# Geometric Observers for Dynamically Evolving Curves

Marc Niethammer, Patricio A. Vela, and Allen Tannenbaum

*Abstract*— **This paper proposes a deterministic observer design for visual tracking based on non-parametric implicit (level-set) curve descriptions. The observer is continuous-discrete, with continuous-time system dynamics and discrete-time measurements. Its state-space consists of an estimated curve position augmented by additional states (e.g., velocities) associated with every point on the estimated curve. Multiple simulation models are proposed for state prediction. Measurements are performed through standard static segmentation algorithms and optical-flow computations. Special emphasis is given to the geometric formulation of the overall dynamical system. The discrete-time measurements lead to the problem of geometric curve interpolation and the discrete-time filtering of quantities propagated along with the estimated curve. Interpolation and filtering are intimately linked to the correspondence problem between curves. Correspondences are established by a Laplace-equation approach. The proposed scheme is implemented completely implicitly (by Eulerian numerical solutions of transport equations) and thus naturally allows for topological changes and subpixel accuracy on the computational grid.**

*Index Terms*— **Geometric Observer, Level Set Method, Curve Evolution, Visual Tracking**

## I. INTRODUCTION

The filtering of sensed data is a practical necessity when using the data to inform a feedback process. Real world signals, as derived from sensing devices, have noise and disturbances that must be dealt with prior to incorporating the data into the feedback loop. Vision sensors suffer from these, plus other unique drawbacks, therefore it is sensible (and often essential) to consider the filtering of the feedback information provided from visual sensors. Visual sensors are fundamentally different from traditional sensors (gyros, accelerometers, range sensors, GPS, etc.) in the sense that the true output for use in the feedback loop is usually not directly obtained from the sensor proper, but is extracted using a computer vision algorithm. This paper analyzes curve evolution approaches (known as *active contours* in computer vision) within the context of observer theory in order to arrive at a deterministic, recursive filtering strategy.

Filtering methodologies for computer vision may be divided in three broad (not necessarily mutually exclusive) subcategories:

(1) **Pre-filtering**: Direct filtering of the image information obtained from the vision sensors, followed by the application of a computer vision algorithm.

(2) **Internal-state-filtering**: Filtering the internal states associated with a computer vision algorithm, based on the image information obtained from the vision sensors.

(3) **Post-filtering**: Direct application of a computer vision algorithm to the data obtained from the vision sensors, followed by the filtering of the output of the computer vision algorithm.

To differentiate these methodologies, assume the objective is to track the centroid of a moving object given noisy image information from a vision sensor. For (1), the images are spatio-temporally filtered, the object is extracted, and the centroid is computed given the extracted object (the segmentation). For (3), the object is segmented from an image, the centroid is computed, and the centroid position is filtered given the centroids from previous image frames. For (2), the object itself is modeled dynamically (system states being position, shape, velocity, etc.), the object states are filtered based on the spatio-temporal image information of the vision sensor, and the centroid is extracted from the internal state of the modeled object.

*a) Observer Theory:* Observers are internal-state-filters. They are a classical concept in control- and estimation-theory [1], where system states need to be reconstructed from measurement data. They all have- but are not limited to- the following fundamental observer ingredients: **(O1)** a dynamical state model and a measurement model of the system to be observed for state and measurement **prediction**; **(O2)** a **measurement** methodology (e.g., a device to measure velocity, a thermometer, an object segmentation, etc.); and **(O3)** an **error correction** scheme to reconcile measurement

Marc Niethammer is with the Department of Computer Science, University of North Carolina, Chapel Hill. Patricio A. Vela and Allen Tannenbaum are with the School of Electrical and Computer Engineering, Georgia Institute of Technology.

Digital Object Indentifier 10.1109/TPAMI.2008.28

and prediction for state **estimation**. Irrespective of the observer similarities (O1)-(O3), observer approaches differ in terms of the system and measurement class for which they are designed, and the estimation method being employed. Observers may be deterministic, such as the Luenberger observer [2], or probabilistic, such as the Kalman filter [3] and its derivatives (the unscented Kalman filter, the extended Kalman filter), as well as particle filters [4].

Constructing observers for finite-dimensional nonlinear dynamical systems in complete generality (including noise processes) is difficult [5], [6]. The governing equations get complex even for systems with finite-dimensional state spaces [4], [7]. Observer design for infinite-dimensional (possibly nonlinear) systems is even more challenging and still in its infancy [8].

*b) Problem Description:* This paper focuses on the design of an observer for the infinite-dimensional space of closed curves[1]. To make the observation problem tractable, we propose here, as an initial step towards observer design on the space of curves, a deterministic observer, where the system model is time-continuous and the measurements are time-discrete. Special emphasis is given to a geometric formulation of the infinite-dimensional state observer. The proposed approach may be viewed as a geometric filtering method for the class of computer vision algorithms using curve evolutions (e.g., active contours, or snakes), where the fundamental observer building blocks (O1-O3) are interpreted in the context of dynamic curve evolution.

A fundamental issue that has not been fully explored is how one should resolve the curve when the curve measurements are not faithful to the truth. Essentially equivalent to this is the problem of interpreting the curve signal when the image properties imply unobservability or poor observability of the target. Typical solutions introduce complicated segmentation models or utilize low-dimensional, knowledge-based curve representations. Instead, this paper proposes a temporal filtering strategy for the static segmentations arising from an image sequence. The resulting geometric observer should generate a signal that has improved temporal and spatial properties of the curve proper that cannot be achieved through the sequential application of equivalent static curve evolution algorithms, and is robust with respect to unknown disturbances (e.g., measurement and system noise) and – especially as regards visual tracking – with respect to unmodeled dynamics.

*1) Intellectual precursors:* To impose spatial consistency, static segmentations frequently rely on expected shape, statistical, or appearance information to drive the segmentation process. Some curve-based filtering methodologies use spatially-consistent segmentations for each image frame [9]. Alternatively, dynamic segmentations arising from video sequences may incorporate temporal consistency: an expected object state will be conditioned on previous image measurements.

Bayesian estimation approaches have been central to a number of finite-dimensional curve-based tracking approaches. Blake and Isard [10] proposed a particle-filter based curve tracking framework for evolving the coefficients of a B-spline representation of a curve. More recent work has focused on combining particle filters with shape models, both for curve representations [11], [12], [13] as well as in combination with active shape models [14], [15]. Learned shape-based priors have been combined with jointly learned shape-dynamics using Gaussian probability distributions in [16]. Curve-tracking may also be formulated as a region-tracking problem [17].

While research in curve evolution based methods for segmentation has expanded to the point that there are a large variety of methods for achieving a desired segmentation[2] and a variety of finite-dimensional tracking methods exist (see paragraph above), curve evolution based filtering and tracking has received less attention. When online estimation is not required, the process may be regarded as a segmentation problem in space and time. Standard volumetric segmentation procedures can be used to perform tracking over the space-time volume, however, modeling temporal changes can still be beneficial for estimation performance. An example is the work by Papadakis and Mémin [18] for batch estimation using a data assimilation approach. The current focus is on online estimation methods to, ultimately, facilitate visual feedback control. Fundamental challenges for online curve-based filtering and tracking are the prediction (O1), and the error correction and estimation (O3) steps.

*2) Curve error correction:* To influence the state estimation by the measurements, one needs to define what constitutes a measurement and how measurements affect state estimation, leading to curve error correction. Error correction for curve-based observers, i.e., the construction of a corrected estimate of the current state given the predicted and measured states, is intimately linked to curve registration.

Blake and Isard [10] approach this problem in the

[1] The approach may be extended to objects of codimension one in space.

[2] Covering this diversity of approaches is beyond the scope of this text.

context of particle filtering, by defining the probability of a measurement conditioned on the system state. This conditional probability is based on the likelihood of image features along a pre-specified number of normals to the curve corresponding to candidates for the true measurement. Curve registration is implicitly encoded in the search along normal directions; corresponding points are expected to be found in a direction normal to the curve. Direct curve point correspondences are not explicilty established since the curve is represented by a low-dimensional B-spline space. Peterfreund [19] uses image gradient and optical flow measurements along the curve. The method does not compute correspondences, but relies on measurements computed at the current curve position. Rathi *et al.* [11] define measurement probabilities conditioned on the system state based on a curve evolution energy using the system state as its initial condition. Measurement and system dynamics are merged, creating spatial correspondences between curves by a curve evolution process that does not get run to steady state. Jackson *et al.* [20] use a registration approach, where a chosen similarity measure quantifies the difference between the estimated object and the actual image measurements over a finite number of image frames, subject to the degrees of freedom imposed on the registration by the motion group of choice.

In contrast, this paper is interested in establishing *explicit*, elastic, correspondences between curve points on the predicted, the measured, and the estimated curves. Further, this paper seeks to develop a non-parametric geometric observer framework for *dynamically evolving curves*, where every curve point is endowed with additional state information; the observer model requires a correction method for both curve position and the additional states. The primary difficulty associated with direct error correction between closed curves is the fact that two curves will, generically, not be identical, meaning that there is now a data association problem to be solved between the state information for the two curves, where the space of closed curves is infinite-dimensional and has a non-trivial topology [21], [22]. To be compatible with the observer framework proposed in this paper, a curve correspondence method should establish a homotopy between curves (assuming that curve parts are neither created or destroyed) to be used for curve interpolation. Prior work in this area includes methods based on defining metrics on the space of curves [23], [24], [25], [21], [22] and geometric distance based approaches, frequently combined with additional matching terms [26], [27], [28]. The class of matching methods based on defining metrics on the space of curves naturally results in curve interpolation paths. Correspondence schemes allowing for the deletion and addition of curve segments exist as well [29], [30]. Alternatively, curve correspondence computations may be performed by representing the curve by the inside and outside regions it separates and performing nonlinear registration using these regions [31], borrowing from the rich class of registration methods developed, in particular, for medical image registration problems, such as large deformation metric mappings [32] and optimal mass transport [33].

*3) Contribution:* The contributions of this paper are best understood when examined within the context of observer theory and design for infinite-dimensional nonlinear systems. The proposed observer incorporates a dynamical model of the computer vision algorithm's evolving states, here a closed curve and an associated vector fiber, and also a correction algorithm for moderating between the predicted and measured states. The consequences of this procedure are analogous to finite-dimensional systems. Appropriately designed, the observer will improve the active contour segmentation procedure while resulting in smoother state estimations and track signals.

The method proposed in this paper utilizes an infinite-dimensional motion model whose motion gets implicitly constrained by the measurement (which itself can be unconstrained, shape-constrained, area-based, etc.). Under the defined infinite-dimensional motion model, objects with changing topology are trackable; a global motion model describing the arbitrary splitting or merging of an object is not required. The present work also defines a method to establish one-to-one correspondences between the measured and the estimated curves in order to exchange information between them for dynamic filtering. The method is geometric in nature, as it involves the computation of a homotopy between two curves that, in turn, induces a mechanism for interpolating the state information. The geometric nature of the homotopy leads to intuitively tunable gains for position and state filtering. The gains are not derived from an underlying statistical model for the system uncertainties.

*4) Outline:* Section II briefly reviews curve evolution equations, related notation, and level set implementations. The overall specification of the observer framework is given in Section III, followed by a discussion of the individual components (O1)-(O3): the possible motion models of the system, Section IV, potential measurement models, Section V, and the correction step, Section VI. Examples to validate the claims are given in Section VII. Concluding remarks and possible further research directions follow in Section VIII.

## II. Curve and State Evolution

This section briefly outlines standard material from [34], [35], [36], to which we refer the reader for relevant details.

This paper uses smooth closed planar curves to represent image segmentations. The space of smooth closed planar curves, denoted by $C^\infty(S^1;\mathbb{R}^2)$, forms an infinite-dimensional manifold [34], meaning that there is no finite set of parameters which uniquely characterizes the space. Moreover, there is no unique description of a given element of such a space. When dealing with the evolution of curves, an additional temporal parameter is added to the curve description ($C^\infty(S^1 \times \mathbb{R}^+;\mathbb{R}^2)$), and so planar curve evolution may be described as the time-dependent mapping in the following manner: $\mathcal{C}(p,t) : S^1 \times [0,\tau) \mapsto \mathbb{R}^2$, where $p \in [0,1]$ is the curve's parametrization on the unit circle $S^1$, $\mathcal{C}(p,t) = [x(p,t),\ y(p,t)]^T$, such that $\mathcal{C}(0,t) = \mathcal{C}(1,t)$.

In this work, we will also use the level set methodology [35], [36]. Hence, we represent the one-parameter family of closed curves $\mathcal{C}$ implicitly by the family of level set functions $\Psi : \mathbb{R}^2 \times [0,\tau) \to \mathbb{R}$, where the trace of the curve on the domain is related to the zero level-set of the implicit function by

$$\Psi(0,t)^{-1} = trace(\mathcal{C}(\cdot,t)).$$

Many times, the level set function $\Psi$ is chosen to be a signed distance function, defined by

$$\begin{cases} \|\nabla\Psi\| = 1, \text{ almost everywhere}, \\ \Psi(\boldsymbol{x}) = 0, \ \forall \boldsymbol{x} \in \mathcal{C}, \\ \Psi(\boldsymbol{x}) < 0, \ \forall \boldsymbol{x} \in int(\mathcal{C}), \\ \Psi(\boldsymbol{x}) > 0, \ \forall \boldsymbol{x} \in ext(\mathcal{C}). \end{cases}$$

where $int(\mathcal{C})$ and $ext(\mathcal{C})$ denote the *interior* and *exterior*, respectively, of the curve $\mathcal{C}$.

Given a curve evolution equation

$$\mathcal{C}_t = \boldsymbol{v},$$

where $\boldsymbol{v}$ is a velocity vector and subscripts denote partial derivatives, the corresponding level set evolution equation is

$$\Psi_t + \boldsymbol{v}^T\nabla\Psi = 0,$$

where $\boldsymbol{v}$ is appropriately extended to be defined over the domain of $\Psi$ [36]. The unit inward normal, $\mathcal{N}$, and the signed curvature, $\kappa$, are given by

$$\mathcal{N} = -\frac{\nabla\Psi}{\|\nabla\Psi\|}, \quad \kappa = \nabla \cdot \frac{\nabla\Psi}{\|\nabla\Psi\|}.$$

### A. Curves with Vector Fibers

In order to define a dynamical model for curve evolution, the curve state space needs to be expanded to include curve velocities. The corresponding space is now a vector bundle [34], in particular, it is the tangent bundle. Briefly, a *vector bundle* is a family of vector spaces parameterized by a manifold (the *base space*), in our case $C^\infty(S^1;\mathbb{R}^2)$, the space of smooth closed planar curves. To each point of the manifold a vector space is associated (the *vector fiber* over that point), and these vector spaces vary smoothly. This means that locally the vector bundle is diffeomorphic to the cross product of a subspace of $C^\infty(S^1;\mathbb{R}^2)$, and a given model vector space, $W$. For a one-parameter family of curves $C^\infty(S^1 \times \mathbb{R}^+;\mathbb{R}^2)$, an element $\boldsymbol{w} \in W$ may be described as vector-valued function defined on $trace(\mathcal{C}(\cdot,t)), t \in [0,\tau]$. For the implicit representation of the family of curves $C^\infty(S^1 \times \mathbb{R}^+;\mathbb{R}^2)$ represented by the one-parameter family of level set functions $\Psi$ as above, the corresponding time-varying vector fiber element may be given by a family of functions, $\boldsymbol{w} : \mathbb{R}^2 \times [0,\tau) \to \mathbb{R}^2$, and thus we can obtain the curve velocities by evaluating $\boldsymbol{w}$ on $\Psi(0,t)^{-1}$ for each $t \in [0,\tau]$.

When defining the evolution or deformation of a curve by the vector $\boldsymbol{v}$, the transport of the fiber quantities with the curve must also be defined [34]. This is typically done by lifting the dynamics of the base space into the vector bundle. The lifted transport of the fiber element, $\boldsymbol{w}_0$, in the implicit representation is induced by the curve evolution through the advection equation,

$$\begin{cases} \boldsymbol{w}(\cdot,0) = \boldsymbol{w}_0, \\ \boldsymbol{w}_t + D\boldsymbol{w} \cdot \boldsymbol{v} = 0, \end{cases} \tag{1}$$

where $D\boldsymbol{w}$ denotes the Jacobian of $\boldsymbol{w}$, which is integrated in step with the level set $\Psi$ [36].

## III. General Observer Structure

In the classical observer framework, as described by Luenberger [2], there are prediction and measurement components. Prediction incorporates the dynamical assumptions made regarding the plant or, in the context of visual tracking, the movement of the object. To evolve the overall estimated curve, the prediction influence has to be combined with the measurement influence, leading to the correction step.

The observer to be defined is a continuous-discrete observer, i.e., the system evolves in continuous time with available measurements at discrete time instants $k \in \mathbb{N}_0^+$,

$$\begin{pmatrix}\mathcal{C}\\ \boldsymbol{w}\end{pmatrix}_t = \begin{pmatrix}\boldsymbol{v}(\mathcal{C},\boldsymbol{w},t,\boldsymbol{s}_1(t))\\ \boldsymbol{f}(\mathcal{C},\boldsymbol{w},t,\boldsymbol{s}_2(t))\end{pmatrix}, \qquad \text{and}$$

$$\boldsymbol{z}_k = \boldsymbol{h}_k\left(\begin{pmatrix}\mathcal{C}\\ \boldsymbol{w}\end{pmatrix}, \boldsymbol{m}_k\right),$$

where $\boldsymbol{s}_1$, $\boldsymbol{s}_2$ and $\boldsymbol{m}_k$ are the system and measurement noises respectively, $\mathcal{C}$ represents the curve position, $\boldsymbol{w}$ denotes additional states transported along with $\mathcal{C}$ (e.g., velocities), and $(\cdot)_k$ denotes quantities given at discrete time points $t_k$.

The addition of: **(1)** a prediction model for the active contour, **(2)** a measurement model for the active contour, and **(3)** a correction step to the evolution and measurement, form the general observer structure. The prediction and measurement models

$$\begin{pmatrix}\hat{\mathcal{C}}\\ \hat{\boldsymbol{w}}\end{pmatrix}_t = \begin{pmatrix}\hat{\boldsymbol{v}}(\hat{\mathcal{C}},\hat{\boldsymbol{w}},t)\\ \hat{\boldsymbol{f}}(\hat{\mathcal{C}},\hat{\boldsymbol{w}},t)\end{pmatrix}, \qquad \hat{\boldsymbol{z}}_k = \hat{\boldsymbol{h}}_k\begin{pmatrix}\hat{\mathcal{C}}\\ \hat{\boldsymbol{w}}\end{pmatrix}, \tag{2}$$

simulate the system dynamics and measurement process (the hat denotes simulated quantities).

For simplicity, consider the case when the complete state is measurable, $\boldsymbol{h}_k = \mathrm{id}$ (the identity map). The proposed continuous-discrete observer is

$$\begin{pmatrix}\hat{\mathcal{C}}\\ \hat{\boldsymbol{w}}\end{pmatrix}_t = \begin{pmatrix}\hat{\boldsymbol{v}}(\hat{\mathcal{C}},\hat{\boldsymbol{w}},t)\\ \hat{\boldsymbol{f}}(\hat{\mathcal{C}},\hat{\boldsymbol{w}},t)\end{pmatrix}, \qquad \hat{\boldsymbol{z}}_k = \begin{pmatrix}\hat{\mathcal{C}}_k\\ \hat{\boldsymbol{w}}_k\end{pmatrix}, \tag{3}$$

$$\begin{pmatrix}\hat{\mathcal{C}}_k(+)\\ \hat{\boldsymbol{w}}_k(+)\end{pmatrix} = \Phi\left(X_{err}; \begin{pmatrix}\hat{\mathcal{C}}_k(-)\\ \hat{\boldsymbol{w}}_k(-)\end{pmatrix}, \begin{pmatrix}\mathcal{C}_k\\ \boldsymbol{w}_k\end{pmatrix}; \begin{pmatrix}K_k^{\mathcal{C}}\\ \boldsymbol{K}_k^{\boldsymbol{w}}\end{pmatrix}\right)$$

where $(-)$ denotes the time just before a discrete measurement, $(+)$ the time just after the measurement, $\Phi$ is a correction function depending on the gain parameters $K_k^{\mathcal{C}}$ (a scalar) and $\boldsymbol{K}_k^{\boldsymbol{w}}$ (a matrix), and $(X_{err})$ is the error vector field. Figure 1 shows the observer structure as given in Equation (3). In what follows, these observer components as applied to closed curves are described in further detail.

## IV. Priors

The prediction model is a motion prior describing the evolution of the closed curve, and possibly its vector fiber, in time. Several motion priors are described in this section. Motion priors are problem dependent and should model as precisely as possible the dynamics of the object(s) to be tracked, consequently the given general-purpose priors should be substituted by more accurate priors when available. Priors should not depend on the current, underlying image (measurement) information. In this section, only dynamics of the curve proper are defined; the evolution of the implicit representation can be derived using the concepts from Section II.

### A. Curve Evolution

*1) Static Prior:* The simplest possible prior is the static prior, i.e., no motion at all,

$$\hat{\mathcal{C}}_t = 0.$$

*2) Constant Velocity Prior:* The next simplest prior would be the constant velocity prior,

$$\hat{\mathcal{C}}_{tt} = 0.$$

*3) Quasi-dynamic Prior:* Suppose that the richness of the target motion precluded an accurate dynamic model, but that an available instantaneous motion model, $X_{est}$, of the visual information existed. The instantaneous velocity information could be used to propagate forward the curve,

$$\hat{\mathcal{C}}_t = (X_{est} \circ \mathcal{C} \cdot \mathcal{N})\mathcal{N},$$

where $X_{est}$ is the estimated instantaneous velocity field. One example would be to use the optical flow vector field as a motion prior (see Section V-B for details on optical flow computations).

*4) Dynamic Elastic Prior:* The dynamic elastic prior is based on the dynamic active contour [37], which minimizes the action integral

$$\mathcal{L} = \int_{t=t_0}^{t_1} \int_0^1 \left(\frac{1}{2}\mu\|\hat{\mathcal{C}}_t\|^2 - a\right)\|\hat{\mathcal{C}}_p\| dp dt, \tag{4}$$

with $\mu$ a mass constant and $a$ a scalar regularization field. Whereas $a$ is image-dependent in the case of a dynamic active contour, it is image-*independent* for the dynamic elastic prior; $a$ is a design parameter for curve regularization. The resulting dynamic elastic prior is

$$\mu\hat{\mathcal{C}}_{tt} = -\mu(\mathcal{T}\cdot\hat{\mathcal{C}}_{ts})\hat{\mathcal{C}}_t - \mu(\hat{\mathcal{C}}_t\cdot\hat{\mathcal{C}}_{ts})\mathcal{T} - \left(\frac{1}{2}\mu\|\hat{\mathcal{C}}_t\|^2 - a\right)\kappa\mathcal{N} - (\nabla a\cdot\mathcal{N})\mathcal{N}$$

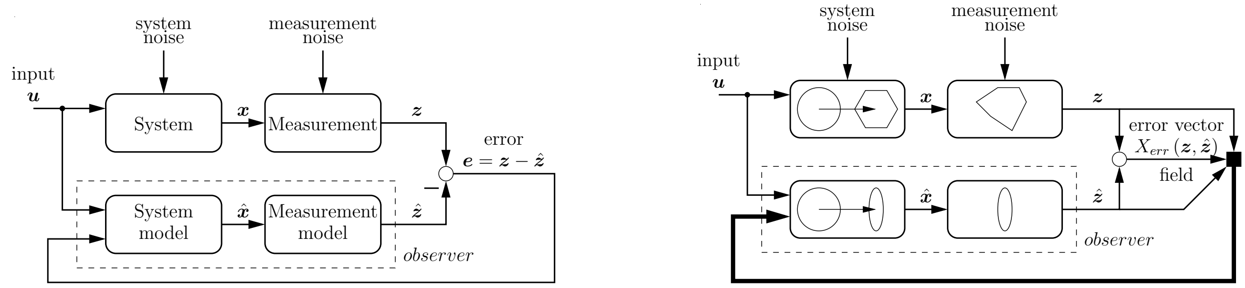
or, when restricted to normal curve propagation, is

$$\mu\hat{\mathcal{C}}_{tt} = \left(\frac{1}{2}\mu\|\hat{\mathcal{C}}_t\|^2 + a\right)\kappa\mathcal{N} - (\nabla a\cdot\mathcal{N})\mathcal{N} - \frac{1}{2}\mu(\|\hat{\mathcal{C}}_t\|^2)_s\mathcal{T}, \tag{5}$$

which simplifies to

$$\mu\beta_t = \left(\frac{1}{2}\mu\beta^2 + a\right)\kappa - (\nabla a\cdot\mathcal{N})$$

where $s$ denotes arc-length, $\mathcal{T}$ is the unit tangent vector to $\hat{\mathcal{C}}$, and $\beta$ is the normal speed in the direction of $\mathcal{N}$ [37].



(a) Classical Euclidean Observer Structure.

(b) Symbolic Observer Structure for Curves.

Fig. 1. *General Observer Structure.* In the Euclidean case, subtractions and additions are used for estimation error computations and state corrections. For curves, estimation errors are represented through an error vector field, relating the prediction to the measurement. The error vector field, $X_{err}$, defines a homotopy between the predicted and the measured curves.

### B. Vector Fiber Evolution

While only velocities get propagated with the curve in this paper, the dynamical model may include additional vector fiber states. These could for example be local estimates of state uncertainty or marker particles. A motion model would be needed for these additional states.

## V. MEASUREMENTS

Measurements are used to drive the observer model's states to the true system states. The predicted measurements are based on the current state of the observer. Standard computer vision algorithms can be used to come up with the "real" measurement. This observer setup has two crucial advantages:

- Any standard (static or dynamic) segmentation algorithm can be employed for the measurement. While the dynamic model is a model of a dynamically evolving curve, the measurement can utilize, for example, area-based or region-based segmentation algorithms.
- Static and dynamic approaches incorporating shape information exist [38], [39]. If these approaches are used for the measurement curve, shape information can be induced into the infinite dimensional model without the need for explicit incorporation of the shape information into the dynamical model.

### A. Position Measurements

A candidate segmentation method is the geodesic active contour

$$\mathcal{C}_t = g\kappa\mathcal{N} - (\nabla g \cdot \mathcal{N})\mathcal{N}, \tag{6}$$

where $\kappa$ denotes curvature, $\mathcal{N}$ is the unit inward normal and $g$ denotes a stopping function that is small at edges [40], [41]. Alternatively, one can use active contours without edges [42]

$$\mathcal{C}_t = \big(\mu\kappa - \nu - \lambda_1(u_0 - c_1)^2 + \lambda_2(u_0 - c_2)^2\big)\mathcal{N}, \tag{7}$$

where $\mu$ is a curvature gain, $\nu$ favors small curves, $\lambda_1$ and $\lambda_2$ are balancing constants for the curve interior and the curve exterior contributions, $u_0$ denotes image intensity and $c_1$ and $c_2$ are the mean image intensities inside and outside the curve respectively. Additional contour measurement options can be found by referencing the active contour segmentation literature.

### B. Measurement of Vector Fiber Quantities

Measurements of the vector fiber quantities are performed based on the location of the position measurements, if possible. In the case of dynamically evolving curves, velocities need to be measured on the measured curve. Given sufficient gradient information at the boundaries of the measured curve, normal optical flow can be computed as

$$\begin{pmatrix} u \\ v \end{pmatrix} = -I_t \frac{\nabla I}{\|\nabla I\|^2},$$

where $I : \mathbb{R}^2 \mapsto \mathbb{R}$ (usually image intensity). If there is insufficient gradient information, regularization may be performed to obtain the optical flow field reliably; one possibility being the computation of Horn and Schunck optical flow [43], minimizing

$$\mathcal{L}(I) = \lambda \iint (I_x u + I_y v + I_t)^2 \; dx \; dy \\ + \iint \left(\|\nabla u\|^2 + \|\nabla v\|^2\right) \; dx \; dy,$$

where $I$ denotes image intensity, subscripts denote partial derivatives, and $u$ and $v$ denote the velocities in $x$ and $y$ directions respectively.

## VI. Error Correction

The observer design proposed in this paper requires a methodology to compare the predicted curve configuration to the measured curve configuration. Minimally, comparison requires establishing unique correspondences between points on the two curves, e.g., defining a diffeomorphic homotopy between the two curves. The homotopy will be obtained through the flow of an error vector field defined between the two curves. Section VI-A constructs the error vector field. The curve correction homotopy naturally follows. Section VI-B discusses the propagation of state information along the error vector field and the implicit computation of signed distance functions. Section VI-C describes the correction procedure for the curve configuration (curve + fiber).

### A. The Error Vector Field

The error vector field to be defined is the manifold analogue to the measurement residual of an observer. Due to the geometry of the space of closed curves, there is no unique way to define the error vector field; its construction is a design choice when defining an observer for closed curves. Here, the method chosen is a Laplace equation based approach [44], [45], [46], whose error vector field induced flow is a diffeomorphism, which is easy to implement and fast to compute.

The variational formulation leading to the Laplace problem is

$$\underset{u}{min} \int \|\nabla u\|^2 \; d\Omega, \tag{8}$$

such that $trace(\mathcal{C}_0) = u^{-1}(0)$ and $trace(\mathcal{C}_1) = u^{-1}(1)$, where $\mathcal{C}_0$ is the source curve (the measurement curve) and $\mathcal{C}_1$ is the target curve (the predicted curve). Its solution requires careful construction of the interior and boundary conditions. The source curve and the target curve define the following solution domain decomposition of the total space $\Omega$, $R := \overline{int(\mathcal{C}_0) \ominus int(\mathcal{C}_1)}$, $R_{pi} := int(\mathcal{C}_0) \cap int(\mathcal{C}_1)$, and $R_{lo} := \Omega \setminus \left(\overline{R \cup R_{pi}}\right)$, where $int(\mathcal{C})$ denotes the interior of the curve $\mathcal{C}$ and $\ominus$ is the set-symmetric difference; see Figure 2(a) for a depiction of the domains.

To simplify the solution of (8) computationally, set the boundary conditions to 0 for the interior curve parts $(\partial\overline{R_{pi}} \setminus (\mathcal{C}_0 \cap \mathcal{C}_1))$ and to 1 for the exterior curve parts $(\partial\left(R \cup R_{pi}\right))$. The exterior and the interior curve parts may comprise of subsets of $\mathcal{C}_0$ *and* $\mathcal{C}_1$ if $\mathcal{C}_0$ and $\mathcal{C}_1$ intersect. These boundary conditions lead to a globally continuous solution, whose gradient field may have regions of reversed direction with respect to the gradient field of the original formulation (8). The reversed orientation is readily corrected to yield the proper gradient field; the quantity of interest for observer design purposes. Via the calculus of variations, a solution to (8) in the domain enclosed by the source and target curves *with modified boundary conditions* satisfies

$$\Delta u_s(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in R, \tag{9}$$

where $\Delta = \nabla^2$, with the boundary conditions

$$\begin{aligned} u_s(\boldsymbol{x}) &= 0, \; \boldsymbol{x} \in \partial\overline{R_{pi}} \setminus \left(\mathcal{C}_0 \cap \mathcal{C}_1\right), \\ u_s(\boldsymbol{x}) &= 1, \; \boldsymbol{x} \in \partial\left(R \cup R_{pi}\right), \end{aligned} \tag{10}$$

which is a simple reformulation of the minimization problem (8) based on the domain decomposition depicted in Figure 2. Of note, the solution of (10) is sufficient to define the error vector field. However, to facilitate easy numerical computations of the error vector field, we extend the solution to the remainder of the image domain[3], by solving an additional Laplace equation on $R_{lo}$ and a Poisson equation[4] on $R_{pi}$:

$$\Delta u_{pi}(\boldsymbol{x}) = c \quad \boldsymbol{x} \in \overline{R_{pi}}, \; c > 0 \tag{11}$$

$$\Delta u_{lo}(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in \overline{R_{lo}}, \tag{12}$$

with boundary conditions

$$\begin{aligned} &u_{pi}(\boldsymbol{x}) = 0, \; \boldsymbol{x} \in \partial\overline{R_{pi}}, \\ &u_{lo}(\boldsymbol{x}) = 1, \; \boldsymbol{x} \in \partial\left(R \cup R_{pi}\right), \\ &u_{lo}(\boldsymbol{x}) = 2, \; \boldsymbol{x} \in \partial\Omega. \end{aligned} \tag{13}$$

The combined solution

$$u(\boldsymbol{x}) = \begin{cases} u_{lo}(\boldsymbol{x}), & \boldsymbol{x} \in R_{lo}, \\ u_{pi}(\boldsymbol{x}), & \boldsymbol{x} \in R_{pi}, \\ u_s(\boldsymbol{x}), & \boldsymbol{x} \in R \end{cases} \tag{14}$$

defines the error vector field $X_{err}$ on $\Omega$,

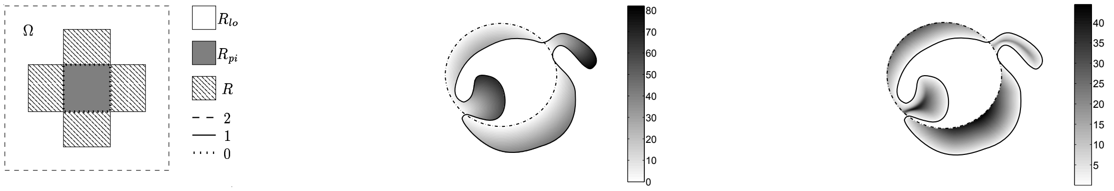
$$X_{err}(\boldsymbol{x}) := \begin{cases} \nabla u / \|\nabla u\|, & \boldsymbol{x} \in R, \\ \nabla u_o / \|\nabla u_o\|, & \boldsymbol{x} \in R_{lo}, \\ \nabla u_i / \|\nabla u_i\|, & \boldsymbol{x} \in R_{pi} \end{cases} \tag{15}$$

on $\Omega$ via the normalized gradient[5]. To illustrate the behavior of $X_{err}$ assume the measured curve is $\mathcal{C}_0$ and the predicted curve is $\mathcal{C}_1$, where $\mathcal{C}_0$ is strictly interior of $\mathcal{C}_1$. Then $X_{err}$ becomes a vector field flowing the

[3] Note, that the combined solution will be continuous everywhere, but not necessarily differentiable on $C_0$ and $C_1$. However, by construction the gradient directions will align.

[4] See Evans [47] for background on the Laplace equation and the Poisson equation.

[5] The essential part of the solution lies in $R$, defining solutions inside and outside of $R$ as described ensures monotonicity of the Laplace solution $u$ outside of $R$. The monotonicity is convenient for the later definition of the zero level set which otherwise would need to be restricted to lie within $R$.



(a) Solution domain decomposition. (b) Distance from estimate to measurement. (c) Distance from measurement to estimate.

Fig. 2. The topology and geometry of curve comparison.

measured curve into the predicted curve at unit speed. Flowing at unit speed implies that particles starting at $C_0$ and flowing according to $X_{err}$ will reach $C_1$ at different times (proportional to the distance covered), an essential property for the geometric interpolation procedure to be proposed in Section VI-B.

The advantage of the Laplace based correspondence scheme is that it is fast, parametrization-free, and admits topological changes. The main disadvantage is that it may lead to unwanted correspondences since it is not invariant to translations, rotations or scale. In particular curve intersection points remain fixed (i.e., they get identified with each other), which is unnatural in case of a translational motion. One may obtain the desired invariances by preceding the computations with a procedure for image registration based on similarity transformations (i.e., by working on the space of shapes). Nevertheless, sensible correspondences can be computed as long as $int(\mathcal{C}_0) \cap int(\mathcal{C}_1) \neq \emptyset$, otherwise source and target are in severe disagreement and a loss-of-track procedure needs to be employed (e.g., declaring the measurement as the estimate, leaving the estimate unchanged, or registering estimate and measurement). For visual tracking we can assume (given a reasonable motion model) that the displacements between curves will be small and thus the proposed approach will lead to reasonable correspondences.

### B. Information Transport and State Interpolation

The error vector field $X_{err}$ defined in Equation (15 of Section VI-A will be used to geometrically interpolate between two curves, thereby defining the curve correction homotopy $\Phi(X_{err}; \hat{\mathcal{C}}, \mathcal{C}; K^{\mathcal{C}})$ and also inducing a state correction homotopy $\Phi(X_{err}, (\hat{\mathcal{C}}, \hat{\boldsymbol{w}}), (\mathcal{C}, \boldsymbol{w}); K)$. Geometric interpolation is achieved by measuring the distance between correspondence points along the characteristics that connect them, defined by the error vector field, and subsequent flow up to a certain percentage of this distance. The error vector field will be used to define distance from a curve since Euclidean distances are not desirable for complicated curve shapes [38]. The procedure entails solving a series of associated transport equations (the transport equation was described in Section II-A).

Given a flow field $X$ and a particle $p$ initially located at $\boldsymbol{x}_0$, its traveling distance, $d$, at position $\boldsymbol{x}$ is defined as the arc-length of the characteristic curve, defined by having local tangents aligned with $X$, connecting $\boldsymbol{x}_0$ and $\boldsymbol{x}$. To measure traveling distances from a complete set of initial locations, as specified by $d^{-1}(\cdot, 0)$, solve

$$\begin{cases} d(\cdot, 0) = 0, \\ d_\tau + \frac{1}{\|X\|} X^T \nabla_x d = 1, \end{cases} \tag{16}$$

where $\tau$ is an artificial time parameter for the PDE, $X \neq 0$ is assumed and $d : \mathbb{R}^2 \times \mathbb{R}^+ \to \mathbb{R}$. Equation (16) is a Hamilton-Jacobi equation for which efficient numerical methods exist [48]. See Figure 2 for sample distance computations.

*1) Curve Correction Homotopy:* When $\hat{\Psi}$ and $\Psi$ implicitly represent the curves $\hat{\mathcal{C}}$ and $\mathcal{C}$, the interpolation can be accomplished implicitly to subpixel accuracy. To compute the traveling distances, the approach for level set reinitialization proposed in [49] will be modified. Specifically, to determine the traveling distance from each curve to the other along $X_{err}$ compute

$$\begin{aligned} \hat{d}_\tau + S(\hat{\Psi}) X_{err}^T \nabla_x \hat{d} = S(\hat{\Psi}), \qquad & \hat{d}(\boldsymbol{x}, 0) = \hat{\Psi}, \\ (d_m)_\tau + S(\Psi) X_{err}^T \nabla_x d_m = S(\Psi), \qquad & d_m(\boldsymbol{x}, 0) = \Psi, \end{aligned} \tag{17}$$

where $\hat{d}$ is the traveling distance associated with the predicted curve and $d_m$ is the same for the measured curve, $\tau$ is an artifical time parameter for the PDE, and

$$S(x) := \begin{cases} 0, & \text{if } \|x\| \leq 1, \\ \frac{x}{\sqrt{\epsilon + x^2}}, & \text{otherwise.} \end{cases}$$

Here, $S(\boldsymbol{x})$ denotes a smoothed sign function used for bidirectional measurement of traveling distance along $X_{err}$ (positive distance values) and in the opposite direction of $X_{err}$ (negative distance values). The results are signed distance level set functions whose distances conform to travel along the error vector field $X_{err}$. The distance error functions, $\hat{d}$ and $d_m$, obtained by solving Equation (17) are point-wise interpolated to yield the interpolated distance function $d_i$,

$$d_i = (1-\alpha)\hat{d} + \alpha d_m, \ \alpha \in [0,1], \tag{18}$$

which subsequently gets redistanced according to

$$(\hat{\Psi}_i)_\tau + S(\hat{\Psi}_i^0)\|\nabla_x \hat{\Psi}_i\| = S(\hat{\Psi}_i^0), \qquad \hat{\Psi}_i(\boldsymbol{x},0) = d_i, \tag{19}$$

arriving at the corrected distance function $\hat{\Psi}_i$, The weighting factor $\alpha$ geometrically interpolates the estimated and the measured curve. Several interpolations are illustrated in Figure 3(a).

*2) Vector Fiber Transport:* In order to compare and correct the vector fiber quantities, they need to be transported to the new interpolated curve, located within the region $R$. The measured quantities $\boldsymbol{w}$ and the estimated quantities $\hat{\boldsymbol{w}}$ utilize the corresponding transport equations

$$\begin{aligned} (\boldsymbol{p})_\tau + S(\Psi)X_{err}^T \nabla_x(\boldsymbol{p}) = 0, &\quad \boldsymbol{p}(\boldsymbol{x},0) = \boldsymbol{w}, \text{ and} \\ (\hat{\boldsymbol{p}})_\tau + S(\hat{\Psi})X_{err}^T \nabla_x \hat{\boldsymbol{p}} = 0, &\quad \hat{\boldsymbol{p}}(\boldsymbol{x},0) = \hat{\boldsymbol{w}}, \end{aligned} \tag{20}$$

respectively, to propagate the values throughout the domain, defining $\boldsymbol{p}, \hat{\boldsymbol{p}} : \mathbb{R}^2 \to W$.

### C. Performing the Error Correction

The error correction scheme builds on the results of Sections VI-A and VI-B. We assume that the correction function can be written as

$$\Phi\left(X_{err}; \begin{pmatrix} \hat{\mathcal{C}}_k(-) \\ \hat{\boldsymbol{w}}_k(-) \end{pmatrix}, \begin{pmatrix} \mathcal{C}_k \\ \boldsymbol{w}_k \end{pmatrix}; \begin{pmatrix} K_k^{\mathcal{C}} \\ \boldsymbol{K}_k^{\boldsymbol{w}} \end{pmatrix}\right) = \begin{pmatrix} \Phi^{\mathcal{C}}\left(X_{err}; \hat{\mathcal{C}}_k(-), \mathcal{C}_k; K_k^{\mathcal{C}}\right) \\ \Phi^{\boldsymbol{w}}\left(X_{err}; \begin{pmatrix} \hat{\mathcal{C}}_k(-) \\ \hat{\boldsymbol{w}}_k(-) \end{pmatrix}, \begin{pmatrix} \mathcal{C}_k \\ \boldsymbol{w}_k \end{pmatrix}; \boldsymbol{K}_k^{\boldsymbol{w}}\right) \end{pmatrix}.$$

The error correction for the curve position is then

$$\hat{\mathcal{C}}_k(+) = \Phi^{\mathcal{C}}\left(X_{err}; \hat{\mathcal{C}}_k(-), \mathcal{C}_k; K_k^{\mathcal{C}}\right), \tag{21}$$

which amounts to curve interpolation as per the implicit method described in Section VI-B.1,

$$trace(\hat{\mathcal{C}}_k(+)) = \hat{\Psi}_i(0)^{-1}, \text{ with } \alpha = K_k^{\mathcal{C}}.$$

Vector fiber information need to be exchanged and compared between the measured and the estimated curves (see Figure 3). Further, the final filtering results needs to be associated with $\hat{\mathcal{C}}_k(+)$. This is accomplished by the error correction for the vector fiber

$$\hat{\boldsymbol{w}}_k(+) = \Phi^{\boldsymbol{w}}\left(X_{err}; \begin{pmatrix} \hat{\mathcal{C}}_k(-) \\ \hat{\boldsymbol{w}}_k(-) \end{pmatrix}, \begin{pmatrix} \mathcal{C}_k \\ \boldsymbol{w}_k \end{pmatrix}; \boldsymbol{K}_k^{\boldsymbol{w}}\right)$$

which amounts to point-wise filtering for each component $i$ computed as

$$\begin{aligned} (\hat{\boldsymbol{w}}_i)_k(+) = (\hat{\boldsymbol{p}}_i)_k + (\boldsymbol{K}_{ij})_k^{\boldsymbol{w},\boldsymbol{w}}\left((\boldsymbol{p}_j)_k - (\hat{\boldsymbol{p}}_j)_k\right) \\ + (\boldsymbol{K}_i)_k^{\boldsymbol{w},\mathcal{C}}\left(d_m - \hat{d}\right) \end{aligned}$$

and evaluated at $\hat{\mathcal{C}}_k(+)$, where repeated indices are summed over and $\boldsymbol{K}_k^{\boldsymbol{w}}$ is assumed to be block-diagonal and decomposes into a gain matrix for the fiber quantities ($\boldsymbol{K}_k^{\boldsymbol{w},\boldsymbol{w}}$) and for the curve position error ($\boldsymbol{K}_k^{\boldsymbol{w},\mathcal{C}}$). Computation of $\boldsymbol{p}$ and $\hat{\boldsymbol{p}}$ was described in Section VI-B.2. Table I gives a description of the overall geometric observer algorithm. The choice of gain is problem dependent and should be done based on experimental evidence.

```
Geometric observer algorithm:
  repeat
    1) Propagate curve under the prediction model
    (Section IV) for the time-span between two image
    measurements (usually given by the camera frame rate).
    Initial conditions are given by the current observer state.

    2) Obtain curve measurements by image segmentation,
    optical flow, etc. (Section V).

    3) Reconcile internal observer state with the measure-
    ments by error correction (Section VI):
      a) Establish the error vector field (to induce corre-
      spondences; Section VI-A).
      b) Flow measurements and observer states along the
      error vector field (Section VI-B).
      c) Perform update of the internal observer state
      (Section VI-C).
  until end of tracking sequence
```

TABLE I

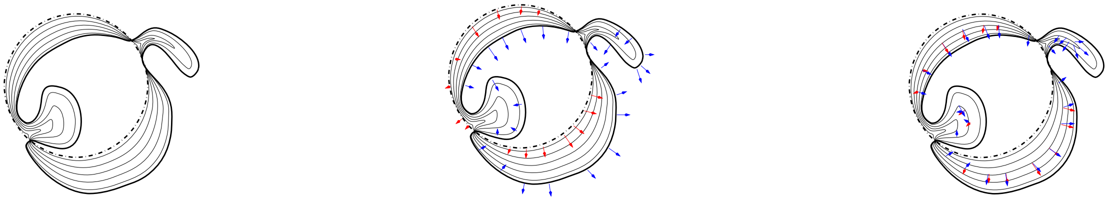
DESCRIPTION OF THE GEOMETRIC OBSERVER ALGORITHM.

## VII. Results

This section contains a variety of example scenarios utilizing the active contour to analyze video sequences. For each example, a performance metric is given to compare the standard implementation versus the observer-based implementation. It must be emphasized that for all of the sequences, the associated parameter settings for generating the measurements were precisely the same between the standard implementation and the observer-based implementation.

Every example is meant to highlight a specific property of the observer framework:

(A) The synthetic tracking example illustrates curve prediction, measurement, and estimation for an elastically deforming object with and without topology changes (Section VII-A).



(a) The dash-dotted curve gets geometrically interpolated to the solid curve.

(b) How does one compare the fiber quantities of the two curves?

(c) Comparison occurs after fiber quantities are pushed-forward to the same curve.

Fig. 3. Performing the curve and fiber correction.

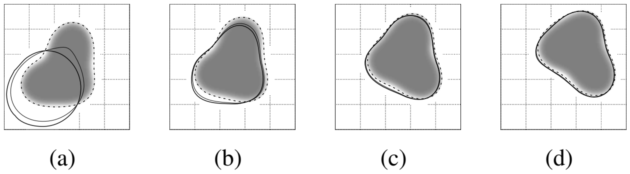


Fig. 4. Tracking of a single blob (Frames 0, 5, 10, and 15). The predicted (bold, solid), measured (dash-dotted), and corrected (solid) curves are depicted.

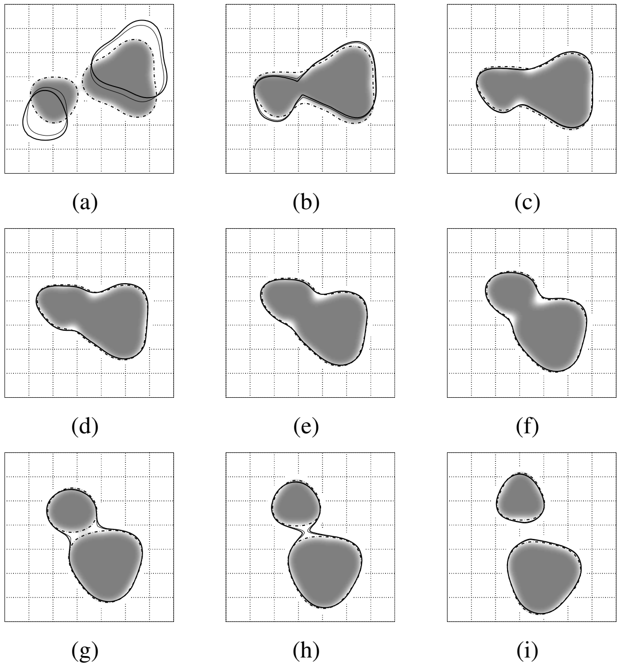


Fig. 5. Tracking of two blobs (Frames 0, 5, 10, 15, 20, 25, 30, 35, and 40). The predicted (bold, solid), measured (dash-dotted), and corrected (solid) curves are depicted.

(B) The tracking through turbulence example shows the utility of the observer framework to impose temporal coherence (Section VII-B).

(C) The bio-membrane tracking example illustrates the behavior of the dynamic elastic motion prior when the system becomes locally unobservable for a short time period (Section VII-C).

(D) The fish tracking example demonstrates the advantages of a curve-based filtering scheme over pre-filtering of images (Section VII-D).

(E) The aerial tracking example demonstrates how the observer-framework can be integrated with a particle-filter based sensing strategy (Section VII-E).

### A. Synthetic Tracking Example

To demonstrate proof-of-concept and to highlight some properties of the proposed observer, the observer is applied to the tracking of a synthetically created, elastically deforming blob.

The setup of the algorithm is as follows: **(1) Prior:** Due to the deformations of the blobs, the dynamic elastic prior was chosen with $\mu = 1$ and $a = 0.05$. **(2) Measurement:** The Chan and Vese region based evolution, Equation (7), generated the measurements. Horn and Schunck optical flow calculations generated the normal speed measurements. **(3) Correction:** The error correction gains were $K_k^{\mathcal{C}} = 0.25$, $K_k^{\boldsymbol{w},\mathcal{C}} = 0$, and $K_k^{\boldsymbol{w}} = 0.25$.

*Results:* Figures 4 and 5 show tracking results on a single blob and on two blobs respectively. Initial conditions (the bold solid curves) were chosen far from the initial measurement curves (dash-dotted curves). For the single blob case, the estimated curve (solid curve) converges nicely to the measurement curve over time. For the two blobs case, the estimated curve undergoes topological changes[6]; the curves merge and split naturally.

### B. Tracking Through Deep Turbulence

This example extends prior work on tracking of a high-speed projectile through deep turbulence using knowledge-based segmentation [50], where the objective is to precisely track a target being sensed by a laser-based imaging system. Due to the large distance traveled by the laser, turbulence is a primary source of uncertainty in

[6] Strictly speaking, the Laplace-equation correspondence approach will not be able to establish correspondences for a set of measure zero when topological changes occur and may require special computational handling for the critical points.

the visual signal. Turbulence introduces intensity fluctuations and minor random shifts of the target on the image plane. The authors' experiences are such that smoothing of the noisy image does not improve performance due to the non-deterministic effects of turbulence. Furthermore, smoothing of the track signal degrades performance or leads to negligible improvements (in the best of cases). The knowledge-based segmentation algorithm, which is an adaptive thresholding algorithm utilizing statistical analysis of the image together with Bayes' rule for tracking, has shown significant improvement in track fidelity while preserving bandwidth. The Bayesian step operates like an observer modulating the current likely measurement with the prior observation. Here we demonstrate that observer-based active contours may function in a manner similar to Bayesian segmentation.

The observer design comprises of the following procedures: **(1) Prior:** Because the system already has a coarse outer-loop tracker, the target is expected to lie within the vicinity of the image center, meaning that a static prior can be used. **(2) Measurement:** In order to replicate the statistical measurement model of the Knowledge-Based Segmentation algorithm, the measurements are based on the active contour analogue to likelihood maximization [51],

$$\mathcal{C}_t = \left(\frac{I \circ \mathcal{C} - \mu_1}{\sigma_1^2}\right)^2 - \left(\frac{I \circ \mathcal{C} - \mu_2}{\sigma_2^2}\right)^2 + \log\frac{\sigma_1}{\sigma_2} + \nu\kappa\mathcal{N},$$

where the means, $\mu_1$ and $\mu_2$, and standard deviations, $\sigma_1$ and $\sigma_2$, are computed based on the interior and exterior statistics, respectively, of the contour. **(3) Correction:** Since the model is a first-order static model, only the position gain is needed. It is $K_k^{\mathcal{C}} = 1/3$. **(4) Track point:** The track point for the target is found by utilizing a weighted centroid to find an interior point on the target. From this point, the contour segment in the direction of travel of the target is sought. A local average of this tip segment generates the track point (depicted in Figure 6).

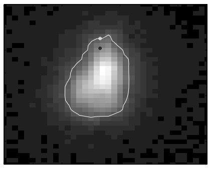


Fig. 6. Demonstration of tip-seeking for tracking.

*Results:* The results of applying the active contour measurement algorithm (denoted AC) and also the observer-augmented version (denoted ACO) to three image sequences are provided in Table II (each sequences is 2500 frames long). Two performance measures are indicated, the first being cross-correlation with the ground truth (higher is better), and the second being standard deviation of the error with respect to the ground truth (lower is better). The improvement percentage is with respect to the weighted centroiding algorithm (CENT), the currently accepted standard for this tracking problem. The observer framework improves the active contour performance by 30%, resulting in similar performance to the knowledge-based segmentation algorithm (KBS).

### C. Bio-Membrane Tracking.

The example scenario presented here involves the tracking of a bio- membrane. Imaging at the microscopic scale requires magnifying optics, resulting in a signal to noise ratio lower than that of traditional imaging systems. In addition to noisy imagery, the control objectives are in conflict with the observation objectives. In Figure 7, the red arrow points to a laser-controlled bead interacting with the bio-membrane (blue arrow). The interaction is used to estimate the mechanical properties of the bio-membrane, to control the location of the membrane, and/or to deform the membrane boundary shape [52], [53], [54], [55]. The act of measuring the cell membrane elasticity involves pressing up against the membrane with a manipulator, i.e., the laser-controlled bead, temporarily creating an obstruction, an unobservable component of the bio-membrane, and/or a disturbance of the bio-membrane.

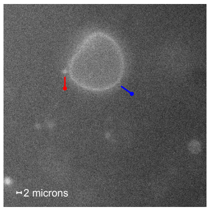


Fig. 7. Control bead and bio-membrane.

The observer setup for the bio-membrane sequence is as follows, **(1) Prior:** Because of the elasticity of the bio-membrane the normal dynamic elastic prior is used, Equation (5). **(2) Measurement:** For the position measurements, edge-based geodesic active contours are used, Equation (6), within a scale-space pyramid. Three averaging filters of differing support are applied to the image (5x5, 9x9, 15x15), smoothing it. Edge-based active contour evolution is applied to each of the smoothed images starting with the smoothest image and ending with the least smoothed. The velocity measurements are obtained from multi-grid optical flow calculations [56]. **(3) Correction:** The position correction gain is $K_k^{\mathcal{C}} = 0.5$ and the normal speed gain is $K_k^{\boldsymbol{w}} = 0.2$.

*Results:* Figure 8 has snapshots of the bio-membrane tracking sequence. The red contour is the active contour, whereas the blue contour is a manual segmentation using splines. Note that both are able to grossly track the bio-membrane, however in addition to tracking it, the curvature along the curve is required meaning that the contour must approximate well the bio-membrane while also being smooth. Figure 8(c) depicts the smoothness index[7]

[7] We define the smoothness index as $SI = \frac{1}{2\pi}\int |\kappa|\, ds - 1$, where $\kappa$ denotes curvature and $s$ arc-length.

| Algorithm | Axis | Run: 01 | 02 | 03 | Metric | Improvement |
|---|---|---|---|---|---|---|
| CENT | long | 0.831 | 0.838 | 0.827 | Cross Corr. | |
| | short | 0.975 | 0.972 | 0.969 | | |
| | long | 634 | 635 | 652 | Std. Dev. | |
| | short | 304 | 300 | 281 | | |
| KBS | long | 0.938 | 0.938 | 0.933 | Cross Corr. | |
| | short | 0.980 | 0.980 | 0.975 | | |
| | long | 374 | 399 | 371 | Std. Dev. | 40% |
| | short | 292 | 276 | 264 | | 6% |
| AC | long | 0.866 | 0.870 | 0.860 | Cross Corr. | |
| | short | 0.978 | 0.977 | 0.972 | | |
| | long | 535 | 533 | 542 | Std. Dev. | 16% |
| | short | 289 | 276 | 267 | | 6% |
| ACO | long | 0.936 | 0.938 | 0.930 | Cross Corr. | |
| | short | 0.979 | 0.978 | 0.973 | | |
| | long | 371 | 386 | 366 | Std. Dev. | 44% |
| | short | 289 | 275 | 265 | | 6% |

TABLE II
OPEN-LOOP TARGET TRACKING COMPARISON.

of the bio-membrane over time for three different contour signals, the standard (static) active contour segmentation, the observer-based active contour segmentation, and a manual segmentation. It is based on the computation of local absolute curvature around the closed contour; a perfect circle would result in a zero smoothness index. The observer-based segmentation has a lower curvature based smoothness index (mean of 1.8) when compared to the standard active contour (mean of 3.0), and also has a lower standard deviation (0.3 vs. 0.5 respectively), making it closer to the manual segmentation (mean of 1.2 and standard deviation of 0.2).

Using the temporal history of the signal leads to a smoother contour. Furthermore, the temporal history of the curve allows the observer-based active contour to be somewhat robust to disturbances. The disturbances come in the form of the bead proper, the image noise (mostly, but not entirely, smoothed out by the scale-space pyramid), and the blurring of the membrane on some frames. The blurring generates a shallow potential well for the geodesic active contour, which, in turn, leads to ambiguous minima for the active contour meaning descreased measurement smoothness. Although increasing the curvature-based component of the curve evolution would result in a smoother curve, the increased contribution of the curvature term overwhelms the image-based contribution to the flow during low observability leading to incorrect segmentations.

On frame 5, the proximity of the bead to the bio-membrane causes a disturbance in the measurement. From frame 25 to 27, the bead penetrates the bio-membrane (right side of middle row) causing a deformation in the bio-membrane boundary. After penetration, the interior motion of the bead continues to induce disturbances in the bio-membrane.

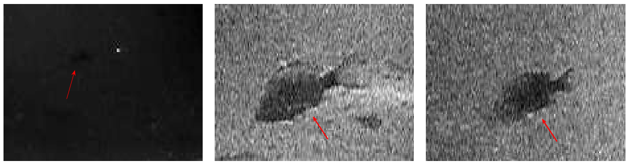
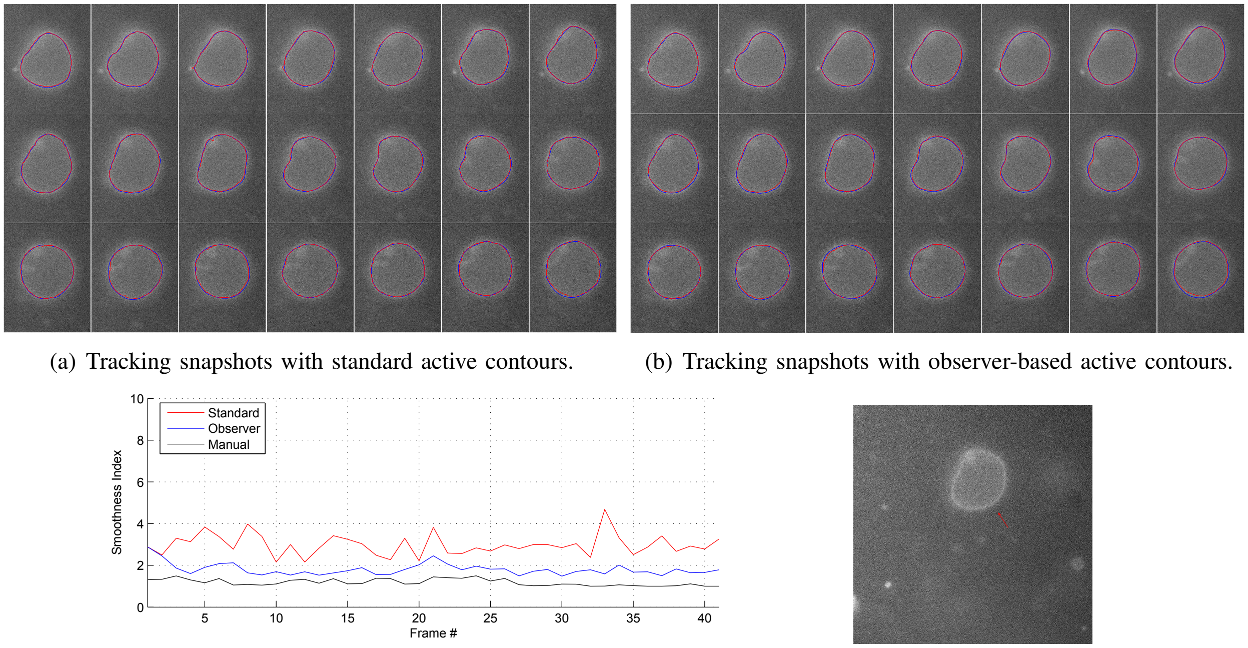


Fig. 9. Sample cropped portions of the aquarium sequences.

### D. Noisy and Low Observability Fish Tracking

One motivating factor for incorporating an observer into the active contour framework is to adequately deal with low observable imagery. Imaging sensors are designed to operate optimally within specific environmental conditions; deviation from these conditions leads to increased noise and/or reduce visibility of targets. Although the target may be visibly apparent by examining the temporal sequence, on any given frame the target may be incomplete or have misleading visual content. The image sequences of this example demonstrate such a situation; they were taken in low-light conditions and have significant noise. Examples of some of the issues associated with the image sequences can be found in Figure 9. The cropped samples suffer from one or more of the following: low contrast/poor visibility, pixel intensity values which cause it to blend into the background, missing information due to the sampling rate, or high noise.



(a) Tracking snapshots with standard active contours.

(b) Tracking snapshots with observer-based active contours.

(c) Smoothness index based on local curvature.

(d) Frame 20 of sequence.

Fig. 8. Comparison of standard versus observer-based tracking for bio-membrane sequence. The snapshots of the tracking sequence were taken every odd frame (1 to 41, from left to right). The blue curve in both snapshot series is the human generated contour, whereas the red curve is the appropriate active contour. The smoothness index of all three contour signals is also plotted.

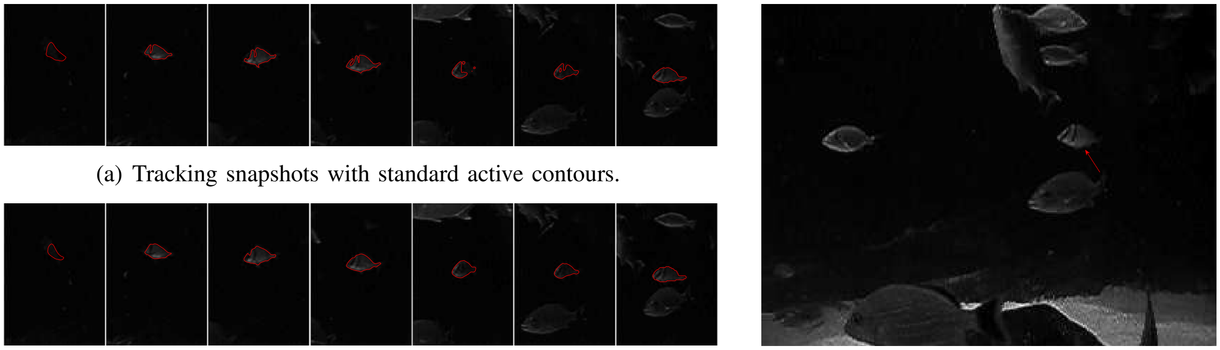
The observer setup for the aquarium sequences is as follows, **(1) Prior:** Because of the time-varying dynamics of the fish the normal dynamic model was chosen, Equation (5). **(2) Measurement:** For the position measurements, the Chan and Vese region based evolution was chosen, Equation (7). The velocity measurements are obtained from multi-grid Horn and Schunck optical flow calculations. **(3) Correction:** The position correction gain is $K_k^{\mathcal{C}} = 0.35$ and the normal speed gain is $K_k^{\boldsymbol{w}} = 0.35$.

*Results:* The Chan and Vese active contour algorithm was used to track all eight of the aquarium sequences, as was an observer-based implementation of the same algorithm. In both cases, the targeted fish are tracked throughout, however the performance differs between the two methods. Tracking performance characteristics of the two algorithms is done through visual inspection, analysis of the contour signal, and analysis of the resulting track points.

Figures 10 and 11 depict two tracking scenarios comparing the resulting contours to the actual visual observations; the fish swim from right to left, therefore the image sequences are depicted with frame number increasing from right to left. Both scenarios are difficult to segment on a per frame basis due to missing or low observable target information, as can be seen in Figures 10(a) and 11(a), where portions of the fish are not properly segmented. The observer-based tracker is able to utilize the temporal information to arrive at a more appropriate contour segmentation, Figures 10(b) and 11(b). Sequence 4 is difficult to track due to the time-varying blending of the target fish intensities with the background intensities. A similar problem occurs in sequence 5, which is exacerbated when a larger fish swims over the target fish causing a lowering of the overall illumination, and concomitantly reducing the observability of the fish (a blurred portion of the larger fish's tail can be seen in the third frame from the left). The observer is able to more adequately deal with the time-varying imaging characteristics.

Figures 12(a) and 12(b) contain the smoothness index of the contour signal and the smoothness index of the track point signal, respectively, for all aquarium sequences. As explored in the bio-membrane example, an added benefit of the observer framework is to result in smoother signals, both in terms of the segmenting contour and the associated track point. The first two sequences differ significantly from the remaining ones because the fish in them are about ten times smaller than the other fish (sequence 1 is depicted as the left-most sample in Figure 9).

As with the vesicle sequence, smoothing the contour using the curvature weight would result in a loss of measurement (this is slightly visible in Figure 11) Lowering the curvature term to capture more of the fish would cause the observation to be noisier, and in

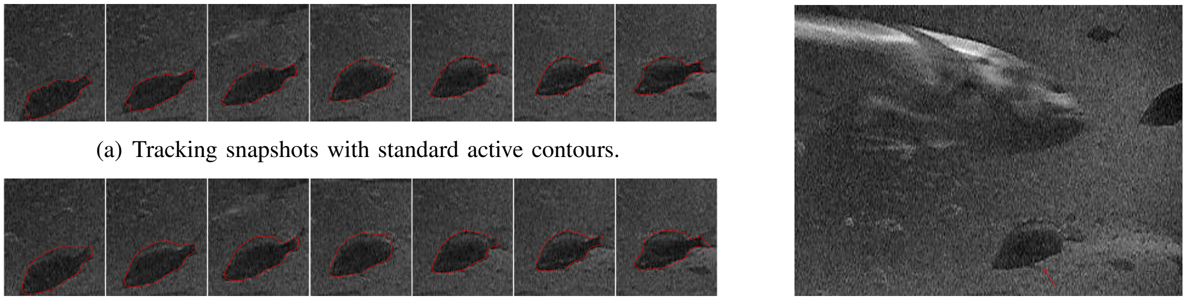


(a) Tracking snapshots with standard active contours.

(b) Tracking snapshots with observer-based active contours.

(c) Frame 25 of sequence 4.

Fig. 10. Comparison of standard versus observer-based tracking for aquarium sequence 4. On a frame per frame basis, there is not enough information to preserve the fish, but overall there is sufficient information to track.

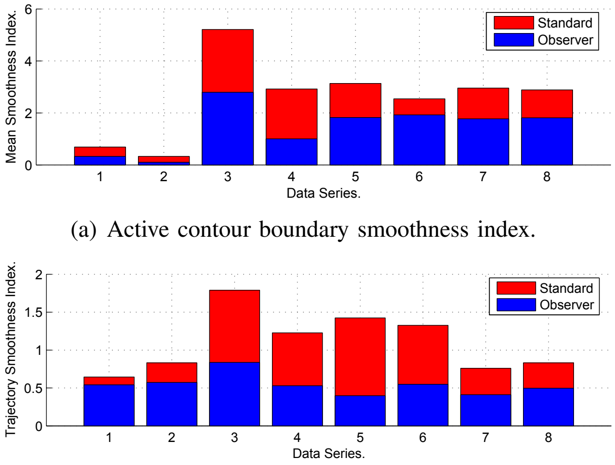


(a) Tracking snapshots with standard active contours.

(b) Tracking snapshots with observer-based active contours.

(c) Frame 25 of sequence 4.

Fig. 11. Comparison of standard versus observer-based tracking for aquarium sequence 5. The angle of the fish and the source of the lighting renders the top portion of the fish poorly observable. The observer-based contour more faithfully captures the geometry of the fish.



(a) Active contour boundary smoothness index.

(b) Track signal smoothness index.

Fig. 12. Contour signal and track signal smoothness indices for the aquarium series.

some cases would cause bleeding into the background of the active contour. Smoothing of the image is not possible since this would obliterate what little image-based information is available to the observer, further degrading the measurements. For example, examine the top portion of the middle fish in Figure 11; smoothing destroys the little intensity information available, which is providing enough information for the observer to track. It should also be noted that the sequences with similar image characteristics utilized the same parameters setting for the active contours and for the observer framework. In isolation, the parameter choice was not optimal on each sequence for the standard active contour, however, good tracking was possible for all sequences when the observer was introduced. Ideally, one would like the system to be adaptive and robust to imaging conditions. The observer-based framework confers some degree of robustness to the parameters settings, but does fail once the measurement model persistently fails.

### *E. Integrating with Particle Filters*

The final tracking scenario discusses the implementation of a particle filtering algorithm [57] for aerial tracking. A closer examination of the implementations described in [11], and summarized in Table III reveals that the particle filter is not a particle filter implementation in the traditional sense. Strictly speaking, the particle filter framework of [11] provides a mechanism to start at multiple initial guesses in order to arrive at the proper observation. The algorithm is more like RANSAC and

```
Particle sensing algorithm:
    for i = 1, . . . , $N$, do
        Draw states $x_0^{(i)}$ from prior $p(x_0)$.
    end
    for $k = 1, 2, \ldots ,$ do
        Propagate forward the samples under prediction model.
        Perform $L$ steps of curve evolution for each sample.
        Evaluate the importance weights.
        Normalize the weights.
        Select the sample with the largest weight to be the
          measurement.
        Resample from the particle distribution to obtain $N$
          particles with weights $\frac{1}{N}$.
    end
```

TABLE III

Description of the particle sensing algorithm of [11].

other randomized strategies that seek to determine a measurement or estimate based on a search over a subset of the true search domain/dataset. In the flight scenario, the unknown camera motion results in random translations of the target on the image plane (justifying the authors' use of only translations in the described particle framework). Due to the random shifts, the initial guess for the active contour gradient descent need not lie within the potential well of the true observation. Consequently, the particles act as random initial guesses for gradient descent; at least one of the particles should be within the vicinity of the potential well associated with the energy to minimize. Choosing a sufficient amount of particles will lead to high confidence in the occurance of this event.

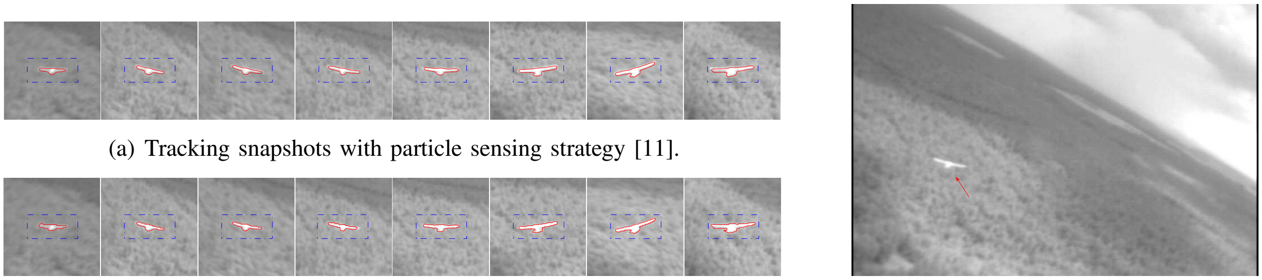
In deriving the final estimated observation, there is no predicted model to compare against. The authors therefore give special care to the post-processing of the samples to ensure that the observation strategy does not overly rely on the image information; the sensing strategy limits the number of iterations in the gradient descent procedure for active contours and does not go to steady state. This can lead to erroneous results during measurement model failure, as there is no natural way of knowing *a priori* what the optimal number of iterations should be, nor whether this will be constant for a given image sequence. Strictly speaking then, the strategy proposed in [11] is a particle sensor; it uses a collection of particles to infer the best current measurement.

In the observer-based strategy proposed here, the contour goes to steady state. If the initial gradient search location is incorrect, the contour will vanish (either by shrinking to a point, or blowing up to incorporate the entire sensing window), allowing that particle to be rejected. The best measurement is compared against the predicted measurement and corrected to arrive at the final observation. Having a predicted dynamic model completely independent of the particle measurements provides robustness to measurement error, while also minimizing measurement error since the particles are now allowed to reach steady-state. The proposed algorithm is as follows, **(1) Prior:** In order to capture the rolling moments of the target plane, a constant velocity model was used. **(2) Measurement:** The position measurements were obtained using the particle sensing strategy found in Table III, except that the active contour evolution is allowed to reach steady-state. **(3) Correction:** The position correction gain is $K_k^{\mathcal{C}} = 0.45$ and the normal speed correction gain using positional error information is $K_k^{\boldsymbol{w},\mathcal{C}} = 0.001$ (with $K_k^{\boldsymbol{w},\boldsymbol{w}} = 0$).

*Results:* The two algorithms were applied to three image sequences; see Figures 13 to 15. In the figures, the blue dash-dotted box depicts the sensing window of the winning particle and the red contour/lines depict the foreground/background boundaries segmenting the target from the background. Both algorithms utilized fifteen particles for all sequences. In the first sequence, both are able to track through the entire sequence. In the second sequence, there is a piece of dirt on the lens introducing an imaging disturbance that the particle sensor captures when the target enters the region of the image plane where the dirt lies. It is a minor disturbance, but does introduce an error in the tracking signal. In the last sequence, there is large interframe motion leading to motion blur, most likely a wind-induced disturbance on the following plane. The motion blur causes problems for the particle sensor algorithm due to the limited amount of allowed evolution and the lack of comparison against a predicted model that is independent of the imagery. For the particle sensor, an incorrect patch of the image results in a lower energy and is thus incorrectly classified as the target. Because of the limited number of gradient descent iterations, some final results may appear to be valid when in fact they are not. In contrast, the proposed particle filtering model is able to attenuate the disturbance and does not lose track.

## VIII. Conclusions and Future Work

*a) Conclusions:* This paper proposed a geometric design methodology for observing or filtering dynamically evolving curves over time. State measurements are performed via static measurement methods, making the framework flexible and powerful, since any relevant technique may be used for the measurement step, including methods incorporating shape information. In comparison to most previous approaches, the model is infinite-dimensional. Further, the observer framework is geometric, leading to geometrically meaningful observer

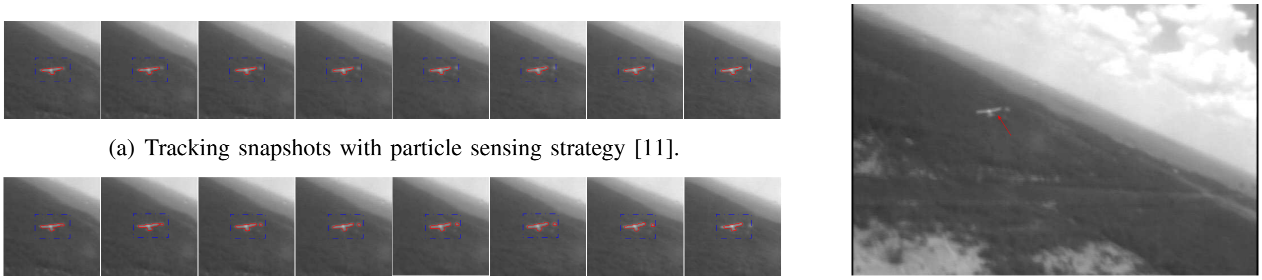


(a) Tracking snapshots with particle sensing strategy [11].

(b) Tracking snapshots with proposed particle filtering strategy.

(c) Frame 40 of flight sequence 1.

Fig. 13. Comparison of particle sensing strategy versus the proposed particle filtering strategy for flight sequence 1. On standard imagery, they have equivalent performance with the difference that the observer-based strategy will have a smoother contour.

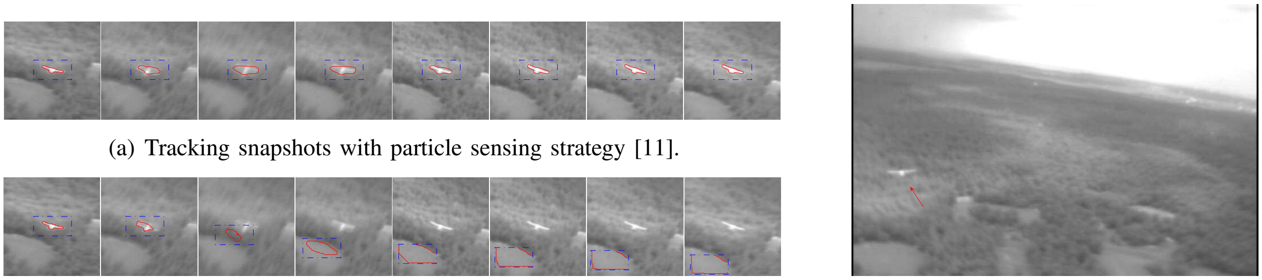


(a) Tracking snapshots with particle sensing strategy [11].

(b) Tracking snapshots with proposed particle filtering strategy.

(c) Frame 77 of flight sequence 2.

Fig. 14. Comparison of particle sensing strategy versus the proposed particle filtering strategy for flight sequence 2. The dirt speck on the lens is captured by the particle sensor method, whereas the proposed particle filter method rejects it.



(a) Tracking snapshots with particle sensing strategy [11].

(b) Tracking snapshots with proposed particle filtering strategy.

(c) Frame 90 of flight sequence 3.

Fig. 15. Comparison of particle sensing strategy versus the proposed particle filtering strategy for flight sequence 3. The camera experiences motion blur on the captured imagery. The particle sensor cannot track through this temporary model failure.

gains. For the sake of simplicity in our presentation, this paper was restricted to dynamically evolving curves (see Section II), however, the overall scheme extends to closed hypersurfaces of codimension one.

To assess the behavior of the observer framework in comparison to different tracking strategies and scenarios, it was applied to a variety of tracking examples, highlighting different aspects of the methodology. The examples showed the utility of curve-based temporal filtering in the presence of noise, for low observability, and in combination with a particle-based sensing strategy.

*b) Future Work:* The positive results obtained in this investigation suggest several avenues to explore regarding the control-theoretic analysis of computer vision algorithms for performance and robustness enhancement. In particular, alternative methods for obtaining image measurements and establishing state-correspondences (in the case of this paper, a homotopy between curves) across time. Unlike for finite-dimensional systems, there is no longer a unique comparison and correction method, therefore this step becomes a part of the observer design process. Secondly, a principal omission in the currently proposed observer scheme is the lack of statistical information, i.e., a “measure-of-trust” for measurements and observed states. Introducing statistical information will require the notion of a mean shape of curves and curve covariances [28]. Future work will introduce adaptive observer gains based on curve statistics.

Most of the sequences contained only about 100-200 frames, which is quite low when considering the amount of time one would like to track. This, unfortunately, was a consequence of the image sequences available for testing. We expect to generate longer video sequences to assess the long-term robustness and stability of the observer framework. Furthermore, since the examples and analysis in this paper were open-loop, i.e., prerecorded imagery, investigating closed-loop scenarios to validate and extend the results obtained will be part of future research. This entails the active, real-time use of the observer output for closed-loop tracking.

## Acknowledgment

This work was supported in part by grants from NSF, AFOSR, ARO, MURI, MRI-HEL as well as by a grant from NIH (NAC P41 RR-13218) through Brigham and Women's Hospital. This work is part of the National Alliance for Medical Image Computing (NAMIC), funded by the National Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 EB005149. Information on the National Centers for Biomedical Computing can be obtained from `http://nihroadmap.nih.gov/bioinformatics`.

## References

[1] A. Gelb, Ed., *Applied Optimal Estimation*, 15th ed. MIT Press, 1999.

[2] D. G. Luenberger, "An introduction to observers," *IEEE Transactions on Automatic Control*, vol. 16, no. 6, pp. 596–602, 1971.

[3] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[4] M. Arulampalam, S. Maskell, N. Gordon, T. Clapp, D. Sci, T. Organ, and S. Adelaide, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transaction on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[5] S. K. Mitter, "Filtering and stochastic control: A historical perspective," *IEEE Control Systems Magazine*, vol. 16, no. 3, pp. 67–76, 1996.

[6] S. V. Lototsky, "Problems in statistic of stochastic differential equations," Ph.D. dissertation, University of Southern California, 1996.

[7] R. N. Miller, E. F. Carter, and S. T. Blue, "Data assimilation into nonlinear stochastic models," *Tellus A*, vol. 51, pp. 167–194, 1999.

[8] A. V. Wouwer and M. Zeitz, *Control Systems, Robotics and Automation, Theme in Encyclopedia of Life Support Systems*. EOLSS Publishers, 2001, ch. State estimation in distributed parameter systems.

[9] A. Yilmaz, X. Li, and M. Shah, "Object contour tracking using level sets," 2004.

[10] A. Blake and M. Isard, *Active Contours*. Springer Verlag, 1998.

[11] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi, "Tracking Deforming Objects Using Particle Filtering for Geometric Active Contours," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 8, pp. 1470–1475, 2007.

[12] Y. Rathi, N. Vaswani, and A. Tannenbaum, "A generic framework for tracking using particle filter with dynamic shape prior," *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1370–2007, 2007.

[13] N. Vaswani, A. Yezzi, Y. Rathi, and A. Tannenbaum, "Time-varying finite dimensional basis for tracking contour deformations," in *Proceedings of the Conference on Decision and Control (CDC)*, 2006, pp. 1665–1672.

[14] S. K. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Transactions on Image Processing*, vol. 13, no. 11, pp. 1491–1506, 2004.

[15] F. Dornaika and F. Davoine, "On appearance based face and facial action tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 9, pp. 1107–1124, 2006.

[16] D. Cremers, "Dynamical statistical shape priors for level set-based tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1262–1273, 2006.

[17] A.-R. Mansouri, "Region tracking via level set PDEs without motion computation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 947–961, 2002.

[18] N. Papadakis and E. Mémin, "Variational optimal control technique for the tracking of deformable objects," in *Proceedings of the International Conference on Computer Vision*. IEEE, 2007.

[19] N. Peterfreund, "Robust tracking of position and velocity with Kalman snakes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, pp. 564–569, 1999.

[20] J. D. Jackson, A. J. Yezzi, and S. Soatto, "Tracking deformable moving objects under severe occlusions," in *Proceedings of the Conference on Decision and Control*. IEEE, 2004.

[21] P. W. Michor and D. Mumford, "Riemannian geometries on spaces of plane curves," http://arxiv.org/abs/math/0312384.

[22] A. Yezzi and A. Mennucci, "Conformal metrics and true "gradient flows" for curves," in *Proceedings of the International Conference on Computer Vision*, 2005, pp. 913–919.

[23] L. Younes, "Computable elastic distances between shapes," *SIAM Journal on Applied Mathematics*, vol. 58, no. 2, pp. 565–586, 1998.

[24] ——, "Optimal matching between shapes via elastic deformations," *Image and Vision Computing*, vol. 5-6, pp. 381–389, 1999.

[25] E. Klassen, A. Srivastava, W. Mio, and S. H. Joshi, "Analysis of planar shapes using geodesic paths on shape spaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 3, pp. 372–383, 2004.

[26] I. Cohen, N. Ayache, and P. Sulger, "Tracking points on deformable objects using curvature information," INRIA, Tech. Rep. 1595, 1991.

[27] R. Basri, L. Costa, D. Geiger, and D. Jacobs, "Determining the similarity of deformable shapes," *Vision Research*, vol. 38, pp. 2365–2385, 1998.

[28] G. Charpiat, O. Faugeras, and R. Keriven, "Approximations of shape metrics and application to shape warping and empirical shape statistics," *Foundations of Computational Mathematics*, pp. OF1–OF58, 2004.

[29] H. D. Tagare, D. O'Shea, and D. Groissier, "Non-rigid shape comparison of plane curves in images," *Journal of Mathematical Imaging and Vision*, vol. 16, no. 1, pp. 57–68, 2002.

[30] T. B. Sebastian, P. N. Klein, and B. B. Kimia, "On aligning curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 1, pp. 116–124, 2003.

[31] M. I. Miller and L. Younes, "Group actions, homeomorphisms, and matching: A general framework," *International Journal of Computer Vision*, vol. 41, pp. 61–84, 2001.

[32] M. F. Beg, M. I. Miller, A. Trouvé, and L. Younes, "Computing large deformation metric mappings via geodesic flows of diffeomorphisms," *International Journal of Computer Vision*, vol. 61, no. 2, pp. 139–157, 2005.

[33] S. Angenent, S. Haker, and A. Tannenbaum, "Minimizing flows for the Monge-Kantorovich problem," *SIAM Journal of Mathematical Analysis*, vol. 35, pp. 61–97, 2003.

[34] R. Abraham, J. E. Marsden, and R. Ratiu, *Manifolds, tensor analysis, and applications: 2nd edition*. New York, NY, USA: Springer-Verlag New York, Inc., 1988.

[35] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, ser. Applied Mathematical Sciences. Springer-Verlag, 2003, vol. 153.

[36] J. Sethian, *Level Sets Methods and Fast Marching Methods*. Cambridge University Press,, 1999.

[37] M. Niethammer, A. Tannenbaum, and S. Angenent, "Dynamic active contours for visual tracking," *IEEE Transactions on Automatic Control*, vol. 51, no. 4, pp. 562–579, 2006.

[38] A. Yezzi and S. Soatto, "Deformotion: Deforming motion, shape average and the joint registration and approximation of structures in images," *International Journal of Computer Vision*, vol. 53, no. 2, pp. 153–167, 2003.

[39] D. Cremers and S. Soatto, "A pseudo-distance for shape priors in level set segmentation," in *Proceedings of the International Workshop on Variational, Geometric and Level Set Methods in Computer Vision*. IEEE, 2003, pp. 169–176.

[40] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi, "Conformal curvature flows: From phase transitions to active vision," *Arch. Rat. Mech. Anal.*, vol. 134, pp. 275–301, 1996.

[41] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *Int. J. Comp. Vis.*, vol. 13, pp. 5–22, 1997.

[42] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[43] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Atificial Intelligence*, vol. 23, pp. 185–203, 1981.

[44] E. Pichon, D. Nain, and M. Niethammer, "A Laplace equation approach for the validation of image segmentation," in preparation.

[45] A. Duci, A. J. Yezzi, S. K. Mitter, and S. Soatto, "Shape representation via harmonic embedding," in *Proceedings of the International Conference on Computer Vision*, 2003, pp. 656–662.

[46] A. Yezzi and J. L. Prince, "A PDE approach for measuring tissue thickness," in *CVPR*, vol. 1, 2001, pp. 87–92.

[47] L. Evans, *Partial Differential Equations*. American Mathematical Society, 1998.

[48] C. Y. Kao, S. Osher, and Y.-H. Tsai, "Fast sweeping methods for static Hamilton-Jacobi equations," UCLA, Tech. Rep. 03-75, 2003.

[49] M. Sussman, P. Smereka, and S. Osher, "A level set approach for computing solutions to incompressible two-phase flow," *Journal of Computational Physics*, vol. 114, pp. 146–159, 1994.

[50] P. A. Vela, M. Niethammer, G. D. Pryor, A. R. Tannenbaum, R. Butts, and D. Washburn, "Knowledge-based segmentation for tracking through deep turbulence," to appear in the *IEEE Transactions on Control Systems Technology*, 2007.

[51] M. Rousson and R. Deriche, "A variational framework for active and adaptive segmentation of vector valued images," in *IEEE Workshop on Motion and Video Computing*, 2002.

[52] J. Li, M. Dao, C. Lim, and S. Suresh, "Spectrin-level modeling of the cytoskeleton and optical tweezers stretching of the erythrocyte," *Biophysical Journal*, vol. 88, no. 5, pp. 3707–3719, 2005.

[53] Y. Tseng, J.-H. Lee, I. Jiang, T. Kole, and D. Wirtz, "Micro-organization and visco-elasticity of the interphase nucleus revealed by particle nanotracking," *Journal of Cell Science*, vol. 117, no. 10, pp. 2159–2167, 2004.

[54] G. Dong, N. Ray, and S. Acton, "Intravital leukocyte detection using the gradient inverse coefficient of variation," *IEEE Trans. Med. Imag.*, vol. 24, no. 7, pp. 910 – 924, 2005.

[55] S. Suresh, J. Spatz, J. Mills, A. Micoulet, M. Dao, C. Lim, M. Beil, and T. Seufferlein, "Connections between single-cell biomechanics and human disease states: Gastrointestinal cancer and malaria," *Acta Biomaterialia*, vol. 1, p. 1630, 2005.

[56] W. Enkelmann, "Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences," *Computer Vision, Graphics, and Image Processing*, vol. 43, no. 2, pp. 150–177, 1988.

[57] A. Doucet, Ed., *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

**Marc Niethammer** received his Diplom-Ingenieur in Engineering Cybernetics from the Universität Stuttgart, Germany in 2000. He earned a Masters of Science in Engineering Science and Mechanics, a Masters of Science in Applied Mathematics, and a Ph.D. in Electrical and Computer Engineering from the Georgia Institute of Technology in 1999, 2002, and 2004 respectively. After working in medical imaging as a postdoctoral research fellow and subsequently instructor in the Psychiatry Neuroimaging Laboratory, at Brigham and Women's, Harvard Medical School, he joined the Department of Computer Science at the University of North Carolina, Chapel Hill, as an assistant professor in 2008. His research interests include computer vision, medical image analysis, and control- and estimation-theory.

**Patricio A. Vela** earned his B.S. in 1998 and his Ph.D. in 2003 at the California Institute of Technology in the area of geometric nonlinear control. After working as a post-doctoral researcher in computer vision at the Georgia Institute of Technology, he joined the School of Electrical and Computer Engineering of the Georgia Institute of Technology as an assistant professor in 2005. His research interests lie in the geometric perspectives to control theory and computer vision.

**Allen Tannenbaum** attended Columbia University where he received his B.A. from Columbia in 1973, and his Ph.D. from Harvard University in mathematics in 1976. He has been a faculty member at the Georgia Institute of Technology since 1999. He has played a leading role in developing new mathematical techniques for various engineering problems in systems and control, vision, signal processing, and cryptography.