Learning the Dynamics and Time-Recursive Boundary Detection of Deformable Objects

Walter Sun, *Member, IEEE*, Müjdat Çetin, *Member, IEEE*, Raymond Chan, and Alan S. Willsky, *Fellow, IEEE*

Abstract—We propose a principled framework for recursively segmenting deformable objects across a sequence of frames. We demonstrate the usefulness of this method on left ventricular segmentation across a cardiac cycle. The approach involves a technique for learning the system dynamics together with methods of particle-based smoothing as well as nonparametric belief propagation on a loopy graphical model capturing the temporal periodicity of the heart. The dynamic system state is a low-dimensional representation of the boundary, and the boundary estimation involves incorporating curve evolution into recursive state estimation. By formulating the problem as one of state estimation, the segmentation at each particular time is based not only on the data observed at that instant, but also on predictions based on past and future boundary estimates. Although this paper focuses on left ventricle segmentation, the method generalizes to temporally segmenting any deformable object.

Index Terms—Cardiac imaging, curve evolution, graphical models, image segmentation, learning, left ventricle (LV), level sets, magnetic resonance imaging, particle filtering, recursive estimation, smoothing.

I. INTRODUCTION

TEMPORAL segmentation of deformable objects is a problem encountered in many fields, including medical imaging, video coding, surveillance, and oceanography. In video sequences, the use of semantic object tracking is useful for compression [1], [2]. In the area of surveillance, there is often an interest in tracking known objects [3], [4], e.g., humans, whose motion involves nonrigid transformations. In oceanography, the desire to segment and track oceanic fronts is of interest to navigators, scientists, and oil rig drill teams [5]–[7]. While the methodology we propose can be applied to any of these temporal segmentation problems, we focus on a medical application, in particular the segmentation of the left ventricle (LV) of the heart across a cardiac cycle. This is a problem of interest because the left ventricle's proper function, pumping oxygenated blood to the entire body, is vital for normal human activity.

Having segmentations of the LV over time allows cardiologists to assess the dynamic behavior of the human heart. One quantitative measure of the health of the LV is ejection fraction (EF), the percentage volume of blood transmitted out of the LV in a given cardiac cycle. To compute EF, we need to have segmentations of the LV at multiple points in a cardiac cycle; in particular at end diastole (ED) and end systole (ES). In addition, observing how the LV evolves throughout an entire cardiac cycle allows physicians to determine the health of the myocardial muscles. Segmented LV boundaries can also be useful for further quantitative analysis. For example, past work [8], [9] on extracting the flow fields of the myocardial wall assumes the availability of LV segmentations throughout the cardiac cycle.

The automated segmentation of the LV endocardium in bright blood cardiac magnetic resonance (MR) images is a nontrivial process. First, the image intensities of the cardiac chambers vary due to differences in blood velocity [10]; blood that flows into the ventricles produces higher intensities in the acquired image than blood which remains in the ventricles [11]. Furthermore, locating the LV is complicated by the presence of the right ventricle and aorta jointly with the LV in many images of the heart. In addition, automatic segmentation of low signal-to-noise ratio (SNR) cardiac images (e.g., body coil MR or ultrasound [12]–[14]) is difficult because intensity variations can often obscure the LV boundary.

Using segmentations from a previous or future frame can aid in the segmentation of the object in the current frame. During a single cardiac cycle, which lasts approximately 1 second, the heart contracts from end diastole to end systole and expands back to end diastole. Over this time, MR systems can acquire approximately 20 images of the heart. Because adjacent frames are imaged over a short time period (approximately 50 ms), the LV boundaries exhibit strong temporal correlation. Consequently, previous and subsequent LV boundaries provide information regarding the location of the current LV boundary. Using such information is particularly useful for low SNR images, where the observation from a single frame alone may not provide enough information for a good segmentation. Fig. 1 shows the results of static segmentations compared with results obtained using the approach described in this paper. Our method exploits the dynamics of the heart and incorporates information from past and future frames.

The incorporation of dynamic models into the tracking of dynamically evolving shapes is an area of recent and growing interest both for the problem of cardiac tracking and for more
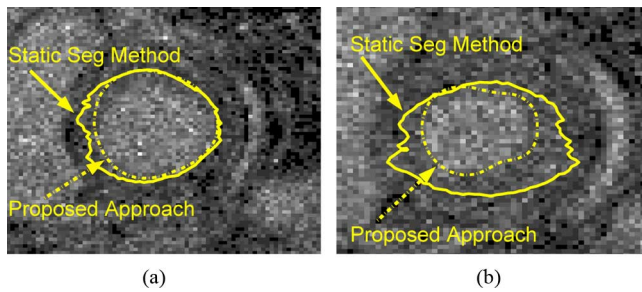
Fig. 1. Sample segmentations of the LV, one near end diastole (a) and the other near end systole (b), using a static segmentation method and our proposed recursive framework.

general application. In particular, in the work of Chalana *et al.* [14] and Jolly *et al.* [15] causal processing is performed in which the segmentation from the most recent frame is used to initiate a local search for the segmentation in the current frame—thus, implicitly using a simple, linear random walk model for curve evolution in which the best prediction of the current curve is the previous one. Zhou *et al.* [16] allow more complex linear dynamics on a landmark-based representation for the curves (thus introducing the issue of correspondence) but constrain the dynamics to be known and linear with Gaussian uncertainty and perform best linear unbiased estimation at each time fusing together the causal prediction from the preceding frame and the noisy measurements from the current frame. Senegas *et al.* [17] use sample-based methods (using sequential Monte Carlo) for causal shape tracking on a finite-dimensional representation of the shape space using spherical harmonics. Their prediction step implicitly defines an assumed dynamic model for cardiac motion that can be thought of as linear and Gaussian, with "pinned" distributions at both the start and end of the cycle. In particular, the mean shapes at ED and ES are determined from training data, and, given the shape estimate in a particular frame, the prediction is assumed to be Gaussian with mean shape given by a linear combination of the current estimate and a fraction of the difference between the mean shapes at ED and ES (hence, driving the means toward these pinned values at the end points). Vaswani *et al.* [18]–[20] also use a particle-based method for causal shape tracking, but they do this directly in a variational framework that leads to level-set based representations and particles that represent samples of the intrinsically infinite-dimensional curve. Finally, Cremers [21] uses level sets and finite-dimensional representations using principal component analysis to represent shapes and learns linear, Gaussian dynamic models for this finite-dimensional representation.

As in Cremers' approach, our method uses level set representations in concert with finite-dimensional projections of curves onto principal components. However, in contrast to essentially all of the previous methods, we do not constrain our dynamics to be linear or Gaussian and, in fact, use nonparametric, information-theoretic methods to learn statistical models for evolution of these finite-dimensional projections. As in Senegas' and Vaswani's work, we use sample-based methods for the fusing of our dynamic model with the dynamic image sequence. However, we perform noncausal smoothing in order to take advantage of frames throughout the cardiac cycle in order to enhance

segmentation at each frame. Moreover, if we exploit the periodicity of cardiac motion (as is done implicitly through the "pinning" operation in Senegas' work) we are led to the problem of performing smoothing on a graph consisting of a single cycle, requiring the use of new particle-based methods appropriate for inference on loopy graphs. In addition, while the core of our approach involves sampling and inference on finite-dimensional projections of the shape space (as opposed to the purely infinite-dimensional framework of Vaswani), by coupling level set representations into our graphical structure appropriately, our ultimate boundary estimates are in fact curves generated via curve evolution and, hence, are not constrained to a finite-dimensional space.

In summary, the major contributions of our approach are (a) the use of nonparametric, information-theoretic methods to learn the dynamics of shape evolution; (b) the development of particle-based smoothing algorithms (including ones based on a single-cycle graph to capture periodicity of cardiac motion) that allow segmentation throughout the LV cycle using all available data; (c) the incorporation of level set methods in a statistically consistent manner in order to generate boundaries not constrained to reside in the finite-dimensional domain in which dynamics have been learned; and (d) the demonstration of the effectiveness of our approach (both qualitatively and quantitatively) on cardiac imagery.

In Section II, we describe the probabilistic model structure and components on which our segmentation technique is based. In Section III, we introduce a finite-dimensional state representation for the LV boundaries, and present an information theoretic method for learning the system dynamics. Section IV contains descriptions of the sample-based approximate inference algorithms we use. This includes estimation algorithms for filtering and smoothing on a Markov chain, as well as on a single cycle graph. Furthermore, this section also includes the incorporation of curve evolution methods into our framework. In Section V, we present experimental results on cardiac MR images demonstrating the effectiveness of our approach. We conclude in Section VI.

II. PROBABILISTIC MODEL STRUCTURE

We formulate the dynamic segmentation problem as the estimation of the posterior distribution of the boundary[1] at each discrete time t based on the data available. In this section, we describe the structure and components of the probabilistic model we use in solving that estimation problem. Fig. 2 is a graphical model representation [22], [23] of the statistical structure we propose for describing data and cardiac dynamics over a single cardiac cycle. In this graphical model, circular and rectangular nodes correspond to hidden and observed random variables, respectively, and the graph encodes the statistical dependency structure of the model as we describe. Let us discretize a cardiac cycle into T time points, corresponding to the T temporal frames in MR imagery of a single cardiac cycle. At each time point t, we observe the image data y_t, which are noisy measurements of the blood and tissue intensity field. Let f_t be a coarse description of the intensities inside and outside the LV (in

[1]We use the terms "boundary" and "curve" interchangeably throughout this paper.
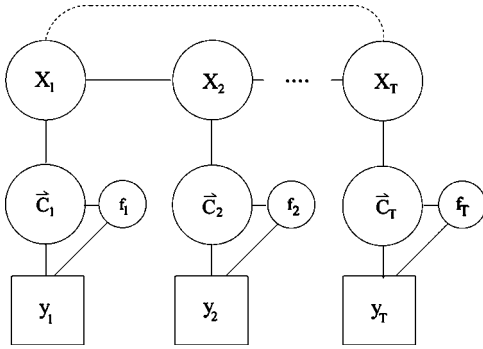
Fig. 2. Graphical model representing the statistical model structure used in our work.

particular, while other, more complex descriptions can be used, we take f_t to be the 2-D vector consisting of the mean intensity inside the LV and the mean intensity outside). Let $\vec{C}_t$ denote the LV boundary at time t. We note that $\vec{C}_t$ is an infinite-dimensional curve that evolves in time. We also define a variable X_t, which is a low-dimensional representation of $\vec{C}_t$, and which we use as the the dynamic system state. The structure in Fig. 2 implies a Markov structure for the dynamical model of X_t.

Let us assume that $t = 1$ corresponds to end systole (ES). As we move from left to right in Fig. 2, we go through various points in the cardiac cycle, e.g., through end diastole (ED) somewhere in the middle, back to ES towards the right end. Note the dashed line connecting X_1 and X_T. This line indicates the possibility of a graphical link to exploit the periodicity of the cardiac cycle, capturing the fact that although X_1 and X_T are not temporal neighbors in absolute time, they are neighbors in the cardiac cycle. We present techniques including and excluding this link.

One key aspect of the graphical structure in Fig. 2 is that the state variables X_t are low-dimensional representations of the LV boundaries $\vec{C}_t$. This is motivated by the fact that we are interested in learning the cardiac dynamics and that it is reasonable to expect that statistical significance of this learning will be enhanced if we use a compressed, finite-dimensional representation for the curves representing the LV wall. Based on this observation, our dynamical models, i.e., the graphical links across time, are on these finite-dimensional representations. Of course the actual observed data y_t involve the real curve $\vec{C}_t$ and the intensities f_t, which is captured in our model through the other graphical connections. Note that this can be interpreted as a finite-dimensional hidden Markov model capturing cardiac dynamics.

We note that because of what we know about the cardiac cycle, we expect the dynamics—i.e., the graphical links among the X_t's from time to time to be time-varying. In our work, we develop and use such time-varying models. We also note that the time indexing puts ED and ES at particular frames with some uncertainty in those frames. The models we develop also handle such uncertainties.

Next we describe particular components of our model, that is, probabilistic relationships between the random variables represented in the graphical model in Fig. 2 (we assume that the cardiac cycle is cyclic so that the state following X_T is X_1).

This involves a *likelihood model* $p(y_t|f_t, \vec{C}_t)$, an *intensity prior model* $p(f_t|\vec{C}_t)$, a *curve prior model* $p(\vec{C}_t|X_t)$, and a *dynamical model for the system state* $p(X_t|X_{t-1})$. Note that each of these models can be associated with a single edge or multiple edges in Fig. 2, and also that due to the Markov structure, these are the only relationships required to characterize the joint distribution of the variables involved. Note that our loopy graph represents a Markov random field. Namely, if we condition on any two nonadjacent nodes, the two separated chains are independent of each other.

A. Curve Prior Model ($p(\vec{C}_t|X_t)$)

As we mentioned, the state X_t is a finite-dimensional approximate representation of the boundary $\vec{C}_t$. A formal definition of X_t will follow in Section III-A. To allow for variability around the curve represented by X_t, we use the following model for curve $\vec{C}_t$:

$$p(\vec{C}_t|X_t) \propto exp(-D^2(\vec{C}_t, X_t)) \tag{1}$$

where $D^2(\vec{C}, X_t)$ measures the deviation of $\vec{C}_t$ from X_t by the following formula:[2]

$$D^2(\vec{C}_t, X_t) \equiv \int_{\vec{C}_t} d_{X_t}^2(s)ds \tag{2}$$

where $d_{X_t}(s)$ is the distance of point s on $\vec{C}_t$ from the curve represented by X_t [24]. Note that other distance measures could be used, as well. In addition, if one were interested in a curve length penalty or any other constraint on the boundary, such regularization can be incorporated within the curve prior model.

B. Intensity Prior Model ($p(f_t|\vec{C}_t)$)

Although we noted that differing blood velocities cause differences in intensities within the left ventricle, these differences are not systematic enough to be included in a spatial intensity model. It is then reasonable to assume intensities exhibit statistical homogeneity within regions. Models based on such assumptions are widely used in image segmentation, one well-known example being the Mumford–Shah model [25]. One extreme of that idea is to model intensities simply as having constant means inside and outside the region bounded by the curve, as in Chan and Vese's work [26]. Taking a similar perspective to that in [26], we model mean intensities as a constant bright intensity representing blood in $R_{in}(\vec{C}_t)$ (the region inside the LV boundary) and a constant darker one representing the myocardium in $R_{out}(\vec{C}_t)$ (the region immediately outside the LV boundary).

In [26], the constant intensity values in each region are determined in a completely data-driven fashion during the segmentation process. However, in the cardiac imaging application, we consider here, we can acquire statistical information about these intensities from training images. Therefore, we use a prior model for the intensity values inside and outside the LV. In particular, let $f_t = [f_t^{R_{in}(\vec{C}_t)} f_t^{R_{out}(\vec{C}_t)}]^T$ be a 2-D random vector, with independent components. Here, $f_t^{R_{in}(\vec{C}_t)}$ and $f_t^{R_{out}(\vec{C}_t)}$ in-

---

[2]This deviation is not a metric because it is not symmetric (in general, $D(X, Y) \neq D(Y, X)$) from its definition, but it provides a reasonable curve prior.

dicate the random variables for the constant intensities inside and immediately outside the LV, respectively. Note that we explicitly indicate the dependence of the regions on $\vec{C}_t$. Based on the distribution of intensities observed in the training data, we determined that these variables can be accurately modeled by log-normal distributions, leading to the following prior model:

$$p(f_t|\vec{C}_t) \propto exp\left(-\frac{(\log f_t^{R_{\text{in}}(\vec{C}_t)} - u)^2}{2\sigma_u^2}\right)$$
$$\cdot exp\left(-\frac{(\log f_t^{R_{\text{out}}(\vec{C}_t)} - v)^2}{2\sigma_v^2}\right). \quad (3)$$

The model parameters $u, v, \sigma_u^2, \sigma_v^2$ are learned from segmented training data (see Section V).

C. Likelihood Model $(p(y_t|f_t, \vec{C}_t))$

Given the simple intensity model described above, variations in the observed data, such as those due to differences in blood velocity [11], are accounted for by the following multiplicative noise model:

$$y_t(z) = \begin{cases} f_t^{R_{\text{in}}(\vec{C}_t)} \cdot n(z), & z \in R_{\text{in}}(\vec{C}_t) \\ f_t^{R_{\text{out}}(\vec{C}_t)} \cdot n(z), & z \in R_{\text{out}}(\vec{C}_t) \end{cases} \quad (4)$$

where $y_t(z)$ is the observed data at time t, spatial location z; and $n(z)$ is a spatially independent, identically distributed log-normal random field with $\log n(z)$ a Gaussian random field having zero mean and variance σ_n^2.

Given the two intensities $f_t^{R_{\text{in}}(\vec{C}_t)}$, $f_t^{R_{\text{out}}(\vec{C}_t)}$, and the location of a particular boundary $\vec{C}_t$, then $\log y_t(z)$ is normally distributed with mean $\log f_t^{R_{\text{in}}(\vec{C}_t)}$ if that particular pixel is inside the LV boundary (and $\log f_t^{R_{\text{out}}(\vec{C}_t)}$ if it is outside), and variance σ_n^2. Consequently, the likelihood of the entire observed image y_t at time t can be written as

$$p(y_t|f_t, \vec{C}_t) \propto \exp\left(-\int_{z \in R^{\text{in}}(\vec{C}_t)} \frac{(\log y_t(z) - \log f_t^{R_{\text{in}}(\vec{C}_t)})^2}{2\sigma_n^2}\right.$$
$$\left. -\int \frac{(\log y_t(z) - \log f_t^{R_{\text{out}}(\vec{C}_t)})^2}{2\sigma_n^2}\, dz\right). \quad (5)$$

D. State Dynamics $(p(X_t|X_{t-1}))$

As Fig. 2 indicates, the statistical structure of the state process consists of a Markov chain, corresponding to the solid edges in the figure, possibly augmented by an additional statistical constraint, indicated by the dashed edge, to capture (statistical) periodicity. As this latter piece is easily described-and can be viewed as conditioning the Markov chain on this additional piece of information, the major component needed to complete our statistical model is that of capturing the Markovian dynamics between two temporally-neighboring states X_{t-1} and X_t in the evolution of the LV, and can be associated with the the edges between the X's in Fig. 2. The perspective we take in this paper is to learn a time-varying dynamical model from off-line training data (using expert-segmented LV boundaries),
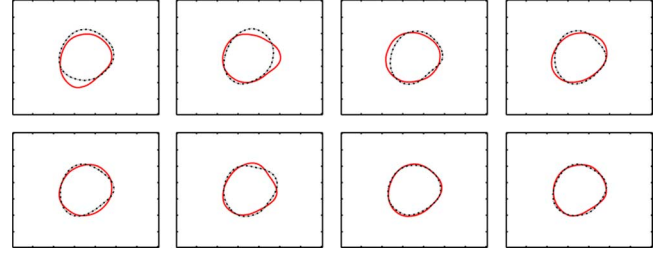

Fig. 3. Illustration of LV shape variability. $\bar{\psi} \pm \sigma_i \psi_i$ for the first eight primary modes of variability ($i = 1, 2, \ldots, 8$, left to right). Solid curve represents $\bar{\psi} + \sigma_i \psi_i$ while dashed represents $\bar{\psi} - \sigma_i \psi_i$.

and then use that model in the automatic segmentation process. Using a finite-dimensional approximate representation X_t of the boundaries $\vec{C}_t$ makes the learning problem better posed and the learned models more useful for prediction. In Section III, we describe this representation, and an information-theoretic approach for learning the forward dynamics $p(X_t|X_{t-1})$, as well as the backward dynamics $p(X_{t-1}|X_t)$.

III. System Dynamics

In this section, we describe our finite-dimensional representation of LV boundaries and propose a technique for learning the boundary dynamics.

A. Boundary Representation

The set of LV boundaries have different internal areas and different shapes across a cardiac cycle and across patients. We want to represent these boundaries in a simple, low-dimensional, yet accurate, manner. For a given boundary (a 1-D curve in 2-D space), we propose a representation using a signed distance level set function [27] (a 2-D surface in a 3-D space) where the level set function takes an absolute value equal to its distance from the curve. Given a training set of expert LV segmentations, we first find the area of each LV boundary and normalize the boundaries (across patients and over time) with respect to area. While there are a variety of approaches that are possible for specifying finite-dimensional projections of shapes, we use one that has been used with substantial success in other contexts [28], [29]. Specifically, we use principal component analysis (PCA) on the signed distance functions (SDFs) of the normalized boundaries to obtain a finite-dimensional basis for the shapes [28], [29]. Given this basis, we can represent the SDF of each shape by a linear combination of the basis elements. Thus, our finite-dimensional shape space consists of all curves with SDFs that are linear combinations of this finite set of basis vectors [and where the probabilistic relationship between actual curves and their finite-dimensional representations is given by (1)].

A large part of the shape variability is contained in the first few modes of variability. For our particular application, 97% of the variability can be captured by the first eight modes. Fig. 3 shows these eight primary modes of variability. Thus, we can approximate the SDF (ψ) of each shape by

$$\psi \approx \bar{\psi} + \sum_{i=1}^{8} \alpha^{(i)} \psi_i \quad (6)$$

where $\bar{\psi}$ is the mean level set and ψ_i is the ith mode of variability. It should be noted that there is no guarantee that the approximation will yield a single connected closed boundary. This is a limitation of using PCA on signed distance level set functions for shape representation. However, since the shapes are generally simple (curvature does not change drastically around a boundary), we have not observed any problems using SDFs. In addition to the shape, the cross-sectional area of the left ventricle (A_t) is important. Area needs to be added to the state because we previously normalized the curves before determining the shape variability. Thus, we then have a finite-dimensional approximation for each boundary consisting of the area and $\vec{\alpha} = [\alpha^{(1)}; \alpha^{(2)}; \ldots; \alpha^{(8)}]^T$

$$X_t = \begin{bmatrix} A_t \\ \vec{\alpha}_t \end{bmatrix}. \tag{7}$$

B. Learning the System Dynamics

In this work, we propose and use a dynamical system model based on statistical learning, rather than a physics-based model. This is motivated by the fact that purely physics-based models [30]–[33] are usually not readily available at the level of granularity one would need in an image analysis problem, and existing models may require high-dimensional states and/or a complex set of differential equations that model the interaction between the parts of the evolving system, which can pose computational difficulties and require specification/tuning/identification of many parameters.

We propose a method for learning the dynamics of deformable objects from training data. The closest work to ours appears to be that of Cremers [21]. However, unlike [21], which is based on linear models with Gaussian uncertainty, in our framework, we learn the dynamics using information-theoretic ideas and nonparametric statistics, without limiting them to being linear or Gaussian. One might wonder if this additional modeling capacity comes with an unreasonable cost. Our approach to learning is based on the technique in [34], which provides a nonparametric, yet computationally tractable approach to learning the dynamics. Our approach can directly estimate high-order Markov models by identifying the functionals of the past that have maximal mutual information with the future. However, for simplicity of development, we present only the first-order model of this type, with some specialization to take into account what we know about cardiac dynamics from ED to ES and back again.

Our approach involves two steps. First we learn maximally informative statistics about the past (or future) for the prediction of the current state. The second step involves nonparametrically estimating the (forward and backward) transition densities.[3] We describe these steps in the following subsections.

C. Maximally-Informative Statistics

We are interested in learning the statistical relationship between X_{t-1} and X_t and one might estimate the joint density

[3]If we are only performing causal processing, we only need the forward densities; however, for smoothing and for estimation when we include the periodicity constraint, we need both forward and backward transition densities.

only if an extensive amount of training data were available. Although we have achieved significant dimensionality reduction by representing the LV dynamics in terms of X's rather than $\vec{C}$'s, the degrees of freedom in the dynamical model can still be too many, especially if we have limited training data. Ideally, we would like to consider only the portion of the state X_{t-1} that is statistically pertinent to the prediction of X_t. It should be noted that we are not building a reduced-order model but simply positing that there is a reduced-dimensional part of the full-state X_t that can be used for accurate prediction of the next state for the forward model or the past state for the backward model. To this end, let us introduce a function $q_{t-1}(X_{t-1})$ which seeks to reduce dimensionality yet capture all of the information in X_{t-1} that relates to X_t.[4] This is achieved exactly only when $I(X_t; X_{t-1}) = I(X_t; q_{t-1}(X_{t-1}))$, where $I(X_t; X_{t-1})$ is the mutual information between X_t and X_{t-1}. Practically, however, information is lost when we introduce the function q_{t-1}. By the data processing inequality, $I(X_t; q_{t-1}(X_{t-1})) \leq I(X_t; X_{t-1})$, with equality only in the ideal case; when there is no loss of information. In practice, even if we cannot generally make $I(X_t; q_{t-1}(X_{t-1}))$ equal to $I(X_t; X_{t-1})$, we can try to make it as large as possible. As a result, we pose the problem as finding q_{t-1} that maximizes $I(X_t; q_{t-1}(X_{t-1}))$. This makes $q_{t-1}(X_{t-1})$ a maximally-informative statistic instead of a sufficient statistic. Our use of reduced-dimensional variables as maximally informative statistics for both forward and backward prediction can be interpreted as a form of learning with constraints, in this case imposing the constraint that the functional of the present needed for forward or backward prediction has reduced dimension.

Let us now briefly describe how we perform that optimization, namely the maximization of mutual information between X_t and $q_{t-1}(X_{t-1})$. The mutual information expression is given by

$$I(X_t; q_{t-1}(X_{t-1})) = H(X_t) + H(q_{t-1}(X_{t-1})) \\ - H(X_t, q_{t-1}(X_{t-1})) \tag{8}$$

where $H(x)$ denotes the entropy of the random variable x. The training data provide samples of X_t together with information about the cardiac phase at each time. We estimate the entropies in (8) based on such samples, using leave one out resubstitution. In particular, given N equally weighted samples $x_t^{(i)}$ of $p(X_t)$, we approximate this density using a kernel density estimate with Gaussian kernels. Let us define $k(X; x_t^{(i)}, \sigma^2)$ to be a Gaussian kernel with mean $x_t^{(i)}$ and variance σ^2, where σ^2 is determined by a method such as that in [35]. Then, the entropy estimate of $q_t(X_t)$ is

$$H(q_t(X_t)) \\ = -\frac{1}{N} \sum_{i=1}^{N} \log \left(\frac{1}{N-1} \sum_{j \neq i} k(q_t(x_t^{(i)}); q_t(x_t^{(j)}), \sigma^2) \right). \tag{9}$$

[4]Note that we can define a function in the "backward" direction in a completely analogous fashion.

Taking the derivative with respect to any parameter a of the function q_t yields

$$
\begin{aligned}
\frac{\partial H(q_t(X_t))}{\partial a} \\
= -\frac{1}{N} \sum_{i=1}^{N} \Bigg[& \frac{1}{\sum_{j \neq i} k\left(q_t(x_t^{(i)}); q_t(x_t^{(j)}), \sigma^2\right)} \\
& \cdot \sum_{j \neq i} -\frac{q_t(x_t^{(i)}) - q_t(x_t^{(j)})}{\sigma^2} \\
& \cdot k\left(q_t\left(x_t^{(i)}\right); q_t\left(x_t^{(j)}, \sigma^2\right)\right) \\
& \cdot \left(\frac{\partial q_t(x_t^{(i)})}{\partial a} - \frac{\partial q_t(x_t^{(j)})}{\partial a}\right) \Bigg].
\end{aligned} \tag{10}
$$

By applying (10) to the second term of (8) (and using a similar derivation to find the derivative of the joint entropy of the third term), we can determine the gradient of $I(X_t; q_{t-1}(X_{t-1}))$. At each iteration, we move in the direction of the gradient to maximize mutual information, continuing until convergence.

At this point, we make a simplification, and constrain $q_t(\forall t)$ to be a linear function. We can then write

$$
q_{t-1}(X_{t-1}) = Q_{t-1} X_{t-1} = \begin{bmatrix} Q_A & Q_{A\vec{\alpha}} \\ Q_{\vec{\alpha}A} & Q_{\vec{\alpha}} \end{bmatrix}_{t-1} X_{t-1}. \tag{11}
$$

The problem now reduces to finding the parameters of Q_t. It should be noted that the linearity assumption for q_t does not mean that the resulting dynamic model is either linear or Gaussian [34], just that we choose statistics for the conditional densities that are linear functions, but the resulting transition densities that are learned (see Section III-D) will involve non-linear functions of these statistics.

In the discussion thus far, we have discussed finding a time-varying q_t. In order to have an accurate estimate of the parameters of q_t, there must be sufficient training data. Practically, we may not have enough data to learn a different q_t for each t. For our particular training set, we learn the dynamics separately in the two distinct phases of the cardiac cycle: q_S for the systolic phase, when oxygenated blood leaves the LV, and q_D is for the diastolic phase, when the LV fills itself with blood.

D. Learning the Forward and Backward Densities

Given the maximally informative statistics q_S and q_D, we now discuss how we obtain the forward densities (backward densities can be learned in a completely analogous manner) based on training data. Note that given a forward model, there is a precisely defined model in reverse time which is completely consistent. It is, thus, worth noting that there are not two different models but rather two different representations of the same model. Such models play a variety of roles including their use in defining optimal smoothing algorithms [36]–[39], algorithms that use all data in a time interval to estimate the chain at every point in the interval. These algorithms can be thought of as message-passing algorithms in which messages

are passed both forward and backward, each using the corresponding model (forward and backward) in the message-passing stage.

The procedure we describe in this section produces densities of the form $p(X_t|q_{t-1}(X_{t-1}))$ which we use as approximate representations of the state dynamics $p(X_t|X_{t-1})$. Let us consider a time instant in the systolic phase. Given samples of X_t and X_{t-1}, we construct a kernel density estimate of the joint density $p_{X_t, q_S(X_{t-1})}(X_t, q_S(X_{t-1}))$, from which we can obtain the desired forward transition density by conditioning: $p_{X_t|q_S(X_{t-1})}(X_t|q_S(X_{t-1}))$. Note that although we do not learn a different maximally informative statistic for each time in the systolic and diastolic phases, our framework allows for the learning of a different transition density for each time, resulting in a time-varying dynamical model. However, in practice, this requires the availability of sufficient training data to support such learning. When the number of training samples is moderate, one could impose a time-invariant model for the systolic phase, and one for the diastolic phase (and, hence, use all data in that phase, rather than only at a particular time, for learning), as we do in our experiments in Section V.

What we have described up to this point produces different dynamical models for the systolic and diastolic phases. To use the right model during the segmentation process, one would need to know the cardiac phase at particular frames. However, as there is uncertainty and patient-to-patient variability in the precise location of ED and ES—and since our transition models switch at these times (e.g., at ES, the point of minimum area, cardiac motion changes from the systolic to the diastolic phase)—we need to incorporate this variability into our statistical model. We do this first by using training data to learn the probability that the heart is in each phase at particular times in the cycle and use this to define a "transition region" over which the transition density is taken as a mixture density based on q_S and q_D.

In particular, we assume that we are in the systolic (diastolic) phase between frames 1 and 6 (12 and 20). For the region in between, we apply a combination of the two dynamics according to the mixture probability determined empirically. For instance, 14.3% of the training examples reach end systole at frame 6 and transition to the diastolic phase from that frame onward. Based on that, when constructing $p(X_7|q_6(X_6))$, 14.3% of the samples of come from the forward density using q_D while 85.7% come from the forward density using q_S.

To demonstrate the usefulness of the mixture model, Fig. 4(a) shows predictions obtained for the eighth frame in a test sequence using a mixture dynamic model constructed for that time instant. Since we have access to the underlying LV sequence, we actually know that this particular test sequence is in the systolic phase between frames 7 and 8. Of course, this information is not available in a real data processing setting. Note the bimodal nature of the predictions due to sampling from the two densities, acknowledging the uncertainty about the phase, rather than making a strict (and potentially wrong) choice. As described in Section IV, such predictions are then combined with data. Sample boundaries after such a data update are shown in Fig. 4(b), together with the ground truth, which shows that
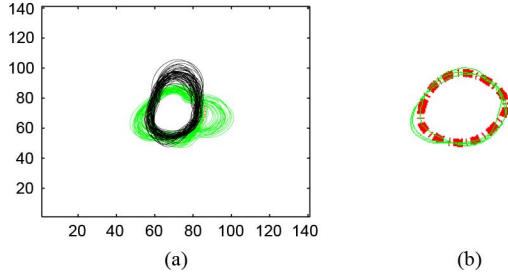
Fig. 4. (a) Predictions for frame 8 in a test sequence. Samples in black come from $p(X_8|q_D(X_7))$ while those in green come from $p(X_8|q_S(X_7))$. (b) Most likely, boundaries after incorporating the data at frame 8 superimposed upon the truth (red dash-dotted curve). A reasonably accurate set of curve estimates are obtained after the data update at frame 8 (See Section IV).

by providing predictions from both phases, we allow the data to properly weigh in for the right phase.

IV. INFERENCE ALGORITHMS

In this section, we describe our approach to dynamic segmentation based on the graphical model of Fig. 2, the component models described in Section II, and the dynamic model for X learned using the method discussed in Section III. Thanks both to the nonparametric form of the forward and backward transition densities for X and, more generally, the complex nature of the Bayesian inference problems we wish to solve, we are led to the use of sampling—i.e., particle-based algorithms [40], [41] to approximate the densities and likelihood functions required in the inference procedure. However, there are two issues that preclude the straightforward use of standard particle filtering methods in our problem. The first is that in addition to considering causal filtering, we are also interested not only in noncausal smoothing (i.e., using all data to estimate cardiac state at each point in the cycle) but also in performing this smoothing when we also take into account the periodicity of the cycle, which adds a loop to the graphical model in Fig. 2 (illustrated by the dashed edge). The second is that a full particle representation of the inference problem of LV segmentation would involve generating particles for the infinite-dimensional curves, $\vec{C}$. While an approach to doing this has recently been developed (for models in which curve dynamics are described by linear-quadratic variational formulations) [18]–[20], we have chosen to avoid this additional complexity through an approximation that involves solving level-set based curve evolutions during the sampling procedure.

A. Causal Filtering

In the filtering problem, the goal is to estimate X_t for any t based on $y_{1:t} = \{y_\tau|\tau \leq t\}$. Based on the graphical model in Fig. 2, the posterior density $p(X_t|y_{1:t})$ can be written as

$$
p(X_t|y_{1:t})
$$
$$
\propto \int_{f_t} \int_{\vec{C}_t} p(y_t|f_t, \vec{C}_t)p(f_t|\vec{C}_t)p(\vec{C}_t|X_t)
$$
$$
\cdot p(X_t|y_{1:t-1})d\vec{C}_t df_t \tag{12}
$$

where $p(X_t|y_{1:t-1})$ is the prediction density

$$
p(X_t|y_{1:t-1}) = \int_{X_{t-1}} p(X_t|X_{t-1})p(X_{t-1}|y_{1:t-1})dX_{t-1}. \tag{13}
$$

As in standard particle filtering algorithms (Appendix I), we assume the availability of a sample-based version of the density $p(X_{t-1}|y_{1:t-1})$ at the preceding iteration. Suppose that we represent $p(X_{t-1}|y_{1:t-1})$ by an equally-weighted set of N samples $x_{t-1}^{(i)}(i \in \{1, 2, \ldots, N\})$[5]

$$
p(X_{t-1}|y_{1:t-1}) \approx \frac{1}{N} \sum_{i=1}^{N} \delta\left(X_{t-1} - x_{t-1}^{(i)}\right). \tag{14}
$$

Using the condition density $p(X_t|q_{t-1}(X_{t-1}))$ as an approximation for $p(X_t|X_{t-1})$ and applying the N samples to (13), we sample from the learned forward transition density for each of these samples $(pX_t|q_{t-1}(X_{t-1} = x_{t-1}^{(i)}))$ to generate a sampled version of the one-step-ahead predicted density $p(X_t|y_{1:t-1})$. In particular, for each $x_{t-1}^{(i)}$, we generate M equally weighted samples, resulting in the following estimate:

$$
p(X_t|y_{1:t-1}) \approx \frac{1}{MN} \sum_{i=1}^{MN} \delta\left(X_t - x_{t|t-1}^{(i)}\right). \tag{15}
$$

Given this density and set of particles, the objective then is to provide a sample-based approximation to the density $p(X_t|y_{1:t})$ in (12). In standard particle filtering applications—in which the measurements depend directly on X, all this involves is a reweighting of the particles in the one-step-ahead predicted density, $p(X_t|y_{1:t-1})$, using the likelihood function for the observations conditioned on each particle. However, as can be seen in (12), the computation of these new weights in our case is complicated by the two intervening variables in our graphical model, namely f_t and $\vec{C}_t$, as (12) requires a marginalization over these variables. One way to deal with that complication would be to approximate the marginalization operation by choosing the $(f_t, \vec{C}_t)$ pair which maximizes the integrand of (12).

One principled manner in which to do this would be to use each each particle $x_{t|t-1}^{(i)}$ to generate its own pair of samples of f_t and $\vec{C}_t$ by solving the optimization problem

$$
(f_t^{*,i}, \vec{C}_t^{*,i}) = \arg\max_{f_t, \vec{C}_t} \log\left(p(y_t|f_t, \vec{C}_t)p(f_t|\vec{C}_t)\right.
$$
$$
\left. \cdot p(\vec{C}_t|X_t = x_{t|t-1}^{(i)})\right) \tag{16}
$$

and then to weight that particle by the exponential of the maximizing value of the right-hand side of (16) (and finally to renormalize these weights so that they sum to 1). This would, of course, require generating one particle pair $(f_t^{*,i}, \vec{C}_t^{*,i})$ for each particle in $p(X_t|y_{1:t-1})$, a computationally intensive task. Instead, we have found it effective to generate a *single particle*

[5]For simplicity, our representation here is based on delta functions. One could use any other kernel function instead.

pair by optimizing an average over all particles $x_{t|t-1}^{(i)}$ of the right-hand sides of (16), i.e., by solving

$$(f_t^*, \vec{C}_t^*) = \arg\max_{f_t, \vec{C}_t} \log \int_{X_t} p(y_t | f_t, \vec{C}_t)$$
$$\cdot p(f_t | \vec{C}_t) p(\vec{C}_t | X_t) p(X_t | y_{1:t-1}) dX_t. \quad (17)$$

Substituting expressions from (1), (3), (5), and (15) into (17), and using the fact that

$$\int_{X_t} p(\vec{C}_t | X_t) p(X_t | y_{1:t-1}) dX_t$$
$$\propto \frac{1}{MN} \sum_i^{MN} \exp\left(-D^2(\vec{C}_t, x_{t|t-1}^{(i)})\right) \quad (18)$$

we obtain

$$(f_t^*, \vec{C}_t^*) = \arg\min_{f_t, \vec{C}_t} E(f_t, \vec{C}_t) \quad (19)$$

where

$$E(f_t, \vec{C}_t) = -\log p(y_t | f_t, \vec{C}_t) - \log p(f_t | \vec{C}_t)$$
$$- \log \int_{X_t} p(\vec{C}_t | X_t) p(X_t | y_{1:t-1}) dX_t$$
$$= \left(\int_{z \in R_{\text{in}}(\vec{C}_t)} \frac{(\log y_t(z) - F_t^{\text{in}}(\vec{C}_t))^2}{2\sigma_n^2} dz \right.$$
$$+ \int_{z \in R_{\text{out}}(\vec{C}_t)} \frac{(\log y_t(z) - F_t^{\text{out}}(\vec{C}_t))^2}{2\sigma_n^2} dz \right)$$
$$+ \left(\frac{(F_t^{\text{in}}(\vec{C}_t) - u)^2}{2\sigma_u^2} + \frac{(F_t^{\text{out}}(\vec{C}_t) - v)^2}{2\sigma_v^2} \right)$$
$$+ \log \left[\frac{1}{MN} \sum_i^{MN} \exp(-D^2(\vec{C}_t, x_{t|t-1}^{(i)})) \right] \quad (20)$$

where $F_t^{\text{in}}(\vec{C}_t) = \log f_t^{R_{\text{in}}(\vec{C}_t)}$ and $F_t^{\text{out}}(\vec{C}_t) = \log f_t^{R_{\text{out}}(\vec{C}_t)}$. Practically, we initialize the minimization using the mean shape from the training data.

The solution to this variational problem is related to joint boundary and field estimation methods—e.g., those based on the so-called Mumford–Shah functional [26], [42], [43]—and can be solved in an analogous manner, using coordinate descent and curve evolution (see Appendix II).

At the end of this process, we have essentially replaced (12) by

$$p(X_t | y_{1:t})$$
$$\propto p(y_t | f_t^*, \vec{C}_t^*) p(f_t^* | \vec{C}_t^*) p(\vec{C}_t^* | X_t) p(X_t | y_{1:t-1}). \quad (21)$$

Now we can use (21) to re-weight our particle-based representation

$$p(X_t | y_{1:t}) = \sum_{i=1}^{MN} w_t^{(i)} \delta(X_t - x_{t|t-1}^{(i)}), \quad (22)$$

$$w_t^{(i)} = \frac{1}{Z} p(y_t | f_t^*, \vec{C}_t^*) p(f_t^* | \vec{C}_t^*) p$$
$$\cdot (\vec{C}_t^* | X_t = x_{t|t-1}^{(i)}) \quad (23)$$

with Z chosen so that the weights sum to 1. Finally, we can resample this density to N equally weighted samples (see Appendix I). Thus, from equally weighted samples representing the posterior at $t-1$, we have arrived at equally-weighted samples for the posterior at t.

B. Noncausal Smoothing

In the smoothing problem, the goal is to estimate X_t for any t based on y_τ for all $\tau \in [1, T]$ in the cardiac cycle (we denote this observation sequence by $y_{1:T}$). Based on the graphical model in Fig. 2, the posterior density of interest, $p(X_t | y_{1:T})$, can be written as

$$p(X_t | y_{1:T})$$
$$\propto \int_{f_t} \int_{\vec{C}_t} p(y_t | f_t, \vec{C}_t) p(f_t | \vec{C}_t) p(\vec{C}_t | X_t)$$
$$\cdot p(X_t | y_{(1:T) \backslash t}) d\vec{C}_t df_t \quad (24)$$

where $y_{(1:T) \backslash t}$ denotes all data in $y_{1:T}$ except y_t. Note that the likelihood, intensity prior, and curve prior terms are the same as in the filtering formulation of (12), but the prediction term is now conditioned on $y_{(1:T) \backslash t}$. Note that this formulation is valid for smoothing on a chain as well as inference on a single cycle graph (i.e., when the dashed edge in Fig. 2 is included). Let us first consider the chain case.

As in the filtering problem, consider the following approximate representation of (24)

$$p(X_t | y_{1:T})$$
$$\propto p(y_t | f_t^*, \vec{C}_t^*) p(f_t^* | \vec{C}_t^*) p(\vec{C}_t^* | X_t) p(X_t | y_{(1:T) \backslash t}). \quad (25)$$

We will describe how to compute f_t^* and $\vec{C}_t^*$ in this case, but for the time being, assume we have them. We are then left with the computation of (25) at each point in time, which is a classical smoothing problem or, equivalently, an inference problem on the graph in Fig. 2 (for the moment without the dashed edge). The solution can be found in many sources, and, with an eye toward including the dashed edge, we describe this in terms of message-passing also known as belief propagation [44]. In particular, the Markovian structure of X implies that (25) has the following form:

$$p(X_t | y_{1:T}) \propto p(y_t | f_t^*, \vec{C}_t^*) p(f_t^* | \vec{C}_t^*) p(\vec{C}_t^* | X_t) p(X_t)$$
$$\cdot \prod_{\tau \in N(t)} m_{\tau t}(X_t) \quad (26)$$

where the neighbors $N(t)$ for each time t are times just before and after, i.e., $t-1$ and $t+1$ (except for the end points 1 and T which have neighbors on only one side), and where the "messages" $m_{\tau t}(X_t)$ represent a summary of all of the information provided by points in time before (for $\tau = t-1$) or after ($\tau = t+1$)—i.e., they are likelihood functions capturing all of the information relevant for inference at time t contained in the dynamical relationships and data either before or after t.

These messages themselves have recursive structure inherited from the graph. Adopting graphical model notation and terminology [22], [23], these recursions involve the specification of the probabilistic model in terms of *potential functions* corresponding to each node/time in our graph and to each edge. In our case, the node potentials capture all of the statistical information localized to time t alone, i.e.,

$$\psi_t(X_t, y_t) = p(y_t|f_t^*, \vec{C}_t^*)p(f_t^*|\vec{C}_t^*)p(\vec{C}_t^*|X_t)p(X_t). \quad (27)$$

The edge potentials are given by

$$\psi_{t,\tau}(X_t, X_\tau) = \frac{p(X_t, X_\tau)}{p(X_t)p(X_\tau)} = \frac{p(X_t|X_\tau)}{p(X_t)} = \frac{p(X_\tau|X_t)}{p(X_\tau)} \quad (28)$$

where the last two forms in (28) suggest how these expressions are used. In particular, if $\tau = t - 1$, the next-to-last expression in (28) involves the forward prediction density, while the last expression involves the backward prediction density.

Using these quantities, the messages in (26) satisfy the following relationship:

$$
\begin{aligned}
m_{\tau t}&(X_t) \\
&\propto \int_{X_\tau} \psi_{t,\tau}(X_t, X_\tau)\psi_\tau(X_\tau, y_\tau) \\
&\quad \cdot \prod_{s \in N(\tau)\backslash t} m_{s\tau}(X_\tau)dX_\tau
\end{aligned}
\quad (29)
$$

where $N(\tau) \backslash t$ denotes all neighbors of node X_τ except X_t. Note in our case that for the graph of Fig. 2 (without the dashed line), each of these sets consists of a singleton (except for $t = 1$ and $t = T$ for which these sets are empty). In particular, for $\tau = t-1$, $N(\tau)\backslash t$ is the single point $t-2$ and (29) corresponds precisely to the causal filtering operation, in which we take the information from the preceding time $(t - 2)$, incorporate the available information at time $t - 1$ (the multiplication by the node potential at time $\tau = t - 1$), and then predict ahead one time step (the multiplication by the edge potential from $t - 1$ to t followed by the integration to marginalize out time $t - 1$. Analogously, if $\tau = t + 1$, (29) corresponds to a backward propagation of information.

These fixed-point equations can be solved iteratively via different methods for *message-scheduling*, i.e., for choosing the order in which messages are updated. For a chain, one approach is to perform smoothing by a two-sweep procedure, moving from the leftmost node of the chain to the rightmost one and then back again. Such an algorithm yields the correct answers at the end of these two sweeps. The same idea can be generalized to tree-structured graphs. At the other extreme one can update all messages simultaneously—i.e., we can simply apply the fixed point equations iteratively

$$
\begin{aligned}
m_{\tau t}^n&(X_t) \\
&\propto \int_{X_\tau} \psi_{t,\tau}(X_t, X_\tau)\psi_\tau(X_\tau, y_\tau) \\
&\quad \cdot \prod_{s \in N(\tau)\backslash t} m_{s\tau}^{n-1}(X_\tau)dX_\tau.
\end{aligned}
\quad (30)
$$

For inference on chains (and on tree-structured models more generally), convergence occurs once the messages from each node have propagated to every other node in the graph.

Such message computations can be performed exactly only in special cases—e.g., when everything is Gaussian—in which these functions have fixed finite-dimensional parameterizations. This leads to the need for sample-based methods analogous to those used in particle filtering. In our work, we use two such sampling-based algorithms. The first, the forward-backward algorithm of Doucet *et al.* [41], is a two-sweep procedure. The forward step of the algorithm uses particle filtering as explained in Section IV-A. In the backward pass, it then re-weights the particles that were obtained through the forward filtering process to obtain the posterior $p(X_t|y_{1:T})$ at each time t. The second algorithm we consider and use is nonparametric belief propagation (NBP) [45]. In NBP, particle representations serve as weighted mixture distribution approximations (in particular Gaussians sums) of the true messages. The NBP algorithm stochastically approximates the parallel message propagation operation in (30) and thus provides a consistent nonparametric estimate of the outgoing message. The message products in (26) and (29) are computed using an efficient local Gibbs sampling procedure. One of the most interesting ideas in NBP is a method for sampling from such products without explicitly constructing them. A detailed description of NBP can be found in [45]. Here, we only hightlight a number of aspects of our framework differentiating it from a standard application of NBP.

The first involves the dependence of the node potentials on f_t^* and $\vec{C}_t^*$. Let us describe how we compute these quantities at each iteration [with the resulting values then used to update the node potentials in (27) for use in NBP]. Let $p^{(n)}(X_t|y_{(1:T)\backslash t})$ denote the density of the product of the incoming messages to node X_t at iteration n, [i.e., the product of messages in (30)], represented nonparametrically as

$$p^{(n)}(X_t|y_{(1:T)\backslash t}) \approx \frac{1}{N}\sum_{i=1}^{N} w_{t|T\backslash t}^{(i)}[n]k(X_t; x_{t|T\backslash t}^{(i)}[n], \Lambda) \quad (31)$$

where N is the number of samples, $x_{t|T\backslash t}^{(i)}[n]$ denotes the i sample, and $k(X_t; x_{t|T\backslash t}^{(i)}[n], \Lambda)$ denotes a Gaussian kernel with mean $x_{t|T\backslash t}^{(i)}[n]$ and covariance Λ. This is analogous to (15) for the filtering problem. The objective then is to provide an approximation to the posterior $p(X_t|y_{1:T})$ using a sample-based version of (26). This requires the computation of f_t^* and $\vec{C}_t^*$, for which we use a similar procedure to that in the filtering context: at every iteration n we carry out the following optimization:

$$(f_t^*[n], \vec{C}_t^*[n]) = \arg\min_{f_t, \vec{C}_t} E_n(f_t, \vec{C}_t) \quad (32)$$

where

$$
\begin{aligned}
E_n(f_t, \vec{C}_t) = &-\log p(y_t|f_t, \vec{C}_t) - \log p(f_t|\vec{C}_t) \\
&-\log \int_{X_t} p(\vec{C}_t|X_t)p^{(n)}(X_t|y_{(1:T)\backslash t})dX_t \quad (33)
\end{aligned}
$$

which is essentially the same as (20) except for the last term. Given the form of $p^{(n)}(X_t|y_{(1:T)\backslash t})$ in (31), this term involves

a convolution of the Gaussian kernel with $p(\vec{C}_t|X_t)$. To simplify that operation, just for this step of our procedure, we approximate the Gaussian kernel with a Dirac delta function, through which the last term in (33) becomes

$$\log\left[\frac{1}{N}\sum_i w_{t|T\backslash t}^{(i)}[n]\exp(-D^2(\vec{C}_t, x_{t|T\backslash t}^{(i)}[n]))\right] \quad (34)$$

where D is as defined in (2). Once the values of $f_t^*[n]$ and $\vec{C}_t^*[n]$ are found to minimize this functional we update the node potential $\psi_t^{(n)}(X_t, y_t)$ (note that here we indicate the dependence of the potential on the iteration n explicitly) using (27). Then the NBP belief update procedure [45] can be used to generate samples from the posterior density $p(X_t|y_{1:T})$.

Another point worth mentioning is the way we represent the edge potentials in passing messages in different directions on an edge. For complete consistency, we should use precisely the same potential for messages being sent in either direction. This is not an issue in Doucet's forward-backward algorithm, as in that approach no new particles are generated in the backward sweep—they are simply reweighted. However, for NBP both forward and backward messages are used. For messages passed forward (i.e., $m_{(t-1)t}$) in the chain, the pairwise potential used in (29) is $\psi_{t-1,t}(X_{t-1}, X_t) = p(X_t|X_{t-1})/p(X_t)$ which requires the forward conditional density. Meanwhile, for messages passed backward in time (i.e., $m_{t(t-1)}$), we write the pairwise potential as $\psi_{t,t-1}(X_t, X_{t-1}) = p(X_{t-1}|X_t)/p(X_{t-1})$ which requires the backward conditional density. Since we learn the forward and backward densities separately (as described in Section III), the approximations to $\psi_{t-1,t}$ and $\psi_{t,t-1}$ used in our algorithm are not precisely equal. Rather than introducing a method to enforce equality (a difficult thing to do with particle-based densities) we do not enforce it. As we will see, this is not a significant issue. Moreover, such an approximation is not unprecedented. Sigal *et al.* [46], for instance, learn the conditional densities separately in the process of tracking motion.

Finally, we turn to incorporating the additional dashed edge in Fig. 2 in order to more tightly couple the LV boundary at the start and at the end of a complete cycle. Specifying this graphical model involves introducing a single additional edge potential to our existing model. We define this edge potential in exactly the same way as the other edge potentials in our framework, and learn the associated transition densities from training data. On tree-structured graphs, exact inference can be achieved using belief propagation (BP) for discrete or Gaussian variables. However, the formulae in (26) and (29) are no longer valid if there are loops in the graph. Inference for general loopy graphs is a challenging problem on which considerable current work is being performed. For example, methods do exist to "correct" belief propagation for the case of a single loop as in Fig. 2 (with the dashed edge). That said, it is also true that in many cases—especially when the loops are long [47]—simply applying the fixed point equation of (30)—i.e., performing local belief propagation updates neglecting the fact that there are loops in the graph—works well. Note that there is no counterpart to Doucet's forward-backward algorithm for loopy graphs, and, hence, to implement loopy belief propagation—i.e., the iterative application of (30) to the graph of Fig. 2—we use NBP.

V. EXPERIMENTAL RESULTS

A. Setup

We apply the proposed technique on 2-D mid-ventricular slices of MR data, but we also note that we can in principle apply the method to 3-D data. The dataset we use contains twenty-frame time sequences of breath-hold cardiac MR images, each representing a single cardiac cycle with the initial frame gated (synchronized) with an electrocardiogram (EKG) signal. We do not consider arrhythmias because only patients having sustained and hemodynamically-stable arrhythmias, a rare situation, can be practically imaged and analyzed. Anonymized data sets were obtained from the Cardiovascular MR-CT Program at Massachusetts General Hospital. Our training set consists of 42 cardiac cycles of 20 frames each for a total of 840 images, acquired from five patients. The segmentations on the training set were carried out manually by both radiologists as well as researchers whose results were reviewed by radiologists. All of the training and test sets come from healthy patients. If we had a rich enough set of training data, the model would theoretically still work with unhealthy patients. However, if the training set included some types of abnormal cardiac behavior and then we encountered yet a new/different type of abnormality in the test set there would be no guarantee how well it would work. Stated another way, healthy training data behave uniformly. If we had training data for a certain abnormality, the model certainly should be able to handle test data with cardiac behavior similar to this abnormality. The difficulty is finding a rich set of data for specific abnormalities. In the interest of brevity, representative results are presented here. However, we refer the reader to [48] for the full illustration of results.

In the training phase, we take the 840 images as well as manually segmented boundaries, normalize each boundary with respect to area, and then perform PCA to extract the first eight primary modes of variability as described in Section III-A. Using the PCA coefficients as well as the areas, we form the state vector X_t for each boundary sample at a particular time t. We then estimate the parameters of the intensity prior (Section II-B) $u, v, \sigma_u^2, \sigma_v^2$ by computing sample means and sample variances in high-SNR MR images. For the outside region $R_{\text{out}}(\vec{C}_t)$, we take a five-pixel wide band around the boundary (to provide context, the average area of the LV at end diastole is 1889 pixels while that at end systole is 380 pixels). We choose $R_{\text{out}}(\vec{C}_t)$ to have width five in an effort to ensure that this region contains only myocardial muscle, while providing a large enough region to have meaningful statistics.

Next we learn the maximally informative statistics as described in Section III-C. For Q_t, we assume that there is no interaction between the area and shape of the object. So, $Q_{A\vec{\alpha}} = 0$ and $Q_{\vec{\alpha}A} = 0$ in (11).[6] Thus, the learning of the area dynamics can be separated from that of the shape dynamics. For the shape

[6]Given a rich enough training data set, one can allow interactions between the area and shape when learning Q_t, but in this work, we assume a block diagonal Q_t matrix.

statistic $Q_{\vec{\alpha}}$, we have empirically determined that going from one to two dimensions led to a significant improvement, while subsequent increases in dimension did not lead to substantial gains in capturing shape dynamics. Thus we have taken the statistic Q_t to be 3-D (two dimensional for shape and one for area), so that Q_A is scalar and $Q_{\vec{\alpha}}$ is 2×8. From the training data, we learn the 17 parameters (1 for Q_A, 16 for $Q_{\vec{\alpha}}$) for each function Q. After learning the maximally informative statistics, we learn the transition densities as described in Section III-D. Although our framework is general enough to learn a different dynamical model at each time t, we chose to learn a single model for the systolic phase and a single model for the diastolic phase. The next step is to learn the dynamical models in the transition region, through mixtures of systolic and diastolic models. We determine the percentage contributions at each time t based on the percentage of the training data samples being at a particular phase at that time. As a result, we obtain a different mixture density, hence a different dynamic model, at each time point in the transition region.

Our test set consists of 234 cardiac cycles of 20 frames each, acquired from 26 patients, different from the patients used for the training set. Each cardiac cycle test set was estimated independently. Given a test image sequence to be segmented, we first pick a noise variance σ_n^2 (see Section II-C) based on the values empirically determined from the training data. We then use one of the inference algorithms described in Section IV to perform segmentation. To initialize our algorithms with some priors on the boundaries, we use the mean shape and area from the training data samples for that time instant. Thus, the test sets do not require user intervention for setup although one could theoretically provide the 'truth' for $t = 1$ to obtain a better set of segmentations. For causal filtering, and for forward-backward smoothing, we do this just for $t = 1$, whereas for parallel message passing in NBP we do it for each time t. To terminate the parallel message passing in NBP, at each iteration we compute the Kullback-Leibler (KL) divergence between successive estimates of the posterior density $p(X_t | y_{1:T})$ at each node, and stop when the maximum KL divergence over all nodes gets smaller than some predefined threshold. We use 200 particles to represent each of the densities involved in our algorithm. While the algorithm was created for accuracy and has not been optimized, the convergence time per image frame was about 30 s on a desktop personal computer having a single-core Xeon 2.2-GHz processor running MATLAB version 6.5 on a Linux OS.

B. Evaluation Metrics

We measure accuracy by computing the dice coefficient [49] (commonly used for evaluation of segmentations in medical imaging) between the segmentation and the manually segmented truth

$$\text{Dice}(R_1, R_2) \equiv \frac{2A(R_1 \cap R_2)}{A(R_1) + A(R_2)} \tag{35}$$

where R denotes the region inside a boudnary, $R_i \cap R_j$ represents the intersection of regions R_i and R_j, and $A(R_i)$ is the
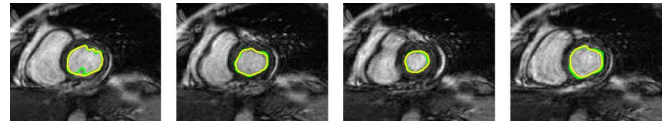

Fig. 5. LV segmentation results (yellow) of smoothing by NBP on a chain, on high-SNR observations. Ground truth shown in green.

area of region R_i. The dice measure evaluates to 1 when two regions are a perfect match and 0 when the regions are a complete mismatch.

1) Evaluation of LV Boundary Estimates $\vec{c}_t^*$: To determine the accuracy of a set of boundary estimates, we compute the dice measure between each of the estimates $\vec{C}_t^*$ and the ground truth at that time, and then average over all of the dice coefficients computed. For instance, if we have four cardiac cycles of 20 frames in our test set, we compute the dice measure for each of the 80 segmentations and then determine their average. We refer to this average as the *dice boundary coefficient*.

2) Evaluation of Samples of the Posterior of the State X_t: In addition to examining the accuracy of the boundary estimates, it may be instructive to evaluate the quality of the samples of the posterior density of X_t. To quantitatively examine samples, we can determine the dice coefficient between each sample from a posterior density $p(X_t | y_{1:T})$ and ground truth. Since none of these samples are expected to be as accurate as the LV boundary estimate $\vec{C}_t^*$, we expect the average dice coefficient from the samples to be smaller than the dice coefficient for the LV boundary estimate at the same frame. To determine the accuracy of samples, we compute the average across the samples in all of the frames analyzed. For example, if we have four cardiac cycles of 20 frames in our test set, with each posterior represented by 50 samples, then we compute the dice measure for all 4000 samples and then determine the average of these dice coefficients. Henceforth, we refer to this average as the *dice sample coefficient*.

We note that one of the advantages of the particle-based approach is that having the posterior distribution in principle allows us to understand more than simply looking at the dice sample coefficient. In particular having a density for shape conditioned on data and a dynamical model gives us a picture of the variability intrinsic to the problem-i.e., a representation of the posterior density is more informative about the remaining uncertainty after estimation than simply giving a single estimate.

C. Smoothing Results on the Chain

1) NBP: First we consider smoothing results on the chain using NBP. Figs. 5 and 6 show representative estimates $\vec{C}_t^*$ of the LV boundary for high and low SNR data, together with ground truth. The dice boundary coefficient in high SNR is 0.9214, while for low SNR it is 0.8909. All the dice coefficient computations in this paper are based on the test data consisting of 234 cardiac cycles.

When we constrast these results to causal filtering (dice boundary coefficient of 0.8654 for high SNR and 0.8210 for low SNR), we observe that smoothing results are more accurate. We would expect this result because these estimates effectively incorporate more data. Table I contains the numbers we have
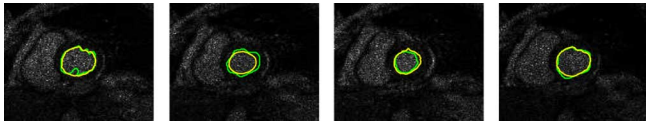
Fig. 6.   LV segmentation results (yellow) of smoothing by NBP on a chain, on low-SNR observations. Ground truth shown in green.

TABLE I
DICE BOUNDARY COEFFICIENTS ACHIEVED BY VARIOUS ALGORITHMS

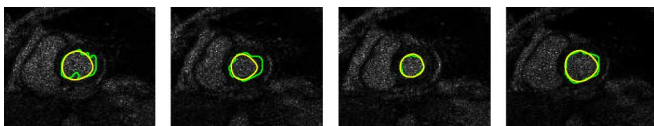| SNR	Shape Prior	Causal Filtering	Smoothing by Fw/Back	Smoothing by NBP	Loopy NBP
High	0.8516	0.8654	0.9210	0.9214	0.9292
Low	0.8035	0.8210	0.8836	0.8909	0.9069

Fig. 7.   LV segmentation results (yellow) of the forward-backward smoothing algorithm on low SNR observations. Ground truth shown in green.
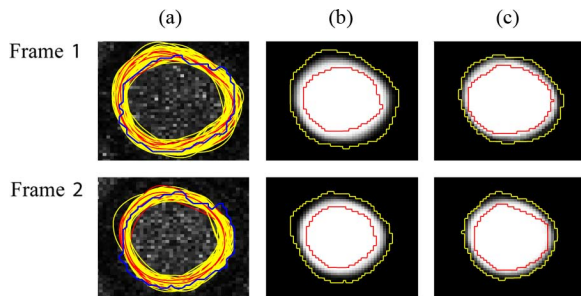

Fig. 8.   Comparison of filtering and forward-backward smoothing on low SNR data. (a) Samples of the filtering posterior $p(X_t|y_{1:t})$ are shown in yellow, while samples of the smoothing posterior $p(X_t|y_{1:T})$ are shown in red. The manually segmented truth is shown in blue. (b) Image which illustrates the distribution of the filtering samples, with the two curves (yellow and red) indicating the largest and smallest curves in the samples and the gray scale variations showing the frequency that a given pixel is contained within the curve samples (black indicating zero occurrences and white indicating that the point is contained in all of the sampled curves). (c) Image which shows the distribution of the smoothing samples in a similar manner to (b), with the two curves again indicating the largest and smallest curves.

mentioned here. For causal filtering-based segmentation examples, please see [43].

2) Forward-Backward Method: We next present results for smoothing on a Markov chain using the forward-backward method. Fig. 7 shows the segmentations using low SNR data. The dice boundary coefficient is 0.8836. This result is similar to the 0.8909 obtained by NBP on the same Markov chain and better than the 0.8210 obtained using filtering. Note that filtering is essentially just the forward step of the forward-backward procedure. Table I contains the numbers we have mentioned here.

To provide a visual comparison of filtering and smoothing, Fig. 8(a) shows the most representative samples of the posterior density $p(X_t|y_{1:t})$ for filtering (yellow) and the posterior density $p(X_t|y_{1:T})$ for forward-backward smoothing (red), to-

TABLE II
DICE SAMPLE COEFFICIENTS FOR MARKOV CHAIN SMOOTHING USING THE FORWARD-BACKWARD METHOD COMPARED WITH THAT FROM FILTERING. THE VALUES ARE BASED ON THE 50 MOST HEAVILY WEIGHTED SAMPLES OF THE POSTERIOR

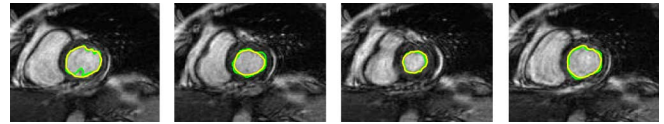| Dice *sample* coefficient	Causal Filtering	Smoothing by Forward-Backward
High SNR	0.7683	0.8452
Low SNR	0.6951	0.7904

Fig. 9.   LV segmentations (yellow) obtained from loopy NBP on a high-SNR cardiac MR image sequence. Ground truth shown in green.
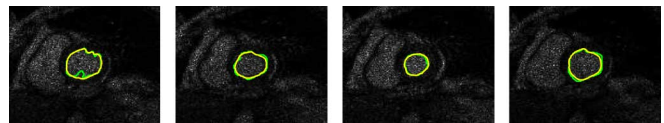

Fig. 10.   LV segmentations (yellow) obtained from loopy NBP on a low SNR cardiac MR sequence. Ground truth shown in green.

gether with the manually-segmented truth (blue) for two representative frames. Qualitatively, we observe that the more erroneous curves mostly belong to the filtering posterior. Fig. 8(b) and 8(c) shows the variability of the curves in a different way for filtering and smoothing, respectively. This suggests that the smoothing posterior is tighter. Quantitatively, for low-SNR data, the dice sample coefficient is 0.6951 for filtering and 0.7904 for forward-backward smoothing. Table II shows the results of the dice sample coefficient for high and low SNR data.

D. Single Cycle Graph Results

Figs. 9 and 10 show the segmentations obtained using loopy NBP on the high and low SNR data, respectively. The results are in general very accurate. For high-SNR data, the dice boundary coefficient is 0.9292, while that for low SNR data is 0.9069. The results show an improvement of estimates over those from smoothing on a chain. When we examine the segmentations at each frame more carefully (not shown here), we observe that the improvement provided by the loopy model is most significant near the beginning and end of the cardiac cycle, as one would expect.

E. Comparison of Estimates With Static Segmentations

We also visually demonstrate the benefits of our approach by comparing our segmentation results with the static, shape prior-based segmentation algorithm of Tsai *et al.* [29]. Fig. 11 shows our results using NBP on a loopy graphical model together with the results of the method in [29] on the same data. The static segmentation method of [29] uses the same training data as our approach for learning the shape priors. We qualitatively observe that the loopy NBP approach leads to more accurate segmentations. This observation is supported quantitatively
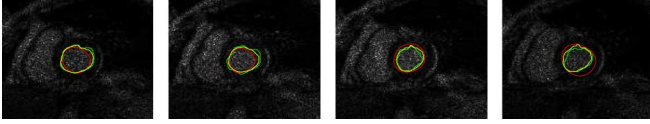
Fig. 11. Comparison of the LV boundary estimates of loopy NBP with static segmentations using a shape prior (dark red) based on [29]. Ground truth is shown in green.

by the dice boundary coefficients in Table I. The static segmentation method achieves 0.8516 and 0.8035, for the high and low SNR cases, respectively, while the corresponding numbers for loopy NBP are 0.9292 and 0.9069.

VI. CONCLUSION

In this paper, we have presented an approach for time-recursive segmentation of evolving deformable objects. In particular, we have proposed nonparametric, information-theoretic methods to learn the time-varying dynamics of such deformable objects from training data, with the specific application of LV evolution in cardiac imaging. Using a Bayesian framework, we have developed particle-based smoothing algorithms that allow time-recursive segmentation throughout the LV cycle using all available data. In our development, we have used not only Markov chains for the temporal LV evolution, but also a single-cycle graph to capture periodicity of cardiac motion. We have adapted and used message passing algorithms for approximate solution of these inference problems. In this framework, we have used finite-dimensional approximate representations of the LV boundary as state variables, rather than the infinite-dimensional LV boundaries themselves, in order to improve the reliability and accuracy of the learning process. However, we have also incorporated level set methods in a statistically consistent manner into this framework in order to generate boundaries not constrained to reside in the finite-dimensional domain in which dynamics have been learned and exploited. We have demonstrated the effectiveness of our approach (both qualitatively and quantitatively) on cardiac MR imagery. These results exhibit the improvements provided by our approach over static segmentation methods.

APPENDIX I
PARTICLE FILTERS

For dynamical systems whose state evolution involves non-parametric densities, sample-based methods, such as particle filters [40], [41], can be used for recursive state estimation. These sequential Monte Carlo (MC) techniques represent the density through a discrete set of weighted samples drawn from that density which in turn approximate the density through a kernel density estimate.

Suppose X_t is a Markov process we want to estimate based on measurements y_t that represent noisy observations of the underlying process X_t. To determine the posterior distribution

$p(X_t|y_{1:t})$, where $y_{1:t} \equiv [y_1, y_2, \ldots, y_t]$, the set of all observations from the initial time to time t, we apply Bayes' Rule

$$p(X_t|y_{1:t}) = p(X_t|y_t, y_{1:t-1}) = \frac{p(X_t, y_t|y_{1:t-1})}{p(y_t)}$$
$$\propto p(y_t|X_t, y_{1:t-1}) \int_{X_{t-1}} p(X_t|X_{t-1}, y_{1:t-1})$$
$$\cdot p(X_{t-1}|y_{1:t-1}) dX_{t-1}. \quad (36)$$

This expression simplifies to

$$p(X_t|y_{1:t})$$
$$\propto p(y_t|X_t) \int_{X_{t-1}} p(X_t|X_{t-1}) p(X_{t-1}|y_{1:t-1}) dX_{t-1} \quad (37)$$

because X_t is Markov and the observations y_t are independent from each other conditioned on their associated state X_t. From this equation, we can observe the recursive nature of the problem. In particle filtering, the continuous distribution at any time t is approximated by a sample-based representation

$$p(X_t|y_{1:t}) \approx \sum_{i=1}^{M} w_t^{(i)} k\left(x; x_t^{(i)}\right) \quad (38)$$

where $k(x; x_t^{(i)})$ represents kernel centered at $x_t^{(i)}$. Suppose that at time $t-1$, we have samples $x_{t-1}^{(i)}$ having weights $w_{t-1}^{(i)}$ to approximate $p(X_{t-1}|y_{1:t-1})$. The points propagate through the dynamics to sample points $x_t^{(i)}$ having weights $w_{t|t-1}^{(i)}$. These points are then weighted by the likelihood, namely

$$w_t^{(i)} = \frac{1}{Z} p\left(y_t|x_t^{(i)}\right) w_{t|t-1}^{(i)} \quad (39)$$

where Z is a normalizing constant which ensures that the weights sum to 1.

One of the key concepts of particle filters is the idea of importance sampling. If a given distribution $p(x)$ cannot be sampled, we resort to sampling from a proposal density $\pi(x)$ and apply the weight of $p(x)/\pi(x)$ to the samples. The weighting for each sample $x_t^{(i)}$ is necessary to adjust for the fact that we sample from $\pi(x)$ rather than $p(x)$, the distribution of interest. Mathematically

$$w_t^{(i)} \propto \frac{p(x_t^{(i)})}{\pi(x_t^{(i)})} \quad (40)$$

where in order to maintain a valid probability distribution, we normalize the weights such that $\sum_i w_t^{(i)} = 1$.

Another feature of particle filters is resampling. After a few iterations in the recursive filtering process, most of the particle weight may become concentrated among a small set of samples, leading to a degenerate distribution. The process of resampling can mitigate this problem. Resampling involves generating a new set of samples $x_t^{(j)}$ from the approximate representation $p(x) \approx \sum_{i=1}^{M} w_t^{(i)} k(x; x_t^{(i)})$. Resampling can be accomplished in many different ways [40].

APPENDIX II
FORMULAE FOR COORDINATE DESCENT

Given the functional of (20), we solve the problem using co-ordinate descent.[7] For the f_t step, we fix the boundary $\vec{C}_t$ and compute the intensity estimates. Using the assumption that f_t is piecewise constant, the f_t which minimizes $E(f_t, \vec{C}_t)$ for a given $\vec{C}_t$ is

$$
\begin{aligned}
f_t^{R_{\text{in}}(\vec{C}_t)} &= \exp\left(\frac{\frac{1}{\sigma_n^2} \int_{R_{\text{in}}(\vec{C}_t)} \log y_t(z)dz + \frac{u}{\sigma_u^2}}{\frac{1}{\sigma_n^2} \int_{R_{\text{in}}(\vec{C}_t)} dz + \frac{1}{\sigma_u^2}} \right) \\
f_t^{R_{\text{out}}(\vec{C}_t)} &= \exp\left(\frac{\frac{1}{\sigma_n^2} \int_{R_{\text{out}}(\vec{C}_t)} \log y_t(z)dz + \frac{v}{\sigma_v^2}}{\frac{1}{\sigma_n^2} \int_{R_{\text{out}}(\vec{C}_t)} dz + \frac{1}{\sigma_v^2}} \right). \quad (41)
\end{aligned}
$$

For a given f_t, we apply curve evolution to find the $\vec{C}_t$ that minimizes $E(f_t, \vec{C}_t)$. To accomplish this, we compute the first variation of $E(f_t, \vec{C}_t)$ with respect to $\vec{C}_t$ and move in that direction. The formula for the first variation is

$$
\begin{aligned}
\frac{\partial \vec{C}_t}{\partial \tau}(z) = -\Big[& \alpha(F_t^{\text{out}}(\vec{C}_t) - F_t^{\text{in}}(\vec{C}_t)) \\
& \cdot (2\log y_t(z) - F_t^{\text{out}}(\vec{C}_t) - F_t^{\text{in}}(\vec{C}_t)) \\
& + 2\beta_u \frac{(F_t^{\text{in}}(\vec{C}_t) - u)}{A_{\text{in}}}(F_t^{\text{in}}(\vec{C}_t) - \log y_t(z)) \\
& + 2\beta_v \frac{(F_t^{\text{out}}(\vec{C}_t) - v)}{A_{\text{out}}}(F_t^{\text{out}}(\vec{C}_t) - \log y_t(z)) \\
& + \frac{1}{Q} \sum_i \exp(-D^2(\vec{C}_t, x_{t|t-1}^{(i)}))(\nabla d^2(z, x_{t|t-1}^{(i)}) \cdot \vec{N} \\
& + d^2(z, x_{t|t-1}^{(i)})\kappa(z))\Big]\vec{N}
\end{aligned} \quad (42)
$$

where $\alpha = 1/2\sigma_n^2$, $\beta_u = 1/2\sigma_u^2$, $\beta_v = 1/2\sigma_v^2$, $Q = \sum_i \exp(-D^2(\vec{C}_t, x_{t|t-1}^{(i)}))$, A_{in} is the area of R_{in} and A_{out} is the area of R_{out}, $\kappa(z)$ is the curvature of $\vec{C}$ at z, $\vec{N}$ is the unit outward normal of $\vec{C}$ at z, and τ is an iteration-time parameter used during the curve evolution process. The computation of the first variation relies on four separate derivations of curve flows [24], [26], [50], [51].

REFERENCES

[1] C. Gu and M. C. Lee, "Semiautomatic segmentation and tracking of semantic video objects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 5, pp. 572–584, May 1998.

[2] B. Gunsel, M. Ferman, and A. Tekalp, "Temporal video segmentation using unsupervised clustering and semantic object tracking," *J. Electron. Imag.*, vol. 7, pp. 592–604, 1998.

[3] F. Lv, J. Kang, R. Nevatia, I. Cohen, and G. Medioni, "Automatic tracking and labeling of human activities in a video sequence," presented at the IEEE Int. Workshop Performance Evaluation of Tracking and Surveillance (in ECCV), 2004.

[4] H. Buxton and S. Gong, "Advanced visual surveillance using Bayesian networks," presented at the IEEE Workshop on Context-Based Vision, 1995.

[5] C. R. Williams, J. L. Bamber, A. Wilmshurst, N. Stapleton, and J. Scott, "Detection and tracking of oceanic thermal boundaries using passive microwave data," in *Proc. Int. Geoscience and Remote Sensing Symp.*, 2000, vol. 3, IEEE, pp. 1092–1094.

[6] A. Gangopadhyay and A. R. Robinson, "Feature-oriented regional modeling of oceanic fronts," *Dyn. Atmos. Oceans*, vol. 36, no. 1-3, pp. 201–232, 2002.

[7] W. Sun, M. Cetin, W. Thacker, T. Chin, and A. Willsky, "Variational approaches on discontinuity localization and field estimation in sea surface temperature and soil moisture," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 2, pp. 336–350, Feb. 2006.

[8] J. Duncan, A. Smeulders, F. Lee, and B. Zaret, "Measurement of end diastolic shape deformity using bending energy," *Comput. Cardiol.*, pp. 277–280, 1988.

[9] J. C. McEachen , II and J. S. Duncan, "Shape-based tracking of left ventricular wall motion," *IEEE Trans. Med. Imag.*, vol. 16, no. 3, pp. 270–283, Mar. 1997.

[10] G. K. von Schulthess, *The Effects of Motion and Flow on Magnetic Resonance Imaging.* New York: Springer Verlag, 1989.

[11] A. Goshtasby and D. A. Turner, "Segmentation of cardiac cine MR images for extraction of right and left ventricular chambers," *IEEE Trans. Med. Imag.*, vol. 14, no. 1, pp. 56–64, Jan. 1995.

[12] G. Hamarneh and T. Gustavsson, "Combining snakes and active shape models for segmenting human left ventricle in echocardiographic images," *Comput. Cardiol.*, vol. 27, pp. 115–118, 2000.

[13] X. Papademetris, A. J. Sinusas, D. P. Dione, and J. S. Duncan, "Estimation of 3D left ventricular deformation from echocardiography," *Med. Imag. Anal.*, vol. 5, pp. 17–28, 2001.

[14] V. Chalana, D. T. Linker, D. R. Haynor, and Y. Kim, "A multiple active contour model for cardiac boundary detection on echocardiographic sequences," *IEEE Trans. Med. Imag.*, vol. 15, no. 3, pp. 290–298, Mar. 1996.

[15] M.-P. Jolly, N. Duta, and G. Funka-Lee, "Segmentation of the left ventricle in cardiac MR images," in *Proc. IEEE Int. Conf. Computer Vision*, 2001, vol. 1, pp. 501–508.

[16] X. S. Zhou, D. Comaniciu, and A. Gupta, "An information fusion framework for robust shape tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 1, pp. 115–129, Jan. 2005.

[17] J. Senegas, T. Netsch, C. A. Cocosco, G. Lund, and A. Stork, "Segmentation of medical images with a shape and motion model: A Bayesian perspective," in *Proc. Computer Vision Approaches to Medical Image Analysis (CVAMIA) and Mathematical Methods in Biomedical Image Analysis (MMBIA) Workshop*, 2004, pp. 157–168.

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

[19] N. Vaswani, A. Yezzi, Y. Rathi, and A. Tannenbaum, "Particle filters for infinite (or large) dimensional state spaces—parts 1 and 2," presented at the IEEE Int. Conf. Acoustics, Speech, and Signal Processing, 2006.

[20] 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.*, vol. 29, no. 8, pp. 1470–1475, Aug. 2007.

[21] D. Cremers, "Dynamical statistical shape priors for level set-based tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1262–1273, Aug. 2006.

[22] S. L. Lauritzen, *Graphical Models.* Oxford, U.K.: Clarendon, 1996.

[23] M. I. Jordan, "Graphical models," *Statist. Sci.*, vol. 19, pp. 140–155, 2004.

[24] Y. Chen, H. Tagare, S. Thiruvenkadam, F. Huang, D. Wilson, K. Gopinath, R. Briggs, and E. Geiser, "Using prior shapes in geometric active contours in a variational framework," *Int. J. Comput. Vis.*, vol. 50, no. 3, pp. 315–328, 2002.

[25] D. Mumford and J. Shah, "Boundary detection by minimizing functionals I," presented at the IEEE Conf. Computer Vision and Pattern Recognition, 1985.

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

[27] J. A. Sethian, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Material Science.* Cambridge, U.K.: Cambridge Univ. Press, 1996.

[28] M. Leventon, "Statistical models in medical image analysis," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, 2000.

[7]This development generalizes in a straightforward fashion to the analogous optimization problem (33) in smoothing.

[29] A. Tsai, A. Yezzi, W. Wells, C. Tempany, D. Tucker, A. Fan, W. E. Grimson, and A. Willsky, "A shape-based approach to the segmentation of medical imagery using level sets," *IEEE Trans. Med. Imag.*, vol. 22, no. 2, Feb. 2003.

[30] A. McCulloch, J. B. Bassingthwaighte, P. J. Hunter, D. Noble, and T. L. Blundell, "Computational biology of the heart: From structure to function," *Progr. Biophys. Molec. Biol.*, vol. 69, no. 2-3, pp. 153–155, 1998.

[31] M. Sermesant, C. Forest, X. Pennec, H. Delingette, and N. Ayache, "Deformable biomechanical models: Applications to 4D cardiac image analysis," *Med. Imag. Anal.*, vol. 7, no. 4, pp. 475–488, 2003.

[32] P. C. Shi, A. J. Sinusas, R. T. Constable, and J. S. Duncan, "Volumetric deformation analysis using mechanics-based data fusion: Applications in cardiac motion recovery," *Int. J. Comput. Vis.*, vol. 35, no. 1, pp. 65–85, 1998.

[33] H. F. Liu and P. C. Shi, "Biomechanically constrained multiframe estimation of nonrigid cardiac kinematics from medical image sequences," presented at the Int. Conf. Inverse Problems and Numerics, 2002.

[34] A. Ihler, "Maximally informative subspaces: nonparametric estimation for dynamical systems," M.S. thesis, Massachusetts Inst. Technol., Cambridge, MA, 2000.

[35] P. Hall, S. J. Sheather, M. C. Jones, and J. S. Marron, "On optimal data-based bandwidth selection in kernel density estimation," *Biometrika*, vol. 78, no. 2, pp. 263–269, 1991.

[36] J. E. Wall, A. S. Willsky, and N. R. Sandell, "On the fixed-interval smoothing problem," *Stochastics*, vol. 5, pp. 1–42, 1981.

[37] G. C. Verghese and T. Kailath, "A further note on backwards markovian models," *IEEE Trans. Inf. Theory*, vol. 25, no. 1, pp. 121–124, Jan. 1979.

[38] B. D. O. Anderson and T. Kailath, "Forwards and backwards models for finite-state markov processes," *Adv. Appl. Probab.*, vol. 11, no. 1, pp. 118–133, 1978.

[39] B. D. O. Anderson and I. B. Rhodes, "Smoothing algorithms for nonlinear finite-dimensional systems," *Stochastics*, vol. 9, pp. 139–165, 1983.

[40] 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.

[41] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statist. Comput.*, vol. 10, no. 3, pp. 197–208, 2000.

[42] A. Tsai, A. Yezzi, and A. S. Willsky, "Curve evolution implementation of the Mumford–Shah functional for image segmentation, denoising, interpolation, and magnification," *IEEE Trans. Image Process.*, vol. 10, no. 8, pp. 1169–1186, Aug. 2001.

[43] W. Sun, M. Çetin, R. Chan, V. Reddy, G. Holmvang, V. Chandar, and A. Willsky, "Segmenting and tracking the left ventricle by learning the dynamics in cardiac images," presented at the Int. Conf. Information Processing in Medical Imaging, 2005.

[44] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. San Mateo, CA: Morgan Kaufman, 1988.

[45] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky, "Nonparametric belief propagation," presented at the IEEE Conf. Computer Vision and Pattern Recognition. IEEE, 2003.

[46] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard, "Tracking loose-limbed people," presented at the IEEE Conf. on Computer Vision and Pattern Recognition. IEEE, 2004.

[47] Y. Weiss, "Correctness of local probability propagation in graphical models with loops," *Neural Comput.*, vol. 12, pp. 1–41, 2000.

[48] W. Sun, "Learning the dynamics of deformable objects and recursive boundary estimation using curve evolution techniques," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, 2005.

[49] L. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, pp. 297–302, 1945.

[50] J. Kim, M. Cetin, and A. S. Willsky, "Nonparametric shape priors for active contour-based image segmentation," *Signal Process.*, vol. 87, no. 12, pp. 3021–3044, 2007.

[51] A. Yezzi, A. Tsai, and A. Willsky, "A fully global approach to image segmentation via coupled curve evolution equations," *J. Vis. Commun. Image Represent.*, vol. 13, pp. 195–216, 2002.

Walter Sun (S'94–M'03) received the B.S. degree from the Georgia Institute of Technology, Atlanta, and the M.S., E.E., and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, in 1995, 1997, 2004, and 2005, respectively, all in electrical engineering.

He is an NDSEG Graduate Fellowship recipient. He is currently with Microsoft Corporation, Redmond, WA, and is also an Adjunct Professor at Seattle University in the Electrical and Computer Engineering Department. His research interests include statistical signal and image processing, computer vision, stochastic finance, multimedia search, and image coding.

Müjdat Çetin (S'98–M'02) received the Ph.D. degree from Boston University, Boston, MA, in 2001, in electrical engineering.

From 2001 to 2005, he was with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge. Since September 2005, he has been an Assistant Professor at Sabanci University, Istanbul, Turkey. His research interests include statistical signal and image processing, inverse problems, computer vision, data fusion, wireless sensor networks, biomedical information processing, radar imaging, brain computer interfaces, machine learning, and sensor array signal processing.

Dr. Çitin has received a number of awards, including the 2008 Turkish Academy of Sciences Young Scientist Award and the 2007 Elsevier Signal Processing Journal Best Paper Award. He has served in various organizational capacities, including special session organizer, session chair, and technical program committee member for a number of conferences including the IEEE International Conference on Acoustics, Speech, and Signal Processing; the SPIE Conference on Algorithms for Synthetic Aperture Radar Imagery; the IEEE Statistical Signal Processing Workshop; the IEEE International Conference on Image Processing; and the EURASIP European Signal Processing Conference. He has also served as the Technical Program Co-Chair for the 2006 IEEE Turkish Conference on Signal Processing and Communications Applications.

Raymond Chan received the B.S. degree in biomedical engineering and electrical engineering from the Johns Hopkins University, Baltimore, MD, in 1992, the M.S. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, in 1996, and the Ph.D. degree from the Harvard-MIT Division of Health Sciences and Technology in 2001.

His research expertise is in interventional guidance, imaging biomarkers for clinical studies, multimodality cardiovascular imaging, and computation methods for estimation of anatomical and functional information from biomedical data. At the Massachusetts General Hospital, he was an Instructor of radiology and a Principal Investigator for Philips Research North America with joint appointments in the Cardiovascular MR, PET, & CT program, the Wellman Center for Photomedicine, the Cardiac Arrhythmia Unit, and the Echocardiography Unit. He is now a Research Director for the Philips Healthcare Clinical Research Board in the western United States.

Alan S. Willsky (S'70–M'73–SM'82–F'86) joined the Massachusetts Institute of Technology, Cambridge, in 1973, and is the Edwin Sibley Webster Professor of Electrical Engineering and Co-Director of the Laboratory for Information and Decision Systems. He was a founder of Alphatech, Inc., and Chief Scientific Consultant, a role in which he continues at BAE Systems Advanced Information Technologies. From 1998-2002, he served on the U.S. Air Force Scientific Advisory Board. He has received several awards, including the 1975 American Automatic Control Council Donald P. Eckman Award, the 1979 ASCE Alfred Noble Prize, the 1980 IEEE Browder J. Thompson Memorial Award, the IEEE Control Systems Society Distinguished Member Award in 1988, the 2004 IEEE Donald G. Fink Prize Paper Award, and Doctorat Honoris Causa from Université de Rennes in 2005. He has delivered numerous keynote addresses and is co-author of the text *Signals and Systems*. His research interests are in the development and application of advanced methods of estimation, machine learning, and statistical signal and image processing.