

***Suivi de primitives complexes sur une séquence d'images***

Benedicte Bascle, Patrick Bouthemy †, Rachid Deriche, Francois Meyer †

**N° 2428**

Decembre 1994

PROGRAMME 4

Robotique,  
image  
et vision ***Rapport  
de recherche*****1994**

## Suivi de primitives complexes sur une séquence d'images

Benedicte Bascle, Patrick Bouthemy †, Rachid Deriche, Francois Meyer †

Programme 4 — Robotique, image et vision  
Projet Robotvis

Rapport de recherche n° 2428 — Decembre 1994 — 22 pages

**Résumé :** Cet article présente une nouvelle approche pour le suivi de primitives complexes sur une séquence d'images. Notre approche intègre les principes du suivi de contours par la méthode des contours actifs et l'analyse de mouvement à base de régions. Tout d'abord, un contour actif effectue le suivi du contour de la région et effectue la segmentation de la région. Ensuite le mouvement de la région extraite est estimé par une analyse dense et multi-résolution du mouvement des points de la région, basée sur les gradients spatio-temporel de l'image. Enfin, l'estimation du mouvement obtenue est utilisée (via un filtrage temporel) pour prédire la position de la région dans l'image suivante. Cette prédiction permet d'initialiser le contour actif dans l'image suivante, et donc ainsi de guider l'algorithme de suivi vers la zone de l'image où la région se trouve probablement. Ainsi, ces deux approches complémentaires collaborent réellement et échangent des informations de manière à effectuer un meilleur suivi. L'algorithme fournit aussi la trajectoire des points de la région, et une estimation de son mouvement apparent. La méthode est illustrée par des résultats expérimentaux sur des images réelles.

**Mots-clé :** Suivi, contours actifs, analyse du mouvement, champ des vitesses, filtrage, prédiction

*(Abstract: pto)*

†IRISA/INRIA, Campus Universitaire de Beaulieu, 35042 Rennes Cedex. This research was funded by the GDR-PRC "Man-Machine Communication" Computer Vision program. Cet article est paru dans les Actes d'ICPR'94, Jerusalem, Oct. 1994. Email: bascle@sophia.inria.fr

# Tracking complex primitives in an image sequence

**Abstract:** This paper describes a new approach to track complex primitives along image sequences. This approach integrates the ideas of snake-based contour tracking and region-based motion analysis. First, a snake tracks the region outline and performs segmentation. Then the motion of the extracted region is estimated by a dense analysis of the apparent motion over the region, using spatio-temporal image gradients. Finally, this motion measurement is filtered to predict the region location in the next frame, and thus to guide (i.e. to initialize) the tracking snake in the next frame. Therefore, these two approaches collaborate and exchange information to overcome the limitations of each of them. The combined algorithm performs object tracking and estimates both the apparent motion and the trajectory of the object in the image. The method is illustrated and validated by experimental results on real images.

**Key-words:** Tracking, snakes, motion analysis, velocity field, filtering, prediction

# 1 Introduction

Tracking an object through a long period of time is an important issue in dynamic scene analysis: it provides for instance a good basis for the estimation of 3D motion and/or structure [Sha86] [BC86]. First investigations were concerned with the tracking of points [SJ87] and edge segments [CSD88] [DF90] [GCDVF93]. However, this data is sparse and sensitive to occlusion. Consequently, interest was driven to the tracking of more complex and global primitives, such as regions or entire objects. Among the latter approaches, two large classes of algorithms emerge: snake-based contour tracking [KWT87] [CB92] [CAS92] [CB92] [TS92] [BD93b] and region-based tracking [BC86] [MB92b] [Sch82]. Besides, some authors investigated model-based tracking, where a 3D polyhedral model of the object of interest is known [KDTN92] [Low92] [SD91]. However, to track objects of various shapes, as considered in this paper, snake-based contour tracking and motion-based region segmentation are more appropriate methods. The first method, based on snakes, is particularly well adapted to tracking due to the active behavior of snakes (also called active contours), provided the motion is slow enough. It is also quite precise in the extraction of object boundaries. Still, the approach is not so robust to large displacements and is limited to the information given by edges (no information about the interior of the region is used). In particular, objects with a lot of texture are not well handled. Snakes also require an appropriate initialization. Moreover, snakes give rough estimations of boundary motion (if they do), since they rely only on the edge information (not on the whole region) and perform no filtering on the motion estimations. On the contrary, the second type of methods, motion-based region segmentation and region-based tracking, exploits the full region information, and estimates region motion quite precisely. Moreover, filtering is applied to smooth the estimated motions. Besides, this method is not very sensitive to the magnitude of the displacement, nor to partial occlusion. Furthermore, motion-based region segmentation is able to detect moving objects in the image and thus do not require any initialization. Nevertheless, segmentation gives rather rough estimations of region boundaries. As can be seen, these two methods have complementary strengths and shortcomings.

This paper describes an original combination of these two types of approaches, which overcomes some of the limitations of each of them. First, the moving object is detected by a motion-based segmentation algorithm [BF93] [BL93]. Then a snake-based contour tracking algorithm is used to track and segment the object along the image sequence. It is based on B-spline snakes with motion constraints [BD93b] [BD93a]. Indeed B-splines can describe various shapes. Moreover motion constraints improve the robustness of the tracking algorithm. Thereafter, the motion of the region delineated by the snake is estimated using a region-based motion analysis approach. It is derived from the region-based tracking approach proposed by [MB92a]. It consists in a dense and multi-resolution estimation of an affine velocity field over the region. Additionally the obtained motion measurements are smoothed using Kalman filtering. This temporal filtering is further applied to predict the expected position of the region in the next frame and thus to initialize the snake tracker in the next image. As a result, snake-based contour tracking and region-based motion analysis strongly cooperate in this combined approach: motion estimation relies on the snake-based segmentation, whereas

it provides to the tracking snake a prediction of the region location. Such a joined approach has rather good tracking performances, as illustrated by the experimental results presented in the last section of this paper.

The paper is organized as follows: the first section describes the global framework of our tracking approach. The second and third sections present in more detail the modules of the approach: respectively object segmentation and tracking, and motion estimation and prediction. Finally the last section presents experimental results to illustrate and to validate the approach.

## 2 Outline of the tracking approach

This section describes our tracking approach that combines snake-based contour tracking and region-based motion analysis. The whole experimental platform that has been developed is summarized in Figure 1. Below are presented the main steps of this approach and the interactions between the different modules:

**INITIALIZATION ON FIRST IMAGE OF THE SEQUENCE** As snake-based tracking requires an initialization, motion-based segmentation [BF93] is used to initialize tracking in the first image of the sequence. Indeed motion-based segmentation can detect a moving object in the image and generate a rough mask of this object. Then a first outline of the object 2D projection can be deduced from this mask using classical edge detection and linking. After B-spline approximation, this estimation of the object contour can be improved by a free deformable B-spline curve, so as to better fit image edges [BD92]. This provides an initialization to the snake-based tracking algorithm. However, since motion-based segmentation is rather unprecise, it is applied only once to initialize the snake on the first image of the sequence, and the segmentation of the following images of the sequence is achieved by a snake-based tracking algorithm.

**CONTOUR TRACKING BY A SNAKE WITH MOTION CONSTRAINTS** Then the region outline is simultaneously extracted and tracked along the image sequence by a snake. A snake is a deformable curve representing the contour. From its starting point, its position is optimized in the image by energy minimization, so that the equilibrium state of the snake corresponds to the position of the image contour. Here the chosen snake is a deformable B-spline curve with motion constraints [BD93b] [BD93a]. Indeed B-splines can describe various types of shapes. They also have nice smoothness properties [BBB87]: it is useful to regularize the noisy edge information issued from the image. Besides, motion constraints are applied to stabilize the snake, so that the convergence becomes quicker and robuster. These motion constraints consist in imposing a motion model to the displacement of the deformable curve. Thus the curve is matched globally to the object boundary in the next frame, with the result that the point-to-point correspondences between the curves are regular along them. This is important if the pointwise trajectography of the object in the image is to be recorded. Indeed fully free deformable curves tend to present fallacious sliding

effects of the curve along the image contour, which do not correspond to true object motion but to snake internal dynamics: they are due to the local fitting behavior of the unconstrained snake which causes local dilation and shrinking. And whereas these effects are almost negligible for small displacements, they can become disturbing for large motions. It can be noted that other solutions than motion constraints have been proposed to treat this problem. For example, Cohen *et al* [CAS92] use high curvature points as landmarks to guide the matching process. With respect to the same problem, Curwen and Blake [CB92] impose "weak" shape assumptions on the deformable contour (up to an affine transform). In this paper, a 2D motion model is enforced on the snake, as proposed by [BD93b] [BD93a]. It is affine, since such a 2D displacement model can reasonably approximate the 2D motion of a 3D rigid object if perspective is weak. Then these motion constraints are relaxed in order to refine the extracted curve (to deal with non-rigid deformations for instance). Indeed, during this refinement step, sliding effects are likely to be small, since the displacement model should be able to describe the largest part of the curve displacement. Thus region segmentation and tracking is performed by a B-spline snake with affine-plus-deformable motion. Such an approach gives good instantaneous estimations of the region outlines in the images of the sequence. However, these estimations can sometimes be noisy. Therefore temporal smoothing - achieved by Kalman recursive filtering - is used to reduce the noise. Thus the trajectories of the control points parameterizing the boundary of the region (estimated by the B-spline snake) are filtered using a decoupled  $\alpha, \beta$  tracker for each control point coordinate [DF90]. Thus our combined tracking algorithm evaluates the regularized trajectories of the region border points: this is useful to study the region deformations.

**REGION SEGMENTATION** This region segmentation and tracking is performed by the snake. Such a snake-based segmentation is usually rather accurate, compared to motion-based segmentation for instance. However, the estimation of motion given by the snake is not so satisfactory since it relies only on the information available along the border of the region, and does not consider the interior of the region. Thus, to estimate motion, the algorithm presented in this paper uses a region-based approach, which is likely to be more stable than an contour-based approach.

**REGION-BASED MOTION ESTIMATION** So, given the region segmentation delivered by the snake, the region motion is estimated using a region-based approach inspired by motion-based region segmentation [MB92b]. A 2D affine displacement model is assumed. The parameters of this affine displacement are deduced from the parameters of a 2D affine velocity model. Moreover, this approach is multi-resolution. The interest of this method is that it uses the dense information given by the interior of the region (i.e. the spatio-temporal derivatives of intensity) to estimate motion, instead of the sparse information provided by contours. Thus this approach to motion estimation has rather good precision. It is also quite robust to noise and occlusion, and can handle large displacements, due to a multi-resolution scheme. Besides, here motion estimation is not used to perform region segmentation: it is provided beforehand by the snake. Indeed, motion-based segmentation often provides poor estimations of region boundaries, whereas snakes usually estimate contours with good accuracy. Thus this motion analysis procedure gives a good instantaneous estimation of the

region motion. However these estimations remain relatively noisy. Therefore temporal filtering is performed on them: a Kalman filter with constant speed is applied to each parameter of the region affine motion model. Thus regularized estimates of the region affine motion are given by the algorithm.

**CONTOUR PREDICTION** Moreover, this Kalman filter, which performs temporal smoothing on the estimations of region affine motion, is exploited to provide a prediction of the region motion in the next frame. Thus, by applying the predicted motion to the control points of the current region, the geometry and position of the region in the next frame can be predicted. This prediction is then employed to initialize the snake - which tracks the region contour - in the next frame. Thus the snake search for the region edge is directed towards the image area where the region is to be expected. Such a prediction step is likely to improve the snake tracking performances. Indeed, snakes, which search for the closest edge from their starting point, are sensitive to initialization and can be trapped in local minima. Thus, the use of a prediction clearly reduces the possibility of tracking failure or mismatching by the snake, especially if the contour displacement is large. This way, motion analysis, which relies on the segmentation provided by snakes, operates back on snake tracking and supplies a prediction that should improve the snake tracking capabilities. Thus the tracking approach presented in this paper truly and closely combines snake-based tracking and region-based motion analysis.

**REMARK ABOUT ROBUSTNESS TO OCCLUSION** It is to be noted that this algorithm is rather robust to partial occlusions. Indeed region-based motion estimation is not very sensitive to occlusion since it relies on a dense information. Moreover snakes with motion constraints perform global matching between the template shape extracted in the previous frame and the image contour, so that they are also rather robust to partial occlusions. However, motion constraints on the snake should not be relaxed (for shape refinement) in the case of occlusion, since free deformable B-splines are more sensitive to local information.

### 3 Object segmentation and tracking

This section details how the object is tracked and segmented in each image. In fact tracking is performed by a B-spline snake with motion constraints [BD93b]. Indeed a contour is modelled as a B-spline curve, because such a representation can describe most real-world shapes rather realistically. Then tracking is performed by a deformable B-spline with strong motion constraints, that is to say a parametric displacement model (such as rigid, affine or homographic) is imposed on the deformable curve. Indeed, whereas a classical deformable B-spline [MSMM90] [BD92] is parameterized by the coordinates of its control points, the deformable B-spline with motion constraints [BD93b] is parameterized by the displacement model parameters. Thus this tracking approach performs reliable matching between the points of consecutive contours, due to the motions constraints. This tracking method also is rather quick and robust, due to the limited number of degrees of freedom. Tracking is finally completed by a refinement step, during which motion constraints are relaxed. This

final step is necessary to perform accurate tracking if the chosen displacement model does not exactly describe reality, and especially to deal with non-rigid objects. The equations of this deformable model are presented below.

Let  $C_0(u)$  be the B-spline curve corresponding to the contour of the first image. It is obtained from motion-based segmentation [BF93], which generates a mask of the moving object in the first image. The outline of this mask is extracted by an edge detection algorithm. Then it is approximated - in the least-square sense - by a B-spline curve and optimized by a deformable B-spline curve, so as to better fit the edges of the grey-level image [MSMM90] [BD92]. This B-spline curve is the template shape. As a B-spline, it can be parameterized by linear combination of basis functions  $B_i^k(u)|_{i=0..m}$  - where  $k$  is the degree (plus one) of the B-spline - and control points  $(Vx_{i0}, Vy_{i0})|_{i=0..m}$ , as follows:

$$C_0(u) = \begin{bmatrix} x_0(u) \\ y_0(u) \end{bmatrix} = \sum_{i=0}^m \begin{bmatrix} Vx_{i0} \\ Vy_{i0} \end{bmatrix} B_i^k(u) \quad (1)$$

The contour is then tracked by a deformable B-spline from the initial position given by  $C_0$  to its position  $C(u)$  in the next image. To this purpose, as already mentioned, a 2D affine displacement model is imposed on the deformable curve. Therefore, the curve  $C(u)$  to be found in the next image is described as an affine transform of the template shape, so that it can be written as:

$$C(u) = \begin{bmatrix} x(u) \\ y(u) \end{bmatrix} = A \begin{bmatrix} x_0(u) \\ y_0(u) \end{bmatrix} + T = A \left( \sum_{i=0}^m \begin{bmatrix} Vx_{i0} \\ Vy_{i0} \end{bmatrix} B_i^k(u) \right) + T = \sum_{i=0}^m \left( A \begin{bmatrix} Vx_{i0} \\ Vy_{i0} \end{bmatrix} + T \right) B_i^k(u) \quad (2)$$

As B-splines share the property of affine invariance, the affine transform can be applied either to the control points or to the curve points, as shown by the above formula.

Then the parameters of the curve  $C(u)$  are optimized until the curve fits image contours, by means of energy minimization (as classically done for snake techniques). Of course, the curve parameters are here those of the affine transform, the template shape remaining unchanged. As the deformable curve seeks edges, its associated energy is written as the mean of the intensity gradient along the curve (with a minus sign):

$$E = -\frac{1}{|C|} \int_C |\nabla I|(x, y) du \quad (3)$$

The energy minimization is carried out using Euler-Lagrange dynamics; the system is considered to be massless and embedded in a viscous medium. Therefore the dynamic equations of the parameter  $a_j$  of the affine transform  $(A, T)$  (which parametrize the deformable curve) are the following:

$$\gamma_{a_j} \frac{da_j}{dt} = -\frac{\partial E}{\partial a_j} = \frac{1}{|C|} \int_C \left( \frac{\partial |\nabla I|}{\partial x} \frac{\partial x}{\partial a_j} + \frac{\partial |\nabla I|}{\partial y} \frac{\partial y}{\partial a_j} \right) du \quad (4)$$

As a result of the use of a displacement model, the deformable curve is globally matched to the image edges. Thus the algorithm is rather robust and the point-to-point matching between the curves is fairly reliable.



Finally the motion constraints imposed by the displacement model may be relaxed, in order to refine the estimated contour. In this final stage, the B-spline curve becomes fully deformable, like in [MSMM90] [BD92]. This refinement step improves the precision of the contour extraction in the case where part of the object deformations (like non-rigid deformations) cannot be described by the selected displacement model, if no occlusion has occurred.

The final curve extracted in a frame becomes the new template shape which is in its turn tracked to the next frame. Thus the template shape is continuously updated during tracking.

It can be noted that this approach makes use of two different scales to better extract the contours. Indeed, the calculation of the image forces (related to the intensity gradient) governing the snake motion is preceded by a smoothing of the image. First, when global tracking of the object is performed using a deformable B-spline with affine displacement model, the smoothing factor is large. And finally, when the extracted edge is locally refined by a free deformable B-spline, a small smoothing factor is applied to the image.

Furthermore, the trajectories of the control points parameterizing the region contour, and measured by the snake, are smoothed using Kalman filtering. Indeed these trajectories are relatively noisy and their regularization provides a better description of the region deformations. Thus each coordinate of the control points is filtered independently using a Kalman filter called the  $\alpha, \beta$  tracker [DF90]. Indeed Kalman filtering is a statistical and recursive approach to estimate a time-varying vector  $X_t$  from noisy measurements  $V_t$ , given a dynamic system model, the uncertainty associated with the model and the uncertainty associated with the measures. Here each vector to estimate, called the state vector  $X_t$ , is the position  $x$  and velocity  $\dot{x}$  of the given control point coordinate. A constant velocity model is assumed. The position  $x$  of the control point coordinate can be measured directly. The steady state Kalman filter that can be written for this system is called the  $\alpha, \beta$  tracker. It generates the estimations associated with each state vector (i.e. each control point coordinate) using the following equations:

$$\begin{aligned} x_{t/t} &= -(\alpha + \beta - 2)x_{t-1/t-1} - (1 - \alpha)x_{t-2/t-2} + \alpha v_t + (-\alpha + \beta)v_{t-1} \\ \dot{x}_{t/t} &= -(\alpha + \beta - 2)\dot{x}_{t-1/t-1} - (1 - \alpha)\dot{x}_{t-2/t-2} + \beta(v_t - v_{t-1}) \end{aligned} \quad (5)$$

The reader interested by more details on the performances of such a filter is kindly referenced to [DF90].

## 4 Motion estimation and prediction

Given the segmentation performed by the snake, as described in the previous section, the region motion is then estimated using a region-based approach inspired from motion-based region segmentation [MB92b] [BF93]. Indeed such a region-based approach to motion estimation is likely to be more stable than a contour-based approach. A 2D affine displacement model is used to capture the motion of the tracked region between two successive frames. The computation of this affine displacement relies on the spatio-temporal derivatives of the

intensity function and is embedded in a multi-resolution scheme. This module is described below:

**The region evolution model** As mentioned before, the region  $R$  in the image at time  $t + 1$  is assumed to be the result of a 2D affine transform,  $(\mathbf{A}, \mathbf{T})$ , of the region  $R$  in the image at time  $t$ . Hence every point  $(x(t), y(t)) \in R$  at time  $t$  is located at  $(x(t + 1), y(t + 1))$  at time  $t + 1$ , with :

$$\begin{pmatrix} x \\ y \end{pmatrix}(t + 1) = \mathbf{A}(t) \begin{pmatrix} x \\ y \end{pmatrix}(t) + \mathbf{T}(t) \quad (6)$$

These regions have already been delineated by a snake and we need to determine the parameters of this affine displacement model  $(\mathbf{A}, \mathbf{T})(t)$ . In fact, these parameters can straightforwardly be derived from the parameters of a 2D affine velocity model  $(\mathbf{M}, \mathbf{b})$ . Indeed, the 2-D velocity field within the region  $R$  can be approximated with an affine model :

$$\forall (x, y) \in R, \quad \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix}(t) = \mathbf{M}(t) \begin{pmatrix} x \\ y \end{pmatrix}(t) + \mathbf{b}(t) \quad (7)$$

Such affine models of the 2D motion field have also been used by [Ber92b, Bur89, Neg92]. Indeed, such a model can describe a large class of motions: rotation, scaling, shear, ... It can also handle deformable motions to a certain extent. Moreover, an affine model of the 2-D velocity field conveys valuable and robust information about the 3-D motion and structure [MB92a].

If  $(x, y)^T(t + 1)$  is expanded in Taylor series to the first order, it can be written as follows (where  $\delta t$  is the time step between two successive frames):

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

From (6), (7) and above, it comes (with  $\mathbf{I}_2$  being the  $2 \times 2$  identity matrix):

$$\mathbf{A}(t) = \mathbf{I}_2 + \delta t \mathbf{M}(t) \quad \text{and} \quad \mathbf{T}(t) = \delta t \mathbf{b}(t) \quad (8)$$

Thus the affine displacement  $(\mathbf{A}(t), \mathbf{T}(t))$  can be deduced from the affine model  $(\mathbf{M}, \mathbf{b})(t)$  of the 2-D motion field. To estimate  $(\mathbf{M}, \mathbf{b})(t)$ , a multi-resolution scheme has been developed. Therefore the method provides accurate and robust estimates of the motion parameters, even in the case of large displacements. It is presented below.

**Multi-resolution estimation of the motion parameters** The idea is to estimate the motion parameters with a ‘‘coarse-to-fine’’ strategy. First a rough estimate of the parameters is obtained at the lowest resolution. Then it is refined using the higher resolution images. Similar hierarchical methods have already been employed for the estimation of optical flow [Enk88] [Ana89] and parametric motion models [Ber92a] [MB92a].

Two low pass Gaussian pyramids are built, for the images at time  $t$  and  $t + \delta t$ . For each segmented region, the pyramid provides the delineations of the regions at each level. First,

the six affine velocity parameters of the region are estimated at the lowest resolution level  $L$ , with a least-squares fit to normal flows. To do this, we use the well-known image flow constraint equation [Hor81] that relates the motion field  $\mathbf{v}(x, y)$  to the spatial gradient of the image intensity  $\nabla \mathbf{I}(x, y)$  and its temporal partial derivative  $I_t(x, y)$ :

$$\nabla \mathbf{I}(x, y) \cdot \mathbf{v}(x, y) + I_t(x, y) = 0 \quad (9)$$

Thus the affine velocity model  $\mathbf{v}_{(\mathbf{M}, \mathbf{b})}(x, y)$  given by (7) should verify equation (9).

Each image in the Gaussian pyramid is a blurred and sub-sampled (of a factor 2 in both directions) version of its predecessor. The displacements measured in pixel at level  $l + 1$  are twice as less as the displacements at level  $l$ . As a result, the optical flow constraint equation (9), which assumes that the image motion is small, can be more easily applied at the lowest resolution level  $L$ , and the parameters  $(\mathbf{M}^L, \mathbf{b}^L)$  are estimated over the region  $R^L$  at level  $L$  using a least-square procedure based on relation (9).

Then, we refine the estimate using the higher resolution levels, with an equation similar to (9). For the clarity of the presentation, 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  $\mathbf{v}_{(\mathbf{M}, \mathbf{b})^l}(\mathbf{p}^l)$  be the affine model of the velocity at location  $\mathbf{p}^l$ :

$$\mathbf{v}_{(\mathbf{M}, \mathbf{b})^l}(\mathbf{p}^l) \triangleq \mathbf{M}^l \cdot \mathbf{p}^l + \mathbf{b}^l \quad (10)$$

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:

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

In fact (9) is the differential version of the above equation. Since  $\delta \mathbf{p}^l = \mathbf{v}_{(\mathbf{M}, \mathbf{b})^l}(\mathbf{p}^l) \cdot \delta t$ , we wish to 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  $\mathbf{p}^{l+1}$ . 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  $l$ :

$$\delta^2 \mathbf{p}^l \triangleq \delta \mathbf{p}^l - 2\delta \hat{\mathbf{p}}^{l+1} \quad (11)$$

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

Let us define  $\Delta \mathbf{M}^l \triangleq \mathbf{M}^l - \widehat{\mathbf{M}}^{l+1}$ , and  $\Delta \mathbf{b}^l \triangleq \mathbf{b}^l - 2\widehat{\mathbf{b}}^{l+1}$ , as the refinement of the motion parameters to be estimated at the current level  $l$ .

It can be shown, using a first order expansion of  $I$  about  $\mathbf{p}^l + 2\delta \hat{\mathbf{p}}^{l+1}$ , that:

$$\nabla \mathbf{I}(\mathbf{p}^l + 2\delta \hat{\mathbf{p}}^{l+1}, t + \delta t) \cdot (\Delta \mathbf{M}^l \mathbf{p}^l + \Delta \mathbf{b}^l) \delta t + I(\mathbf{p}^l + 2\delta \hat{\mathbf{p}}^{l+1}, t + \delta t) - I(\mathbf{p}^l, t) = 0 \quad (12)$$

Equation (12) is linear with respect to  $\Delta \mathbf{M}^l$  and  $\Delta \mathbf{b}^l$ . Least-squares estimates  $(\Delta \widehat{\mathbf{A}}^l, \Delta \widehat{\mathbf{b}}^l)$ , of  $(\Delta \mathbf{A}^l, \Delta \mathbf{b}^l)$  can thus be obtained.

At each level we have :

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

Rather than incrementally warping the image at time  $t$  toward the image at time  $t + 1$ , as proposed in [Ber92b] and in [Ber92a], we use an “incremental” version (12) of the image flow constraint equation at each level. If the initial estimate is close enough to the real displacement  $\delta p^l$ , then the small incremental displacement will be accurate.

Finally, bilinear interpolations of  $I$  and  $\nabla I$  have to be computed in (12) for points which are not on the image grid.

**Recursive estimation of the motion parameters** The multi-resolution method gives instantaneous measurements of the motion parameters. 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 (e.g. when a tracked region is being totally occluded). Indeed, in this case we can generate the optimal predicted estimate, given the previous measurements, of each of the motion parameters, as explained in [Med69].

In the absence, in the general case, of any explicit simple analytical function describing the evolution of the variables, a Taylor-series expansion of each component of  $(\widehat{\mathbf{M}}, \widehat{\mathbf{b}})(t)$  yields a local approximation of the temporal evolution of the parameter. Even though using the first three terms of the expansion (i.e. constant acceleration model) results in a more precise model, we have noticed that a filter based on a first order expansion (i.e. constant velocity model) 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, even in situations of long occlusions. Moreover, we have observed in many sequences that the correlation coefficients between the six components of  $(\widehat{\mathbf{M}}, \widehat{\mathbf{b}})(t)$  are negligible. Assuming that the six variables are indeed independent, we have decided to decouple the six filters. Let  $a_j(t), j = 1, \dots, 6$ , be one of the component of  $(\widehat{\mathbf{M}}, \widehat{\mathbf{b}})(t)$ , and let  $\tilde{a}_j(t)$  be the instantaneous measurement of  $a_j(t)$ , given by the multi-resolution algorithm. The following dynamic system provides a local approximation to the temporal evolution of each motion parameter  $a_j$  :

$$\begin{cases} \begin{bmatrix} a_j \\ \dot{a}_j \end{bmatrix} (t+1) = \begin{bmatrix} 1 & \delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a_j \\ \dot{a}_j \end{bmatrix} (t) + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \end{bmatrix} (t) \\ \tilde{a}_j(t) = a_j(t) + \beta(t) \end{cases} \quad (13)$$

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

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

A standard Kalman filter generates recursive estimates of each motion parameter. The initialization of the Kalman filter is thoroughly discussed in [MB92b].

The estimates of the 2D affine velocity model delivered by the filter are then used to compute the 2D affine displacements ( $\mathbf{A}$ ,  $\mathbf{T}$ ) according to relation (8). Furthermore, a prediction of the region affine displacement in the next frame is calculated. It is then used to initialize in the next frame the position of the snake tracking the region contour. This helps solving tracking ambiguities that the snake may encounter. In conclusion, the two main modules of our approach, contour tracking and motion estimation, truly cooperate, since motion estimation relies on the segmentation provided by tracking, whereas tracking is guided in each new image by the motion prediction.

## 5 Experimental results

This section presents experimental results illustrating the tracking algorithm presented in this paper - combining snake-based contour tracking and region-based motion analysis. These tests were carried out on real images of different types: a moving arm, an highway scene and a moving person.

The first image sequence shows moving cars on a highway (fig. 2 and 3). Tracking was initialized in the first image of the sequence using motion-based region segmentation, as described in section 2. Indeed, the segmentation algorithm provides a map of the moving objects (fig. 2a), from which pixel chains can be extracted using an edge point extraction and linking technique. One of these chains describes the car, yet the silhouette is very unprecise (fig. 2b). It was approximated by a B-spline and optimized using a fully deformable B-spline snake, so as to accurately fit image edges (fig. 2c). This curve is then used to initialize tracking. Besides, in this image sequence, the displacement of the tracked car between two frames is quite large (see fig. 3a showing all the contours extracted along the image sequence), especially at the beginning, so that classical snake-based approaches are not well adapted to track it. On the contrary our approach performs correct and accurate tracking, due to the enforcement of a motion model on the snake (which increases its robustness) and to the use of a prediction to guide the tracking snake in the region of interest (this prediction is based on our region-based motion analysis scheme). The last few images of the sequence (fig. 2f) also illustrate the robustness of the approach with respect to partial occlusion: tracking continues correctly despite a beginning occlusion of the car. Figure 3b displays the filtered trajectories of the car contour points. These trajectories are quite coherent and close to what can be expected, though still a bit noisy.

The second image sequence shows the head of a man moving in front of the camera (see fig. 3 and 4). First, the person moves leftward, then rightward. It is tracked using snakes with affine-plus-deformable motion, as described in section 2. This approach handles this example quite correctly, despite deformations due to perspective (such as scaling effects) and temporary occlusion of the ears for instance. Moreover, the head motion is estimated using the dense and multi-resolution method presented in the previous section; it is then

possible, through temporal filtering of the motion estimation, to predict the head position in the next frame and initialize the snake. This prediction (see fig. 4d) is rather good and it improves the quality of snake tracking since the head motion is sometimes large and the background is cluttered. The overall approach is quite robust since tracking continues correctly despite the motion change of direction. It can be noted that the edge detection is a bit disturbed in the 14th image (fig. 4c) by the cluttered background, but this error is corrected in the following images. Fig.3c and d show the contours tracked during motion. Fig. 4f also displays the trajectories of some of the outline points, estimated by the tracking snake and filtered over time. It can be seen that these trajectories are quite regular, due to the global matching performed by snakes with motion constraints.

The third image sequence shows a moving arm (fig. 5 and 6). This example is interesting because the motion is articulated (flexion then stretching of the arm). Thus tracking is more difficult. Moreover, the affine motion model assumed by the approach is not a very good approximation of the object 2D motion. However the arm is tracked successfully. This example illustrates the usefulness of relaxing the snake motion constraints on each image in a second phase. First, global tracking of the arm is performed between two successive images thanks to the affine motion constraints, despite a large displacement. But this affine model is not sufficient to describe the arm motion. Therefore, in a second step, motion constraints are relaxed and the snake freely fits the arm outline. Thus the arm contour is tracked and extracted with a relatively good precision, despite the blur present along the arm in the image. The arm outlines extracted along the image sequence are shown in fig. 6b. The algorithm also evaluates the trajectories of some of the arm contour points (fig. 6a): these look like what a human viewer would expect. In this example, the prediction (see fig. 5c) is not very precise due to the bad adequacy of the affine model to the real motion. But it is still useful since it helps to solve ambiguities between the arm contour and the face shadow: without the prediction, the snake is confused by the shadow contour which is close and fails to track the arm correctly.

## 6 Conclusion

The experimental results illustrate the good tracking performances of the approach for various types of motions and objects. This approach also proved rather robust to partial occlusion. The distinctive feature of this method (that gives it its robustness and efficiency) is the combination of snake-based contour tracking and region-based motion analysis. A snake tracks the region outline and thus performs segmentation. This segmentation is exploited by a region-based motion estimation algorithm inspired from motion-based region segmentation. Then the motion estimation module generates a prediction of the shape and location of the region in the next frame, through filtering. This prediction is used to guide the tracking snake: it helps the snake to solve ambiguities during tracking. Besides, tracking can be automatically initialized in the first image of the sequence using motion-based region segmentation. The algorithm is completed by a trajectography module which estimates the filtered trajectories of the object outline points.

In the future, it is planned to treat total occlusions. The approach will also be finalized by several applications, such as the estimation of time-to-collision, the spatio-temporal surface generated by the tracked object contour or 3D structure and/or motion.

## Références

- [Ana89] Anandan, P. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision, Vol 2:283–310*, 1989.
- [BBB87] R. Bartels, J. Beatty, and B. Barsky. *An Introduction to Splines for use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann, 1987.
- [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.
- [BD93a] B. Bascle and R. Deriche. Energy-based methods for 2d curve tracking, reconstruction and refinement of curves of 3d curves and applications. *Proceedings of Geometric Methods in Computer Vision, San Diego, July, 1993*.
- [BD93b] B. Bascle and R. Deriche. Stereo matching, reconstruction and refinement of 3 d curves using deformable contours. *Proceedings of 4th ICCV, Berlin, 11-14 May, 1993*.
- [BD92] B. Bascle and R. Deriche. Features extraction using parametric snakes. In *Proceedings 11th IAPR*, volume 3, pages 659–662, August 30, September 3 92.
- [Ber92a] Bergen, J.R. and Anandan, A. and Hanna, K. and Hingorani, R. Hierarchical model-based motion estimation. *Proc. of ECCV-92, S.Margherita Ligure, Italy, pages 237–252, Springer-Verlag, 1992*.
- [Ber92b] Bergen, J.R. and Burt, P.J. and Hingorani, R. and Peleg, S. A three-frame algorithm for estimating two-component image motion. *IEEE Trans. PAMI, Vol.14, No.9:pp 886–896*, Sept 1992.
- [BF93] P. Bouthemy and E. François. Motion segmentation and qualitative dynamic scene analysis from an image sequence. *Intern. J. Comput. Vis.*, 10(2):157–182, 1993.
- [BL93] P. Bouthemy and P. Lalande. Recovery of moving object masks in an image sequence using local spatio-temporal contextual information. *Optical Engineering*, Vol.32, No.6:pp 1205–1212, June 1993.
- [Bur89] Burt, P.J. and Bergen, J.R. and Hingorani, R. and Kolczynski, R. and Lee, W.A. and Leung, A. and Lubin, J. and Shvaytser, H. Object tracking with a moving camera. *Workshop on Visual Motion*, pages pp 2–12, March 1989.

- [CAS92] I. Cohen, N. Ayache, and P. Sulger. Tracking points on deformable objects using curvature information. In *Proc. 2nd ECCV, Santa Margherita Ligure, Italy*, May 1992.
- [CB92] R. Cipolla and A. Blake. Surface orientation and time to contact from image divergence and deformation. In *Proc. 2nd ECCV, Santa Margherita Ligure, Italy*, pages 187–202, May 1992.
- [CB92] R. Curwen and A. Blake. *Active Vision*, chapter Dynamic Contours: Real-time Active Splines, pages 39–57. MIT Press, 92.
- [CSD88] J.L. Crowley, P. Stelmazyk, and C. Discours. Measuring image flow by tracking edge-lines. In *Proc. 2nd Int. Conf. Computer Vision, pp 658-664, Florida*, Dec 1988.
- [DF90] R. Deriche and O.D. Faugeras. Tracking line segments. In *First European Conference on Computer Vision, p. 259-268, Antibes, France*, April 1990.
- [Enk88] Enkelmann, W. Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences. *Comput. Vision, Graphics Image Process.*, 43:150–177, 1988.
- [GCDVF93] B. Giau-Checa, R. Deriche, T. Viéville, and O.D. Faugeras. Tracking segments in a monocular sequence of images. *INRIA Research Report 2113, Sophia Antipolis*, 1993.
- [Hor81] Horn, B.K.P. and Schunck, B.G. Determining optical flow. *Artificial Intelligence, Vol.17:pp 185-203*, 1981.
- [KDTN92] D. Koller, K. Daniilidis, T. Thórhallson, and H.H. Nagel. Model-based object tracking in traffic scenes. In G. Sandini, editor, *Proc. of Second European Conference on Computer Vision, ECCV-92, Santa Margherita, Italy*, pages 437–452. Springer-Verlag, 1992.
- [KWT87] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *First International Conference on Computer Vision*, pages 259–268, June 1987.
- [Low92] D.G. Lowe. Robust model-based motion tracking through the integration of search and estimation. *Intern. J. Comput. Vis.*, 8(2):113–122, 1992.
- [MB92a] F. Meyer and P. Bouthemy. Estimation of time-to-collision maps from first order motion models and normal flows. In *Proc. 11th Intern. Conf. on Pattern Recognition (ICPR'92), The Hague, Netherlands*, volume 1, pages 78–82, Sept 1992.
- [MB92b] F. Meyer and P. Bouthemy. Region-based tracking in an image sequence. In G. Sandini, editor, *Proc. 2nd European Conf. on Computer Vision (ECCV'92), Santa Margherita Ligure, Italy*, pages 476–484. Springer-Verlag, May 1992.



- 
- [Med69] Meditch, J.S. Stochastic optimal linear estimation and control. *McGraw-Hill*, 1969.
- [MSMM90] A. Menet, P. Saint-Marc, and G. Medioni. Active contour models: Overview, implementation, and applications. In *IEEE Conf. Syst. Man. Cyb. L.A.*, Nov 90.
- [Neg92] Negahdaripour, S. and Lee, S. Motion recovery from image sequences using only first order optical flow information. *Intern. J. Comput. Vis.*, 9(3):163–184, 1992.
- [Sch82] Schalkoff, R.J. and McVey, E.S. A model and tracking algorithm for a class of video targets. *IEEE Trans. PAMI*, Vol.PAMI-4, No.1:2–10, Jan 1982.
- [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, NJ*, pages 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. In *IEEE Pami*, Vol 9, No 1, pp 56-73, Jan 1987.
- [TS92] D. Terzopoulos and R. Szeliski. *Active Vision*, chapter Tracking with Kalman Snakes, pages 3–20. MIT Press, 92.