Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects

Nikos Paragios and Rachid Deriche

Abstract—This paper presents a new variational framework for detecting and tracking multiple moving objects in image sequences. Motion detection is performed using a statistical framework for which the observed interframe difference density function is approximated using a mixture model. This model is composed of two components, namely, the static (background) and the mobile (moving objects) one. Both components are zero-mean and obey Laplacian or Gaussian law. This statistical framework is used to provide the motion detection boundaries. Additionally, the original frame is used to provide the moving object boundaries. Then, the detection and the tracking problem are addressed in a common framework that employs a geodesic active contour objective function. This function is minimized using a gradient descent method, where a flow deforms the initial curve towards the minimum of the objective function, under the influence of internal and external image dependent forces. Using the level set formulation scheme, complex curves can be detected and tracked while topological changes for the evolving curves are naturally managed. To reduce the computational cost required by a direct implementation of the level set formulation scheme, a new approach named Hermes is proposed. Hermes exploits aspects from the well-known front propagation algorithms (Narrow Band, Fast Marching) and compares favorably to them. Very promising experimental results are provided using real video sequences.

Index Terms—Front propagation, geodesic active contours, level set theory, motion detection, tracking.

1 Introduction

The problem of detecting and tracking moving objects has a wide variety of applications in computer vision such as coding, video surveillance, monitoring, augmented reality, and robotics. Additionally, it provides input to higher level vision tasks, such as 3D reconstruction and 3D representation. This paper addresses the problem using boundary-based information to detect and track several nonrigid moving objects over a sequence of frames acquired by a static observer.

During the last decade, a large variety of motion detection algorithms have been proposed. Early approaches for motion detection rely on the detection of temporal changes. Such methods [1] employ a thresholding technique over the *interframe difference*, where pixelwise differences or block differences (to increase robustness) have been considered. The difference map is usually binarized using a predefined threshold value to obtain the motion/no-motion classification. A step forward in this direction was the use of statistical tests [2] that were constrained to pixelwise independent decisions. These tests assume intrinsically that the detection of temporal changes is equivalent to the motion detection. However, this assumption is valid when either large displacements appear or the object projections are sufficiently textured, but fails in the case of moving objects that preserve uniform regions. To avoid this limitation, temporal change detection masks (Gabor spatio-temporal change detectors) and filters have also been considered [3]. The use of these masks improves the efficiency of the change detection algorithms, especially in the case where some a priori knowledge about the size of the moving objects is available, since it can be used to determine the type and the size of the masks. On the other hand, these masks have limited applicability since they cannot provide an invariant change detection model (with respect to size, illumination) and cannot be used without an a priori context-based knowledge.

To overcome this, global energy frameworks that use more complex primitives and apply some spatial constraints to the segmentation process were introduced. The motion detection problem is formulated within a global minimization framework that combines attraction and regularity-based terms. In that direction, the spatial Markov Random Fields have been widely used [4], [5], [6] and motion detection has been considered as a statistical estimation problem. The optimal segmentation map is obtained by maximizing the a posteriori segmentation probability given the observed data. The optimization problem turns to be equivalent to the minimization of a global objective function and is usually performed using stochastic (Mean-field, Simulated Annealing) or deterministic relaxation algorithms (Iterated Conditional Modes, Highest Confidence First). Although MRF-based estimation is a very powerful paradigm, usually it is time consuming (especially when the solution space is large), a fact that constitutes a serious shortcoming.

Tracking goes further than motion detection and requires extra motion-based measurements, specifically the segmentation of the corresponding motion parameters. There are numerous research efforts dealing with the *tracking* problem and the existing approaches can be mainly classified in two distinct categories:

- *N. Paragios is with Siemens Corporate Research, Department of Imaging and Visualization, 755 College Road East, Princeton, NJ 08540. E-mail: nikos@cr.siemens.com.*
- *R. Deriche is with the Computer Vision and Robotics Group (RobotVis), I.N.R.I.A. Sophia Antipolis, BP. 93, 2004 Route de Lucioles, 06902 Sophia Antipolis Cedex, France. E-mail: der@sophia.inria.fr.*

Manuscript received 9 Nov. 1998; accepted 27 Aug. 1999.
Recommended for acceptance by R. Szeliski.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 108208.

- **Motion-based** approaches rely on robust methods for grouping visual motion consistencies over time [7], [8], [9]. These methods are relatively fast but have considerable difficulties in dealing with nonrigid movements and objects.
- **Model-based** approaches impose high-level semantic representation and knowledge and, therefore, are more reliable compared to the motion-based ones [10], [11], [12], [13], [14]. The model space can be either the real 3D world, or the 2D projection (image) space. These methods suffer from high computational costs for complex models due to the need for coping with scaling, translation, rotation, and deformation.

In both cases, tracking is performed using measurements provided by geometrical or region-based properties of the tracked object. In this direction, there exist two main approaches:

- The **boundary-based** approaches (also referred to as edge-based) rely on the information provided by the object boundaries (image-based shape properties) [15], [16],
- The **region-based** approaches rely on information provided by the entire region such as texture and motion-based properties [9], [5].

The idea of using boundary-based features to provide tracking has been widely adopted. The boundary-based features (edges) are well-adapted to the tracking problem since they provide reliable information which does not depend on the motion type, or object shape. Usually, the boundary-based tracking algorithms employ active contour models, like snakes [17], balloons [18], [19], and geodesic active contours [20], [21]. These models are energy-based [20], [21] or geometric-based minimization approaches [22] that evolve an initial curve under the influence of external potentials, while it is being constrained by internal energies. The snakes are usually parameterized (using B-splines) and the solution space is constrained to have a predefined shape [18], [15], [23], [16]. These methods require an accurate initialization step since the initial contour converges iteratively toward the solution of a partial differential equation. In the case of the geodesic active contour models, there is not such a constraint since the former are relatively free of the initialization step. Besides, these models are not parameterized and can be used to track objects that undergo nonrigid motion [24].

On the other hand, the region-based methods use a motion estimation/segmentation technique. In this case, the estimation of the target's velocity is based on the correspondence between the associated target regions at different time instants [25], [26], [5]. This operation is usually time consuming (a point-to-point correspondence is required within the whole region) and is accelerated by the use of parametric motion models that describe the target motion with a small set of parameters. The use of these models introduces the difficulty of tracking the real object boundaries in cases with nonrigid movements/objects, but increases robustness due to the fact that information provided by the whole region is exploited.

This paper describes a unified approach for the *detection* and *tracking* of moving objects by the propagation of curves [29]. Thus, an original scheme is proposed that may be viewed as a geodesic active motion detection and tracking model which basically attracts the given curves to the bottom of a potential wellcorresponding to the boundaries of the moving objects.

Initially, a statistical analysis is performed and is used to provide the motion-based information. According to this modeling phase, the assumption that the observed *interframe difference* density function is a two component mixture model is considered. These two components are zero-mean and correspond to the static (background) and to the mobile population (moving objects). Then, using Bayes rule, the conditional objects boundary probability given the observed data is estimated. This information is used to solve the object detection problem. Additionally, by assuming a smooth background, the input frame can be used directly to provide an accurate object tracking result. The detection and the tracking problem are dealt with simultaneously, using a geodesic active contour model that permits an initial curve to evolve towards a minimum length geodesic active curve that takes into account the desired image characteristics under the influence of internal and external forces. The objective function is minimized using a gradient descent method, where the associated partial differential equation is implemented using the level-set methodology [30], [31], which provides some very nice properties. Topological changes (splitting and merging) are naturally handled, intrinsic geometrical properties can be estimated directly from the level-set frame, and a quite stable numerical approximation scheme is applicable. For the front propagation problem, two well-known schemes are used, namely the Narrow Band [32] and the Fast Marching [31] approaches. A new scheme is proposed, called HERMES, which is also evaluated compared to the existing schemes. Finally, in order to further reduce the execution time, a multiscale approach has also been considered. Very promising experimental results are provided using real video sequences.

The most closely related work with the one proposed in this paper can be found in [24] and, more recently, in [27], [28].

In [24], a three step approach is proposed which is very different from the unified approach presented in this article. There, the authors following their previous work on geodesic active contours, 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. At this step, very unstable measurements can be obtained. Following this, a PDE is designed to move the contours to the boundary of the moving objects. These contours are then used as initial estimates of the contours in the next image and the process iterates.

More recently, in [27], a front propagation approach that couples two partial differential equations to deal with the problems of object tracking and sequential segmentation was proposed. Additionally, in [28], a new, efficient numerical implementation of the geodesic active contour model has been proposed which was applied to track objects in movies.

The paper is organized as follows: Section 2 introduces the level set formulation and the geodesic active contour model which is the basis of our approach. Besides, the existing front propagation algorithms are briefly explained

in this section, while a new approach (the HERMES algorithm) is proposed. Section 3 illustrates the detection and the tracking approach, by proposing a unified model for both problems. Finally, Section 4 presents discussion, experimental results, and concluding remarks.

2 Propagating Curves

2.1 Level Set Theory

Let $[C : [0,1] \rightarrow \mathcal{R}^2,\ p \rightarrow C(p)]$ be a parameterized closed initial planar curve in a Euclidean plane and $C(p,t)$ the family of curves which is generated by the movement of an initial curve $C_0(p)$ in the direction of its inward Euclidean normal vector $\mathcal{N}$. We assume that the speed of this movement is a scalar function $[F]$ of the curvature $\mathcal{K}$:

$$\begin{cases} C_t = F(\mathcal{K})\,\mathcal{N} \\ C(p,0) = C_0(p). \end{cases} \tag{1}$$

In order to implement the curve evolution according to the above equation, we can consider a Lagrangian approach and produce the associated equations of motions for the position vector $(x,y) = C(p)$. These positions are updated using a difference approximation scheme. The main drawback of this approach is that the evolving model is not capable of dealing with topological changes of the moving front.

This can be avoided by considering the approach of Osher and Sethian [30]. According to them, the initial curve $C_0(p)$ is represented by a zero-level set ($\phi = 0$) function of an initial surface z (Fig. 1)

$$\left[z = (x,y,\phi(x,y,t)) \in \mathcal{R}^3\right].$$

Using (1) and taking the derivative of $\phi(x,y,t) = 0$ with respect to time and space, the following associated equation of motion for the surface $\phi()$ can then be easily derived:

$$\begin{cases} \phi_t = -F(\mathcal{K})\,|\nabla\phi| \\ \phi(C_0(p),0) = 0, \end{cases} \tag{2}$$

where $|\nabla\phi|$ denotes the gradient norm. Thus, there is a connection between the family of moving curves $C(p,t)$ and the family of one parameter evolving surfaces $\phi(x,y,t)$. This connection is due to the fact that the zero level set values of the function ϕ always yields to the moving front. This propagation framework has numerous advantages. More specifically, the evolving function ϕ always remains a function as long as F is smooth. However, the propagating curve $C(p)$ may change topology as the function ϕ evolves. Additionally, due to the fact that $\phi(p,t)$ remains a function during its evolution, numerical simulations may be developed very easily and an explicit finite difference approach is possible. Finally, intrinsic geometric properties of the curve can be estimated directly from the level set function, (e.g., normal, curvature) and the method can be very easily extended to deal with problems in higher dimensions [33].

2.2 Geodesic Active Contours

Let $I : [0,a] \times [0,b] \rightarrow \mathcal{R}^+$ be a given input in which the task of extracting the object contours is considered.

The geodesic active contour model [20], [21] was introduced as a geometric alternative for snakes [17], [34] and aims at finding the curve $(C)(p)$ that minimizes the following energy:

Fig. 1. Level set methodology and curve propagation. The left column shows the evolving level set function, while, on the right, the corresponding curve that is the zero level set values of the surface is illustrated. The mechanism that allows changes of topology is also demonstrated.

$$E[(C)(p)] = \int_0^1 \underbrace{g(\ |\nabla I(C(p))|\ )}_{Boundary\ attraction}\ \underbrace{|\ \dot{C}(p)\ |}_{Regularity}\, dp, \tag{3}$$

where

- $\dot{C}(p)$ is the partial derivative of curve with respect to its parameter p

$$\left[\dot{C}(p) = \frac{\partial C}{\partial p}(p)\right]$$

  and
- $g(.)$ is a monotonically decreasing function such that $g(r) \rightarrow 0$ as $r \rightarrow \infty$, and $g(0) = 1$ (e.g., a Gaussian function [35]).

The energy interpretation is clear since, when the detection of the object contours is equivalent to finding the geodesic curve that best takes into account the desired image characteristics (edges) [36].

This objective functions is minimized by solving the associated Euler-Lagrange equation. According to it, the flow that deforms the initial curve $C(p,0) = C_0(p)$ toward the minima of (3) is given by a steady state solution of:

$$C_t = [g(|\nabla I|)\mathcal{K} - \nabla g(|\nabla I|)\cdot\mathcal{N}]\mathcal{N}, \tag{4}$$

where t denotes the time as the contour evolves and $\mathcal{N}$ is the inward Euclidean normal vector to the curve $C(p,t)$.

By introducing the level set formulation, curve C can be considered as the zero level set of a function $\phi : [0,a] \times [0,b] \rightarrow \mathcal{R}$. This model is parameter-free, as well as topology-free, since different topologies of the zero level set correspond to the same topology of ϕ. Based on (1), it can be shown that a steady state solution of this geodesic problem is given by:

$$\phi_t = g(|\nabla I|)\mathcal{K}|\nabla\phi| + \nabla g(|\nabla I|)\cdot\nabla\phi, \tag{5}$$

where $\phi(x,y,0) = \phi_0(x,y)$ and the curve C is represented by the zero level values of ϕ. The normal $\mathcal{N}$, as well as the curvature value $\mathcal{K}$, can be estimated directly from the level-set function ϕ

$$\left[\mathcal{N} = -\frac{\nabla\phi}{|\nabla\phi|}, \mathcal{K} = \mathbf{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right].$$

The geodesic active contour model compares favorably to the classical snake due to the fact that it does not depend on **the curve parameterization**. Furthermore, due to the level set implementation, **topological changes are naturally handled**, which allows detection of all the objects which appear in the image plane without knowing their exact number.

2.3 Front Propagation Algorithms

A direct implementation of (2), (5) involves the reestimation at the level set function of all image pixels (not simply the zero level set corresponding to the front itself). This front propagation method is computationally expensive due to numerous useless calculations that are performed for pixels which are out of interest during the front propagation. In order to overcome this drawback, two different methods have been proposed: the *Narrow Band* [32], which is still computationally expensive, and the *Fast Marching* [31] that is a very fast front propagation framework but has limited applicability. Here, a new method is proposed, called the *Hermes* algorithm [29], that combines the rapidity of *Fast Marching* with the generality of *Narrow Band*.

2.3.1 Narrow Band Approach

The key idea is to deal only with pixels which are close to the latest position of the zero level-set contour in both directions (inward and outward). This is known as *Narrow Band* Approach, which has been initially proposed in [37] and extensively analyzed and optimized in [32]. 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 affect the evolution process. Therefore, only pixels close to the current contour position are considered. A set of narrow band pixels is defined around the latest contour position and the level set function is updated only within this band. The problem is that the curve position changes dynamically (from iteration to iteration). As a consequence, the Narrow Band also has to be updated from iteration to iteration. This will increases the cost dynamically (in terms of complexity), thus, the contour position and the set of narrow band pixels are updated only in cases where the contour is very close to the borders of the current band. A significant cost reduction is achieved through this approach, but the cost still remains considerable.

2.3.2 Fast Marching Approach

This algorithm has been proposed in [38], [39] and can be used in cases with monotonically advancing fronts, that is, fronts moving with a velocity $[F]$ which is always positive (or negative), leading to a particular stationary level set equation for the crossing time $[T]$ given by $[\ |\nabla T|F = 1\ ]$.

This method provides an extremely fast scheme to solve problems of the above form and relies on a coupling of the narrow band methodology with a fast heap-sort algorithm. The key idea is the observation that the information propagates "one way," that is, from smaller values of T to larger ones. Thus, the front is swept ahead in an upwind fashion by considering a set of pixels in a narrow band around the existing front and, to march this narrow band forward, freezing the values of existing pixels and bringing new ones into the narrow band structure. The main handicap of this approach is that it requires propagation velocities that have constant sign, thus it cannot be applied to cases with curvature-dependent speed functions.

2.3.3 Hermes Algorithm

In this section, a new approach is proposed that combines the *Narrow Band* and the *Fast Marching* method by employing the idea of a selective propagation (*Fast Marching*) over a relatively small window (*Narrow Band*). Thus, the curve propagation is speeded up by introducing the idea of propagating the front in a pixel that evolves **faster** at each step.

A given level set PDE can be rewritten as:

$$\phi^{t+1}(x,y) = \phi^t(x,y) + \mathcal{V}(x,y,\phi^t)\ \ \delta t. \tag{6}$$

This PDE indicates that the front is propagating differently with respect to the image and the geometric properties of each pixel. The *Narrow Band* method updates the level set within a band which may contain pixels with *zero*-valued propagation velocities. In these pixels, the front remains static, resulting in a significant amount of redundant computations.

This problem can be avoided by introducing the idea of evolving the front very locally according to the associated propagation velocities. This leads to a **smart** *Narrow Band* method that uses ideas from *Fast Marching* (e.g., fastest pixel), resulting in a drastic decrease of the required computational cost. Thus, at each step, the proposed approach selects the pixel of the front, preserving the highest absolute propagation velocity, and performs a local evolution to the level set frame within a circular window centered on this pixel. An overview of the algorithm is shown in Fig. 2, while a detailed description follows:

A. **Initialization Procedure.** The level set function is initialized as a signed distance function from the front. The maximum distance (in terms of absolute value) is considered to be the radius of the active window (see **step B**). Then, the front pixels are introduced to the active list [SETACTIVEPIXELS()].

B. **Marching Procedure.** At each step, the pixel [c] with the highest absolute propagation velocity is selected from the active list [SELECTHIGHESTVELOCITY()]. If there are several pixels with equal propagation velocities, then the **first in, first out (FIFO)** rule is applied with respect to their entrance times in the active list.

   1. A centralized circular window is defined around this pixel $[c]$ (the radius of this window is usually two) and the level set function is updated locally within this window. To ensure stability, different time steps for the pixels $[s]$ within this window are used. These time steps are inversely proportional to the geometric distance from the window center

$$\left[\delta t(s) = \frac{\delta t}{1 + ||s - c||}\right].$$

   This modification does not affect the propagation process since, if the front moves, this will

```
SetActivePixels();
iterationNumber = 0;
while ( iterationNumber < maxIteration ){
    iterationNumber++;
    site = SelectHighestVelocity();
    EvolveLevelSetLocally(site);
    FindFrontLocally(site);
    RemoveInActivePixels(site);
    AddNewActivePixels(site);
    ReInitializeLevelSetLocally(site);
    UpdateAffectedVelocities(site);
    if ( ReachConvergence() )
        break;
    }
FindLatestFrontPosition();
exit();
```

Fig. 2. Hermes algorithm in pseudosymbolic language.

first happen for the selected pixel (e.g., highest propagation velocity), which is also associated with the most important time step. This operation is suspended if the front moves or a certain number of iterations is reached [EVOLVELEVELSETLOCALLY()]. As for the propagation velocities within the local window, they are estimated only at the beginning and they are not affected by the local changes of the level set function.

2. When the local level set evolution is completed, then the front is extracted within the active window by considering the zero level set crossings [FINDFRONTLOCALLY()] and the level set function is reinitialized locally [REINITIALIZELEVELSETLOCALLY(.)]. The modified local front [current] is compared to the one before evolution [previous]. The following cases are considered for the pixels within the local window:
   - If a pixel is a new front point (it belongs in the current front, while it was not a front pixel before the local evolution), then it is added to the active list [ADDNEWACTIVEPIXELS()],
   - If a pixel is an ex-front point (it was part of the front before but not anymore), then it is removed from the active list [REMOVEINACTIVEPIXELS()],
   - Finally, if a pixel belongs to the current and the previous front, then it is affected (in terms of the propagation velocity) by the local changes of the level set function. Hence, a reestimation of its velocity is performed [UPDATEAFFECTEDVELOCITIES()].

C. **Control Procedure.** If a certain number of iterations is reached or the front does not further move [REACHCONVERGENCE()], then the operation is completed, and the final front position is extracted [FINDLATESTFRONTPOSITION(.)], otherwise, **step B** is repeated.

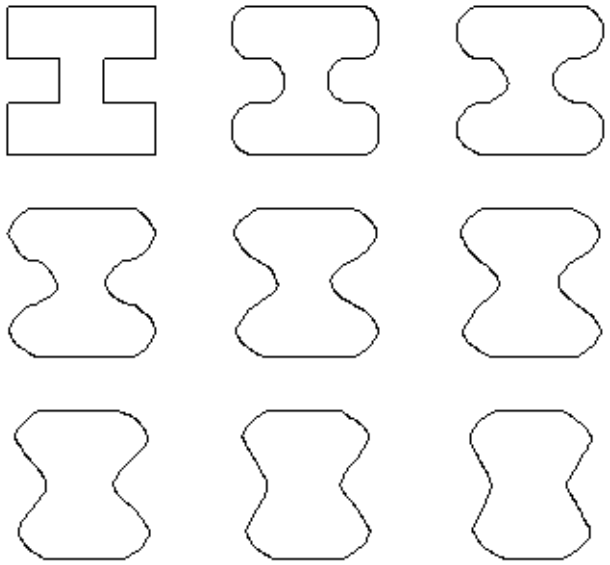

Fig. 3. Mean curvature flow (top to bottom, left to right) according to the **HERMES** algorithm $[\phi_t = \mathcal{K}|\nabla\phi|]$ (e.g., minimizing the length). The curvature values are estimated directly from the level set function using a 3×3 window and the max/min mode scheme. The propagation grid is 100×100.

The key issue for an efficient implementation of the *Hermes* algorithm is a fast way for locating the grid pixel among the front pixels having the highest propagation velocity. This can easily be done using a variation of a heap-sort algorithm [40]. Initially, all the active pixels are sorted in a heap-sort (so that the highest velocity member can be easily located). When the propagation velocity of a pixel is changed, then it is bubbled upward (or downward) until it reaches its correct position. Whenever a new pixel is added to the heap-sort, it is placed at the end and the processed in the same manner.

The proposed algorithm does not solve exactly the given partial differential equation in terms of the intermediate levels of the curve evolution since these levels are not always compatible with the expected ones. This is due to the locality that has been introduced during the curve propagation. On the other hand, the final solution is obtained very fast and respects the curvature-based geometrical constraints that are imposed by the energy functional. Besides, the *Hermes* algorithm is independent of the form of the speed function and is capable of coping with a large variety of level-set applications in image processing (i.e., curvature based speed functions (Fig. 3), positive/negative speed functions, etc.). Additionally, it has an efficient implementation with a high convergence rate.

A fundamental issue is raised here; specifically, whether it is more important to obtain the final solution by respecting the constraints that are imposed by the motion equation on the intermediate levels, rather to obtain the same final solution much more rapidly by loosely respecting these constraints in the intermediate levels. The *Hermes* algorithm follows the second path, where the constraints are at least respected in the final solution.

3 Detection and Tracking

3.1 Defining the Model

Let $I(s;t)$ be the current and $I(s;t-1)$ be the previous input frame and let $D(s;t)$ be the *interframe* gray level difference frame:

$$D(s;t) = I(s;t) - I(s;t-1). \tag{7}$$

The motion detection problem can be viewed as a "binary" decision for each pixel on the frame grid. There exist two possible cases, the *static* one corresponding to a pixel that belongs to the background in both frames and the *mobile* case corresponding to a pixel that belongs to a moving object in the current or in the previous frame. Let $p_D(d)$ be the probability density function of the observed *interframe* difference frame (histogram of the interframe difference). This density function is assumed to be a mixture model of a static and a mobile component. Let $p_S(d)$ be the conditional static probability and let $p_M(d)$ be the conditional mobile probability given the observed *interframe difference* data [6]. These probability density functions are assumed to be homogeneous (i.e., independent of pixel location).

The observed difference values are assumed to be obtained by selecting an hypothesis $L \in \{S : static,\ M : mobile\}$ with a priori probability P_L and, then, selecting a value d according to the probability law $p_L(d)$. Thus, the observed probability density function can be decomposed as:

$$p_D(d) = P_S\ p_S(d) + P_M\ p_M(d). \tag{8}$$

It is assumed that the conditional probability density functions are zero-mean and follow the Laplacian

$$p_x(d) = \frac{\lambda_x}{2} e^{-\lambda_x |d|} \tag{9}$$

or the Gaussian law

$$p_x(d) = \frac{1}{\sqrt{2\pi}\sigma_x} e^{-\frac{d^2}{2\sigma_x^2}}. \tag{10}$$

The static component is zero-mean since this population contains the difference values that are estimated from the projected intensities of the same 3D point at different time instants. Additionally, it is assumed that the mobile objects preserve uniform regions and important difference values appear at the occluded and disoccluded regions of the background due to the objects motion. These parts are expressed statistically by the tail of the conditional density function.

The estimation of the unknown parameters of this model $\{(P_L, \Theta_L) : L \in \{S : static,\ M : mobile\}\}$, where $[\Theta = \{\lambda|\sigma\}]$ is done using a gradient descent method derived by the maximum likelihood principle [41], where the parameter vector $(\hat{P}, \hat{\Theta})$ is the value of $(\hat{P}, \hat{\Theta})$ that maximizes the joint density

$$p(\mathcal{X}|P, \Theta) = \prod_{d \in D} p_D(d|P, \Theta). \tag{11}$$

An example of this analysis is illustrated in Fig. 4, where the first, second, and third order moments are used to provide an initial solution.

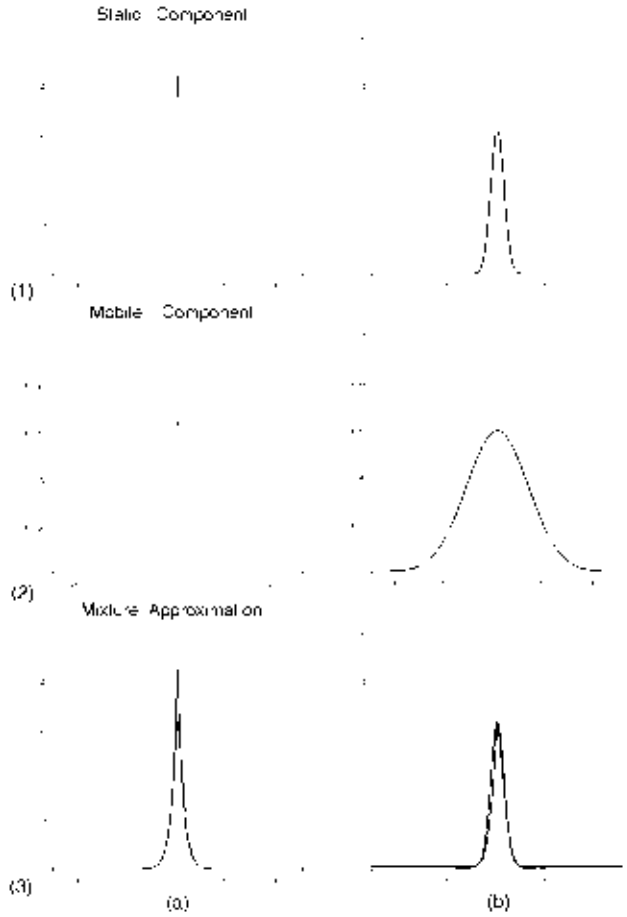

Fig. 4. Motion detection-based statistics for *player sequence* (Fig. 5). (a) Using **Laplacian** modes, *Iterations number: 14*, *Mean Approximation error* $\approx$ *:0.00032.* (b) Using Gaussian modes, *Iterations number: 06*, *Mean approximation error* $\approx$ *0.00047*. (1) Static component, (2) mobile component, and (3) input density function (gray line), mixture approximation (black line).

3.2 Setting the Energy

3.2.1 Detection Part

Let s be a grid location, and N be a partition of its neighborhood into two $[N_R(s), N_L(s)]$ local regions. Additionally, let $D(N(s))$ be the corresponding difference data in this neighborhood.

If $[p(B|D(N(s)))]$ is the probability of s being at the boundaries between a moving object and the static background, then according to the Bayes rule is given by:

$$\begin{aligned} p(B|D(N(s))) &= \frac{p_B(D(N(s)))}{p(D(N(s)))} p(B) \\ &= \frac{p_B(D(N(s)))}{p(D(N(s))|B \cup \bar{B})} p(B) \\ &= \frac{p_B(D(N(s)))}{p_B(D(N(s))) + p_{\bar{B}}(D(N(s)))}, \end{aligned} \tag{12}$$

where

- $p_B(d)$ is the conditional boundary probability,
- $p_{\bar{B}}(d)$ is the conditional nonboundary probability,
- and $p(B)$ is the a priory boundary probability, which is a constant scale factor and thus can be ignored.

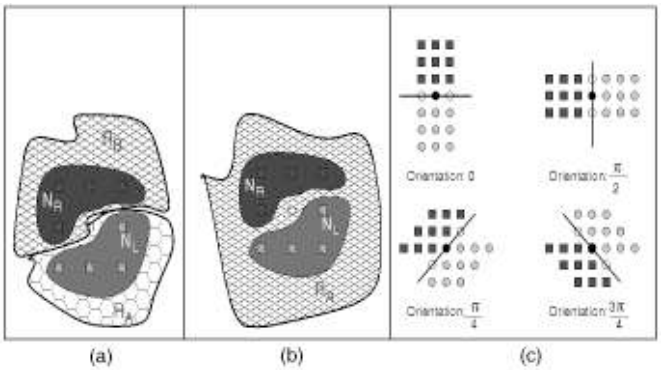

Fig. 5. (a) Neighborhood partition that indicates a **boundary** pixel, (b) neighborhood partition that indicates a **nonboundary** pixel, and (c) neighborhood partitions that are considered.

The conditional boundary/nonboundary probability can be easily estimated since

1. If s is a boundary pixel, then there exists a partition of its neighborhood $[N_L(s), N_R(s)]$, where the most probable assignment for $N_L(s)$ is the *Static* hypothesis while for $N_R(s)$ is the *Mobile* hypothesis or vice-versa (Fig. 5a),
   boundary condition:
$$\left[\begin{array}{l}(N_R \in static \cap N_L \in mobile) \\ \qquad \cup (N_R \in mobile \cap N_L \in static)\end{array}\right],$$
2. If s is not a boundary pixel, then, for every possible neighborhood partition, the most probable assignment for $N_L(s)$, as well as for $N_R(s)$, is either the *Static* or the *Mobile* hypothesis (Fig. 5b).
   nonboundary condition:
$$\left[\begin{array}{l}(N_R \in static \cap N_L \in static) \\ \qquad \cup (N_R \in mobile \cap N_L \in mobile)\end{array}\right].$$

As a consequence, the conditional boundary $[p_B()]$ and nonboundary $[p_{\bar{B}}()]$ probabilities can be estimated directly from known quantities using the following formulas:

$$\begin{aligned} p(D(N(s))|B) &= p(D(N(s))|[N_R \in static \cap N_L \in mobile]) \\ &\quad + p(D(N(s))|[N_R \in mobile \cap N_L \in static]) \\ &= p_S(D(N_L(s)))\ p_M(D(N_R(s))) \\ &\quad + p_M(D(N_L(s)))\ p_S(D(N_R(s))) \end{aligned}$$

$$\begin{aligned} p(D(N(s))|\bar{B}) &= p(D(N(s))|[N_R \in static \cap N_L \in static]) \\ &\quad + p(D(N(s))|[N_R \in mobile \cap N_L \in mobile]) \\ &= p_S(D(N_L(s)))\ p_S(D(N_R(s))) \\ &\quad + p_M(D(N_L(s)))\ p_M(D(N_R(s))). \end{aligned}$$

Since the probability that pixel s lies on the boundary of a moving object is defined, the next problem is to define the neighborhood partition. Four different partitions of the neighborhood are considered, namely the vertical, the horizontal, and along the two diagonals (Fig. 5c). These partitions can be obtained by assuming four different orientations

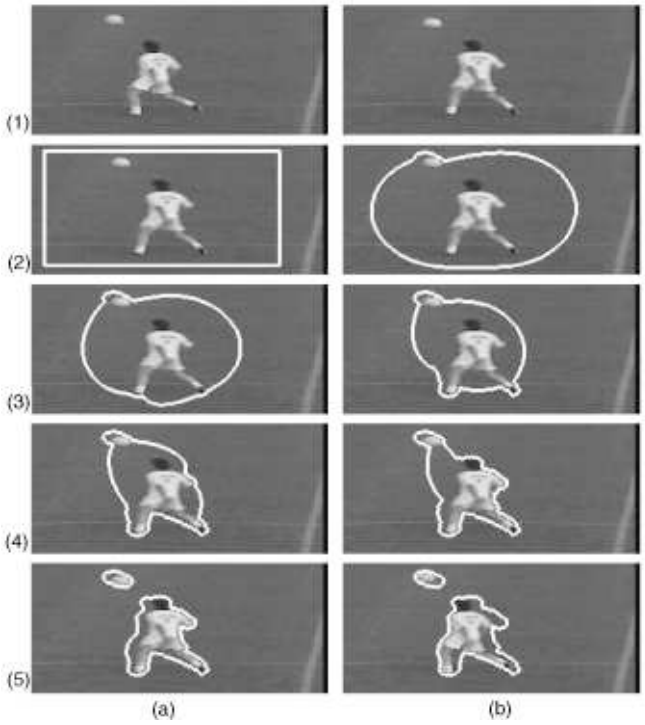

Fig. 6. Motion detection for the soccer sequence. Two consecutive frames are shown in the first row and the curve propagation in the other rows (left to right, top to bottom). Curve propagation: Narrow Band $[C_t = \{g(p_B)\mathcal{K} + \nabla\ g(p_B) \cdot \mathcal{N}\}\mathcal{N}]$. (1a) Previous frame, (1b) current frame, (2a) initial curve, (5a) motion detection area projected in the current frame, and (5b) motion detection area projected in the previous frame.

$$\left[\theta = \left\{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\right\}\right],$$

while the corresponding neighborhood regions are 3×3 directional windows. Using these partitions, the following vector of boundary probabilities is obtained:

$$B(s) = \left[\begin{array}{c} \frac{p_B(D(N(s)),0)}{p_B(D(N(s)),0)+p_{\bar{B}}(D(N(s)),0)} \\ \frac{p_B\left(D(N(s)),\frac{\pi}{4}\right)}{p_B\left(D(N(s)),\frac{\pi}{4}\right)+p_{\bar{B}}\left(D(N(s)),\frac{\pi}{4}\right)} \\ \frac{p_B\left(D(N(s)),\frac{\pi}{2}\right)}{p_B\left(D(N(s)),\frac{\pi}{2}\right)+p_{\bar{B}}\left(D(N(s)),\frac{\pi}{2}\right)} \\ \frac{p_B\left(D(N(s)),\frac{3\pi}{4}\right)}{p_B\left(D(N(s)),\frac{3\pi}{4}\right)+p_{\bar{B}}\left(D(N(s)),\frac{3\pi}{4}\right)} \end{array}\right], \tag{13}$$

where the different elements correspond to the different neighborhood partitions (with respect to θ). The highest element of the above vector is assigned to the motion detection boundary image $\{I_D\}$.

Then, the detection and tracking problem is expressed using the framework of energy minimization; under the conditions of a geodesic active problem; an energy function is associated to the given curve that has to be minimized with respect to the curve length and position:

$$E[(C)(p)] = \int_0^1 g(I_D(C(p)))\left|\dot{C}(p)\right|, \tag{14}$$

where

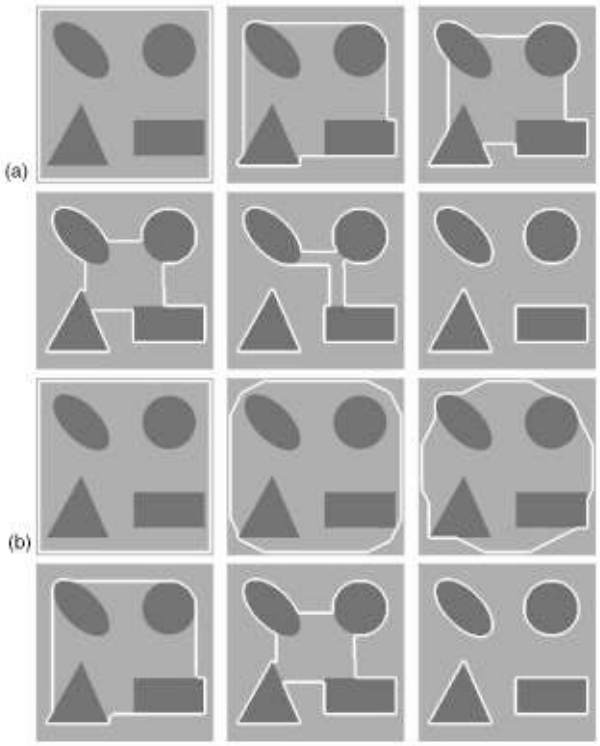

Fig. 7. Front propagation with Hermes algorithm: (a) $\phi_t = F|\nabla\phi|$, (b) $\phi_t = [F \mathcal{K} |\nabla\phi| + \nabla F \cdot \nabla\phi]$. In both cases, the F is a Gaussian function that captures the object boundaries (see Section 4.1).

$$\left[g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \right]$$

is a Gaussian function.

Using this energy functional, the geodesic active contour framework permits the detection of the areas corresponding to moving objects. In the case of objects with holes, this result is not equivalent to the moving area since there are some regions inside the estimated contour which correspond to static image regions. These regions can be easily detected by defining a curve inside the motion detection area that evolves outward, guided by the same propagation forces.

3.2.2 Tracking Part

Complete motion detection is not equivalent to temporal change detection. This is because the area estimated as moving between two successive images corresponds to the union of the moving object locations in these images. The goal is to track the objects in these images. To achieve this, the following modification of the geodesic active contour model is required:

$$E[(C)(p)] = \int_0^1 \left(\underbrace{\gamma\ g(I_D(C(p)), \sigma_D)}_{Motion\ Detection\ Term} + \underbrace{(1-\gamma)\ g(|\nabla I(C(p);t)|, \sigma_T)}_{Tracking\ Term} \right) \underbrace{\left|\dot{C}(p)\right|}_{Regularity} dp,$$

where σ_D, σ_T are the variances of the Gaussian functions that capture the motion detection and the tracking information. The detection term forces the curve to converge towards the moving area, avoiding edges or static objects. On the other hand, since the curve includes the moving area, this term is close to zero. The tracking term is then used for evolving the curve until it coincides with the exact location of the moving object. The assumption that the region between the object position and the motion detection result preserves significant edges only at the object boundaries is made. Finally, $\gamma \in [0, 1]$ is a parameter which balances the contribution of the detection and the tracking term. Selecting a value for γ close to *zero*, the model is forced to detect the moving objects (using a geodesic curve) while a value close to *one*, results in a geodesic active contour model.

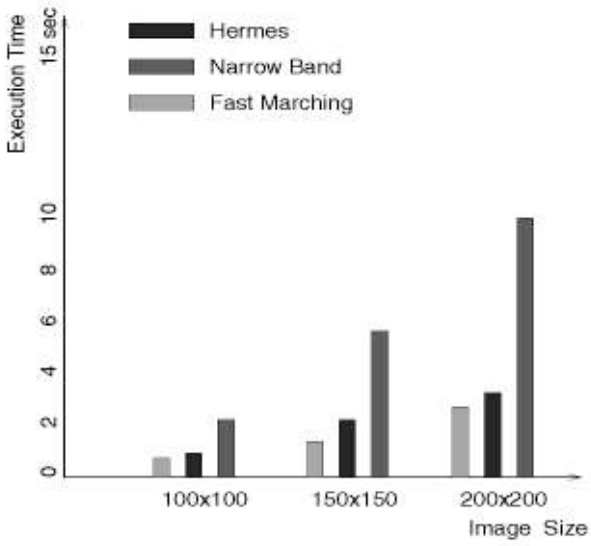

Fig. 8. Front propagation algorithms computational cost. Evolution: $[\phi_t = F|\nabla\phi|]$, time step: 0.1, narrow band size: 10, it Hermes window radius: 2.

3.3 Multiscale Approach

In order to further reduce the computational cost, a multiscale technique is proposed that can be combined with the front propagation algorithms. Specifically, a Gaussian pyramid of images is built upon the full resolution image and similar geodesic contour problems are defined across the different levels. This multiresolution structure is then utilized according to a coarse-to-fine strategy. In other words, an extrapolation of the current contour from a level with low resolution to levels with finer contour configuration takes place. Usually, this technique is applied to a pyramid with two or three levels. At the low resolution levels, the detection problem is solved by setting the parameter γ equal to *one*. Additionally, at the finest resolution level, the tracking problem is dealt with by setting the parameter γ equal to *zero* and the initial curve (corresponding to the detection) is evolved toward the real object boundaries.

3.4 Minimizing the Energy

Following the work on geodesic active contours presented in the previous section, the minimization of the defined geodesic active contour-based motion detection and

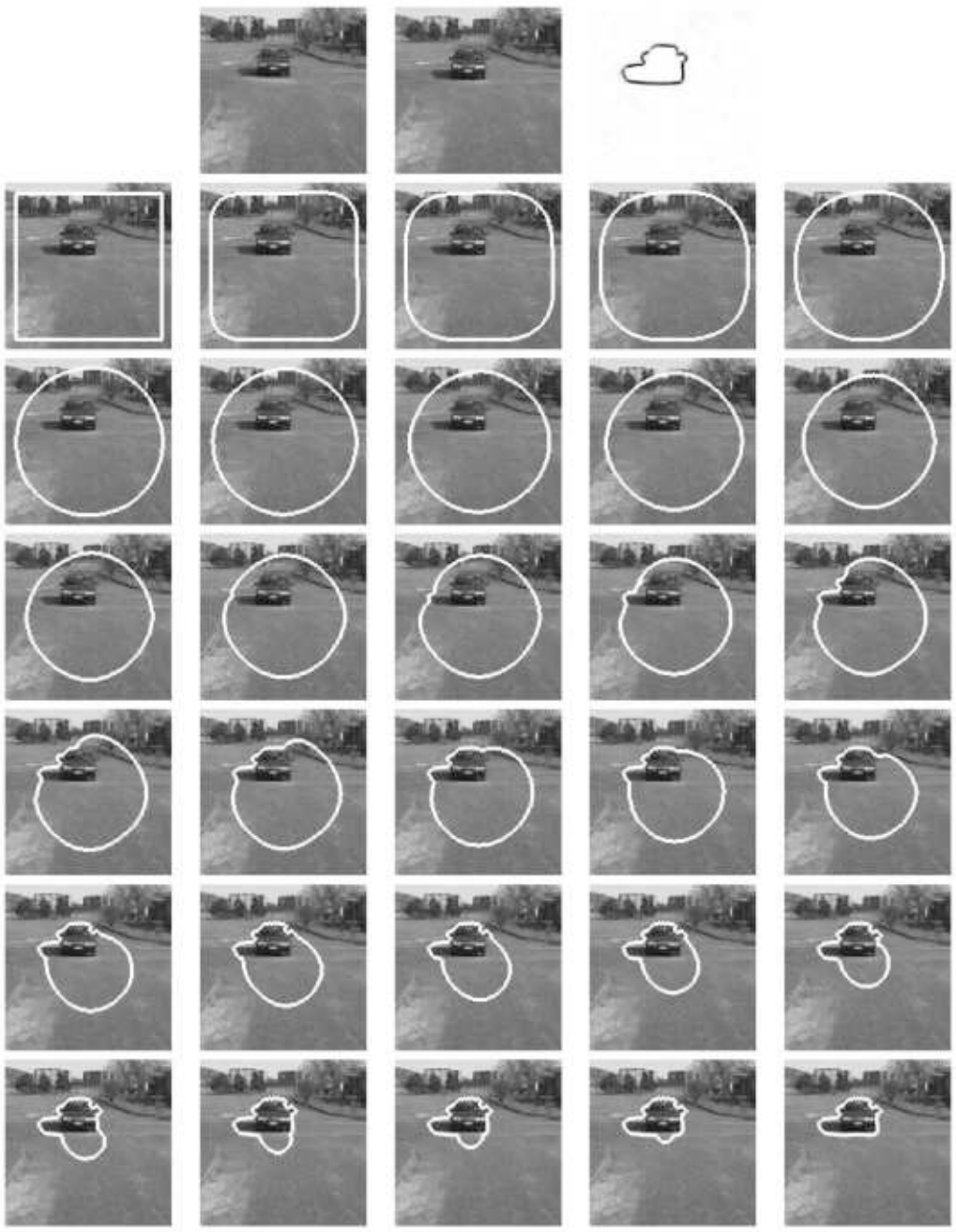

Fig. 9. Motion detection for the car sequence ($\gamma = 1$). Image size: 128×128. First row: first, second frame and motion detection boundaries. Front propagation algorithm: *Narrow Band*, algorithm parameters: $[band\_size = 8,\ time\_step = 0.5]$, computational cost: *1 min, 16 sec.*

tracking objective function is transformed into a problem of geodesic computation. Besides, the associated Euler-Lagrange PDE is solved using the level set formulation [30]. The detection and tracking are viewed as a front propagation problem under the influence of internal and external image dependent forces:

$$\phi_t = \left\{ \gamma \left(g(I_D, \sigma_D)\mathcal{K} + \nabla g(I_D, \sigma_D) \cdot \frac{\nabla \phi}{|\nabla \phi|} \right) + (1-\gamma)\left(g(|\nabla I|, \sigma_T)\mathcal{K} + \nabla g(|\nabla I|, \sigma_T) \cdot \frac{\nabla \phi}{|\nabla \phi|} \right) \right\} |\nabla \phi|. \tag{15}$$

This resulting PDE equation is then solved using techniques borrowed from hyperbolic conservations laws [30] and the approaches presented in Section 2.

3.5 Implementation Issues

The proposed algorithm is self-sufficient and works as follows: Given the first two frames, an arbitrary curve is initialized at the borders of the frame that converges to the motion detection area using the proposed framework (Fig. 6, Fig. 9, and Fig. 10). Following this, the tracking module is applied that moves this curve to the boundaries of the moving objects in both frames (two propagations are performed using the same initial conditions).

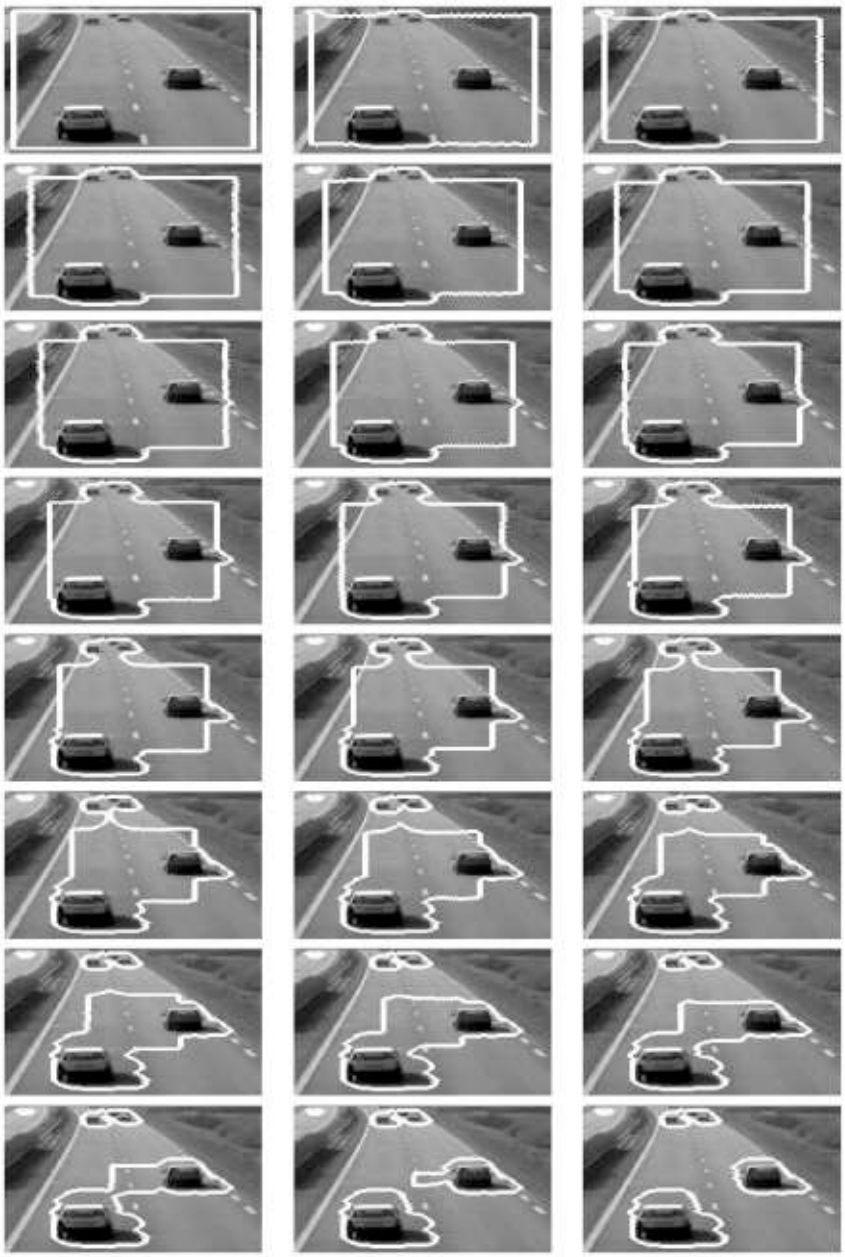

Fig. 10. Motion detection for highway sequence ($\gamma = 1$). Image size: 128×128. Front propagation algorithm: *Fast marching*. The PDE (15) is modified as follows: $[\ \phi_t = g(I_D(s), \sigma_D)\ |\nabla\phi|\ ]$, where the function $g(.)$ is not curvature-based to meet the algorithm requirements. Computational cost: *15 sec*.

Then, the next frame is considered. Due to the fact that the geodesic active contour framework relies on a nonparameterized curve and evolves an initial curve toward one direction (constrained by the curvature effect), it demands a *specific* initialization step. Thus, the initial curve should be either in the interior to the real object boundaries or be exterior to them. Moreover, cases where some parts of the object are within the initial curve while, some other parts are outside, cannot be naturally handled.

Taking this into account, the tracking result of the previous frame is used to initialize the curve in the current frame (Fig. 11a, Fig. 12a, and Fig. 13a). This solution is adopted due to the fact that the curve defined by this result is completely included in the expected motion detection area between the current and the previous frame. Hence, to achieve motion detection, a propagation in an outward direction is considered (Fig. 11b, Fig. 12b, and Fig. 13b). Then, using the final position of the motion detection curve, the tracking module is activated in the current frame and the curve moves towards the real object boundaries in an inwards direction (Fig. 11c, Fig. 12c, Fig. 13c).

4 Discussion, Conclusions, Summary

Before presenting experimental results on real-word video sequences that have been used to validate the proposed approach, the behavior, as well as the efficiency of the proposed front propagation algorithms, is examined.

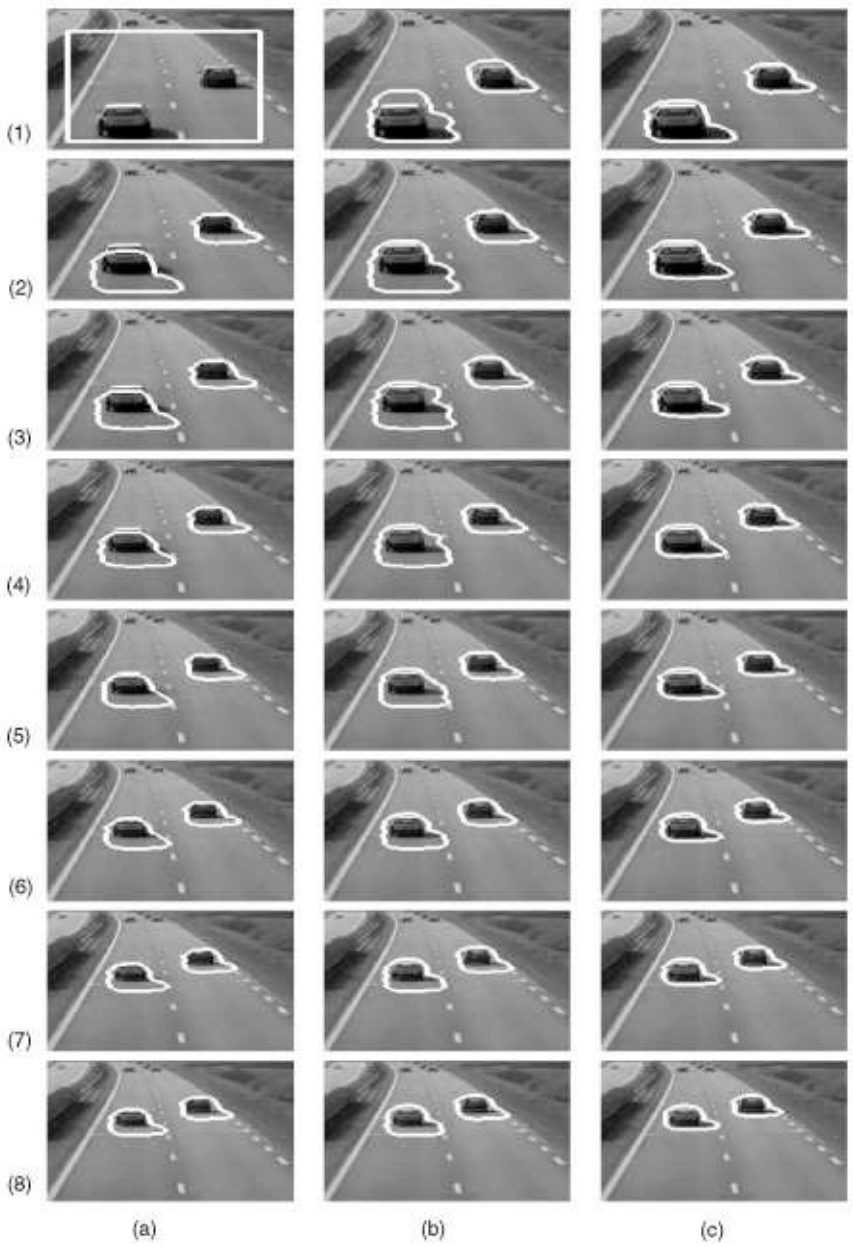

Fig. 11. Motion detection and tracking for highway sequence (1, 2, 3, 4, 5, 6, and 7). Front propagation algorithm: *Hermes*. (a) Initial curve, (b) motion detection result (between the current and the previous frame), and (c) tracking result. This sequence satisfies the constraints imposed by our approach. The background is sufficiently smooth, while the consecutive object positions are overlapped.

4.1 Propagation Algorithms

Two aspects are considered, namely the computational cost and the application field. As it concerns the computational cost, a front propagation equation that can be implemented using the proposed, as well as the existing algorithms, is selected: $[\phi_t = F|\nabla\phi|]$. In order to provide real and reliable tests, function F is chosen to be a Gaussian function that attracts the object boundaries of a given frame (Fig. 7) and is given by:

$$\left[F((x,y)|\sigma) = \frac{1}{\sqrt{2\pi}\sigma e}^{\frac{-|\nabla I(x,y)|^2}{2\sigma^2}}\right].$$

This particular image has been selected since it demands changes of topology and contains objects with important curvature values. The three different algorithms have been applied within this problem framework. The time step was equal to 0.1 for all algorithms, while the band size was 10 pixels for the *Narrow Band* and the window radius for the *Hermes* was equal to 2. Besides, to determine the computational cost with respect to the image size, the same image has been used in different scales (e.g., 100×100, 150×150, 200×200). The computational cost (in terms of execution time[1]) for all algorithms is show in Fig. 8. According to this cost diagram, the *Fast Marching* algorithm is the fastest one,

1. A Sun ULTRA 10, with 128MB memory and a CPU at 299 MHz has been used.

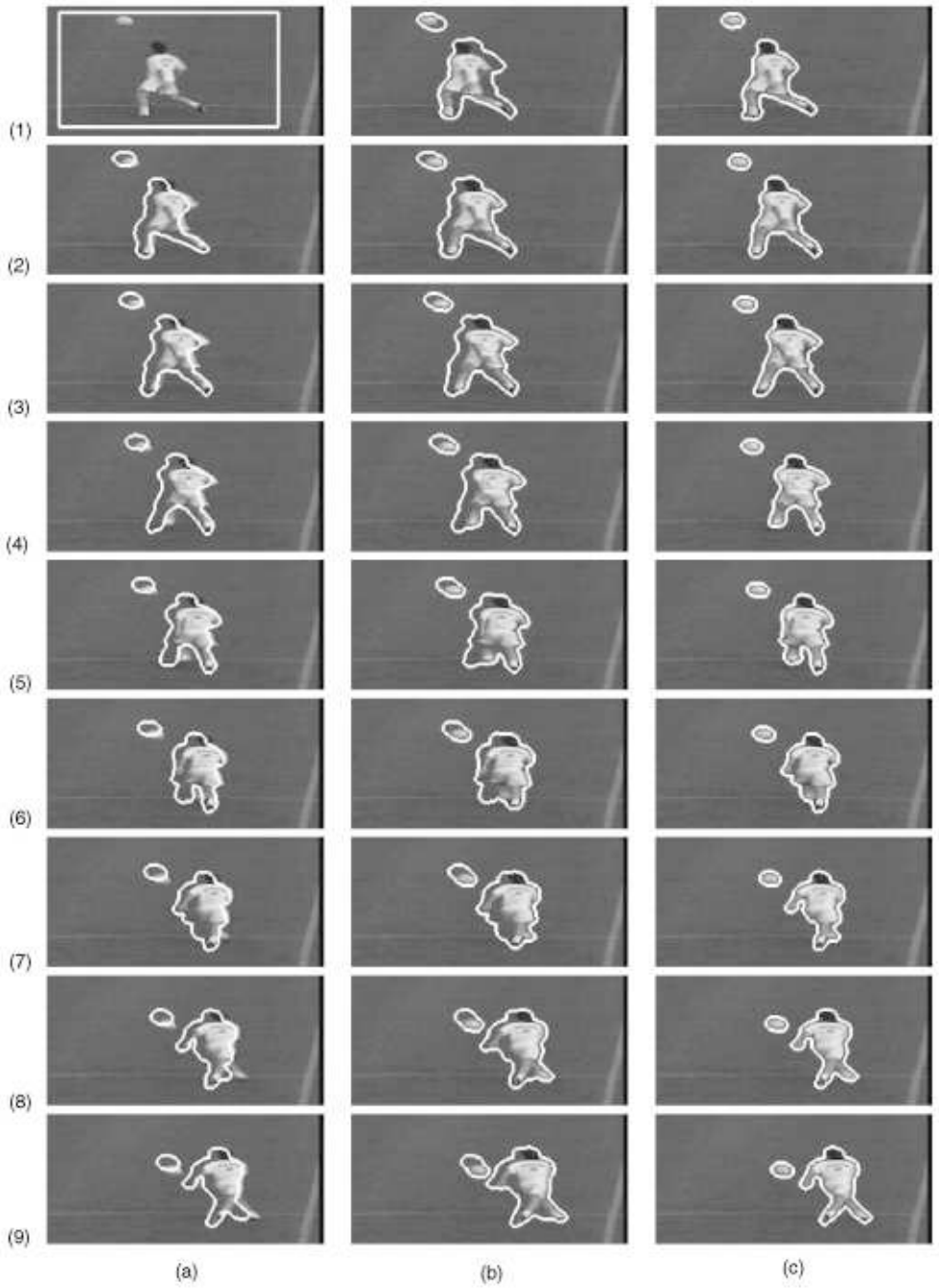

Fig. 12. Motion detection and tracking for player sequence (1, 2, 3, 4, 5, 6, 7, 8, and 9). Front propagation algorithm: *Hermes*. (a) Initial curve, (b) motion detection result (between the current and the previous frame), and (c) Tracking result. This sequence satisfies the constraints imposed by our approach. The background is sufficiently smooth, while the consecutive object positions are overlapped.

while the *Hermes* algorithm seems to have almost the same computational cost with *Fast Marching*.

As it concerns the applicability, the Narrow Band and the Hermes algorithm do not present any any limitations. Hence, these algorithms can be used in cases with **nonconstant sign** speed functions as well as in cases with curvature-dependent speed functions (the case considered in this paper). Contrary to this, the *Fast Marching* is constrained by the assumption of speed functions of constant sign.

4.2 Detection and Tracking

For the detection part, very satisfactory results have been obtained. The boundaries of the moving areas are successfully determined using the proposed *statistical model*. Then, this information is expressed within the framework of a geodesic active contour that deforms an initial curve toward several curves that correspond to the different moving areas (Fig. 6, Fig. 9, and Fig. 10).

The *tracking* part behaves differently with respect to the detection part. In the case where the moving objects are surrounded by a smooth area, or the object motion is

Fig. 13. Motion detection and tracking for Pierre and Imed sequence (1, 2, 3, ,4, and 5). Front propagation algorithm: *Hermes*. (a) Initial curve, (b) motion detection result (between the current and the previous frame), and (c) tracking result. This sequence satisfies the constraints imposed by our approach. The background is smooth, while the consecutive object positions are overlapped. Besides, occlusions are not handled.

small, very satisfactory results have been obtained (Fig. 11, Fig. 12, and Fig. 13). When, on the other hand, there is a textured background close to the objects, the proposed model has some limitations and fails to recover the exact object position.[2] Moreover, if the object displacements are suffiently large and, therefore, there are not any overlapping regions between their two consectutive positions, then both modules fail (detection and tracking). In that case, the motion dection area in the current frame corresponds to the tracking result in the previous frame and the object's position is lost. This can be avoided by introducing a motion estimation step that is going to be used for the curve initialization. Although this idea seems to be promising, it cannot be used within the current framework. The geodesic active contour framework evolves a curve in one direction (positive data-dependent propagation values) constrained by the curvature effect (negative propagation values). Hence, if the initial curve does not completely encircle the object, or if it is not totally surrounded by it, then the complete recovery of the object's area is problematic. This case may arise very often in sequences or images with nonrigid objects due to the fact that the motion estimates are not very accurate

2. This constraint can be easily removed by assuming a background reference frame. In that case the tracking boundaries can be determined using a similar approach with the one used for the motion detection boundaries. Since the case of a stationary camera is considered, the demand of having a background reference frame is reasonable especially for long time applications.

along the object boundaries. However, according to the adopted framework, the initial curve which encircles the tracking result has to be shrunk during the tracking step. Finally, the proposed approach cannot deal with the occlusion case, in which a single curve will be produced by the proposed framework, that corresponds to the complete are covered by the moving objects (Fig. 13).

The main advantage of our appraoch is the ability of dealing with nonrigid objects and movements using a very simple model that does not require important *prior* information. This is because a nonparameterized geodesic active contour model is proposed that does not require a preprocessing step (object parameterization) and can be easily adapted to the current object shape can be obtained, especially in cases with nonrigid objects. Moreover, cases with rigid objects are also appropriately handled (Fig. 11). However, it is certain that in cases of rigid objects, a classic parameterized snake-based approach behaves better and is more robust.

4.3 Summary

To summarize, in this paper, a very simple framework has been proposed for detecting and tracking moving objects in a sequence of images using the front propagation theory and the level-set methodology. According to this framework, both problems are treated simultaneously by propagating a regular curve, first toward the motion detection and, then, toward the tracking boundaries under the influence data-dependent forces. The motion detection boundaries are determined using a probabilistic edge detector that is based on the analysis of the interframe difference using a mixture model, while the tracking boundaries are determined using an edge detector applied to the input image. Then, an objective function is defined that transforms detection and tracking into a geodesic computation problem. This function is minimized using a gradient descent method and the obtained PDE motion equation is implemented using the level set theory. In order to further reduce the computational cost, a multiscale approach is considered which permits moving objects to be tracked with considerable speedup.

Thus, the main contributions of the proposed model are the following:

- A general framework is proposed that links the minimization of a geodesic active contour objective function to the detection and the tracking of moving objects. This framework is implemented using the level set method and can successfully deal with the challenging problem of tracking nonrigid objects that cannot be easily parameterized. Moreover, within this framework, changes of topology of the object being tracked are allowed and scenes in which objects split and merge from frame to frame due to, for example, occlusions, can be tracked. Examples where this happen are traffic control, tracking in sport videos, and human-computer interaction.
- A new front propagation algorithm is proposed that combines the existing ones and can be applied successfully to a wide variety of applications with very low computational cost.

4.4 Future Work

The proposed approach is limited to boundary-based information. A possible extension could be the incorporation of region-based tracking modules [42], [43] to the existing framework that will increase robustness. In that direction, motion detection/segmentation, texture-based [42], and motion-based features [43] (optical flow) to yield the correspondence and free the model from the initial conditions can be considered. Furthermore, the use of geometric-based features ("light" object representations) and multiphase propagation of curves will permit the method to deal with occlusions. A coupled Geodesic Active Contour framework that incorporates different information forms (boundary, region) of different nature (edges, intensities, texture, motion) and can perform tracking under various conditions (static or mobile observer) is the future direction of our work.

Various experimental results (in MPEG format), including the ones shown in this article, can be found at: http://www.inria.fr/robotvis/demo.

Acknowledgments

The authors would like to acknowledge the anonymous reviewers and M. Lourakis for their helpful comments and suggestions which have improved the quality of this paper. Nikos Paragios was affiliated with the Computer Vision and Robotics Group (RobotVis) of INRIA Sophia Antipolis, France. This work was funded in part under the VIRGO research network (EC Contract No. ERBFMRX-CT96-0049) of the TMR Program.

References

[1] N. Diehl, "Object-Oriented Motion Estimation and Segmentation in Image Sequences," *IEEE Trans. Image Processing*, vol. 3, pp. 1,901-1,904, Feb. 1990.
[2] H.H. Nagel, G. Socher, H. Kollnig, and M. Otte, "Motion Boundary Detection in Image Sequences by Local Stochastic Tests," *Proc. European Conf. Computer Vision*, vol. II, pp. 305-315, 1994.
[3] S. Liou and R. Jain, "Motion Detection in Spatio-Temporal Space," *Computer Vision, Graphics, and Image Understanding*, 1989.
[4] T. Aach and A. Kaup, "Bayesian Algorithms for Adaptive Change Detection in Image Sequences Using Markov Random Fields," *Signal Processing: Image Comm*, vol. 7, pp. 147-160, 1995.
[5] J.M. Odobez and P. Bouthemy, "Robust Multiresolution Estimation of Parametric Motion Models," *J. Visual Comm. and Image Representation*, vol. 6, pp. 348-365, 1995.
[6] N. Paragios and G. Tziritas, "Adaptive Detection and Localization of Moving Objects in Image Sequences," *Signal Processing: Image Comm.*, vol. 14, pp. 277-296, 1999.
[7] T. Broida and R. Chellappa, "Estimation of Objects Motion Parameters from Noisy Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, pp. 90-99, Aug. 1986.
[8] F. Meyer and P. Bouthemy, "Region-Based Tracking Using Affine Motion Models in Long Image Sequences," *CVGIP: Image Understanding*, vol. 60, pp. 119-140, 1994.
[9] B. Bascle and R. Deriche, "Region Tracking through Image Sequences," *Proc. IEEE In'l Conf. Computer Vision*, pp. 302-307, 1995.
[10] D. Lowe, "Robust Model Based Motion Tracking through the Integration of Search and Estimation," *In'l J. Computer Vision*, vol. 8, pp. 113-122, 1992.
[11] D. Koller, K. Daniilidis, and H.H. Nagel, "Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes," *Int'l J. Computer Vision*, vol. 10, pp. 257-281, 1993.
[12] J. Wang and E. Adelson, "Representing Moving Images with Layers," *IEEE Trans. Image Processing*, vol. 3, pp. 625-638, 1994.

[13] J. Rehg and T. Kanade, "Model-Based Tracking of Self-Occluding Articulated Objects," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 612-617, 1995.
[14] D. Gavrila and L. Davis, "3-D Model-Based Tracking of Humans in Action: A Multi-View Approach," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 1996.
[15] M. Isard and A. Blake, "Contour Tracking by Stochastic Propagation of Conditional Density," *Proc. European Conf. Computer Vision* vol. I, pp. 343-356, 1996.
[16] F. Leymarie and M. Levine, "Tracking Deformable Objects in the Plane Using an Active Contour Model," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, pp. 617-634, 1993.
[17] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *Int'l J. Computer Vision,* vol. 1, pp. 321-332, 1988.
[18] A. Blake and M. Isard, *Active Contours.* Springer-Verlag, 1997.
[19] D. Cohen, "On Active Contour Models and Balloons," *CVGIP: Image Understanding,* vol. 53, pp. 211-218, 1991.
[20] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic Active Contours," *Proc. IEEE Int'l Conf. Computer Vision,* 1995.
[21] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi, "Gradient Flows and Geometric Active Contour Models," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 810-815, 1995.
[22] R. Malladi, J. Sethian, and B. Vemuri, "Shape Modeling with Front Propagation: A Level Set Approach," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, pp. 158-175, 1995.
[23] D. Koller, J. Weber, and J. Malik, "Robust Multiple Car Tracking with Occlusion Reasoning," *Proc. European Conf. Computer Vision,* vol. I, pp. 189-196, 1994.
[24] V. Caselles and B. Coll, "Snakes in Movement," *SIAM J. Numerical Analysis,* vol. 33, pp. 2,445-2,456, 1996.
[25] M. Black, "Combining Intensity and Motion for Incremental Segmentation and Tracking over Long Image Sequences," *Proc. European Conf. Computer Vision,* pp. 485-493, 1992.
[26] F. Heitz and P. Bouthemy, "Multimodal Estimation of Discontinuous Optical Flow Using Markov Random Fields," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, pp. 1,217-1,232, 1993.
[27] M. Bertalmio, G. Sapiro, and G. Randall, "Morphing Active Contours," *Proc. Int'l Conf. Scale-Space Theories in Computer Vision,* pp. 46-57, 1999.
[28] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky, "Fast Geodesic Active Contours," *Proc. Int'l Conf. Scale-Space Theories in Computer Vision,* pp. 34-45, 1999.
[29] N. Paragios and R. Deriche, "A PDE-Based Level Set Approach for Detection and Tracking of Moving Objects," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 1,139-1,145, 1998.
[30] S. Osher and J. Sethian, "Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on the Hamilton-Jacobi Formulation," *J. Computational Physics,* vol. 79, pp. 12-49, 1988.
[31] J. Sethian, *Level Set Methods.* Cambridge Univ. Press, 1996.
[32] D. Adalsteinsson and J. Sethian, "A Fast Level Set Method for Propagating Interfaces," *J. Computational Physics,* vol. 118, pp. 269-277, 1995.
[33] O. Faugeras and R. Keriven, "Variational Principles, Surface Evolution, PDE's, Level Set Methods and the Stereo Problem," *IEEE Trans. Image Processing,* vol. 7, pp. 336-344, 1998.
[34] D. Terzopoulos, A. Witkin, and M. Kass, "Constraints on Deformable Models: Recovering 3D Shape and Nonrigid Motion," *Artificial Intelligence,* vol. 36, pp. 91-123, 1988.
[35] R. Deriche and O. Faugeras, "Les EDP en Traitement des Images et Vision par Ordinateur," *Traitement du Signal,* vol. 13, 1996. ftp://ftp-robotvis.inria.fr/pub/html/Papers/deriche-faugeras:96b.ps.gz.
[36] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi, "Conformal Curvature Flows: From Phase Transitions to Active Vision," *Archive of Rational Mechanics and Analysis,* vol. 134, pp. 275-301, 1996.
[37] D. Chop, "Computing Minimal Surfaces via Level Set Curvature-Flow," *J. Computational Physics,* vol. 106, pp. 77-91, 1993.
[38] J. Sethian, "A Fast Marching Level Set Method for Monotonically-Advancing Fronts," *Proc. Nat'l Academy of Science,* vol. 93, pp. 1,591-1,694, 1996.
[39] J. Sethian, "A Review of Theory, Algorithms, and Applications of Level Set Methods for Propagating Interfaces," *Acta Numerica.* Cambridge Univ. Press, 1996.
[40] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms.* MIT Press, McGraw-Hill, 1990.
[41] R. Duda and P. Hart, *Pattern Classification and Scene Analysis.* John Wiley & Sons, 1973.
[42] N. Paragios and R. Deriche, "Unifying Boundary and Region-Based Information for Geodesic Active Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. II:300-305, 1999.
[43] N. Paragios and R. Deriche, "Geodesic Active Regions for Motion Estimation and Tracking," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 688-674, 1999.

Nikos Paragios received the BS and MS degrees in computer science (with honors) from the Department of Computer Science at the University of Crete, Greece, in 1994 and 1996, respectively. He is a PhD candidate in computer engineering at the School of Computer Engineering of the University of Nice-Sophia Antipolis. His research interests are in the areas of image processing and computer vision and include image/texture segmentation, motion estimation/analysis/tracking, and applications of computer vision in medical images.

Rachid Deriche graduated from Ecole Nationale Supérieure des Télécommunications, Paris, in 1979, and received the PhD degree in mathematics from the University of Paris XI, Dauphine, in 1982. He is currently a research director at INRIA Sophia-Antipolis in the computer vision and robotics group. His research interests are in computer vision and image processing and include partial differential equations applied to IP and CV, low-level vision, motion analysis and visual tracking, calibration, and stereo, image sequence analysis. More generally, he is very interested in the application of mathematics to computer vision and image processing. He has authored and coauthored more than 100 scientific papers. To find out more about his research and some selected publications, visit at http://www.inria.fr/robotvis/personnel/der/der-eng.html.