A PDE-based Level-Set Approach for Detection and Tracking of Moving Objects

Nikos K. Paragios Rachid Deriche
INRIA, 2004 Route des Lucioles, BP 93
06902 Sophia Antipolis Cedex, FRANCE
e-mail:{nparagio, der}@sophia.inria.fr

Abstract

This papers presents a framework for detecting and tracking moving objects in a sequence of images. Using a statistical approach, where the inter-frame difference is modeled by a mixture of two Laplacian or Gaussian distributions, and an energy minimization based approach, we reformulate the motion detection and tracking problem as a front propagation problem. The Euler-Lagrange equation of the designed energy functional is first derived and the flow minimizing the energy is then obtained. Following the work by Caselles et al [CKS95] and Malladi et al [MSV95, MSV93], the contours to be detected and tracked are modeled as geodesic active contours evolving toward the minimum of the designed energy, under the influence of internal and external image dependent forces. Using the level set formulation scheme of Osher and Sethian [OS88], complex curves can be detected and tracked and topological changes for the evolving curves are naturally managed. To reduce the computational cost required by a direct implementation of the formulation scheme of Osher and Sethian [OS88], a new approach exploiting aspects from the classical Narrow Band [AS95] and Fast Marching [Set96] methods is proposed and favorably compared to them. In order to further reduce the CPU time, a multi-scale approach has also been considered. Very promising experimental results are provided using real video sequences.

1 Introduction

Detection and *tracking* of moving objects in a sequence of images are problems arising in numerous applications of computer vision and image coding. This paper deals with these two problems of motion analysis supposing a static scene with moving objects.

In this paper, we propose a novel approach for the *detection* and *tracking* of moving objects. Both problems are stated under a unified model which follows a level-set methodology. The inter-frame difference is modeled as a mixture of two Laplacians or Gaussians distributions. Using this analysis, we generate a new image, based on the input images which exhibit large gradient values only around the moving area. The use of geodesic active contours is then introduced. For the front propagation problem, three well-known schemes are used, Classic, Narrow Band [AS95] and Fast Marching [Set96] approaches. A new scheme is proposed, called Hermes, which is also evaluated and compared to the existing schemes. In order to achieve a faster algorithm, the front propagation methods are combined with a classic multi-scale approach.

Simultaneously to this work, the idea of applying the curve evolution theory to the tracking problem has been recently presented by Caselles and Coll in [CC96]. However this sequentially three step approach is very different from the unified approach we present in this article. Following the work on geodesic active contours by [CKS95], they first start by detecting the contours of the objects to be tracked. An estimation of the velocity vector field along the detected contours is then performed using a completely separate approach, and finally another PDE is designed to move the contours to the boundary of the moving objects. These contours are then used as initial estimate of the contours in the next image, and the process is repeated.

In the next section, we introduce the proposed *detection and tracking* model. The existing front-propagation methods are briefly described and evaluated in comparison with our approach in Section 3. Finally, Section 4 presents experimental results, and concluding remarks.

2 Detection and Tracking

2.1 Defining the model

Let $D = \{d(X), X \in R^2\}$ denote the *inter-frame* gray level difference image with

$$d(X) = I(X; t+1) - I(X; t)$$

The detection problem consists of a "binary" label (static or mobile) for each pixel on the image grid. Let $p_{D|st}(d|st)$ (resp. $p_{D|mob}(d|mob)$) be the probability density function of the observed inter-frame difference under the static (resp. mobile) hypothesis. These probability density functions are supposed to be homogeneous, *i.e.* independent of the pixel location and usually they follow the Laplacian or Gaussian law, that is:

$$p(d) = \frac{\lambda}{2} e^{-\lambda|d|} \;,\; p(d) = \frac{1}{2\sigma\sqrt{\pi}} e^{-\frac{d^2}{2\sigma^2}}$$

We suppose that the probability density function of the observed inter-frame difference is a mixture distribution of two different populations (static and mobile), given by

$$p_D(d) = P_{st}\; p_{D|st}\,(d|st) + P_{mob}\; p_{D|mob}(d|mob)$$

where P_{st} (P_{mob}) is the *a priori* probability of static (mobile) case. In this mixture distribution problem, the principle of Maximum Likelihood is used to obtain iteratively an estimation of the four unknown parameters $(P_{st}, P_{mob}, \{\lambda_{st}, \lambda_{mob}\}||\{\sigma_{st}, \sigma_{mob}\})$ [DH73], from the observed distribution of grey level inter-frame differences [PPT$^+$96] (Fig. 1).

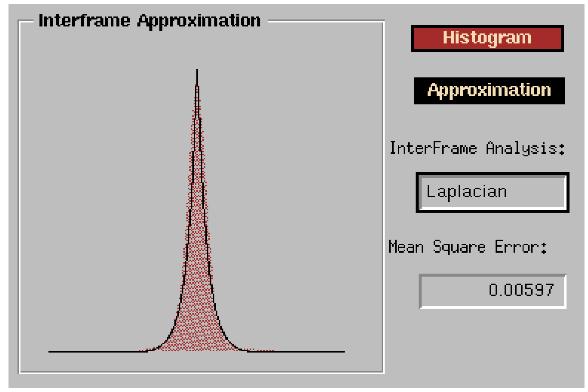

Figure 1: Mixture decomposition for inter-frame difference for *Football match sequence*

2.2 Setting the Measurement Image

The contour of the *moving area* corresponds to the pixels where there is a **transition** between their labels and the labels of their neighbors. A pixel X (a site at the image grid) defines the contour of the moving area if its label is **static** (**mobile**) and there is at least a neighborhood pixel Y with the label **mobile** (**static**). For each pixel location X and a neighborhood pixel Y, we define the following energy terms:

$$\begin{aligned} E_{trans}(X,Y) = \;& p_{D|st}(d(X)|st) \cdot p_{D|mob}(d(Y)|mob) \\ & + p_{D|mob}(d(X)|mob) \cdot p_{D|st}(d(Y)|st) \\ E_{smooth}(X,Y) = \;& p_{D|st}(d(X)|st) \cdot p_{D|st}(d(Y)|st) \\ & + p_{D|mob}(d(X)|mob) \cdot p_{D|mob}(d(Y)|mob) \end{aligned}$$

Taking into account these energy terms, we propose a unified model concerning the detection as well as the tracking part. For the detection part, we create a new image ($F(X)$) based on the inter-frame difference analysis and we combine this image with the input image $I(X;t)$ (resp. $I(X;t+1)$) in order to extract the precise location of the given object at time instant t (for the tracking part). This new image should have large gradient values in the boundaries of the moving area. In order to achieve this, we define for each pixel the following measurement:

$$F(X) = \max_{Y \in n_{g(X)}} \left\{ \frac{E_{trans}(X,Y)}{E_{smooth}(X,Y)} \right\}$$

where $n_{g(X)}$ denotes the neighborhood of pixel (X) (second-order with 8 pixels).

It is clear from the above definition that the generated image would present large magnitudes only in the pixels around the boundary of the moving area (Fig. 2). To avoid the noise influence, we can perform a smoothing operation before the estimation of this image. For instance a median filtering technique could be applied in the image of the *inter-frame difference*.

2.3 Geodesic Active Detection-Tracking

Additionally the complete motion detection is not equivalent to temporal change detection. In other words, the moving estimated area between two successive images corresponds to the union of the moving object locations in these images. Taking into account this remark, we express the problem of detection and tracking using the framework of energy minimization. We associate an energy function to the given curve, and we try to find (for a given parameter λ) the curve $C(p,t)$ that minimizes the following energy:

$$E(C(p)) = (1-\lambda) \underbrace{\int_0^1 |C'(p)|^2 dp}_{\mathbf{E}_{internal}(C)} +$$

$$\underbrace{\lambda \int_0^1 (\gamma \overbrace{g(|\nabla F(C(p))|)}^{\text{detection term}} + (1-\gamma) \overbrace{g(|\nabla I(C(p);t)|)}^{\text{tracking term}})^2 dp}_{\mathbf{E}_{image}(C)}$$

where $\lambda \in [0,1]$ is a positive constant which balances the contribution of the two energy terms. The energy term $E_{internal}(C)$ accounts for the expected spatial properties (*i.e. smoothness*) of the contour while the energy term $E_{image}(C)$ stands for the *attraction* energy term of the curve towards the objects contour.

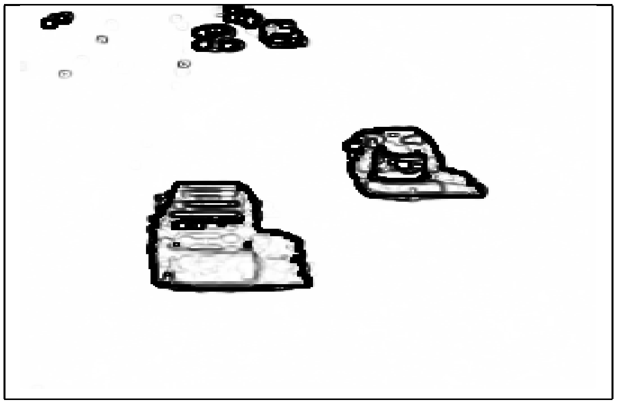

Figure 2: Generated image for: *Autoroute Sequence*

The detection term forces the curve to fit the moving area, avoiding edges or static objects. Once the curve is around the moving area, this term is close to zero. The tracking term, is used for evolving the curve until it reaches the exact location of the moving object. Concerning g, it is a monotonically decreasing function such that $\lim_{r\to\infty} g(r) = 0$ and $g(0) = 1$. Finally $\gamma \in [0, 1]$ is a parameter which balances the contribution of the detection and the tracking terms. Selecting a γ value close to 1, we push the model to detect the moving objects, while with a value close to 0 we have a classic geodesic active contour model. In the general case, γ must have a value close to 0.75. Since we use a multi-scale approach (as we are going to explain later), the selection of the γ value is not so difficult. Thus at the low resolution levels, we use a value close to 1, while at the high resolution levels this value is close to 0 (the moving area has already been reached by the contour).

For simplicity reasons concerning the mathematical formulas, we define the following function

$$h(C(p)) = |\gamma \cdot g(|\nabla F(C(p))|) + (1-\gamma) \cdot g(|\nabla I(C(p); t))|$$

Following the approach developed in [CKS95], it can be proved that minimizing $E(C(p))$ leads to a geodesic curve in a Riemmanian space with a new metric, and the problem can be shown to be equivalent to the minimization of:

$$E_0(C(p)) = \int_0^L h(C(p))ds \quad (1)$$

where ds is the Euclidean arc-length element, and L the Euclidean length of $C(p)$. In other words, when we try to detect an object, we try to find the best minimal length that takes into account the image characteristics.

We minimize (1) by solving the associated Euler-Lagrange equation. According to this equation, the flow that deforms the initial curve towards the local minima of (1) is given by the steady state solution of:

$$C_t = (h(C)\mathcal{K} - \nabla h(C) \cdot \mathcal{N})\,\mathcal{N} \quad (2)$$

where $\mathcal{N}$ denotes the inward Euclidean normal vector to the curve C, and $\mathcal{K}$ the Euclidean curvature. Equation (2) can be implemented by updating the position vector p using a difference approximation scheme. The main drawback of this approach is that the evolving model is not capable to deal with topological changes of the moving front. This could be avoided by introducing the work of Osher and Sethian [OS88]. In such a case the curve $C(p, t)$ is represented by the zero level set of a smooth and continuous function $\Phi : \mathbf{R^2} \times [0, t) \to \mathbf{R}$ given by $\{\mathbf{X} \in \mathbf{R^2} : \Phi(X, t) = 0\}$. Since $C(p, t)$ is on the zero level set, it satisfies

$$\Phi(C, t) = 0 \quad (3)$$

By differentiating equation (3) with respect to time, and then with respect to the curve parameter, the following associated equation of motion for the surface Φ can then be derived:

$$\Phi_t = \Big\{\gamma \cdot (g(|\nabla F(C)|) \cdot \mathcal{K} + \nabla g(|\nabla F(C)|) \cdot \tfrac{\nabla\Phi}{||\nabla\Phi||}) +$$
$$(1-\gamma) \cdot (g(|\nabla I(C;t)|) \cdot \mathcal{K} + \nabla g(|\nabla I(C;t)|) \cdot \tfrac{\nabla\Phi}{||\nabla\Phi||})\Big\}\,|\nabla\Phi|$$

where the value of $\mathcal{K}$ comes from the level-set of Φ. In our case g is considered to be: $g(r) = 1/(1 + |r|^p)$, $p = 1, 2$. See our previous work in [DF96], for different functions.

This model is parameter free, as well as topology-free since different topologies of zero level-set correspond to the same topology of Φ. The main difference between the classical snake models and the curve evolution model is the independence of the topology due to the level-set representation. This allows detection of all the objects which appear in the image plane, without knowing their exact number. Additionally, we don't have any evolution problems concerning the contour initialization, due to the fact that the two terms of the velocity magnitude are both equal to zero only at the contours points. This pushes the contour to evolve even if it has been initialized far away from the solution.

The resulting PDE (acting on Φ evolution) is then solved as described in [OS88], using the approaches proposed in the next section.

3 Front Propagation Algorithms

A direct implementation approach of the resulting PDE involves the re-estimation of the characteristic image of all the level set pixels (not simply the zero level set corresponding to the front itself). This front evolution method is computationally very expensive, due to many useless operations that are performed during the front propagation (especially in pixels which are out of interest). In order to overcome this drawback two different methods have been proposed: the "Narrow Band" method [AS95] and the "Fast Marching" method [Set96].

3.1 Narrow Band Approach

The key idea is to deal only with pixels which are close to the latest estimation of the zero level-set contour in both directions (inwards and outwards). This is known as Narrow Band Approach [AS95]. Since the curve evolution is smoothly performed according to the Euler-Lagrange equations, the use of pixels which are far away from the current contour does not effect the evolving process. Thus we can work only with pixels around to the current contour estimation. In that case a set of narrow band points is defined around the latest contour estimation and the evolving Euler-Lagrange equation is performed only for these points. A significant cost reduction is achieved through this approach (compared to the classic method), but certainly the cost still remains considerable.

3.2 Fast Marching Approach

Consider a special case of a front moving with a speed $\mathcal{F} = \mathcal{F}(x, y)$, $\mathcal{F} > 0$. Let now consider a monotonically advancing front whose level-set equation is of the form: $\Phi_t = \mathcal{F}(x, y)||d\Phi||$. Let $T(x, y)$ be the time at which the curve crosses the site (x, y). In this time the surface $T(x, y)$ satisfies the equation: $||\nabla T|| \cdot \mathcal{F} = 1$. This equation simply says that the gradient of arrival time surface is inversely proportional to the speed of the front. Using the above equation and an approximation for the gradient norm $|\nabla T|$, (proposed in [AS95]) we are looking for the solution of:

$$[max(max(D^x_- T, 0), -min(D^x_+ T, 0))^2 + \\ max(max(D^y_- T, 0), -min(D^y_+ T, 0))^2] = \frac{1}{F^2_{xy}}$$

where

$$D^x_- T(x, y) = \frac{T(x,y) - T(x,y-1)}{2}, \quad D^x_+ T(x, y) = \frac{T(x,y+1) - T(x,y)}{2}$$
$$D^y_- T(x, y) = \frac{T(x,y) - T(x-1,y)}{2}, \quad D^y_+ T(x, y) = \frac{T(x+1,y) - T(x,y)}{2}$$

Concerning the gradient approximation, different forms could also be used ([RT92]). Since according to gradient approximation equation, information are propagated in "one way" (that is from smaller to larger values of T), it is possible to build the solution outwards of the smallest time value T. The idea is to sweep the front ahead in an upwind fashion (by considering a set of pixels in narrow band around the existing front), and to march this narrow band forwards (freezing the values of existing pixels and bringing new ones into narrow band structure) [AS95].

The algorithm is composed of two basic steps: the *initialization* and the *marching* one. Initially, a user-defined contour initialization takes place. All the pixels belonging to the contour are set to *Alive* and their T value to zero. For all *Alive* pixels, we select their neighbors, set them as *Narrow Band* pixels and initialize their T value equal to $\frac{1}{F}$. The rest of the pixels are assumed to be *FarAway* pixels with a T value close to infinite. During the *marching* step, the *Narrow Band* pixel with the smallest T value is selected and set *Alive* in each iteration. The idea is to propagate the information from the lowest values to the biggest ones. All the neighborhood sites of the selected pixel which are *FarAway* are moved to *Narrow Band*. A re-estimation of T values (according to the quadratic approximation equation) for the *Narrow Band* neighborhood pixels takes place. The algorithm stops when all pixels have been labeled *Alive*.

Despite the fact that *Fast Marching* requires low computational cost, it has a *limited set of applications*. The *main drawback is the necessity to have only positive or negative speed function during the front evolution*. Additionally, this algorithm can be used only for cases with location-dependent speed function. As a consequence, we *can't use this method in a case where there is a curvature-dependent speed function, a very popular case which appears in a large variety of image vision applications*, such as the one we are considering in this article (motion detection and tracking applications).

3.3 Hermes Algorithm

We propose in this section a new approach that combines the existing ones and is capable of propagating fronts without any limitations on their speed function (Fig. 3). An alternative idea for the propagation of a given front is to work with the "strongest" point at each step. Since the PDE has been defined for the level set evolution, the contour depends on the time step and the velocity which varies from point to point. For the general case, we write

$$\Phi^{t+1}(x, y) = \Phi^t(x, y) + \mathcal{V}(x, y, \Phi)\delta t$$

Since the velocity $\mathcal{V}(x, y, \Phi)$ in many cases is estimated according to image characteristics, there are some pixels for which the front evolves in a faster way compared to the others. The key idea of this approach, is to evolve the contour according to the velocity values.

```
SetInitialContour()
SetActivePixels()
while ( LargestVelocity() !=0 ) {
    site = SelectLargestVelocity()
    EvolveLevelSet(site,neighbors)
    AddNeighborsActive(site)
    ReestimateVelocity(site,neighbors)
    if ( ActivePixels > Threshold){
        SmoothCharImage()
        FindContourPosition()
        SetActivePixels()
    }
}
exit()
```

Figure 3: Hermes Algorithm

We propose an algorithm which at each step selects the pixel with the largest velocity and evolves its characteristic value. First we initialize the contour and we set both the contour pixels and their neighbors as *active points*. We select from the active pixels the one with the largest velocity and we evolve it for a certain number of iterations as well as the values of the neighborhood pixels. Since there are modified image values, there are pixels which are affected (in terms of current velocity). For these pixels, we estimate their velocity. If there are neighborhood pixels which are not *active* we label them as *active*. Additionally, there is a special velocity cost term, which counts only for the selection of the "strongest" pixel (not for the level-set evolution), and depends on the number of times at which this pixel has been selected. Periodically we find the contour position in order to avoid the creation of a large set of *Active* pixels and we perform a smooth operation in the characteristic image, since this image is partially motivated. The algorithm stops when all the contour points (according to the latest estimation) have velocities close to zero, or there are no more *Active* pixels.

The key issue for an efficient version of the Hermes algorithm lies on a fast way of locating the grid point among Active points with the with the biggest velocity. For this reason, a variation of a heapsort algorithm is used. Initially all the Active points are sorted in a heapsort (so that the biggest member can be easily located). When a point is removed from the heapsort, the values of its neighbors are recomputed, and the results are bubbled upwards until they reach their correct locations. Moreover, whenever we want to add a point to the heapsort, we put it at the end and we process it in the same way.

The proposed algorithm can deal with cases at which *Fast Marching* cannot to be used. It

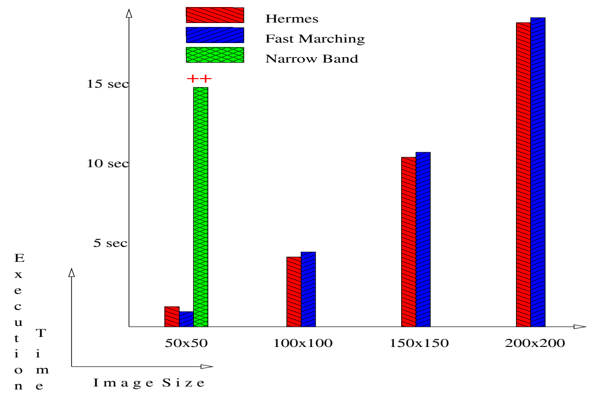

Figure 4: Computational Cost Diagram, for *Front Propagation* algorithms. The cost of *Narrow Band* approach does not appear for all image sizes, since it is much more greater then the cost of the two other approaches.

is completely independent on the form of the speed function, that is capable to cope with a large variety of level-set applications in image processing (*i.e* curvature based speed functions, etc.)(Fig. 6). Additionally, an efficient implementation has high convergence rate and can determine the result very quickly. In order to test the behavior of the proposed approach, we used it for the problem of geodesic active contour detection together with Narrow Band and Fast Marching. We used a synthetic image with four objects where we added gaussian noise. According to our experiments (Fig. 4) Hermes provides a little bit less computational cost than *Fast Marching*, while the cost of Narrow Band is at least ten times more. The cost of Hermes and Fast Marching algorithms seems to be proportional to the image size.

3.4 Computational Issues

In order to reduce further the computational cost, we propose a multi-scale technique which can be used combined with the front propagation algorithms. Thus, a Gaussian pyramid of images is built upon the full resolution image and similar geodesic contour problems are defined through the different levels. This multi-resolution structure is then utilized according to a coarse-to-fine strategy. In other words, an extrapolation of the current contour from level with low resolution to levels with finer contour configuration takes place. This extrapolation scheme is used as an initial contour, and a new contour evolution is performed. Usually this technique is applied to a pyramid with two or three levels.

Sequence (size)	Levels	Narrow Band	Fast Marching	Hermes Algorithm
Player	0	11.3 min	11.5 sec	5.8 sec
(*208x80*)	1	2.7 min	6.4 sec	3.1 sec
	2	1.4 min	4.6 sec	3.3 sec
Autoroute	0	14.8 min	11.2 sec	8.1 sec
(*128x128*)	1	3.8 min	4.8 sec	3.4 sec
	2	2.2 min	3.8 sec	3.3 sec
Football	0	30^{+} min	40.5 sec	30.1 sec
match	1	7.1 min	18.4 sec	12.9 sec
(*408x144*)	2	3.4 min	13.2 sec	9.8 sec

Figure 5: Computational Cost (CPU time)

4 Experimental Results - Discussion

Real-word video sequences have been used to test and validate the proposed approach.

For the detection part, we obtained very satisfactory results. The generated image, based on the inter-frame difference mixture analysis, fits exactly to the assumptions done during its creation (Fig. 2). In addition, the use of level-set approach allows to deal with a large variety of objects movements.

The *tracking* part doesn't always give the same quality of results as the detection part. In the case where the moving objects are surrounded by a smooth area, the quality of the solution is very close to the optimal (Fig. 6, 7). On the other hand, there are some limitations for the model, since it is not capable to deal with cases where there is a texture background (with edges) closed to the objects. Concerning the level-set implementation, two different well-known approaches have been implemented and a new one is proposed. These approaches are evaluated concerning their computational cost [1] and their set of applications (Fig. 5).

Summarizing, we have considered a level-set approach for *detection* and *tracking* in image sequences. The main contribution of our approach is a framework for detecting and tracking moving objects in a sequence of images using of level-set methodology. Based on the inter-frame difference, we are able to create an image where the detection and tracking can be viewed as a geodesic computation problem. A *new, very fast algorithm*, - **which can be used under any case of evolving speed** - for the front propagation problem, is proposed and compared with the existing well-known methods, which have been also implemented. In order to further reduce the computational cost, we use a multi-scale approach combined with the different algorithms of front propagation, which permits to track moving objects very fast.

An extended version of this paper can be found at [PD97]. Additionally various experimental results

[1] A SPARCstation20 with 64MB memory and a CPU at 75 MHz has been used

(a)

(b)

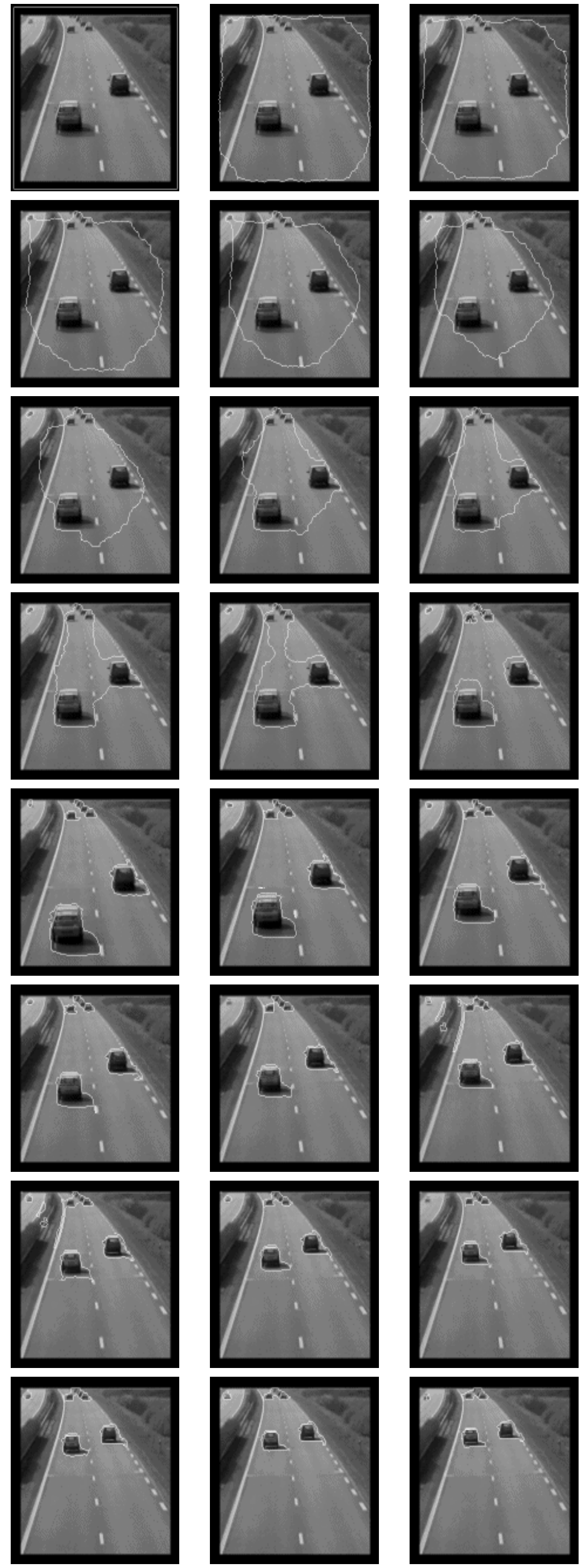

Figure 6: *Autoroute Sequence* (a) *Curve evolution* projected at the first image for *Autoroute Sequence* (left to right) with Hermes Algorithm. (b) *Tracking* Result (left to right).

(in MPEG format), including the ones shown in this article, can be found at: *http:// www.inria.fr/robotvis/personnel/nparagio/demos.html*

Acknowledgments

This work was funded in part under the VIRGO research network (EC Contract No ERBFMRX-CT96-0049) of the TMR European Community Programme.

References

[AS95] D. Adalsteinsson and J. A. Sethian. A Fast Level Set Method for Propagating Interfaces. *Journal of Computational Physics*, 118(2):269–277, 1995.

[CC96] V. Caselles and B. Coll. Snakes in Movement. *SIAM Journal on Numerical Analysis*, 33:2445–2456, December 1996.

[CKS95] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *Proceedings of the 5th International Conference on Computer Vision* [icc95], pages 694–699.

[DF96] Rachid Deriche and Olivier Faugeras. Les EDP en Traitement des Images et Vision par Ordinateur. *Traitement du Signal*, 13(6), 1996.

[DH73] R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* New York: Willey-Interscience, 1973.

[icc95] Boston, MA, June 1995. IEEE Computer Society Press.

[MSV93] R Malladi, J.A. Sethian, and B.C. Vemuri. A topology independent shape modeling scheme. *SPIE*, 2031:246, 1993.

[MSV95] R. Malladi, J. A. Sethian, and B.C. Vemuri. Shape modeling with front propagation: A level set approach. *PAMI*, 17(2):158–175, February 1995.

[OS88] S. Osher and J. Sethian. Fronts propagating with curvature dependent speed : algorithms based on the Hamilton-Jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.

[PD97] N. Paragios and R. Deriche. A PDE-based Level Set Approach for Detection and Tracking of Moving Objects. Technical Report 3173, INRIA, France, May 1997. http://www.inria.fr/rapports/sophia/RR-3173.html.

[PPT+96] N. Paragios, P. Perez, G. Tziritas, C. Labit, and P. Bouthemy. Adaptive Detection of Moving Objects, using Multiscale Techniques. In *Proceedings of the 3th International Conference on Image Processing*, volume 1, pages 525–528, 1996.

[RT92] E. Rouy and A. Tourin. A Viscosity Solutions Approach to Shape-from-Shading. *SIAM J. Numer. Anal.*, 29(3):867–884, June 1992.

[Set96] J. A. Sethian. A Fast Marching Level Set Method for Monotonically Advancing Fronts. In *Proc. Nat. Ac. Science*, volume 93, pages 1591–1694, 1996.

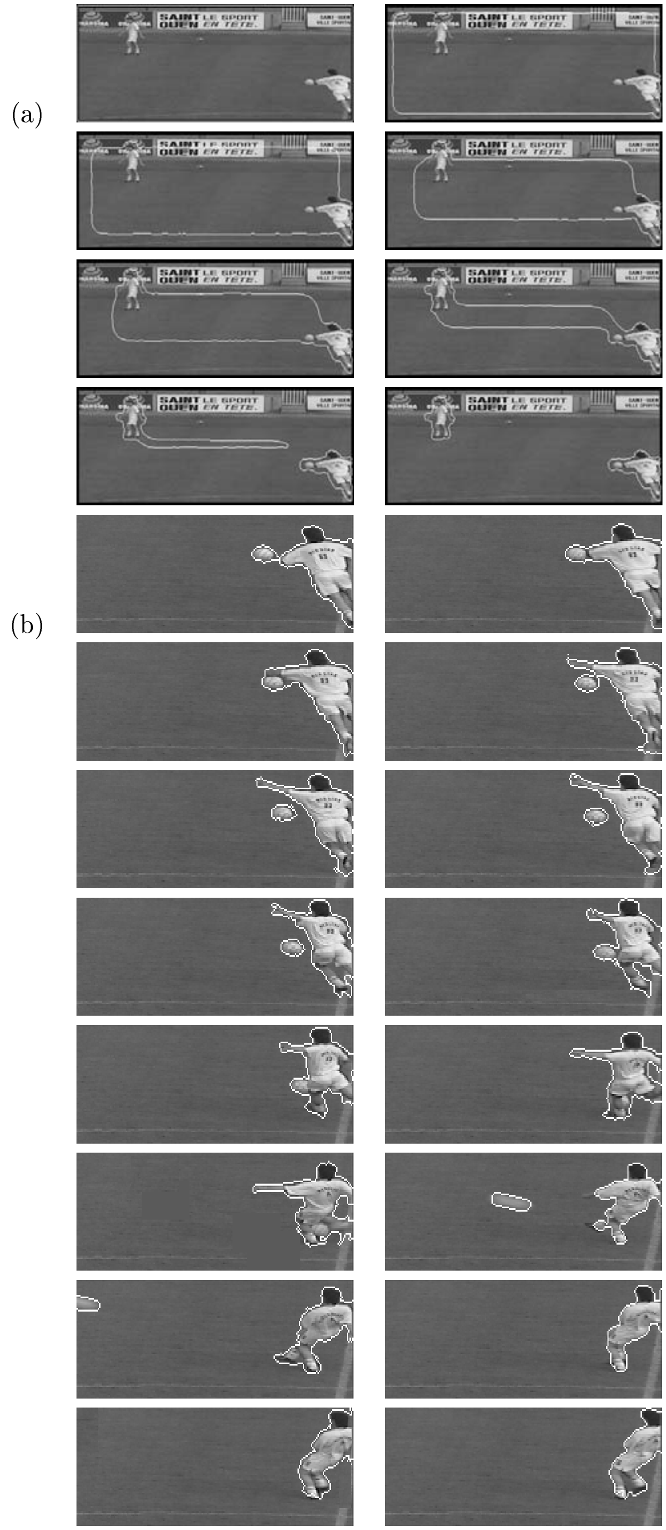

Figure 7: *Football match* Sequence (a) Curve evolution projected at the first frame (left to right) with Hermes Algorithm. (b) *Tracking* Result (left to right).