Time-varying Finite Dimensional Basis for Tracking Contour Deformations

Namrata Vaswani*, Anthony Yezzi**, Yogesh Rathi**, Allen Tannenbaum**
*Dept. of ECE, Iowa State University, Ames, IA 50011, namrata@iastate.edu
**School of ECE, Georgia Tech, Atlanta, GA 30332

***Abstract*— We consider the problem of tracking the boundary contour of a moving and deforming object from a sequence of images. If the motion of the "object" or region of interest is constrained (e.g. rigid or approximately rigid), the contour motion can be efficiently represented by a small number of parameters, e.g. the affine group. But if the "object" is arbitrarily deforming, each contour point can move independently. Contour deformation then forms an infinite (in practice, very large), dimensional space. Direct application of particle filters for large dimensional problems is impractical, due to the reduction in effective particle size as dimension increases. But in most real problems, at any given time, "most of the contour deformation" occurs in a small number of dimensions ("effective basis") while the residual deformation in the rest of the state space ("residual space") is "small". The effective basis may be fixed or time varying. Based on this assumption, we modify the particle filtering method to perform sequential importance sampling only on the effective basis dimensions, while replacing it with deterministic mode tracking in residual space (PF-MT). We develop the PF-MT idea for contour tracking. Techniques for detecting effective basis dimension change and estimating the new effective basis are presented. Tracking results on simulated and real sequences are shown and compared with past work.**

I. Introduction

We would like to causally segment a moving/deforming object(s) from a sequence of images. This is formulated as a problem of tracking the boundary contour of the object, i.e. computing an "optimal" estimate of the state (contour and contour velocity) at the current time using all observations (images) until the current time. We denote the state at time t by X_t and the observation by Y_t. Any "optimal" state estimate can be computed once the posterior, $\pi_t(X_t) \triangleq p(X_t|Y_{1:t})$, is computed or approximated, e.g. MAP or MMSE. The general problem formulation is depicted in Fig. 1. The state dynamics is assumed to be Markovian. The observed image is assumed to be a noisy and possibly nonlinear function of the contour. The image likelihood given the contour ("observation likelihood") may be multimodal or heavy tailed. Since the state space model is nonlinear and multimodal, we study particle filtering(PF) [1], [2], [3] solutions to the tracking problem.

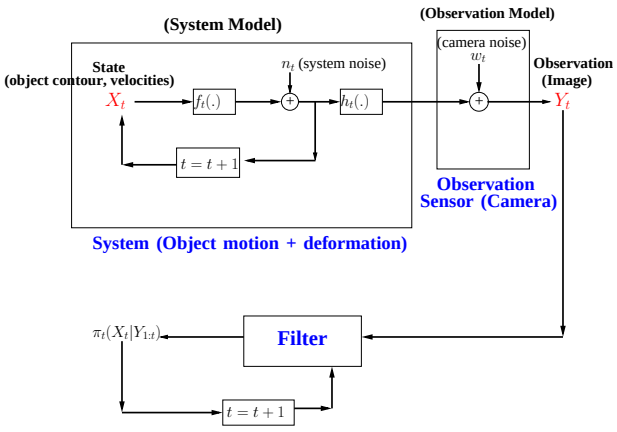


Fig. 1. Problem Formulation

A continuous closed curve (contour)[4] is the smooth locus of points traced out by the mapping of the unit interval into $\mathbb{R}^2$. *Deforming contours* occur either due to changing region of partial occlusions or when the object of interest is actually deforming its shape over a time or space sequence of images. An example of the first kind is shown in Fig. 5(a), where the contour representing the left part of the car deforms as it moves under the pole. Examples of the second kind are a beating heart, moving animals or humans, or the cross-sections of different parts of a 3D object like the brain, in consecutive MRI slices, e.g. Fig. 6. Most biological images contain deforming objects/regions. Contour tracking has many applications in medical image analysis, e.g. sequential segmentation of volume images (Fig. 6); tracking heart regions [5], [6] or image guided surgery.

The observation likelihood is often multimodal due to background objects (clutter) which are partially occluded by the "object of interest" (e.g. see Fig. 2) or due to an object which partially occludes the "object of interest" (e.g. the two contour modes shown in Fig. 5(a), 5(b)) or due to low contrast imagery (e.g. see Fig. 6 or [6]). Heavy tailed or multimodal observation likelihoods occur when the observation noise has occasional outliers (e.g. see Fig. 3).

Early work on contour tracking [7], [8], [9], [10] used the Kalman filter to track a fixed number of marker points [11] uniformly chosen on the initial contour or a fixed parametric representation, such as B-spline control points [8]. The Kalman filter can only handle additive and unimodal observation noise and so the observation needs to be an observed contour extracted from the image by searching in the vicinity of the predicted contour. The seminal work of [12] (Condensation) introduced particle filters (PF) [1], [2], [3] to tackle multimodal (and possibly nonlinear) observation likelihoods that occur due to clutter or occlusions. It allowed directly using the image (or the edge map) as the observation. But it only tracked on the 6-dim space of affine deformations.

Many recent works on contour tracking, e.g. [13], [14], [6]

use the level set representation [15], [16] of a contour and propose different types of approximate linear observers for contour deformation and/or for global motion. The level set method [15], [16] provides a way to implicitly represent and deform a continuous contour on a fixed pixel grid and thus automatically handles changes in contour length or topology. Specifically, [13] computes the current contour estimate as an approximate linear combination of the predicted contour and the observation likelihood mode nearest to it (and similarly for global motion). We call this general technique a Posterior Mode Tracker [17], since it can be understood as computing the mode, $\hat{X}_t$, of $p^*(X_t) \triangleq p(X_t|\hat{X}_{t-1}, Y_t)$ and approximating the posterior, π_t, by a Dirac delta function (δ) at $\hat{X}_t$. Setting, $\pi_{t-1}(X_{t-1}) \approx \delta(X_{t-1} - \hat{X}_{t-1})$, one can easily see that $\pi_t(X_t) \approx p^*(X_t)$ i.e. $\hat{X}_t$ is also the mode of the posterior. Thus it *implicitly assumes that the posterior is effectively unimodal* (has only one significant mode which is near $\hat{X}_{t-1}$). This may not hold when there are multiple distinct or overlapping objects.

In [18], we combined the ideas of [12] and [13] to handle more general situations. A PF was used to track affine deformations, while an approximate linear observer was defined to estimate the non-affine deformation for each affine deformed contour particle. In doing this, the *implicit assumption is that the posterior of non-affine deformation is unimodal*. This is valid for many practical problems shown in [18] where the non-affine deformation per frame is small, e.g. a rigid object tracked by a perspective camera with frequent viewpoint changes, or approximately rigid objects, e.g. human body contour from a distance. *But in other situations, where local deformation per frame are large, there may be more than one non-affine mode for the same affine deformation value and the same image, i.e. posterior of non-affine deformation may be multimodal.* This is demonstrated in Fig. 2 (overlapping objects separated by non-affine deformation) and in Fig. 3 (multiple modes due to outlier image and due to overlapping objects). Another example is the car sequence of Fig. 5, where one may want to either track the whole car or only the portion to the left of the street light (the two contour modes are separated by non-affine deformation). [18] tracked the full car by using a special occlusion handling method (which penalized deviations from a rigid car template).

A. Main Idea

To address the problems of [18], we need an importance sampling step [2] in the PF that also samples from the space of non-affine deformations. For deforming objects, each contour point can move independently and hence the contour deformation forms an infinite (in practice, very large), dimensional space. PF on such a large dimensional space is impractical due to the reduction in effective particle size [2] as dimension increases. But in most real problems, *at any given time, "most of the contour deformation" occurs in a smaller number of dimensions ("effective basis") while the deformation in the rest of the state space (space of "residual deformations") is "small". The effective basis may be fixed or time varying.* This is the "large dimensional state spaces (or LDSS)" property, introduced in [17], [19], applied to deforming contours. In other words, the deformation "signal" is approximately bandlimited (spatially), with the approximate cut-off frequency being much smaller than the maximum measurable frequency, 0.5Hz. This idea is detailed in [20].

Using the LDSS property, we proposed to modify the PF method to perform sequential importance sampling [2] only on the effective basis dimensions, while replacing it with deterministic mode tracking (MT) in residual space [17], [19]. In this work, we develop the PF-MT idea for contour tracking using global translation and deformation velocity at subsampled contour locations interpolated using a B-spline basis as the effective basis. Detecting change in effective basis dimension (when contour length or deformation frequency changes) and estimating the new effective basis is discussed. In practice, explicitly tracking local deformation (even with deformation velocity tracked at only $K = 6$ locations around the contour) is extremely beneficial as can be seen from the last rows of Figs. 2 and 3.

We stress the difference from Condensation [12] which uses B-spline control points to approximate the contour itself and hence requires many more sample points for accurate representation. This is because the maximum spatial frequency of deformation cannot be larger (is usually much smaller) than that of the two contours[1] from which it is computed. Also, our effective basis is similar to that of [21] which proposed an annealing based technique for segmentation. Another PF method that also improves effective particle size by reducing PF dimension is Rao-Blackwellization [22]. But it requires that a part of the state have a linear Gaussian state space model. PFs with time-varying dimension have been used in other contexts cited in [19].

The paper is organized as follows. We give the form of the state space model in Section II. The PF-MT algorithm for contour tracking and PF-MT-TV for dealing with time-varying effective basis is explained in Section III. Experimental results on simulated and real sequences are given in Section IV. Conclusions, open issues are given in Section V.

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, contour velocity) is denoted by X_t. A block diagram is shown in Fig. 1. The contour at t can be represented as $C_t = C_t(p) = [C_t^x(p), C_t^y(p)]$, $p \in [0, 1]$. The parametrization is not unique, i.e. all re-parameterizations of the parameter p of the form $\tilde{p} = f(p)$, where $f : [0, 1] \rightarrow [0, 1]$ is continuous and strictly monotonic, yield the same contour [4]. The outward normal to contour C_t at p is denoted by $\overrightarrow{\mathbf{N}}(C_t(p))$ or by $\overrightarrow{\mathbf{N}}_t(p)$. Denote the space of contours [23] by $\mathcal{S}$. Then the tangent space to $\mathcal{S}$ at C_t will be [23] the space of all non-tangential velocities (velocities along the normal to C_t at each point), since

[1]The radius of the osculating circle [4] as a function of arclength is treated as the contour "signal". Deformation (along the contour normal) is the difference of the two consecutive contour "signals". When performing a linear operation, new frequencies cannot be introduced.

tangential velocity only re-parameterizes the contour [4]. We use v_t to denote the vector of normal velocities. Also, we use the notation $\mathcal{N}(x;\mu,\Sigma) \triangleq \frac{1}{\sqrt{2\pi|\Sigma|}} e^{-x^T\Sigma^{-1}x}$. We represent the contour C_t using the level set method [15]. Contour dimension at t is denoted by M_t.

A. System Model

The state at any time t, consists of the contour, its normal deformation velocity, and the global translational velocity. Because of the LDSS property (described in Section I-A), "most of the contour deformation" occurs in a smaller number of dimensions, K, which form the "effective basis". Thus, we split v_t as $v_t(p) = B_s(p)v_{t,s} + B_r(p)v_{t,r} + \overrightarrow{\mathbf{N}}_t(p)^T\rho_t$ where B_s denotes the effective basis directions for contour deformation (with translation removed) while B_r denotes the basis spanning the residual space. $v_{t,s}$, $v_{t,r}$ denote the corresponding coefficients. $\rho_t \in \mathbb{R}^2$ denotes the global x-y translation vector. We use velocity at K subsampled locations interpolated onto the entire contour using B-spline interpolation functions as the effective basis. This is explained in Section II-B. We assume that ρ_t, $v_{t,s}$ follow a first order autoregressive (AR) model, while $v_{t,r}$ is assumed temporally independent.

Let the observations arrive every τ time instants, i.e. arrive at $t = n\tau$, $n = 1, 2, ...$ Denote $X_{n\tau}$ by X_n. The system dynamics of $X_n = [C_n, v_{n,s}, v_{n,r}, \rho_n]$ can be expressed as:

$$C_n = \tilde{C}_n + B_r(C_{n-1})v_{n,r}\overrightarrow{\mathbf{N}}(\tilde{C}_n) \quad (1)$$

$$\tilde{C}_n = C_{n-1} + [B_s(C_{n-1})v_{n,s} + \overrightarrow{\mathbf{N}}_{n-1}^T\rho_n]\ \overrightarrow{\mathbf{N}}_{n-1} \quad (2)$$

$$v_{n,s} = A_s v_{n-1,s} + \nu_{n,s}, \quad \nu_{n,s} \sim \mathcal{N}(0, \Sigma_s) \quad (3)$$

$$v_{n,r} = \nu_{n,r}, \quad \nu_{n,r} \sim \mathcal{N}(0, \Sigma_r) \quad (4)$$

$$\rho_n = A_\rho \rho_{n-1} + \nu_{n,\rho}, \quad \nu_{n,\rho} \sim \mathcal{N}(0, \Sigma_\rho) \quad (5)$$

where $B_s \triangleq B_s(C_{n-1})$ is defined by (6) or by (7). Note, $v_{n,r}$ is actually not part of the state vector (since no element of the next state, X_{n+1}, depends on $v_{n,r}$). The above discretization assumes that

Assumption 1: The observation interval τ is small enough (compared to $B_s v_{n,s} + B_r v_{n,r} + \overrightarrow{\mathbf{N}}_{n-1}^T\rho_n$) so that $\overrightarrow{\mathbf{N}}_{n-1}$ is also approximately normal to C_n.

B. Geometric and Parametric Effective Basis

Contour motion using the level set method is naturally implemented using a B-spline basis that parameterizes velocity of a contour point based on its location on the x-y plane *(geometric effective basis)*. There are many possible ways to define a geometric basis, e.g. see [21]. For example, one dimensional parameterizations can be obtained by using the turning angle (angle made by the tangent with the x axis) or the radial angle (angular coordinate of the contour point w.r.t. the centroid of the contour's inside region, $[\mu_n^y, \mu_n^y]$) as the parameter. In our implementations, we use the radial angle (angular coordinate of the contour point w.r.t. the centroid of the contour's inside region, $[\mu_n^y, \mu_n^y]$), as the parameter. A velocity sample $v_{n,s,j}, j = 1, 2, ..K$ is assigned to each angular region. For e.g., a four dimensional basis is obtained by allocating one velocity sample to one quadrant of the x-y plane and smoothing across quadrant boundaries using B-spline interpolation. Thus

$$B_s(C_n)(p) \triangleq B_s(\underline{\theta}^*)(\theta(C_n(p))), \text{ where}$$
$$\theta(C_n(p)) \triangleq \arctan[\frac{C_n^x(p) - \mu_n^x}{C_n^y(p) - \mu_n^y}] \quad (6)$$

and the vector $\underline{\theta}^*$ contains K basis points (called "knots" [24]) which are uniformly chosen at angular distance $\alpha_s = 2\pi/K$ apart. $B_s(\underline{\theta}^*)(p)$ contains the K B-spline basis functions for a closed cubic B-spline [24] with knots $\underline{\theta}^*$.

A geometric basis automatically handles changes in contour topology. But *it cannot be used if one would like to independently deform two or more points of a contour that have the same radial angle, but are far if one moves along the contour arclength.* Such applications can be handled by a *parametric effective basis* which parameterizes velocity based on the arclength [4] of the contour point, $s(C_n(p))$, w.r.t. an initial starting point. The basis points split the contour arclength into K regions and a velocity sample, $v_{n,s,j}, j = 1, 2, ..K$, is assigned to each region. We initially place the basis points ("knots") on the contour uniformly at $\alpha_s = L/K$ arclength distance apart where L is the contour length. As the contour deforms, the knots also move on the contour. Thus

$$B_s(C_n)(p) \triangleq B_s(\underline{s}(\underline{x}_n^*, C_n))(s(C_n(p))) \quad (7)$$

where $\underline{x}_n^*$ has components, $\underline{x}_{n,j}^*, j = 1, 2, ..K$ which denote the x-y location of the j^{th} knot. The vector $\underline{s}$ has components $\underline{s}_j, j = 1, ..K$ which denote the arclength location of the j^{th} knot w.r.t. a fixed starting point. Given a contour, C, there is an invertible mapping between $\underline{x}^*$ and $\underline{s}$. The forward mapping, $\underline{s}(\underline{x}^*)$ is: $\underline{s}_1(\underline{x}^*, C) = 0$ and

$$\underline{s}_j(\underline{x}^*, C) = \underline{s}_{j-1} + \frac{1}{L} arclen(\underline{x}_j^*, \underline{x}_{j-1}^*, C), \text{ for } j = 2, 3, ..K$$

$$arclen(\underline{x}_j^*, \underline{x}_{j-1}^*, C) \triangleq \sum_{m=m_{j-1}}^{m_j} ||C(p_m) - C(p_{m-1})||, \text{ where}$$
$$m_j \triangleq \arg\min_m ||\underline{x}_j^* - C(p_m)|| \quad (8)$$

Here $arclen()$ is the arclength [4] between the two consecutive knot locations, $\underline{x}_j^*, \underline{x}_{j-1}^*$. The inverse mapping is:

$$\underline{x}_j^*(\underline{s}, C) = C(\underline{s}_j). \quad (9)$$

The knot locations, $\underline{x}_n^*$ move along with the contour, i.e. for all $j = 1, 2, ..K$, they follow:

$$\underline{x}_{n,j}^* = \underline{x}_{n-1,j}^* + v^*(p_j)\overrightarrow{\mathbf{N}}(p_j),\ p_j \triangleq \underline{s}_j(\underline{x}_{n-1}^*, C_{n-1}) \quad (10)$$

$v^*(p) = B_s(p)v_{n,s} + \overrightarrow{\mathbf{N}}_{n-1}^T(p)\rho_n$ is the term inside [ ] on the right hand side of (2). After sometime, some knots may come "too close" to each other, while others may go "too far". This requires a change in effective basis (Section III-B).

The parametric basis is useful when change in topology is not allowed. In implementation, we can detect topology change of contour particles and assign a zero likelihood to particles for which topology change occurs.

C. Observation Model

The observation at time n, Y_n, is the image at n and the edge map derived from it. We assume that Y_n, depends only on C_n (and not on the velocity), i.e. the observation likelihood, $p(Y_n|X_n) = p(Y_n|C_n)$. Many observation models have been proposed - these can be classified as "region based", e.g. [25], [18], [6], or "edge based", e.g. [12] or "motion based", e.g. [10].*Using a good observation model is a critical issue, but we have not addressed it here.* In this paper, we use a product of the simple region-based observation likelihood of [18] which was motivated by the Chan and Vese model [25] and the edge-based observation likelihood proposed in Condensation [12]. This combines the advantages of a region based approach (robustness to blurred edges and 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 combined model is 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).

III. PF-MT-TV (Particle Filter with Mode Tracker and Time Varying Basis)

We first explain a generic particle filtering (PF) algorithm [1], [3], [2]. A PF outputs at each time n, a cloud of N particles, $\{X_n^i\}$ with weights $\{w_n^i\}$ whose empirical measure $\pi_n^N(X_n) \triangleq \sum_{i=1}^N w_n^i \delta(X_n - X_n^i)$ closely approximates the true posterior, $\pi_n(X_n) \triangleq p(X_n|Y_{1:n})$. Here $\delta(X-a)$ denotes the Dirac delta function at a. It starts with sampling N times from π_0 at $n = 0$ to approximate it by $\pi_0^N(X_0)$. For each $n > 0$, it approximates the Bayes recursion for going from π_{n-1}^N to π_n^N using importance sampling. This consists of:

1) *Importance Sampling (IS):* Sample $X_n^i \sim q(X_n^i)$, for $i = 1, 2...N$. The importance sampling density, q, can depend on X_{n-1}^i and Y_n.
2) *Weighting:* Compute the weights: $w_n^i = \frac{\tilde{w}_n^i}{\sum_{j=1}^N \tilde{w}_n^{(j)}}$, where $\tilde{w}_n^i = w_{n-1}^i \frac{p(Y_n|X_n^i)p(X_n^i|X_{n-1}^i)}{q(X_n^i)}$.
3) *Resampling:* Replicate particles in proportion to their weights & reset w_n^i [2]. Set $n \leftarrow n+1$ & go to step 1.

A. PF-MT (Particle Filter with Mode Tracker)

Since the effective particle size decreases with increasing system noise dimension, direct application of PF becomes impractical for large dimensional problems. *We propose to replace the PF by the following: importance sample only on the effective basis dimensions, and replace the importance sampling step by a deterministic Mode Tracking (MT) step in the residual space [17].* This idea, which we call PF-MT, assumes that the effective basis dimension K is large enough to ensure that Assumptions 2 and 3, given below, hold [17].

Assumption 2: The total residual deformation variance, $\Delta_{tot} = trace(\Sigma_r)$ is small enough so that the posterior in residual space, $p^{**,i}$ (defined below), is unimodal.

$$p^{**,i}(v_{n,r}) \triangleq p(v_{n,r}|v_{n,s}^i, \rho_n^i, X_{n-1}^i, Y_n) \propto p(Y_n|\tilde{C}_n^i + B_r v_{n,r}\overrightarrow{\mathbf{N}})\ \mathcal{N}(v_{n,r}; 0, \Sigma_r) \quad (11)$$

When Assumption 2 holds, one can use the following importance sampling strategy [17]: sample $v_{n,s}^i$, ρ_n^i from their state transition priors; compute $\tilde{C}_n^i$ using (2); and sample $v_{n,r}^i$ from a Gaussian approximation [26], $\mathcal{N}(m_n^i, \Sigma_{IS}^i)$, to $p^{**,i}$ about its mode, m_n^i. Finally, compute C_n^i using (1). Now, by conditional variance identity [27], $\mathbb{E}_{Y_n}[\Sigma_{IS}^i] \approx \mathbb{E}_{Y_n}[Variance(p^{**,i})] \leq \Sigma_r$. Thus if the following assumption holds, we can replace importance sampling from $\mathcal{N}(m_n^i, \Sigma_{IS}^i)$ by [17] deterministically setting $v_{n,r}^i = m_n^i$. We call this the Mode Tracking (MT) approximation.

Assumption 3: The total residual deformation variance, $\Delta_{tot} = trace(\Sigma_r)$ is small enough, i.e. $\Delta_{tot} < \Delta_{tot,bnd}$, so that with high probability, there is little error in replacing a random sample from $\mathcal{N}(m_n^i, \Sigma_{IS}^i)$ by m_n^i.

Based on the above ideas, we develop the PF-MT algorithm for contour tracking.

1) Importance Sampling on Effective Basis Dimensions: 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 (2), $\forall i = 1, 2, ..N$. We implement (2) using the level set method [16], [15].

The contour, C_n, is represented as the zero level set of a "level set function", denoted ϕ_n, i.e. C_n is the collection of all points $\{x \in \mathbb{R}^2 : \phi_n(x) = 0\}$ [15], [16] (x denotes the x-y coordinates). The direction of the gradient of ϕ_n, $\nabla\phi_n(x)$, is along the normal, $\overrightarrow{\mathbf{N}}_n$. The level set evolution corresponding to contour evolution given by (2), is: $\tilde{\phi}_n^i(x) = \phi_{n-1}^i(x) + v_{extend}(x)||\nabla_x \phi_{n-1}^i(x)||$ where v_{extend} is the normal extension [15], [16] of $B_s(p)v_{n,s}^i + \overrightarrow{\mathbf{N}}_{n-1}^T(p)\rho_n^i$ onto non-zero level sets. This implementation assumes that Assumption 1 holds. *If it does not hold (either τ large or motion fast), then (2) will have to be implemented using multiple iterations within one observation interval. Also, if the narrowband level set method [15], [16] is used, multiple iterations may be required to implement (2), depending on the velocity magnitude and the narrowband width.*

B_s and its extension onto all level sets need to be computed at each step. This can be done without computing the zero level set (contour) for the geometric basis[2].For the parametric basis, at each iteration, (i) the contour needs to be computed; it needs to always be traversed in the same order (say clockwise); and starting point correspondence needs to be maintained; (ii) the basis points need to be moved along with the contour using (10) and (iii) the B-spline interpolation functions need to be recomputed using the current arclength distance between the basis points.

2) Mode Tracking on Residual Space: This involves computing the mode, m_n^i, of $p^{**,i}$; setting $v_{n,r}^i = m_n^i$; and computing C_n^i using (1), $\forall i = 1, 2..N$. Computing m_n^i

[2]if the requirement of normal extension velocities [15], [16] is relaxed

Algorithm 1 PF-MT-TV: PF-MT for a Time Varying Effective Basis. Going from π^N_{n-1} to $\pi^N_n(X_n)$, $X^i_n = [C^i_n, \rho^i_n, v^i_{n,s}, v^i_{n,r}]$

1) *Importance Sample (IS) on effective basis:* $\forall i$,
sample $v^i_{n,s} \sim \mathcal{N}(A_s v^i_{n-1,s}, \Sigma_s)$, sample $\rho^i_n \sim \mathcal{N}(A_\rho \rho^i_{n-1}, \Sigma_\rho)$ and compute $\tilde{C}^i_n$ using (2).
2) *Mode Tracking (MT) in residual space:* $\forall i$,
(exact) compute m^i_n (mode of $p^{**,i}$ defined in (11)), set $v^i_{n,r} = m^i_n$ and compute C^i_n using (1). Or
(approximate) compute C^i_n by starting with $\tilde{C}^i_n$ & running k iterations of Gradient Descent to minimize $E(C_n) \triangleq -\log[p(Y_n|C_n)]$.
3) *Weight & Resample:* Compute w^i_n using (14) and resample [2].
4) *Detect Effective Basis Change:* as explained in Section III-B. If needed, go to step 5, else $n \leftarrow n+1$, go to step 1.
5) *Change Effective Basis:*
   a) Compute $K_{new} = L/\alpha_s$ where L is the length of the most likely contour particle and α_s is the desired distance between basis points.
   b) $\forall i$, reallocate the knots uniformly and evaluate the new basis $B_{K_{new}}(C^{(i)}_n) \triangleq B_{K_{new},i}$.
   c) $\forall i$, project $v^i_{n,s}$ into the new basis: $v^i_{n,s} \leftarrow (B^T_{K_{new},i} B_{K_{new},i})^{-1} B^T_{K_{new},i} B_i v^i_{n,s}$ and set $B_i \leftarrow B_{K_{new},i}$.
   d) $n \longleftarrow n+1$ and go to step 1.

and C^i_n requires being able to compute B_r. But B_r is the solution of $B_r B_r^T = I - B_s(B_s^T B_s)^{-1} B_s^T$. Since B_s depends on C^i_{n-1}, it will need to be computed at each n and for all i, which is very expensive. By using some approximations, we avoid having to compute B_r. Define $E(C_n) \triangleq -\log[p(Y_n|C_n)]$. We have shown [17] that if Assumption 2 holds, m^i_n can be computed by starting with $v_{n,r} = 0$ as initial guess and running k iterations (for some k) of gradient descent to minimize $E(\tilde{C}^i_n + B_r v_{n,r})$ w.r.t. $v_{n,r}$. If we also allow change along B_s, the k gradient descent iterations to minimize E as a function of $v_{n,r}$ can be replaced by k gradient descent iterations to minimize E as a function of C_n (skips the need to compute B_r). This assumes that:

Assumption 4: We replace (1) and (4) by:

$$C_n = \tilde{C}_n + v_{n,r} \overrightarrow{\mathbf{N}}(\tilde{C}_n), \; v_{n,r} \sim \mathcal{N}(0, \Delta I) \tag{12}$$

Gradient descent is implemented using the standard level set method[15], [16]. We start with $\phi^{(1)} = \tilde{\phi}^i_n$ (level set function corresponding to $\tilde{C}^i_n$) as initial guess and run k iterations of gradient descent to minimize E, i.e. run k iterations of the type $\phi^{(r+1)} = \phi^{(r)} + v_{extend} ||\nabla_x \phi^{(r)}||$ where v_{extend} is normal extension [15], [16] of $(\nabla_C E)$ onto non-zero level sets. After k iterations, we get is ϕ^i_n. Its zero level set is C^i_n.

3) Weighting and Resampling: This involves computing the weights, w^i_n, $\forall i = 1, 2..N$, and resampling [2]. w^i_n is computed as:

$$w^i_n = \frac{\tilde{w}^i_n}{\sum_{j=1}^N \tilde{w}^j_n}, \; \tilde{w}^i_n \triangleq w^i_{n-1} \frac{p(Y_n|C^i_n)\, \mathcal{N}(v^i_{n,r}; 0, \Sigma_r)}{\mathcal{N}(m^i_n; m^i_n, \Sigma^i_{IS})}. \tag{13}$$

Using Assumption 4, the above only requires knowing $||v^i_{n,r}||^2$. Since in the mode tracking step we minimize directly over C^i_n, we never compute $v^i_{n,r}$. We can replace $||v^i_{n,r}||$ by any easily computable distance, d, between C^i_n and $\tilde{C}^i_n$ (or between ϕ^i_n and $\tilde{\phi}^i_n$) without much error in practice. In our experiments, we use the set symmetric distance.Also, since $\mathbb{E}_{Y_n}[\Sigma^i_{IS}] \leq \Sigma_r$, when Σ_r is small, one can replace Σ^i_{IS} by Σ_r. This makes the denominator a constant (can be removed). Thus, the weights can be computed as:

$$w^i_n = \frac{\tilde{w}^i_n}{\sum_{j=1}^N \tilde{w}^j_n}, \; \tilde{w}^i_n = w^i_{n-1} p(Y_n|C^i_n)\, e^{-\frac{d^2(C^i_n, \tilde{C}^i_n)}{\Delta}} \tag{14}$$

B. Time-Varying Effective Basis: PF-MT-TV

The effective basis dimension needs to be large enough so that the mode tracking approximation in residual space can be justified at each n, i.e. we need to satisfy Assumptions 2 and 3 at each n. Assume that we know the maximum allowable value of the distance between consecutive basis points, α_s to ensure that $\Delta_{tot} < \Delta_{tot,bnd}$ and it remains constant with time. Computing α_s is discussed in [20]. For a given α_s, $K = L/\alpha_s$ and thus K needs to change when contour length changes significantly. As the contour deforms, both its total length and arclength distance between consecutive basis points changes. For the parametric basis, there is a need to change effective basis if this distance becomes significantly smaller (starting to estimate noise)[3] or significantly larger (residual deformation too large) than α_s. This is done as follows. Choose a $\alpha_{s,min} < \alpha_s$ and $\alpha_{s,max} > \alpha_s$. We declare a need to change dimension, 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 $K = \lceil L/\alpha_s \rceil$ for the most likely contour particle. Uniformly allocate the K basis points on the arclength of all contour particles and compute the new effective basis functions. *Entire algorithm is summarized in Algorithm 1.*

For geometric basis, deformation is a function of radial angle and so total angular "length" L=2π remains fixed.The only time K changes is when α_s=1/(2f_{min})

[3]Assumptions 2 and 3 require $\Delta_{tot} < \Delta_{tot,bnd}$ which only translates to an upper bound on the distance between consecutive basis points, α_s. But, in practice, if the distance between basis points becomes too small, the PF starts estimating noise (demonstrated in Fig. 4) because the velocities at the different basis points are assumed to be uncorrelated. Thus, distance becoming too small also needs to be detected and corrected.

changes.Change in contour length results in change in frequency response of deformation as a function of radial angle.

IV. Simulation and Experimental Results

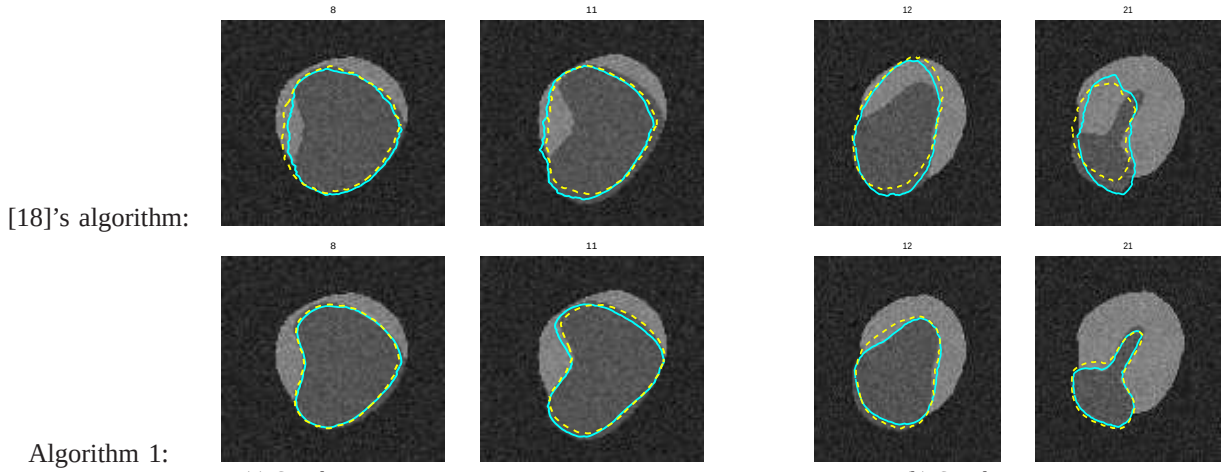
Since the posterior can be multimodal, plotting the "average" contour is not useful. In all figures, we plot two contours with the largest posterior (largest weights). The largest weight contour is shown as a solid cyan line, the second one as a dotted yellow line. In Figs. 2 and 3, we demonstrate some examples of situations where [18] will not work. We simulated the image sequence of Fig. 2(a) as follows. The contour of the white (light grey) object in the background was simulated by starting with a circle at $n = 0$ and using the system model described in (2)-(3) with $K = 6$ knots defining a geometric basis. We used $\Sigma_s = I$, $A_s = 0.5I$ and $\Sigma_\rho = 0.25I$. The residual deformation was assumed to be zero, i.e. we set $C_n = \tilde{C}_n$. The intensity of each pixel inside the contour was taken to be i.i.d Gaussian distributed with mean $u2 = 130$ and variance $\sigma^2_{obs} = 100$. The contour of the grey deforming object (object of interest) was also simulated with a model similar to the one above with the difference that a non-zero drift term, $\mu = [0\ 0\ 0\ 0\ 0\ 2]^T$ was added to the right hand side (RHS) of (3). This introduced a non-zero bias in the velocity dynamics of the 6^{th} knot, resulting in an inward motion with non-zero average velocity at any n. The mean intensity of this object was $u0 = 85$ and variance was again $\sigma^2_{obs} = 100$. The outer (black) background had mean intensity, $v0 = 45$ and same variance.

We tracked the grey object using the particle filter described in Algorithm 1 with $N = 45$ particles and K fixed. During tracking, we used all simulation parameters with the exception that we set $\mu = 0$. Instead, to track the non-zero bias in the velocity, we increased the system noise variance of the 6^{th} knot to 5, i.e. we used $\Sigma_s = diag([1\ 1\ 1\ 1\ 1\ 5])$. Residual deformation was tracked using the Mode Tracking step of Algorithm 1 with $k = 1$ GD iteration. Observation likelihood was defined as explained in Section II-C and it had a strong mode at the grey object and a weak mode at the white (light grey) object (due to the edge-based component). Some frames of the tracked sequence are shown in the second row of Fig. 2(a). In the first row, we show tracking of the same sequence using Affine PF-MT (algorithm of [18]). This algorithm used the space of affine deformations as the effective basis. All non-affine deformation was treated as "residual deformation" (tracked using a method similar to Mode Tracking of Algorithm 1). Since there are two distinct OL modes with roughly the same affine deformation (w.r.t. a circle) and since non-affine deformation per frame is large, the posterior of non-affine deformation is multimodal. So the contours often get stuck to the wrong mode in the Mode Tracking step. $k = 4$ GD iterations were used for tracking the residual (non-affine) deformation. As demonstrated in [20], increasing GD iterations does not improve tracking. We would like to clarify that we did not learn the affine deformation parameters (and hence it is not a fair comparison), but we did change the values a number of times until best possible results were obtained. The sequence of Fig. 2(b) was simulated in a manner similar to Fig. 2(a), with the difference that now knot locations 3 and 5 were made to move inside, i.e. $\mu = [0\ 0\ 1\ 0\ 1\ 0]^T$ is added to the RHS of (3). Here again Affine PF-MT[18] loses track.

The sequence of Fig. 3 was generated using a parametric basis with $K = 6$ basis points. Starting with a circle, one knot was made to move inside by adding a non-zero drift term, $\mu = [0\ 0\ 0\ 0\ 0\ 2]^T$, to the RHS of (3). Outlier observations similar to the one shown in the second column were simulated at every even frame starting at $t = 10$. This was done by increasing the observation noise and by setting $u2 = u0 = 85$. Before $n = 10$, the grey object is well approximated by affine deformation of a circle and hence is in track using both algorithms. But after $t = 10$, [18] gets stuck in the wrong mode due to the outlier observation. Since it does not generate samples for local deformation, it is unable to get to back to the correct mode fast enough (outliers appear every other frame). For this example, increasing GD iterations will only worsen the loss of track. On the other hand, Algorithm 1 is able to get back to the correct mode quickly, because it samples the space of local deformations.

In Fig. 4, we demonstrate the need to change the effective basis dimension, K. In the image sequence shown, the contour length keeps reducing because of inward motion of knots 3 and 5 (simulated by adding a non-zero drift $\mu = [0\ 0\ 1\ 0\ 1\ 0]^T$) to (3). We used a parametric basis here and $\alpha_s = 35$. While generating the sequence, K reduces from 6 to 4 at $n = 15$ and to 3 at $n = 23$. While tracking using Algorithm 1, 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 results are shown in the first three columns. In the last column, we show what happens if we do not allow change in K (use $K = 6$). When the knots come too close, independent velocity samples at these points often erroneously result in a contour with self-intersections (which breaks). All such contours get assigned zero weights. The contour particles that remain with non-zero weights are those which started expanding erroneously.

Fig. 5 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 tracking of the full car is not as accurate because we do not enforce closeness to a rigid car template as is done in [18] and [14]. A geometric basis was used. Fig. 6 shows sequential segmentation of a set of MRI slices of different cross-sections of the brain. We show results on segmenting brain tumor (grey-white region) in Fig. 6(a). A geometric basis was used. The low contrast in the images results in a large number of weak observation likelihood modes, very near the true one. There is intensity variation across the sequence and hence the edge likelihood helps remain in track. Preliminary results on sequentially segmenting the right ventricle (inside black region) are shown in Fig. 6(b).



(a) Simulation sequence 1 (b) Simulation sequence 2

Fig. 2. Multiple local deformation modes & large non-affine deformation per frame (simulation examples). First row: using Affine PF-MT[18] with $k = 4$ GD iterations. Second row: using Algorithm 1 with $k = 1$ GD iteration. *The solid contour is the particle with largest posterior weight, the dotted one has the second most largest weight.*

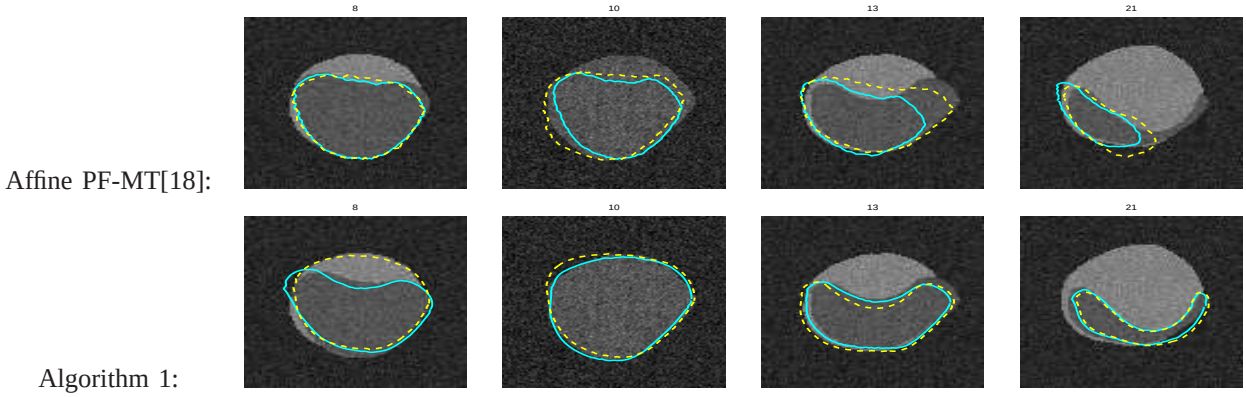


Fig. 3. Tracking through outlier observations (simulation example). At and after n=10, every even frame was an outlier observation similar to frame 10 shown in the second column above. First row: using Affine PF-MT[18] with $k = 4$ GD iterations. Second row: using Algorithm 1. First three plots use $k = 0$ GD iterations. Last plot uses $k = 1$ GD iteration.

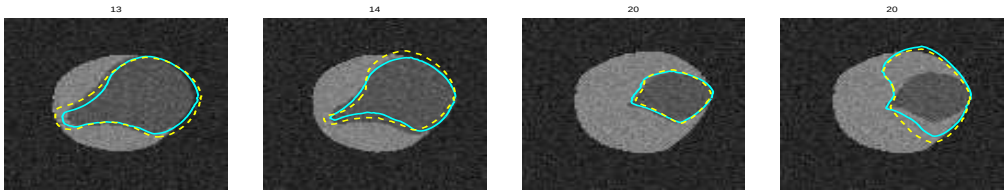


Fig. 4. The need to change K. The grey object deforms and keeps reducing in size which requires reducing K. The tracking is not great because only N=15 particles were used. In the last column, we show what happens if we keep tracking with $K = 6$ all the time. Some contours develop self intersections resulting in zero weight assigned to them (not shown). The ones with non-zero weight are those which did not self-intersect because they started expanding instead (shown).

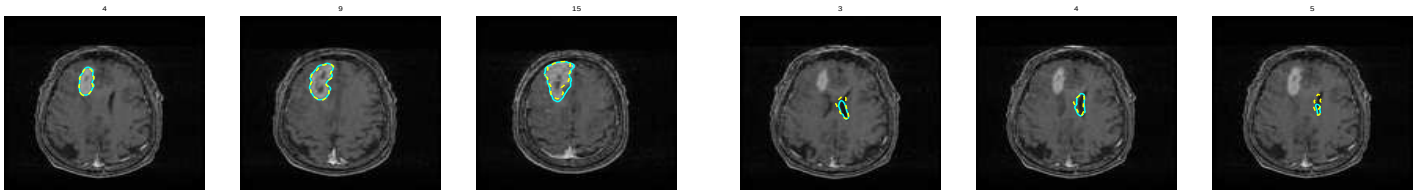
V. Conclusion and Open Issues

A new algorithm for tracking deforming contours is proposed, which uses the fact that in most problems, at any given time, most of the contour deformation occurs in a small number of dimensions (effective basis) while the deformation in the rest of the dimensions (residual space) in small. The dimension of the effective basis may change over time. Note that the proposed algorithm can also be used with other types of effective basis, e.g. PCA basis [28], [6] and also with other representations of the contour (other than level sets).

There are many open issues. The appropriate choice of effective basis is not clear. A second issue is the choice of effective basis dimension, K, and how to change K for both

(a) Tracking the contour for the car to left of the pole (b) Tracking the full car

Fig. 5. Tracking a car through partial occlusion by a pole.



(a) Tracking the tumor (grey-white region) (b) Attempt to track the right ventricle (black region in the center)

Fig. 6. Tracking the tumor (grey-white region) and the ventricle (black region in the center) in a brain MRI sequence. Sequence provided by Dr. Viren Amin of Iowa State University.

types of bases. When changing K while tracking, one also needs to deal with errors in estimating K, for e.g. using the ideas introduced in [19]. A very important implementation issue is the choice of observation models and the use of efficient resampling techniques [3], for large dimensional problems. Application 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, is currently being explored. For most medical image sequences, large amounts of hand-segmented training data can be obtained and hence learning the system dynamics can greatly improve the results. This is discussed in [20].

REFERENCES

[1] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, "Novel approach to nonlinear/nongaussian bayesian state estimation," *IEE Proceedings-F (Radar and Signal Processing)*, pp. 140(2):107–113, 1993.

[2] 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 Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

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

[4] G. Sapiro, *Geometric Partial Differential Equations and Image Processing*, Cambridge University Press, January 2001.

[5] B. Georgescu, X. S. Zhou, D. Comaniciu, and B. Rao, "Real-time multi-model tracking of myocardium in echocardiography using robust information fusion," in *Intl. Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2004.

[6] 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 Technical Report 2642*, Feb 2005.

[7] R.W. Brockett and A. Blake, "Estimating the shape of a moving contour," in *IEEE Conf. Decision and Control (CDC)*, 1994.

[8] A.Blake, M.Isard, and D.Reynard, "Learning to track curves in motion," in *IEEE Conf. Decision and Control (CDC)*, 1994.

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

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

[11] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *Int. Journal of Computer Vision*, vol. 1, pp. 321–331, 1987.

[12] M. Isard and A. Blake, "Condensation: Conditional Density Propagation for Visual Tracking," *Intl. Journal of Comp. Vision*, pp. 5–28, 1998.

[13] J. Jackson, A. Yezzi, and S. Soatto, "Tracking deformable moving objects under severe occlusions," in *IEEE Conf. Decision and Control (CDC)*, 2004.

[14] M. Niethammer and A. Tannenbaum, "Dynamic level sets for visual tracking," in *IEEE Conf. Decision and Control (CDC)*, 2004.

[15] J. A. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, 2nd edition, 1999.

[16] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, Springer Verlag, 2003.

[17] N.Vaswani, A.Yezzi, Y.Rathi, and A.Tannenbaum, "Particle filters for infinite (or large) dimensional state spaces-part 1," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2006.

[18] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi, "Particle filtering for geometric active contours and application to tracking deforming objects," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[19] N.Vaswani, "Particle filters for infinite (or large) dimensional state spaces-part 2," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2006.

[20] N. Vaswani, Y. Rathi, A. Yezzi, and A. Tannenbaum, "Pf-mt (particle filter with mode tracker) for tracking contour deformations," *Submitted*, 2006.

[21] O.Juan, R.Keriven, and G.Postelnicu, "Stochastic mean curvature motion in computer vision: Stochastic active contours," in *VLSM*, 2004.

[22] T. Schn, F. Gustafsson, and P.J. Nordlund, "Marginalized particle filters for nonlinear state-space models," *IEEE Trans. Sig. Proc.*, 2005.

[23] A.Yezzi and A.Mennucci, "Conformal metrics and true "gradient flows" for curves.," in *IEEE Intl. Conf. on Computer Vision (ICCV)*, 2005.

[24] David F. Rogers and J. Alan Adams, *Mathematical Elements for Computer Graphics*, WCB/McGraw-Hill, 1990.

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

[26] A. Doucet, "On sequential monte carlo sampling methods for bayesian filtering," in *Technical Report CUED/F-INFENG/TR. 310, Cambridge University Department of Engineering*, 1998.

[27] G. Casella and R. Berger, *Statistical Inference*, Duxbury Thomson Learning, second edition, 2002.

[28] M. Leventon, E. Grimson, and Olivier Faugeras, "Statistical shape influence in geodesic active contours," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2000, pp. 1316–1324.