar to (13).

For the clarity of the presentation and to lighten the notation we will drop the variable t in $\mathbf{M}^l(t)$, and in $\mathbf{w}^l(t)$ in the following : and we will only note $\mathbf{v}_{\Theta^l}$ to refer to $\mathbf{v}_{\Theta^l}(\mathbf{p}^l)$. With $\mathbf{M}^l$ and $\mathbf{w}^l$ defined by (5), we have :

$$\mathbf{v}_{\Theta^l} = \mathbf{M}^l.\mathbf{p}^l + \mathbf{w}^l$$

Since $\delta \mathbf{p}^l = \mathbf{v}_{\Theta^l}(\mathbf{p}^l).\delta t$ we are interested in the displacement $\delta \mathbf{p}^l$ at location $\mathbf{p}^l$. Equation (9) can be formulated as follows :

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

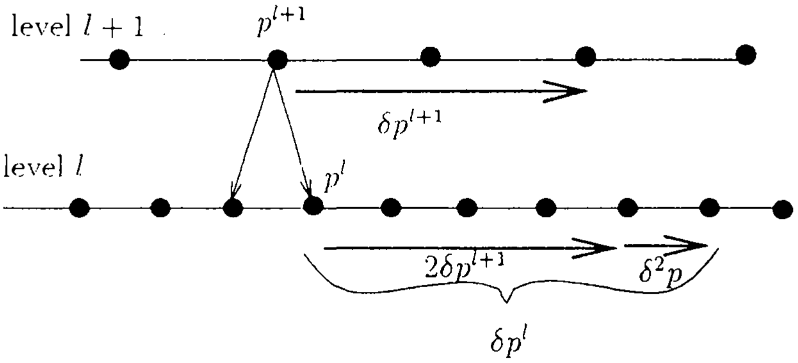

Figure 5: Hierarchical estimation of the motion parameters

Let us assume that $\mathbf{p}^{l+1}$ is the father of $\mathbf{p}^l$ at level $l+1$. And let $\delta\widehat{\mathbf{p}}^{l+1}$ be the displacement estimated, at level $l+1$, at location $\mathbf{p}^{l+1}$. We can project $\delta\widehat{\mathbf{p}}^{l+1}$ 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 $\delta\widehat{\mathbf{p}}^{l+1}$ on the level l is equal to : $2\delta\widehat{\mathbf{p}}^{l+1}$. It serves as an initial estimate to compute $\delta\mathbf{p}^l$. (See Fig. 5).

It follows from above :

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

where $\delta^2\mathbf{p}^l$ is the incremental estimate to be computed at level l. Expanding I in Taylor series about $\mathbf{p} + 2\delta\mathbf{p}^{l+1}$. and dropping higher order terms. leads to :

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

Because

$$\delta\widehat{\mathbf{p}}^{l+1} = \mathbf{v}_{\widehat{\Theta}^{l+1}}(\mathbf{p}^{l+1}).\delta t$$

it implies :

$$I(\mathbf{p}^l+2\mathbf{v}_{\widehat{\Theta}^{l+1}}(\mathbf{p}^{l+1}).\delta t.t+\delta t)+\nabla\mathbf{I}(\mathbf{p}^l+2\mathbf{v}_{\widehat{\Theta}^{l+1}}(\mathbf{p}^{l+1}).\delta t.t+\delta t).(\mathbf{v}_{\Theta^l}-2\mathbf{v}_{\widehat{\Theta}^{l+1}}(\mathbf{p}^{l+1}))\delta t = I(\mathbf{p}^l,t)$$

We define

$$\Delta\mathbf{M}^l = \mathbf{M}^l - \widehat{\mathbf{M}}^{l+1} \text{ and } \Delta\mathbf{w}^l = \mathbf{w}^l - 2\widehat{\mathbf{w}}^{l+1}$$

Consequently :

$$\nabla\mathbf{I}(\mathbf{p}^l+2\mathbf{v}_{\widehat{\Theta}^{l+1}}(\mathbf{p}^{l+1}).\delta t.t+\delta t).(\Delta\mathbf{M}^l\mathbf{p}^l+\Delta\mathbf{w}^l)\delta t+I(\mathbf{p}^l+2\mathbf{v}_{\widehat{\Theta}^{l+1}}(\mathbf{p}^{l+1}).\delta t,t+\delta t)-I(\mathbf{p}^l,t) = 0 \quad (14)$$

and we have :

$$2\mathbf{v}_{\widehat{\Theta}^{l+1}}(\mathbf{p}^{l+1}) = \widehat{\mathbf{M}}^{l+1}\mathbf{p}^l + 2\widehat{\mathbf{w}}^{l+1}$$

where

$$\widehat{\mathbf{M}}^{l+1} = \sum_{k=l+1}^{L-1} \Delta\widehat{\mathbf{M}}^k + \widehat{\mathbf{M}}^L \text{ and } \widehat{\mathbf{w}}^{l+1} = \sum_{k=l+1}^{L-1} 2^{k-(l+1)}\Delta\widehat{\mathbf{w}}^k + 2^{L-(l+1)}\widehat{\mathbf{w}}^L$$

It is possible to give a geometric representation of (14). For the clarity of the figure we represent the image in one dimension only in Fig. 6. The interpretation is absolutely the same in two dimensions.

If we use (13) to estimate the displacement at location p^l we will obtain the estimate $\widetilde{\Delta p^l}$ given by :

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

where ∇I is the slope of the intensity function in the one dimensional case. We can notice that due to the magnitude of the displacement the above estimation is rather bad. On the other hand if we assume that we have an initial estimate of the displacement, $2\delta p^{l+1}$, we can use (14) to obtain an incremental estimate, $\delta^2 p^l$:

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

where $\nabla I(p^l + 2\delta p^{l+1}. t + dt)$ is the slope of the image intensity function at location $p^l + 2\delta p^{l+1}$ and at time $t + dt$. If the initial estimate is close enough to the real displacement δp^l, then the incremental estimate will be accurate. Indeed, as shown in Fig. 6, since the initial estimate is in the vicinity of $p^l + \delta p^l$ the approximated equation (13) holds.

Equation (14) is linear with respect to $\Delta\mathbf{M}^l$ and $\Delta\mathbf{w}^l$. As explained in the monoresolution case we obtain with (14) maximum likelihood estimates $\Delta\widehat{\mathbf{M}}^l$ and $\Delta\widehat{\mathbf{w}}^l$ of $\Delta\mathbf{M}^l$ and $\Delta\mathbf{w}^l$.

At the lowest resolution level L we use (13) :

$$\nabla\mathbf{I}(\mathbf{p}^L.t).(\mathbf{M}^L\mathbf{p}^L + \mathbf{w}^L)\delta t + I(\mathbf{p}^L, t + \delta t) - I(\mathbf{p}^L, t) = 0 \tag{15}$$

At this level we derive estimates $\widehat{\mathbf{M}}^L(t)$ and $\widehat{\mathbf{w}}^L(t)$ of $\mathbf{M}^L(t)$ and $\mathbf{u}^L(t)$. We have at each level a measurement model similar to (11) :

$$\left\{\begin{array}{lcccc} \Delta\widehat{\mathbf{M}}^l(t) & = & \Delta\mathbf{M}^l(t) & + & n_{\Delta\widehat{\mathbf{M}}^l}(t) \\ \Delta\widehat{\mathbf{w}}^l(t) & = & \Delta\mathbf{w}^l(t) & + & n_{\Delta\widehat{\mathbf{w}}^l}(t) \end{array}\right.$$

where $n_{\Delta\widehat{\mathbf{M}}^l}(t)$ and $n_{\Delta\widehat{\mathbf{w}}^l}(t)$ are two sequences of zero-mean gaussian white noise of covariance $cov(n_{\Delta\widehat{\mathbf{M}}^l})$ and $cov(n_{\widehat{\mathbf{w}}^L})$. Each of these covariance matrix is similar to (12).

We retain the final estimates at the highest resolution level : $\widehat{\mathbf{M}}^0$ and $\widehat{\mathbf{w}}^0$.

$$\left\{\begin{array}{lcccc} \widehat{\mathbf{M}}^0 & = & \sum_{l=0}^{L-1} \Delta\widehat{\mathbf{M}}^l & + & \widehat{\mathbf{M}}^L \\ \widehat{\mathbf{w}}^0 & = & \sum_{l=0}^{L-1} 2^l \Delta\widehat{\mathbf{w}}^l & + & 2^L \widehat{\mathbf{w}}^L \end{array}\right.$$

Combining the two preceding equations leads to :

$$\left\{\begin{array}{lcccc} \widehat{\mathbf{M}}^0(t) & = & \mathbf{M}^0(t) & + & n_{\widehat{\mathbf{M}}^0}(t) \\ \widehat{\mathbf{w}}^0(t) & = & \mathbf{w}^0(t) & + & n_{\widehat{\mathbf{w}}^0}(t) \end{array}\right. \tag{16}$$

with :

$$\left\{\begin{array}{lcccc} n_{\widehat{\mathbf{M}}^0} & = & \sum_{l=0}^{L-1} n_{\Delta\widehat{\mathbf{M}}^l} & + & n_{\widehat{\mathbf{M}}^L} \\ n_{\widehat{\mathbf{w}}^0} & = & \sum_{l=0}^{L-1} 2^l n_{\Delta\widehat{\mathbf{w}}^l} & + & 2^L n_{\widehat{\mathbf{w}}^L} \end{array}\right.$$

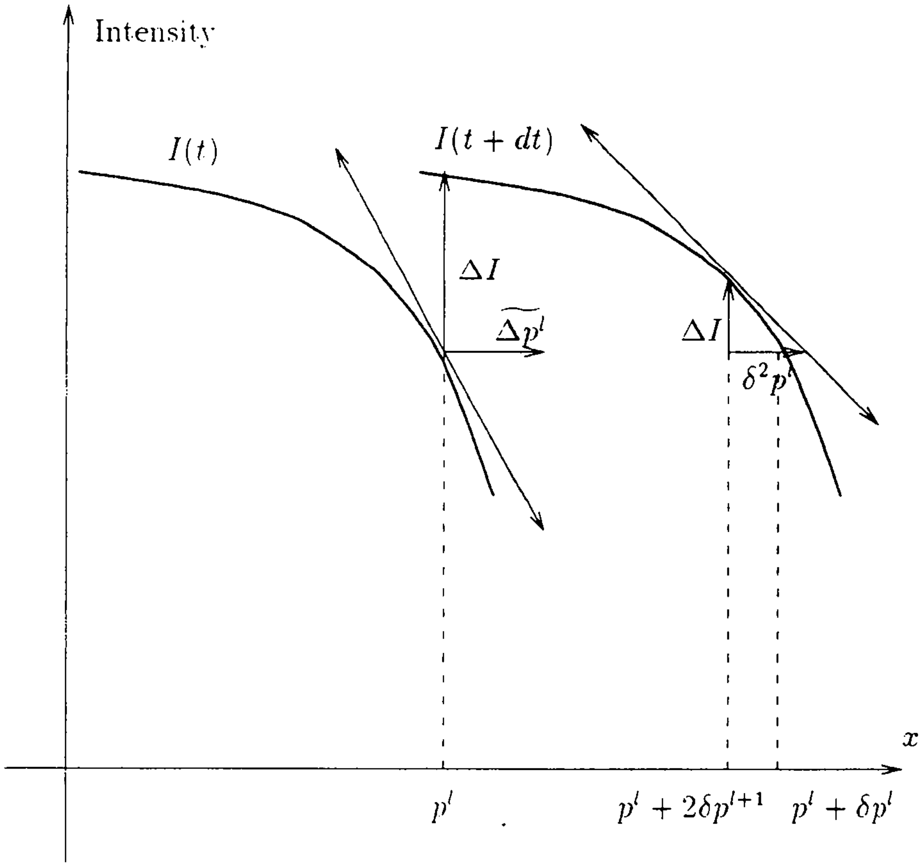

Figure 6: Geometric interpretation in one dimension of the computation of the incremental estimate $\delta^2 p^l$

Assuming that the noise vectors $n_{\Delta}\widehat{\mathbf{M}}^l, l = 0, \dots, L-1$ and $n_{\widehat{\mathbf{M}}^L}$ are independent we derive the expression of the covariance matrix of $n_{\widehat{\mathbf{M}}^0}$.

$$cov(n_{\widehat{\mathbf{M}}^0}) = \sum_{l=0}^{L-1} cov(n_{\Delta\widehat{\mathbf{M}}^l}) + cov(n_{\widehat{\mathbf{M}}^L}) \tag{17}$$

With a similar assumption we obtain the covariance matrix of $n_{\widehat{\mathbf{w}}^0}$:

$$cov(n_{\widehat{\mathbf{w}}^0}) = \sum_{l=0}^{L-1} 4^l cov(n_{\Delta\widehat{\mathbf{w}}^l}) + 4^L cov(n_{\widehat{\mathbf{w}}^L}) \tag{18}$$

4.2.3 Comparing the Monoresolution and Multiresolution Approaches

Both methods have been tested on sequences of real images. It appears that the multiresolution approach yields more accurate estimates. The accuracy of the parameters has been tested on a sequence of real images with known synthetic motion. We present present in the Results Section (Sect. 4.2.5) an experiment performed on a sequence of real images to compare the measurements with the ground truth determined by independent means.

To determine the number of levels of the pyramid, we take into account the size of each region present in the segmented image. At each successive level the region area is divided by four. Therefore at the lowest resolution level the least square estimation of the motion parameters may give erroneous results if there are not enough points any longer within the region. In such case the matrix in (10) can happen to be singular. The size of each region has to be balanced against the expected magnitude of the motion to determine the maximum level of the pyramid. Practically speaking we use two level, three level and four level pyramids.

Finally the multiresolution approach requires more memory storage and more computations. Bilinear interpolations of I and $\nabla\mathbf{I}$ have to be computed in (14) for points which are not on the grid.

4.2.4 Recursive Estimation of the Motion Parameters

The monoresolution or multiresolution methods provide us with instantaneous measurements of the motion parameters. Even if multiresolution estimates are generally less noisy, we need to filter these measurements to generate more accurate and more stable estimates. Furthermore the dynamic system which describes the temporal evolution of the motion parameters will provide us with estimates of these parameters at each instant, even when measurements are not present to update the prediction. In such a case the dynamic system generates only estimates. This is the case when a tracked region is being occluded.

In the absence, in the general case, of any explicit simple analytical function describing the evolution of the variables, we use a Taylor-series expansion of each function about t $= t_k$. Is is possible to derive a closed form expression of the motion parameters in some special cases. For instance, if the tracked object is moving with constant velocity along the optical axis toward the camera, then the coefficients of the matrix $\mathbf{M}(t)$ have the general expression :

$$\frac{at+b}{ct+d}$$

Therefore it is not possible to obtain, with a dynamic linear system, an accurate description of the evolution of the parameters which is valid for all t. However the following dynamic system provides a local approximation of the temporal evolution of each of the motion parameters. After having experimented with different approximations, it appears that using the first three terms performs a good tradeoff between the complexity of the filter and the accuracy of the estimates.

We have observed on many sequences that the correlation coefficients between the six components of $\widehat{\Theta}(t)$ are negligible. For this reason, we have decided to decouple the six variables. The advantage is that we work with six separate Kalman filters. Finally we can derive the following measurement model for each component of Θ, from (11) in the monoresolution case, or from (16), in the multiresolution case :

$$\hat{\mu}(t) = \mu(t) + \lambda(t) \tag{19}$$

where μ is any of the six components of Θ, and $\lambda(t)$ is a sequence of zero-mean Gaussian white noise of variance ${\sigma^2}_\lambda$. ${\sigma^2}_\lambda$ is the corresponding diagonal coefficient of the covariance matrix in (12) for the monoresolution estimation scheme, or in (17) and (18) for the multiresolution estimation scheme.

Let $\Xi(t) = [\mu(t), \dot{\mu}(t), \ddot{\mu}(t)]^T$ be the state vector, where μ is any of the six motion parameters : the four components of $\mathbf{M}(t)$ and the two components of $\mathbf{w}(t)$. $z(t)$ is the corresponding component of $\widehat{\Theta}(t)$ given by the mono or multiresolution method. We derive the following linear dynamic system :

$$\left\{ \begin{array}{rcll} \Xi(t+1) & = & \mathbf{\Phi}\Xi(t) & + \ \gamma(t) \\ z(t) & = & \mathbf{C}(t)\Xi(t) & + \ \lambda(t) \end{array} \right. \tag{20}$$

$\gamma(t)$ and $\lambda(t)$ are two sequences of zero-mean Gaussian white noises of covariance matrix $\mathbf{R}$, and variance σ_λ^2 respectively, with

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 1 & \frac{1}{2} \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{C} = [1 \ \ 0 \ \ 0]$$

The covariance matrix $\mathbf{R}(t)$ of $\gamma(t)$ is given by, as shown in Appendix A :

$$\mathbf{R} = \sigma_{\mathbf{R}}^2 \begin{bmatrix} \frac{1}{20} & \frac{1}{8} & \frac{1}{6} \\ \frac{1}{8} & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{6} & \frac{1}{2} & 1 \end{bmatrix}$$

4.2.5 Experiments

We present hereafter two experiments on real data that illustrate the performance of the filter. The plots represent the evolution of the six kinematic variables, with respect to time, [see (5)] :

$$\mathbf{M}(t) = \begin{bmatrix} \mathbf{M}_{1,1} & \mathbf{M}_{1,2} \\ \mathbf{M}_{2,1} & \mathbf{M}_{2,2} \end{bmatrix} \qquad \text{and} \qquad \mathbf{w}(t) = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

The robot sequence

The first experiment was performed on real images to compare the measurements given by our method with the ground truth determined by independent means. It also illustrates the comparison between monoresolution estimation and multiresolution estimation of the motion parameters.

The sequence *robot*, (32 images), 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 pined on a wall. The constant velocity of the camera, in the camera coordinate system, 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 deg), i.e. the image plane is not parallel to the wall. The parameters of the affine model are calculated analytically with the well known expressions relating the first order approximation of the optical flow with the structure and the kinematics parameters of a planar surface undergoing rigid motion, [FB90]. Since we have $\beta = 0$ and $\gamma = 0$ for the considered motion we did not plot these two coefficients.

For each parameter we plotted :

- the estimate obtained with the monoresolution method,
- the estimate obtained with the multiresolution approach, for :
  - 2 levels,
  - 3 levels,
  - 4 levels.
- the true value of the parameter.

Because there is only one region in the image we also estimated the parameters with a 5 level pyramid. But no significant improvement could be noticed. This phenomenon is in fact more general. If the estimation is correct for a certain level, using more levels will give essentially the same result. The level at which the improvement stops depends only on the magnitude of the true parameter to be estimated.

Comparative performance results from the different levels are displayed in Figs. 7, 8, 9, which show respectively the estimates of : $\mathbf{w}_1(t)$, $\mathbf{M}_{1,1}(t)$ and $\mathbf{M}_{2,2}(t)$. We can notice the good correspondence between the 4 level multiresolution measurements and the true parameters. Because the magnitude of the motion is important (4 to 6 pixels per frame) we need 4 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.

The van sequence

The second sequence of 66 real images (Fig. 24) illustrates a case of occlusion. The scene takes place at a crossroad. A white van is comming from the left of the picture and going to the right (Fig. 24). A black car is driving behind the van at the same speed. A white car is coming from the opposite side of the road. While crossing the image from right to left with an oblique trajectory in the image plane, it is comming closer to the camera. Thus its projection is getting bigger. 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 plotted each of the six motion parameters of the car before, and after occlusion. The moment the car disappears, we do not have any motion measurements. Nevertheless we are able to predict the value of each parameter using the dynamic model (20). For almost every motion parameter the prediction generated with the dynamic system coincides with the motion parameters of the reappearing car. The dynamic model (20) is thus accurate enough to generate predictions during long time intervals. Because of the magnitude of the displacement we need a three level pyramid to estimate $\mathbf{w}$. On the other hand the tracked region, i.e. the white car, has a small area in the segmented image (refer to Fig. 25). Consequently, as explained in Sect. 4.2.3 the least square estimation of the affine parameters at the lowest level is rather noisy. Thus final estimates of $\mathbf{w}_1$ and $\mathbf{w}_2$ are also noisy, (see Figs. 14, 15). However the Kalman filter efficiently provides stable estimates of the two coefficients, (see Figs. 14,15). We notice on Fig. 11 that the prediction of $\mathbf{M}_{2,1}$ is not accurate due to the last measurements (frame 30, and later) which give a different tendency to the prediction. It should be pointed out that theoretically none of the motion parameters are equal to zero since the car is undergoing a complex motion in the scene.

This experiment illustrates the efficiency of the motion filter to provide accurate motion parameters even in the case of complex motion behaviour, and in the presence of complete and long occlusions.

4.3 Results

We present the results of two experiments done on real images.

4.3.1 The parking lot sequence

The first sequence (Fig. 22) contains two cars manoeuvring in a parking lot. The first car in the foreground is rotating and moving toward the camera. The second one in the background is translating toward the camera. It partially hides a third car which is static. The camera pans the scene from right to left. The image background consists of trees, some of them being stirred by the wind (in particular in the centre of the image plane).

The top half of this figure shows three images from the original sequence, at time t_1, t_4 and t_9, while bottom half offers a view of the results of the tracking at the corresponding instants. We can observe in Fig.23 the trajectory of the regions in the image plane. The variance σ_ξ^2 is taken equal to 0.001. The results shows the efficiency of the approach to track multiple regions in an image sequence.

4.3.2 The van sequence

The sequence has been presented in the above section. The polygons representing the tracked regions are superimposed onto the original pictures at time t_{24}, t_{38}, t_{52}, and t_{66}. The corresponding segmented pictures at the same instants are presented on Fig. 25. The black car is driving so closely behind the van that the segmentation is unable to split the two objects (Fig. 25). From the beginning of the sequence until the car disappears the algorithm accurately tracks the region, updating the position and motion of the car with measurements. (Fig. 24).

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. Indeed the prediction coincides with the reappearing car. It is important to note that the white car is accelerating all along the sequence, as shown in Fig. 16 and Fig. 17. We have processed the same sequence with the basic region-based tracking algorithm with a constant velocity model. We noticed that the prediction was very late when the car pops up. This emphasizes the efficiency of the affine model of the velocity, even when objects are accelerating. It should be pointed out that we did not initiated a new track with the reappearing region. We only superimposed the prediction generated with the filter which tracked the region before occlusion. This association between tracks is left for future research as explained in the next section.

This example illustrates the good performance of the region-based tracking in the presence of a total occlusion.

5 Future Work

This work has investigated a new approach to the tracking of region in an image sequence. The work emphasized on the maintenance of the tracking. A necessary future work involves the design of a high level data association module, which will go beyond the simpler techniques discussed in a previous section. This module should control the measurement-to-track assignment in case of conflict situations. It should also solve the problem of track initiation and deletion. When a track is not updated for an extensive period of time, it should be deleted. Alternatively if a measurement can not be assigned to an existing track a new track should be initiated. Multiple target tracking techniques [Bar78],[Bla86], should provide a theoretical framework to deal with these issues. This high level data-association module integrated with the region tracking algorithm will offer an efficient solution to the issue of tracking multiple regions in long and complex image sequences.

6 Conclusion

This paper has explored an original approach to the issue of tracking objects in a sequence of monocular images. We have presented a new region-based tracking method which delivers dense trajectory maps. It allows to directly handle entities at an "object-level". It exploits the output of a motion-based segmentation step which takes into account normal flows and first order visual motion models. A first version of the algorithm has been derived, with a simple kinematic state model of the region. Experiments with a sequence of real images with synthetic motion have shown the ability of the method to track a region in presence of partial occlusion. Then we have proposed a complete version of the method. This algorithm relies on two interacting filters : a geometric filter which predicts and updates the region position and shape, and a motion filter which gives a recursive estimation of the motion parameters of the region. We have developed a multiresolution method to provide reliable motion parameters. Experiments have been carried out on real images to validate the performance of the method. A case of a complete occlusion on real images has been successfully handled. The promising results obtained indicate the strength of the "region approach" to the problem of tracking objects in sequences of images.

A Obtaining the discrete state equations

We consider a system where the third derivative of the state variable, $x(t)$, with respect to time is only white noise. Its behavior is described with the following first-order vector-matrix differential equation :

$$\dot{\mathbf{x}}(t) = H(t)\mathbf{x}(t) + Bw(t)$$

where :

$$\mathbf{x}(t) = [x(t), \dot{x}(t), \ddot{x}(t)]^T \qquad H = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \qquad \text{and} \qquad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

where $w(t)$ is a sequence of zero-mean Gaussian white noise. The solution of the time-invariant linear differential system is, [Wib71] :

$$\mathbf{x}(t) = \epsilon^{H(t-t_0)}\mathbf{x}(t_0) + \int_{t_0}^{t} \epsilon^{H(t-\tau)} Bw(\tau)d\tau$$

Using the above equation we get the discrete formulation of the system at the instants $t_k = kT$. The resulting linear difference equation is :

$$\mathbf{x}((k+1)T) = \mathbf{x}(k+1) = \mathbf{\Phi}\mathbf{x}(k) + \xi(k)$$

with

$$\mathbf{\Phi} = \epsilon^{HT} = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \tag{21}$$

and

$$\xi(k) = \int_{kT}^{(k+1)T} \epsilon^{H((k+1)T-\tau)} B(\tau)w(\tau)d\tau = \int_{kT}^{(k+1)T} \begin{bmatrix} \left(\frac{(k+1)T-\tau}{2}\right)^2 \\ (k+1)T-\tau \\ 1 \end{bmatrix} w(\tau)d\tau$$

$\xi(k)$ is therefore a sequence of zero-mean Gaussian white noise.

$$\xi(k) = \int_{kT}^{(k+1)T} \epsilon^{H((k+1)T-\tau)} B(\tau)w(\tau)d\tau = \int_{kT}^{(k+1)T} \begin{bmatrix} n_1(\tau) \\ n_2(\tau) \\ n_3(\tau) \end{bmatrix} d\tau$$

The covariance matrix $\mathbf{Q}(k)$ of $\xi(k)$ is :

$$\mathbf{Q}(k) = \int_{kT}^{(k+1)T} \begin{bmatrix} w_1(\tau)w_1(\tau) & w_1(\tau)w_2(\tau) & w_1(\tau)w_3(\tau) \\ w_2(\tau)w_1(\tau) & w_2(\tau)w_2(\tau) & w_2(\tau)w_3(\tau) \\ w_3(\tau)w_1(\tau) & w_3(\tau)w_2(\tau) & w_3(\tau)w_3(\tau) \end{bmatrix} d\tau$$

It follows :

$$\mathbf{Q}(k) = \sigma_w^2 \begin{bmatrix} \frac{T^5}{20} & \frac{T^4}{8} & \frac{T^3}{6} \\ \frac{T^4}{8} & \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^3}{6} & \frac{T^2}{2} & T \end{bmatrix} \tag{22}$$

B Filter initialization

We consider the following dynamic system :

$$\left\{ \begin{array}{rcll} \mathbf{x}(k+1) & = & \mathbf{\Phi}\mathbf{x}(k) & + \ \xi(k) \\ z(k) & = & \mathbf{C}\mathbf{x}(k) & + \ \eta(k) \end{array} \right. \tag{23}$$

with

$$\mathbf{x}(k) = [x(k), \dot{x}(k), \ddot{x}(k)]^T \qquad \mathbf{\Phi} = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{C} = [1 \ \ 0 \ \ 0] \tag{24}$$

and

$$\xi(k) = [\xi_1(k), \xi_2(k), \xi_3(k)]^T$$

$\xi(k)$ and $\eta(k)$ are two sequences of zero-mean Gaussian white noise. The variance of $\eta(k)$ is σ_η^2, and the covariance matrix of $\xi(k)$ is given by (22) :

$$\mathbf{Q}(k) = \sigma_\xi^2 \begin{bmatrix} \frac{T^5}{20} & \frac{T^4}{8} & \frac{T^3}{6} \\ \frac{T^4}{8} & \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^3}{6} & \frac{T^2}{2} & T \end{bmatrix}$$

We need to specify the value of the initial estimate and the covariance matrix of this estimate. The initial estimate is build at time 2T. We use approximations of the velocity and the acceleration at the corresponding instant. We obtain :

Theorem 1 *The following vector :*

$$\hat{\mathbf{x}}(2) = \left[z(2), \frac{\frac{3}{2}z(2) - 2z(1) + \frac{1}{2}z(0)}{T}, \frac{z(2) - 2z(1) + z(0)}{T^2} \right]^T$$

is an unbiased estimate of the vector $\mathbf{x}$ *at time* $2T$*. Where* $\mathbf{x}$ *and* z *are defined in (23) and (24). Its covariance matrix is :*

$$cov(\hat{x}) = \begin{bmatrix} \sigma_\eta^2 & \frac{3}{2T}\sigma_\eta^2 & \frac{1}{T^2}\sigma_\eta^2 \\ \frac{3}{2T}\sigma_\eta^2 & \frac{T^3}{3}\sigma_\xi^2 + \frac{13}{2T^2}\sigma_\eta^2 & \frac{9T^2}{40}\sigma_\xi^2 + \frac{6}{T^3}\sigma_\eta^2 \\ \frac{1}{T^2}\sigma_\eta^2 & \frac{9T^2}{40}\sigma_\xi^2 + \frac{6}{T^3}\sigma_\eta^2 & \frac{23T}{30}\sigma_\xi^2 + \frac{6}{T^4}\sigma_\eta^2 \end{bmatrix}$$

Proof

We define the error between the estimate and the state vector :

$$\tilde{\mathbf{x}} = \begin{bmatrix} \tilde{x} \\ \tilde{\dot{x}} \\ \tilde{\ddot{x}} \end{bmatrix} = \hat{\mathbf{x}} - \mathbf{x}$$

Let us calculate $\tilde{\mathbf{x}}$. First we have :

$$\tilde{x}(2) = x(2) - z(2) = \eta(2) \tag{25}$$

In order to obtain $\tilde{\dot{x}}$ and $\tilde{\ddot{x}}$ we will derive the expression of : $z(0)$, $z(1)$, $z(2)$, $\dot{x}(2)$ and $\ddot{x}(2)$ with respect to $x(0)$, $\dot{x}(0)$ and $\ddot{x}(0)$ and the noise variables.

On one hand we have

$$\begin{aligned}
\dot{x}(2) &= \dot{x}(1) + T\ddot{x}(1) + \xi_2(1) \\
\dot{x}(2) &= \dot{x}(0) + T\ddot{x}(0) + \xi_2(0) \\
&\quad + T\ddot{x}(0) + T\xi_3(0) + \xi_2(1)
\end{aligned}$$

Hence :

$$\dot{x}(2) = \dot{x}(0) + 2T\ddot{x}(0) + \xi_2(0) + T\xi_3(0) + \xi_2(1) \tag{26}$$

In a similar way :

$$\ddot{x}(2) = \ddot{x}(0) + \xi_3(0) + \xi_3(1) \tag{27}$$

On the other hand, using the same approach we can derive the following relations.

Firstly :

$$z(0) = x(0) + \eta(0) \tag{28}$$

Secondly :

$$z(1) = x(0) + T\dot{x}(0) + \frac{T^2}{2}\ddot{x}(0) + \xi_1(1) + \eta(1) \tag{29}$$

Finally :

$$z(2) = x(0) + 2T\dot{x}(0) + \ddot{x}(0) - \xi_1(0) + T\xi_2(0) + \frac{T^2}{2}\xi_3(0) + \xi_1(1) + \eta(2) \tag{30}$$

Using (28), (29), (30) we get :

$$\hat{\dot{x}}(2) = \dot{x}(0) + 2T\ddot{x}(0) - \frac{1}{2T}\xi_1(0) + \frac{3}{2}\xi_2(0) + \frac{3T}{4}\xi_3(0) + \frac{3}{2T}\xi_1(1) + \frac{1}{T}(\frac{3}{2}\eta(2) - \eta(1) + \frac{1}{2}\eta(0)) \tag{31}$$

and :

$$\hat{\ddot{x}}(2) = \ddot{x}(0) - \frac{1}{T^2}\xi_1(0) + \frac{1}{T}\xi_2(0) + \frac{1}{2}\xi_3(0) + \frac{1}{T^2}\xi_1(1) + \frac{1}{T^2}(\eta(2) - 2\eta(1) + \eta(0)) \tag{32}$$

At last since

$$\tilde{\dot{x}}(2) = \dot{x}(2) - \hat{\dot{x}}(2) \qquad \text{and} \qquad \tilde{\ddot{x}}(2) = \ddot{x}(2) - \hat{\ddot{x}}(2)$$

we obtain from (31) and (32) the following expression of the error vector :

$$\tilde{\dot{x}}(2) = -\frac{1}{2T}\xi_1(0) + \frac{1}{2}\xi_2(0) - \frac{T}{4}\xi_3(0) + \frac{3}{2T}\xi_1(1) - \xi_2(1) + \frac{1}{T}(\frac{3}{2}\eta(2) - \eta(1) + \frac{1}{2}\eta(0)) \tag{33}$$

and

$$\tilde{\ddot{x}}(2) = -\frac{1}{T^2}\xi_1(0) + \frac{1}{T}\xi_2(0) - \frac{1}{2}\xi_3(0) + \frac{1}{T^2}\xi_1(1) - \xi_3(1) + \frac{1}{T^2}(\eta(2) - 2\eta(1) + \eta(0)) \tag{34}$$

From equations (25), (33) and (34) it is obvious that $\widehat{x}$ is an unbiased estimate of $\mathbf{x}(2)$. With (25), (33) and (34) it is possible to obtain the expression of the covariance matrix of the estimate : $cov(\widehat{x})$ with respect to σ_η^2 and σ_ξ^2.

We have

$$cov(\widehat{x}) = E[\tilde{x}\tilde{x}^T]$$

After some calculations we get :

$$cov(\widehat{x}) = \begin{bmatrix} \sigma_\eta^2 & \frac{3}{2T}\sigma_\eta^2 & \frac{1}{T^2}\sigma_\eta^2 \\ \frac{3}{2T}\sigma_\eta^2 & \frac{T^3}{3}\sigma_\xi^2 + \frac{13}{2T^2}\sigma_\eta^2 & \frac{9T^2}{40}\sigma_\xi^2 + \frac{6}{T^3}\sigma_\eta^2 \\ \frac{1}{T^2}\sigma_\eta^2 & \frac{9T^2}{40}\sigma_\xi^2 + \frac{6}{T^3}\sigma_\eta^2 & \frac{23T}{30}\sigma_\xi^2 + \frac{6}{T^4}\sigma_\eta^2 \end{bmatrix} \tag{35}$$

References

[Adi85] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Trans. PAMI*, Vol 7:384–401, July 1985.

[Adi89] G. Adiv. Inherent ambiguities in recovering 3D motion and structures from a noisy flow field. *IEEE Trans. PAMI*, Vol.11:477–489, May 1989.

[Ana89] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, Vol 2:283–310, 1989.

[AWB87] J. Aloimonos, I. Weiss, and A. Bandopadhay. Active vision. *Proc. of the 1st Int. Conf. on Computer Vision, London*, pp 35–54, June 1987.

[Bar78] Y. Bar-Shalom. Tracking methods in a multitarget environment. *IEEE Trans. on Automatic Control*, pp 618–626, August 1978.

[BBH*89] 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. *IEEE Workshop on Visual Motion*, pp 2–12, March 1989.

[BC86] T.J. Broida and R. Chellappa. Estimation of objects motion parameters from noisy images. *IEEE Trans. PAMI*, Vol.8, No.1:90–99, Jan. 1986.

[Ber88] F. Bergholm. Motion from flow along contours: a note on robustness and ambiguous cases. *Int. Journal on Computer Vision*, Vol.3:395–415, 1988.

[BH83] A.R. Bruss and B.K.P. Horn. Passive navigation. *Computer Vision, Graphics and Image Processing*, Vol.21:3–20, 1983.

[Bla86] S.S. Blackman. *Multiple-Target Tracking with Radar Applications*. Artech House, 1986.

[Bro87] R.W. Brockett. Gramians, generalized inverses, and the least squares approximation of optical flow. *Proc. Int. Conf. on Robotics and Automation, Raleigh, North Carolina*, 1987.

[Bur84] P.J. Burt. The pyramid as a structure for efficient computation. In A. Rosenfeld, editor, *Multiresolution Image Processing and Analysis*, pages pp 6–35, Springer-Verlag, 1984.

[CMMR89] P. Cox, H. Maitre, M. Minoux, and C. Ribeiro. Optimal matching of convex polygons. *Pattern Recognition Letters*, Vol 9 No 5:327–334, June 1989.

[CSD88] J.L. Crowley, P. Stelmaszyk, and C. Discours. Measuring image flow by tracking edge-lines. *Proc. 2nd Int. Conf. Computer Vision, Tarpon Springs, Florida*, pp 658–664, Dec. 1988.

[DF90] R. Deriche and O. Faugeras. Tracking line segments. *Proc. 1st European Conf. on Computer Vision, Antibes*, pp 259–268, April 1990.

[Enk88] W. Enkelmann. Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences. *Computer Vision, Graphics and Image Processing*, Vol.43:150–177, 1988.

[FB90] E. François and P. Bouthemy. Derivation of qualitative information in motion analysis. *Image and Vision Computing Journal*, Vol.8, No.4:279–287, Nov. 1990.

[FB91] E. François and P. Bouthemy. Multiframe-based identification of mobile components of a scene with a moving camera. *Proc. CVPR, Hawaii*, pp 166–172, June 1991.

[FLT87] O.D. Faugeras, F. Lustman, and G. Toscani. Motion and structure from motion from point and line matching. *Proc. 1st Int. Conf. on Computer Vision, London*, pp 25–34, 1987.

[Gam84] J.P. Gambotto. Correspondence analysis for target tracking in infrared images. *Proc. 7th Int. Conf. on Pattern Analysis and Machine Intelligence, Montreal, Vol I*, pp 526–529, 1984.

[Gel74] Arthur Gelb. *Applied Optimal Estimation*. MIT Press, 1974.

[Gor89] G. L. Gordon. On the tracking of featureless objects with occlusion. *Proc. Workshop on Visual Motion, Irving California*, pp 13–20, March 1989.

[HS81] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, Vol.17:pp 185–203, 1981.

[HW88] B.K.P Horn and E.J. Weldon. Direct methods for recovering motion. *Int. Jal of Computer Vision*, Vol.2:51–76, 1988.

[JJ88] C.P. Jerian and R. Jain. *Structure from motion: A critical analysis of methods*. Technical Report CSE-TR-04-88, University of Michigan, 1988.

[LH86] Y.C. Liu and T.S. Huang. Estimation of rigid body motion using straight line correspondences. *Proc. of IEEE Workshop on Motion: Representation and Analysis, Kiawah Island Ressort, Charleston, South Carolina*, pp 47–52, May 1986.

[MB92] F. Meyer and P. Bouthemy. Estimation of time-to-collision maps from first order motion models and normal flows. *Proc. 11th Conf. on Pattern Recognition, The Hague*, 1992.

[MSA85] A. Mitiche, S. Seida, and J.K. Aggarwal. Determining position and displacement in space from images. *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, San Francisco*, pp 504–509, June 1985.

[NY86] S. Negahdaripour and A. Yuille. Direct passive navigation II: analytical solution for quadratic patches. *AI Memo 876, MIT AI Laboratory*, March 1986.

[PR90] S. Peleg and H. Rom. Motion based segmentation. *Proc. 10th IEEE Conf. on Pattern Recognition, Atlantic City*, pp 109–113, 1990.

[PS85] F.P. Preparata and M.I. Shamos. *Computational Geometry, An Introduction.* Springer-Verlag, 1985.

[SD91] 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 New-Jersey*, pp 256–261, October 1991.

[Sha86] H. Shariat. The motion problem: how to use more than two frames. *Ph-D Thesis, Institute for Robotics and Intelligent Systems, School of Engineering, Univ. of Southern California, USC-IRIS Report No 202*, Oct. 1986.

[SJ87] I. K. Sethi and R. Jain. Finding trajectories of feature points in a monocular image sequence. *IEEE Trans. PAMI*, Vol. PAMI-9, No 1:56–73, January 1987.

[SM82] R.J. Schalkoff and E.S. McVey. A model and tracking algorithm for a class of video targets. *IEEE Trans. PAMI*, Vol.PAMI-4, No.1:2–10, Jan. 1982.

[ST90] G. Sandini and M. Tistarelli. Active tracking strategy for monocular depth inference over multiple frames. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.12, No.1:13–27, Jan. 1990.

[WD84] K. Wall and P.E. Danielsson. A fast sequential method for polygonal approximation of digitized curves. *Computer Vision, Graphics and Image Processing*, Vol. 28:220–227, 1984.

[Wib71] D.M. Wiberg. *State Space and Linear Systems.* Mc Graw-Hill, 1971.

[WKS87] A.M. Waxman, B. Kamgar-Parsi, and M. Subbarao. Closed-form solutions to image flow equations for 3D-structure and motion. *Int. Journal of Computer Vision*, No 3:239–258, 1987.

[YM86] Y. Yasumoto and G. Medioni. Robust estimation of three-dimensional motion parameters from a sequence of image frames using regularization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No.4:464–471, July 1986.

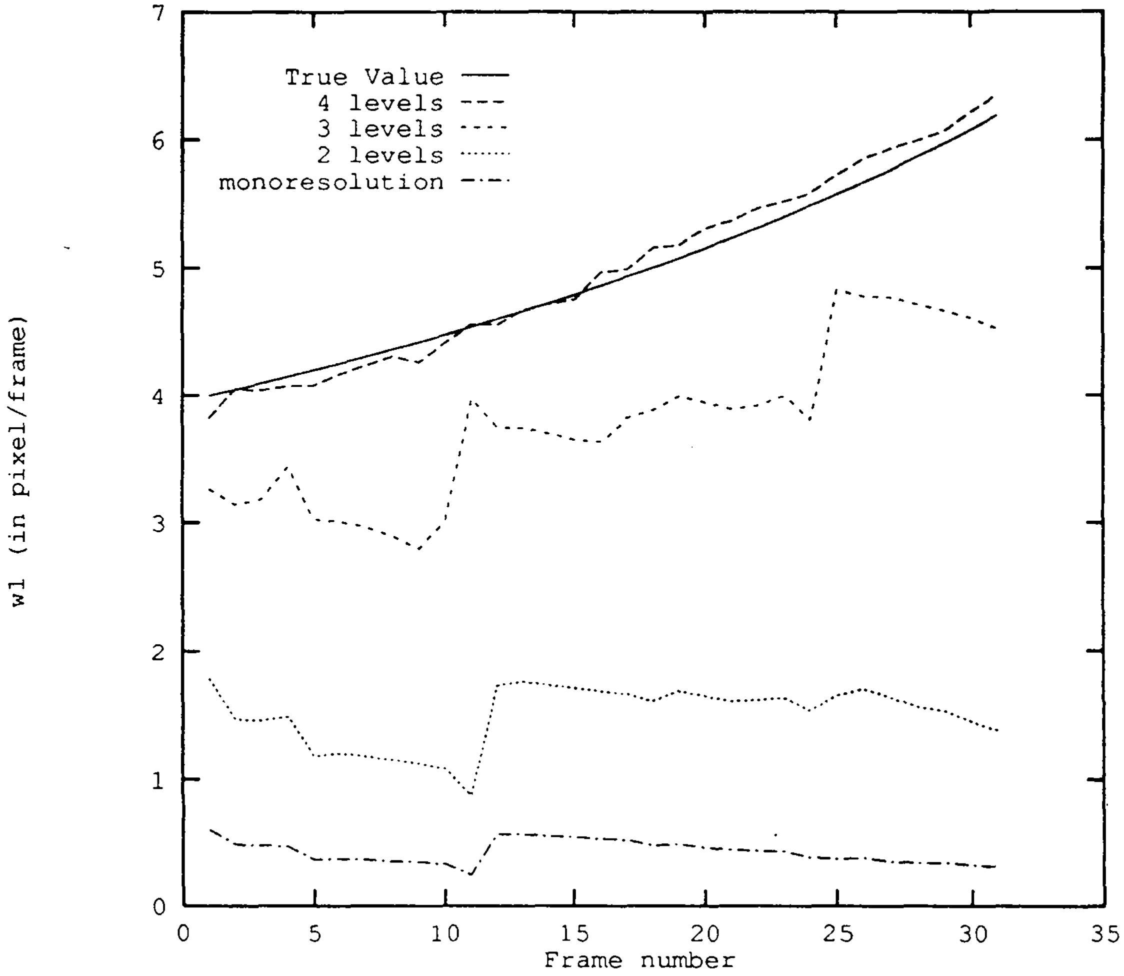

Figure 7: *True value, multiresolution estimate for different pyramid levels, and monoresolution estimate, of the motion parameter :* $\mathbf{w}_1$.

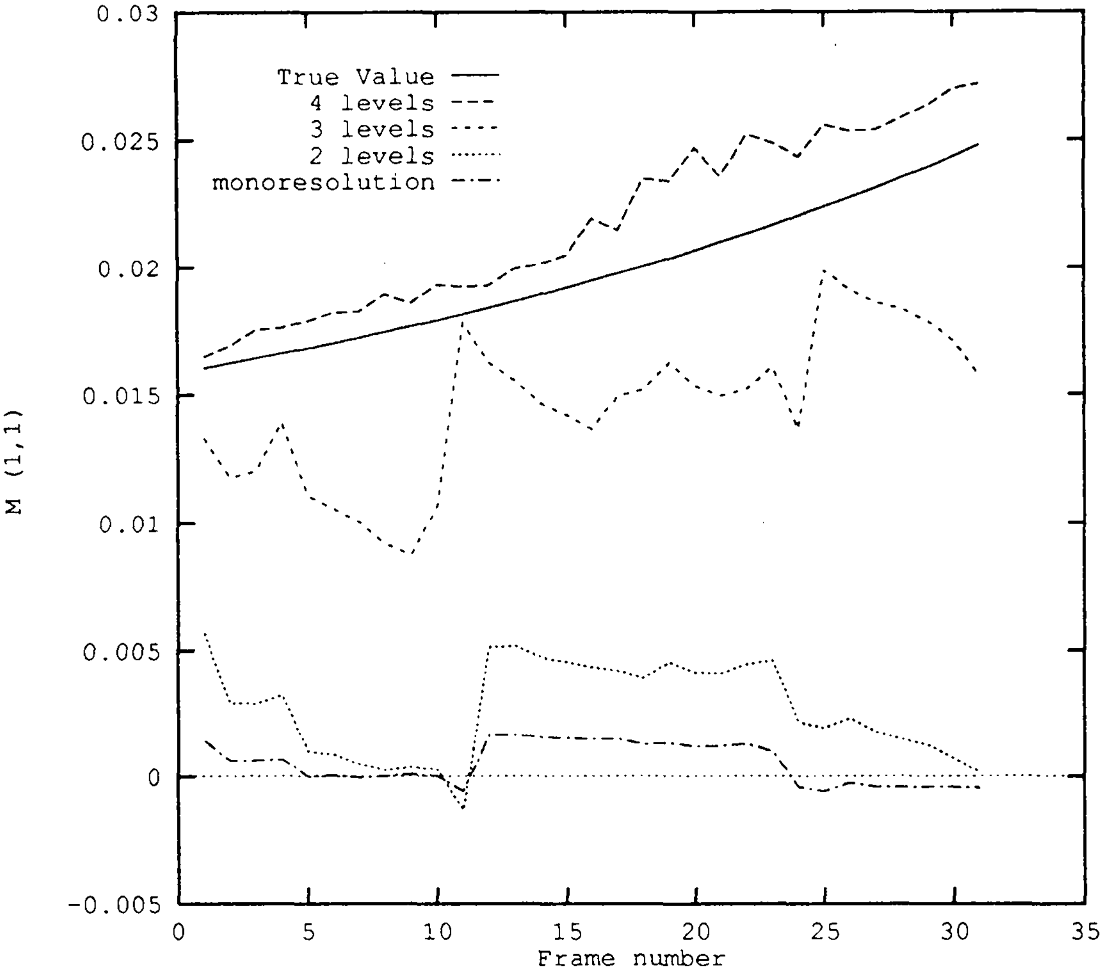

Figure 8: *True value, multiresolution estimate for different pyramid levels, and monoresolution estimate, of the motion parameter :* $\mathbf{M}_{1,1}$.

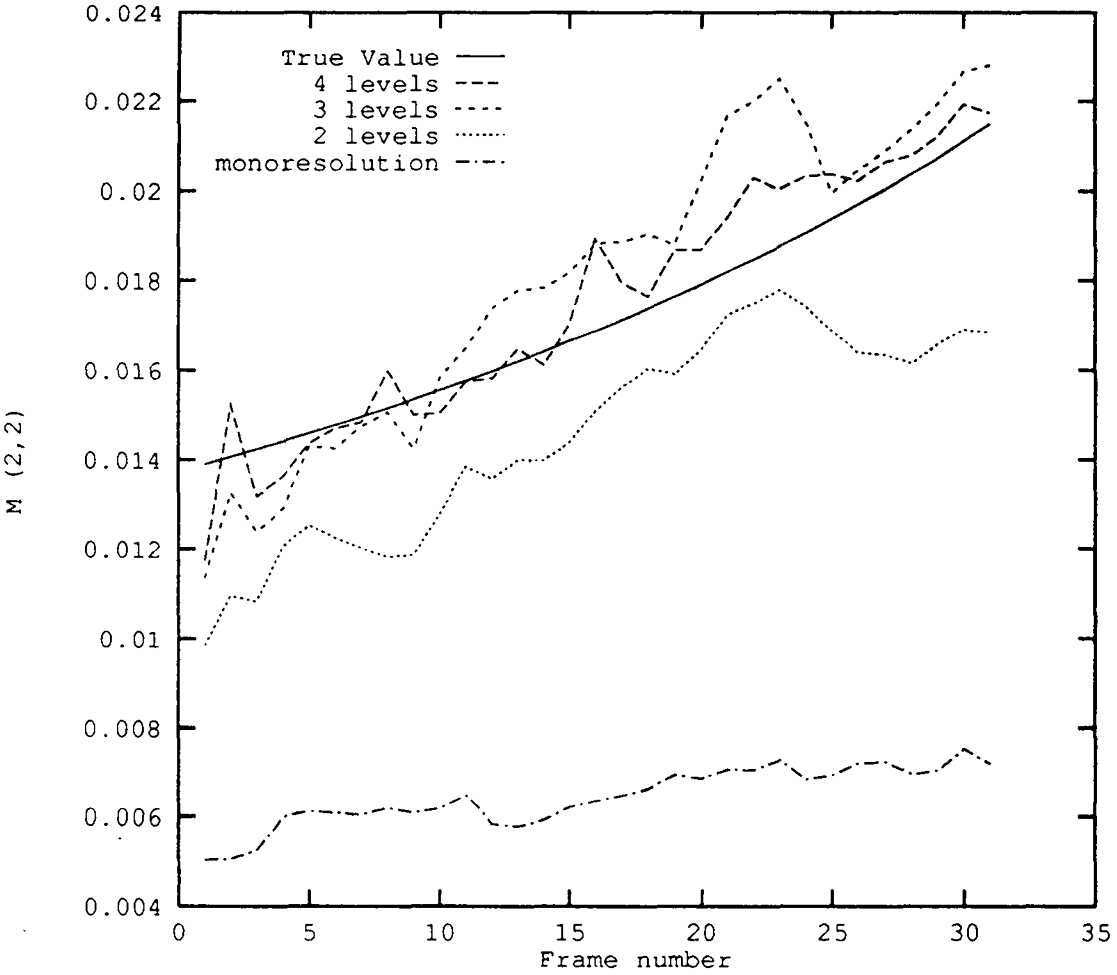

Figure 9: *True value, multiresolution estimate for different pyramid levels, and monoresolution estimate, of the motion parameter :* $\mathbf{M}_{2,2}$.

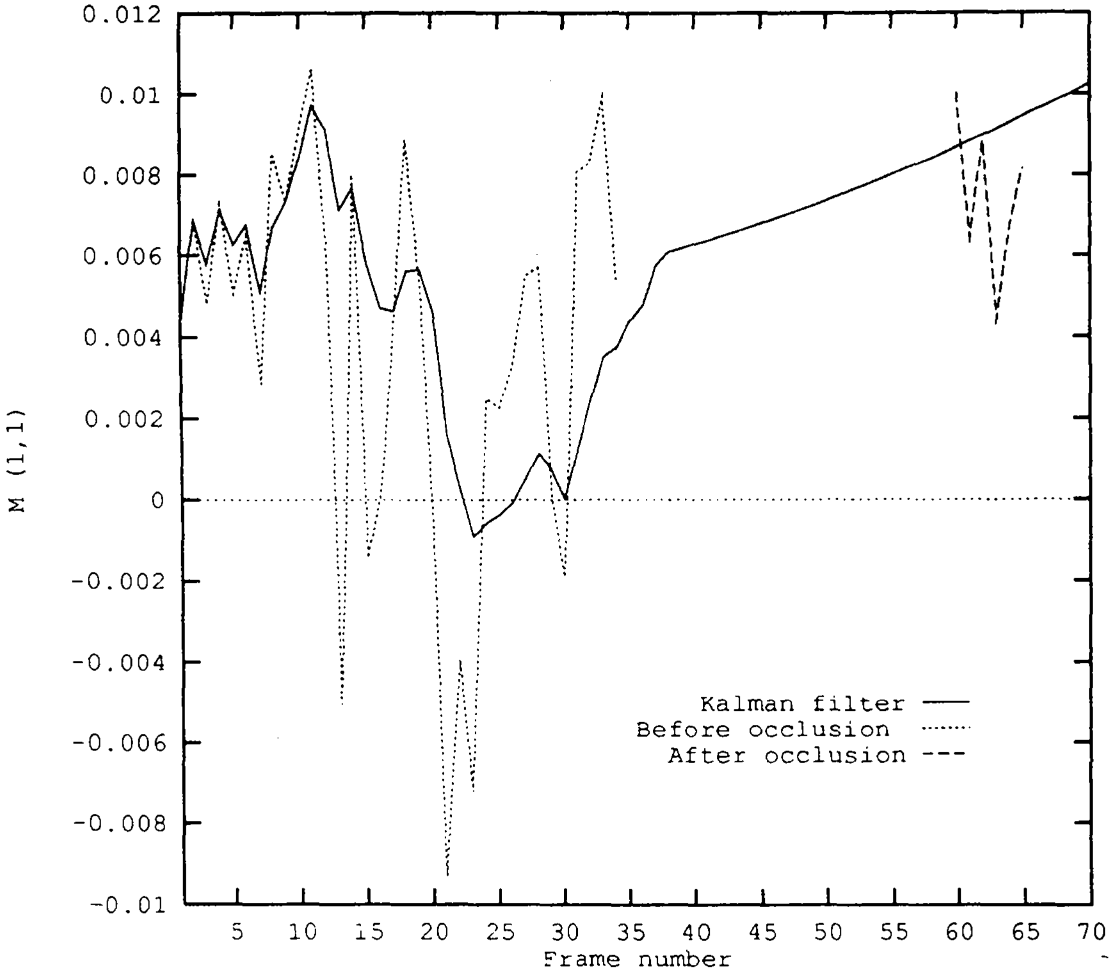

Figure 10: *Measurements before and after occlusion, and Kalman estimates of the motion parameter* $\mathbf{M}_{1,1}$.

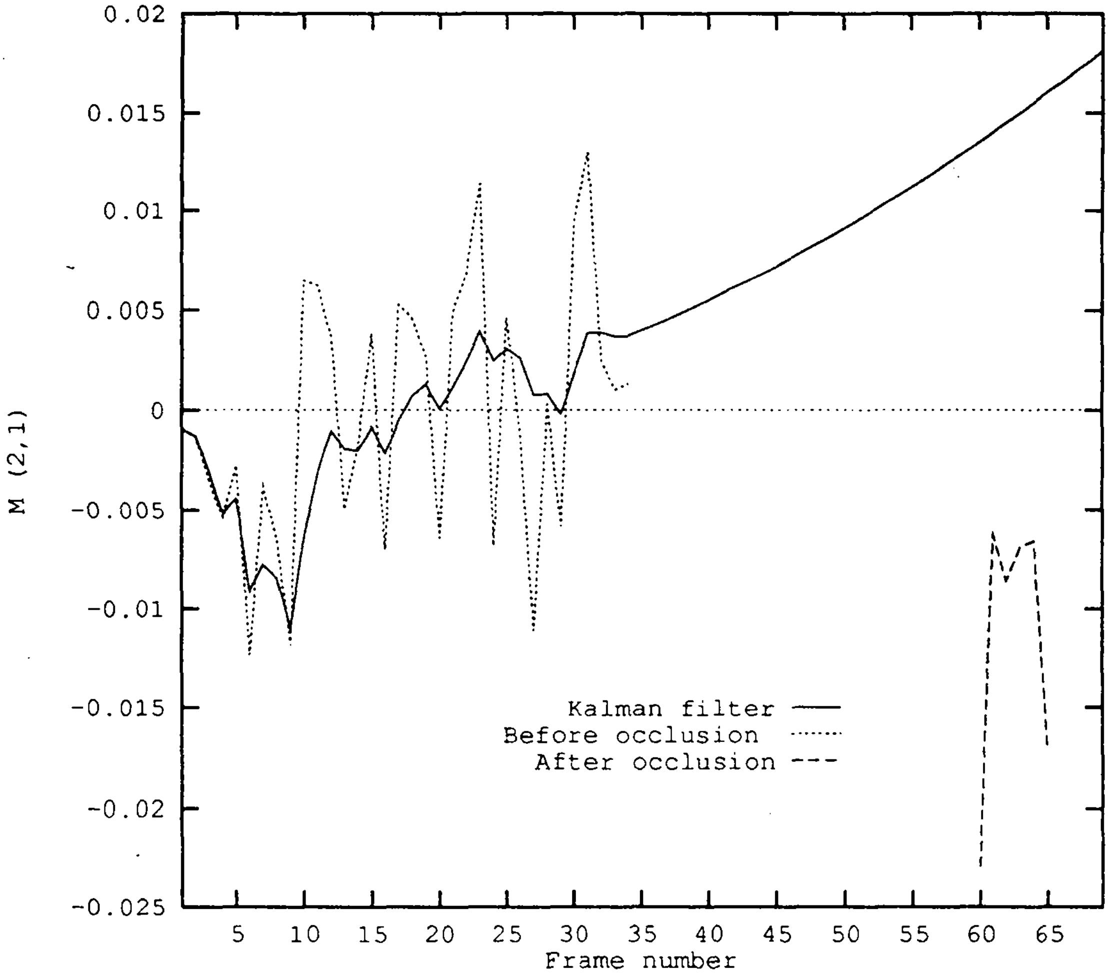

Figure 11: *Measurements before and after occlusion, and Kalman estimates of the motion parameter* $\mathbf{M}_{2,1}$.

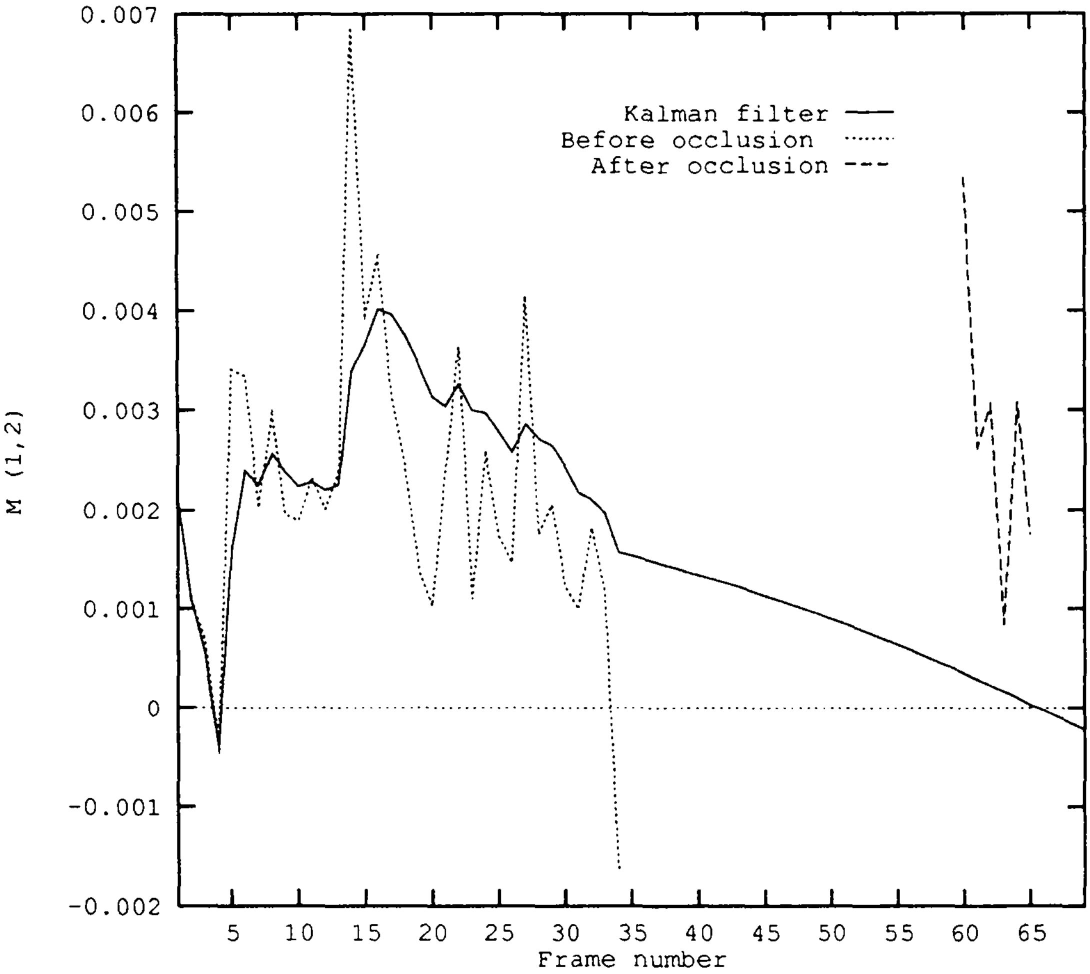

Figure 12: *Measurements before and after occlusion, and Kalman estimates of the motion parameter* $\mathbf{M}_{1,2}$.

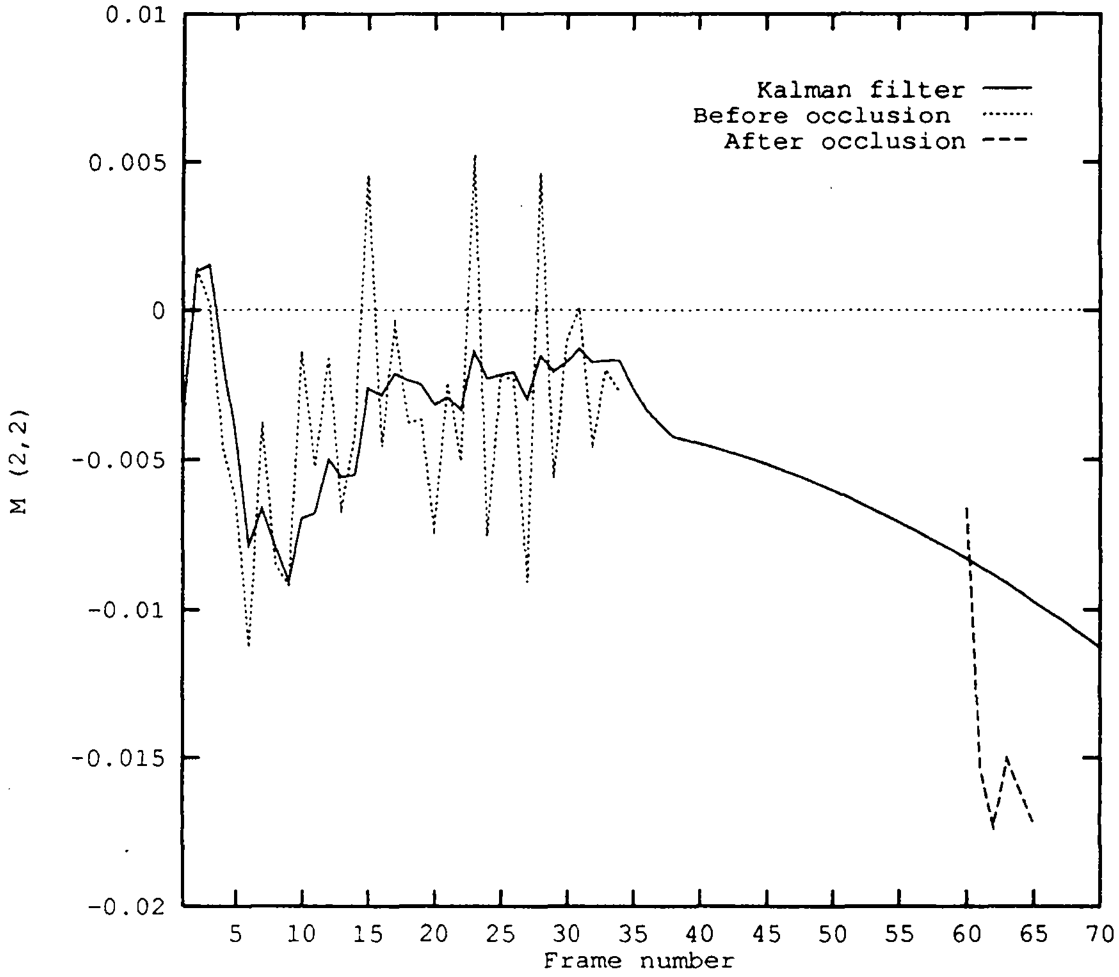

Figure 13: *Measurements before and after occlusion, and Kalman estimates of the motion parameter* $\mathbf{M}_{2,2}$.

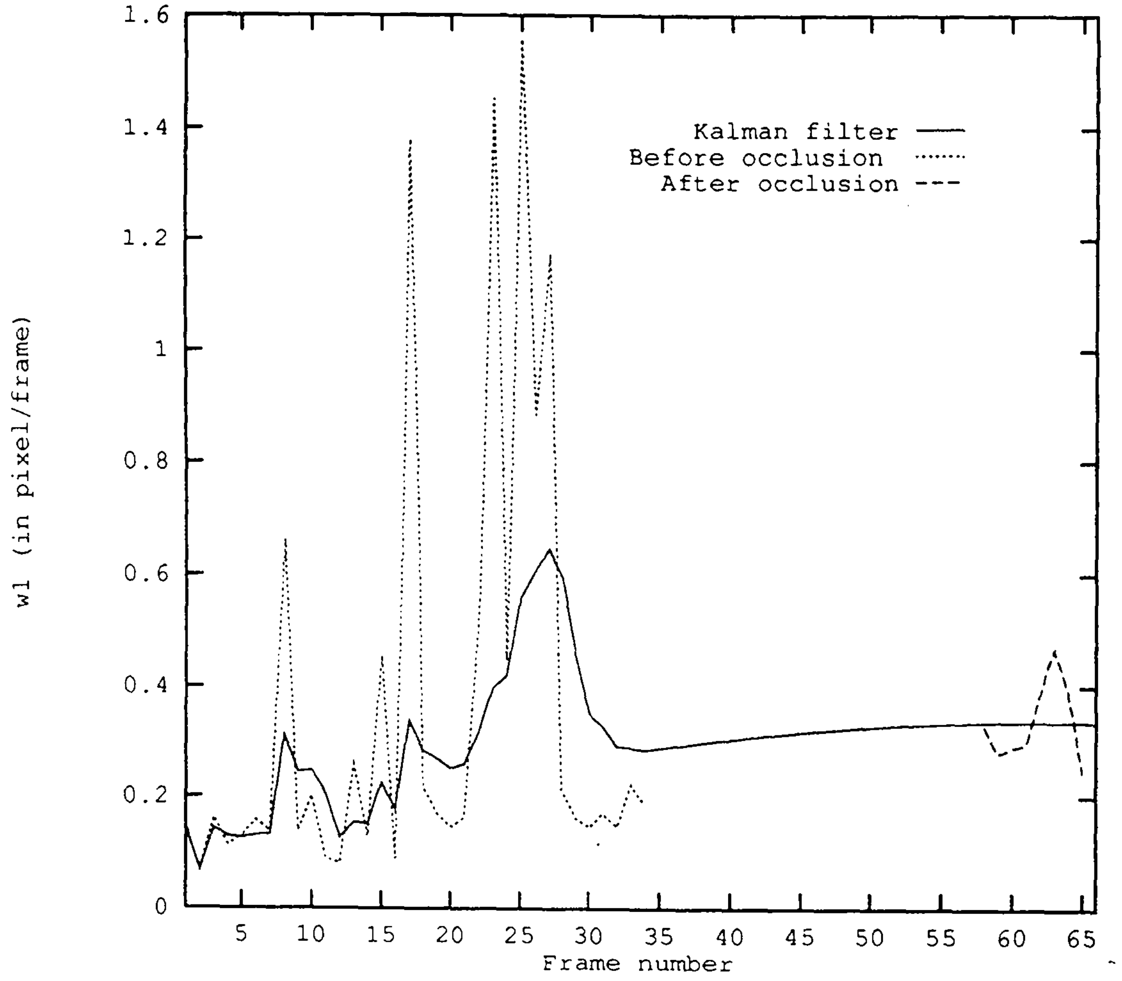

Figure 14: *Measurements before and after occlusion, and Kalman estimates of the motion parameter* $\mathbf{w}_1$.

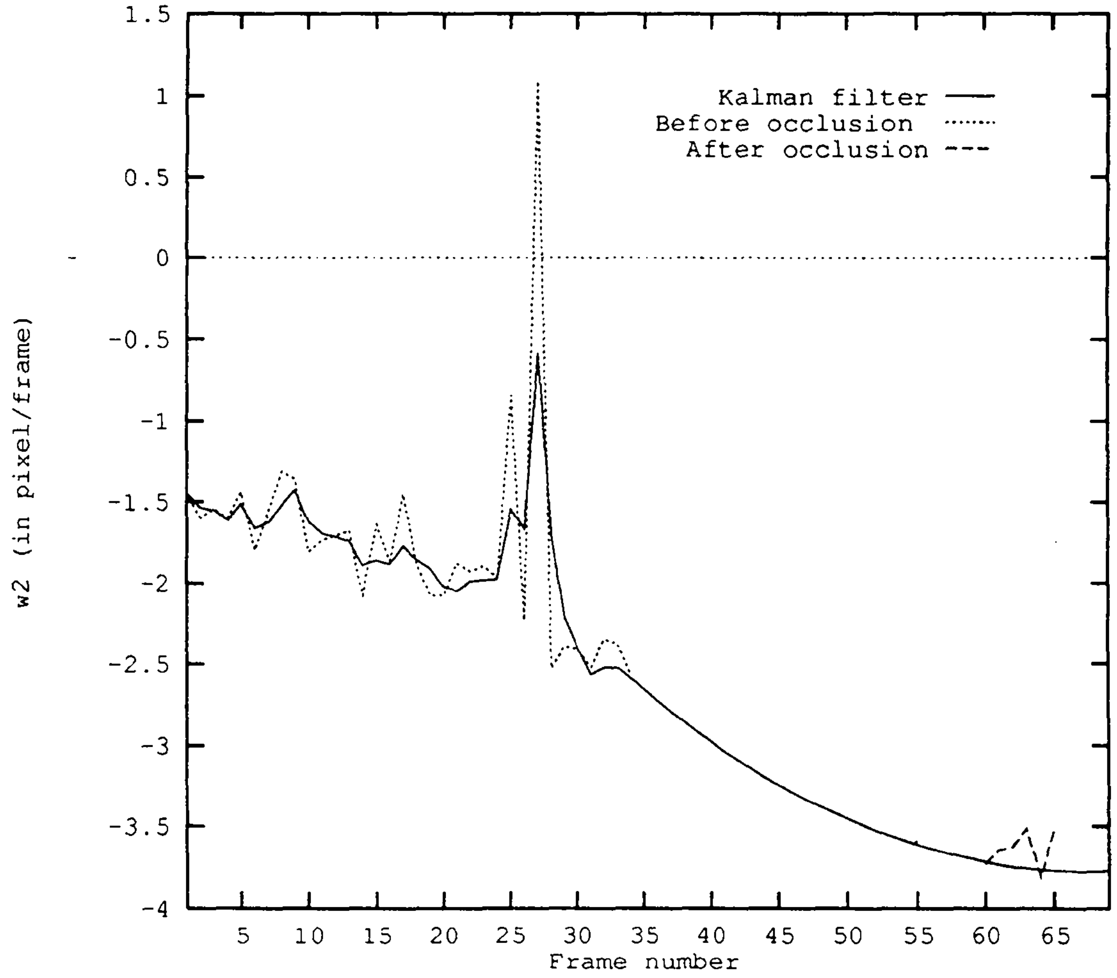

Figure 15: *Measurements before and after occlusion, and Kalman estimates of the motion parameter* $\mathbf{w}_2$.

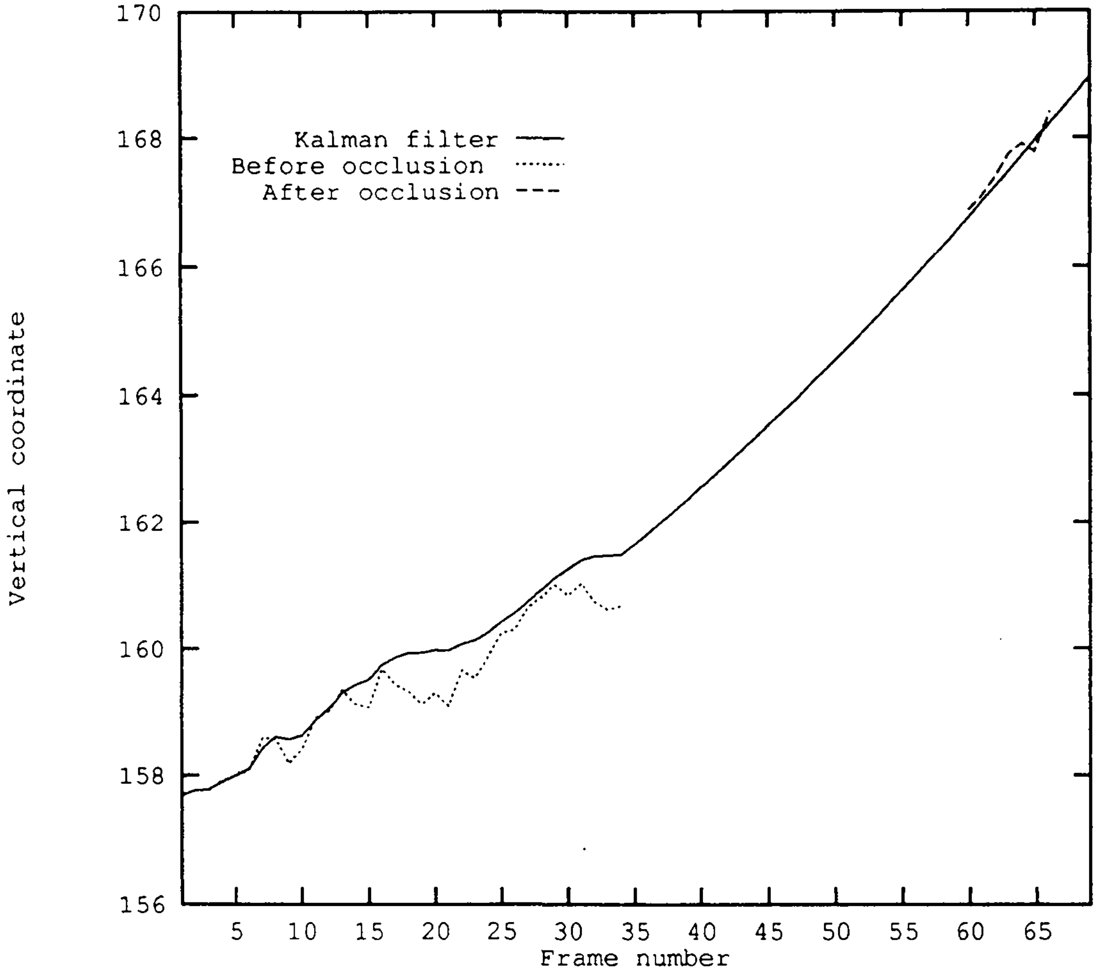

Figure 16: *Measurements before and after occlusion, and Kalman estimates of the vertical coordinate of the centroid of the tracked region*

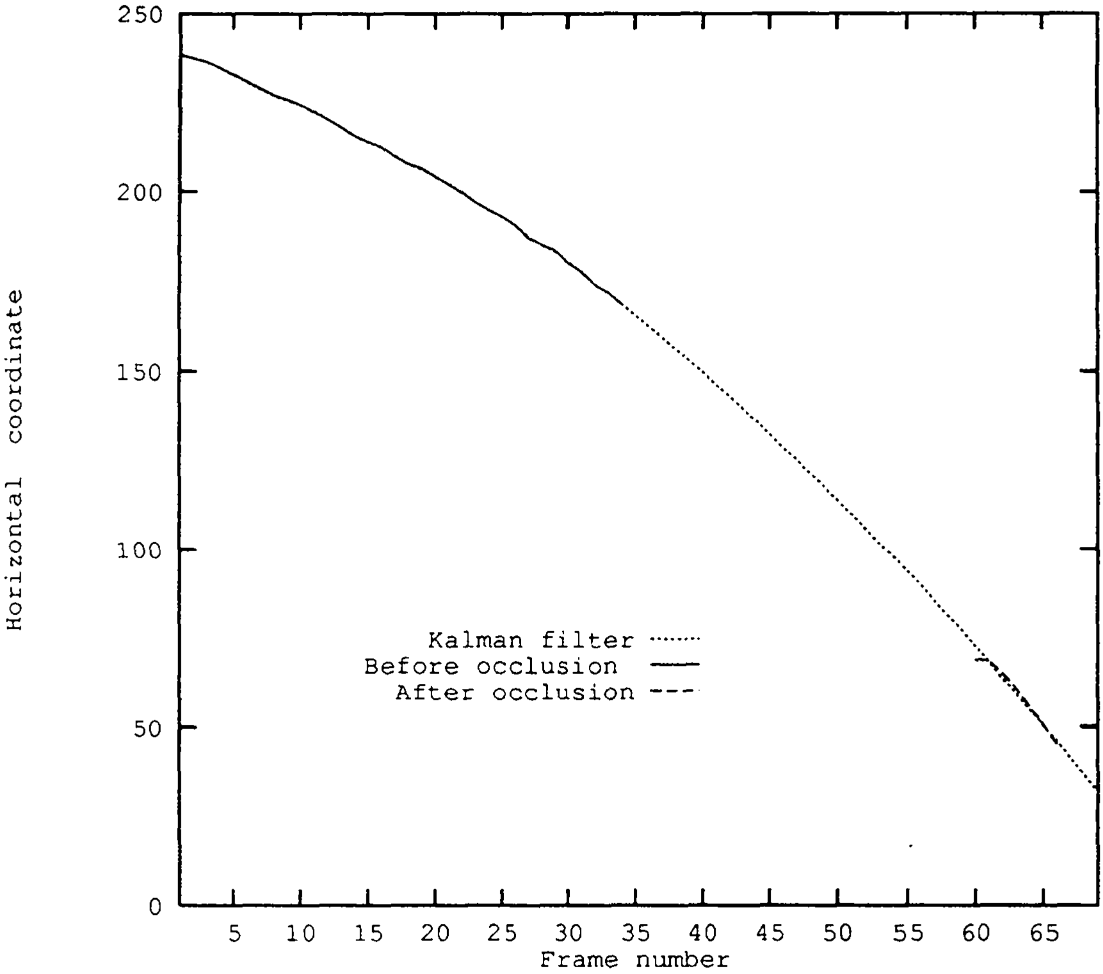

Figure 17: *Measurements before and after occlusion, and Kalman estimates of the horizontal coordinate of the centroid of the tracked region*

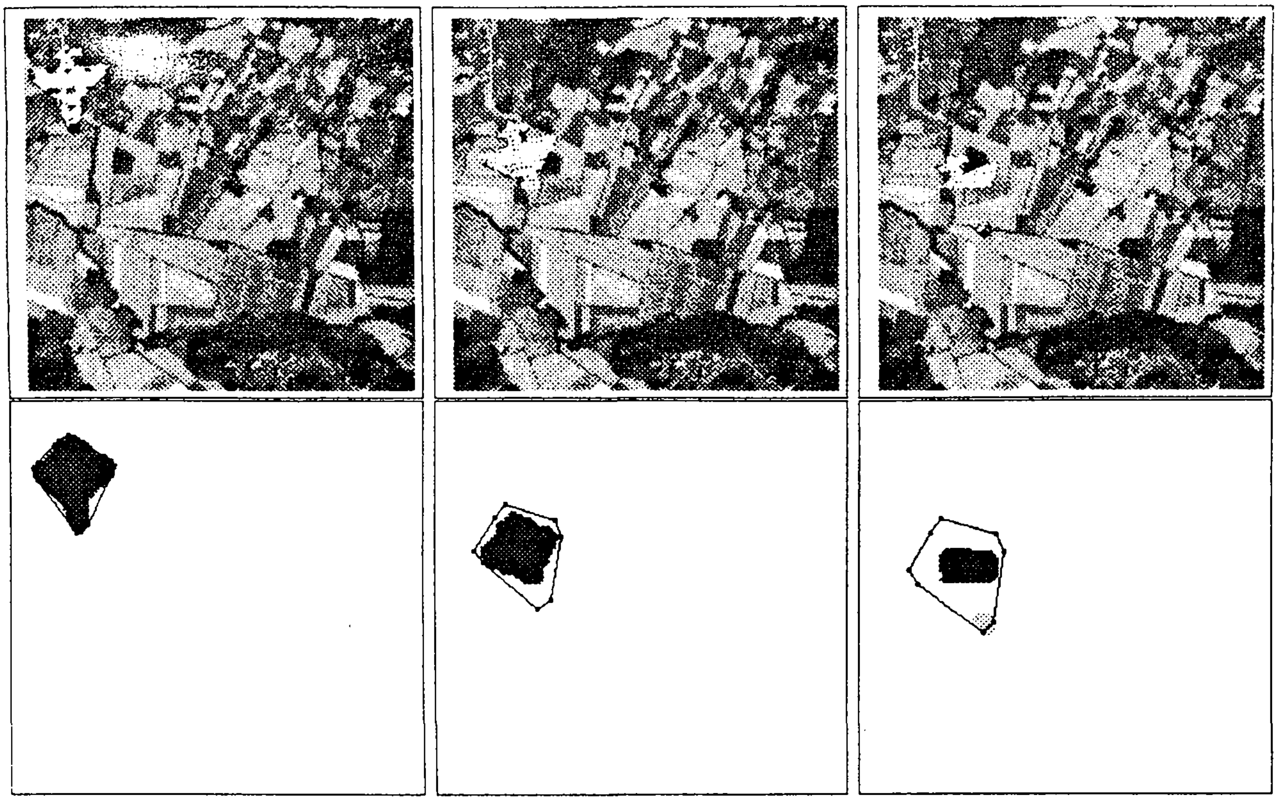

Figure 18: *Sequence airplane : top : original images, bottom : tracking (black polygon) superimposed on the motion-based segmentation, at time t_1, t_{16} and t_{23}.*

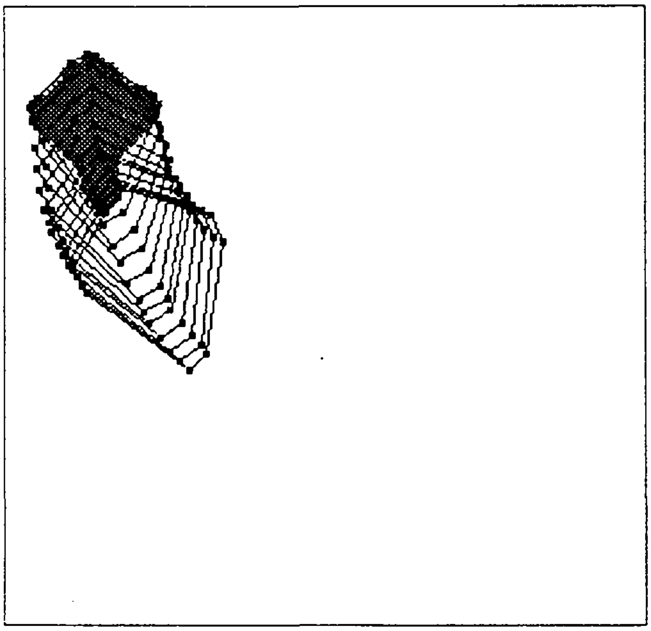

Figure 19: *Sequence airplane : the region trajectory in the image plane from time t_1 to time t_{23}*

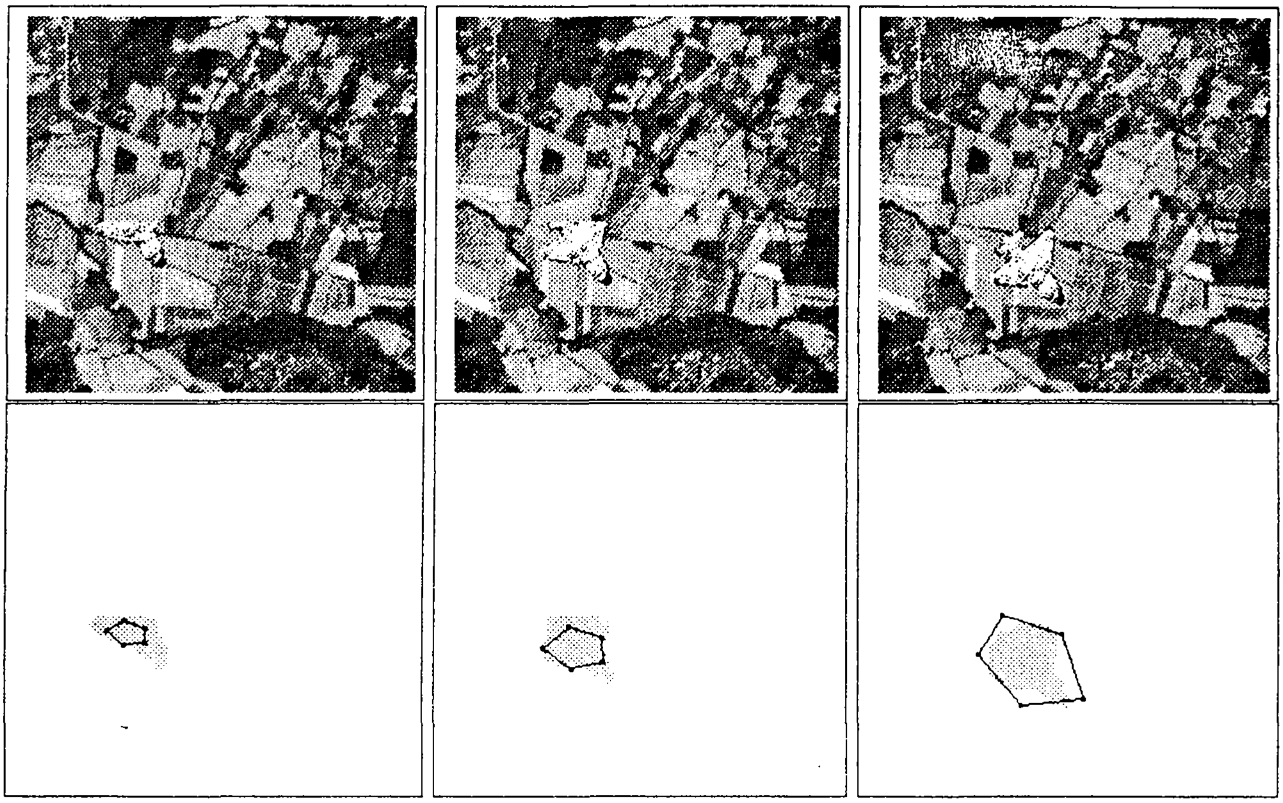

Figure 20: *Sequence airplane ; top : original images, bottom : tracking (black polygon) superimposed on the motion-based segmentation, at time* t_{30}, t_{35} *and* t_{41}.

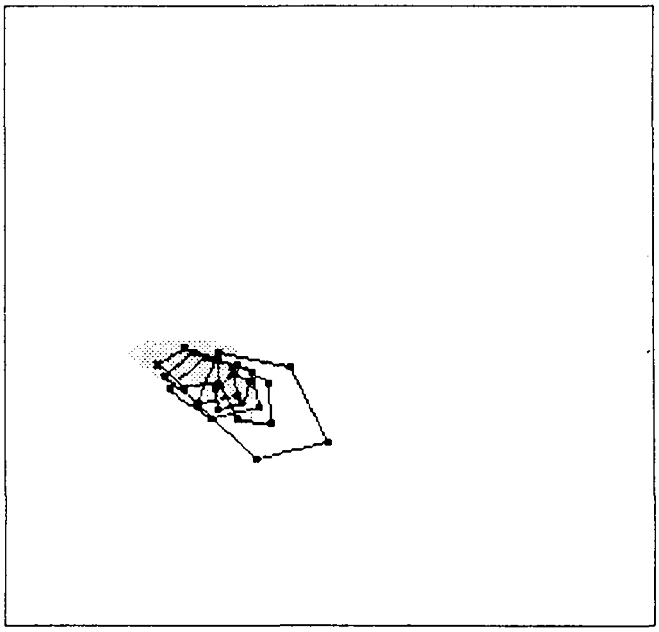

Figure 21: *Sequence airplane : the region trajectory in the image from time* t_{30} *to time* t_{41}

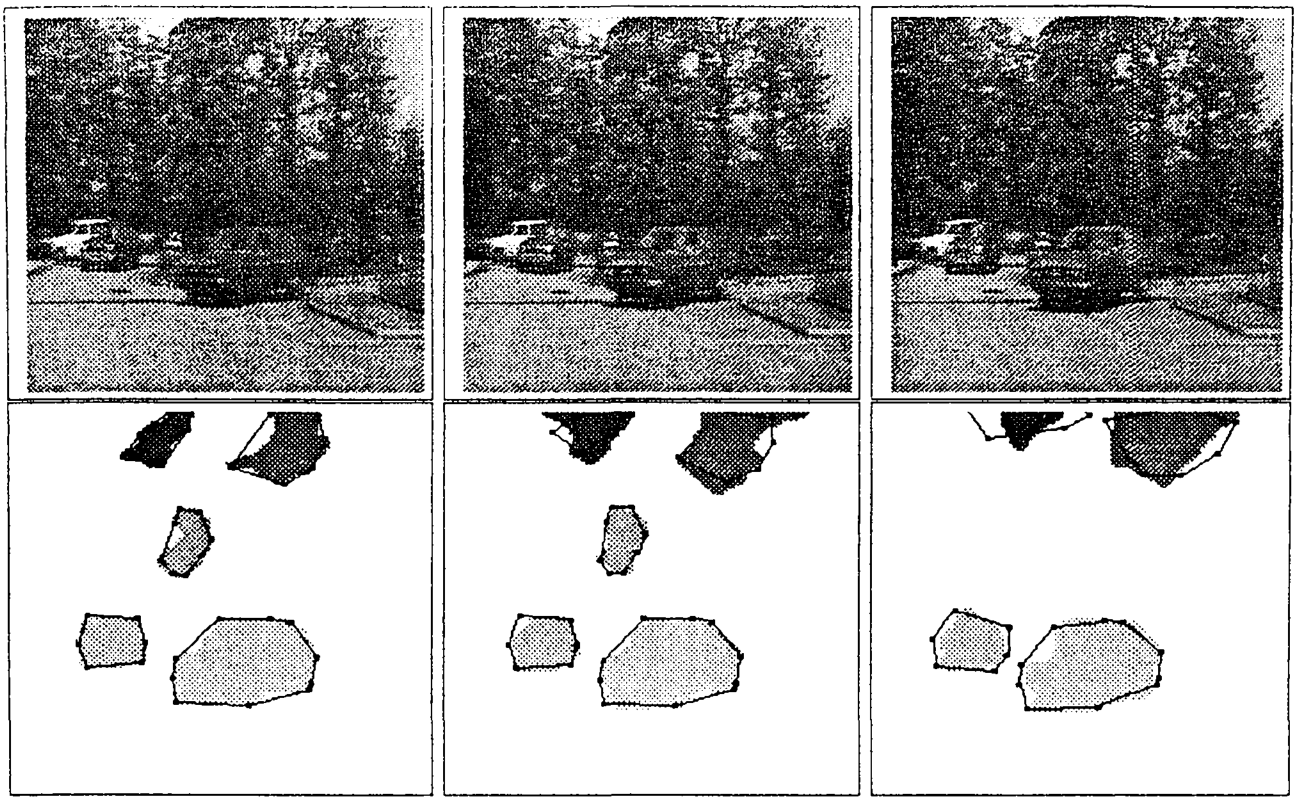

Figure 22: *Sequence parking ; top : original images, bottom : tracking (black polygons) superimposed on the motion-based segmentation, at time t_1, t_4 and t_9.*

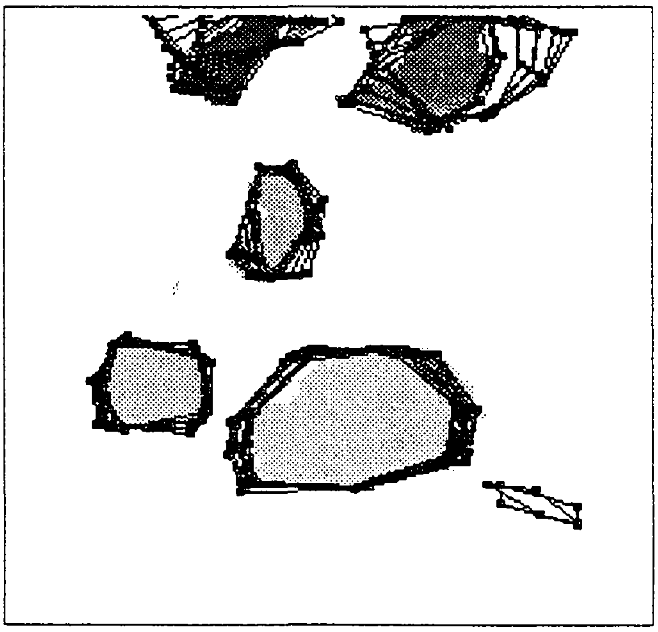

Figure 23: *Sequence parking ; the region trajectory in the image from time t_1 to time t_9*

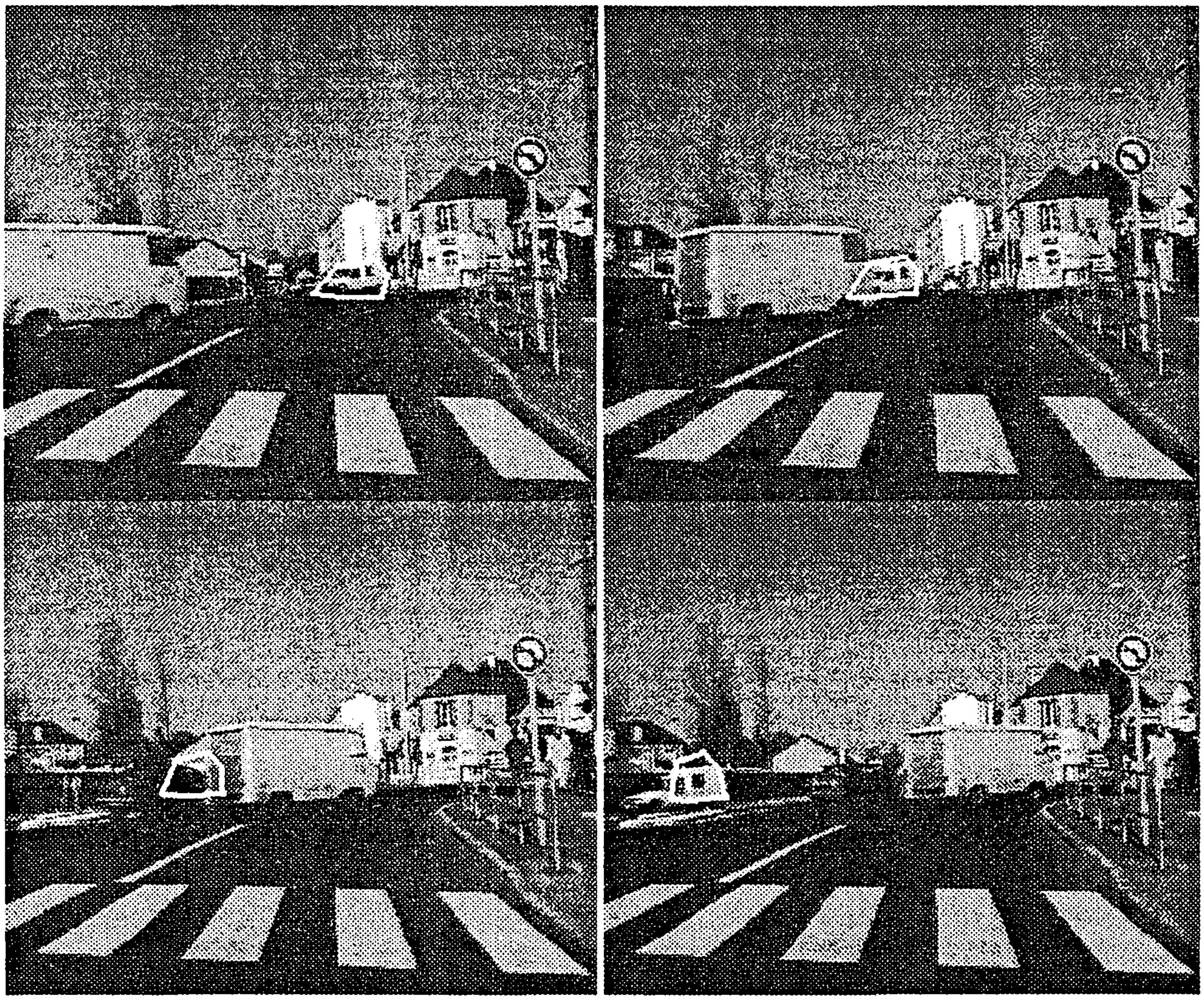

Figure 24: Left to right, top to down : original images with superimposed tracked region (white polygon) at instants t_{24}, t_{38}, t_{52}, t_{66}

Figure 25: Left to right. top to down : segmented images at instants t_{24}, t_{38}, t_{52}, t_{66}

LISTE DES DERNIERES PUBLICATIONS INTERNES PARUES A L'IRISA

PI 655 DIFFUSION ON SCALABLE HONEYCOMB NETWORKS
Dominique DESERABLE
Avril 1992, 24 pages.

PI 656 CAUSALITY ORIENTED SHARED MEMORY FOR DISTRIBUTED SYSTEMS
Michel RAYNAL, Masaaki MIZUNO, Mitch NEILSEN
Avril 1992, 8 pages.

PI 657 ALGORITHMES PARALLELES POUR L'ANALYSE D'IMAGE PAR CHAMPS MARKOVIENS
Etienne MEMIN, Fabrice HEITZ
Mai 1992, 74 pages.

PI 658 MULTISCALE SIGNAL PROCESSING : ISOTROPIC RANDOM FIELDS ON HOMOGENEOUS TREES
Bernhard CLAUS, Ghislaine CHARTIER
Mai 1992, 28 pages.

PI 659 ON THE COVARIANCE-SEQUENCE OF AR-PROCESSES. AN INTERPOLATION PROBLEM AND ITS EXTENSION TO MULTISCALE AR-PROCESSES
Bernhard CLAUS, Albert BENVENISTE
Mai 1992, 36 pages.

PI 660 EXCEPTION HANDLING IN COMMUNICATING SEQUENTIAL PROCESSES DESIGN, VERIFICATION AND IMPLEMENTATION
Jean-Pierre BANATRE, Valérie ISSARNY
Mai 1992, 38 pages.

PI 661 REACHABILITY ANALYSIS ON DISTRIBUTED EXECUTIONS
Claide DIEHL, Claude JARD, Jean-Xavier RAMPON
Juin 1992, 18 pages.

PI 662 RECONSTRUCTION 3D DE PRIMITIVES GEOMETRIQUES PAR VISION ACTIVE
Samia BOUKIR, François CHAUMETTE
Juin 1992, 40 pages.

PI 663 FILTRES SEMANTIQUES EN CALCUL PROPOSITIONNEL
Raymond ROLLAND
Juin 1992, 22 pages.

PI 664 REGION-BASED TRACKING IN AN IMAGE SEQUENCE
François MEYER, Patrick BOUTHEMY
Juin 1992, 50 pages.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique

ISSN 0249-6399