

# Fast Incorporation of Optical Flow Into Active Polygons

Gozde Unal, *Member, IEEE*, Hamid Krim, *Senior Member, IEEE*, and Anthony Yezzi, *Member, IEEE*

**Abstract**—In this paper, we first reconsider, in a different light, the addition of a prediction step to active contour-based visual tracking using an optical flow and clarify the local computation of the latter along the boundaries of continuous active contours with appropriate regularizers. We subsequently detail our contribution of computing an optical flow-based prediction step directly from the parameters of an active polygon, and of exploiting it in object tracking. This is in contrast to an explicitly separate computation of the optical flow and its *ad hoc* application. It also provides an inherent regularization effect resulting from integrating measurements along polygon edges. As a result, we completely avoid the need of adding *ad hoc* regularizing terms to the optical flow computations, and the inevitably arbitrary associated weighting parameters. This direct integration of optical flow into the active polygon framework distinguishes this technique from most previous contour-based approaches, where regularization terms are theoretically, as well as practically, essential. The greater robustness and speed due to a reduced number of parameters of this technique are additional and appealing features.

**Index Terms**—Level-set methods, motion estimation, object tracking, optical flow, polygon evolution, region-based active contours.

## I. INTRODUCTION

THE problem of tracking moving objects in digital video data remains of great research interest in computer vision on account of various applications in video surveillance, monitoring, robotics, and video coding. Algorithms for extracting and tracking moving objects or targets over time in a video sequence are essential components in such applications. Tracking methods may be classified into two categories [1]: 1) motion-based approaches, which use motion segmentation of temporal image sequences by grouping moving regions over time, and by estimating their motion models [2]–[7]; 2) model-based approaches exploit some model structure to generally combat noisy conditions in the scene. Objects are usually tracked using a template of a three-dimensional (3-D) object such as 3-D models in [8]–[12]. Usage of this high-level

semantic information yields robust algorithms at a high computational cost. Another classification of object tracking methods due to [1] based on the type of information that the tracking algorithm uses is as follows. 1) Boundary-based methods use the boundary information along the object's contour. Methods using snake models such as [13]–[16], employ parameterized snakes (such as B-splines), and constrain the motion by assuming certain motion models, e.g., rigid, or affine. In [13], a contour's placement in a subsequent frame is predicted by an iterative registration process, where rigid objects, and rigid motion are assumed. In another tracking method with snakes [17], the motion estimation step is skipped, and the snake position from any given image frame is carried to the next frame. Other methods employ geodesic active contour models [1], and [18], which also assume rigid motion and rigid objects. 2) Region-based methods, such as [2]–[4], [19], [45] on the other hand, segment a temporal image sequence into regions with different motions. Regions segmented from each frame by a motion segmentation technique are matched to estimate motion parameters [20]. They usually employ parametric motion models, and they are computationally more demanding than boundary-based tracking methods because of region-matching cost.

Another tracking method, referred to as geodesic active regions [21], incorporates both boundary-based and region-based approaches, and an affine motion model is assumed. In feature-based trackers, one usually seeks similar features in subsequent frames. For instance, in [22], the features in subsequent frames are matched by a deformation of a current feature image onto the next feature image, and a level-set methodology is used to carry out this approach. One of the advantages of our technique is that of avoiding to have to match features, e.g., boundary contours, in a given image frame to those on successive ones. In another paper by Mansouri [23], the region competition functional is utilized in tracking in a time-varying image sequence rather than segmenting in a still image. Instead of having a motion estimation step, the region tracking problem is posed as a Bayesian estimation problem, and solved in a variational framework. While both this method and that proposed herein fall into a general variational framework, the former is distinguished by its Bayesian setting in a level-set contour model. Similarly, there are other approaches to object tracking which use posterior density estimation techniques [24], [25]. All of these algorithms similarly share a high computational cost, which we mean to avoid in this paper.

Motion of objects in a 3-D real world scene is projected onto two-dimensional (2-D) image plane, and this projected motion, referred to as “apparent motion,” or “2-D image motion,” or

Manuscript received April 13, 2003; revised May 8, 2004. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Christine Guillemot.

G. Unal is with the Intelligent Vision and Reasoning Department, Siemens Corporate Research, Princeton, NJ 08540 USA (e-mail: gozde.unal@siemens.com).

H. Krim is with the Electrical and Computer Engineering Department, North Carolina State University, Raleigh, NC 27695 USA (e-mail: ahk@eos.ncsu.edu).

A. Yezzi is with the School of Electrical Engineering, Georgia Institute of Technology, Atlanta, Atlanta, GA 30332-02501 USA (e-mail: ayezzi@ece.gatech.edu).

Digital Object Identifier 10.1109/TIP.2005.847286

sometimes also as “optical flow,” is to be estimated. In a time-varying image sequence,  $I(x, y, t) : [0, a] \times [0, b] \times [0, T] \rightarrow \mathbb{R}^+$ , ( $a, b, T \in \mathbb{R}^+$ ) image motion may be described by a 2-D vector field  $\mathbf{V}(x, y, t)$ , which specifies the direction and speed of the moving target at each grid point  $(x, y)$ , and time  $t$ . The measurement of visual motion is equivalent to computing  $\mathbf{V}(x, y, t)$  from  $I(x, y, t)$  [26].

As a popular approach to motion estimation, differential techniques [27], [28] assume that a point in a 3-D shape, when projected onto a 2-D image plane, has a constant intensity over time, and the corresponding optical flow constraint equation may be obtained as ( $\mathbf{x} = (x, y)$ )

$$\frac{\partial I(\mathbf{x}, t)}{\partial t} + \nabla I(\mathbf{x}, t) \cdot \mathbf{V}(\mathbf{x}, t) = 0. \quad (1)$$

This constraint is, however, insufficient for solving for both components of  $\mathbf{V}(\mathbf{x}, t) = (u(\mathbf{x}, t), v(\mathbf{x}, t))$ , and additional constraints on the velocity field are required to address the ill-posed nature of the problem. Horn and Schunck, in their pioneering work [27], combined the optical flow constraint with a global smoothness constraint on the velocity field. Imposing the regularizing smoothness constraint on the velocity over the whole image yields an oversmoothed motion estimates at the discontinuity regions such as occlusion boundaries and edges. Attempts to reduce the smoothing effects along steep edge gradients included variations such as incorporating an oriented smoothness constraint [29], or a directional smoothness constraint in a multiresolution framework [30]. Hildreth [26] proposed imposing the smoothness constraint on the velocity field only along contours extracted from time-varying images.

Another popular approach to tracking is based on Kalman filtering theory. The dynamical snake model of Terzopoulos and Szeliski [31] introduces a time-varying snake which moves until its kinetic energy is dissipated. The potential function of the snake represents image forces, and a general framework for a sequential estimation of contour dynamics is presented. The state space framework is indeed well adapted to tracking for not only sequentially processing time varying data but for also increasing robustness against noise as well. The dynamic snake model of [31] along with a motion control term are expressed as the system equations, whereas the optical flow constraint and the potential field are expressed as measurement equations by Peterfreund [32], [33]. The state estimation is performed by Kalman filtering. An analogy may be obtained here, since a state prediction step which uses the new information of the most current measurement is essential to our technique.

A generic dynamical system may be written as

$$\begin{aligned} \dot{\mathbf{P}}(t) &= \mathbf{F}(\mathbf{P}(t)) + \mathbf{V}(t) \\ \mathbf{Y} &= \mathbf{H}(\mathbf{P}(t)) + \mathbf{W}(t) \end{aligned} \quad (2)$$

where  $\mathbf{P}$  is the state vector (here the coordinates of a set of vertices of a polygon),  $\mathbf{F}$  and  $\mathbf{H}$  are the nonlinear vector functions describing the system dynamics and the output, respectively,  $\mathbf{V}$  and  $\mathbf{W}$  are noise processes, and  $\mathbf{Y}$  represents the output of the system. Only the output  $\mathbf{Y}$  of the system is measured and its observation over time serves in achieving one the main goals of model-based feedback control, namely to infer the complete

state  $\mathbf{P}$ . A rich literature on state observation [34] solves the system (2). This entails a sufficiently close approximation of the dynamical system and an accounts for noise effects, model uncertainties, and measurement errors. The implementation invokes an output error term designed to push the states of the simulated system toward the states of the actual system. The observer equations may then be written as

$$\dot{\hat{\mathbf{P}}}(t) = \hat{\mathbf{F}}(\hat{\mathbf{P}}(t)) + L(t)(\mathbf{Y}(t) - \hat{\mathbf{H}}(\hat{\mathbf{P}}(t))) \quad (3)$$

where  $L(t)$  is the error feedback gain, determining the error dynamics of the system. One of the most influential ways in designing this gain is the Kalman filter [35], where  $L(t)$  is usually called the Kalman gain matrix  $K$  which, designed to minimize the mean square estimation error (the error between simulated and measured output) exploits the known or estimated statistics of the Gaussian noise processes  $\mathbf{V}(t)$  and  $\mathbf{W}(t)$ .

In visual tracking, a sampled continuous reality is available, i.e., objects being tracked move continuously, but we are only able to observe the objects at specific times (e.g., depending on the frame rate of a camera), making  $\mathbf{Y}$  measurements available select sampled time instant  $t$ . This requires a slightly different observer which couples underlying continuous dynamics *and* sampled measurements. A continuous-discrete extended Kalman filter is obtained and given by the state estimate propagation equation

$$\dot{\hat{\mathbf{P}}}(t) = \hat{\mathbf{F}}(\hat{\mathbf{P}}(t)) \quad (4)$$

and the state estimate update equation

$$\hat{\mathbf{P}}_k(+) = \hat{\mathbf{P}}_k(-) + K_k(\mathbf{Y}_k - \mathbf{H}_k(\hat{\mathbf{P}}_k(-))) \quad (5)$$

where  $+$  denotes values after the update step,  $-$  values obtained from (4), and  $k$  is the sampling index. We assume that  $\mathbf{P}$  contains the  $(x, y)$  coordinates of the vertices of the active polygon. We note that (4) and (5) then correspond to a two step approach to tracking: 1) state propagation and 2) state update.

In our approach, given a time-varying image sequence, and initially outlined boundary contours of an object, step 1) is a **prediction step**, which predicts the position of a polygon at time step  $k$  based on its position and the optical flow field along the contour at time step  $k - 1$ . This is like a state update step. Step 2) refines the position obtained by step 1) through a spatial segmentation, referred to as a **correction step**, which is like a state propagation step. Past information is only conveyed by means of the location of the vertices and the motion is assumed to be piecewise constant from frame to frame.

#### A. Our Contribution

The novelty of our contribution, is in computing and utilizing an optical flow-based prediction step integrated in the parameters of an active polygonal model. This is in contrast to the usual addition of a prediction step to active contours using an optical flow with additional adapted regularizers. Our approach indeed naturally embeds a regularization effect in the polygonal model by integrating measurements along its edges. This, as a result, obviates the need for *ad hoc* regularizing terms to the optical flow computations.

Our proposed tracking approach may be viewed as model based because we will fully exploit a polygonal approximation model of objects to be tracked. The polygonal model results from a system of ordinary differential equations developed in [36]. More specifically, and with minimal assumption on the shape or boundaries of a target object, an initialized generic active polygon on an image, yields a flexible approximation model of an object. The tracking algorithm is, hence, an adaptation of this model and is inspired by evolution models which use region-based data distributions to capture polygonal object boundaries [36]. A fast numerical approximation of an optimization of a newly introduced information measure first yields a set of coupled ODEs, which, in turn, define a flow of polygon vertices to enclose a desired object.

We note that we avoid a time-varying estimation approach of [32], [33] which is based on boundary dynamic snake models and velocity estimation (either model based or optical flow based). In contrast to the active polygon model, the contour representation of most previous snake-based techniques, is a numerically sensitive snake model, which has to be sampled equidistantly in space. In our velocity estimation step, we achieve robustness to noise by a spatial averaging of image information over polygon edges. This is, again, in contrast to averaging optical flow measurements over time. In addition, our approach exploits the correlation among neighboring pixels to achieve robustness.

To better contrast existing continuous contour tracking methods to those based on polygonal models, and clarify differences between them, we proceed to a parallel development in this sequel. When used in formulating a tracking problem, active contours are considered continuous. This consists of moving each sample point on the contour with a velocity which preserves curve integrity. Under noisy conditions, however, the velocity field estimation usually requires regularization following its typical initialization as the normal component of the moving target boundaries, as shown in Fig. 1. The polygonal approximation of a target on the other hand, greatly simplifies the prediction step by only requiring a velocity field at the vertices as illustrated in Fig. 1. The resulting reduced number of vertices is clearly well adapted to man-made objects and is appealing in its simple and fast implementation.

This paper is organized as follows. In Section II, as noted earlier, for a better comparison context of tracking techniques, we present a continuous contour tracker, with an additional smoothness constraint. In Section III, we present a polygonal tracker and compare it to the continuous tracker. We provide simulation results and conclusions in Section IV.

## II. TRACKING WITH ACTIVE CONTOURS

Although there is a vast literature and a large diversity of models incorporating optical flow, we present an explanation and implementation on use of optical flow in visual tracking to establish a common ground among all these techniques which is useful to clarify the general idea, to fairly compare our results and show the benefits of novelties of our contribution.

Evolution of curves is a widely used technique in various applications of image processing such as filtering, smoothing,

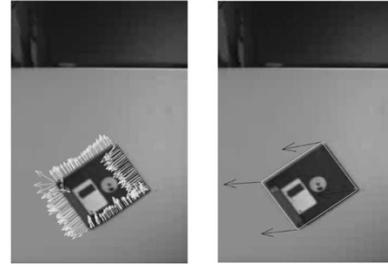


Fig. 1. Left: Velocity vectors normal to local direction of boundaries of an object which is translating horizontally toward left. Right: Velocity vectors at vertices of the polygonal boundary.

segmentation, tracking, registration, to name a few. Curve evolutions consist of propagating a curve via partial differential equations (PDEs). Denote a family of curves by  $\mathcal{C}(p, t') = (\mathcal{X}(p, t'), \mathcal{Y}(p, t'))$ , a mapping from  $\mathcal{I} \subset \mathbb{R} \times [0, T'] \rightarrow \mathbb{R}^2$ , where  $p \in \mathcal{I}$  is a parameter along the curve, and  $t'$  parameterizes the family of curves. This curve may serve to optimize an energy functional over a region  $R$  and, thereby, serve to capture contours of given objects in an image with the following [36], [37]

$$E(\mathcal{C}) = \int \int_R f(x, y) dx dy = \oint_{\mathcal{C}=\partial R} \langle \mathbf{F}, \mathbf{N} \rangle ds \quad (6)$$

where  $\mathbf{N}$  denotes the outward unit normal to  $\mathcal{C}$  (the boundary of  $R$ ),  $ds$  the Euclidean arclength element, and where  $\mathbf{F} = (F^1, F^2)$  is chosen so that  $\nabla \cdot \mathbf{F} = f$ . Toward optimizing this functional, it may be shown [37] that a gradient flow for  $\mathcal{C}$  with respect to  $E$  may be written as

$$\frac{\partial \mathcal{C}}{\partial t'} = f \mathbf{N} \quad (7)$$

where  $t'$  denotes the evolution time variable for the differential equation.

### A. Continuous Tracker With Optical Flow Constraint

Image features, such as edges or object boundaries, are often used in tracking applications. In the following, we will similarly exploit such features in tandem with an optical flow constraint which serves to predict a velocity field along object boundaries. This in turn is used to move the object contour in a given image frame  $I(\mathbf{x}, t)$  to the next frame  $I(\mathbf{x}, t + 1)$ . If a 2-D vector field  $\mathbf{V}(\mathbf{x}, t)$  is computed along an active contour, the curve may be moved with a speed  $\mathbf{V}$  in time according to

$$\frac{\partial \mathcal{C}(p, t)}{\partial t} = \mathbf{V}(p, t).$$

This is effectively equivalent to

$$\frac{\partial \mathcal{C}(p, t)}{\partial t} = (\mathbf{V}(p, t) \cdot \mathbf{N}(p, t)) \mathbf{N}(p, t)$$

as it is well known that a reparameterization of a general curve evolution equation is always possible, which in this case yields an evolution along the normal direction to the curve [38]. The velocity field  $\mathbf{V}(\mathbf{x})$  at each point on the contour at time  $t$  may, hence, be represented in terms of a parameter  $p$  as  $\mathbf{V}(p) =$

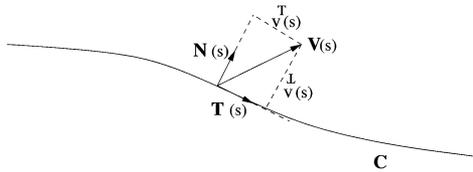


Fig. 2. Two-dimensional velocity field along a contour.

$v^\perp(p)\mathbf{N}(p) + v^T(p)\mathbf{T}(p)$ , with  $\mathbf{T}(p)$  and  $\mathbf{N}(p)$  respectively denoting unit vectors in the tangential and normal directions to an edge (Fig. 2).

Using (1), we may proceed to compute an estimate of the orthogonal component  $v^\perp$ . Using a set of local measurements derived from the time-varying image  $I(\mathbf{x}, t)$  and brightness constraints, would indeed yield

$$v^\perp(\mathbf{x}, t) = \frac{-I_t(\mathbf{x}, t)}{\|\nabla I(\mathbf{x}, t)\|}. \quad (8)$$

This provides the magnitude of the velocity field in the direction orthogonal to the local edge structure. It may in turn be used to write a curve evolution equation which preserves a consistency between two consecutive frames

$$\frac{\partial \mathbf{C}(p, t)}{\partial t} = v^\perp(p, t)\mathbf{N}(p, t), \quad 0 \leq t \leq 1. \quad (9)$$

An efficient method for implementation of curve evolutions, due to Osher and Sethian [39], is the so-called, level-set method. The parameterized curve  $\mathbf{C}(p, t)$  is embedded into a surface, which is called a level-set function  $\Phi(x, y, t) : \mathbb{R}^2 \times [0, T] \mapsto \mathbb{R}$ , as one of its level sets. This leads to an evolution equation for  $\Phi$ , which amounts to evolving  $\mathbf{C}$  in (7), and written as

$$\frac{\partial \Phi}{\partial t} = -f\|\nabla \Phi\|. \quad (10)$$

The prediction of the new location of the active contour on the next image frame of the image sequence can, hence, be obtained as the solution of the following PDE

$$\frac{\partial \Phi}{\partial t} = -v^\perp\|\nabla \Phi\|, \quad 0 \leq t \leq 1. \quad (11)$$

In the implementation, a narrowband technique which solves the PDE only in a band around the zero level set is utilized [40]. Here,  $v^\perp$  is computed on the zero level set and extended to other levels of the narrowband [39]. Most active contour models require some regularization to preserve the integrity of the curve during evolution, and a widely used form of this regularization is the arc length penalty. Then, the evolution for **the prediction step** takes the form

$$\frac{\partial \Phi}{\partial t} = \alpha\kappa - v^\perp\|\nabla \Phi\|, \quad 0 \leq t \leq 1 \quad (12)$$

where  $\kappa(x, y, t)$  is the curvature of the level-set function  $\Phi(x, y, t)$ , and  $\alpha \geq 0 \in \mathbb{R}$  is a weight determining the desired amount of regularization.

Upon predicting the curve at the next image frame, an update/correction step is usually required in order to refine the position of the contour on the next image frame. One typically exploits region-based active contour models to update the contour or the level-set function. These models assume that the image consists of a finite number of regions, that are characterized by a predetermined set of features or statistics such as means, and variances. These region characteristics are in turn used in the construction of an energy functional of the curve which aims at maximizing a divergence measure among the regions. One simple and convenient choice of a region-based characteristic is the mean intensity of regions inside and outside a curve [41], [42], which leads the image force  $f$  in (10) to take the form

$$f(x, y) = (u - v)(I(x, y) - u + I(x, y) - v) \quad (13)$$

where  $u$  and  $v$  respectively represent the mean intensity inside and outside the curve. Region descriptors based on information-theoretic measures or higher order statistics of regions may also be employed for increasing the robustness against noise and textural variations in an image [36]. **The correction step** is, hence, carried out by

$$\frac{\partial \Phi}{\partial t'} = \alpha'\kappa - f\|\nabla \Phi\|, \quad 0 \leq t' \leq T' \quad (14)$$

on the next image frame  $I(x, y, t + 1)$ . Here,  $\alpha' \geq 0 \in \mathbb{R}$  is included as a very small weight to help preserve the continuity of the curve evolution, and  $T'$  is an approximate steady-state reaching time for this PDE.

To clearly show the necessity of the prediction step in (12) in lieu of an update step alone, we show in the next example a video sequence of two marine animals. In this clear scene, a curve evolution is carried out on the first frame so that the boundaries of the two animals are outlined at the outset. Several images from this sequence shown in Fig. 3 demonstrate the tracking performance with and without prediction respectively in (rows 3 and 4) and (rows 1 and 2). This example clearly shows that the prediction step is crucial to a sustained tracking of the target, as a loss of target tracking results rather quickly without prediction. Note that the continuous model's "track loss" is due to the fact that region-based active contours are usually based on nonconvex energies, with many local minima, which may sometimes drive a continuous curve into a single point, usually due to the regularizing smoothness terms.

In the noisy scene of Fig. 4 (e.g., corrupted with Gaussian noise), we show a sequence of frames for which a prediction step with an optical flow-based normal velocity, may lead to a failed tracking on account of excessive noise. Unreliable estimates from the image at the prediction stage are the result of noise. At the correction stage, on the other hand, the weight of the regularizer, i.e., the arc length penalty, requires a significant increase. This, in turn, leads to rounding and shrinkage effects around the target object boundaries. This is tantamount to saying that the joint application of prediction and correction cannot guarantee an assured tracking under noisy conditions as may be seen in Fig. 4. One may indeed see that the active contour loses track of the rays after some time. This is a strong indication that additional steps have to be taken into account in

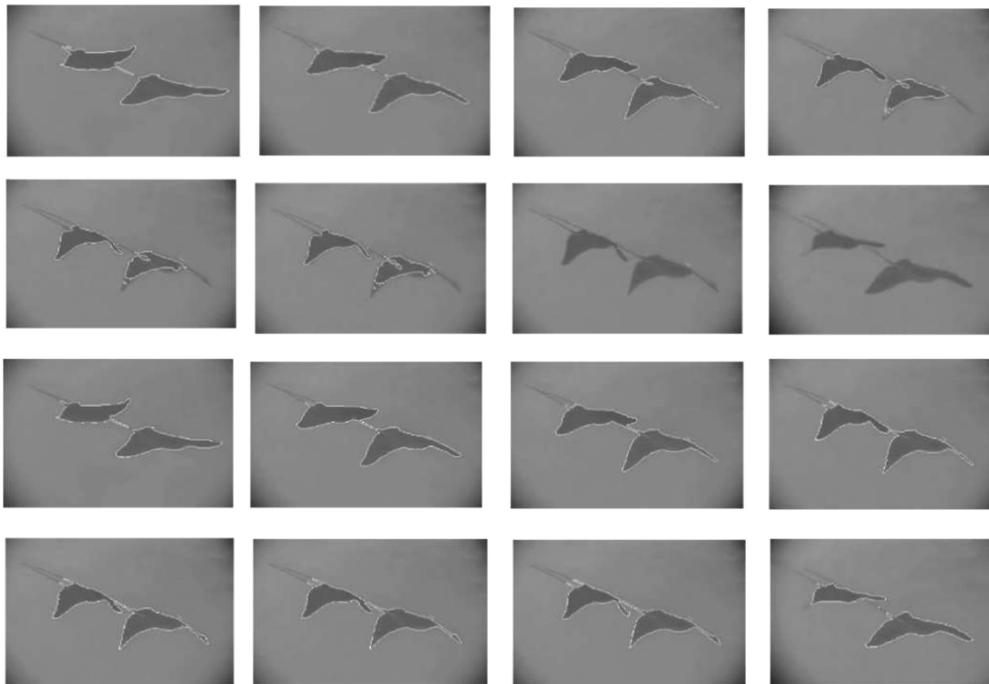


Fig. 3. Two rays are swimming gently in the sea (frames 1, 10, 15, 20, 22, 23, 24, and 69 are shown left to right, top to bottom). Rows 1 and 2: Tracking without prediction. Rows 3 and 4: Tracking with prediction using optical flow orthogonal component.

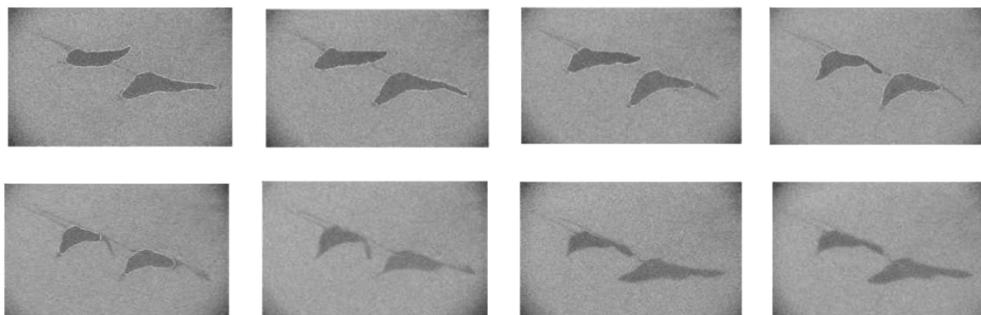


Fig. 4. Two-rays-swimming video noisy version (frames 1, 8, 13, 20, 28, 36, 60, and 63 are shown). Tracking with prediction using optical flow orthogonal component.

reducing the effect of noise. This may be in the form of regularization of the velocity field used in the prediction step.

### B. Continuous Tracker With Smoothness Constraint

Due to a well-known aperture problem, a local detector can only capture the velocity component in the direction perpendicular to the local orientation of an edge. Additional constraints are, hence, required to compute the correct velocity field. A smoothness constraint, introduced in [26] relies on the physical assumption that surfaces are generally smooth, and generate a smoothly varying velocity field when they move. There are still infinitely many solutions. A single solution may be obtained by finding a smooth velocity field that exhibits the least amount of variation among the set of velocity fields which satisfy the constraints derived from the changing image. The smoothness of the velocity field along a contour can be introduced by a familiar approach such as  $\int_{\mathcal{C}} \|\partial \mathbf{V} / \partial s\|^2 ds$ . Image constraints may be satisfied by minimizing the difference between the measurements  $v^\perp$  and the projection of the velocity field  $\mathbf{V}$  onto the normal

direction to the contour, i.e.,  $\mathbf{N}$ . The overall energy functional, thus defined by Hildreth [26], is given by

$$E(\mathbf{V}) = \int_{\mathcal{C}} \left\| \frac{\partial \mathbf{V}}{\partial s} \right\|^2 ds + \beta \int_{\mathcal{C}} [\mathbf{V} \cdot \mathbf{N} - v^\perp]^2 ds \quad (15)$$

where  $\beta$  is a weighting factor that expresses the confidence in the measured velocity constraints. The estimate of the velocity field  $\mathbf{V}$  may be obtained by way of minimizing this energy. This is in turn carried out by seeking a steady state solution of a PDE corresponding to the Euler Lagrange equations of the functional. In light of our implementation of the active contour model via a level-set method, the target object's contour is implicitly represented as the zero level set of a higher dimensional embedding function  $\Phi$ . The solution for the velocity field  $\mathbf{V}$ , defined over an implicit contour embedded in  $\Phi$ , is obtained with additional constraints such as derivatives that depend on  $\mathbf{V}$  which are intrinsic to the curve (a different case where data defined on a surface embedded into a 3-D level-set function is given in [43]). Following the construction in [43], the smoothness constraint of the velocity field, i.e., the first term in (15), corresponds to the

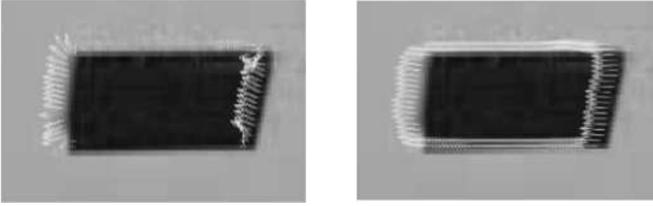


Fig. 5. Velocity normal to local direction of boundaries of an object which is translating horizontally is shown on the left, and the velocity field computed from (16) is given on the right (with  $\beta = 0.1$ , a time step of 0.24, and number of iterations = 400).

Dirichlet integral with the intrinsic gradient, and using the fact that the embedding function  $\Phi$  is chosen as a signed distance function, the gradient descent of this energy can be obtained as

$$\frac{\partial \mathbf{V}}{\partial t'} = \mathbf{V}_{ss} - \beta \left( \mathbf{V} \cdot \frac{\nabla \Phi}{\|\nabla \Phi\|} - v^\perp \right) \frac{\nabla \Phi}{\|\nabla \Phi\|}, \quad 0 \leq t' \leq T'. \quad (16)$$

Also by construction, the extension of the data defined on the curve  $\mathcal{C}$  over the narrowband satisfies  $\nabla \mathbf{V} \cdot \nabla \Phi = 0$ , which helped lead to (16) (here the gradient operator  $\nabla$  also acts on each component of  $\mathbf{V}$  separately). This PDE can be solved with an initial condition taken as the  $v^\perp \mathbf{N}$ , to provide estimates for the full velocity vector  $\mathbf{V}$  at each point on the contour, indeed at each point of the narrowband.

A blowup of a simple object subjected to a translational motion from a video sequence is shown in Fig. 5. The velocity vector at each sample point on the active contour moves from one frame to the next. The initial normal velocities are shown on the left, and the final velocity field is obtained as a steady state solution of the PDE in (16) and is shown on the right. It can be observed that the correct velocity on the boundary points, is closely approximated by the solution depicted on the right. Note that the zero initial normal speeds over the top and bottom edges of the object have been corrected to nonzero tangential speeds as expected.

The noisy video sequence of two-rays-swimming, shown in the previous section, is also tested with the same evolution technique, replacing the direct normal speed measurements  $v^\perp$  by the projected component of the estimated velocity field, that is  $\mathbf{V} \cdot \mathbf{N}$  as explained earlier. It is observed in Fig. 6 that the tracking performance is, unsurprisingly, improved upon utilizing Hildreth's method, and the tracker kept a better lock on objects. This validates the adoption of a smoothness constraint on the velocity field. The noise presence, however, heavily penalizes the length of the tracking contours, which in turn, leads to severe roundedness in the last few frames. If we furthermore consider its heavy computational load, we realize that the continuous tracker with its Hildreth-based smoothness constraint is highly impractical. Note that this last approach offers a new light on active contour tracking with a smoothness constraint, and helps us introduce our new approach.

In an attempt to address these problems and to better consider issues related to speed, we next propose a polygonal tracker nearly an order of magnitude faster than the most effective continuous tracker introduced in the previous sections. The advantage of our proposed technique is made clear by the resulting tracking speeds of various approaches displayed in Fig. 7. It is

readily observed that the smoothness constraint on the velocity field of a continuous tracker significantly increases the computation time of the algorithm, and that a more robust performance is achievable.

### III. POLYGONAL TRACKER

The goal of this section is to propose and develop a simple and efficient region-based tracking algorithm well adapted to polygonal objects. The idea is built on the insights gained from both the continuous tracker model and the polygon evolution model introduced in [36]. While technically, conventional discrete particle-based implementations of active contours (going back to the snake model of Kass, Witkin, and Terzopoulos [44]) may be considered polygonal models, there is a key difference both philosophically and numerically between these more conventional "active contours" implemented discretely as finely sampled polygons and what we are calling "active polygons." This difference is that conventional discrete implementations of active contours are point-wise in their treatment of the model. Namely, forces derived from image measurements to evolve each vertex are obtained only in the vicinity of that particular vertex. In our case, however, the basic element of the discrete model is no longer the vertex but instead the edges between vertices along which image information is "accumulated" in order to determine the motion of the vertex. This represents a variation on the mathematical treatment of the model as well as its numerics. In conventional snakes, it is desirable to have very closely spaced vertices since many sample points are needed to approximate the underlying "nonpolygonal" model (otherwise very little image information will be used) whereas in an "active polygon" it is desirable to have very few vertices in order to yield longer edges (which in turn allow for more averaging of image information along edges). Its suitability to tracking problems and its amenability to Kalman Filter-inspired prediction and correction steps make it an all around good choice as we elaborate next.

#### A. Velocity Estimation at Vertices

We presented in [36] gradient flows which could move polygon vertices so that an image domain be parsed into meaningfully different regions. Specifically, we considered a closed polygon  $\mathbf{P}$  as the contour  $\mathcal{C}$ , with a fixed number of vertices, say  $n \in \mathbb{N}$ ,  $\{\mathbf{P}_1, \dots, \mathbf{P}_n\} = \{(x_i, y_i), i = 1, \dots, n\}$ . The first variation of an energy functional  $E(\mathcal{C})$  in (6) for such a closed polygon is detailed in [36]. Its minimization yields a gradient descent flow by a set of coupled ordinary differential equations (ODEs) for the whole polygon, and, hence, an ODE for each vertex  $\mathbf{P}_k$ , and given by

$$\frac{\partial \mathbf{P}_k}{\partial t'} = \mathbf{N}_{1,k} \int_0^1 p f(L(p, \mathbf{P}_{k-1}, \mathbf{P}_k)) dp + \mathbf{N}_{2,k} \int_0^1 (1-p) f(L(p, \mathbf{P}_k, \mathbf{P}_{k+1})) dp \quad (17)$$

where  $\mathbf{N}_{1,k}$  (resp.  $\mathbf{N}_{2,k}$ ) denotes the outward unit normal of edge  $(\mathbf{P}_{k-1} - \mathbf{P}_k)$  (respectively,  $(\mathbf{P}_k - \mathbf{P}_{k+1})$ ), and  $L$  parameterizes a line between  $\mathbf{P}_{k-1}$  and  $\mathbf{P}_k$  or  $\mathbf{P}_k$  and  $\mathbf{P}_{k+1}$ . We note

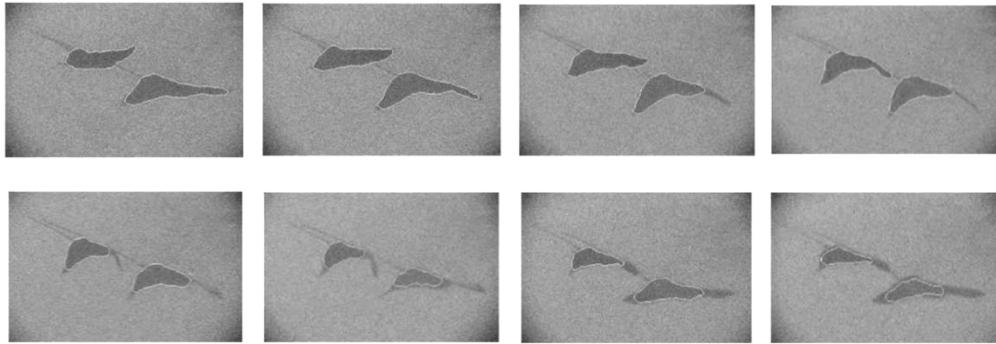


Fig. 6. Two-rays-swimming video noisy version (frames 1, 8, 13, 20, 28, 36, 60, and 63 are shown). Tracking with prediction using full optical flow computed via (16).

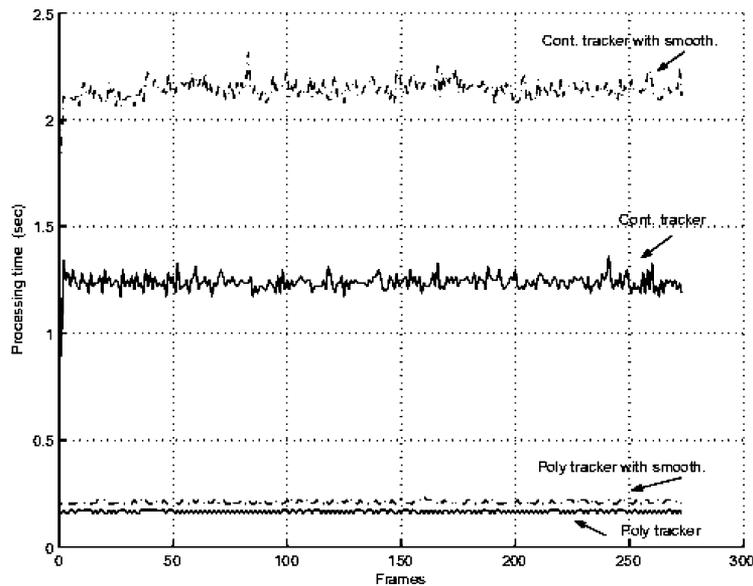


Fig. 7. Speed comparisons among different trackers introduced in this paper. From top to bottom, plots depicted are continuous tracker with smoothness constraint, continuous tracker, polygonal tracker with smoothness constraint, and polygonal tracker.

the similarity between this polygonal evolution equation which may simply be written in the form

$$\frac{\partial P_k}{\partial t} = f_1 N_{1,k} + f_2 N_{2,k}$$

and the curve evolution model given in (7), and recall that each of  $f_1$  and  $f_2$  corresponds to an integrated  $f$  on both neighboring edges of vertex  $P_k$ . Whereas each point of the curve in the continuous model moves as a single entity driven by a functional  $f$  of local as well as global quantities, each polygon edge in the proposed approach moves as a single unit moved along by its end vertices. The latter motion is in turn driven by information gleaned from two neighboring edges via  $f$ . In addition to the pertinent information captured by the functional  $f$ , its integration along edges provides an enhanced and needed immunity to noise and textural variability. This clear advantage over the continuous tracker, highlights the added gain from a reduced number of well separated vertices and its distinction from snake-based models.

The integrated spatial image information along adjacent edges of a vertex  $P_k$  may also be used to determine the speed and direction of a vertex on a single image, as well as to estimate

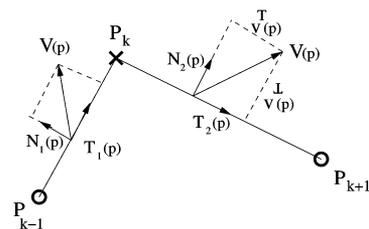


Fig. 8. Two-dimensional velocity field along two neighbor edges of a polygon vertex.

its velocity field on an active polygon laid on a time-varying image sequence. The estimated velocity vector at each vertex  $P_k$  using the two adjacent edges is schematically illustrated in Fig. 8.

The velocity field  $V(x, y)$  at each point of an edge may be represented as  $V(p) = v^\perp(p)N_i(p) + v^T(p)T_i(p)$ , where  $T_i(p)$  and  $N_i(p)$  are unit vectors in the tangential and normal directions of edge  $i$ . Once an active polygon locks onto a target object, the unit direction vectors  $N$  and  $T$  may readily be determined. A set of local measurements  $v^\perp$  (8) obtained from the optical flow constraint yield the magnitude of a velocity field

in an orthogonal direction of a local edge structure. Instantaneous measurements are unfortunately insufficient to determine the motion, and an averaged information is shown to be critical for an improved point velocity estimation. To that end, we utilize a joint contribution from two edges of a vertex to infer its resultant motion. Specifically, we address the sensitivity of the normal velocity measurements to noise by their weighted integration along neighboring edges of a vertex of interest. This leads to our prediction equation of vertex velocity

$$\frac{\partial \mathbf{P}_k}{\partial t} = \mathbf{V}_k = \mathbf{N}_{1,k} \int_0^1 p v^\perp(L(p, \mathbf{P}_{k-1}, \mathbf{P}_k)) dp + \mathbf{N}_{2,k} \int_0^1 (1-p) v^\perp(L(p, \mathbf{P}_k, \mathbf{P}_{k+1})) dp \quad (18)$$

for  $k = 1, \dots, n$ . To introduce further robustness, and to achieve more reliable estimates in the course of computing  $v^\perp$ , we may make use of smoother spatial derivatives (larger neighborhoods).

To fully exploit the vertices of an underlying polygon, our tracking procedure is initialized by delineating target boundaries by either region-based active polygon segmentation or manually. The prediction step of the velocity vector is carried out in (18), which in turn determines the locations of the polygon vertices at the next time instance on  $I(x, y, t + 1)$ . In a discrete setting, the ODE simply corresponds to

$$\mathbf{P}_k(t + 1) = \mathbf{P}_k(t) + \mathbf{V}_k(t) \quad (19)$$

if the time step in the discretization is chosen as 1.

The update step of the tracking seeks to minimize the deviation between current measurement/estimate of vertex location and predicted vertex location, by applying (17). Since both the prediction as well as the update stages of our technique call for a polygonal delineation of a target contour, a global regularizing technique we introduced in great detail in [36] is required to provide stability. Specifically, it makes use of the notion of an electrostatic field among the polygon edges as a means of self repulsion. This global regularizer technique provides an evolution without degeneracies and preserves the topology of the evolving polygon as a simple shape. The polygon-based segmentation/approximation of a target assumes an adequate choice of the initial number of vertices. Should this prior knowledge be lacking, we have developed a procedure which automatically adapts this number by periodic additions/deletions of new/redundant vertices as the case may be [36]. In some of the examples given below, this adaptively varying number of vertices approach is lumped together with the update step and will be pointed out in due course.

One may experimentally show that the velocity estimation step (prediction) of the polygonal tracker indeed improves performance. The following sequence in Fig. 9 shows a black fish swimming among a school of other fish. Tracking which only uses the spatial polygonal segmentation with an adaptive number of vertices (i.e., just carries the active polygon from one image frame onto the next one after a number of spatial segmentation iterations) may lose track of the black fish. In particular, as one notes in Fig. 9, a partial occlusion of the black

fish leads to a track loss (frame marked by LOST). The active polygon may be re-initialized after the occlusion scene (frame marked by RE-INITIALIZED), but to no avail as another track loss follows as soon as the fish turns around (second frame marked by LOST).

On the other hand, and as may be observed in Fig. 10, the polygonal tracker with the prediction step could follow the black fish under rougher visibility conditions such as partial occlusions and small visibility area when the fish is making a turn around itself. A successful tracking continues for all 350 frames of the sequence. This example demonstrates that the tracking performance is improved with the addition of the optical flow estimation step, which, as described earlier, merely entails the integration of the normal optical flow field along the polygon adjacent edges to yield a motion estimate of a vertex.

### B. Polygonal Tracker With Smoothness Constraint

A smoothness constraint may also be directly incorporated into the polygonal framework, with in fact much less effort than required by the continuous framework in Section II-B. In the prediction stage, an initial vector of a normal optical flow may be computed all along the polygon over a sparse sampling on edges between vertices. A minimization of the continuous energy functional (15) is subsequently carried out by directly discretizing it, and taking its derivatives with respect to the  $x$  and  $y$  velocity field components. This leads to a linear system of equations which can be solved by a mathematical programming technique, e.g., the conjugate gradients as suggested in [26]. We have carried out this numerical minimization in order to obtain the complete velocity field  $\mathbf{V}$  along all polygon edges. For visualizing the effect of the smoothness constraint on the optical flow, a snapshot from a simple object in translational motion is shown in Fig. 11 where the first picture in a row depicts the normal optical flow component  $v^\perp \mathbf{N}$  initialized over the polygon. In this figure, the first row corresponds to a clean sequence, whereas the second row corresponds to the noisy version of the former. The velocity at a vertex may be computed by integrating according to (18), and shown in the second picture in a row. The complete velocity  $\mathbf{V}$  obtained as a result of the minimization of the discrete energy functional is shown in the third picture. It is observed that the estimated velocity field is smooth, satisfies the image constraints, and very closely approximates the true velocity. This result may be used in the active polygon framework by integrating the velocity field along a neighbor edge pair of each vertex  $\mathbf{P}_k$  for yet additional improvement on the estimate  $\mathbf{V}_k$

$$\mathbf{V}_k = \int_0^1 p \mathbf{V}(L(p, \mathbf{P}_{k-1}, \mathbf{P}_k)) dp + \int_0^1 (1-p) \mathbf{V}(L(p, \mathbf{P}_k, \mathbf{P}_{k+1})) dp \quad (20)$$

$k = 1, \dots, n$

as demonstrated on the right in Fig. 11 for  $n = 4$ . The active polygon can now be moved directly with (19) onto the next image frame. The update step follows the prediction step to continue the process. We, however, note that this full velocity estimation is not required in our algorithm even though it produces

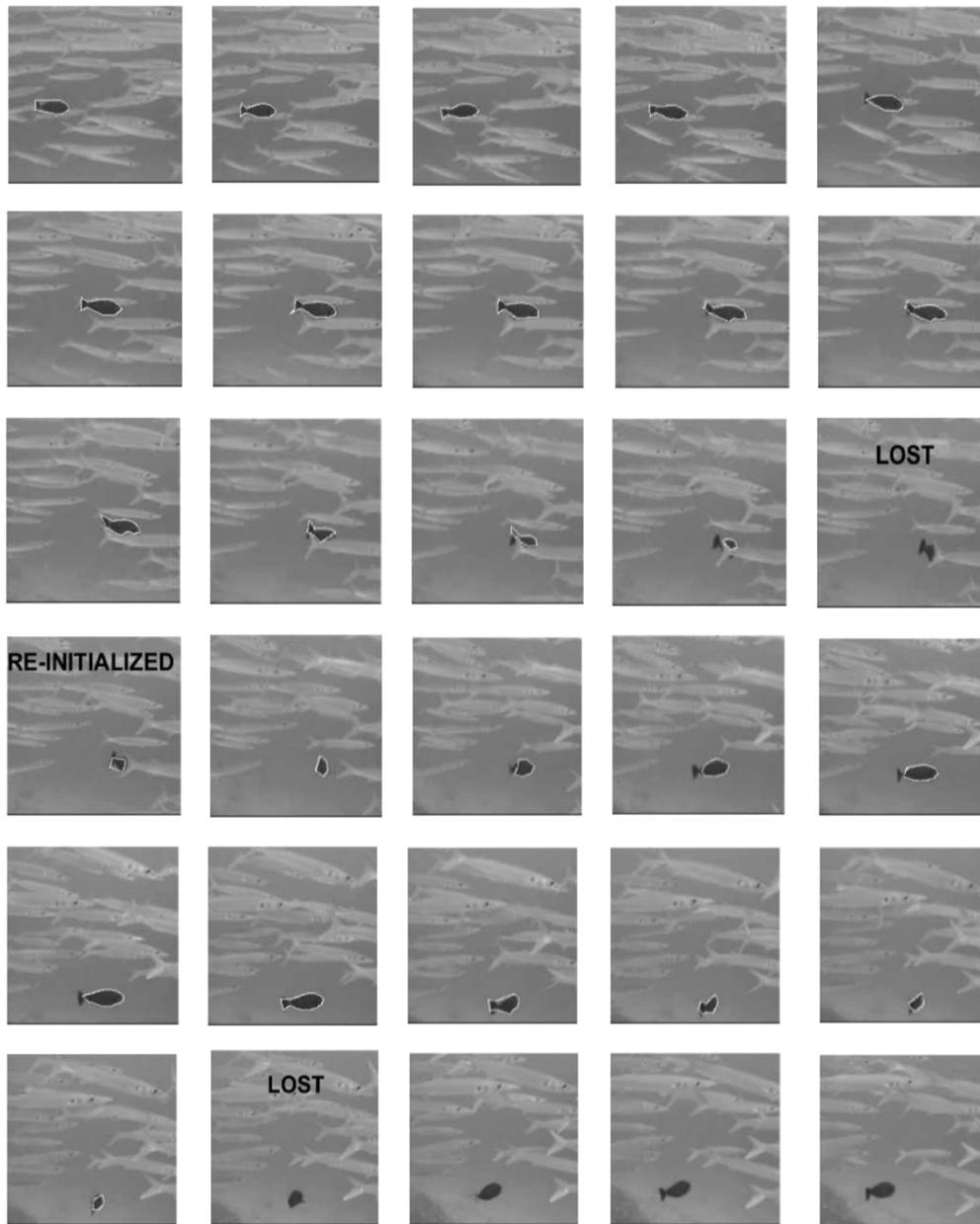


Fig. 9. Black fish swims among a school of other fish. Polygonal tracker with only the correction stage may lose track of the black fish when it is partly occluded by other fish, or turning backward.

a better velocity estimate, and we do not include this step in our algorithm for the experimental results.

#### IV. DISCUSSIONS AND RESULTS

In this section, we substantiate our proposed approach by a detailed discussion contrasting it to existing approaches, followed by numerical experiments.

##### A. Comparison Between the Continuous and the Polygonal Approaches

With a limited number of parameters, our active polygon-based model tracking framework enjoys several advantages over a continuous tracker, such as speed, reduced sensitivity to smoothness, inherent robustness to noise by integration of OF

on a group of pixels over the target boundaries. A comparison between the continuous and the polygonal approaches may be made on the basis of the following.

- If the true velocity field  $\mathbf{V}$  were to be *exactly* computed, the polygonal model would move the vertices of the polygon directly with the full velocity onto the next frame by  $\partial\mathbf{C}/\partial t = \mathbf{V}$  with unnecessary update. Such information could not, however, be so readily used by a continuous tracker, as its update would require a solution to a PDE  $\partial\Phi/\partial t = (\mathbf{V} \cdot \mathbf{N})\mathbf{N}$  (by level-set method). The zero level-set curve motion, as a solution to the PDE, only depends on the normal component of the velocity vector, and is, hence, unable to account for the complete direction of the velocity. Moreover, additive noise in continuous contours causes irregular displacements of

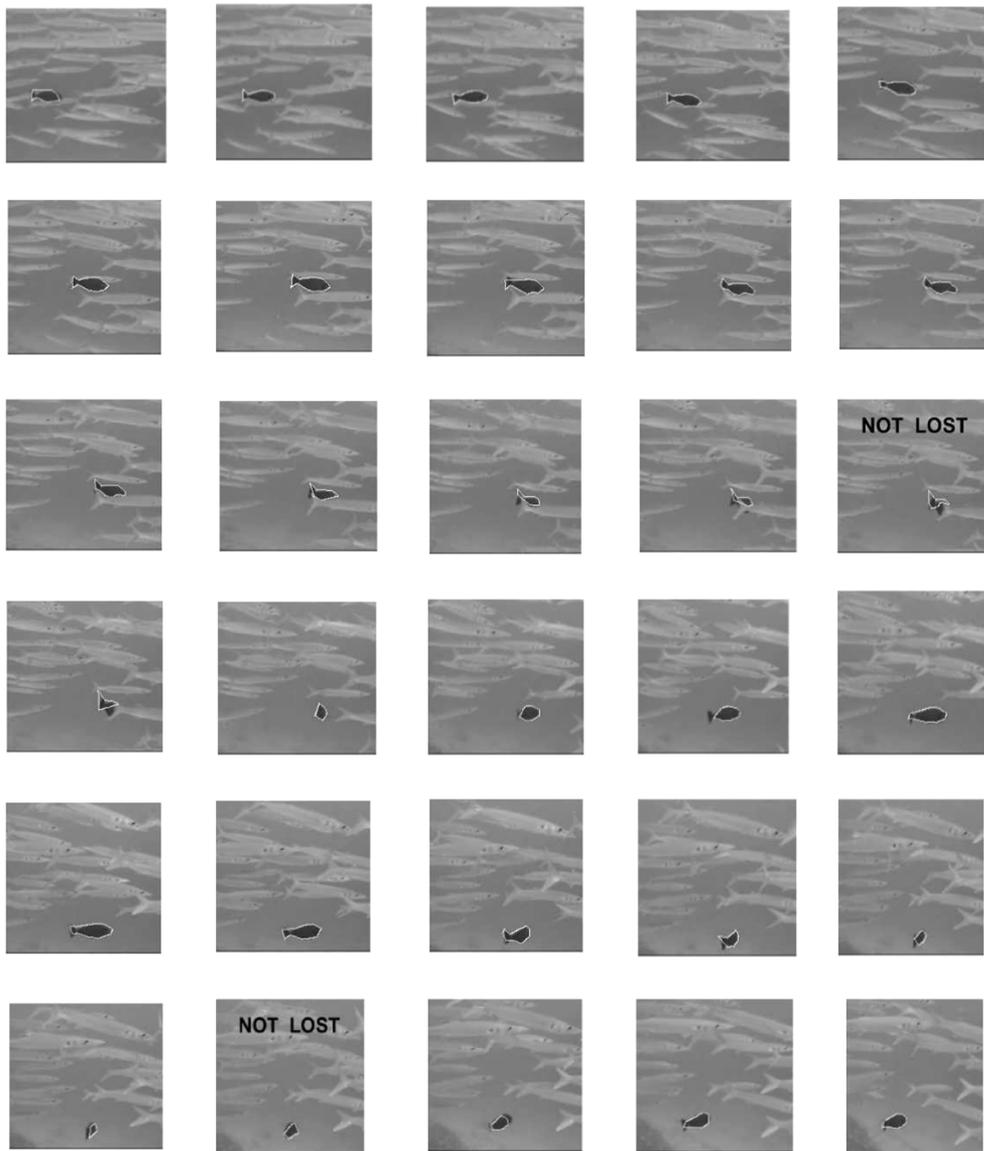


Fig. 10. Black fish swims among a school of other fish. Polygonal tracker with prediction stage successfully tracks the black fish even when there is partly occlusion or limited visibility.

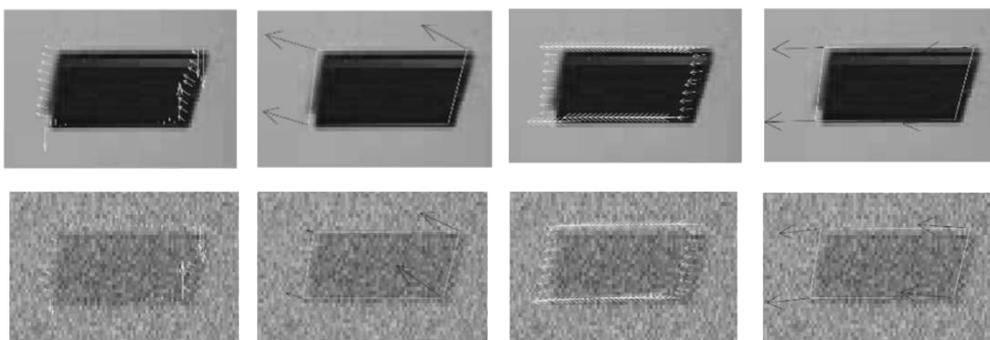


Fig. 11. Object is translating horizontally. Row 1: Clean version. Row 2: Noisy version. (Left to right) Picture 1: Velocity normal to local direction of boundaries. 2: Overall integrated velocity at the vertices from picture 1. 3: Velocity field computed through minimization of (15) with conjugate gradients technique. 4: Overall integrated velocity field at the vertices.

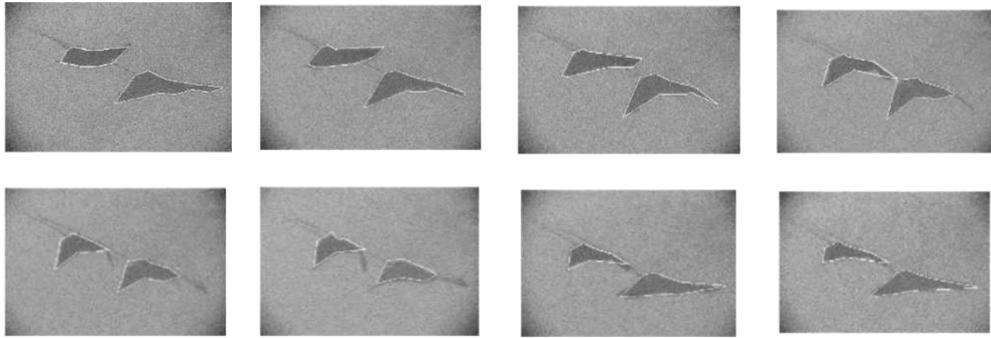


Fig. 12. Two-rays-swimming video noisy version (frames 1, 8, 13, 20, 28, 36, 60, and 63 are shown). Tracking via active polygons with prediction using optical flow normal component.

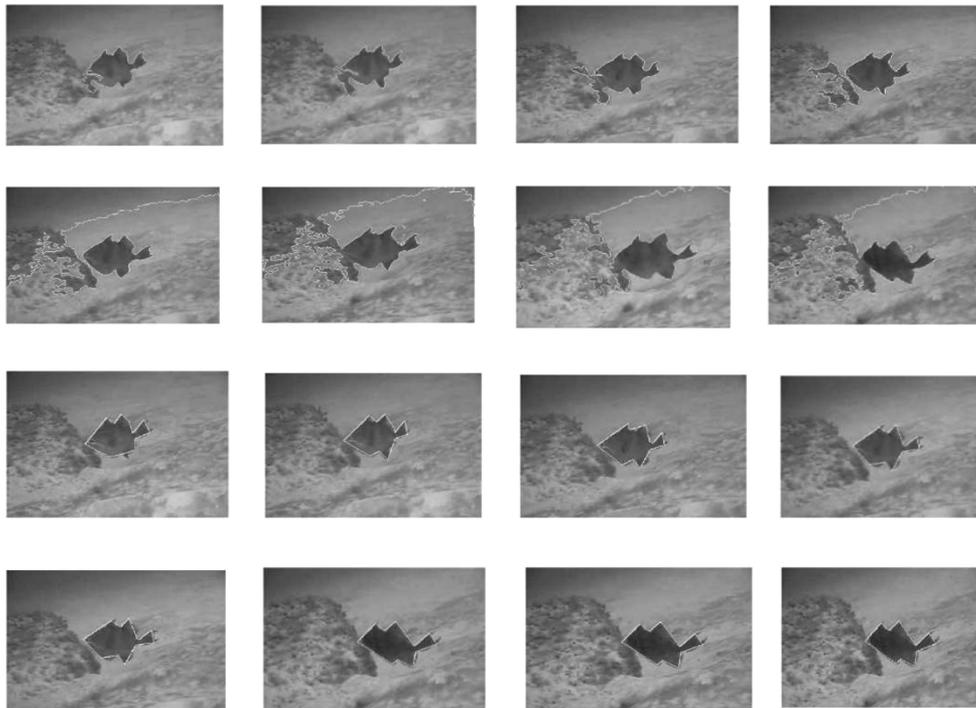


Fig. 13. A swimming fish in a rocky terrain in the sea (frames 1, 10, 20, 30, 40, 70, 110, and 143 are shown left to right, top to bottom). Rows 1 and 2: Continuous tracker fails to track the fish. Rows 3 and 4: Polygonal tracker successfully tracks the fish.

contour points, breakups and others. The well-separated vertex locations of the polygonal model, on the other hand takes full advantage of the complete optical flow field to avoid such problems.

- The polygonal approach owes its robustness to an averaging of information along edges adjacent to a moving vertex and gathered at all pixels; this is in contrast to a pixelwise information of the continuous model. The noisy video sequence of two-rays-swimming constitutes a good case study to unveil the pros and cons of both approaches. The continuous tracker, via a level-set implementation autonomously handles topological changes, and conveniently takes care of multiply connected regions, here the two swimming animals. Adapting the polygonal model to allow topology changes may be done by observing the magnitudes of its self-repulsion forces (which kicks in when polygonal edges are about to cross each other). This term can communicate to us

when and where a topological change should occur. For our intended applications, we do not pursue this approach. Handling multiple targets is easier than handling topology changes though, because the models we developed can be extended to multiple polygons which evolve separately with coupled ODEs.<sup>1</sup> Snapshots from the noisy two-rays-swimming sequence illustrate the polygonal tracker (here, for the sake of example, two animals could be separately tracked and the results are overlaid) in Fig. 12. The ability of the continuous tracker to automatically handle topological changes, is overshadowed by its sensitivity to noise which is likely to cause breakdown making this property less pronounced. The prediction and update steps, inspired from a statistical filtering perspective endow the polygonal approach with

<sup>1</sup>We note that this methodology does not make any distinction when two objects cross or partially superpose, and a multihypothesis tracking may be useful in such cases.

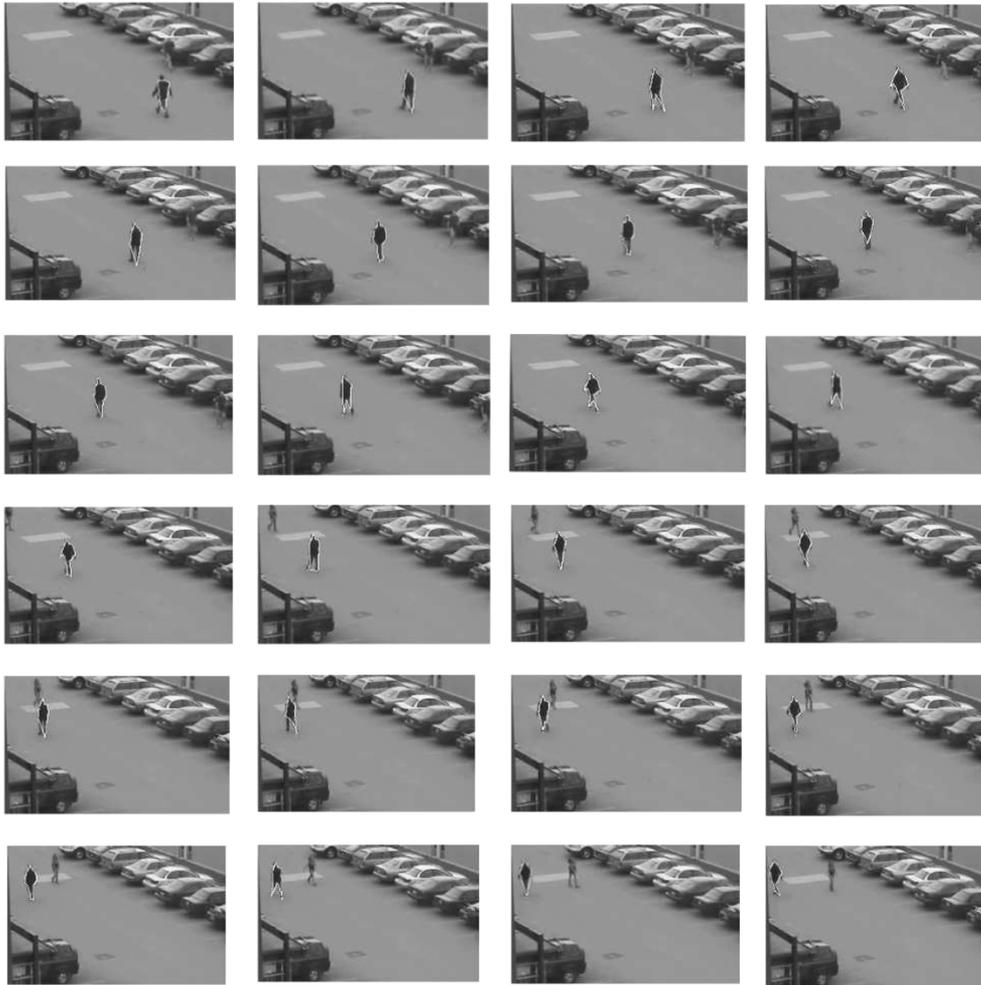


Fig. 14. Walking person (frames shown left to right, top to bottom) is tracked by the polygonal tracker.

much sought robustness. As already seen in Figs. 4 and 6, shrinkage and rounding effects may be very severe in the presence of a significant amount of noise in the scene due to necessary large regularization in continuous tracking. This is in contrast to the electrostatic forces used in conjunction with the polygonal model as well as the latter's resilience to external textural variability. We also note here that the region-based descriptor  $f$  used in the update step is the same in both the continuous and polygonal tracker examples shown, and is as given in (13).

- The lower number of degrees of freedom present in moving a polygon makes leaking through background regions more unlikely than for a continuous curve being easily attracted toward unwanted regions. This is illustrated by an example shown in Fig. 13, with a fish swimming in a rocky sea terrain. As the background bears similar region characteristics as the fish, the continuous tracker with its ease in split and merge encloses unrelated regions other than the target fish in the update step. The polygonal contour, in this case, follows the fish by preserving the topology of its boundaries. This is also an illustration for handling topological changes automatically may be either an advantage or a disadvantage.

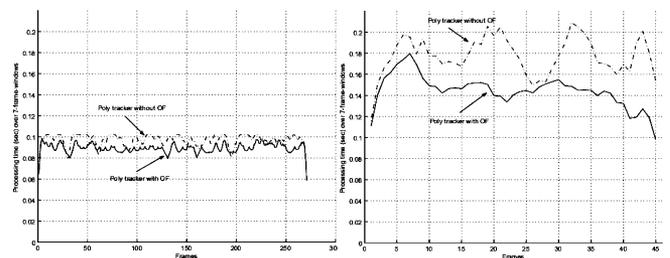


Fig. 15. Processing time (averaged over seven window frames) versus frames (left) for the original sequence and (right) for the sequence subsampled by six in time.

- The speed performance of the polygonal tracker is superior to that of the continuous tracker. A comparison is given in Fig. 7, where the plots depict the computation speed versus frames for both the polygonal and the continuous models. The polygonal tracker with or without the smoothness constraint is approximately eight times faster than the continuous model with or without the smoothness constraint.
- The intrinsic regularity of the proposed polygonal tracker is due to a natural regularizer term which prevents edge crossings, and has a significant influence only in close proximity of this near-pathological case.

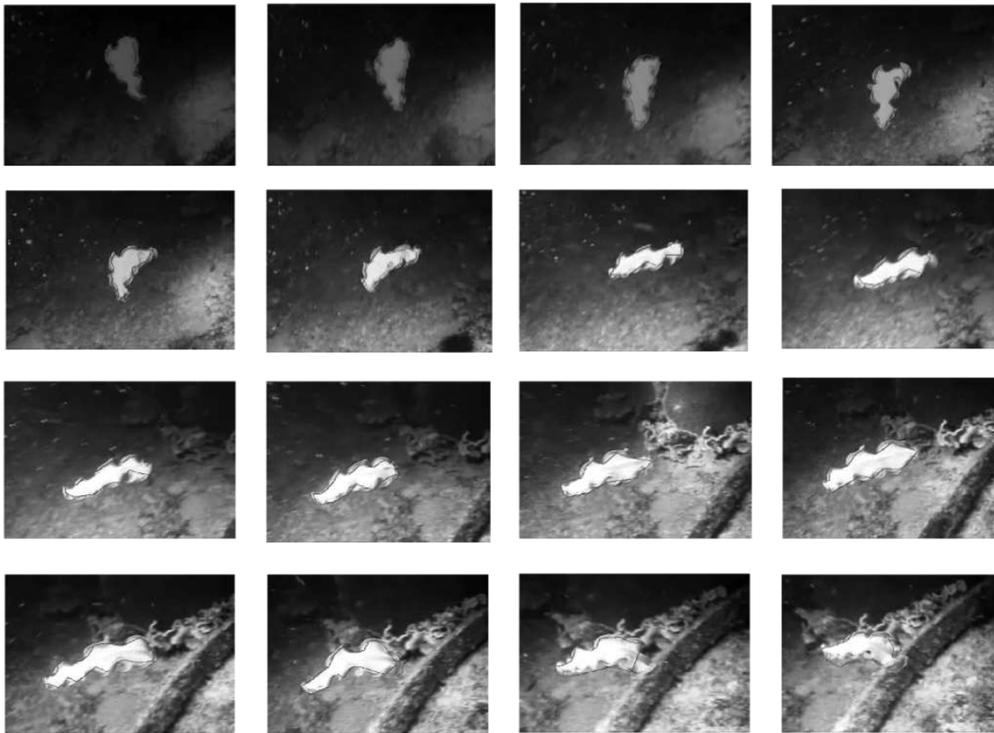


Fig. 16. Flatworm swims in a textured sea terrain (frames 1, 8, 14, 21, 28, 35, 42, 49, 56, 63, 70, 77, 85, 92, 99, and 104 are shown left to right, top to bottom). Polygonal tracker successfully tracks the flatworm.

### B. Experimental Results

Fig. 14 illustrates tracking in snapshots from a video sequence of a person walking in a parking lot.<sup>2</sup> The insertion of a prediction step in the tracking methodology is to speed up the computations by helping the active polygon glide onto a new image frame in the sequence, and smoothly adapt to displaced object's boundaries. The temporal resolution of the given sequence is quite high, and the scene changes from frame to frame are minimal. Nonetheless, when we plot the speeds of the polygonal tracker with and without the velocity prediction as depicted in Fig. 15 (left), we observe that the former is faster, confirming the expected benefit of the prediction step. To verify this effect for a sequence with lower temporal resolution, we decimate in time the sequence by six, and plot the speeds in Fig. 15 (right). When the temporal resolution of the sequence is decreased, the processing time for each frame increases as expected for both tracking methods. Although our velocity prediction scheme gives rough estimates, the tracking polygon is mapped to a position which is closer to the new object position in the new scene or frame. This is reflected in the given speed plots where the polygonal tracker without the prediction step, takes longer to flow the polygon toward the desired object boundaries.

The polygonal tracker with its ability to utilize various region-based descriptors could be used for tracking textured objects on textured backgrounds. A specific choice based on an information-theoretic measure [36] whose approximation uses high-order moments of the data distributions leads the image-based integrand  $f$  in (17) to take the form

<sup>2</sup>The sequences presented as snapshots here can be viewed at <http://users.ece.gatech.edu/~gozde/tracking1>.

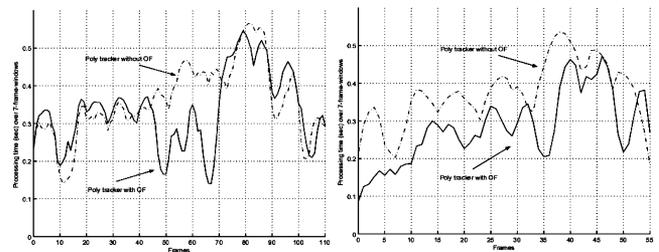


Fig. 17. Processing time (averaged over seven window frames) versus frames (left) for the original sequence and (right) for the sequence subsampled by two in time.

$f = \sum_{j=1}^m (u_j - v_j) ((G_j(I) - u_j) + (G_j(I) - v_j))$  with functions  $G$  chosen as for instance  $G_1(\xi) = \xi e^{-\xi^2/2}$ , and  $G_2(\xi) = e^{-\xi^2/2}$ . When using the descriptor  $f$  in our update step with an adaptive number of vertices, a flatworm swimming at the sea bottom may be captured through the highly textured sequence by the polygonal tracker in Fig. 16. The speed plots in Fig. 17 depict the speeds for the tracker with and without prediction. The figure on the right is for the original sequence (whose plot is given on the left) which is temporally subsampled by two. Varying the number of vertices to account for shape variations of the worm slows down the tracking in general. As expected, the tracker with prediction still performs faster than the tracker without prediction. The difference in speeds becomes more pronounced in the subsampled sequence on the left. Similarly, a clownfish on a host anemone shown in Fig. 18, may also be tracked in a highly textured scene. The continuous trackers we have introduced in this paper, do not provide a continuous tracking in either of these examples, and

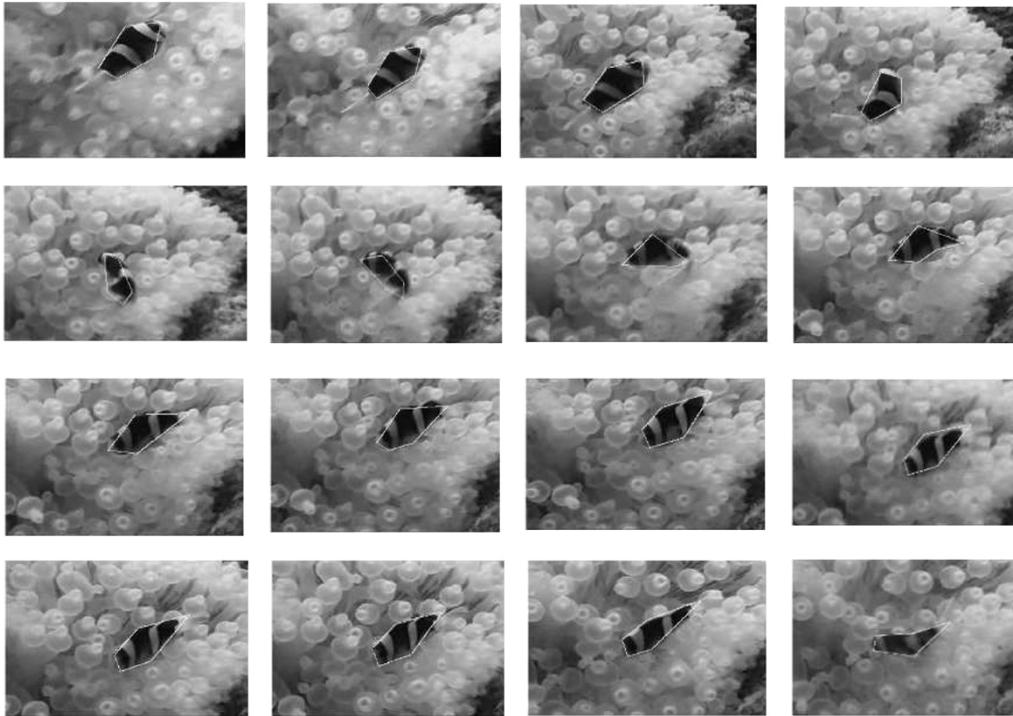


Fig. 18. Clownfish with its textured body swims in its host anemone (frames 1, 13, 39, 59, 64, 67, 71, 74, 78, 81, 85, 95, 105, 120, 150, and 155 are shown left to right, top to bottom). Polygonal tracker successfully tracks the fish.

they split, leak to background regions, and completely lose track of the target.

### C. Conclusion

In this paper, we have presented a simple but efficient approach to object tracking combining active contours framework with the optical-flow based motion estimation. Both curve evolution and polygon evolution models are utilized to carry out the tracking. The ODE model obtained for the polygonal tracker, can act on vertices of a polygon for their intra-frame as well as inter-frame motion estimation according to region-based characteristics as well as the optical-flow field's known properties. The latter is easily estimated from a well-known image brightness constraint. We have demonstrated by way of examples and discussion that our proposed tracking approach effectively and efficiently moves vertices through integrated local information with a resulting superior performance. We, moreover, note that no prior shape model assumptions on targets are made, since any shape may be approximated by a polygon. While the topology-change property provided by continuous contours in the level-set framework is not attained, this limitation may be an advantage if the target region stays simply connected. We also note that we avoid widely used assumptions in many object tracking methods which also exploit a motion detection step, e.g., a static camera. A motion detection step may also be added to this framework to make the algorithm more unsupervised in detecting motion in the scene, or handle the presence of multiple moving targets in the scene.

### ACKNOWLEDGMENT

The authors would like to thank M. Niethammer at The Georgia Institute of Technology for helpful discussions on the tracking techniques using the Kalman filtering theory.

### REFERENCES

- [1] N. Paragios and R. Deriche, "Geodesic active contours and level sets for the detection and tracking of moving objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 3, pp. 266–280, Mar. 2000.
- [2] F. G. Meyer and P. Bouthemy, "Region-based tracking using affine motion models in long image sequences," *Comput. Vis., Graph. Image Process.*, vol. 60, no. 2, pp. 119–140, 1994.
- [3] B. Bascle and R. Deriche, "Region tracking through image sequences," in *Proc. Int. Conf. Computer Vision*, 1995, pp. 302–307.
- [4] J. Wang and E. Adelson, "Representing moving images with layers," *IEEE Trans. Image Process.*, vol. 3, no. 5, pp. 625–638, May 1994.
- [5] T. Broida and R. Chellappa, "Estimation of object motion parameters from noisy images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 1, pp. 90–99, Jan. 1986.
- [6] J. Badenas, J. M. Sanchiz, and F. Pla, "Motion-based segmentation and region tracking in image sequences," *Pattern Recognit.*, vol. 34, pp. 661–670, 2001.
- [7] F. Marques and V. Vilaplana, "Face segmentation and tracking based on connected operators and partition projection," *Pattern Recognit.*, vol. 35, pp. 601–614, 2002.
- [8] D. Koller, K. Daniilidis, and H. H. Nagel, "Model-based object tracking in monocular image sequences of road traffic scenes," *Int. J. Comput. Vis.*, vol. 10, no. 3, pp. 257–281, 1993.
- [9] J. Rehg and T. Kanade, "Model-based tracking of self-occluding articulated objects," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1995, pp. 612–617.
- [10] D. Gavrial and L. Davis, "3-d model-based tracking of humans in action: A multi-view approach," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1996, pp. 73–80.
- [11] D. Lowe, "Robust model based motion tracking through the integration of search and estimation," *Int. J. Comput. Vis.*, vol. 8, no. 2, pp. 113–122, 1992.

- [12] E. Marchand, P. Bouthemy, F. Chaumette, and V. Moreau, "Robust real-time visual tracking using a 2D-3D model-based approach," in *Proc. Int. Conf. Computer Vision*, 1999, pp. 262–268.
- [13] M. O. Berger, "How to track efficiently piecewise curved contours with a view to reconstructing 3D objects," in *Proc. Int. Conf. Pattern Recognition*, 1994, pp. 32–36.
- [14] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *Proc. Eur. Conf. Computer Vision*, 1996, pp. 343–356.
- [15] Y. Fu, A. T. Erdem, and A. M. Tekalp, "Tracking visible boundary of objects using occlusion adaptive motion snake," *IEEE Trans. Image Process.*, vol. 9, no. 12, pp. 2051–2060, Dec. 2000.
- [16] C. Papin, P. Bouthemy, E. Memin, and G. Rochard, "Tracking and characterization of highly deformable cloud structures," presented at the Eur. Conf. Computer Vision, 2000.
- [17] F. Leymarie and M. Levine, "Tracking deformable objects in the plane using an active contour model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 6, pp. 617–634, Jun. 1993.
- [18] V. Caselles and B. Coll, "Snakes in movement," *SIAM J. Numer. Anal.*, vol. 33, no. 12, pp. 2445–2456, 1996.
- [19] B. Basclé and R. Deriche, "Stereo matching, reconstruction and refinement of 3d curves using deformable contours," in *Proc. Int. Conf. Computer Vision*, 1993, pp. 421–430.
- [20] J. Badenas, J. Sanchiz, and F. Pla, "Using temporal integration for tracking regions in traffic monitoring sequences," in *Proc. Int. Conf. Pattern Recognition*, 2000, pp. 1125–1128.
- [21] N. Paragios and R. Deriche, "Geodesic active regions for motion estimation and tracking," INRIA, Tech. Rep., 1999.
- [22] M. Bertalmio, G. Sapiro, and G. Randall, "Morphing active contours," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 7, pp. 733–737, Jul. 2000.
- [23] A.-R. Mansouri, "Region tracking via level set pdes without motion computation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 947–961, Jul. 2002.
- [24] A. Blake and M. Isard, *Active Contours*. London, U.K.: Springer Verlag, 1998.
- [25] B. Li and R. Chellappa, "A generic approach to simultaneous tracking and verification in video," *IEEE Trans. Image Process.*, vol. 11, no. 5, pp. 530–544, May 2002.
- [26] E. C. Hildreth, "Computations underlying the measurement of visual motion," *AI*, vol. 23, pp. 309–354, 1984.
- [27] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *AI*, vol. 17, pp. 185–203, 1981.
- [28] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Imaging Understanding Workshop*, 1981, pp. 121–130.
- [29] H. H. Nagel and W. Enkelmann, "An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 5, pp. 565–593, May 1986.
- [30] S. V. Fogel, "The estimation of velocity vector fields from time-varying image sequences," *CVGIP: Image Understanding*, vol. 53, no. 3, pp. 253–287, 1991.
- [31] D. Terzopoulos and R. Szeliski, *Active Vision*. Cambridge, MA: MIT Press, 1992, pp. 3–20.
- [32] N. Peterfreund, "Robust tracking of position and velocity with kalman snakes," *Pattern Anal. Mach. Intell.*, vol. 21, no. 6, pp. 564–569, 1999.
- [33] —, "The velocity snake: Deformable contour for tracking in spatio-velocity space," *CVGI*, vol. 73, no. 3, pp. 346–356, 1999.
- [34] D. G. Luenberger, "An introduction to observers," *IEEE Trans. Automat. Control*, vol. AC-16, no. 4, pp. 596–602, Apr. 1971.
- [35] A. Gelb, Ed., *Applied Optimal Estimation*. Cambridge, MA: MIT Press, 1974.
- [36] G. Unal, A. Yezzi, and H. Krim, "Information-theoretic active polygons for unsupervised texture segmentation," *IJCV*, vol. 62, no. 3, pp. 199–220, 2005.
- [37] S. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and bayes/MDL for multiband image segmentation," *IEEE Pattern Anal. Mach. Intell.*, vol. 18, no. 9, pp. 884–900, Sep. 1996.
- [38] B. Kimia, A. Tannenbaum, and S. Zucker, "Shapes, shocks, and deformations I," *Int. J. Comput. Vis.*, vol. 31, pp. 189–224, 1995.
- [39] S. Osher and J. Sethian, "Fronts propagating with curvature dependent speed: Algorithms based on the hamilton-jacobi formulation," *J. Comput. Phys.*, vol. 49, pp. 12–49, 1988.
- [40] D. Peng, B. Merriman, S. Osher, H.-K. Zhao, and M. Kang, "A PDE-based fast local level set method," *J. Comput. Phys.*, vol. 255, pp. 410–438, 1999.
- [41] T. Chan and L. Vese, "An active contour model without edges," in *Proc. Int. Conf. Scale-Space Theories in Computer Vision*, 1999, pp. 141–151.
- [42] A. Yezzi, A. Tsai, and A. Willsky, "A fully global approach to image segmentation via coupled curve evolution equations," *J. Vis. Commun. Image Representation*, vol. 13, pp. 195–216, 2002.
- [43] M. Bertalmio, L. Cheng, S. Osher, and G. Sapiro, "Variational problems and partial differential equations on implicit surfaces," *J. Comput. Phys.*, vol. 174, no. 2, pp. 759–780, 2001.
- [44] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, Jan. 1988.
- [45] D. Cremers and S. Soatto, "Motion competition: A variational framework for piecewise parametric motion segmentation," *Int. J. Comput. Vis.*, vol. 62, pp. 249–265, 2005.

**Gozde Unal** (S'96–M'03) received the Ph.D. degree in electrical engineering from North Carolina State University, Raleigh, in 2002.

She was a Postdoctoral Fellow at Georgia Institute of Technology, Atlanta, from Fall 2002 to Spring of 2003, and visited HP Labs, Palo Alto CA, during summer of 2003. She joined the Intelligent Vision and Reasoning Department, Siemens Corporate Research, Princeton NJ, in Fall 2003. Her research interests include variational techniques with connections to information theory and probability theory, applications to various computer vision and image processing problems such as stereo camera calibration, 2-D/3-D image segmentation and registration, filtering and enhancement, and stochastic particle systems.

**Hamid Krim** (M'80–SM'98) received all of his degrees in electrical engineering.

As a Member of Technical Staff at AT&T Bell Labs, Murray Hill, NJ, he worked in the areas of telephony and digital communication systems/subsystems. Following an NSF Postdoctoral Fellowship at Foreign Centers of Excellence, LSS/University of Orsay, Paris, France, he became a Research Scientist at the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, performing and supervising research. He is presently on the faculty in the Electrical and Computer Engineering Department, North Carolina State University, Raleigh. His research interests are in statistical signal and image analysis and mathematical modeling with a keen emphasis on applied problems.

**Anthony Yezzi** (M'99) received the Ph.D. degree from the Department of Electrical Engineering, University of Minnesota, Minneapolis, in 1997.

After completing a postdoctoral research position in LIDS, Massachusetts Institute of Technology, Cambridge, he began his faculty position at the Georgia Institute of Technology, Atlanta, as an Assistant Professor in 1999. Currently, he is an Associate Professor with the School of Electrical Engineering at the Georgia Institute of Technology. He has consulted for a number of medical imaging companies, including GE, Picker, and VTI. His research lies primarily within the fields of image processing and computer vision. His work within these fields includes anisotropic diffusion for image smoothing, active contours, segmentation, multiframe shape from shading, stereoscopic reconstruction, and shape analysis. His work in anisotropic smoothing and segmentation has been largely motivated and directed toward problems in medical imaging applied to MRI, ultrasound, CT, and OCT modalities. Two central themes of his research, in general, are curve/surface evolution theory from differential geometry and partial differential equations.