

# Region tracking via level set PDEs without motion computation

Abdol-Reza Mansouri

INRS-Télécommunications, Institut National de la Recherche Scientifique  
Place Bonaventure, P.O. Box 644, Montréal, Québec, Canada H5A 1C6  
`mansouri@inrs-telecom.quebec.ca`

## Abstract

Tracking regions in an image sequence is a challenging and difficult problem in image processing and computer vision, and at the same time, one that has many important applications: Automated video surveillance, video database search and retrieval, automated video editing, ... So far, numerous approaches to region tracking have been proposed. Many of them suffer from excessive constraints imposed on the motion of the region being tracked and need an explicit motion model (e.g. affine, euclidean, ...). Some, which do not need a parametrized motion model, rely instead on a dense motion field. By and large, most rely on some kind or other of motion information. Those which do not use any motion information use instead a model of the region being tracked, typically by assuming strong intensity boundaries, or constraining the shape of the region to belong to a parametrized family of shapes. In this paper, we propose a novel approach to region tracking that derives from a Bayesian formulation. The novelty of the approach is twofold: First, no motion field or motion parameters need to be computed. This removes a major burden since accurate motion computation has been and remains a challenging problem and the quality of region tracking algorithms based on motion critically depends on the computed motion fields and parameters. The second novelty of this approach is that very little a priori information about the region being tracked is used in the algorithm. In particular, unlike numerous tracking algorithms, no assumption is made on the strength of the intensity edges of the boundary of the region being tracked, nor is its shape assumed to be of a certain parametric form. The problem of region tracking is formulated as a Bayesian estimation problem, and the resulting tracking algorithm is expressed as a level set partial differential equation. We present further extensions to this partial differential equation, allowing the possibility of including additional information in the tracking process, such as priors on the region's intensity boundaries, and we briefly discuss the issue of numerical implementation. Very promising experimental results are provided using numerous real image sequences with natural object and camera motion.

## I. INTRODUCTION

Tracking of regions in an image sequence is one of the most important, and yet, difficult problems in video processing (object-based video database search, automated image manipulation, video surveillance), video compression (object-based coding such as in MPEG-4), and computer vision (scene analysis and interpretation).

A number of different approaches to region tracking have been developed to date, starting from the early tracking algorithms which concentrated on tracking feature points [1] and edge segments [2], [3]. Since the ultimate goal of a region tracking algorithm is to track whole regions and not only feature points in the region, strong assumptions had to be made both on the shape of the region and on the motion of the points, in order to allow the region to be recovered from its feature points. In the simplest case, the motion field would be assumed to be translational and the shape of the region constant throughout the sequence. This would allow feature point tracking using simple cross-correlation computations between successive images in the sequence. A slightly higher degree of generality would be attained by allowing the motion to be euclidean and using rotational-invariant cross-correlation measures. The notion of tracking a region through its feature points was extended to three dimensions by [4]. There, a three dimensional polyhedral model was assumed, and computed from feature points in the image. Once the three dimensional model was computed, it would be reprojected onto the image plane. Clearly, all these algorithms perform well when the two-dimensional (or three-dimensional) shape of the region can be easily modeled and stays relatively constant over the image sequence. This is not the case in many tracking problems of interest, where the region of interest changes shape throughout the sequence, corresponding to deformable objects (e.g. blood cells [5]). There, feature point based tracking is not appropriate since the shape of the region cannot be related to the position of the feature points in any simple way. To overcome these limitations of region tracking based on simple feature points, region tracking algorithms based on active contours were developed [5], [6], [7], [8]. Here, no assumption is made on the shape of the region (other than overall smoothness); yet, a very strong assumption is made on the boundaries of the region, that is they have to correspond to strong intensity boundaries. In this way, an active contour is shrunk from an initial contour (large enough so as to englobe the region of interest) under a velocity field which locks the contour to intensity edges. In the subsequent frame, this contour is used as initial condition and is locked onto the intensity edges of the region being tracked. To further constrain the motion of the active contour, additional a priori models on the contour displacement have been proposed [9]. Although relatively good results have been obtained, the active contour approach is limited to small displacements and uses only the boundary of the region being tracked, and not the region itself. For this class of algorithms, region tracking

is essentially a byproduct of intensity boundary detection. This is not so with region based tracking algorithms, which use region information and not merely boundary information. In this regard, approaches dual to edge detection have been proposed in [10], aiming at maximizing the intensity homogeneity inside the region to be tracked and inside the background as well. Once region information is used, the motion field can be computed over the region and used to compute the position of the region in the next frame. Furthermore, this motion field can be temporally filtered in order to yield better estimates of the region position. The main drawback of region based techniques is that boundaries are not accurately computed. This imposes a first restriction on the magnitude of motion between subsequent frames: The smaller the motion, the more accurately it is computed. But this is not enough. Indeed, due (usually) to excessive smoothing, motion is least accurately computed around motion boundaries, and these often coincide with boundaries of regions we wish to track. This suggests that motion information alone is not sufficient to guarantee reliable tracking over long sequences. This has led researchers to combine both edge-based and region-based approaches.

A desirable feature of any tracking algorithm, first proposed and demonstrated by [11], is that the tracking algorithm allow changes in the topology of the region being tracked. Indeed, it may be the case that a region of interest split into sub-regions over a number of frames, or that conversely, a number of regions of interest merge into one region. Such changes in topology, which may be difficult to handle when region shape is parametrized, follow instead naturally whenever tracking is formulated as a level set partial differential equation [11], [12], [13].

In [11], tracking is expressed as the solution of a system of coupled partial differential equations: While one image is morphed into another via a particular level set partial differential equation, the boundary of the region being tracked follows the flow obtained from this equation via another level set equation. Since the evolution equation for morphing is obtained by gradient descent on an  $L^2$  norm of image differences, this tracking algorithm can be applied only when the displacements involved are small, typically on the order of a few pixels. In [12], tracking is expressed as minimum description length estimation using a geometric description length criterion, and knowledge of a dense or parametrized motion field is assumed. The resulting tracking performance is critically dependent on the quality of the given motion field or parameters, as shown in the results. The problem addressed in [13] is that of detection and tracking of moving objects in image sequences. In the proposed algorithm, a detection step forces a closed curve to converge towards moving areas of an image, while a tracking step evolves the curve even further, until it coincides with the exact boundary of the moving object. The detection step is based on temporal change detection, while the tracking step is actually nothing other than a classical intensity boundary detection algorithm using active contours and implemented using

level sets. Thus, the tracking algorithm proposed in [13] is not a generic tracking algorithm, and since it is merely a byproduct of simple intensity boundary detection, it suffers from very stringent constraints; in particular, the region to be tracked is assumed to have strong intensity boundaries, and the background is assumed to be smooth. Also, since the tracking step relies heavily on the preceding moving region detection step, the region being tracked is assumed to be moving, while the background is assumed to be stationary.

In this paper, we consider the *generic* problem of tracking regions in image sequences, be they moving or stationary, and we attempt to do so using only the least number of explicit constraints on the region being tracked. In particular, other than overall smoothness, we do not assume the region's shape to belong to a parametrized family, nor do we relate it to a three-dimensional model, nor do we assume that the region undergoes rigid motion. Also, in stark contrast to [13], we do not make the simplifying assumption that its boundary coincides with strong intensity boundaries, nor do we assume that the region to be tracked is moving on a fixed and uniform background. Furthermore, unlike numerous motion-based approaches (e.g. [12]) , we do not compute any motion field or motion parameters or feature points for motion computation, and contrary to [11], we do not restrict tracking to small displacements. To be sure, we will have to make some assumptions on the way to our proposed tracking algorithm; however, these assumptions will be expressed probabilistically and will be made on particular probability distributions. As a result, these assumptions will constrain the tracking problem only implicitly, as will be discussed in the paper. We believe such implicit probabilistic assumptions are more suited to the tracking problem than the explicit assumptions commonly used; this is demonstrated in the experimental results, where it is shown that our proposed algorithm yields precise tracking over numerous frames for real image sequences with substantially different motion, shape, and intensity boundary structure.

The starting point of our algorithm is a basic observation model that expresses the fact that the region's luminance/chrominance statistics vary little from frame to frame in the sequence. Clearly, without such an assumption, tracking would not make much sense. Although we do not assume any particular underlying motion, we assume the maximum range of motion in between image frames is given. This does not reduce the generality of the proposed algorithm and is used only to reduce its computational complexity. We pose region tracking as a bayesian estimation problem, and at each step on our way to solving this estimation problem, we use the minimum assumptions necessary to make the problem tractable. This bayesian estimation problem is ultimately formulated as an energy minimization problem, leading to the solution via Euler-Lagrange descent equations. These Euler-Lagrange equations are themselves formulated as level set partial differential equations, leading to a stable numerical implementation,

in addition to allowing the proposed tracking algorithm to retain the desirable feature of being topology independent, and allowing regions being tracked to freely split and merge. The tracking algorithm we propose is thus a particular level set evolution equation, and as will be seen in the experimental results, the tracking obtained is remarkably accurate despite the variety of test images used and despite the small number - two- of parameters to set, one of which is the same for all test runs. Although our proposed tracking algorithm makes no use of intensity boundary, shape, or motion information, we indicate precisely in the paper where and how such information, whenever available, could be incorporated in the tracking algorithm.

The paper is organized as follows. In Section II a formulation of region tracking as bayesian estimation is proposed. In Section III, the solution of this bayesian estimation problem in terms of Euler-Lagrange descent equations is provided. Then, in Section III-C the equivalent formulation of these equations in terms of level set partial differential equations is given, and the numerical solution of these evolution equations is very briefly discussed. The estimation of the probability distribution functions which form the cornerstone of this algorithm is discussed in Section IV, together with basic extensions. In Section V, our proposed level set evolution equation for tracking is given, together with numerous extensions. The paper ends with experimental results in Section VI followed by conclusions.

## II. REGION TRACKING AS BAYESIAN ESTIMATION

Let  $(\mathbf{I}^k)_k$  be a sequence of images (indexed by nonnegative integers with integer  $k$  representing time instant  $k$ ) with common domain  $\Omega$  (an open subset of  $\mathcal{R}^2$ ). Let  $\mathbf{R}_0 \subset \Omega$  be a region in the image at time  $n$  ( $\mathbf{I}^n$ ) and let  $\mathbf{R}_1 \subset \Omega$  be the corresponding (unknown) region in the image at time  $n+1$  ( $\mathbf{I}^{n+1}$ ), that we seek to estimate. The problem of tracking  $\mathbf{R}_0$  from time instant  $n$  to time instant  $n+1$  can thus be formulated as the problem of *estimating*  $\mathbf{R}_1$  given  $\mathbf{I}^n, \mathbf{I}^{n+1}$ , and  $\mathbf{R}_0$ . In order to perform this estimation, we need to define an observation model. For simplicity, we shall assume that all our images are scalar-valued. Extensions to vector-valued images will be indicated where appropriate and are, for the most part, immediate. Once this observation model is defined, probability estimates can be computed and optimized.

### A. Observation model

We consider a given (finite or infinite) set  $\Gamma$  of diffeomorphisms  $\psi : \Omega \rightarrow \Omega$ .  $\Gamma$  is the set of allowed geometrical transformations. We assume there exists a mapping  $\phi \in \Gamma$  such that  $\phi(\mathbf{R}_0) = \mathbf{R}_1$  (and hence  $\phi(\mathbf{R}_0^c) = \mathbf{R}_1^c$ ), and such that

$$\mathbf{I}^{n+1} \circ \phi(\mathbf{x}) = \mathbf{I}^n(\mathbf{x}) + \mu(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega, \quad (1)$$

where  $\mu$  denotes a stationary zero-mean Gaussian white noise process with variance (covariance matrix, in the vector case)  $\sigma^2$ . This is a formulation of the constraint that the image function  $\mathbf{I}^{n+1}$  defined over  $\mathbf{R}_1$  be in agreement with the image function  $\mathbf{I}^n$  defined over  $\mathbf{R}_0$ , as is implicit in most tracking algorithms. Note that this model does not take into account occlusions or uncovered regions. To take these into account, we would either have to model them explicitly or factor them in the noise process, which would then be neither stationary nor white. Nevertheless, the observation model as defined above is a reasonable working model. It is important to note that motion based tracking algorithms directly attempt to estimate  $\phi$  among all transformations in  $\Gamma$ , and then estimate  $\mathbf{R}_1$  as being given by  $\phi(\mathbf{R}_0)$ . To make the estimation of  $\phi$  tractable,  $\Gamma$  is then constrained in various ways: For example, by assuming the transformations in  $\Gamma$  are close to the identity transformation, thereby leading to the use of differential techniques, or by assuming  $\Gamma$  is a small-dimensional space described by only a few parameters, leading to least-squares techniques for estimating these parameters.

In this paper, we propose a different approach to tracking, and rather than estimating  $\phi$ , we shall attempt to estimate  $\mathbf{R}_1$ . The main benefit of this approach is that no specific assumptions on  $\Gamma$  need to be made, and the resulting tracking algorithm will not be constrained to particular types of motion.

### B. MAP formulation

The Maximum A Posteriori estimate  $\hat{\mathbf{R}}_1$  of  $\mathbf{R}_1$  given  $\mathbf{I}^n$ ,  $\mathbf{I}^{n+1}$ , and  $\mathbf{R}_0$  is found by maximizing the a posteriori probability  $P(\mathbf{R}_1 = \mathbf{R} | \mathbf{I}^n, \mathbf{I}^{n+1}, \mathbf{R}_0)$  over subsets  $\mathbf{R}$  of  $\Omega$ . That is,

$$\hat{\mathbf{R}}_1 = \arg \max_{\mathbf{R} \subset \Omega} P(\mathbf{R}_1 = \mathbf{R} | \mathbf{I}^n, \mathbf{I}^{n+1}, \mathbf{R}_0). \quad (2)$$

Using Bayes' rule, we can thus write

$$\hat{\mathbf{R}}_1 = \arg \max_{\mathbf{R} \subset \Omega} P(\mathbf{I}^{n+1} | \mathbf{I}^n, \mathbf{R}_0, \mathbf{R}_1 = \mathbf{R}) P(\mathbf{R}_1 = \mathbf{R} | \mathbf{I}^n, \mathbf{R}_0).$$

The first expression on the right hand side above is the likelihood of observing image  $\mathbf{I}^{n+1}$  given image  $\mathbf{I}^n$ , the region  $\mathbf{R}_0$  in the image at time  $n$ , and the corresponding region  $\mathbf{R}_1$  in the image at time  $n + 1$ . This likelihood function summarizes part of our assumptions about the behavior of the region to be tracked, since it allows us to find the maximum likelihood estimate of  $\mathbf{I}^{n+1}$  from the knowledge of  $\mathbf{I}^n$ ,  $\mathbf{R}_0$ , and  $\mathbf{R}_1$ . This likelihood term will embody our assumptions pertaining to the constancy (with respect to the sequence index  $n$ ) of the luminance or the chrominance of the region being tracked. The second term on the right-hand-side above is the prior term on  $\mathbf{R}_1$ . Note that unlike usual priors, it is a conditional probability, with conditioning on  $\mathbf{I}^n$  and region  $\mathbf{R}_0$ . This probability term embodies our geometrical assumptions on how the region  $\mathbf{R}_0$  should evolve from image  $\mathbf{I}^n$  to image  $\mathbf{I}^{n+1}$  of the sequence. For example, the assumption

that  $\mathbf{R}_1$  should differ from  $\mathbf{R}_0$  by an euclidean transformation would affect only this probability term. If we allow the deformation of a region throughout the sequence to be independent of its luminance or chrominance characteristics, then this term reduces to  $P(\mathbf{R}_1 = \mathbf{R}|\mathbf{R}_0)$ . If, in addition, we allow the shape of the region being tracked from  $\mathbf{I}^n$  to  $\mathbf{I}^{n+1}$  to be independent of its shape in  $\mathbf{I}^n$ , then this last term further reduces to  $P(\mathbf{R}_1 = \mathbf{R})$ . This yields the highest degree of generality for the tracking algorithm in that the shape of the region being tracked can freely change from one frame to the next. Although such a degree of generality is not always warranted, we will allow it precisely in order not to constrain the deformations of the region being tracked to belong to a narrow specific class.

To reduce the likelihood term  $P(\mathbf{I}^{n+1}|\mathbf{I}^n, \mathbf{R}_0, \mathbf{R}_1 = \mathbf{R})$  to a more tractable expression, we make two fundamental assumptions:

- *Conditional independence:* We assume that  $\forall \mathbf{x}, \mathbf{y} \in \Omega, \mathbf{x} \neq \mathbf{y}$ , the conditional probabilities of  $\mathbf{I}^{n+1}(\mathbf{x})$  and  $\mathbf{I}^{n+1}(\mathbf{y})$  are independent given  $\mathbf{I}^n, \mathbf{R}_0, \mathbf{R}_1$ . That is,

$$P(\mathbf{I}^{n+1}(\mathbf{x}), \mathbf{I}^{n+1}(\mathbf{y})|\mathbf{I}^n, \mathbf{R}_0, \mathbf{R}_1 = \mathbf{R}) = P(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0, \mathbf{R}_1 = \mathbf{R})P(\mathbf{I}^{n+1}(\mathbf{y})|\mathbf{I}^n, \mathbf{R}_0, \mathbf{R}_1 = \mathbf{R}).$$

- *Partial dependence:* We assume that for any  $\mathbf{x} \in \Omega$ , the conditional probability of  $\mathbf{I}^{n+1}(\mathbf{x})$  given  $\mathbf{I}^n, \mathbf{R}_0, \mathbf{R}_1$  is a function only of  $\mathbf{I}^n, \mathbf{R}_0$  and the membership (or lack of membership) of  $\mathbf{x}$  in  $\mathbf{R}_1$ . In other words, we write

$$P(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0, \mathbf{R}_1 = \mathbf{R}) = P(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0, \chi_{\mathbf{R}}(\mathbf{x})),$$

where  $\chi_{\mathbf{R}}$  is the indicator function of the set  $\mathbf{R}$ .

The conditional independence assumption would follow directly from the observational model (1) if we could determine  $\phi$  uniquely from the knowledge of  $\mathbf{R}_0$  and  $\mathbf{R}_1$ . This happens whenever the class of allowed diffeomorphisms  $\Gamma$  is small enough that there exists only one diffeomorphism  $\phi$  in  $\Gamma$  for which  $\phi(\mathbf{R}_0) = \mathbf{R}_1$ . Thus, the conditional independence assumption is essentially an assumption about the set of allowed geometrical transformations  $\Gamma$ , akin to constraints on motion which are imposed in numerous tracking algorithms (e.g. translational motion, euclidean, affine). We believe however that constraining  $\Gamma$  indirectly through this probabilistic condition is better suited to tracking and provides a much more general tracking algorithm, as opposed to constraining  $\Gamma$  to be a particular set of geometric transformations. Furthermore, although the conditional independence assumption is false in general, it is a reasonable assumption and will provide us with a tractable model. The partial dependence assumption, on the other hand, stipulates that the dependence of a point on  $\mathbf{R}_1$  (given  $\mathbf{I}^n$  and  $\mathbf{R}_0$ ) can be factored through the dependence only on its membership in  $\mathbf{R}_1$ . Such an assumption would hold if the image function  $\mathbf{I}^n$  were to have a constant value on  $\mathbf{R}_0$  and a constant value on  $\mathbf{R}_0^c$ . More generally, it

would also hold if  $\mathbf{I}^n(\mathbf{R}_0) \cap \mathbf{I}^{n+1}(\mathbf{R}_0^c)$  were empty. Note that this assumption is analogous to the assumption of strong intensity boundaries which is imposed in active contour based tracking algorithms. Once again, we believe that formulating the constraint that the region being tracked and the background have distinct luminance/chrominance characteristics probabilistically through the partial dependence assumption is better suited to tracking than assuming strong intensity boundaries. Indeed, in the latter case, the tracking algorithm is easily derailed by spurious intensity edges which are not on the region's boundary. The partial dependence assumption is also a simplifying assumption, since an object's luminance/chrominance characteristics sometimes overlap with those of the background. Yet, locally (e.g. in a small neighborhood of a boundary point of the region), these characteristics are usually distinct.

The partial dependence assumption allows us to define the probability functions

$$P(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0, \mathbf{R}_1 = \mathbf{R}) = \begin{cases} P_{in,\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0), & \text{if } \mathbf{x} \in \mathbf{R}, \\ P_{out,\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0), & \text{if } \mathbf{x} \in \mathbf{R}^c, \end{cases}$$

which are the conditional probabilities that a point in  $\Omega$  be inside or outside of the region being tracked. Here, we have indexed the probability functions by  $\mathbf{x}$  to highlight their dependence on the position variable.

Together, the conditional independence and partial dependence assumptions above allow us to write

$$\hat{\mathbf{R}}_1 = \arg \max_{\mathbf{R} \subset \Omega} \left\{ \left( \prod_{\mathbf{x} \in \mathbf{R}} P_{in,\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0) \right) \left( \prod_{\mathbf{x} \in \mathbf{R}^c} P_{out,\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0) \right) P(\mathbf{R}_1 = \mathbf{R}|\mathbf{I}^n, \mathbf{R}_0) \right\},$$

where  $\mathbf{R}^c$  denotes the complement in  $\Omega$  of the subset  $\mathbf{R}$  of  $\Omega$ . Since one of our objectives in this paper is to illustrate how reliable region tracking can be performed without reliance on intensity boundaries, we shall define a prior on  $\mathbf{R}_1$  that does not make any use of intensity boundaries of  $\mathbf{I}^{n+1}$ . In the next section, we shall indicate how to include priors on  $\mathbf{R}_1$  depending on intensity boundaries. Furthermore, since we also wish to define a generic tracking algorithm that does not assume a particular two-dimensional or three-dimensional model of the region being tracked, we shall define a simple prior on  $\mathbf{R}_1$  that only enforces smoothness of the boundary  $\partial\mathbf{R}_1$  of  $\mathbf{R}_1$ . This will allow our tracking algorithm to retain a substantial degree of generality. It is nevertheless possible to tailor our proposed algorithm to the tracking of specific shapes. In this case, the smoothness prior we shall adopt for the sake of generality and simplicity should be replaced with a prior specific to the shapes being tracked. Such priors have been proposed in [18], and we refer the reader thereto for a detailed discussion of their computation.

Dropping normalizing additive constants, we shall then define  $P(\mathbf{R}_1 = \mathbf{R}|\mathbf{I}^n, \mathbf{R}_0)$  through its



negative log-likelihood:

$$-\log P(\mathbf{R}_1 = \mathbf{R} | \mathbf{I}^n, \mathbf{R}_0) = \lambda_L \oint_{\partial \mathbf{R}} d\rho,$$

which is simply, up to the positive constant  $\lambda_L$ , the length of  $\partial \mathbf{R}$ ,  $d\rho$  being the differential of arc length. We can thus write

$$\hat{\mathbf{R}}_1 = \arg \min_{R \subset \Omega} E(\mathbf{R} | \mathbf{I}^n, \mathbf{I}^{n+1}, \mathbf{R}_0),$$

where

$$\begin{aligned} E(\mathbf{R} | \mathbf{I}^n, \mathbf{I}^{n+1}, \mathbf{R}_0) &= - \int_{\mathbf{R}} \log P_{in, \mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x}) | \mathbf{I}^n, \mathbf{R}_0) d\mathbf{x} \\ &\quad - \int_{\mathbf{R}^c} \log P_{out, \mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x}) | \mathbf{I}^n, \mathbf{R}_0) d\mathbf{x} \\ &\quad + \lambda_L \oint_{\partial \mathbf{R}} d\rho. \end{aligned}$$

This functional, which we call the tracking functional, is the one we shall try to minimize with respect to  $\mathbf{R}$ . Obviously, this functional depends on the probability functions  $P_{in}$  and  $P_{out}$  and these need to be somehow estimated. Before discussing the estimation of these probabilities, however, we shall detail the solution of the tracking functional.

### III. MINIMIZATION OF THE TRACKING FUNCTIONAL

#### A. Tracking independent of intensity boundaries

Let  $\vec{\gamma} : [0, 1] \rightarrow \mathcal{R}^2, s \mapsto \vec{\gamma}(s)$ , be a closed planar curve, oriented counterclockwise, that we use as an estimator of  $\partial \mathbf{R}_1$  (note that the arc parameter  $s$  is not necessarily arc length). Our goal is to find the curve  $\vec{\gamma}^*$  that minimizes the tracking functional

$$\begin{aligned} E(\vec{\gamma} | \mathbf{I}^n, \mathbf{I}^{n+1}, \mathbf{R}_0) &= - \int_{\mathbf{R}_{\vec{\gamma}}} \log P_{in, \mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x}) | \mathbf{I}^n, \mathbf{R}_0) d\mathbf{x} \\ &\quad - \int_{\mathbf{R}_{\vec{\gamma}}^c} \log P_{out, \mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x}) | \mathbf{I}^n, \mathbf{R}_0) d\mathbf{x} \\ &\quad + \lambda_L \oint_{\vec{\gamma}} d\rho, \end{aligned} \tag{3}$$

where  $\mathbf{R}_{\vec{\gamma}}$ , the region enclosed by  $\vec{\gamma}$ , is the estimator of the region  $\mathbf{R}_1$ , while its complement  $\mathbf{R}_{\vec{\gamma}}^c$  is the estimator of the background  $\mathbf{R}_1^c$ . In the last integral above, expressing the prior on the length of  $\vec{\gamma}$ , the curve  $\vec{\gamma}$  has been parametrized by the arc parameter  $s \in [0, 1]$  (note that  $s$  is not necessarily arc length). This functional is identical to the region competition functional proposed by [14] for image segmentation, and the minimization of this functional corresponds to finding the shortest closed curve  $\vec{\gamma}$  that best separates the region being tracked from the background.

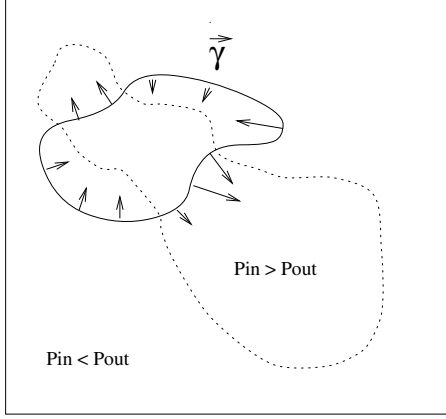


Fig. 1. Curve evolution for tracking.

To find the curve  $\vec{\gamma}^*$  that minimizes  $E(\vec{\gamma}|\mathbf{I}^n, \mathbf{I}^{n+1}, \mathbf{R}_0)$ , we embed  $\vec{\gamma}$  in a one-parameter family  $(\vec{\gamma}(\cdot, t))_{t \geq 0}$  of closed planar curves, such that as  $t \rightarrow \infty$ ,  $\vec{\gamma}(\cdot, t)$  converges to a local minimum of  $E$ . Such a one-parameter family is obtained from the Euler-Lagrange descent equation associated to the minimization of (3), and, omitting  $t$  for simplicity of notation, is given by [14]:

$$\frac{d\vec{\gamma}(s)}{dt} = [\log P_{in, \vec{\gamma}(s)}(\mathbf{I}^{n+1}(\vec{\gamma}(s))|\mathbf{I}^n, \mathbf{R}_0) - \log P_{out, \vec{\gamma}(s)}(\mathbf{I}^{n+1}(\vec{\gamma}(s))|\mathbf{I}^n, \mathbf{R}_0) - \lambda_L \kappa_\gamma(s)] \vec{n}(s), \quad (4)$$

where  $\vec{n}(s)$  is the unit normal to  $\vec{\gamma}$  at  $s$  pointing outward of  $\mathbf{R}_{\vec{\gamma}}$ , and  $\kappa_\gamma$  is the curvature function of  $\vec{\gamma}$  defined as follows:

$$\kappa_\gamma = \frac{\dot{x}\ddot{y} - \ddot{x}\dot{y}}{(\dot{x}^2 + \dot{y}^2)^{3/2}},$$

where  $(\dot{x}, \dot{y}, \ddot{x}, \ddot{y})$  are the first- and second-order derivatives, respectively, of  $\vec{\gamma} = (x, y)$  with respect to  $s$ . The evolution of  $\vec{\gamma}$  can be easily read off the above equation (see Fig. 1): Assuming the curvature to be zero, a boundary point  $\mathbf{x} = \vec{\gamma}(s)$  with higher likelihood of belonging to  $\mathbf{R}_1$  than to  $\mathbf{R}_1^c$  ( $P_{in, \mathbf{x}} > P_{out, \mathbf{x}}$ ) will induce a velocity of  $\vec{\gamma}(s)$  in the outward direction and the curve  $\vec{\gamma}$  will grow so as to englobe the point  $\mathbf{x}$ . Similarly, if  $\mathbf{x} = \vec{\gamma}(s)$  is a point with higher likelihood of belonging to  $\mathbf{R}_1^c$  than to  $\mathbf{R}_1$  ( $P_{out, \mathbf{x}} > P_{in, \mathbf{x}}$ ), then  $\vec{\gamma}$  will move inward at  $\vec{\gamma}(s)$  and will shrink, relinquishing the point  $\mathbf{x}$ . The curvature term, on the other hand, will tend to locally straighten out the curve; convexities ( $\kappa_\gamma > 0$ ) will be pulled in, while concavities ( $\kappa_\gamma < 0$ ) will be pushed out.

### B. Tracking accounting for intensity boundaries

We now indicate a simple extension to the curve evolution equation (4) that allows the incorporation of intensity boundary information during tracking. Though our main objective is to demonstrate reliable tracking without reliance on such constraints, it is important to note

that such additional constraints can be easily incorporated in the proposed formulation. Indeed, in some tracking applications, there may be a very strong correlation between the boundary of the region being tracked and image intensity boundaries, and such a correlation may need to be exploited. In other words, knowledge of the intensity boundaries of  $\mathbf{I}^{n+1}$  may help in the estimation of  $\mathbf{R}_1$ . We thus assume a dependence between  $\mathbf{R}_1$  and the intensity boundaries of  $\mathbf{I}^{n+1}$  which we assume, without loss of generality, can be obtained from  $\vec{\nabla}\mathbf{I}^{n+1}$  (note that any other differential operator may be used for that matter). Returning to equation (2), we may write

$$\begin{aligned}\hat{\mathbf{R}}_1 &= \arg \max_{\mathbf{R} \subset \Omega} P(\mathbf{R}_1 = \mathbf{R} | \mathbf{I}^n, \mathbf{I}^{n+1}, \mathbf{R}_0) \\ &= \arg \max_{\mathbf{R} \subset \Omega} P(\mathbf{R}_1 = \mathbf{R} | \mathbf{I}^n, \mathbf{I}^{n+1}, \vec{\nabla}\mathbf{I}^{n+1}, \mathbf{R}_0),\end{aligned}$$

since  $\vec{\nabla}\mathbf{I}^{n+1}$  is a function of  $\mathbf{I}^{n+1}$ . Using Bayes' rule, we can write

$$\hat{\mathbf{R}}_1 = \arg \max_{\mathbf{R} \subset \Omega} P(\mathbf{I}^{n+1} | \mathbf{I}^n, \vec{\nabla}\mathbf{I}^{n+1}, \mathbf{R}_0, \mathbf{R}_1 = \mathbf{R}) P(\mathbf{R}_1 = \mathbf{R} | \mathbf{I}^n, \vec{\nabla}\mathbf{I}^{n+1}, \mathbf{R}_0).$$

Assuming that  $P(\mathbf{I}^{n+1}(\mathbf{x}) | \mathbf{I}^n, \vec{\nabla}\mathbf{I}^{n+1}, \mathbf{R}_0, \mathbf{R}_1 = \mathbf{R}) = P(\mathbf{I}^{n+1}(\mathbf{x}) | \mathbf{I}^n, \mathbf{R}_0, \mathbf{R}_1 = \mathbf{R})$  for all  $\mathbf{x} \in \Omega$ , everything proceeds as in section II-B, except that the prior probability on  $\mathbf{R}_1$  is now of the form  $P(\mathbf{R}_1 = \mathbf{R} | \mathbf{I}^n, \vec{\nabla}\mathbf{I}^{n+1}, \mathbf{R}_0)$ . We can now impose on  $\mathbf{R}_1$  a prior that not only enforces smoothness of  $\partial\mathbf{R}_1$ , as we did in the previous section, but also one that encourages  $\partial\mathbf{R}_1$  to pass through points in  $\Omega$  where  $\vec{\nabla}\mathbf{I}^{n+1}$  has a large magnitude. Thus, a possible prior on  $\mathbf{R}_1$  could now be given by

$$-\log P(\mathbf{R}_1 = \mathbf{R} | \mathbf{I}^n, \vec{\nabla}\mathbf{I}^{n+1}, \mathbf{R}_0) = \lambda_L \oint_{\partial\mathbf{R}} d\rho - \lambda_I \oint_{\partial\mathbf{R}} \|\vec{\nabla}\mathbf{I}^{n+1}\| d\rho,$$

where the last term favors the alignment of  $\partial\mathbf{R}$  with intensity boundaries of  $\mathbf{I}^{n+1}$ , depending on the magnitude of the positive constant  $\lambda_I$ . In this way, the prior model for  $\partial\mathbf{R}_1$  seeks the shortest closed curve  $\vec{\gamma} : [0, 1] \rightarrow \mathcal{R}^2$  that aligns itself with image intensity boundaries, yielding the tracking functional

$$\begin{aligned}E(\vec{\gamma} | \mathbf{I}^n, \mathbf{I}^{n+1}, \mathbf{R}_0) &= - \int_{\mathbf{R}_{\vec{\gamma}}} \log P_{in, \mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x}) | \mathbf{I}^n, \mathbf{R}_0) d\mathbf{x} \\ &\quad - \int_{\mathbf{R}_{\vec{\gamma}}^c} \log P_{out, \mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x}) | \mathbf{I}^n, \mathbf{R}_0) d\mathbf{x} \\ &\quad + \lambda_L \int_{[0, 1]} \|\dot{\vec{\gamma}}\| ds \\ &\quad - \lambda_I \int_{[0, 1]} \|\vec{\nabla}\mathbf{I}^{n+1}(\vec{\gamma}(s))\| \|\dot{\vec{\gamma}}\| ds.\end{aligned}\tag{5}$$

The complete curve evolution equation leading to a minimum of the functional (5) is given by:

$$\begin{aligned} \frac{d\vec{\gamma}(s)}{dt} = & [\log P_{in,\vec{\gamma}(s)}(\mathbf{I}^{n+1}(\vec{\gamma}(s))|\mathbf{I}^n, \mathbf{R}_0) - \log P_{out,\vec{\gamma}(s)}(\mathbf{I}^{n+1}(\vec{\gamma}(s))|\mathbf{I}^n, \mathbf{R}_0) - \lambda_L \kappa_\gamma(s)] \vec{n}(s) \\ & + \lambda_I \left[ (\mathbf{H}_I \frac{\vec{\nabla} \mathbf{I}^{n+1}}{\|\vec{\nabla} \mathbf{I}^{n+1}\|}) \cdot \vec{n}(s) + \|\vec{\nabla} \mathbf{I}^{n+1}\| \kappa_\gamma(s) \right] \vec{n}(s), \end{aligned} \quad (7)$$

where  $\mathbf{H}_I$  is the Hessian of the image function  $\mathbf{I}^{n+1}$  and “.” denotes the euclidean scalar product. The tangential evolution terms have been discarded from the above equation since flows along the tangent to a curve change only its parametrization and not its geometry. Here again, further priors on  $\vec{\gamma}$  can be included to constrain the shape of the region being tracked, as was discussed in the previous section.

We now discuss the level set representation of the curve evolution equations obtained in the last two sections.

### C. Level Set Representation

Equations (4) and (6) can be solved numerically by discretizing the interval  $[0, 1]$  on which  $\vec{\gamma}$  is defined, thus leading to a representation of  $\vec{\gamma}$  in terms of a finite number of points or nodes. This leads to an *explicit* representation of  $\vec{\gamma}$ . A better alternative is to represent the curve  $\vec{\gamma}$  *implicitly* by the zero level set of a function  $u : \mathcal{R}^2 \rightarrow \mathcal{R}$ , with the region inside  $\vec{\gamma}$  corresponding to  $u > 0$ . There are major advantages to such an implicit representation [15], the most important being numerical stability and topology independence. Indeed, the discretization of a curve using a fixed number of nodes can lead to large approximation errors since the order of discretization error is highly dependent on the evolution of the curve itself. Such discretization problems do not occur with level set representations of curves since each level set function is defined over a fixed discrete grid with uniform spacing. Furthermore, as is well known by the Jordan curve theorem, a closed simple (i.e., non self-intersecting) curve divides the plane into two connected components, one bounded and the other unbounded. A contour evolution equation does not allow for curve splitting and curve merging. Thus, during curve evolution, there will always be one “inside” and one “outside”. On the other hand, curves implicitly defined using level sets can split and merge, freely changing topology during their evolution (Fig. 2).

Since  $\vec{\gamma}$  obeys an evolution equation and the zero level set of  $u$  is assumed to coincide with (the image of)  $\vec{\gamma}$ ,  $u$  must evolve according to a certain evolution equation closely related to that of  $\vec{\gamma}$ . We can thus embed  $u$  in a one-parameter family  $(u(\cdot, t))_{t \geq 0}$ , and we construct the evolution equation that  $u$  has to satisfy so that its zero level set satisfy the evolution equation of  $\vec{\gamma}$ . To achieve such correspondence,  $u$  must satisfy two conditions:

1. The curve  $\vec{\gamma}$  must be a zero level set (iso-level contour) of  $u$ , which leads to:

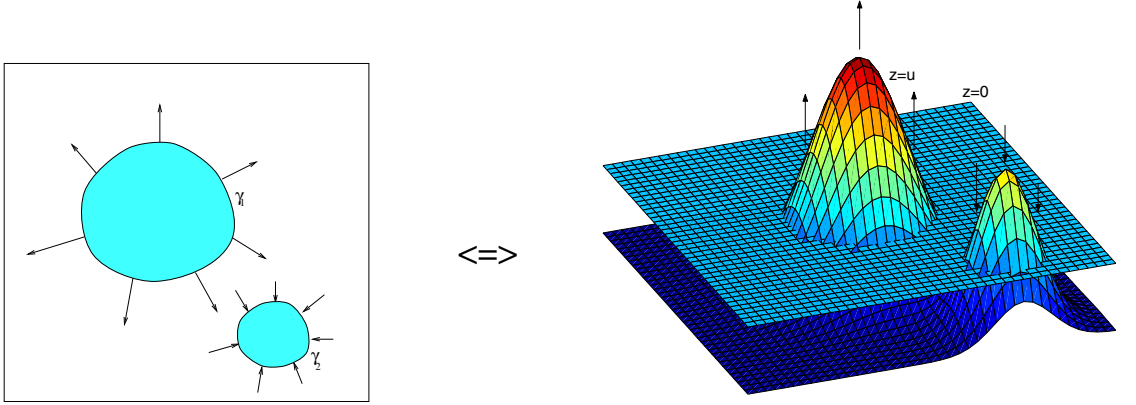


Fig. 2. Schematic representation of the equivalence between the evolution of curves  $\vec{\gamma}_1$ ,  $\vec{\gamma}_2$  and the evolution of function  $u$ .

$$\frac{du(\vec{\gamma}(s, t), t)}{ds} = \frac{\partial u}{\partial x} \frac{\partial x}{\partial s} + \frac{\partial u}{\partial y} \frac{\partial y}{\partial s} = 0.$$

2. The curve  $\vec{\gamma}$  must be a zero level set (iso-level contour) of  $u$  for all time  $t$ , and this leads to:

$$\frac{du(\vec{\gamma}(s, t), t)}{dt} = \frac{\partial u}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial u}{\partial y} \frac{\partial y}{\partial t} + \frac{\partial u}{\partial t} = 0.$$

Based on the two conditions above, one can show that if the evolution of  $\vec{\gamma}$  is described by the equation

$$\frac{d\vec{\gamma}(s, t)}{dt} = F(\vec{\gamma}(s, t))\vec{n}(s, t),$$

where  $F$  is a function defined on  $\mathcal{R}^2$ , then the corresponding evolution of  $u$  is given by:

$$\frac{\partial u(\mathbf{x}, t)}{\partial t} = F(\mathbf{x})\|\vec{\nabla} u(\mathbf{x}, t)\|.$$

As a result, the solution curve to our tracking problem will be given by the zero level set of the function  $u(\cdot, t)$  as  $t \rightarrow \infty$ .

In the case of tracking independent of intensity boundaries, the evolution equation for  $u$  corresponding to equation (4) becomes (omitting  $t$  for simplicity of notation):

$$\frac{\partial u(\mathbf{x})}{\partial t} = [\log P_{in, \mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0) - \log P_{out, \mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0) - \lambda_L \kappa_u(\mathbf{x})]\|\vec{\nabla} u\|, \quad (8)$$

where  $\kappa_u$  is the curvature of the level set of  $u$ , given as a function of  $u$  by:

$$\kappa_u = \vec{\nabla} \cdot \frac{\vec{\nabla} u}{\|\vec{\nabla} u\|} = \frac{u_{xx}u_y^2 - 2u_xu_yu_{xy} + u_{yy}u_x^2}{(u_x^2 + u_y^2)^{3/2}}. \quad (9)$$

On the other hand, the level set evolution equation corresponding to the curve evolution equation (6) for tracking accounting for intensity boundaries (Subsection III-B) is given by:

$$\begin{aligned} \frac{\partial u(\mathbf{x})}{\partial t} = & [\log P_{in,\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0) - \log P_{out,\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0) - \lambda_L \kappa_u(\mathbf{x})] \|\vec{\nabla} u\| \\ & - \lambda_I \vec{\nabla} u^T \mathbf{H}_I \frac{\vec{\nabla} \mathbf{I}^{n+1}}{\|\vec{\nabla} \mathbf{I}^{n+1}\|} + \lambda_I (\|\vec{\nabla} \mathbf{I}^{n+1}\| \kappa_u(\mathbf{x})) \|\vec{\nabla} u\|. \end{aligned} \quad (10)$$

The level set evolution equations (8),(10) are solved by replacing time derivatives by finite differences, and spatial derivatives by approximations using forward and backward differences, as suggested in [15], the key idea of such numerical approximations being that the numerical domain of dependence of a function should contain its mathematical domain of dependence ([15]). We refer the reader to [15] for a presentation and discussion of level set partial differential equation discretization schemes. We also refer the reader to [15] and [16] for a presentation and analysis of algorithms for fast computation of level set evolution equations.

#### IV. ESTIMATION OF THE PROBABILITY FUNCTIONS $P_{in,\mathbf{x}}$ AND $P_{out,\mathbf{x}}$

As was described in section II, the conditional independence and partial dependence assumptions allowed us to formulate our tracking/estimation problem in terms of the two probability functions  $P_{in,\mathbf{x}}$  and  $P_{out,\mathbf{x}}$ . In what follows, we shall first present an approximation to these probability functions. Starting from this approximation, we shall present generalizations in various directions.

##### A. Basic estimation

The membership probability function  $P_{in,\mathbf{x}}$  can be written

$$\begin{aligned} P_{in,\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0) &= P_{\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0, \mathbf{x} \in \mathbf{R}_1) \\ &= \int_{\Gamma} P_{\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0, \mathbf{x} \in \mathbf{R}_1, \phi = \psi) P(\phi = \psi|\mathbf{I}^n, \mathbf{R}_0, \mathbf{x} \in \mathbf{R}_1) d\psi \\ &= \int_{\{\psi \in \Gamma: \psi^{-1}(\mathbf{x}) \in \mathbf{R}_0\}} P_{\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0, \phi = \psi) P(\phi = \psi|\mathbf{I}^n, \mathbf{R}_0) d\psi, \end{aligned}$$

where the second equality is obtained by expressing the probability  $P_{\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0, \mathbf{x} \in \mathbf{R}_1)$  as the marginal probability of  $P_{\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0, \mathbf{x} \in \mathbf{R}_1, \phi = \psi)$  by integration with respect to the transformation  $\phi$ . Also, using the fact that  $\phi$  is a diffeomorphism of the image domain that maps  $\mathbf{R}_0$  onto  $\mathbf{R}_1$  (and hence  $\mathbf{R}_0^c$  onto  $\mathbf{R}_1^c$ ), the probability function  $P(\phi = \psi|\mathbf{I}^n, \mathbf{R}_0, \mathbf{x} \in \mathbf{R}_1)$  is evidently zero over the subset  $\{\psi \in \Gamma: \psi^{-1}(\mathbf{x}) \in \mathbf{R}_0^c\}$  of  $\Gamma$ , hence the third equality.

We now make a major simplifying assumption which will allow us to approximate this last integral: We assume that the probability function  $P_{\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0, \phi = \psi)$  is highly peaked both as a function on  $\{\psi \in \Gamma: \psi^{-1}(\mathbf{x}) \in \mathbf{R}_0\}$  and as a function on  $\{\psi \in \Gamma: \psi^{-1}(\mathbf{x}) \in$

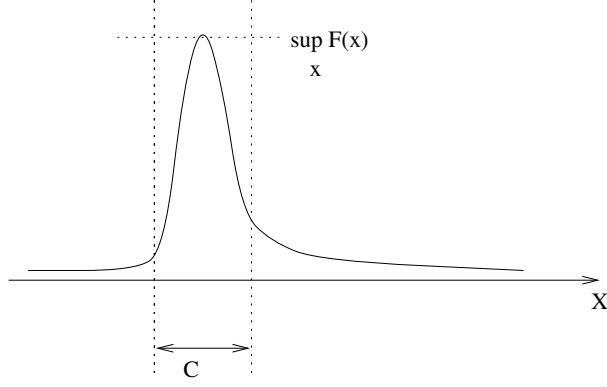


Fig. 3. Approximating  $\int_X F(x)dx$  by  $C \sup_{x \in X} F(x)$ .

$\mathbf{R}_0^c\}$ . In other words, we assume that  $P_{\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0, \phi = \psi)$  is concentrated on a small, family of transformations (see Fig. 3). This is an indirect assumption both on  $\Gamma$  and on the luminance/chrominance characteristics of the image in the region to be tracked and in the background: We are assuming that a small family of (possibly neighboring) transformations of  $\Gamma$  map a point of  $\mathbf{I}^{n+1}$  to a corresponding point of  $\mathbf{I}^n$ . This will happen if the family  $\Gamma$  is small enough, or if the region and background texture are sufficiently non-uniform. This is once again an assumption on the tracking problem expressed in probabilistic form. Consistent with our idea of not using any motion information for tracking, we can assume that the prior probability of  $\phi$  is uniform; extensions of our algorithm to include known a priori motion will be given in a later section. We finally obtain:

$$\begin{aligned} P_{in, \mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0) &= \int_{\{\psi \in \Gamma: \psi^{-1}(\mathbf{x}) \in \mathbf{R}_0\}} P_{\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0, \phi = \psi) P(\phi = \psi|\mathbf{I}^n, \mathbf{R}_0) d\psi \\ &\approx C_{in}(\mathbf{x}) \sup_{\{\psi \in \Gamma: \psi^{-1}(\mathbf{x}) \in \mathbf{R}_0\}} P_{\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0, \phi = \psi), \end{aligned}$$

where  $C_{in}$  is a normalizing function that depends also on  $\mathbf{I}^n$  and  $\mathbf{R}_0$ . For each  $\mathbf{x}$  in the image domain,  $C_{in}(\mathbf{x})$  is a direct function of the extent of the support of  $P_{\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0, \phi = \psi)$  in  $\{\psi \in \Gamma: \psi^{-1}(\mathbf{x}) \in \mathbf{R}_0\}$ . The larger this support, the larger  $C_{in}(\mathbf{x})$  will be.

Performing a similar approximation for  $P_{out, \mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0)$ , we obtain:

$$\begin{aligned} P_{out, \mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0) &= P_{\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0, \mathbf{x} \in \mathbf{R}_1^c) \\ &= \int_{\Gamma} P_{\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0, \mathbf{x} \in \mathbf{R}_1^c, \phi = \psi) P(\phi = \psi|\mathbf{I}^n, \mathbf{R}_0, \mathbf{x} \in \mathbf{R}_1^c) d\psi \\ &= \int_{\{\psi \in \Gamma: \psi^{-1}(\mathbf{x}) \in \mathbf{R}_0^c\}} P_{\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0, \phi = \psi) P(\phi = \psi|\mathbf{I}^n, \mathbf{R}_0) d\psi \\ &\approx C_{out}(\mathbf{x}) \sup_{\{\psi \in \Gamma: \psi^{-1}(\mathbf{x}) \in \mathbf{R}_0^c\}} P_{\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0, \phi = \psi), \end{aligned}$$

where  $C_{out}$  is another normalizing function (that depends also on  $\mathbf{I}^n$  and  $\mathbf{R}_0$ ) that is a direct

function of the extent of the support of  $P_{\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0, \phi = \psi)$  in  $\{\psi \in \Gamma : \psi^{-1}(\mathbf{x}) \in \mathbf{R}_0^c\}$ . Here again, the larger this support, the larger  $C_{out}(\mathbf{x})$  will be.

The precise behaviour of the functions  $C_{in}$  and  $C_{out}$  as functions of  $\mathbf{x}$  depends on the statistics of  $\mathbf{I}^n, \mathbf{I}^{n+1}$ , as well as on the structure of the family  $\Gamma$  of diffeomorphisms. This dependence is furthermore very intricate, and the precise estimation of these functions using local image statistics is the subject of our ongoing research. At this point however, we shall make the last simplifying assumption which will lead us to the solution of our tracking problem, and we shall assume that at each point  $\mathbf{x}$  in the image domain,  $C_{in}(\mathbf{x})$  and  $C_{out}(\mathbf{x})$  are equal, and we shall denote by  $C(\mathbf{x})$  their common value. This assumption essentially means that the set of transformations in  $\Gamma$  that best map  $\mathbf{x}$  to a *background* point in  $\mathbf{I}^n$  has the same measure as the set of transformations that best map  $\mathbf{x}$  to a *region* point in  $\mathbf{I}^n$ . Once again, this is an assumption on the image sequence, but in probabilistic disguise.

Using the observation model (1) and the above expressions, we can thus approximate  $-\log P_{in,\mathbf{x}}$  and  $-\log P_{out,\mathbf{x}}$  by the following expressions:

$$-\log P_{in,\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0) \approx \inf_{\{\psi \in \Gamma : \psi^{-1}(\mathbf{x}) \in \mathbf{R}_0\}} \frac{(\mathbf{I}^{n+1}(\mathbf{x}) - \mathbf{I}^n(\psi^{-1}(\mathbf{x})))^2}{2\sigma^2} - \log C(\mathbf{x}) \quad (11)$$

$$+ \frac{1}{2} \log(2\pi\sigma^2),$$

and

$$-\log P_{out,\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0) \approx \inf_{\{\psi \in \Gamma : \psi^{-1}(\mathbf{x}) \in \mathbf{R}_0^c\}} \frac{(\mathbf{I}^{n+1}(\mathbf{x}) - \mathbf{I}^n(\psi^{-1}(\mathbf{x})))^2}{2\sigma^2} - \log C(\mathbf{x}) \quad (12)$$

$$+ \frac{1}{2} \log(2\pi\sigma^2).$$

A key observation here is that the computation of the negative log likelihoods above is done *pointwise*. As a result, two distinct transformations  $\psi, \xi \in \Gamma$  for which  $\psi^{-1}(\mathbf{x})$  equals  $\xi^{-1}(\mathbf{x})$  will have the same contribution to the infima above and will be redundant. In each of the infima above, therefore, we can quotient the family  $\Gamma$  of diffeomorphisms by the equivalence relation  $\psi \sim \xi \Leftrightarrow \psi^{-1}(\mathbf{x}) = \xi^{-1}(\mathbf{x})$ . The quotient set  $\Gamma / \sim$  can then be identified with a subset of the group  $\Theta$  of plane translations. In the worst case, if  $\Gamma$  is large enough,  $\Gamma / \sim$  can be identified with  $\Theta$  itself. We shall make this worst case assumption and allow  $\Gamma$  to be so large that the search for the above infima reduce to a search over the complete group  $\Theta$ . The only simplification we shall make is to assume that the maximum range of motion from one frame to the next in the sequence is already known a priori. This is not a major restriction since the maximum range of motion of each point is already bounded by the diameter of the image domain; furthermore, in most applications, a maximum range of motion can be assumed with no loss of generality, reducing the computational burden. This allows us to further replace  $\Theta$  with the subset  $\Theta_\delta$  of



plane translations of norm less than or equal to the maximum range of motion  $\delta$ . We obtain:

$$-\log P_{in,\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0) \approx \inf_{\{z:\|\mathbf{z}\|\leq\delta, \mathbf{x}+\mathbf{z}\in\mathbf{R}_0\}} \frac{(\mathbf{I}^{n+1}(\mathbf{x}) - \mathbf{I}^n(\mathbf{x} + \mathbf{z}))^2}{2\sigma^2} - \log C(\mathbf{x}) \quad (13)$$

$$+ \frac{1}{2} \log(2\pi\sigma^2),$$

and

$$-\log P_{out,\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0) \approx \inf_{\{z:\|\mathbf{z}\|\leq\delta, \mathbf{x}+\mathbf{z}\in\mathbf{R}_0^c\}} \frac{(\mathbf{I}^{n+1}(\mathbf{x}) - \mathbf{I}^n(\mathbf{x} + \mathbf{z}))^2}{2\sigma^2} - \log C(\mathbf{x}) \quad (14)$$

$$+ \frac{1}{2} \log(2\pi\sigma^2),$$

where the right hand side is set to a large positive value whenever the search for the infimum is over an empty set. Since in any implementation the image plane  $\Omega$  is discretized, this search reduces to a finite search in a restricted neighborhood of pixel  $\mathbf{x}$ . This is precisely where the computational efficiency of this algorithm lies: Instead of searching for the transformation  $\phi$  in the possibly very large set of diffeomorphisms  $\Gamma$  and then mapping  $\mathbf{R}_0$  to  $\mathbf{R}_1$  via  $\phi$ , likelihoods of membership in  $\mathbf{R}_1$  are independently computed for every point in  $\Omega$  by searching in a much smaller and simpler set of transformations.

## V. LEVEL SET EVOLUTION EQUATIONS FOR TRACKING

### A. Basic equation

Using the probability estimates for  $P_{in,\mathbf{x}}$  and  $P_{out,\mathbf{x}}$  obtained in equations (13) and (14) together with the level set evolution equation (8) yields our proposed level set evolution equation for tracking:

$$\frac{\partial u(\mathbf{x})}{\partial t} = -\left[ \inf_{\{z:\|\mathbf{z}\|\leq\delta, \mathbf{x}+\mathbf{z}\in\mathbf{R}_0\}} (\mathbf{I}^{n+1}(\mathbf{x}) - \mathbf{I}^n(\mathbf{x} + \mathbf{z}))^2 - \inf_{\{z:\|\mathbf{z}\|\leq\delta, \mathbf{x}+\mathbf{z}\in\mathbf{R}_0^c\}} (\mathbf{I}^{n+1}(\mathbf{x}) - \mathbf{I}^n(\mathbf{x} + \mathbf{z}))^2 \right. \\ \left. + \lambda_L \kappa_u(\mathbf{x}) \|\vec{\nabla} u\|, \right] \quad (15)$$

where the noise variance  $\sigma^2$  has been absorbed in the coefficient  $\lambda_L$ . This is the basic equation we shall use. It is important to note that our complete level set evolution equation for tracking depends on only two parameters: The search range  $\delta$ , and the length prior coefficient  $\lambda_L$ .

### B. Using known motion information

Although the basic equation we have proposed in (15) does not use any motion information, it may happen, in some tracking applications, that motion estimates are available. This could be a byproduct of tracking itself in that a trajectory model could be built from consecutive tracking of a region in a sequence, and this trajectory model could be used to predict the region in a subsequent frame. Or it could be the result of a separate process. In any case, it is expected

that knowledge of motion should help in tracking. The extent to which knowledge of motion will improve tracking performance will of course depend on the reliability of the motion estimates. We can assume that this known motion is given as a vector field  $\mathbf{v}$  on the image domain, and relates a point  $\mathbf{x}$  of  $\mathbf{I}^{n+1}$  to a point  $\mathbf{x} + \mathbf{v}(\mathbf{x})$  of  $\mathbf{I}^n$ . To understand how to incorporate knowledge of this vector field in equation (15), it should be noted that the sets over which the infima are computed can be written as  $\{z : \|\mathbf{z}\| \leq \delta, \mathbf{x} + \mathbf{z} \in \mathbf{R}_0\} = \{z : \|\mathbf{z}\| \leq \delta\} \cap \{z : \mathbf{x} + \mathbf{z} \in \mathbf{R}_0\}$  and  $\{z : \|\mathbf{z}\| \leq \delta, \mathbf{x} + \mathbf{z} \in \mathbf{R}_0^c\} = \{z : \|\mathbf{z}\| \leq \delta\} \cap \{z : \mathbf{x} + \mathbf{z} \in \mathbf{R}_0^c\}$ , respectively. In both intersections, the set  $\{z : \|\mathbf{z}\| \leq \delta\}$  is an isotropic neighborhood of the origin in the plane, and can be suitably modified to reflect the known a priori about motion. Assume each motion vector  $\mathbf{v}(\mathbf{x})$  is known with error  $\epsilon(\mathbf{x})$ , i.e. the exact (unobservable) motion vector at  $\mathbf{x}$  is known to be in the closed ball  $\{z : \|\mathbf{z} - \mathbf{v}(\mathbf{x})\| \leq \epsilon(\mathbf{x})\}$  of center  $\mathbf{v}(\mathbf{x})$  and radius  $\epsilon(\mathbf{x})$ . We can immediately incorporate this information in our tracking algorithm by replacing the set  $\{z : \|\mathbf{z}\| \leq \delta\}$  above with the set  $\{z : \|\mathbf{z} - \mathbf{v}(\mathbf{x})\| \leq \epsilon(\mathbf{x})\}$ . This yields the following level set evolution equation:

$$\begin{aligned} \frac{\partial u(\mathbf{x})}{\partial t} = & - \left[ \inf_{\{z : \|\mathbf{z} - \mathbf{v}(\mathbf{x})\| \leq \epsilon(\mathbf{x}), \mathbf{x} + \mathbf{z} \in \mathbf{R}_0\}} (\mathbf{I}^{n+1}(\mathbf{x}) - \mathbf{I}^n(\mathbf{x} + \mathbf{z}))^2 \right. \\ & \left. - \inf_{\{z : \|\mathbf{z} - \mathbf{v}(\mathbf{x})\| \leq \epsilon(\mathbf{x}), \mathbf{x} + \mathbf{z} \in \mathbf{R}_0^c\}} (\mathbf{I}^{n+1}(\mathbf{x}) - \mathbf{I}^n(\mathbf{x} + \mathbf{z}))^2 + \lambda_L \kappa_u(\mathbf{x}) \right] \|\vec{\nabla} u\|. \end{aligned} \quad (16)$$

Note that whenever the motion field is known with absolute precision ( $\epsilon(\mathbf{x}) = 0$  for all  $\mathbf{x}$ ), the set  $\{z : \|\mathbf{z} - \mathbf{v}(\mathbf{x})\| \leq \epsilon(\mathbf{x})\}$  is reduced to the singleton  $\{\mathbf{v}(\mathbf{x})\}$ . In this case, tracking becomes tautological, as can be read from equation (16).

### C. Further extensions

We now present extensions to the basic level set equation (15). These extensions are obtained by refining the estimates of the probability distributions  $P_{in,\mathbf{x}}$  and  $P_{out,\mathbf{x}}$  (without loss of generality, the various normalizing constants have been discarded) ; the corresponding level set evolution equation can then be obtained by substituting these estimates in equation (8). Note that all these extensions can be combined in an obvious manner.

#### C.1 Extension to vector-valued images

The most obvious extension of the approximations described in the previous subsection to the case of vector-valued images is given by assuming that the various image components are uncorrelated and identically (Gaussian) distributed. This allows us to replace the squared difference expression in equations (11) and (12) with the squared norm of a difference. We thus obtain:

$$- \log P_{in,\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x}) | \mathbf{I}^n, \mathbf{R}_0) \approx \inf_{\{\psi \in \Gamma : \psi^{-1}(\mathbf{x}) \in \mathbf{R}_0\}} \frac{\|\mathbf{I}^{n+1}(\mathbf{x}) - \mathbf{I}^n(\psi^{-1}(\mathbf{x}))\|^2}{2\sigma^2},$$

and

$$-\log P_{out,\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0) \approx \inf_{\{\psi \in \Gamma: \psi^{-1}(\mathbf{x}) \in \mathbf{R}_0^c\}} \frac{\|\mathbf{I}^{n+1}(\mathbf{x}) - \mathbf{I}^n(\psi^{-1}(\mathbf{x}))\|^2}{2\sigma^2}.$$

### C.2 Estimation away from $\partial\mathbf{R}_0$

In any tracking sequence, the points which are most likely to be misclassified as being inside or outside of the region being tracked are those which are closest to the region boundary. Those which are further away from the boundary are less prone to such misclassification. This suggests that some measure of robustness can be added by performing the estimation in equations (11) and (12) by discarding points which are  $\epsilon$ -close to the boundary  $\partial\mathbf{R}_0$ . This yields the following approximations:

$$-\log P_{in,\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0) \approx \inf_{\{\psi \in \Gamma: \psi^{-1}(\mathbf{x}) \in \mathbf{R}_0, d(\psi^{-1}(\mathbf{x}), \partial\mathbf{R}_0) > \epsilon\}} \frac{(\mathbf{I}^{n+1}(\mathbf{x}) - \mathbf{I}^n(\psi^{-1}(\mathbf{x})))^2}{2\sigma^2},$$

and

$$-\log P_{out,\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0) \approx \inf_{\{\psi \in \Gamma: \psi^{-1}(\mathbf{x}) \in \mathbf{R}_0^c, d(\psi^{-1}(\mathbf{x}), \partial\mathbf{R}_0) > \epsilon\}} \frac{(\mathbf{I}^{n+1}(\mathbf{x}) - \mathbf{I}^n(\psi^{-1}(\mathbf{x})))^2}{2\sigma^2},$$

where  $d$  denotes a distance function on  $\Omega$ .

### C.3 Using local image structure

The search for the infimum in equations (11) and (12) is done by computing a squared difference expression. This expression is a measure of the mismatch between a point and its postulated corresponding match. Since a considerable amount of image structure is preserved during tracking, it is possible to take this local image structure into account in the search for the infimum. The most obvious way of doing so is to replace the squared difference expression at  $\mathbf{x}$  by a sum of squared differences in a neighborhood of  $\mathbf{x}$ . This yields:

$$-\log P_{in,\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0) \approx \inf_{\{\psi \in \Gamma: \psi^{-1}(\mathbf{x}) \in \mathbf{R}_0\}} \frac{\sum_{\mathbf{y} \in V_{\mathbf{x},\epsilon}} (\mathbf{I}^{n+1}(\mathbf{y}) - \mathbf{I}^n(\psi^{-1}(\mathbf{x})))^2}{2\sigma^2},$$

and

$$-\log P_{out,\mathbf{x}}(\mathbf{I}^{n+1}(\mathbf{x})|\mathbf{I}^n, \mathbf{R}_0) \approx \inf_{\{\psi \in \Gamma: \psi^{-1}(\mathbf{x}) \in \mathbf{R}_0^c\}} \frac{\sum_{\mathbf{y} \in V_{\mathbf{x},\epsilon}} (\mathbf{I}^{n+1}(\mathbf{x}) - \mathbf{I}^n(\psi^{-1}(\mathbf{x})))^2}{2\sigma^2},$$

where  $V_{\mathbf{x},\epsilon}$  is the  $\epsilon$ -ball around  $\mathbf{x}$ . Since the expressions above are not computed pointwise, we cannot reduce the set  $\Gamma$  of diffeomorphisms to a subset of the group of plane translations. Nevertheless, assuming the diffeomorphisms in  $\Gamma$  are close to being locally Euclidean, we can reduce  $\Gamma$  to a subset of the group of plane Euclidean transformations, as long as  $\epsilon$  is small enough. Once again, this allows us to replace the search over a possibly huge space of transformations with a number of independent searches over a much smaller space of transformations.

#### C.4 Extension to arbitrary distance functions

We can generalize the expressions in equations (11) and (12) even further, by completely discarding the assumption of Gaussian noise. This allows us to replace the squared difference expression with any distance function. This distance could be pointwise (e.g. the absolute difference of intensities at two points) or it could be computed over a neighborhood of the points under consideration (e.g. the sum of absolute differences of intensities in a neighborhood of the points). As was mentioned earlier, this does not always allow us to make the reduction from the family  $\Gamma$  of diffeomorphisms to a subset of the group of plane translations. To make this reduction, one has to ensure basic invariance properties of this distance function. This distance function could even be a distance between intensity distributions computed in the neighborhood of the points. Such a distance function, robust to partial occlusions, clutter, rotations, and changes in camera position has been proposed in [17]. Letting  $d$  be such a general distance function, the general level set evolution equation for tracking then becomes

$$\begin{aligned} \frac{\partial u(\mathbf{x})}{\partial t} = & - \left[ \inf_{\{z: \|\mathbf{z}\| \leq \delta, \mathbf{x} + \mathbf{z} \in \mathbf{R}_0\}} d((\mathbf{x}, \mathbf{I}^{n+1}), (\mathbf{x} + \mathbf{z}, \mathbf{I}^n)) - \inf_{\{z: \|\mathbf{z}\| \leq \delta, \mathbf{x} + \mathbf{z} \in \mathbf{R}_0^c\}} d((\mathbf{x}, \mathbf{I}^{n+1}), (\mathbf{x} + \mathbf{z}, \mathbf{I}^n)) \right. \\ & \left. + \lambda_L \kappa_u(\mathbf{x}) \right] \|\vec{\nabla} u\|. \end{aligned}$$

## VI. EXPERIMENTAL RESULTS

We illustrate our tracking algorithm on three real image sequences with natural motion. In all three experiments, an initial contour is manually outlined in frame 0 ( $\mathbf{I}^0$ ) of the sequence; this contour is then tracked from frame 0 to frame 1 ( $\mathbf{I}^1$ ), then from  $\mathbf{I}^1$  to  $\mathbf{I}^2$ , ... and so on until the last frame in the sequence. In all three experiments, region tracking from frame to frame is performed via equation (15), with, for color sequences, the extension described in Subsection V-C.1. All three experiments have been performed with the same value of the length prior coefficient  $\lambda_L = 10$ . The parameter  $\delta$ , on the other hand, which determines the maximum range of motion, has been set to different values.

The first tracking experiment is performed on the *parking* sequence. This is a gray level interlaced sequence with little interframe motion. The only object with any apparent motion is a white car tracing a circular path at the image speed of 1 to 2 pixels per frame. Accordingly, in this experiment, we have set  $\delta = 5$ . The body of the car, which is manually outlined in frame 0 and tracked over 150 frames, changes little shape overall, but its orientation is substantially modified in going from frame 0 to frame 150. The result is shown in Figure 4. Note that the car is accurately tracked over the first 20 frames. At frame 30, however, the contour on the right side of the car is slightly deformed. This can be explained by the fact that the luminance of the right side of the car and that of the road are very close. Hence, the only force acting on

the contour there is due to curvature and tends to shrink the contour. This shrinking continues until frame 50 and the contour then stabilizes. Frames 55 to 70 show changes in topology which can be handled by level set equations. In frame 55, part of the white marking on the road has been misclassified as part of the car. In frame 60 however, the portion of contour englobing the marking is severed from the portion englobing the car, again due to curvature. The same phenomenon is shown in frames 65 and 70. From frame 80 onwards, the car is accurately tracked without any substantial changes in the geometry of the tracking contour. Furthermore, one of the two contours associated to road markings fades away between frames 90 and 100. Note that the final tracking result, as shown in frame 150, is very close to the original, in frame 0. Note that the contour deformation in frame 30 could be avoided through the inclusion of a shape model. Such a shape model could be either included as a prior on the shape of the region being tracked, or used to refine estimates of the probability functions  $P_{in}$  and  $P_{out}$ .

The second experiment is performed on a subsequence of the *manege* sequence, which is an interlaced YUV color sequence. Specifically, we have constructed a new sequence (which we shall call *manege subsequence*) by temporally subsampling the *manege* sequence by a factor of 4, in order to construct longer range motion. Thus, frame  $k$  of *manege subsequence* corresponds to frame  $4k$  of the original *manege* sequence. At frame 0 of *manege subsequence*, the region to be tracked in subsequent frames, namely the body of the sailboat, is manually outlined. Motion of this region from frame to frame in *manege subsequence* is mainly translational, and mainly in the horizontal direction, with an average displacement of about 30 pixels. The vertical displacement of this region is an order of magnitude smaller than its horizontal component. Here, we have set  $\delta = 30$ . We track the region of interest over 10 frames of *manege subsequence*. Since the body of the sailboat undergoes a rotation in space around an axis parallel to the image plane, the shape of its projection on the image plane is modified in going from frame 0 to frame 10. Despite this fact, tracking has been performed very faithfully, as can be seen in Figure 5. Note that the rudder, which is prominent in frame 10, is barely visible in frame 0. Note also the precision with which the contour has tracked the object boundary, given that the interframe displacement is on the order of 30 pixels, and despite the fact that no priors on intensity edges were used. To be sure, some inaccuracies remain in the contour, especially on the top right corner of the sailboat. This is due to the fact that no shape models were used to guide the tracking. Improvements to these tracking results can be obtained by incorporating further a priori knowledge on shape other than mere smoothness. Note also that in this image sequence the background is far from being smooth, containing numerous strong intensity boundaries, and that it is far from being stationary, in stark contrast to the requirements of the algorithm presented in [13].

The third experiment is performed on the *autoroute* sequence, which is a gray level interlaced

sequence. A car is manually outlined in frame 0 and is tracked over 40 frames. Interframe displacement of the car from frame to frame is about 5 pixels. We have therefore set  $\delta = 5$ . Furthermore, the car's size changes substantially from frame 0 to frame 40, as it is shrunk by a factor of 3, and is even partially occluded by the car following it, as can be seen in frame 40. Once again, the initial contour is tracked very accurately, including even the mirror which appears as a small dent on the left side of the car. Note that in frame 10 the contour seems to go somewhat astray, as it locks to portions of the car shadow on the road. This does not persist however, and the contour returns to the car body. Note that overall, only the car body is being tracked, as desired. This would have not been the case, had tracking been based on intensity boundaries. Indeed, the car shadow on the road has strong enough contrast with respect to the road to mislead an intensity edge based tracking algorithm, as can be seen in the results on the *highway* sequence given in [13]. Note also that the tracking is very accurate despite the partial occlusion in frame 40.

To give some idea of the time efficiency of our algorithm, tracking from frame to frame of the *autoroute* sequence took approximately five seconds, on a 450 MHz Pentium-III PC under Linux. No attempt was made at using special techniques to reduce the computations [15], [16], nor at optimizing the code. It is expected that the use of such techniques will allow real-time tracking by our algorithm.

In closing, we illustrate the performance of our algorithm on a sequence with synthetic motion which has been constructed so as to show the limits of the proposed tracking level set equation (15). This sequence, which we call the *camouflage* sequence, has been constructed from a portion of the *aqua* YUV color sequence by cutting out a disc-like shape from the textured vegetation in the center of the image and pasting it so as to create apparent motion from the lower left corner to the upper right corner, as can be seen in Figure 7. The displacement of the disc-shaped object from frame to frame is about 15 pixels, so we have set  $\delta = 20$ . This disc-shaped object is manually outlined in frame 0 of the sequence and is tracked in subsequent frames, using equation (15) with  $\lambda_L = 10$  as before. Note that the tracking is accurate until and including frame 3 of the sequence, despite the fact that already in frame 3, most of the right part of the disc is overlapping a background with texture similar to that of the region. In frame 4 of the sequence, the right side of the tracked region is strongly deformed. This is due to the fact that the region and background textures are so similar there that the probability estimates  $P_{in}$  and  $P_{out}$  are almost identical for most of those points. These two probability functions cancelling each other out (see equation (8)), the curve evolution equation at those points is driven only by curvature. Note that on the left edge of the region, where the region and background textures are distinct, the tracking result remains accurate. In frame 5, the tracked region is even more

deformed, as it is almost completely superimposed on a background with identical texture. This sequence is indeed challenging, and without constraints on shape, the ambiguity between region and background in frames 4 and 5 cannot be resolved.

## VII. CONCLUSIONS

We have presented a novel method for tracking regions in image sequences. The novelty of the method lies in the fact that region tracking is formulated as a Bayesian estimation problem, with no motion model assumed nor any dense motion field computed. Furthermore, no assumptions, other than smoothness, are being made on the shape of the region being tracked nor on its intensity boundaries. In particular, the shape of the region being tracked is not constrained to belong to a particular parametrized family of shapes, nor should the region exhibit strong contrast with respect to the background. The only basic assumption of this algorithm is that a region's luminance/chrominance characteristics be relatively constant from frame to frame. Clearly, this is the minimum assumption on which any tracking algorithm should rely. The tracking algorithm is formulated as a Bayesian estimation problem and its solution provided by a level set partial differential equation. The benefits of this approach are numerical stability of the descent iterations and flexibility in topology. The experimental results we have shown demonstrate that this algorithm yields very accurate tracking results despite the minimality of assumptions on which it is based. Our current research is aimed at providing better estimates of the various probability functions used in the proposed algorithm.

## ACKNOWLEDGMENTS

The author would like to thank the two anonymous reviewers whose detailed questions and comments have contributed to improving the presentation of the proposed algorithm. This research was supported by the Natural Sciences and Engineering Research Council of Canada under strategic grant STR192788. The author would like to thank the CCETT, Rennes, France, the RACE DISTIMA project of the European Community, and the NHK of Japan for kindly providing the image sequences used in this work.

## REFERENCES

- [1] I. K. Sethi and R. Jain, "Finding trajectories of feature points in a monocular image sequence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 1, pp. 56-73, 1987.
- [2] J.L. Crowley, P. Stelmazyk, and C. Discours, "Measuring image flow by tracking edge lines," in *Proceedings of the 2<sup>nd</sup> International Conference on Computer Vision*, pp. 658-664, Florida, December 1988.
- [3] R. Deriche and O.D. Faugeras, "Tracking line segments," in *First European Conference on Computer Vision*, pp. 259-268, France, April 1990.
- [4] D.G. Lowe, "Robust model-based motion tracking through the integration of search and estimation," *International Journal of Computer Vision*, vol. 8, no. 2, pp. 113-122, 1992.

- [5] F. Leymarie, "Tracking Deformable Objects in the Plane Using an Active Contour Model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.15, no. 6, pp. 617-634, 1993.
- [6] M. Kass, A. Witkin, D. Terzopoulos, "Snakes: active contour models," in *First International Conference on Computer Vision*, pp. 259-268, 1987.
- [7] R. Curwen and A. Blake, pp. 39-57, *Active Vision*, MIT Press, 1992.
- [8] D. Terzopoulos and R. Szeliski, pp. 3-20, *Active Vision*, MIT Press 1992.
- [9] A. Blake, R. Curwen, and A. Zisserman, "Affine-invariant contour tracking with automatic control of spatio-temporal scale," *Proceedings of the 4<sup>th</sup> International Conference on Computer Vision*, Berlin, Germany, 1993.
- [10] R. Deriche and T. Blaszk, "Recovering and characterizing image features using an efficient model based approach," *Proceedings of Computer Vision and Pattern Recognition*, 1993.
- [11] M. Bertalmio, G. Sapiro, and G. Randall, "Morphing active contours: a geometric approach to topology-independent image segmentation and tracking," in *Proc. Int. Conf. Image Processing*, vol. III, pp. 318-322, 1998.
- [12] A.-R. Mansouri, A. Olivier, and J. Konrad, "Topology-independent region tracking with level sets," in *Proc. Int. Conf. Image Processing*, vol. III, pp. 66-69, 2000.
- [13] N. Paragios and R. Deriche, "Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no.3 , pp. 266-280, 2000.
- [14] S.C. Zhu and A.L. Yuille, "Region Competition: Unifying Snake/balloon, Region Growing and Bayes/MDL/Energy for Multi-Band Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, 1996.
- [15] J.A. Sethian, *Level Set Methods*, Cambridge University Press, 1998.
- [16] A.-R. Mansouri, T. Chomaud, and J. Konrad, "A comparative evaluation of algorithms for fast computation of level set PDEs with applications to motion segmentation," in *Proc. Int. Conf. Image Processing*, 2001 (to appear).
- [17] D. Comaniciu and V. Ramesh, "Mean shift optimal prediction for efficient object tracking," in *Proc. Int. Conf. Image Processing*, vol. III, pp. 70-73, 2000.
- [18] S.C. Zhu, "Embedding Gestalt Laws in Markov Random Fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no.11, pp. 1170-1187, 1999.



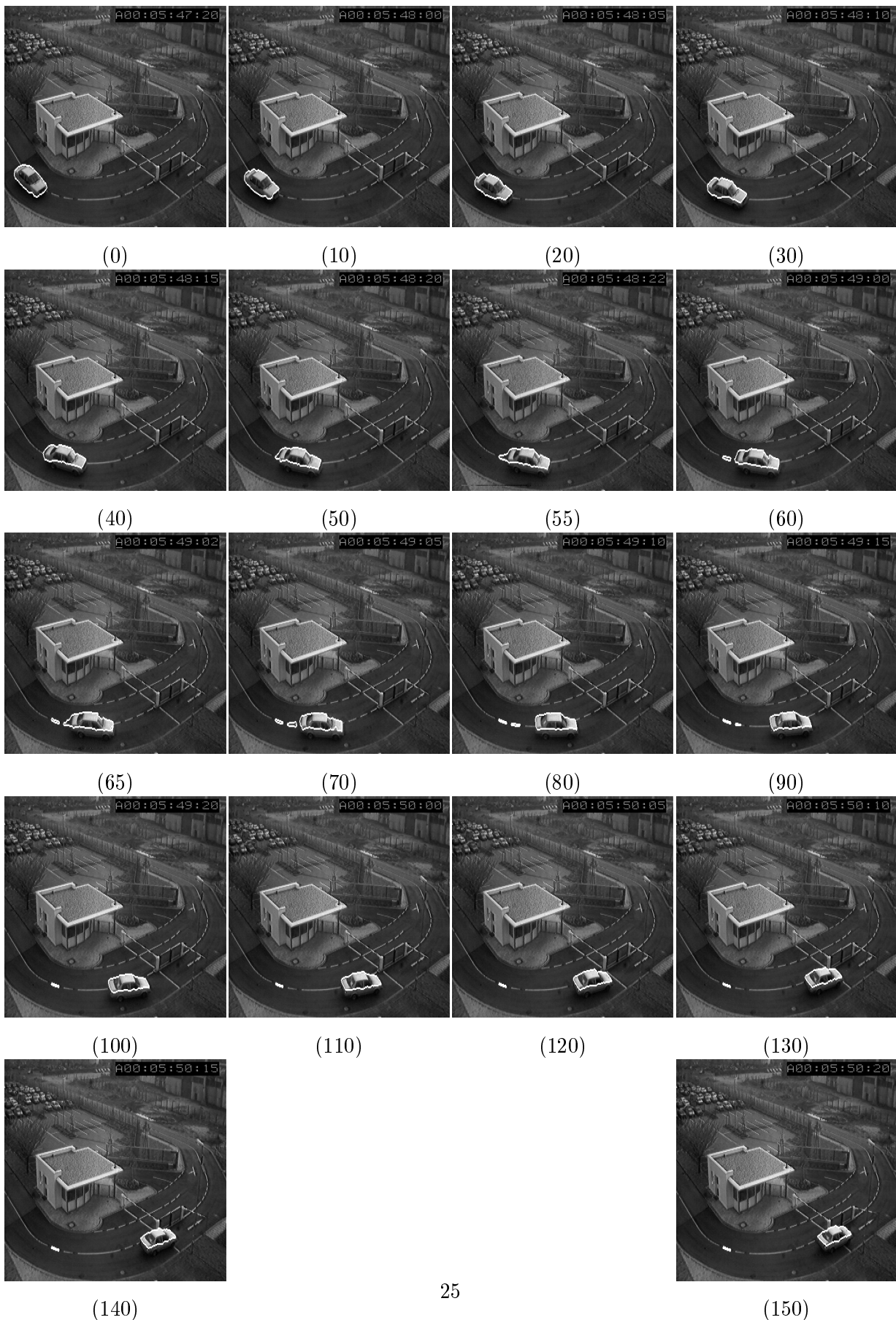


Fig. 4. Tracking on the *parking* sequence. The frame number is indicated below the pictures. Note the change in topology in frames 55 and 60. Note: Contour has been thickened for better visibility.



(0)

(1)



(2)

(3)



(5)

(10)

Fig. 5. Tracking on *manege* subsequence. The frame number is indicated below the pictures. Compare the rudder in frame 0 and in frame 10. Note: Contour has been thickened for better visibility.

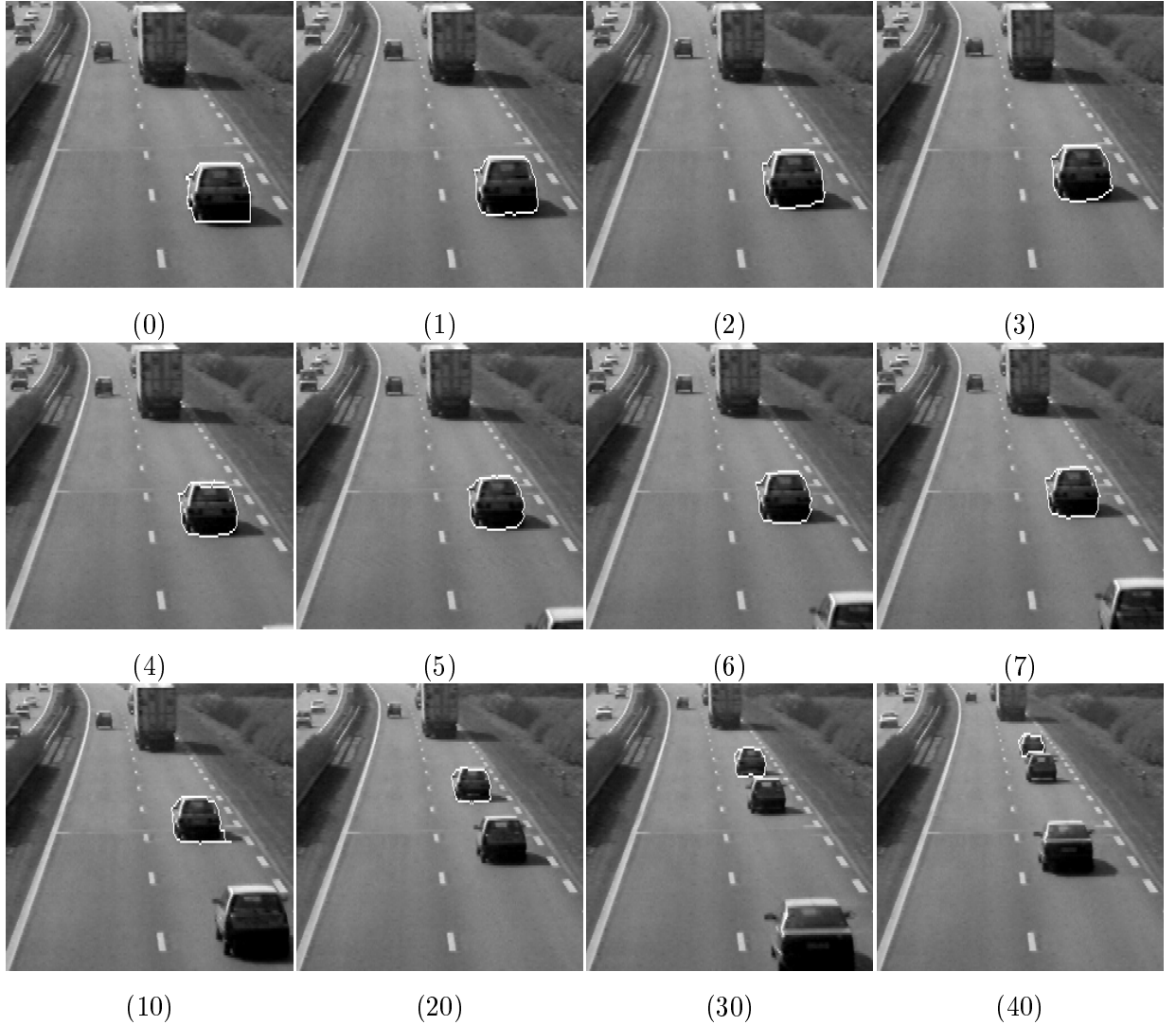


Fig. 6. Tracking on the *autoroute* sequence. The frame number is indicated below the pictures. Note the stability of the contour, despite the change in size of the car, and despite the occlusion in frame 40.



(0)



(1)



(2)



(3)



(4)



(4)

Fig. 7. Tracking on the *camouflage* synthetic sequence. This is a challenging sequence, as the region's texture is identical to a portion of the background's texture. The frame number is indicated below the pictures. Note: Contour has been thickened for better visibility.