Deform PF-MT: Particle Filter With Mode Tracker for Tracking Nonaffine Contour Deformations

Namrata Vaswani, Yogesh Rathi, Anthony Yezzi, and Allen Tannenbaum

***Abstract*—We propose algorithms for tracking the boundary contour of a deforming object from an image sequence, when the nonaffine (local) deformation over consecutive frames is large and there is overlapping clutter, occlusions, low contrast, or outlier imagery. When the object is arbitrarily deforming, each, or at least most, contour points can move independently. Contour deformation then forms an infinite (in practice, very large), dimensional space. Direct application of particle filters (PF) for large dimensional problems is impractically expensive. However, in most real problems, at any given time, most of the contour deformation occurs in a small number of dimensions ("effective basis space") while the residual deformation in the rest of the state space ("residual space") is small. This property enables us to apply the particle filtering with mode tracking (PF-MT) idea that was proposed for such large dimensional problems in recent work. Since most contour deformation is low spatial frequency, we propose to use the space of deformation at a subsampled set of locations as the effective basis space. The resulting algorithm is called deform PF-MT. It requires significant modifications compared to the original PF-MT because the space of contours is a non-Euclidean infinite dimensional space.**

I. Introduction

OUR goal is to causally segment moving and deforming object(s) from a sequence of images. This is formulated as the problem of sequentially estimating the boundary contour of the object, i.e., computing an optimal Bayesian estimate of the state (contour and contour velocity) at the current time using all observations (images) until the current time. This is referred to as "tracking". Any optimal state estimate, e.g., minimum mean squared error (MMSE) or maximum *a posteriori* (MAP), can be computed once the posterior is approximated. The state dynamics is assumed to be Markovian. The observed image is a noisy and possibly nonlinear function of the contour. The observation likelihood (image likelihood given the contour) is often multimodal or heavy tailed as a function of the contour.[1] This can be due to clutter, i.e., due to background objects which are partially occluded by the object of interest (Fig. 1); due to an object which partially occludes the object of interest (Fig. 6); due to low contrast imagery (see Fig. 7 and [2]); or due to outlier noise (Fig. 2). Since the observation likelihood is nonlinear and often multimodal, and since the state dynamics is nonlinear, due to the state space being non-Euclidean, the exact posterior cannot be computed. We study particle filtering [3]–[5] solutions to the tracking problem.

If the motion of the object is constrained, the contour motion can be efficiently represented by a small number of parameters, e.g., the affine group [6], [7]. However, if the object is arbitrarily deforming, each contour point can move independently. Contour deformation then forms an infinite (in practice, very large), dimensional space. Deforming contours occur either due to changing regions of partial occlusions [e.g., Fig. 6(b)] or when the object of interest is actually deforming its shape over a time or space sequence of images, e.g., beating heart, moving animals or humans, or the cross sections of different parts of a 3-D object like the brain, in consecutive MRI slices (Fig. 7). Most biological images contain deforming objects/regions. Contour tracking has many applications in medical image analysis, e.g., sequential segmentation (Fig. 7); tracking heart regions [2] or image guided surgery [8].

Early work on contour tracking used the Kalman filter to track a fixed number of marker points [9]–[11] or a fixed parametric representation, such as B-spline control points [12]. A Kalman filter for a continuous contour was proposed in [13]. The seminal work of Condensation [6] first used the particle filter (PF) [3]–[5] which could track the contour even when the observation likelihood was multimodal, e.g., due to clutter or occlusion. Also, the PF allowed directly using the image, or the edge map, as the observation. However, it tracked on the 6-D space of affine deformations and, hence, could not track large local deformations.

Many recent works on contour tracking [14]–[19] use the level set representation [20] of a contour (which automatically handles contour length/topology changes due to deformation) and propose different types of posterior mode trackers [21] for tracking deforming contours, with different likelihood and prior models and different mode computation methods. However, posterior mode trackers, also called approximate linear observers, implicitly assume that the posterior is effectively unimodal, i.e., has only one significant mode which is near the previously tracked contour. This is quite restrictive and may not hold when there is background clutter, occlusions or low contrast imagery (multimodal likelihood) and fast moving/deforming sequences (broad prior), e.g., see Figs. 2, 7, 6, and also [22]. The work of [18] computes the current contour estimate

Manuscript received January 04, 2007; revised April 08, 2009. First published November 24, 2009; current version published March 17, 2010. A part of this paper was presented in [1]. This work was supported in part by the National Science Foundation Grant ECCS-0725849. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Onur G. Guleryuz.

N. Vaswani is with the Electrical and Computer and Engineering Department, Iowa State University, Ames, IA 50011 USA (e-mail: namrata@iastate.edu).

Y. Rathi is with Harvard Medical School, Boston, MA 02115 USA.

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

A. Tannenbaum is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA, and also with the Electrical Engineering Department, The Technion—Israel Institute of Technology, Haifa, Israel (e-mail: tannenba@gatech.edu).

Digital Object Identifier 10.1109/TIP.2009.2037465

[1]The observation likelihood treated as a function of the state (here contour) has multiple local maxima.

as an approximate linear combination of the predicted contour and the observation likelihood mode nearest to it; and does the same for global motion. In other words, it defines posterior mode trackers separately for deformation and for motion. To address the limitations of posterior mode tracking and of full PF, in [22], we defined a PF to track affine deformations (similar to [6]), while using an approximate linear observer (similar to [18]) to estimate the nonaffine, or local, deformation for each affine deformed contour particle. In doing this, we implicitly assumed that the posterior of nonaffine deformation is unimodal. This is much weaker than the assumptions of most previous work and is valid for many practical problems shown in [22] where either the observation likelihood is unimodal as a function of local deformation or the local deformation per frame is small. However, in other situations, where the object is deforming fast, and there are multiple contour modes in the image separated by local deformation, the posterior of local deformation will be multimodal, e.g., see Figs. 1, 2, 6, and 7.

To relax the assumptions of [22], and, hence, also of [14]–[19], we need an importance sampling step in PF that also samples from the space of local deformations. However, as explained earlier, the space of local deformations is large dimensional. Thus, standard PF is impractical due to the reduction in effective particle size [4] as dimension increases. However, in most real problems, at any given time, most of the contour deformation occurs in a smaller number of dimensions ("effective basis space") while the deformation in the rest of the state space ("residual" space) is small. Thus, the large dimensional state spaces property introduced in [23] holds and so we can apply the PF with Mode Tracker (PF-MT) [23] idea to our problem. A large part of the contour deformation is smooth, i.e., it is low spatial frequency. Hence, we propose to use the space of deformation at a small subsampled set of locations along the contour as the effective basis space. *We call the resulting algorithm Deform PF-MT.*

The development of Deform PF-MT requires significant modifications because the space of contours is a non-Euclidean infinite dimensional space, while PF-MT was originally proposed for a large dimensional Euclidean space. Specifically the following.

1) There is no unique way to parameterize deformation and consequently no unique way to subsample it or to interpolate it from sub-samples. There is also no known way to perform PCA of the space of deformations, except when all contours have small deviations from the mean so that linear PCA is valid [24], [25], [19]. In this work, we develop two possible parameterizations for deformation and their corresponding interpolation algorithms. We define a novel system model for contour deformation dynamics, which can use any one of these two parameterizations This is done in Section II.
2) Since the space of contours is non-Euclidean, the discretization of the above model is valid only under the small motion assumption (see Assumption 1). We show how to implement importance sampling for the proposed model using the level set method, when the small motion assumption is violated. Specifically, we give a method to implement contour motion using level sets when the contour velocity is an unbounded random variable. In our case it is obtained by interpolating a small dimensional Gaussian vector onto all contour points and, hence, its maximum magnitude is unbounded. This is described in Section III.
3) A third contribution is a way to estimate the effective basis dimension and the system model parameters and a way to estimate changes in effective basis when either contour length or spatial frequency of deformation changes. This is described in Section IV.
4) We discuss simulation/experimental results and comparisons with [22] in Section V and conclude in Section VI.

We stress the difference of the current work from Condensation [6] which represented a contour by the B-spline control points of its x and y locations. In the current work, we use 1-D B-splines to only interpolate deformation (normal contour velocity) defined at K subsampled points onto all the contour points. Since the maximum spatial frequency of the deformation signal is usually much smaller than that of the contour signal and since we also use a mode tracking step to track residual deformation, in most cases our algorithm will require much fewer control points for the effective basis deformation than what [6] used for representing the contour. For example, in our experiments, $K = 6$ sufficed.

II. State Space Model

The observation at time t (image and edge map at t) is denoted by Y_t and the state at t (contour, its deformation and its translation) is denoted by X_t. We begin by describing the space of contours and the basic idea of the level set method in Section II-A. Next, we introduce the proposed system model for contour deformation in Section II-B. We briefly describe the observation model (taken from existing work) in Section II-C. Finally, we discuss two ways to parameterize deformation in order to use our system model in practice in Section II-D.

A. Space of Contours and Notation

The contour at time, t, can be represented as $C_t^c(p) = [C_t^{c,x}(p), C_t^{c,y}(p)]$, $p \in [0,1]$. The superscript ^{c} refers to the spatially continuous contour. The parametrization is not unique, i.e., all re-parameterizations of the parameter p of the form $\tilde{p} = f(p)$, where $f : [0,1] \to [0,1]$ is continuous and strictly monotonic, yield the same contour [26, Ch. 1]. In other words, the contour is a geometric entity [26]. The outward normal to C_t^c at p is denoted by $\overrightarrow{\mathbf{N}}(C_t^c(p))$ or by $\overrightarrow{\mathbf{N}}_t^c(p)$. Denote the space of contours by $\mathcal{S}$. The tangent space to $\mathcal{S}$ at C_t^c, denoted $\mathcal{TS}_{C_t^c}$, will be the space of all normal velocities (velocities along the normal to C_t^c at each point), since tangential velocity only re-parameterizes the contour [27], [26]. We use v_t^c to denote the vector of normal velocities (deformation), i.e., $v_t^c \in \mathcal{TS}_{C_t}$.

We use the terms "deformation", "deformation velocity", and "normal velocity" interchangeably. We let $(C_t)_{M_t \times 2} = [(C_t)_1^T, (C_t)_2^T, \ldots (C_t)_{M_t}^T]^T$ denote the spatial discretization of the contour. Here , $(C_t)_m$ is a row vector containing the x and y location of the mth contour point. Similarly, $(\overrightarrow{\mathbf{N}}_t)_{M_t \times 2}$ contains the normal vector (arranged as a row vector) at these points and $(v_t)_{M_t \times 1} = [(v_t)_1, \dots (v_t)_{M_t}]^T$ contains the normal direction deformation at these points.

When implementing contour motion using the level set method [28], [29], C_t consists of the x-y locations of the points of intersection of the contour with the pixel grid. Because of this, (a) M_t changes with time as the contour deforms, and (b) consecutive points are nonuniformly spaced along the parameter (i.e., $|p_{j+1} - p_j| \neq |p_{k+1} - p_k|, j \neq k$). For example, when using the arclength parametrization (i.e., p is the normalized arclength), the arclength distance between $(C_t)_{j+1}$ and $(C_t)_j$ is different from that between $(C_t)_{k+1}$ and $(C_t)_k$ for $j \neq k$. The same also applies for any other parametrization, except the one that forces consecutive points to be uniformly spaced. (c) Also, as the contour deforms, its arclength deforms, resulting in both local and global changes in arclength.

1) More Notation: Global translation is represented by a row vector $(\rho_t)_{1 \times 2}$. The normal direction deformation velocity (after removing translation) is denoted by $(v_t)_{M_t \times 1}$ and this can be split into a small dimensional effective basis deformation, $B_s v_{t,s}$, and residual deformation, $B_r v_{t,r}$. $(v_{t,s})_{K \times 1}$, $(v_{t,r})_{(M_t - K) \times 1}$ are the projections along the basis matrices, B_s, B_r. As we explain later, the effective basis states consist of ρ_t and $v_{t,s}$, and, thus, the effective basis dimension is $K + 2$. After specifying the system model in continuous time, we discretize it using $t = n\tau$ where τ is the discretization interval. We use the subscript, n, to denote the nth discrete time instant.

We use the notation $C_t(p_m)$ and $(C_t)_m$ interchangeably to denote the row vector containing the x-y location of the mth contour point. Similarly, $v_t(p_m)$ or $(v_t)_m$ is the normal direction deformation (scalar) of this contour point.

The max norm (also known as the ℓ_∞ norm) of an M-length vector v is $\|v\|_{\max} := \max_{i=1,\dots M} |v_i|$.

We use $\mathcal{N}(\mu, \Sigma)$ to denote a Gaussian probability density function (pdf) with mean μ and covariance Σ. We use $\mathcal{N}(x; \mu, \Sigma)$ to denote the value of the pdf computed at x. Some other notation specific to computing B_s is defined in Section II-D. This is primarily used only in Algorithms 2 and 3.

2) Level Set Method: We use the level set method [28], [29] to move the contours since it automatically handles contour length changes. C_t is represented implicitly as the zero level set of a 2-D function, denoted $\phi_t(x)$, i.e., C_t is the collection of all points $\{x \in \mathbb{R}^2 : \phi_t(x) = 0\}$ where x denotes the x-y coordinates. In other words, $\phi_t(x)$ satisfies $\phi_t(C_t(p)) = 0, \forall p$. The direction of the gradient, $\nabla \phi_t$, is along the normal to the level set. A common choice of $\phi_t(x)$ is the "signed distance function" [28], [29]. It is well known that level set evolution corresponding to contour evolution of the form $(\partial C_t)/(\partial t)(p) = v_t(p) \overrightarrow{\mathbf{N}}_t(p)$ is given by

$$\frac{\partial \phi_t(x)}{\partial t} = v_{\text{extend},t}(x) \|\nabla \phi_t(x)\| \tag{1}$$

where $v_{\text{extend},t}$ is the "normal extension" of v_t onto all nonzero level sets [28], [29].

B. System Model

The state consists of the contour, C_t, its deformation (normal contour velocity after removing translation), v_t, and the global translational velocity, ρ_t. In [22], we used the space of affine deformations as the effective basis space. In this work, we propose to use the space of global translations (ρ_t) and the space of deformation at a small number, K, of subsampled locations along the contour as the effective basis space. We denote the subsampled deformation by $v_{t,s}$. The effective deformation basis directions, $B_s(C_t)$, are the interpolation functions used to interpolate $v_{t,s}$ to all contour points. The deformation, v_t, can be split into effective deformation, $v_{t,s}$, along $B_s(C_t)$ and deformation, $v_{t,r}$, along the residual basis directions, $B_r(C_t)$, i.e., $v_t = B_s(C_t) v_{t,s} + B_r(C_t) v_{t,r}$. Thus, the state, $X_t = [C_t, \rho_t, v_{t,s}, v_{t,r}]$ with effective basis state, $X_{t,s} = [\rho_t, v_{t,s}]$. In this work, we assume that ρ_t and $v_{t,s}$ follow a first order autoregressive (AR) model, while $v_{t,r}$ is temporally and spatially independent and identically distributed (iid). Thus, the continuous time model for X_t is

$$\begin{aligned}
(\partial C_t)_m &= \rho_t^T dt + [B_s(C_t) v_{t,s} dt + B_r dv_{t,r}]_m (\overrightarrow{\mathbf{N}}_t)_m \\
&\qquad m = 1, \dots M_t \\
dv_{t,s} &= -A'_s v_{t,s} dt + dW_{t,s}, \quad dW_{t,s} \sim \mathcal{N}(0, \Sigma'_s) \\
dv_{t,r} &= dW_{t,r}, \quad dW_{t,r} \sim \mathcal{N}(0, \Delta'_r I) \\
d\rho_t &= -A'_\rho \rho_t dt + dW_{t,\rho}, \quad dW_{t,\rho} \sim \mathcal{N}(0, \Sigma'_\rho)
\end{aligned} \tag{2}$$

where ∂C_t denotes the partial differential of C_t w.r.t. time; $W_{t,s}, W_{t,r}, W_{t,\rho}$ are Brownian motions [30], with dimensions K, $M_t - K$ and 2, respectively; and $_m$ refers to the basis matrix or normal direction computed at the mth contour point (at parameter location p_m). Assume that the observations arrive every τ time instants, i.e., at times $t = n\tau$. Defining $\Sigma_s \triangleq \Sigma'_s \tau^2$, $\Sigma_\rho \triangleq \Sigma'_\rho \tau^2$, $\Delta_r \triangleq \Delta'_r \tau$, $A_s \triangleq [I - A'_s \tau]$, $A_\rho \triangleq [I - A'_\rho \tau]$, the above can be time-discretized as follows:

$$(C_n)_m = (\tilde{C}_n)_m + [B_r(C_{n-1}) v_{n,r}]_m (\overrightarrow{\mathbf{N}}_{n-1})_m \tag{3}$$

$$(\tilde{C}_n)_m \triangleq (C_{n-1})_m + \rho_n^T + [B_s(C_{n-1}) v_{n,s}]_m \, (\overrightarrow{\mathbf{N}}_{n-1})_m \tag{4}$$

$$v_{n,s} = A_s v_{n-1,s} + \nu_{n,s}, \quad \nu_{n,s} \sim \mathcal{N}(0, \Sigma_s) \tag{5}$$

$$v_{n,r} = \nu_{n,r}, \quad \nu_{n,r} \sim \mathcal{N}(0, \Delta_r I) \tag{6}$$

$$\rho_n = A_\rho \rho_{n-1} + \nu_{n,\rho}, \quad \nu_{n,\rho} \sim \mathcal{N}(0, \Sigma_\rho) \tag{7}$$

where $C_n \equiv C_{n\tau}$ and so on and $(\cdot)_m$ denotes the mth row of the matrix or vector (corresponding to the mth contour point). The effective basis matrix, $[B_s(C_{n-1})]_{M_{n-1} \times K}$, will be defined by (11) or by (18), depending on the type of parametrization chosen (we explain this later in Section II-D). The residual basis matrix, $[B_r(C_{n-1})]_{M_{n-1} \times (M_{n-1} - K)}$, satisfies

$$B_r B_r^T = I - B_s \left(B_s^T B_s\right)^{-1} B_s^T, \; B_s \triangleq B_s(C_{n-1}). \tag{8}$$

Note that the discretization in (3) and (4) assumes the following.

Assumption 1 (Small Motion): The observation interval, τ, is small enough, or equivalently, the deformation velocity $B_s v_{n,s} + B_r v_{n,r}$ is slow enough so that $\overrightarrow{\mathbf{N}}_{n-1}$ is also approximately normal to C_n.

Remark 1: This assumption can be violated since $v_{n,s}$ is a Gaussian random vector and so its max norm and, hence, the max norm of $B_s v_{n,s}$, is unbounded. When it is violated, one

```
Algorithm 1 Deform PF-MT. Going from $\pi_{n-1}^N$ to $\pi_n^N(X_n) = \sum_{i=1}^N w_n^{(i)} \delta(X_n - X_n^i)$,  $X_n^i = [C_n^i, \rho_n^i, v_{n,s}^i, v_{n,r}^i]$
1) Importance Sample (IS) on effective basis: For each $i = 1, 2 \ldots N$,
    a) Importance sample $\rho_n^i \sim \mathcal{N}(A_\rho \rho_{n-1}^i, \Sigma_\rho)$ and $v_{n,s}^i \sim \mathcal{N}(A_s v_{n-1,s}^i, \Sigma_s)$.
    b) Compute $\tilde{C}_n^i$ by implementing (4): use Algorithm 2 for radial angle parametrization, and use Algorithm 3 for arclength parametrization.
2) Approximate Mode Tracking (MT) in residual space: For each $i = 1, 2 \ldots N$,
    a) Compute $C_n^i$ by starting with $\tilde{C}_n^i$ and running $G$ iterations of gradient descent to minimize $E(C_n) \triangleq -\log[p(Y_n|C_n)]$. Gradient descent is implemented using the standard level set method (or the narrowband level set method) [28]. Usually $G = 1$ or 2 suffices. This step follows from the approximations explained in Sec. III-C.
3) Weight: For each $i = 1, 2 \ldots N$, compute $w_n^i$ using (26). Resample [4].
```

discrete time step of (3) and (4) is implemented using multiple iterations of the level set method.

Remark 2: Only under the above assumption, the number of points in C_n and in C_{n-1} will be the same. The level set implementation of (3), (4) allows this number to be different.

Remark 3: Both the choice of effective basis and the dynamics of $v_{n,s}$, $v_{n,r}$ or ρ_n can be easily changed in the above model without changing anything else in our framework.

C. Observation Model

The observation at time n, Y_n, is the image at n and its edge map. We assume that Y_n depends only on C_n (and not on the velocity), i.e., the observation likelihood

$$p(Y_n \,|\, X_n) = p(Y_n \,|\, C_n) \propto e^{-E(C_n)} \tag{9}$$

where $E(C_n)$ is a segmentation energy functional. Many ideas have been proposed in literature for segmentation and tracking energies—these can be classified as "region based", e.g., [31], [22], [2]; "edge based", e.g., [6] or "motion based", e.g., [11]. Using a good observation model is important, but we do not address it here. We use a product of the simple region-based likelihood of [22] which was motivated by the Chan and Vese model [31] and the edge-based likelihood proposed in Condensation [6]. This combines the advantages of a region based approach (ability to select the object of interest) with those of an edge based approach (ability to deal with intensity variations across the sequence and with errors in learning the foreground or background object intensities). The observation likelihood equation for defining $p(Y_n \,|\, C_n)$ and the implicit assumptions in using it are given in the Appendix. The resulting likelihood is frequently multimodal with a strong mode at the object of interest (high region and edge likelihood) and a weaker mode at any "object" (high edge likelihood only).

D. Parameterizing and Interpolating Contour Deformation

So far, we have postponed the question of how to parameterize deformation and how to use it to interpolate subsampled deformation. We use 1-D closed cubic B-splines as the interpolation functions since they are twice continuously differentiable (required by the level set method) and provide local control (one control point, $v_{n,s,j}$, affects deformation of only a small region of the contour). Local control makes the interpolation robust to outlier errors in any one control point.

Let $[B^{cb}(\underline{q}^*; q)]_{1\times K}$ denote the closed cubic B-spline computed at parameter location, $q \in [0, 1)$. This can be computed in practice using the cox-de-Boor recursion or directly [32]. The basis points (called "knots" [32]) are $\underline{q}^*{}_j, j = 1, 2 \ldots K$. The subsampled deformation, v_s, interpolated to contour point, C_m, is given by

$$v_m = B^{cb}(\underline{q}^*;\ q(C_m))\, v_s \tag{10}$$

where $q(C_m)$ is the parameter value corresponding to contour point, C_m. *In general, one can replace B-splines by any choice of periodic interpolation functions without changing anything in our framework.* In the next two sub-subsections we show how to use the above interpolation function to parameterize deformation in two different ways.

1) Radial Angle Parametrization: We use the radial angle, $\theta(C_m)$ (angular coordinate of the mth contour point w.r.t. the centroid of the contour's inside region, $[\mu^x, \mu^y]$), normalized by 2π, as the parameter, i.e., $q(C_m) := (\theta(C_m))/(2\pi)$. Thus

$$[B_s(C)]_{m,j} = \left[B^{cb}\left(\frac{\underline{\theta}^*}{2\pi}; \frac{\theta(C_m)}{2\pi}\right)\right]_{1,j} \quad j = 1, \ldots K \tag{11}$$

$$\text{where } \theta(C_m) \triangleq \arctan\left[\frac{C_m^y - \mu^y}{C_m^x - \mu^x}\right] \tag{12}$$

where the vector $\underline{\theta}^*$ contains the K knots (basis points) which are uniformly spaced at angular distance $\alpha_s = (2\pi/K)$ i.e., $\underline{\theta}^*{}_j = ((j-1)2\pi)/(K)$, $j = 1, 2, \ldots K$, and m denotes the mth contour point.

2) Arclength Parametrization: We use the arclength [26], $s(C_m)$, of the contour point C_m w.r.t. an initial starting point, normalized by the total length, $L = L(C)$, as the parameter, i.e., $q(C_m) = (s(C_m))/(L)$. Thus

$$[B_s(C)]_{m,j} = \left[B^{cb}\left(\frac{\underline{s}^*}{L}; \frac{s(C_m)}{L}\right)\right]_{1,j} \quad j = 1, \ldots K \tag{13}$$

$$\text{where } s(C_m) \triangleq \sum_{k=2}^{m} \|C_k - C_{k-1}\| \tag{14}$$

where $L = L(C)$ is the contour length; $\underline{s}^*$ has components $\underline{s}^*{}_j, j = 1, \ldots K$ which denote the arclength location of the jth knot w.r.t. a fixed starting point (that moves with the same velocity as the contour) and m denotes the mth contour point. At time, $n = 0$, we place the knots on the contour uniformly at

$\alpha_s = L/K$ arclength distance apart, i.e., $\underline{s}^*_j = ((j-1)L/K)$, $j = 1,\dots K$. In order to ensure that $v_{n+1,s,j}$ affects deformation of the same part of the contour at time $n+1$ as $v_{n,s,j}$ did at time n, we move the the x-y locations of the knots with the same velocity as that of the contour. Their arclength locations at time $n+1$ form the new knots vector which is used to interpolate $v_{n+1,s}$. We explain this below.

Let $[\underline{x}^*(\underline{s}^*, C)]_{K\times 2}$ contain the x-y locations of the knots $\underline{s}^*$ on the contour C ($\underline{x}^*_j$ contains the x-y location of $\underline{s}^*_j$, $j = 1,\dots K$). It is computed as

$$\begin{aligned} \underline{x}^*_j(\underline{s}^*, C) &= C_{m_j}, \\ m_j &\triangleq \arg\min_m |\underline{s}^*_j - s(C_m)|, \quad j = 2,\dots K. \end{aligned} \tag{15}$$

In particular, $\underline{x}^*_1 = C_1$ is the initial contour point. The inverse mapping, $\underline{s}^*(\underline{x}^*, C)$ is given by: $\underline{s}^*_1(\underline{x}^*, C) = 0$, and

$$\begin{aligned} \underline{s}^*_j(\underline{x}^*, C) &= s(C_{m_j}) \\ m_j &\triangleq \arg\min_m \|\underline{x}^*_j - C_m\|, \quad j = 1,\dots K. \end{aligned} \tag{16}$$

As the contour deforms, the knots also move with the same velocity, i.e., their x-y locations at time n, $\underline{x}^*_{n,j}$, follow

$$\begin{aligned} \underline{x}^*_{n,j} &= \underline{x}^*_{n-1,j} + \rho_n^T + [B_s(C_{n-1})v_{n,s}]_{m_j}(\overrightarrow{\mathbf{N}}_{n-1})_{m_j} \\ m_j &\triangleq \arg\min_m \|\underline{x}^*_{n,j} - (C_{n-1})_m\|, \quad j = 1,\dots K \end{aligned} \tag{17}$$

and at each n, $\underline{s}^*(\underline{x}^*_n, C_n)$ is recomputed using (16). Because of this, the knots may not remain uniformly spaced after some time: some may come "too close", others may go "too far" away compared to α_s. This will require a knot re-allocation (change of effective basis), explained in Section IV-B.

In summary, $B_s(C_n)$ at time n is computed as

$$[B_s(C_n)]_{m,j} = \left[B^{cb}\left(\frac{\underline{s}^*(\underline{x}^*_n, C_n)}{L(C_n)}; \frac{s((C_n)_m)}{L(C_n)}\right)\right]_{1,j} \quad j = 1\dots K \tag{18}$$

and $\underline{x}^*_n$ follows (17); $\underline{s}^*(\cdot,\cdot)$ is defined in (16) and $s(\,\cdot\,)$ is defined in (14).

3) Choosing Parametrization Type: For a given problem, we discuss some preliminary ideas on which parametrization to choose when. Contour motion using the level set method is easily implemented using a x-y location based parametrization of deformation such as radial angle. In this case, the interpolation functions, and, hence, the extension velocities can be directly computed without having to first compute the zero level set (contour). This makes implementation using level sets intuitive and fast. Also such a parametrization can, in principle, automatically handles changes in contour topology.

On the other hand, radial parametrization cannot be used if one would like to independently deform two or more points of a contour that are close along the radial angle, but are far if one moves along the contour arclength. Such applications can be handled by the arclength parametrization. However, level set implementation of the arclength parametrization is expensive because the zero level set (contour) needs to be computed at each iteration (details in Section III-B). Also, it is useful for applications where change in topology is not allowed. In practical implementations, we handle this by detecting topology change of contour particles and assigning a zero likelihood to particles for which topology change occurs.

III. Deform PF-MT

A particle filter (PF) [3], [33], [5] uses sequential importance sampling [33] along with a resampling step [3] to output at each time n, a cloud of N particles, $\{X_n^i\}_{i=1}^N$ with weights $\{w_n^i\}$ whose empirical measure $\pi_{n\,|\,n}^N(X_n) \triangleq \sum_{i=1}^N w_n^i \delta(X_n - X_n^i)$, closely approximates the true posterior, $\pi_{n\,|\,n}(X_n) \triangleq p(X_n\,|\,Y_{1:n})$. Here $\delta(X-a)$ denotes the Dirac delta function at a. For large dimensional problems such as deformable contour tracking, direct application of PF becomes impractical due to the reduction in effective particle size as dimension increases. To address this issue, the PF with Mode Tracker (PF-MT) algorithm was proposed in [23]. *The importance sampling dimension of PF-MT is only* $\dim(X_{n,s})$ *which is much smaller than* $\dim(X_n)$ *and this greatly improves its effective particle size compared to PF.* We first explain the general PF-MT idea and then explain how to implement each step for deformable contour tracking (Deform PF-MT).

A. Main Idea of PF With Mode Tracker (PF-MT)

PF-MT splits the state vector X_n into $X_n = [X_{n,s}, X_{n,r}]$ where $X_{n,s}$ denotes coefficients along a small dimensional "effective basis" while $X_{n,r}$ denotes the "residual" basis coefficients. In our problem, $X_{n,s} = [\rho_n, v_{n,s}]$ and $X_{n,r} = [v_{n,r}, C_n]$. PF-MT importance samples the effective state particles, $X_{n,s}^i$, from their state transition prior, while replacing importance sampling by posterior Mode Tracking (MT) for the residual state particles, $X_{n,r}^i$. Mode Tracking is an approximation to the "Efficient Importance Sampling" technique, proposed in [23], which importance samples $X_{n,r}^i$ from a Gaussian approximation to the residual posterior

$$\begin{aligned} p^{**,i}(X_{n,r}) &\triangleq p\left(X_{n,r}\,|\,X_{n,s}^i, X_{n-1}^i, Y_n\right) \\ &\propto p\left(Y_n\,|\,X_{n,s}^i, X_{n,r}\right) \\ &\quad \times p\left(X_{n,r}\,|\,X_{n-1,r}^i, X_{n,s}^i\right) \end{aligned} \tag{19}$$

about its unique mode (it assumes unimodality of $p^{**,i}(X_{n,r})$). As explained in [23], unimodality of the residual posterior is ensured if the residual space state transition variance is small enough compared to the distance between likelihood modes along residual space. The Mode Tracking approximation deterministically sets $X_{n,r}^i$ equal to the unique mode of $p^{**,i}(X_{n,r})$, and this is a valid approximation when its maximum variance is small enough. This is again ensured if the residual space state transition variance is small enough. For our model, given in (3)–(7), the residual space state transition variance in any direction is Δ_r and thus *PF-MT is applicable if* Δ_r *is "small enough", i.e.,* $\Delta_r < \Delta_{\max}$. Some possible ways to compute $\Delta_{\max}$ are discussed in [23], but most of them are of only theoretical interest. We give the stepwise PF-MT algorithm below.

Algorithm 2 Implementing (4) for the radial parametrization. Computing $\tilde{\phi}_n^i$ (and its zero level set, $\tilde{C}_n^i$, if needed) using ϕ_{n-1}^i (or C_{n-1}^i), $v_{n,s}^i$, ρ_n^i and the knots, $\underline{\theta}^*$, as inputs.

```
Knots $\underline{\theta}_j^* = (j-1)2\pi/K$ for $j = 1, 2, \ldots K$ are defined once. Let $\Omega$ be the region in the x-y plane where the level set function is defined (or is the narrowband region in case narrowband level set method [28] is used).
For each particle, $i$, do
  1) Compute $\mu_n^x, \mu_n^y$ as the centroid coordinates of the region where $\phi_{n-1}^i$ is positive (the inside region of the contour $C_{n-1}^i$).
  2) Compute $\theta_n(x) = \arctan[\frac{x_1 - \mu_n^y}{x_2 - \mu_n^x}]$ for all points $x \in \Omega$.
  3) Compute the B-spline basis functions, $B_{s,extend,n}^i(x) = B^{cb}(\frac{\underline{\theta}^*}{2\pi}; \frac{\theta_n(x)}{2\pi})$, using the cox-de-Boor recursion formula [32] for a closed cubic B-splines, for each $\theta(x)$, $x \in \Omega$.
  4) Compute $v_{extend,n}^i(x) = B_{s,extend,n}^i(x) v_{n,s}^i$ for all $x \in \Omega$.
  5) Compute $P = \lceil ||v_{n,s}^i||_{max} \rceil$ where $||.||_{max}$ denotes the max norm and $\lceil . \rceil$ denote the ceil function (note if Assumption 1 is satisfied, $P = 1$). Iterate the following for $k = 1$ to $P$, beginning with $\phi_{n-1,0}^i(x) = \phi_{n-1}(x)$:
       $$\phi_{n-1,k}^i(x) = \phi_{n-1,k-1}^i(x) + \frac{v_{extend,n}^i(x)}{P} ||\nabla_x \phi_{n-1,k-1}^i(x)||$$
  6) Set $\phi_{n-1,tmp}^i(x) := \phi_{n-1,P}^i(x)$. Implement translation by $\rho_n^i$, i.e. compute $\tilde{\phi}_n^i(x) = \phi_{n-1,tmp}^i(x - \rho_n^i)$. Since $\rho_n^i$ is, in general, not an integer, this requires interpolation using the values of $\phi_{n-1,tmp}^i(x)$ at the four pixels nearest to $x - \rho_n^i$.
  7) Output $\tilde{\phi}_n^i$ (and $\tilde{C}_n^i$).
```

1) Importance Sample $X_{n,s}$ from its state transition prior. For $i = 1, \ldots N$, sample $X_{n,s}^i \sim p(X_{n,s}^i | X_{n-1}^i)$.
2) Mode Track $X_{n,r}$. For $i = 1, \ldots N$, set $X_{n,r}^i = \arg\min_{X_{n,r}} L^i(X_{n,r})$ where $L^i(X_{n,r}) \triangleq -\log p^{**,i}(X_{n,r})$.
3) Weight and Resample. For $i = 1, \ldots N$ compute $w_n^i = (\tilde{w}_n^i)/(\sum_{j=1}^N \tilde{w}_n^j)$, $\tilde{w}_n^i = w_{n-1}^i p(Y_n | X_n^i) p(X_{n,r}^i | X_{n-r}^i, X_{n,s}^i)$. Resample [4].

In the next three subsections, we discuss how each step is developed for Deform PF-MT. The complete algorithm is summarized in Algorithm 1 and its code is available at `http://www.ece.iastate.edu/~namrata/research/ContourTrack.html#code` [34].

B. Deform PF-MT: Importance Sampling on Effective Basis

For each particle, $i = 1, \ldots, N$, this involves sampling $v_{n,s}^i$ and ρ_n^i from their state transition priors, $\mathcal{N}(A_s v_{n-1,s}^i, \Sigma_s)$ and $\mathcal{N}(A_\rho \rho_{n-1}^i, \Sigma_\rho)$ respectively and computing $\tilde{C}_n^i$ using (4).We implement (4) using the level set method. When Assumption 1 holds, the level set evolution corresponding to contour evolution given by (4) is

$$\phi_{n-1,\text{tmp}}^i(x) = \phi_{n-1}^i(x) + v_{\text{extend},n}^i(x) ||\nabla_x \phi_{n-1}^i(x)|| \tag{20}$$

$$\tilde{\phi}_n^i(x) = \phi_{n-1,\text{tmp}}^i\left(x - \rho_n^i\right) \tag{21}$$

where $v_{\text{extend},n}^i$ is the normal extension [28], [29] of $B_s(C_{n-1}^i) v_{n,s}^i$ onto nonzero level sets and B_s is defined either by (18) or by (11). Since ρ_n^i may not be an integer, the implementation of (21) requires interpolation using the values of $\phi_{n-1,\text{tmp}}^i(x)$ at the four pixels nearest to $x - \rho_n^i$. Very often, the deformation per frame is large and Assumption 1 does not hold. In particular, whenever the max norm of $v_{\text{extend},n}^i$ (which is equal to the max norm of $v_{n,s}^i$) is more than 1, the CFL condition [26] will be violated.[2] In this case, (20) needs to be iterated $P = \lceil ||v_{n,s}^i||_{\max} \rceil$ times, replacing $v_{\text{extend},n}^i$ by $v_{\text{extend},n}^i / P$ each time. The implementation of (4) is summarized in Algorithms 2 (radial) and 3 (arclength).

As detailed in Algorithms 2 and 3, $B_s(C_{n-1})$ and its extension onto all level sets need to be computed at each time step n. For the radial parametrization (Algorithm 2), this can be done without computing the zero level set (contour).[3] For the arclength parametrization (Algorithm 3), at each iteration, (i) the contour (zero level set) needs to be computed; it needs to always be traversed in the same order (say clockwise); and starting point correspondence needs to be maintained; and (ii) the basis points need to be moved along with the contour using (17).

C. Deform PF-MT: Mode Tracking on Residual Space

For our problem, the residual posterior

$$p^{**,i}(X_{n,r}) = p^{**,i}(v_{n,r}, C_n)$$
$$\propto \underbrace{p\left(Y_n | \tilde{C}_n^i + B_r^i v_{n,r} \overrightarrow{\mathbf{N}}^i\right) p\left(v_{n,r} \,|\, v_{n-1,r}^i\right)}_{p^{**,i}(v_{n,r})}$$
$$\times \underbrace{\delta\left(C_n - \left(\tilde{C}_n^i + B_r^i v_{n,r} \overrightarrow{\mathbf{N}}^i\right)\right)}_{p^{**,i}(C_n \,|\, v_{n,r})}$$

where $\tilde{C}_n^i + B_r^i v_{n,r} \overrightarrow{\mathbf{N}}^i$ is compact notation for the right hand side of (3). The Dirac delta function is trivially unimodal at $(\tilde{C}_n^i + B_r^i v_{n,r} \overrightarrow{\mathbf{N}}^i)$. So we only need to find the mode

[2]The CFL condition for stability of (20) requires that the max norm of $v_{\text{extend},n}$ be less than one [26] However, note that since we do not evolve a PDE to convergence, it is not clear if the CFL condition needs to be satisfied.

[3]Assuming the requirement of normal extension velocities [28], [29] is relaxed (this is done very often in numerical implementations since it will only result in more frequent re-initialization of level set functions [26]).

Algorithm 3 Implementing (4) for the arclength parametrization. Computing $\tilde{\phi}_n^i$ (and its zero level set, $\tilde{C}_n^i$) and $\underline{x}^{*i}_n$ using ϕ_{n-1}^i (and its zero level set, C_{n-1}^i), $v_{n,s}^i$, ρ_n^i, $\underline{x}^{*i}_{n-1}$ as inputs.

```
For each particle, $i$, do
  1) If $n > 1$, compute $\underline{s}^* = \underline{s}^*(\underline{x}^{*i}_{n-1}, C_{n-1}^i)$ using (16),
     Else (first time instant), set $\underline{s}_j^* = \frac{(j-1)L(C_{n-1}^i)}{K}$, $j = 1, \dots K$ and compute $\underline{x}^{*i}_{n-1,j}$, $j = 1, \dots K$ using (15).
  2) Compute $P = \lceil ||v_{n,s}^i||_{max} \rceil$ where $||.||_{max}$ denotes the max norm and $\lceil . \rceil$ denote the ceil function. Iterate the following, for $k = 1$ to $P$, beginning with $\phi_{n-1,0}^i(x) = \phi_{n-1}(x)$ and $\underline{x}^{*i}_{n-1,0} = \underline{x}^{*i}_{n-1}$
     a) Compute the zero level set $C_{n-1,k-1}^i$ of $\phi_{n-1,k-1}^i$. It should be computed in the correct order (always clockwise or always anticlockwise) and starting point correspondence with previous contour should be maintained.
     b) Compute the arclength locations, $s([C_{n-1,k-1}^i]_m)$ of all contour points $m = 1, 2, \dots M_{n-1}^i$, using (14).
     c) Compute the B-spline basis functions, $B_s(C_{n-1,k-1}^i) = B^{cb}(\frac{\underline{s}^*}{L}; \frac{s([C_{n-1,k-1}^i]_m)}{L})$, for each $m = 1, 2, \dots M_{n-1}^i$, using the cox-de-Boor recursion formula [32] for a closed cubic B-spline.
     d) Compute $B_s(C_{n-1,k-1}^i)\frac{v_{n,s}^i}{P}$ and compute its normal extension [28], $v_{extend,n,k}^i(x)$, $\forall x \in \Omega$.
     e) Compute
        $$\phi_{n-1,k}^i(x) = \phi_{n-1,k-1}^i(x) + v_{extend,n,k}^i(x)||\nabla_x \phi_{n-1,k-1}^i(x)||$$
        $$\underline{x}^{*i}_{n-1,j,k} = \underline{x}^{*i}_{n-1,j,k-1} + [B_s(C_{n-1,k-1}^i)]_{m_j} \frac{v_{n,s}^i}{P} [\overrightarrow{\mathbf{N}}(C_{n-1,k-1}^i)]_{m_j}, \quad m_j \triangleq \arg\min_m ||\underline{x}_j^* - (C_{n-1,k-1}^i)_m||$$
  3) Set $\phi_{n-1,tmp}^i(x) := \phi_{n-1,P}^i(x)$. Implement translation by $\rho_n^i$, i.e. compute $\tilde{\phi}_n^i(x) = \phi_{n-1,tmp}^i(x - \rho_n^i)$.
  4) Implement translation of the knot locations: $\underline{x}^{*i}_{n,j} = \underline{x}^{*i}_{n-1,j,P} + \rho_n^{i\,T}$, $\forall j = 1, 2 \dots K$.
  5) Output $\tilde{\phi}_n^i$ (and $\tilde{C}_n^i$) and $\underline{x}^{*i}_n$.
```

of $p^{**,i}(v_{n,r})$ and set $v_{n,r}^i$ equal to it, and then set $C_n^i = \tilde{C}_n^i + B_r^i v_{n,r}^i \overrightarrow{\mathbf{N}}^i$. Thus, we set $v_{n,r}^i = \arg\min_{v_{n,r}} L^i(v_{n,r})$, where $L^i(v_{n,r}) \triangleq -\log p^{**,i}(v_{n,r})$ is computed as

$$L^i(v_{n,r}) = E\left(\tilde{C}_n^i + B_r^i v_{n,r} \overrightarrow{\mathbf{N}}^i\right) + \frac{v_{n,r}^T v_{n,r}}{2\Delta_r} + \text{const}$$
$$\text{where} E(C) \triangleq -\log p(Y_n|C) \tag{22}$$

This follows from (9) and (6). ${v_{n,r}}^i$ is then used to compute C_n^i by using (3). The minimization of $L^i(v_{n,r})$ is done using gradient descent. The gradient of $L^i(v_{n,r})$ is $\nabla L^i(v_{n,r}) = \sum_m [B_r(\tilde{C}_n^i + B_r^i v_{n,r} \overrightarrow{\mathbf{N}}^i)]_m^T \nabla_{C_m} E(\tilde{C}_n^i + B_r^i v_{n,r} \overrightarrow{\mathbf{N}}^i)(\overrightarrow{\mathbf{N}}^i)_m^T + (v_{n,r})/(\Delta_r)$, where $B_r^i \triangleq B_r(\tilde{C}_n^i)$. Implementing the above using the level set method is very expensive, because for each particle, i, at each gradient descent iteration, k, the gradient computation involves: (a) computing $\tilde{C}_n^i + B_r^i v_{n,r} \overrightarrow{\mathbf{N}}^i$ which may take multiple level set iterations; and (b) computing $B_r(\tilde{C}_n^i + B_r^i v_{n,r} \overrightarrow{\mathbf{N}}^i)$ by solving (8), which is an expensive matrix square root computation.

A much less expensive, but approximate, solution is to change our system model slightly, i.e., replace (3) by the following model:

$$(C_n)_m = (\tilde{C}_n)_m + (v_{n,a})_m \overrightarrow{\mathbf{N}}_m, \quad v_{n,a} \sim \mathcal{N}(0, \Delta_r I). \tag{23}$$

With this model, instead of minimizing only along the residual basis, we minimize along all directions, i.e., we minimize over both $v_{n,s}$ and $v_{n,r}$. Equivalently, we can minimize over C_n, i.e., L^i can be redefined as

$$L^i(C_n) \triangleq E(C_n) + \frac{d^2\left(C_n, \tilde{C}_n^i\right)}{2\Delta_r} \tag{24}$$

where $d(C_n, \tilde{C}_n^i)$ should be the Euclidean norm of v_a, $(v_a)_m := (C_n - \tilde{C}_n^i)_m \overrightarrow{\mathbf{N}}_m^T$. However, except under the small motion assumption (Assumption 1), the difference of two contours is not defined. In fact they may not even have the same length. To handle the general case where small motion may not hold, we set d to be the set symmetric distance [26], [35] between C_n and $\tilde{C}_n^i$, which is always computable for contours. Since the residual space variance, Δ_r, is assumed small, gradient descent will result in only a small change in any coordinate of C_n (or equivalently in any coordinate of $v_{n,s}$ or $v_{n,r}$). In particular, the change in $v_{n,s}$ will be small and thus replacing (3) by (23) is a valid approximation. Gradient descent of (24) can be efficiently implemented using the standard level set method [28], [29] without having to ever compute B_r or the intermediate contours. The small bias introduced by the above can possibly be eliminated using[36].

We have shown in [21] that the minimizer of (24) can be computed by starting with $C_n = \tilde{C}_n^i$ as initial guess and running G iterations (for some G) of gradient descent to minimize $E(C_n)$. Since Δ_r is small, the maximum value of G will also be small and so a heuristic choice of G suffices. In experiments, we use $G = 1$ or 2.

Thus, in summary, we have modified the original mode tracking method in three ways. (a) We replaced mode search along B_r^i by mode search along all directions, which is equivalent to minimization of (24) over C_n. This is a valid

Algorithm 4 Estimating the Effective Deformation Dimension, K, and the System Model Parameters.

Given a training sequence of approximately equal length contours $C_0, \ldots, C_n, ..C_T$, do:

1) *Compute the translation, ρ_n, $\forall n$.* ρ_n is the difference between the centroid of the region inside C_{n-1} and that of the region inside C_n. By abusing notation, denote the centered contour (contour with centroid subtracted) again by C_n.
2) *Compute the deformation, v_n, $\forall n$.* For each point, $[C_{n-1}]_m$, set $[C_{n,perp}]_m$ as the point on C_n at which $[\overrightarrow{\mathbf{N}}_{n-1}]_m$ intersects it. Doing this for each m on C_{n-1} gives $C_{n,perp}$ which has the same number of points as C_{n-1}. Ideally, $[C_{n,perp}]_m - [C_{n-1}]_m$ should be along $[\overrightarrow{\mathbf{N}}_{n-1}]_m$. But due to numerical error, this does not exactly hold. Thus, we compute $[v_n]_m$ as its projection along $[\overrightarrow{\mathbf{N}}_{n-1}]_m$, i.e. $[v_n]_m = ([C_{n,perp}]_m - [C_{n-1}]_m)[\overrightarrow{\mathbf{N}}_{n-1}]_m^T$, for $m = 1, \ldots, M_{n-1}$.
3) **Estimate effective deformation dimension, K, as follows.** (Details are given in Sec. IV-A.)
   a) Resample $v_1(p), ..v_n(p), ..v_T(p)$ to M uniformly spaced points. If p_{min} is the minimum parameter spacing between any two consecutive points, and L_{mean} is the average contour length (for radial parametrization, $L_{mean} = L = 2\pi$), we use $M = \lceil \frac{L_{mean}}{p_{min}} \rceil$.
   b) Compute the deformation power spectral density (PSD) as $S_v(\frac{2\pi k}{M}) = \frac{1}{T}\sum_{n=1}^{T} \frac{|V_n(\frac{2\pi k}{M})|^2}{M}$ for $k = 0, \ldots M-1$. Here V_n is the DFT of resampled v_n.
   c) Compute f_{min} as the smallest frequency for which the maximum PSD value outside f_{min} is smaller than a threshold, Δ_{max}, e.g. 0.05% of total deformation energy. Compute $K = M \cdot 2f_{min}$.
4) *Compute effective basis deformation, $v_{n,s}$, $\forall n$.* Using the computed value of K, compute $B_s(C_{n-1})$ using (11) or (18). Then compute the least squares estimate of $v_{n,s}$ from v_n as $v_{n,s} = (B_s^T B_s)^{-1} B_s^T v_n$ where $B_s := B_s(C_{n-1})$. Notice that since K is smaller than the signal dimension, M_n, $(B_s^T B_s)$ will be invertible.
5) *Compute residual deformation, $v_{n,r}$, $\forall n$,* as $v_{n,r} = v_n - B_s v_{n,s}$.
6) *Estimate A_s, Σ_s using $\{v_{n,s}\}$.* Compute $A_s = R_1 R_0^{-1}$ where $R_1 \triangleq \frac{1}{T-1}\sum_{n=2}^{T} v_{n,s} v_{n-1,s}^T$ and $R_0 \triangleq \frac{1}{T}\sum_{n=1}^{T} v_{n,s} v_{n,s}^T$. Compute $\Sigma_s = \frac{1}{T-1}\sum_{n=2}^{T}[v_{n,s} - A_s v_{n-1,s}][v_{n,s} - A_s v_{n-1,s}]^T$. The matrix, R_0, is invertible if the time sequence length, T, is large compared to K, e.g. T is 2-3 times K.
7) *Estimate A_ρ, Σ_ρ using $\{\rho_n\}$,* exactly as above.
8) Estimate Δ_r as $\Delta_r = \frac{1}{T}\sum_{n=1}^{T} \frac{1}{M_{n-1} - K} v_{n,r}^T v_{n,r}$.

approximation because Δ_r is small and thus change along any direction, and in particular along B_s, will also be small. (b) To define $d(C_n, \tilde{C}_n)$ in (24), the Euclidean distance cannot be used (except under small motion) and so we replace it by the set symmetric distance [26], [35]. (c) Exact mode computation is replaced by starting with $\tilde{C}_n^i$ and running only a few (1–2) iterations of gradient descent to minimize $E(C_n)$. This has been shown to approximate the actual mode in [21].

D. Deform PF-MT: Weighting

This involves computing the weights, $w_n^i, \forall i = 1, 2 \ldots N$. w_n^i is computed as

$$w_n^i = \frac{\tilde{w}_n^i}{\sum_{j=1}^{N} \tilde{w}_n^j}$$
$$\tilde{w}_n^i \triangleq w_{n-1}^i p\left(Y_n | C_n^i\right) \mathcal{N}\left(v_{n,r}^i; 0, \Delta_r I\right) \tag{25}$$

because of the approximation of Section III-C, $\mathcal{N}(v_{n,r}^i; 0, \Delta_r I)$ needs to be replaced by a constant times $\exp[-d^2(C_n, \tilde{C}_n^i)/2\Delta_r]$. Thus, (25) is replaced by

$$w_n^i = \frac{\tilde{w}_n^i}{\sum_{j=1}^{N} \tilde{w}_n^j}$$
$$\tilde{w}_n^i = w_{n-1}^i p\left(Y_n | C_n^i\right) \quad e^{-\frac{d^2(C_n^i, \tilde{C}_n^i)}{2\Delta_r}}. \tag{26}$$

The entire Deform PF-MT is summarized in Algorithm 1.

IV. Parameter Estimation: Offline and Online

In this section, we first discuss how to estimate the effective deformation dimension, K, and the rest of the system model parameters, using a training sequence (offline). Next, we discuss some ideas on how to deal with time-varying contour length or spatial deformation frequency.

A. Estimating K and Other System Model Parameters: Offline

Given a training sequence of contours, we first need to compute the translation and the deformation. This is used to estimate the effective deformation dimension, K, as explained below. For a given K, the algorithm for estimating the rest of the system model parameters $(A_s, A_\rho, \Sigma_s, \Sigma_\rho, \Delta_r)$ uses standard maximum likelihood estimation techniques. The complete algorithm is given in Algorithm 4.

To estimate K, we treat the deformation at time n, $v_n(p)$, as a spatial signal, with p (arclength or radial angle) being the spatial axis. Given a contour sequence, $C_0, C_1, \ldots C_T$, we first estimate the deformation sequence, $v_1(p), \ldots v_n(p), \ldots v_T(p)$, at points $p = p_1, \ldots p_m, \ldots p_{M_{n-1}}$ as explained in the first two steps of Algorithm 4. Note that $v_n(p)$ is computable only at points where C_{n-1} intersects the pixel grid, i.e., the p_m's are nonuniformly spaced. Thus, we need to resample all the $v_n(p_m)$'s to a fixed number, M, of uniformly spaced points before computing their discrete Fourier transform (DFT). If $p_{\min} := \min_k |p_{k+1} - p_k|$ is the minimum parameter spacing between any two consecutive points, and L_{mean} is the average contour length (for radial parametrization, $L_{\text{mean}} = L = 2\pi$), we use $M = \lceil L_{\text{mean}}/p_{\min} \rceil$. After the uniform resampling, the

Algorithm 5 Deform PF-MT-TV: PF-MT for a Time Varying Effective Basis. Going from π_{n-1}^N to $\pi_n^N(X_n)$, $X_n^i = [C_n^i, \rho_n^i, v_{n,s}^i, v_{n,r}^i]$

```
1) Run Algorithm 1.
2) Detect Effective Basis Change. Detect if effective basis change is required as explained in Sec. IV-B. If yes, go to step
   3, else set $n \longleftarrow n+1$ and go to step 1.
3) Change Effective Basis. For each particle, $i = 1, 2, \ldots, N$, do
   a) Compute $K_{new} = L/\alpha_s$ where $L$ is the length of the contour particle with highest weight.
   b) Reallocate the knots uniformly, i.e. set $\underline{s}_{j\,new}^* = \frac{(j-1)L(C_n^i)}{K_{new}}, j = 1, \ldots K_{new}$ and evaluate the new effective basis
      $B_{s,new}(C_n^i) \triangleq B_{s,new}^i$ using (13). Let the old basis $B_s(C_n^i) \triangleq B_s^i$.
   c) Re-project $v_{n,s}^i$ into the new effective basis, i.e. compute $v_{n,s,new}^i = ({B_{s,new}^i}^T B_{s,new}^i)^{-1} {B_{s,new}^i}^T B_s^i v_{n,s}^i$.
   d) Compute ${\underline{x}^*}_{n,new}^i$ for $\underline{s}^*_{new}$ and $C_n^i$ using (15).
   e) Set $v_{n,s}^i \longleftarrow v_{n,s,new}^i$ and ${\underline{x}_n^*}^i \longleftarrow {\underline{x}^*}_{n,new}^i$. Set $n \longleftarrow n+1$ and go to step 1.
```

spatial axis spacing of the consecutive elements of v_n is L/M where $L = L(C_{n-1})$ ($L = 2\pi$ for radial).[4]

From our system model, given in (3)–(7), the deformation is temporally stationary and ergodic. Thus, we can estimate its power spectral density (PSD) [37, p.282] as the average (over the sequence) of the squared DFT magnitudes of $v_n(p)$ weighted by M. We need to average over multiple squared DFT magnitudes since $v_n(p)$ can be spatially nonstationary, e.g., often one region of the contour deforms more than the others. We compute $f_{\min}$ as the smallest frequency for which the maximum PSD value outside $f_{\min}$ is smaller than $\Delta_{\max}$ (this is sometimes called the ϵ-bandwidth with $\epsilon \equiv \Delta_{\max}$). For this bandwidth, and assuming perfect interpolation, by Nyquist's criterion, the maximum distance between basis points needs to be smaller than $\alpha_s = (1/2f_{\min})L/M$. This corresponds to $K = (L/\alpha_s) = M.2f_{\min}$. The above algorithm is summarized in step 3 of Algorithm 4.

For spatially nonstationary deformation, one can either use the strategy described above or assume a model for piecewise spatial stationarity, estimate the boundaries of the different stationary regions, and compute α_s and K for each such region separately. If this can be done accurately, it will be more efficient, e.g., if a large region deforms very little, it can be assigned $K = 0$. We will experiment with this in future work.

B. Dealing With Slow Time-Varying Effective Basis: Online

The effective deformation dimension, K, needs to be large enough so that $\Delta_r < \Delta_{\max}$ (needed to apply PF-MT). For the arclength parametrization, there is a need to change effective basis if either (a) total length or arclength distance between basis points (knots) changes, or (b) the spatial frequency response of the deformation changes and consequently the required distance between basis points, α_s, changes. For radial parametrization, the total angular "length" is always $L = 2\pi$ and also the basis points always remain $2\pi/K$ angular distance apart, i.e., (a) does not occur.

[4]For arclength parametrization, the resampling amounts to shrinking/stretching the spatial axis length from $L(C_{n-1})$ to L_{mean}, followed by uniformly sampling $L(C_{n-1})/M$ arclength distance apart. This changes the frequency content of v_n and, hence, is valid only for a set of approximately equal length contours (small shrinking/stretching).

Change in Total or Local Contour Arclength. Consider the arclength parametrization. Assume that the spatial frequency remains constant and that we know the maximum allowable value of the distance between consecutive basis points, α_s, to ensure that $\Delta_r < \Delta_{\max}$. As the contour deforms, the arclength distance between consecutive basis points, and also its total length, change. There is a need to change effective basis if this distance becomes significantly smaller (starting to estimate noise) or significantly larger (residual deformation too large) than α_s.[5] This is done as follows. Choose $\alpha_{s,\min} < \alpha_s$ and $\alpha_{s,\max} > \alpha_s$.[6] We declare a need to change effective basis, whenever the following occurs for "most" (more than 50%) of the contour particles: the arclength distance between any two consecutive basis points exceeds $\alpha_{s,\max}$ or goes below $\alpha_{s,\min}$. We evaluate the new effective basis as follows: Compute $K_{\text{new}} = \lceil L/\alpha_s \rceil$ where L is the length of the maximum weight contour particle. Uniformly allocate the K_{new} basis points on the arclengths of all contour particles, i.e., set $\underline{s}^*_{\text{new,j}} = ((j-1)L(C_n^i))/(K_{\text{new}}), j = 1, \ldots K_{\text{new}}$ and compute the new effective basis functions using (18) for each contour particle. When the effective basis changes, the old effective basis velocity coefficients' vector, $v_{n,s}^i$ needs to be projected into the new basis. This is done by interpolating $v_{n,s}^i$ using the old effective basis, and then computing a least squares estimate of the coefficients in the new basis, i.e.,

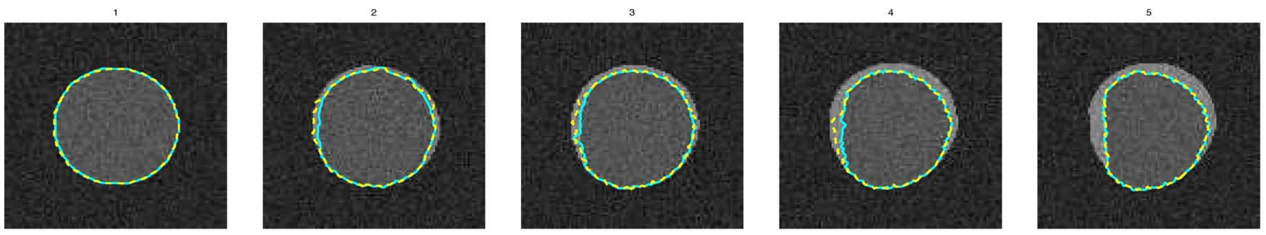
$$v_{n,s,\text{new}}^i = \left({B_{s,\text{new}}^i}^T B_{s,\text{new}}^i\right)^{-1} {B_{s,\text{new}}^i}^T B_s^i v_{n,s}^i \quad (27)$$

where $B_{s,\text{new}}^i = [B_{s,\text{new}}(C_n^i)]_{M_n \times K_{\text{new}}}$ denotes the new effective basis and $B_s^i = [B_s(C_n^i)]_{M_n \times K}$ denotes the old one. One also needs to compute ${\underline{x}^*}_{n,\text{new}}^i$ for $\underline{s}^*_{\text{new}}$ using (15).

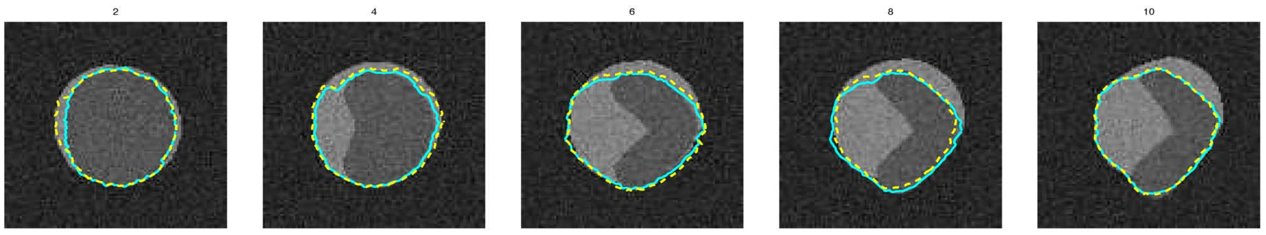
Deform PF-MT with a Time-Varying effective basis (Deform PF-MT-TV) is summarized in Algorithm 5.

[5]PF-MT only requires Δ_r to be small enough which only translates to an upper bound on the distance between consecutive basis points, α_s. However, in practice, if the distance between basis points becomes too small, the PF starts estimating noise (demonstrated in Fig. 4). Thus, distance becoming too small also needs to be detected and corrected.

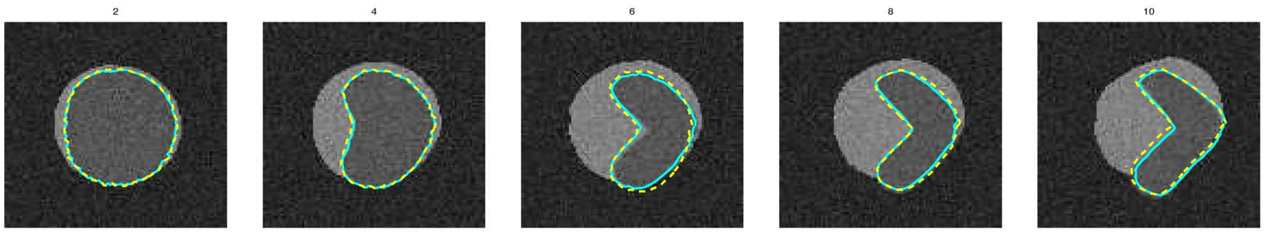
[6]These are chosen heuristically to ensure that the arclength distance between two knot locations does not become too much larger or smaller than α_s. To be perfectly accurate, one should change effective basis every time the distance between any two knots exceeds α_s ($\alpha_{s,\min} = \alpha_s = \alpha_{s,\max}$), but that would require too many basis changes. Also, if $\alpha_{s,\max}$ is too large, knots may go too far (miss tracking important deformation) and if $\alpha_{s,\min}$ is too small, knots may come too close (estimate noise), which is also undesirable.



(a)



(b)



(c)

Fig. 1. *Tracking through two nonaffine contour modes of the likelihood and avoiding distraction by false edges (radial parametrization).* (a) Nonaffine deformation per frame of the dark gray object is small and, hence, affine PF-MT [22] is able to track it. (b) Nonaffine deformation per frame is large and, hence, Affine PF-MT [22] fails even with increasing the gradient descent iterations. Deform PF-MT [Fig. 1(c)] works. The sequences had 15 frames. (a) Small nonaffine deformation. Affine PF-MT [22] with $G = 6$ gradient descent iterations in Mode Tracking (MT) step: works. (b) Large nonaffine deformation. Affine PF-MT [22] with $G = 12$ descent iterations in MT step: fails (similar result obtained with $G = 2$ or $G = 6$ iterations also). (c) Large nonaffine deformation. Deform PF-MT (Algorithm 1) with only $G = 2$ descent iterations in MT step: works (also works with $G = 6$).

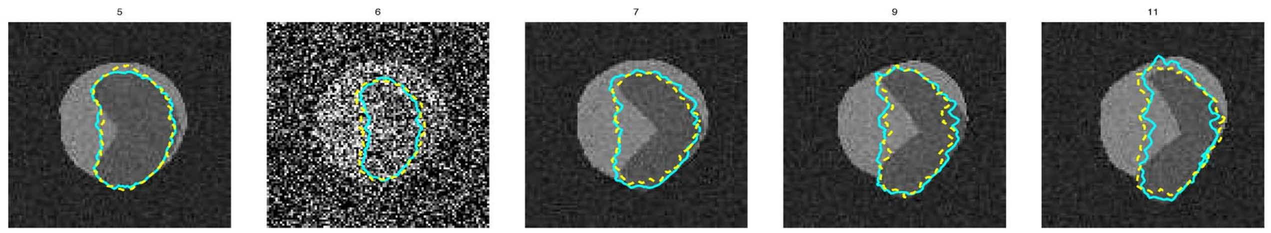
Change in Spatial Frequency Response. If (b) occurs, i.e., if the spatial frequency response of the contour deformation changes over time, it will require re-estimating α_s and, hence, K. This can be done as follows. At each n, use a few previously tracked deformation vectors (i.e., use the PF-MT estimate of $\mathbb{E}[v_l \,|\, Y_{1:l}]$ for $l = n - \tau, \ldots n$) and estimate α_s and K as described in Section IV-A. If α_s changes significantly from its current value, then there is a need to change K. Algorithm 5 can then be applied for arclength parametrization. For radial parametrization, in Algorithm 5, we will need to use $L = 2\pi$, compute $B_{s,\text{new}}$ using (11), and remove step 3d.

V. Simulation and Experimental Results

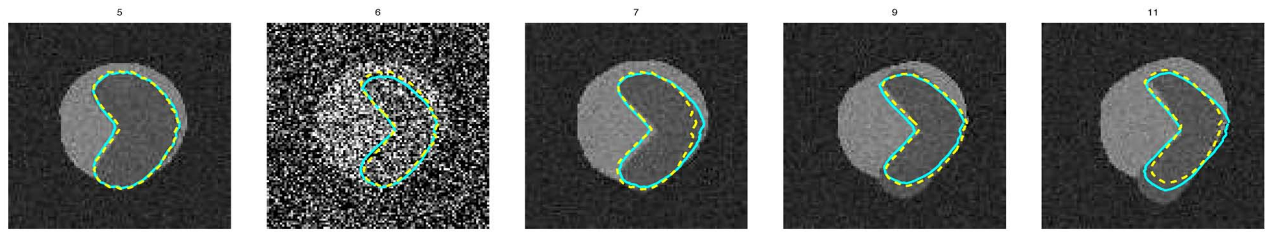
Since the posterior can be multimodal, plotting the posterior "mean" contour is not useful. Also, on the space of contours, there is no clear definition of a "mean" contour. In all the figures, we plot two contours with the largest posterior (largest weights). The largest weight contour is shown as a solid cyan line, and the second largest one as a dashed yellow line.

In Figs. 1, 2, and 3, we show simulated examples of situations where Affine PF-MT [22] will fail but Deform PF-MT (Algorithm 1) will work. *Because Deform PF-MT importance samples on the space of local deformations, it is able to avoid distractions by false nearby edges (Fig. 1), false nearby edges and similar nearby color (Fig. 2) and similar color nearby objects (Fig. 3).* In Fig. 4, we demonstrate the need to change effective basis (the need to use Algorithm 5) as contour length changes for the arclength parametrization. Figs. 1 and 2 use the radial parametrization, while Figs. 3 and 4 use the arclength parametrization. We demonstrate the ability to track through partial occlusions of a car sequence in Figs. 6 and in 7, we show tracking of a brain tumor boundary through sequential MRI slices. In Section V-C, we discuss the computational cost and parallelization ideas.

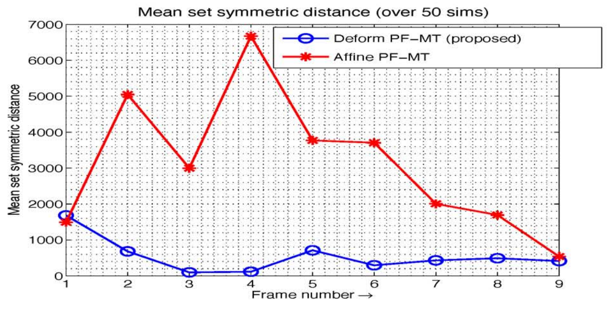
In all simulations, we assume known initial state, i.e., known location of the contour and its velocity at the initial time, $n = 0$. The initial contour estimation is the segmentation problem which is well studied in literature. The purpose of doing this is to only evaluate only the dynamic tracking performance and not incorrect initial state handling. For simulation sequences, we use the simulated value of the contour as the initial contour. For real sequences, the initialization is done as in previous work [22]: We manually mark out a few (5–6) initial points and use the resulting contour as the initial guess to perform Chan-Vese segmentation [31]. Note that our algorithm is *not sensitive* to the choice of initial points as long as they mark out the correct object's region and Chan-Vese is able to segment out the object.



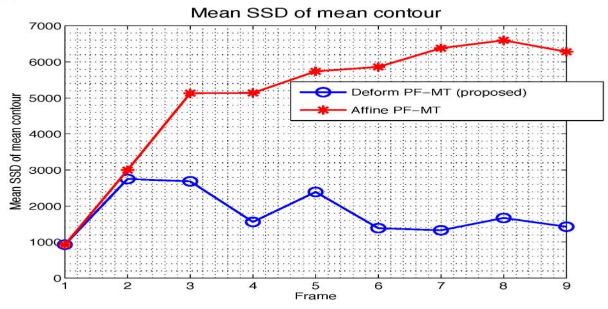
(a)



(b)



(c)



(d)

Fig. 2. *Tracking through outlier noise (radial parametrization).* Starting at $n = 6$, every even frame was an outlier similar to frame 6 shown in the second column. (a) Affine PF-MT [22] loses track. Here, we ensured that frame 5 remains in track (done by removing gradient descent towards edge energy) and use more descent iterations, but Affine PF-MT still loses track. Increasing descent iterations seems to only worsen the loss of track (contour attracted more towards outlier). (b) Deform PF-MT (Algorithm 1) remains in track until $n = 9$, very slight loss of track after that. The sequence had 15 frames. (c), (d) We used the method described in Section V-A2 to generate 50 realizations of contour sequences and the corresponding images (outlier noise added at every even frame for $n > 5$), i.e., we generated sequences similar to the first two rows. For each sequence, we tracked with Deform PF-MT and Affine PF-MT and computed the set symmetric distance of the MAP contour (contour particle with largest weight) from the ground truth (simulated contour) and averaged this distance over the 50 realizations. This is plotted in (c) as a function of time (frame number). In (d), we plot the average set symmetric distance of the "mean" contour from the ground truth. The "mean" contour is computed by resampling the x and y coordinates of all contour particles to a fixed number of points (use the arclength parametrization) and then computing their Euclidean mean. (a) Affine PF-MT [22] with $G = 6$ gradient descent iterations in Mode Tracking (MT) step and ensuring that frame 5 is in track: fails. (b) Deform PF-MT (Algorithm 1) with $G = 2$ descent iterations in MT step: works. (c) Average set symmetric distance of MAP contour particle from truth. (d) Average set symmetric distance of "mean" contour from truth.

A. Simulated Sequences

We demonstrate the ability of Deform PF-MT to track through various types of multimodalities in the image likelihood. We also demonstrate the need to change effective basis dimension when contour length changes.

1) Distraction by False Edges: In Fig. 1, the object-of-interest is the foreground dark gray object. False edges generated by the light gray background object result in a second contour mode of the likelihood. The two modes are separated by nonaffine deformation. In Fig. 1(a), the nonaffine deformation of the object-of-interest is small (slowly deforming) and so it is correctly tracked using Affine PF-MT [22]. However, in Fig. 1(b)–(c), the object-of-interest has large nonaffine deformation per frame (fast deforming) and so Affine PF-MT fails.

We simulated the image sequences as follows. The contour of the background (light gray) object was simulated by starting with a circle at $n = 0$ and using the system model described in (4)-(7) with $K = 6$ (corresponds to $\alpha_s = 2\pi/6$) to move and deform it. We used $\Sigma_s = I_K$, $A_s = 0.5I_K$ and $\Sigma_\rho = 0.25I_2$. The residual deformation was set to zero while simulating, i.e., we set $C_n = \tilde{C}_n$. The contour of the dark gray deforming object (object of interest) was also simulated with a model similar to the one above with the difference that a nonzero drift term, μ, was added in (5). We used $\mu = [0\ 0\ 0\ 0\ 0\ 1]^T$ in Fig. 1(a) (small deformation per frame) but we used $\mu = [0\,0\,0\,0\,0\,2]^T$ in Fig. 1(b)–(c) (large deformation per frame). This introduced a nonzero bias in the velocity dynamics near the sixth knot (basis point), resulting in inward motion of that region with nonzero average velocity at any n. The intensity of each pixel inside the contour of the object of interest (dark gray) was taken to be i.i.d Gaussian distributed with mean $u_1 = 85$ and variance $\sigma^2_{\text{obs},r} = 100$. The mean intensity of the background (light gray) object was $v_1 = 130$ and variance was $\sigma^2_{\text{obs},r} = 100$. The outer (black) background had mean intensity, $v_2 = 45$ and variance, $\sigma^2_{\text{obs},r} = 100$.

We compared the tracking of the dark gray object using Deform PF-MT [Fig. 1(c)] with that using Affine PF-MT [22]

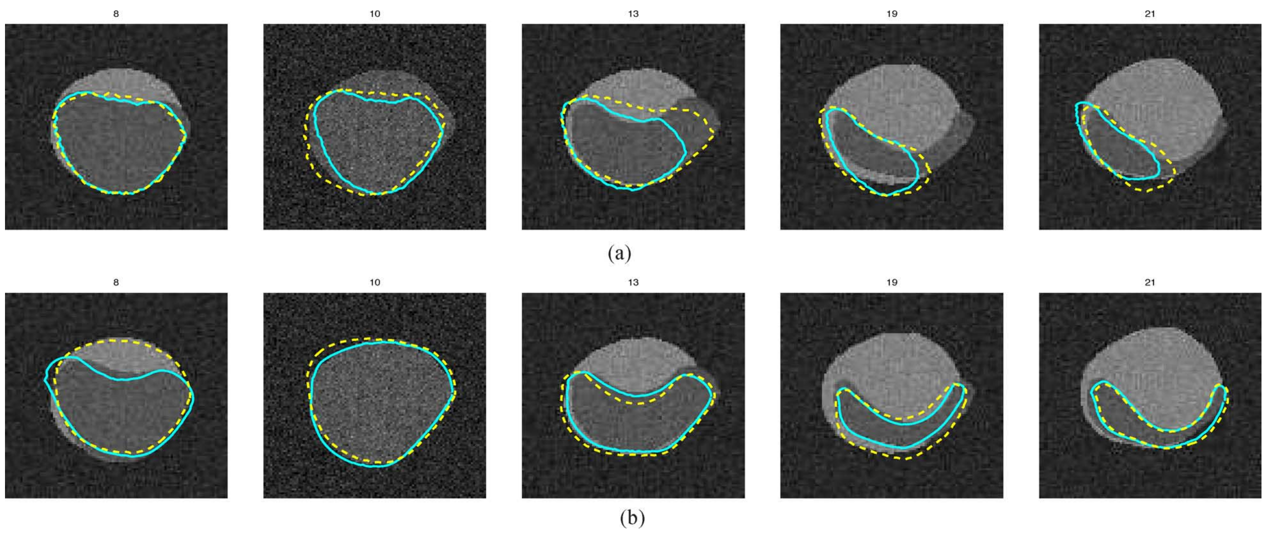


Fig. 3. *Tracking through simulated occlusion (arclength parametrization with $K = 6$ basis points)*. Every even frame beginning $n = 10$ was occluded by an object with the same intensity as the background object. The sequence had 21 frames. (a) Affine PF-MT with $G = 4$ gradient descent iterations: fails, (b) Deform PF-MT (Algorithm 1, parametric): works. First three plots use $G = 0$ descent iterations. Last two plots use $G = 1$.

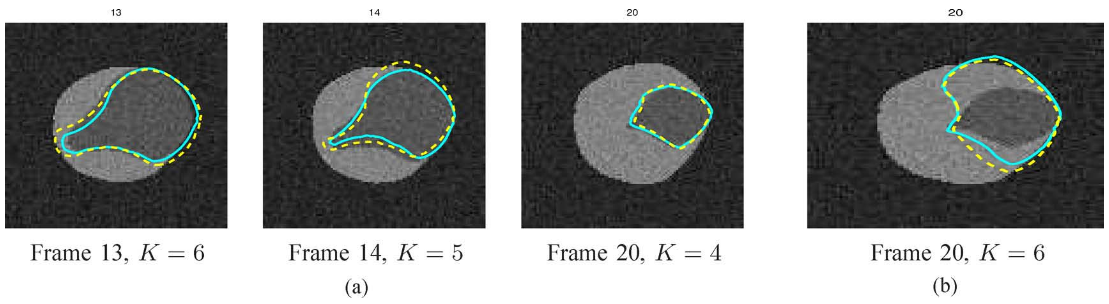


Fig. 4. *Need to change effective basis*. The dark gray object deforms and keeps reducing in size which requires reducing K. (a) Tracked using Deform PF-MT-TV (Algorithm 5). The tracking is not perfect because only $\mathrm{N} = 15$ particles were used. (b) We show what happens if we keep tracking using Deform PF-MT with $K = 6$. Some contours develop self intersections resulting in zero weight assigned to them (not shown). The ones that survive are those which did not self-intersect because they started expanding instead (shown). (a) Tracking using Deform PF-MT-TV. (b) Tracking using Deform PF-MT.

[Fig. 1(b)]. We used $N = 45$ particles in both cases. Observation likelihood was defined as explained in Section II-C. The edge term was taken from [6]. It had two equally strong modes at the background and foreground objects. The region term was the Chan-Vese model [31], [22] with one difference: the background could have two possible intensities v_1 and v_2. This is explained in the Appendix. It had a strong mode only at the object of interest. Thus, the combined likelihood had a strong mode at the object of interest and a weaker mode at the background object. When tracking with Deform PF-MT, we used all simulation parameters with the exception that we set $\mu = 0$. Instead, to track the nonzero bias in the velocity, we increased the system noise variance of the sixth knot to 5, i.e., $\Sigma_s = \mathrm{diag}([1\ 1\ 1\ 1\ 1\ 5])$ was used for tracking. Residual deformation was tracked as explained in the Mode Tracking step (step 2a) of Deform PF-MT with $G = 2$ Gradient Descent iterations in Fig. 1(c) (similar result obtained with $G = 6$ also). As can be seen, just 2 iterations suffice.

In Fig. 1(b), we track the same sequence using Affine PF-MT [22]. This used the space of affine deformations as the effective basis and all nonaffine deformation was treated as "residual deformation" (tracked using a method similar to Mode Tracking of Algorithm 1). We show results with $G = 12$ descent iterations (similar results were obtained even with $G = 2$ or $G = 6$). Since there are two distinct modes with roughly the same affine deformation (w.r.t. a circle) and nonaffine deformation per frame is large, the contours get stuck to the wrong mode in the Mode Tracking step. Increasing descent iterations does not help. We would like to clarify that we did not learn the affine deformation parameters (and, hence, it is not a fair comparison), but we changed the values a number of times until best possible results were obtained.

2) Distraction by Outliers and Quantitative Evaluation: The sequence of Fig. 2 was generated exactly as above, but with large outlier noise added to the image intensity in all even frames starting at $n = 6$. Outlier observations similar to the one shown in the second column of Fig. 2 were simulated by increasing the observation noise to $\sigma^2_{\mathrm{obs},r} = 10000$ in these frames. This results in both false nearby edges and false nearby similar intensity regions. Before and at $n = 5$, the dark gray object is well approximated by affine deformation of a circle and, hence, is in track using both algorithms (if enough descent iterations are used for Affine PF-MT [22]). However, at and after $n = 6$, Affine PF-MT gets stuck in the wrong mode due to the outlier observation. Since it does not importance sample on local deformation, it is unable to get back to the correct mode before

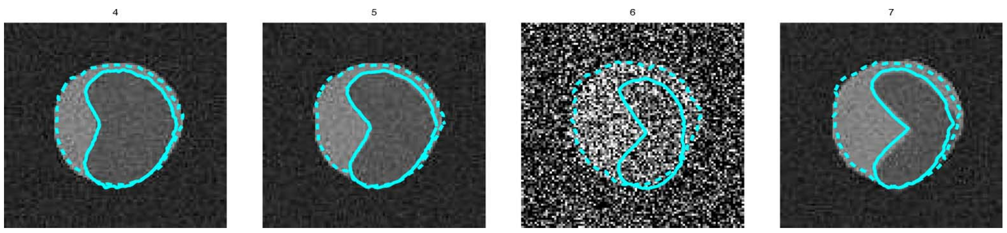


Fig. 5. Example of tracking both foreground and background object simultaneously using Deform PF-MT (same sequence as in Fig. 2).

the next outlier appears (outliers appear every alternate frame). Increasing gradient descent iterations only worsens the loss of track. On the other hand, Deform PF-MT is able to get back to the correct mode and remain in track [Fig. 2(b)].

We simulated 50 realizations of the above sequence and tracked them using both Affine and Deform PF-MT. Average set symmetric distance of the MAP (largest weight) contour from the ground truth is plotted for both algorithms in Fig. 2(c) and (d). We also plot the average set symmetric distance of the posterior "mean" contour. The "mean" was computed by resampling the x and y coordinates of all contour particles (treated as a function of arclength) to a fixed number of points and computing the Euclidean mean. This is a valid "mean" only if the contours have small differences in length or in local arclength.

3) Distraction by Similar Colored Occlusion: The contour and image sequence of Fig. 3 was generated as above, with the difference that now every even frame starting at $n = 10$ was an "occlusion" by an object of the same color as the object-of-interest—see the second column in the figure. A second difference was that the arclength parametrization was used here (the previous two figures used the radial parametrization). Here again Affine PF-MT gets distracted by the similar colored occluder and is unable to come back in track, while Deform PF-MT is able to.

4) Need for a Time-Varying Effective Basis: In Fig. 4, we demonstrate the need to change the effective basis for the arclength parametrization when contour length changes. We used $\alpha_s = 35$ here. In the sequence shown, the contour length keeps reducing because of a positive inward drift added to $v_{n,s,3}$ and $v_{n,s,5}$ (simulated by using a nonzero drift $\mu = [0\ 0\ 1\ 0\ 1\ 0]$ added to $v_{n,s}$). In the simulated sequence, K reduces from 6 to 4 at $n = 15$ and to 3 at $n = 23$. While tracking using Deform PF-MT-TV (Algorithm 5), we detected the need to reduce K from 6 to 5 at $n = 14$, from 5 to 4 at $n = 17$ and to 3 at $n = 23$. The tracking results are shown in Fig. 4(a). In Fig. 4(b), we show what happens if we do not allow change in K, i.e., we track with Deform PF-MT with $K = 6$ and not allowing effective basis change. When the knots come too close, the velocity samples at these points often erroneously result in a contour with self-intersections (which breaks). Whenever a contour breaks, it gets assigned zero weights in our algorithm (since we do not allow topology changes). Thus the only particles that survive are those which started expanding erroneously.

5) Tracking Multiple Contours: As suggested by an anonymous reviewer, we show results for simultaneously tracking both the foreground and background object in Fig. 5. The key idea of our algorithm is to apply an idea similar to the Adaptive PF proposed in [38] in combination with Deform PF-MT to track the different objects. We introduced this idea in [39]. More work is needed to improve the resulting algorithm. This will be part of future work.

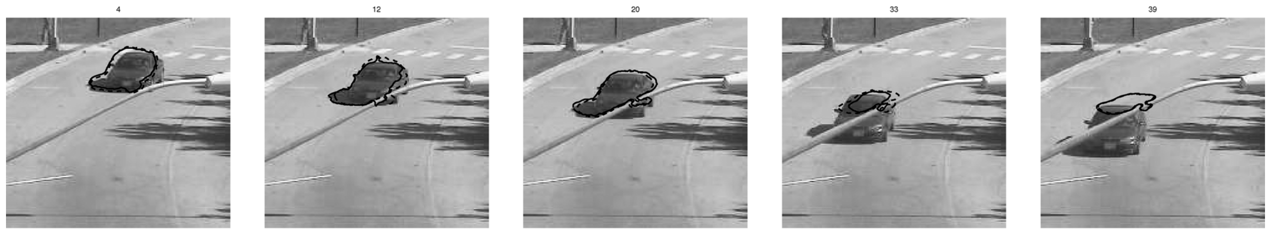
B. Real Sequences: Car and Brain MRI Sequences

Fig. 6 shows a moving car going under a street pole which partially occludes it for some frames. One may want to track the full car or track the portion to the left of the pole or the right portion of the car. We demonstrate the first two cases. The left part of the car was tracked by using $v_1 = \text{average intensity of pole}$, $v_2 = \text{average intensity of road}$ and $u_1 = \text{average intensity of the car}$[7] in the region-based term of the likelihood defined in (31) in the Appendix. For tracking the full car, we used, $v_1 = \text{average intensity of road}$ but $u_1 = \text{average intensity of the car}$ and $u_2 = \text{average intensity of pole}$.

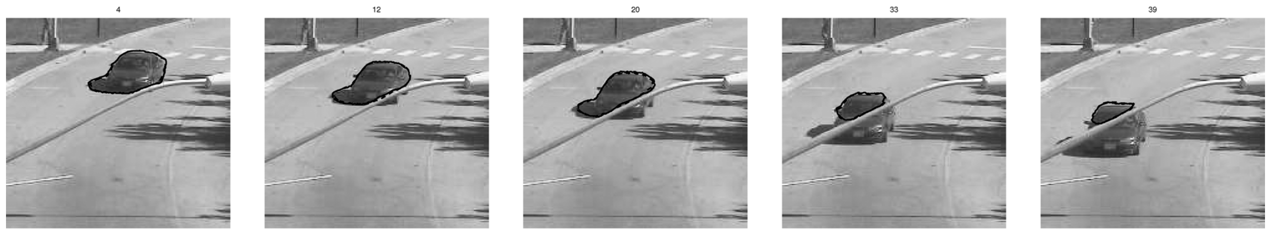
Tracking results for the left car using Affine and Deform PF-MT are shown in Fig. 6(a) and (b), respectively. Full car tracking is shown in Fig. 6(c) and (d). The left part of the car has significant nonaffine deformation over time and, hence, Affine PF-MT [22] fails [Fig. 6(a)] while Deform PF-MT [Fig. 6(b)] is able to track it. The full car tracking using either algorithm is similar, with our method being slightly better. Note that, to actually compare the power of Affine PF-MT with that of Deform PF-MT, we have implemented Affine PF-MT [22] also *without* the occlusion handling method. Hence, it is also about as inaccurate as Deform PF-MT. The results of Affine PF-MT shown in [22] are much more accurate because they include an occlusion handling heuristic (a car template matching term both in the mode tracking step and the particle weighting step).

Fig. 7 shows sequential segmentation of a set of MRI slices of different cross sections of the brain. We show results on segmenting brain tumor (gray-white region) in Fig. 7(a) and (b). The low contrast in the images results in a large number of weak observation likelihood modes, very near the true one. Note also that there is intensity variation across the sequence and, hence, the edge-based likelihood term helps remain in track. Both Affine and Deform PF-MT have similar performance. We were also able to sequentially segment the right ventricle (inside black region), but not very well (not shown). All of this was done

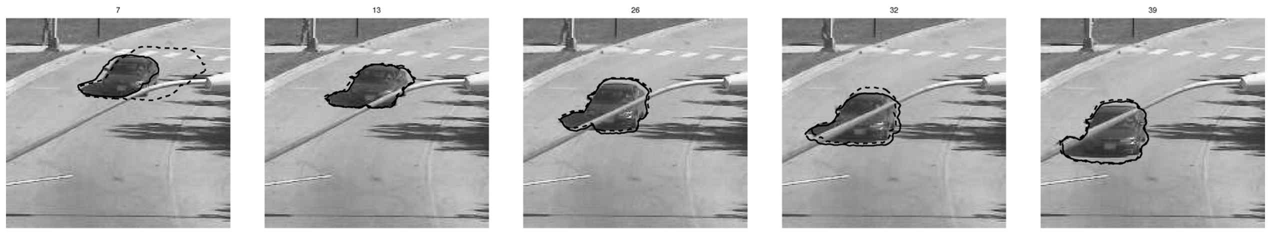
[7]The intensities are initially manually chosen by using the intensity of any one point in each of the regions. At future times, we automatically use the average intensity of the inside of the previous contour. Since the likelihood also contains a region term and an edge term, a precise knowledge of region intensities is not required. In general, they can also be chosen automatically using a more sophisticated segmentation technique taken from any state-of-the-art segmentation paper. However, since the observation likelihood choice is not the focus of this work, this has not been done.



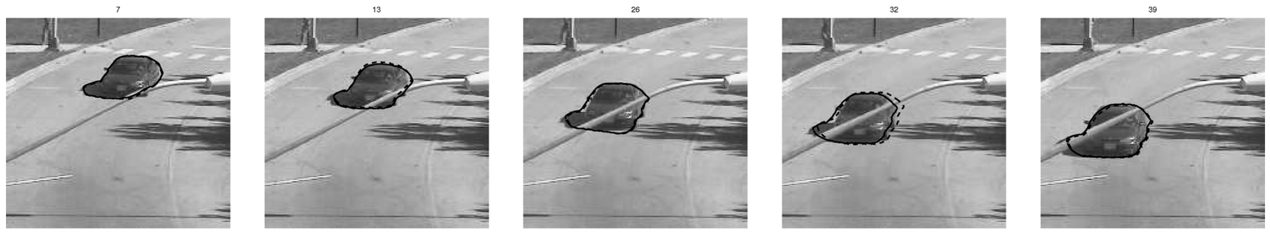
(a)



(b)



(c)



(d)

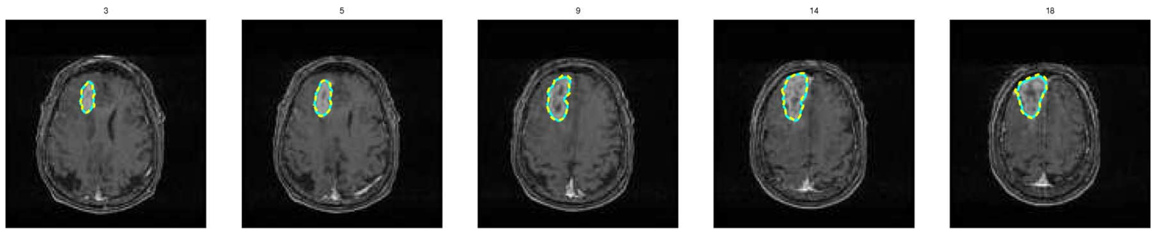
Fig. 6. *Tracking a car through partial occlusion by a pole*. For (a) and (b) (left part of car), $v_1 = \text{average intensity of pole}$, $v_2 = \text{average intensity of road}$ and $u_1 = \text{average intensity of the car}$. For (c) and (d) (full car), $v_1 = v_2 = \text{average intensity of road}$ but $u_1 = \text{average intensity of the car}$ and $u_2 = \text{average intensity of pole}$. The sequence had 40 frames. Note that the left part of the car has significant nonaffine deformation over time and, hence, Affine PF-MT [22] fails [Fig. 6(a)] while Deform PF-MT [Fig. 6(b)] is able to track it. The full car tracking using either algorithm is similar, with our method being slightly better. Note that here we have implemented Affine PF-MT *without* the occlusion handling heuristic of [22] (same for Deform PF-MT also). In this figure, black solid and black dashed lines denote the largest and second largest weight contours. (a) Tracking the contour for the car to left of the pole using Affine PF-MT [22]: fails. (b) Tracking the contour for the car to left of the pole using Deform PF-MT (Algorithm 1): works. (c) Tracking the full car using Affine PF-MT [22]. (d) Tracking the full car using Deform PF-MT (Algorithm 1).

without learning the contour dynamics and, hence, the poor segmentation of the ventricle. Learning its dynamics as explained in Section IV will help improve the results a lot.

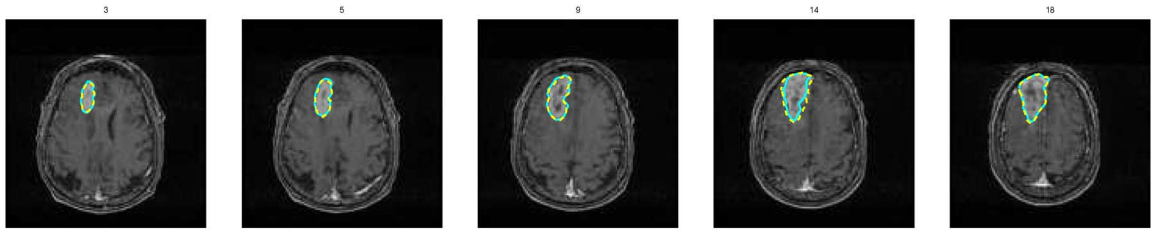
C. Computational Cost and Parallelization

The Deform PF-MT code is available at `http://www.ece.iastate.edu/~namrata/research/ContourTrack.html#code` [34]. The current implementation is an unoptimized MATLAB code (with many obvious inefficiencies) developed only for proof-of-concept. It takes roughly 20 min to track using $N = 45$ particles for $T = 20$ time instants, i.e., time taken per frame is 1 min. This was for a 102×102 image. We expect this to reduce significantly by just removing the inefficiencies in the MATLAB code. However, the more important point is that when we did a MATLAB Profiler analysis, it was observed that 99.7% of the time taken in running Deform PF-MT is spent in the importance sampling, mode tracking and weighting steps. However, note that *importance sampling, mode tracking and weighting are the steps that can be easily parallelized. In fact the parallelization is conceptually very simple*. At each time, t, one just needs to implement the same importance sampling, mode tracking and weighting on N processors. The outputs of all N processors are collected by the $N + 1$th processor which performs resampling and outputs the tracking result. At time $t + 1$, it sends the new observation and the resampled particles from time t to each of the first N processors. This parallel implementation (which is conceptually simple but requires very careful software engineering and programming) is part of ongoing work.

Thus, when the parallel version of Deform PF-MT is developed, we predict that the time taken by Deform PF-MT will reduce to almost $1/N$ of the time it takes to run a nonparallel op-



(a)



(b)

Fig. 7. *Tracking the tumor (gray-white region) in a brain MRI sequence*. Notice the low contrast imagery. Both Affine and Deform PF-MT have similar tracking performance. The sequence had 20 frames. Data obtained from Dr. T. Ryken (University of Iowa). (a) Tracking the tumor (gray-white region) through sequential MR slices of brain using Affine PF-MT [22]. (b) Tracking the tumor (gray-white region) through sequential MR slices of brain using Deform PF-MT (Algorithm 1).

timized C++ code, which itself will be faster than (or at least as fast as) the current unoptimized MATLAB version. Thus, the time taken should reduce to less than $(1-0.997).1+0.997.$ $(1/N)=0.025$ min per frame (1.5 s per frame). There will of course be a little extra overhead for the communication between the $N+1$th (resampling) processor and each of the N (importance sampling/weighting) processors.

VI. Conclusion and Open Issues

A new algorithm for tracking local contour deformations, called Deform PF-MT is developed. Deform PF-MT importance samples the x-y translation and "subsampled" deformation (followed by interpolating it onto the entire contour) while replacing importance sampling by deterministic posterior mode tracking (MT) for the rest of the deformation (residual deformation). A novel system model for contour deformation dynamics is proposed and the algorithm to importance sample from it using the level set method is developed. The effective basis can change with time. We discuss when and how to detect and estimate this change.

There are many open issues. First, the correct way to parameterize deformation is not clear as discussed in Section II-D. Radial angle (or any x-y location based) parametrization is easy to implement and handles topology changes, but there is a nonuniqueness problem if two or more points on the contour have nearly equal radial angle. The arclength parametrization handles this issue, but cannot deal with topology changes and is expensive to implement using the level set method. Second, the approaches for detecting and estimating the change in effective basis, described in Section IV-B, for both types of parametrization need to be studied in more detail. Two different approaches that will be pursued are discussed in [40] and [39].

Deform PF-MT can be made to run in real-time once its parallel implementation is developed, which, as we describe in Section V-C, is conceptually very simple to do. Another important practical issue is to improve the observation models, e.g., the region based observation model can be improved by also tracking object and background intensities as in [2], to allow for illumination variations over time. Finally, applications to medical image sequence segmentation problems, e.g., tracking different regions of an organ such as the brain or the heart, from an MRI sequence or from a more noisier ultrasound sequence, are currently being explored. A key potential application is in image guided surgery where the brain deforms when the skull is opened and there is a need to quickly segment the current brain slice where surgeon is operating. For most medical image sequences, large amounts of hand-segmented training data can be obtained. Current experimental results are without using the parameter estimation algorithm of Section IV, but learning can greatly improve the results.

Appendix
Explaining the Observation Model

The observation at time n, $Y_n=[Y_n^r, Y_n^e]$ where Y_n^r denotes the image and Y_n^e denotes the edge map (locations of all edges detected using Canny's method). The region based likelihood is a slight modification of the Chan-Vese model [31] and the edge term is similar to the model used in Condensation [6]. The observation model can be expressed as follows:

$$\begin{aligned} p(Y_n \mid C_n) &\\ &= p\left(Y_n^e \mid C_n\right) p\left(Y_n^r \middle| C_n\right), \text{ where} \qquad (28)\\ p\left(Y_n^e \middle| C_n\right) &\\ &\propto \exp\left[-E_{\text{ed}}\left(Y_n^e, C_n\right)\right], \text{ where} \qquad (29)\end{aligned}$$

$$E_{\rm ed} \triangleq \frac{1}{\sigma^2_{\rm obs,e}} \times \left[\sum_{j=1}^{M_n} \min\left[\rho, \min_{l=1,2,\ldots P_n} \|Y_n^e(l) - C_n(p_j)\|^2\right] + (P_n - M_n)^+ \rho\right] \quad (30)$$

$$p(Y_n^r|C_n) \propto \exp\left[-E_{\rm cv}(Y_n^r, C_n)\right], \text{ where}$$

$$E_{\rm cv} \triangleq \frac{1}{\sigma^2_{\rm obs,r}} \sum_{x_1=1}^{S_1} \sum_{x_2=1}^{S_2} \left[\chi_{C_n,\rm in}(x)\left(Y_n^r(x) - u_1\right)^2 + (1 - \chi_{C_n,\rm in}(x)) \min_{l=1,2}\left(Y_n^r(x) - v_l\right)^2\right] \quad (31)$$

where $S_1 \times S_2$ is the image size, P_n is the number of edges in the edge map and $\chi_{C_n,\rm in}(x)$ is the indicator function for the region inside the contour C_n. Other parameters are explained while we list the implicit assumptions of the above model.

1) Combined Likelihood: The image, Y_n^r, and the edge map, Y_n^e, are independent given C_n. This assumption is obviously not true but is a convenient way to combine both likelihoods.

2) Assumptions for the Edge Term (30): are as follows.

a) Edge points are generated either by the object contour or by clutter. Each contour point may either generate no edge(missed detection), or may generate any one of the edge points [6]. More than one contour point may generate the same edge.

b) The location of an edge point generated by a contour point is i.i.d. Gaussian distributed about the contour point with variance $\sigma^2_{\rm obs,e}$.

c) The edge points generated by clutter can occur anywhere in the image with uniform probability, $e^{-(\rho)/(\sigma^2_{\rm obs,e})}$. Since the image size is $S_1 \times S_2$ pixels, we get $\rho = \sigma^2_{\rm obs,e} \log_e(S_1 S_2)$.

d) To compute the exact edge likelihood, one would need to evaluate $(P_n + 1)^{M_n}$ likelihood terms corresponding to the different permutations linking a contour point to an edge point and sum them. This will be very expensive. The above model assumes that the observation likelihood given the most likely permutation is much larger than the rest. Thus, it implicitly assumes that (i) the distance of the edge point closest to a contour point is much smaller than that of any other edge point and (ii) when a contour point is missed (not detected), its distance from all edge points is much larger than ρ.

3) Assumptions for the Region Term (31): are as follows.

a) The image is separated into object and background region by the contour. The pixels in the object region are i.i.d. Gaussian distributed with mean intensity u_1, variance $\sigma^2_{\rm obs,r}$.

b) The background pixels are also i.i.d. Gaussian with variance $\sigma^2_{\rm obs,r}$, but have one of two possible mean values, v_1, v_2.

c) $|v_1 - v_2|$ is assumed to be sufficiently large compared to $\sigma_{\rm obs,r}$, so that either the likelihood using v_1 is much larger than that using v_2 or vice versa.

4) Spatially Dependent Observation Noise: The spatially independent observation noise assumption is never true in practice. For the region term, we instead assume in implementation, that $a \times a$ regions (e.g., $a = 4$) in the image have the same observation noise. For the edge term, we replace the summation over all M_n contour points by summation over a subsampled set of M_n/a points.

REFERENCES

[1] N. Vaswani, A. Yezzi, Y. Rathi, and A. Tannenbaum, "Time-varying finite dimensional basis for tracking contour deformations from image sequences," presented at the IEEE Conf. Decision and Control, 2006.

[2] W. Sun, M. Cetin, R. Chan, V. Reddy, G. Holmvang, V. Chandar, and A. Willsky, Segmenting and Tracking the Left Ventricle by Learning the Dynamics in Cardiac Images, MIT Tech. Rep. 2642, Feb. 2005.

[3] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/nongaussian bayesian state estimation," *IEE Proc. F*, vol. 140, no. 2, pp. 107–113, 1993.

[4] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[5] , A. Doucet, N. deFreitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. New York: Springer, 2001.

[6] M. Isard and A. Blake, "Condensation: Conditional density propagation for visual tracking," *Int. J. Comput. Vis.*, pp. 5–28, 1998.

[7] , A. Blake and M. Isard, Eds., *Active Contours*. New York: Springer, 1998.

[8] S. K. Warfield, Real-Time Image Segmentation for Image-Guided Surgery Surgical Planning Lab (SPL, Harvard) [Online]. Available: http://www.spl.harvard.edu:8000/pages/papers/warfield/sc98/index.html, webpage:

[9] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, pp. 321–331, 1987.

[10] D. Terzopoulos and R. Szeliski, *Active Vision*. Cambridge, MA: MIT Press, 1992, pp. 3–20, chapter Tracking with Kalman Snakes.

[11] N. Peterfreund, "Robust tracking of position and velocity with Kalman snakes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 6, pp. 564–569, Jun. 1999.

[12] A. Blake, M. Isard, and D. Reynard, "Learning to track curves in motion," presented at the IEEE Conf. Decision and Control, 1994.

[13] R. W. Brockett and A. Blake, "Estimating the shape of a moving contour," presented at the IEEE Conf. Decision and Control, 1994.

[14] A. 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.

[15] A. Yilmaz, X. Li, and M. Shah, "Contour-based object tracking with occlusion handling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, Nov. 2004.

[16] Y. Shi and W. C. Karl, "Real-time tracking using level sets," presented at the IEEE Conf. Computer Vision and Pattern Recognition, 2005.

[17] M. Niethammer and A. Tannenbaum, "Dynamic level sets for visual tracking," presented at the IEEE Conf. Decision and Control, 2004.

[18] J. Jackson, A. Yezzi, and S. Soatto, "Tracking deformable moving objects under severe occlusions," presented at the IEEE Conf. Decision and Control, 2004.

[19] D. Cremers, "Dynamical statistical shape priors for level set based tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, Aug. 2006.

[20] S. J. Osher and J. A. Sethian, "Fronts propagation with curvature dependent speed: Algorithms based on hamilton-jacobi formulations," *J. Comput. Phys.*, vol. 79, pp. 12–49, 1988.

[21] N. Vaswani, A. Yezzi, Y. Rathi, and A. Tannenbaum, "Particle filters for infinite (or large) dimensional state spaces-part 1," presented at the IEEE Int. Conf. Acoustics, Speech and Signal Processing, 2006.

[22] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi, "Tracking deforming objects using particle filtering for geometric active contours," *IEEE Trans. Pattern Anal. Mach. Intell.*, Aug. 2007.

[23] N. Vaswani, "Particle filtering for large dimensional state spaces with multimodal observation likelihoods," *IEEE Trans. Signal Process.*, Oct. 2008.

[24] M. Leventon, E. Grimson, and O. Faugeras, "Statistical shape influence in geodesic active contours," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2000, pp. 1316–1324, IEEE.

[25] A. Tsai, A. Yezzi, W. Wells, III, C. Tempany, D. Tucker, A. Fan, E. Grimson, and A. Willsky, "Model-based curve evolution technique for image segmentation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2001, vol. 1, pp. 463–468.

[26] G. Sapiro, *Geometric Partial Differential Equations and Image Processing*. Cambridge, U.K.: Cambridge Univ. Press, Jan. 2001.

[27] A. Yezzi and A. Mennucci, "Conformal metrics and true "gradient flows" for curves," presented at the IEEE Int. Conf. Computer Vision, 2005.

[28] J. A. Sethian, *Level Set Methods and Fast Marching Methods*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1999.

[29] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. New York: Springer Verlag, 2003.

[30] A. Papoulis, *Probability, Random Variables and Stochastic Processes*. New York: McGraw-Hill, 1991.

[31] T. Chan and L. Vese, "Active contours without edges," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 266–277, Feb. 2001.

[32] D. F. Rogers and J. Alan Adams, *Mathematical Elements for Computer Graphics*. New York: WCB/McGraw-Hill, 1990.

[33] A. Doucet, On Sequential Monte Carlo Sampling Methods for Bayesian Filtering Cambridge Univ., Dept. Engineering, Tech. Rep., CUED/F-INFENG/TR. 310, 1998.

[34] *Deform PF-MT Code*, [Online]. Available: http://www.ece.iastate.edu/~namrata/research/ContourTrack.html

[35] A. Yezzi and S. Soatto, "Deformotion: Deforming motion, shape average and the joint registration and approximation of structures in images," *Int. J. Comput. Vis.*, vol. 53, no. 2, pp. 153–167, 2003.

[36] D. Schuurmans, "Greedy importance sampling," presented at the Adv. Neur. Info. Proc. Syst., 2000.

[37] G. R. Cooper and C. D. McGillem, *Probabilistic Methods of Signal and System Analysis*, 3rd ed. Oxford, U.K.: Oxford Univ. Press, 1999.

[38] A. Papvasiliou, "A uniformly convergent adaptive particle filter," *J. Appl. Probab.*.

[39] N. Vaswani, "Particle filters for infinite (or large) dimensional state spaces-part 2," presented at the IEEE Int. Conf. Acoustics, Speech and Signal Processing, 2006.

[40] N. Vaswani, "Kalman filtered compressed sensing," presented at the IEEE Intl. Conf. Image Processing, 2008.

Namrata Vaswani received the B.Tech. degree from the Indian Institute of Technology (IIT), Delhi, in 1999, and the Ph.D. degree from the University of Maryland, College Park, in 2004, both in electrical engineering.

Since Fall 2005, she has been an Assistant Professor in the ECE Department, Iowa State University, Ames. Her research interests are in estimation problems in statistical and/or sequential signal processing and biomedical imaging. Her current focus is on large dimensional tracking and on recursive sparse reconstruction problems.

Dr. Vaswani is an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING.

Yogesh Rathi received the B.S. and M.S. degrees from the Birla Institute of Science and Technology, Pilani, India, and the Ph.D. degree from the Georgia Institute of Technology, Atlanta.

He is currently an Instructor at Harvard Medical School, Boston, MA. His research interests lie in applying control theoretic, mathematical, and machine learning concepts to solve problems in medical imaging and computer vision.

Anthony Yezzi received the B.S. and Ph.D. degrees from the Department of Electrical Engineering, University of Minnesota, with minors in mathematics and music.

He is currently a Professor in the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta. His research interests are in image processing and computer vision.

Allen Tannenbaum is a faculty member at the Schools of Electrical and Computer and Biomedical Engineering, Georgia Institute of Technology, Atlanta, and at the Department of Electrical Engineering, The Technion—Israel Institute of Technology, Haifa. He works in image processing and control.