# Variable Length Open Contour Tracking Using a Deformable Trellis

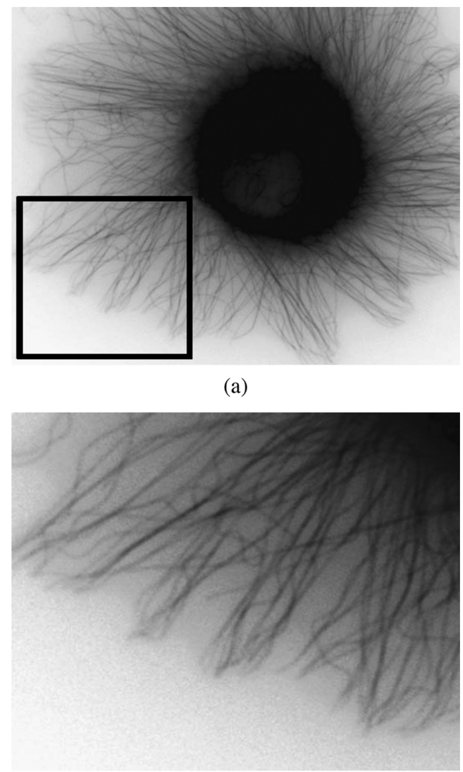
Mehmet Emre Sargin, *Member, IEEE*, Alphan Altinok, *Member, IEEE*, Bangalore S. Manjunath, *Fellow, IEEE*, and Kenneth Rose, *Fellow, IEEE*

***Abstract*—This paper focuses on contour tracking, an important problem in computer vision, and specifically on open contours that often directly represent a curvilinear object. Compelling applications are found in the field of bioimage analysis where blood vessels, dendrites, and various other biological structures are tracked over time. General open contour tracking, and biological images in particular, pose major challenges including scene clutter with similar structures (e.g., in the cell), and time varying contour length due to natural growth and shortening phenomena, which have not been adequately answered by earlier approaches based on closed and fixed end-point contours. We propose a model-based estimation algorithm to track open contours of time-varying length, which is robust to neighborhood clutter with similar structures. The method employs a deformable trellis in conjunction with a probabilistic (hidden Markov) model to estimate contour position, deformation, growth and shortening. It generates a maximum *a posteriori* estimate given observations in the current frame and prior contour information from previous frames. Experimental results on synthetic and real-world data demonstrate the effectiveness and performance gains of the proposed algorithm.**

***Index Terms*—Biomedical image processing, tracking.**

## I. Introduction

OBJECT tracking has been one of the most challenging problems in computer vision with diverse applications. In this context, tracking is commonly aided by contour representation of the object where application specific challenges, such as noise, clutter, etc., are partially addressed by enforcing constraints on the contour topology. In the special case of open contours, typically representing a curvilinear, thin elongated object, the assumption of fixed start and end points restricts the search space and enables fast marching and level set methods to produce very fast estimates of the contour position [1]. However, the invalidity of these assumptions in general, combined with interference of other similar objects, present considerable challenges for open contour tracking. These issues are central to the applications found in the emerging field of bioimage analysis. Typically, curvilinear structures such as blood vessels, dendrites, microtubules, and other filamentous structures are tracked over time for subsequent statistical analysis.



(a)

(b)

Fig. 1. (a) Frame showing the microtubules in a cell, (b) enlarged region.

Biological and specifically microscopy images often exacerbate the difficulties encountered in general imagery. For example, typical fluorescence microscopy images would suffer from additive fluorescence in clutter, or fluorescence degradation over time, known as photobleaching. Furthermore, often similar structures occlude each other or otherwise interfere with the objects of interest which is illustrated in Fig. 1. Finally, many such sub-cellular structures are assembled and disassembled to their building blocks during the course of their function, hence eliminating useful simplifying assumptions about length, fixed start and end points, etc. In this paper, we address two main issues in open contour tracking: (i) natural clutter by other contours, and (ii) time-varying length of the contour. Earlier approaches based on active contours and point matching do

Manuscript received June 16, 2009; revised December 14, 2009 and May 10, 2010; accepted August 21, 2010. Date of publication September 30, 2010; date of current version March 18, 2011. This work was conducted while M.E. Sargin and A. Altinok were with the Department of Electrical and Computer Engineering, University of California, Santa Barbara. This work was supported by Center for Bioimage Informatics under Grant NSF-ITR 033169. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Birsen Yazici.

M. E. Sargin is with Google Inc., Mountain View, CA 94043 USA (e-mail: emresargin@gmail.com).

A. Altinok is with the Biological Network Modeling Center, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: alphan@caltech.edu).

B. S. Manjunath and K. Rose are with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106 USA (e-mail: manj@ece.ucsb.edu; rose@ece.ucsb.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2010.2081680

not adequately address these issues. In the active contour approaches [2], end points are assumed to be fixed, hence any length changes due to end point motion (or growth and shortening) are not captured. On the other hand, tip matching-based tracking schemes [3] are highly prone to clutter.

For the purposes of demonstrating our approach, we applied it to microtubule image sequences which exhibit the challenges described above. Besides being a challenging example for open contour tracking, microtubules are highly important in their own right as their dynamic behavior is a current focus of attention in enhancing the understanding of Alzheimer's disease and cancer. Although the proposed method is designed for curvilinear structures without branches, it can be extended to other similar bioimage problems involving branching tubular structures such as the analysis of dendrites and blood vessels.

### A. Related Work

Most tracking algorithms start with an initial contour and incrementally perform tracking by updating the predicted contour position on each consecutive frame. Therefore, initial contour detection plays an important role in the contour tracking process. Manual initialization is an option in some applications, but is usually not preferred in many other applications due to poor sampling resolution and accuracy.

Satellite imagery is one of the major applications, where the objects of interest are roads, rivers, etc. In [4], the authors present a method based on steerable line profile matching with multiple scales. A robust detection of contour points followed by a linking algorithm based on the line profile orientations is presented in [5]. An alternative strategy is to attach points to the current contour based on the local information [6], where the orientation of the contour and the local filter response are used together to minimize the uncertainty about the point to be attached to contour. Blood vessel extraction is another key application on retinal imaging domain [7], [8]. A detailed review on vessel extraction methods can be found in [9].

The contour position in the current frame is roughly estimated based on the previous positions of the contour, and is then adjusted given the observations in the current frame. The prediction schemes range in complexity from simply using the previous contour position to sophisticated and complex methods. Kalman filter-based prediction and active contour-based adjustment can be combined in a single framework [10], [11]. A statistical framework for active contour-based tracking is presented in [12]. In [13], the authors present a geodesic active contour-based adjustment for tracking. More recent work on active contour and level set-based tracking can be found in [14]–[17].

Active contours provide an iterative framework for contour adjustment starting from an initial contour. While active contours represent an intuitive approach for tracking, it is susceptible to local optimum traps. Motivated by this fact, algorithms based on dynamic programming are proposed to obtain globally optimal solution at reasonable computational complexity [18]–[20]. The Hidden Markov Model (HMM) has been used in tracking of closed contours in [21], [22] to combine different visual cues in a probabilistic framework. HMM provides globally optimal solutions for contour adjustment with the Viterbi decoding which is based on dynamic

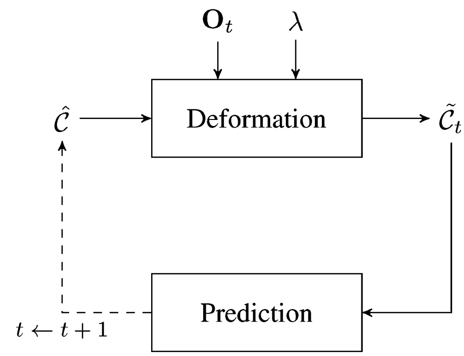



Fig. 2. System block diagram. The output of the current frame is fed to the following frame as the prior estimate.

programming. This critical advantage was exploited in our preliminary work to perform open contour tracking using a deformable trellis with ad hoc parametrization [23]. This paper subsumes [23] and provides a comprehensive method including model parameter learning schemes and overall optimization of the probabilistic model without recourse to ad hoc or manually tuned parameters.

## II. Method

The problem of open contour tracking can be separated into two sub-problems. The first problem is the detection of the open contour on the initial frame. Traditionally, this is the initialization step, where the initial contour position may be marked by the user or it can be estimated by a curve detection algorithm. The second problem is the adjustment of contour point locations on consecutive frames. We will refer to the first problem as *tracing*, and the second problem as *tracking*. In the case of varying contour length, tracking must additionally account for potential extension of the contour length. Thus, either the contour points must be adjusted to cover the extension, necessitating the detection of the new ending point of the contour, or alternatively the tracing paradigm should be incorporated to handle contour extensions. Here, we opt for the second method for the following reasons: (i) detecting the contour end is a challenging task, especially when other contour tips are present in the vicinity, and (ii) adjusting of the contour points to account for both lateral displacement as well as the displacement along the contour is ill posed and considerably harder. We will further see that the shortening and growing cases are different types of problems and therefore should be handled separately.

On each frame at time $t$, the tracking is performed by deforming the predicted contour position, $\hat{\mathcal{C}}$, based on the image observations $\mathbf{O}_t$ and the model $\lambda$. The resulting estimate $\tilde{\mathcal{C}}_t$ is then employed as prior information on the contour position in the next frame, at $t+1$. The initial contour position $\hat{\mathcal{C}}$ at $t=1$ is the output of tracing at $t=0$. A block diagram of tracking method is shown in Fig. 2.

### A. Open Contour Adjustment on a Deformable Trellis

We introduce the *deformable trellis* on a fixed-length contour segment, postponing for later discussion of how to handle length variations. The main idea underlying the deformable trellis is the ability to express (probabilistically) the lateral deformations

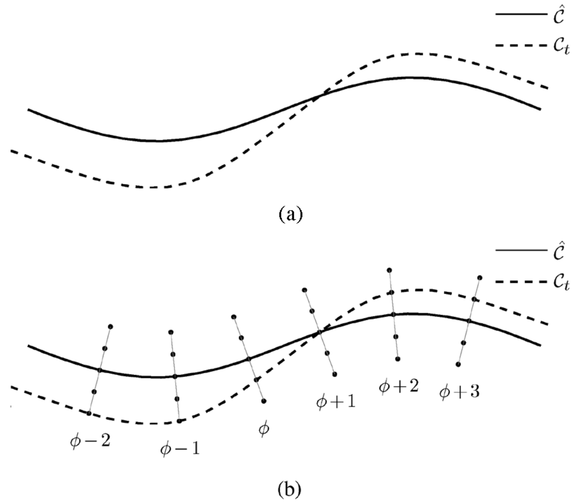



Fig. 3. (a) Possible deformations of $\mathcal{C}$ at $t$ with respect to the estimate $\hat{\mathcal{C}}$, (b) corresponding trellis covering deformations.

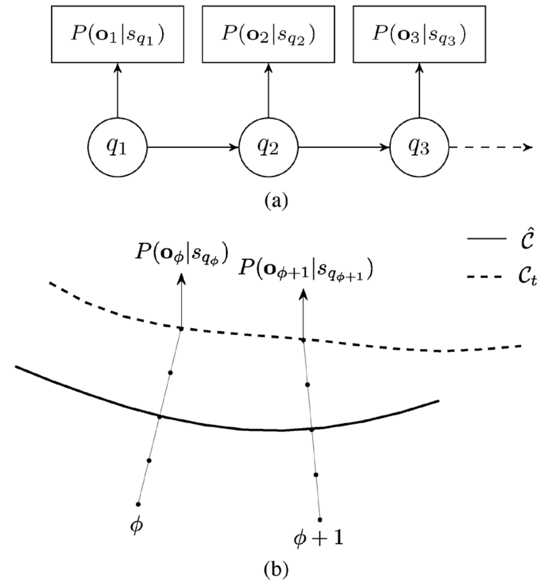



Fig. 4. (a) HMM transitions $\mathbf{A}$ and emissions $\mathbf{B}$ are shown on the trellis, (b) corresponding observation probabilities.

along the contour. Consider a contour $\mathcal{C}$ undergoing a deformation between frames $t$ and $t+1$, Fig. 3(a). We construct a two dimensional lattice along $\hat{\mathcal{C}}$ as follows.

First, we select $M$ equally spaced points along the initial contour estimate $\hat{\mathcal{C}}$. Let $\mathbf{x}_\phi$, $\phi = 1, \ldots, M$, represent these points. At each $\mathbf{x}_\phi$ we draw a normal to the $\hat{\mathcal{C}}$ enumerated by index $\phi$ and define $N$ equally spaced points along these normals. Let $\psi$, $\psi = 1, \ldots, N$, be the indexes of points along the normals. Thus, we create a lattice around $\hat{\mathcal{C}}$ represented by $\mathbf{x}_{\phi,\psi}$, which we refer to as the *trellis*, Fig. 3(b). The initial contour goes through the sequence of points indexed by $\phi = 1, 2, \ldots, M$ and $\psi = (N+1)/2$ ($N$ is odd). Deformations of the contour are obtained on this trellis by varying the values of $\psi$ at each $\phi$. The trellis is reconstructed after each contour position update and evolves with the frames, thereby making the trellis *deformable*. Based on the described structure, we define the contour adjustment problem as *finding the optimal sequence of points* on the trellis (or a path through the trellis) that jointly accounts for the prior on the contour position and the evidence observed on the current frame.

We formulate the problem within a probabilistic framework by mapping the contour position to a state sequence of a hidden Markov model (HMM), [24]. The points along a normal correspond to the states, $s_\psi$, of the HMM, and a path on the trellis corresponds to a *state sequence* of the HMM. The set of all possible *state sequences* is denoted by $\mathbf{Q} = \{\mathbf{q}\}$ where $\mathbf{q}$ is a sample state sequence drawn from $\mathbf{Q}$. Each element of $\mathbf{q}$, denoted by $q_\phi$, is the state on the normal at $\phi$.

We define the image observations as the *emission probabilities* of the HMM. The emission probability at each state reflects the probability of observing the image feature if $\hat{\mathcal{C}}$ passed through that specific state. Thus, the contour adjustment problem can now be expressed as *finding the most probable state sequence* in the HMM, which can be efficiently calculated by the Viterbi algorithm. Given the image observations, $\mathbf{O}_t$, and the HMM, $\lambda$, the optimal state sequence $\hat{\mathbf{q}}$ is given by (Fig. 2)

$$\hat{\mathbf{q}} = \arg\max_{\mathbf{q}\in\mathbf{Q}} P(\mathbf{q}|\mathbf{O}_t, \lambda). \tag{1}$$

The optimal state sequence, $\hat{\mathbf{q}}$, jointly accounts for the initial contour *shape* and the new set of image observations, $\mathbf{O}_t$, which we map back to image coordinates to estimate the new contour position in the current frame by

$$\tilde{\mathcal{C}}_t \leftarrow I(\hat{\mathbf{q}}, \hat{\mathcal{C}}). \tag{2}$$

Note the distinction between the contour positions and the contour shape. The contour positions correspond to the actual image coordinates. The contour shape refers to the possible deformations of the contour, which is represented by the state sequences of $\lambda$. This is the mechanism that makes the trellis deformable by representing the contour shape probabilistically.

### B. Model Construction

A hidden Markov model is concisely defined by $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$, where $\mathbf{A}$, $\mathbf{B}$ and $\pi$ are the transition, emission and the state prior probabilities, respectively [24]. When the initial contour estimate $\hat{\mathcal{C}}$ is available, we derive the parameters of model $\lambda$ as follows. Intuitively, the transition probabilities represent the local flexibility of the contour, i.e., the change of lateral position along the normals at consecutive $\phi$'s. On the model $\lambda$ this is represented by the probability of transitioning from any state $s_{q_\phi}$ at $\phi$ to any state $s_{q_{\phi+1}}$ at $\phi+1$. The emissions, $p(\mathbf{o}_\phi|s_{q_\phi})$, reflect the probability of observing a particular image feature, $\mathbf{o}_\phi$, when the model is in the state $s_{q_\phi}$. These are shown in Fig. 4(a). The details of estimating the model parameters are provided in Section II-G.

Rather than calculating the probability of observing an image feature conditioned *only* on the current state, Fig. 4(b), we could condition it *also* on the state it transitioned from i.e., between the normals at $\phi$ and $\phi+1$ (See Fig. 5(b)). This way, we condition the observations on *state pairs*, which generalizes single state conditioning. This type of HMM is known in the literature as the *arc-emission* HMM, and will be seen to offer significant benefits in this application. We propose arc-emission HMMs, since they allow conditioning the observations on the *direction* of the

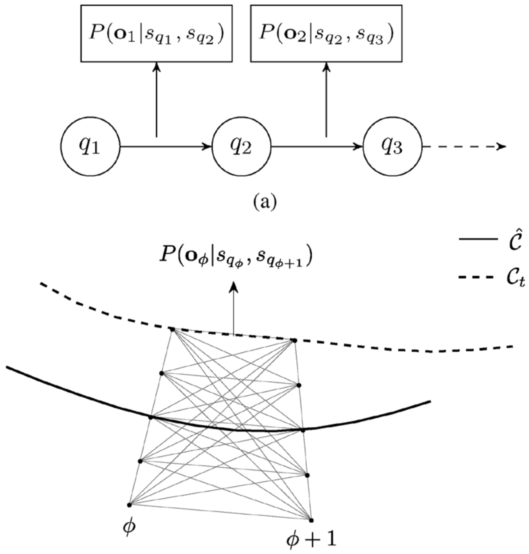



Fig. 5. (a) HMM transitions **A** and arc emissions **B** are shown on the trellis, (b) corresponding observation probabilities.

contour, rather than only location on the contour, Fig. 4(a) and 5(a). While the arc-emission HMM incurs a moderate increase in computational cost, (See Section II-H), it has a significant impact on results as we show in Section III. This is also a major departure from earlier Markovian approaches [21], [22].

### C. Accounting for Length Variations

The contour adjustment method described above assumes that the contour length is fixed. However, the proposed method targets applications where the contour length may be variable. To model the length variations, e.g., shortening or growing of the contour from frame to frame, we use two auxiliary states in the HMM, a *begin state* $s_b$ and an *end state* $s_e$. While the auxiliary states are sufficient for handling the shortening of the contour, the growing case is different as it ventures into areas without prior and requires a further step to extend the trellis in the *a priori* unknown direction of growth. For clarity, we explain the shortening case first and describe the trellis extension (tracing) later.

State $s_b$ is a transient state designed to model shortening at the initial end of the contour, and may be interpreted as "the contour has not yet begun" state. It is only possible to remain in this state or transition out of it, but impossible to later return to it. $s_e$ is an absorbing state that models shortening at the other end of the contour. It is interpreted as "the contour has already ended" state. Once the system transitions to this state it will never transition out. Thus, shortening at the initial end corresponds to late transition out of state $s_b$, and shortening at the other end corresponds to early transition into $s_e$. Since $s_b$ and $s_e$ are transient and absorbing states, respectively, the only allowed transition into $s_b$ is its self transition, and the only allowed transition from $s_e$ is its self transition. Fig. 6 shows the state diagram and Fig. 7 shows the trellis of a simple model. Transition and emission probabilities of $s_b$ and $s_e$ require individual treatment (see Section II-F).

### D. Deformable Trellis Extension to Capture Growth

The proposed model can capture various deformations along the contour as well as shortening. The optimal representation

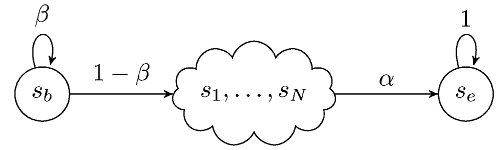



Fig. 6. Auxiliary states $s_b$ and $s_e$ and their transitions.

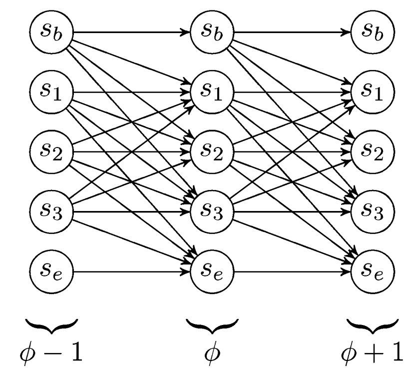



Fig. 7. Trellis of a simple model with $N = 3$.

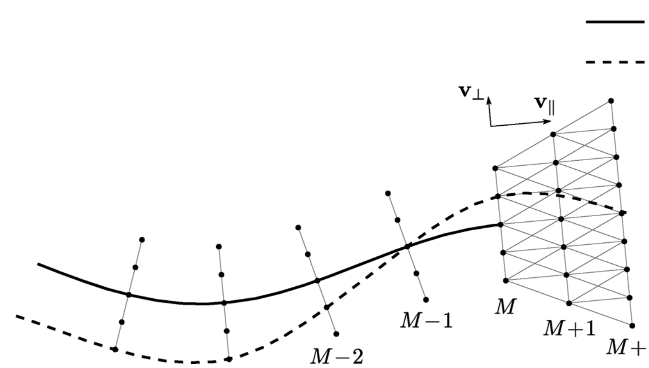



Fig. 8. Construction of extended trellis at $\phi = M$.

will be obtained as long as $\mathcal{C}$ lies in the region that is covered by the deformable trellis. However if $\mathcal{C}$ extends beyond the initial estimate $\hat{\mathcal{C}}$, the prior will lack information about the parts that lie outside the original trellis. In this case, we broaden the trellis by iteratively adding points to both ends. At each iteration, first we construct an *extended trellis* by adding candidate nodes to the ends of the trellis. More precisely, the new nodes are positioned at:

$$\mathbf{x}_{M+\phi,\psi} = \mathbf{x}_{M,(N+1)/2} + \phi \mathbf{v}_{\parallel} + (\psi - \psi')\mathbf{v}_{\perp} \tag{3}$$

where $\psi' = \phi + (N+1)/2$, $\psi = 1, \ldots, N+2\phi$. Furthermore, the vectors associated with the spacing between the nodes along the trellis and the normals are denoted with $\mathbf{v}_{\parallel}$ and $\mathbf{v}_{\perp}$, respectively. A sample extension to the trellis is illustrated in Fig. 8.

The points to be added to the trellis are determined by the most probable path on the *extended trellis*. Including the auxiliary states in the set of candidate states provides a probabilistic way to determine whether to stop or continue the broadening iterations. In this way, we compare the likelihoods of stopping versus continuing the iterations, and stop according to the likelihood test, i.e., as would best explain the observations. Equivalently, we have the auxiliary states appear in the most probable path.

After the broadening phase, we construct the $\tilde{\mathcal{C}}_t$ with the points $\mathbf{x}_{\phi,\hat{q}_\phi}$. The predicted contour $\hat{\mathcal{C}}$ is updated by $\tilde{\mathcal{C}}_t$ so that it will be used as the center of the deformable trellis for the next frame $t+1$.

Here, complex strategies for updating the $\hat{\mathcal{C}}$ can be used (such as Kalman filtering [11]). Depending on the application, even using smoothed version of $\tilde{\mathcal{C}_t}$ may yield useful results.

The overall tracking method which captures contour growth, shortening and deformation is summarized in Algorithm 1.

**Algorithm 1** Open Contour Tracking with Arc-Emission HMM

Require $(\mathbf{A}, \mathbf{B}, \pi)$ of the HMM.
Initialize $\hat{\mathcal{C}}$ on $I_0$ using [25] (or manually).
**for all** $I_t \in \{I_1, \ldots, I_T\}$ **do**
    Construct the trellis around $\hat{\mathcal{C}}$ and extract $\mathbf{O}_t$.
    Initialize $\mathrm{extend}_b = 1$ and $\mathrm{extend}_e = 1$.
    **repeat**
        **if** $\mathrm{extend}_b == 1$ **then**
            Construct extension to trellis at $\phi = 0$.
            Append the features of the extension to $\mathbf{O}_t$.
        **end if**
        **if** $\mathrm{extend}_e == 1$ **then**
            Construct extension to trellis at $\phi = M$.
            Append the features of the extension to $\mathbf{O}_t$.
        **end if**
        Calculate $P(\mathbf{o}_\phi | s_{q_\phi}, s_{q_{\phi+1}})$ $\phi = 1, \ldots, M$.
        Decode $\hat{\mathbf{q}} = \arg\max_{\mathbf{q} \in \mathbf{Q}} P(\mathbf{q}|\mathbf{O}_t, \lambda)$.
        **if** $\hat{q}_1 == b$ **then**
            Set $\mathrm{extend}_b = 0$.
        **end if**
        **if** $\hat{q}_M == e$ **then**
            Set $\mathrm{extend}_e = 0$.
        **end if**
    **until** $\mathrm{extend}_b == 0$ and $\mathrm{extend}_e == 0$
    Update $\hat{\mathcal{C}} = \{\mathbf{x}_{\phi, q_\phi}\}$, $\phi = 1, \ldots, M$.
**end for**

## E. Feature Extraction

The image observation, $\mathbf{O}_t$, is a collection of image features, $\{\mathbf{o}_\phi\}$, extracted on the trellis. The features represent the potential for local ridges in the image. The features are extracted based on the second derivative of the Gaussian filter response at various orientations, as a common practice used in ridge detection. The response of the filter with orientation $\theta$ at point $\mathbf{x}$ is efficiently calculated via $\mathbf{v}(\theta)^T\mathbf{H}(\mathbf{x})\mathbf{v}(\theta)$, where $\mathbf{H}(\mathbf{x})$ is the Hessian evaluated at $\mathbf{x}$, and $\mathbf{v}(\theta) = [\cos\theta \ \sin\theta]^T$, [26].

Feature extraction must be adapted to the HMM-type. Specifically, the state-emission HMM's observations are only related to the current state, while arc emission HMM's observations relate to both the current and the previous state. Thus, an arc emission HMM embeds more information into the evaluation of observations. In this case, two consecutive states through which $\mathcal{C}$ passes provide information on the orientation of $\mathcal{C}$ which is then embedded in the observations. In the case of state-emission HMMs, the orientation must be captured solely by state transitions, compare Fig. 4 and Fig. 5. While the impact due to this difference may not be immediately evident, the results of both emission types confirm the considerable difference in accuracy.

*1) Features for State-Emission HMM:* In state-emission HMM, the observation probabilities depend only on the current state through which $\mathcal{C}$ passes. Thus, the evidence of observing $\mathcal{C}$ on each of the states $s_{q_\phi}$ at $\phi$ is simply evaluated by filter responses at those locations. Recall that on the trellis, the states at a normal $\phi$ are indexed with $\psi$. Thus, the features, $\mathbf{o}_\phi$, computed at the normal at $\phi$ consist of the following filter responses on the states $\psi = 1, 2, \ldots, N$

$$o_{\phi,\psi} = \max_\theta \mathbf{v}(\theta)^T \mathbf{H}(\mathbf{x}_{\phi,\psi}) \mathbf{v}(\theta) \tag{4}$$

which corresponds to the largest eigenvalue of $\mathbf{H}(\mathbf{x}_{\phi,\psi})$ [27].

*2) Features for Arc-Emission HMM:* In the case of arc-emission HMM, observation probabilities are calculated *between* consecutive normals at $\phi$ and $\phi + 1$. Thus, the features are calculated along the line connecting the locations associated with the states $s_{q_\phi}$ and $s_{q_{\phi+1}}$ of the deformable trellis. This setting enables us to quantify the evidence for $\mathcal{C}$ passing through the consecutive states of interest.

The features, $\mathbf{o}_\phi$, extracted between normals at $\phi$ and $\phi + 1$, consist of the following filter responses between the states $q_\phi, q_{\phi+1} = 1, 2, \ldots, N$

$$o_{\phi,q_\phi,q_{\phi+1}} = \int_0^1 \mathbf{v}(\theta)^T \mathbf{H}\left((1-\tau)\mathbf{x}_{\phi,q_\phi} + \tau \mathbf{x}_{\phi+1,q_{\phi+1}}\right) \mathbf{v}(\theta) d\tau \tag{5}$$

where, $\theta$ is the angle normal to the line between $\mathbf{x}_{\phi,\psi}$ and $\mathbf{x}_{\phi+1,\psi'}$. Note that the term $\mathbf{H}(\cdot)$ embeds the orientation into the feature set $\mathbf{o}_\phi$.

## F. Observation Probabilities

We represent the probability density function (PDF) of the features extracted on the foreground (contour) and on the background with $P(x|FG)$ and $P(x|BG)$, respectively. Accordingly, if we consider the state-emission HMM image feature $o_{\phi,\psi}$ extracted on the point $\mathbf{x}_{\phi,\psi}$, the $P(o_{\phi,\psi}|FG)$ and $P(o_{\phi,\psi}|BG)$, represent the likelihood that $\mathbf{x}_{\phi,\psi}$ belongs to the contour and the background, respectively. The likelihoods associated with the arc-emission HMM features $o_{\phi,\psi,\psi'}$ can be defined similarly.

*1) State-Emission Probabilities:* Recall that the set of image features $\mathbf{o}_\phi = \{o_{\phi,\psi}\}$ consists of the features extracted from the points, $\{\mathbf{x}_{\phi,\psi}\}$, that lie on the normal at $\phi$ with $\psi = 1, 2, \ldots, N$. We assume that the elements of $\mathbf{o}_\phi = \{o_{\phi,1}, \ldots, o_{\phi,N}\}$ are conditionally independent of each other given $s_{q_\phi}$ where $q_\phi$ represents the actual state at normal $\phi$:

$$P\left(\mathbf{o}_\phi | s_{q_\phi}\right) \propto \prod_\psi P\left(o_{\phi,\psi} | s_{q_\phi}\right). \tag{6}$$

Intuitively, this assumption states that the image features extracted on the points of a particular normal are independent of each other given the point that contour passes through. Accordingly, we model the observation probabilities $P(o_{\phi,\psi'}|s_\psi)$ as follows:

$$P\left(o_{\phi,\psi}|s_{q_\phi}\right) = \begin{cases} P(o_{\phi,\psi}|FG) & \text{if } \psi = q_\phi \\ P(o_{\phi,\psi}|BG) & \text{else.} \end{cases} \quad (7)$$

The probability of observing $\mathbf{o}_\phi$, given that the system in state $s_{q_\phi}$ can be written as:

$$P\left(\mathbf{o}_\phi|s_{q_\phi}\right) \propto P(o_{\phi,\psi}|FG) \prod_{\psi \neq q_\phi} P(o_{\phi,\psi}|BG). \quad (8)$$

On image pixels, this is equivalent to the probability of observing $\mathcal{C}$ only on the point $\mathbf{x}_{\phi,q_\phi}$, conditioned on the state $s_{q_\phi}$ where $q_\phi = 1, \ldots, N$. Similarly, the observation probability conditioned on an auxiliary state, $s_b$ or $s_e$, is the probability of *not* observing $\mathcal{C}$ on *any* point on the normal at $\phi$.

$$P\left(\mathbf{o}_\phi|s_{q_\phi}\right) \propto \prod_{\psi} P(o_{\phi,\psi}|BG), \quad q_\phi = b, e \quad (9)$$

*2) Arc-Emission Probabilities:* The probability of observing a set of image features $\mathbf{o}_\phi$ extracted between the normals at $\phi$ and $\phi+1$ given that we are moving from state $s_{q_\phi}$ (at $\phi$) to state $s_{q_{\phi+1}}$ (at $\phi+1$) is represented with $P(\mathbf{o}_\phi|s_{q_\phi}, s_{q_{\phi+1}})$. Observation probabilities are set to 0 if the associated state pair transition is not allowed on the transition matrix. Similar to the state-emission case, when both $q_\phi$ and $q_{\phi+1}$ are non-auxiliary states, the observation probability conditioned on the state pairs, $s_{q_\phi}$ and $s_{q_{\phi+1}}$, is associated with the probability that the image feature $o_{\phi,q_\phi;q_{\phi+1}}$ belongs to foreground and the rest of the image features $\{\mathbf{o}_\phi \backslash o_{\phi,q_\phi,q_{\phi+1}}\}$ belong to the background:

$$P\left(\mathbf{o}_\phi|s_{q_\phi}, s_{q_{\phi+1}}\right) \propto P\left(o_{\phi,q_\phi,q_{\phi+1}}|FG\right) \times \prod_{\psi,\psi' \neq q_\phi, q_{\phi+1}} P(o_{\phi,\psi,\psi'}|BG) \quad (10)$$

Observation probability associated with remaining in an auxiliary state at $\phi$ is clearly the probability of not observing $\mathcal{C}$ at these locations. In this case, it is given by the product of probability of not observing the contour, $P(o_{\phi,q_\phi,q_{\phi+1}}|BG)$, through all possible point pairs $(\mathbf{x}_{\phi,q_\phi}, \mathbf{x}_{\phi,q_{\phi+1}})$:

$$P\left(\mathbf{o}_\phi|s_{q_\phi}, s_{q_{\phi+1}}\right) \propto \prod_{\psi,\psi'} P(o_{\phi,\psi,\psi'}|BG). \quad (11)$$

The above probability also applies for transitioning from an auxiliary state to a non-auxiliary state and vice versa. Thus, the arc emission probabilities are evaluated conditionally as follows:

$$P(\cdot) \propto \begin{cases} P\big(o_{\phi,q_\phi,q_{\phi+1}}|FG\big) \times \prod\limits_{\psi,\psi' \neq q_\phi,q_{\phi+1}} P(o_{\phi,\psi,\psi'}|BG) & q_\phi, q_{\phi+1} = 1, \ldots, N \\ \prod\limits_{\psi,\psi'} P(o_{\phi,\psi,\psi'}|BG) & \text{else} \end{cases} \quad (12)$$

### G. Learning Model Parameters

So far, we discussed how image features are related with the emission probabilities of the deformable trellis. Note that this relation is still in terms of the PDFs, $P(x|FG)$ and $P(x|BG)$. These distributions, as well as the transition and prior probabilities of the HMM should be estimated from the training data.

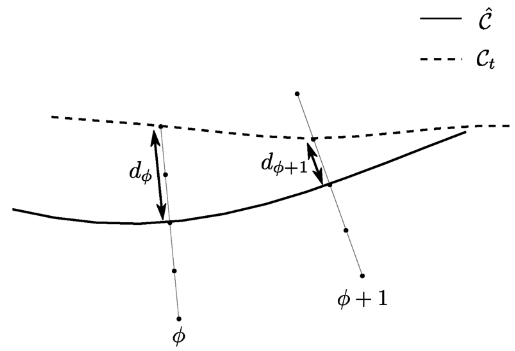



Fig. 9. Manually segmented contour, $\mathcal{C}_t$, superimposed on the predicted contour, $\hat{\mathcal{C}}$. The distance between $\mathcal{C}_t$ and $\hat{\mathcal{C}}$ on the normal at $\phi$ is represented by $d_\phi$.

In this section, we describe a method for learning the parameters of the HMM, $(\mathbf{A}, \mathbf{B}, \pi)$, based on manually segmented open-contours over time. Dynamic and structural statistics of the examples are considered to estimate the transition matrix, $\mathbf{A}$, observation probabilities, $\mathbf{B}$, and the initial state distribution, $\pi$. Intuitively, estimating the model parameters can be thought of as collecting statistics about how contour points effect the movements of neighboring contour points. While theoretically higher order HMMs could be used for a complete representation of this dependence, the resulting computational cost may not be justified in terms of performance gains. Here, we describe the principle using first-order models.

In this work, the components of the HMM are modeled based on Gaussian PDFs. We denote a Gaussian PDF with mean $\mu$ and variance $\sigma^2$ by $\mathcal{N}(\mu, \sigma^2)$. Furthermore, $\mathcal{N}(x; \mu, \sigma^2)$ represents the value of the $\mathcal{N}(\mu, \sigma^2)$ evaluated at $x$.

In Fig. 9 we illustrate a portion of manually segmented contour, $\mathcal{C}_t$, superimposed on the predicted contour, $\hat{\mathcal{C}}$, where the prediction is based on the past manual segmentations. Here, $d_\phi$ is the amount of lateral displacement with respect to the prediction. The amount of bending on normal at $\phi + 1$ with respect to position on the previous normal at $\phi$ is $d_{\phi+1} - d_\phi$. We model the distributions of the $d_\phi$ and $d_{\phi+1} - d_\phi$ for all $\phi$ with two Gaussian PDFs $\mathcal{N}(\mu_l, \sigma_l^2)$ and $\mathcal{N}(\mu_k, \sigma_k^2)$, respectively. The parameters of these distributions are estimated based on the mean and the variance of $d_\phi$'s and $d_{\phi+1} - d_\phi$'s for all $\phi$ on all training sequences.

*1) Initial State Distribution ($\pi$):* We model the distribution of $d_\phi$ for $\phi = 1$ with a Gaussian PDF $\mathcal{N}(\mu_\pi, \sigma_\pi^2)$. The parameters of $\mathcal{N}(\mu_\pi, \sigma_\pi^2)$ are estimated based on the mean and the variance of $d_\phi$ for $\phi = 1$ on all training sequences. Accordingly, the initial state probabilities for non-auxiliary states are modeled with a Gibbs distribution as follows:

$$\pi_\psi \propto \mathcal{N}\left(d_\perp\left(\psi - \frac{N+1}{2}\right); \mu_\pi, \sigma_\pi^2\right) \quad \psi = 1, \ldots, N \quad (13)$$

Here, $d_\perp$ is the spacing between points along the normal. The prior probability $\pi_b$ is set as the probability of not observing $\mathcal{C}$ on the normal at $\phi = 1$. Since we do not expect to start with the end state, $s_e$, we set $\pi_e = 0$.

*2) Observation Distributions ($\mathbf{B}$):* As described in Section II-F, the observation probabilities are modeled based on two PDFs: $P(x|FG)$ and $P(x|BG)$. These distributions are modeled using two Gaussian PDFs, $\mathcal{N}(\mu_{\text{fg}}, \sigma_{\text{fg}}^2)$ and $\mathcal{N}(\mu_{\text{bg}}, \sigma_{\text{bg}}^2)$, respectively. The parameters of these Gaussian PDFs are estimated by the mean and the variance of image features computed from points on contour and on background, respectively. Although these features can be extracted on the training sequences, one can also extract them on the initial frame $I_0$ using $\hat{\mathcal{C}}$ (See Algorithm 1), thereby making the estimated parameters more robust to illumination variations across different contours.

*3) Transition Matrix (*$\mathbf{A}$*):* Each element of the transition matrix, $a_{\psi,\psi'}$, is the probability of moving to state $s_\psi$ given that the system was at state $s_{\psi'}$. Graphical representation of the state transitions is illustrated in Fig. 6. We represent the self transition probability of state $s_b$ with $\beta$. Consequently, the probability of moving from $s_b$ to any one of states $s_\psi$, $\psi = 1,\ldots,N$, is $(1-\beta)/N$ assuming that the each transition is equally likely. The self transition probability of end state $s_e$ is 1 and the probability of moving into $s_e$ is $\alpha$. To cover all possible transitions, we consider a fully connected Markov chain for the non-auxiliary states $s_\phi$, $\psi = 1,\ldots,N$ with the transition matrix $\mathbf{T}$. Here, we focus on estimating $\mathbf{T}$ and discuss the probabilities associated with auxiliary state transitions later.

A straightforward approach to estimate $\mathbf{T}$ would be counting the occurrences of the state transitions on the training data. In the literature, this technique is known as Viterbi learning and provides a computationally efficient way to learn the transition matrix. However, it requires large amount of training data for reliable estimation of $N^2$ parameters. Moreover, associating $\mathcal{C}_t$ with a state along each normal results in distortion due to the finite precision sampling of states along normals. Considering these drawbacks, we propose a method for robust estimation of $\mathbf{T}$ using a parametric representation. In this way, we reduce the number of parameters to estimate and make use of the statistics of $d_\phi$ and $d_{\phi+1}-d_\phi$ which is calculated with sub-pixel accuracy. In the proposed method, we first represent the steady state distribution of the system with a set of parameters and learn them from the training data. Then we relate the model parameters to the steady state distribution and adjust to parameters such that the desired steady state distribution is reached.

In the parametric representation, the steady state distribution is modeled with a Gibbs distribution

$$P(s_\psi) \propto \mathcal{N}\left(d_\perp\left(\psi - \frac{N+1}{2}\right); \mu_l, \sigma_l^2\right) \quad \psi = 1,\ldots,N \tag{14}$$

Note that $\mu_l$ and $\sigma_l^2$ are the mean and the variance of $d_\phi$, respectively. Similarly, each row of $\mathbf{T}$ is also modeled with a Gibbs distribution. In literature, this type of distribution is known as the Gibbs distribution. The mean at each row, $\psi$, represents the most likely state that follows the $s_\psi$, whereas the variance represents the likelihood of transitioning to other states. Estimation of separate parameters for each row is still not desirable since it requires large amount of training data. Therefore, we propose to use the mean and the variance of $d_{\phi+1}-d_\phi$ (i.e., $(\mu_k, \sigma_k^2)$), as the parameters of the Gibbs distributions for each row. We then further allow some deviation across rows through \emph{tilting} coefficients, $\mathbf{w} = [w_\psi]$

$$\mathbf{T} = [t_{\psi',\psi}] = P(s_\psi|s_{\psi'}) \propto w_\psi \mathcal{N}\left(d_\perp(\psi-\psi'); \mu_k, \sigma_k^2\right). \tag{15}$$

In literature, this is known as the tilted-Gibbs distribution. Note that small $\sigma_k^2$ indicates the tendency for self transitions. Oppositely, frequent state changes makes $\sigma_k^2$ large. Now the task is to estimate the tilting coefficients based on the desired steady-state distribution, representing the occupancy rate of the states over infinitely many transitions

$$\begin{aligned} P(s_\psi) &= \sum_{\psi'=1}^{N} P(s_\psi|s_{\psi'})P(s_{\psi'}) \quad \psi = 1,\ldots,N \\ &= w_\psi \sum_{\psi'=1}^{N} f_{\psi'}(\mathbf{w})\mathcal{N}\big(d_\perp(\psi-\psi'); \mu_k, \sigma_k^2\big)P(s_{\psi'}) \end{aligned} \tag{16}$$

Unfortunately, the above equation that relates $\mathbf{w}$ to the steady state distribution is non-linear and no analytical solution exists. However, we can re-write the (16) so that $\mathbf{w}$ becomes fixed point of the following function. This formulation enables us to use fixed point iterations to obtain the optimal $\mathbf{w}$ so that the desired steady state distribution is reached

$$\begin{aligned} f_\psi(\mathbf{w}) &= \frac{P(s_\psi)}{\sum_{\psi'=1}^{N} u_{\psi'}(\mathbf{w})\mathcal{N}\left(d_\perp(\psi-\psi'); \mu_k, \sigma_k^2\right) P(s_{\psi'})} \\ &= w_\psi. \end{aligned} \tag{17}$$

Scaling $\mathbf{w}$ with a positive real number will not change the steady state distribution. Therefore, the optimum $\mathbf{w}$ should be selected on the simplex $\Delta^{N-1}$ (i.e., $\sum_{\psi=1}^{N} w_\psi = 1$).

**Algorithm 2** Learning the Parameters of Arc-Emission HMM

**for all** Training Sequences **do**

Set $\hat{\mathcal{C}}$ as the prediction based on $\mathcal{C}_{t-1}$.

Construct the trellis around $\hat{\mathcal{C}}$.

Calculate $d_\phi$ for all $\phi$'s (See Fig. 9).

Extract the image features for the arcs that are on the contour and on the background.

**end for**

Estimate $(\mu_l, \sigma_l^2)$ using all $d_\phi$'s for all training sequences and for all $\phi$'s.

Estimate $(\mu_k, \sigma_k^2)$ using all $d_{\phi+1} - d_\phi$'s for all training sequences and for all $\phi$'s.

Estimate $(\mu_\pi, \sigma_\pi^2)$ using all $d_1$'s for all training sequences.

Estimate $(\mu_{\mathrm{fg}}, \sigma_{\mathrm{fg}}^2)$ using the image features for the arcs that are on the contour.

Estimate $(\mu_{\mathrm{bg}}, \sigma_{\mathrm{bg}}^2)$ using the image features for the arcs that are on the background.

Initialize $\mathbf{w}' = \{w'_\psi\}$ with random numbers $\in [0,1]$.

Project $w_\psi = w'_\psi/\|\mathbf{w}'\|_1$, $\psi = 1,\ldots,N$.

**repeat**

Set $w'_\psi = f_\psi(\mathbf{w})$, $\psi = 1,\ldots,N$. (Note that $\mathcal{N}(\mu_l, \sigma_l^2)$ and $\mathcal{N}(\mu_k, \sigma_k^2)$ terms are used while calculating $f_\psi(\mathbf{w})$. See (14) and (17) for details.)

Project $w_\psi = w'_\psi/\|\mathbf{w}'\|_1$, $\psi = 1,\ldots,N$.

**until** Convergence

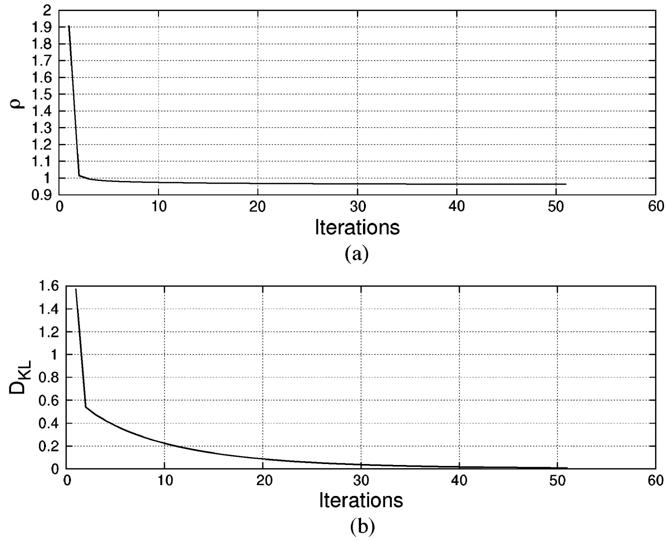



Fig. 10. Convergence plots of the fixed point iterations. (a) Spectral radius of $\mathbf{F}(\mathbf{w}^*)$. (b) Kullback-Leibler divergence [29] between the desired steady state distribution $P(s_\psi)$ and the steady state distribution of $\mathbf{T}$ constructed with $\mathbf{w}^*$.

The necessary condition for convergence of fixed point iterations is $\rho(\mathbf{F}(\mathbf{w}^*)) < 1$ [28]. Here, $\mathbf{F}(\mathbf{w}^*)$ is the Jacobian matrix of $\mathbf{f}(\mathbf{w})$ evaluated at the fixed point, $\mathbf{w}^*$, and $\rho(\mathbf{F}(\mathbf{w}^*))$ is the spectral radius of $\mathbf{F}(\mathbf{w}^*)$. The convergence plots of the fixed point iterations are illustrated in Fig. 10. Based on $\mathbf{T}$, we can write the transition probabilities in matrix form, $\mathbf{A}$, as follows:

$$\mathbf{A} = \begin{bmatrix} (1-\alpha)\mathbf{T}_{N\times N} & \mathbf{0}_{N\times 1} & \alpha\mathbf{1}_{N\times 1} \\ (1-\beta)N^{-1}\mathbf{1}_{1\times N} & \beta & 0 \\ \mathbf{0}_{1\times N} & 0 & 1 \end{bmatrix} \begin{matrix} \}s_1,\ldots,s_N \\ \}s_b \\ \}s_e \end{matrix} . \tag{18}$$

4) *Probabilities $\alpha$ and $\beta$:* The transition probabilities from states $s_\psi, \psi = 1,\ldots,N$, to $s_e$ is assumed to be same and represented with $\alpha$. Similarly, $\beta$ is the self transition probability of $s_b$. These transition probabilities play an important role in the decoding stage even though they are combined with the image observations probabilistically. Depending on the observations, large $\alpha$ might force the optimal path to reach $s_e$ too early. Similarly, optimal path might stay in $s_b$ too long with large $\beta$. To avoid these undesired consequences, we propose to adjust $\alpha$ and $\beta$ based on the statistics of motion and the length of the $\mathcal{C}$.

Given that we start at $s_b$, the expected number of self transitions before we leave $s_b$ is $(1-\beta)^{-1}$ [30]. Accordingly, $(1-\beta)^{-1}$ is set as the average number of normals where $\mathcal{C}$ is not observed (at $\phi = 1$ end). Similarly, starting from states $s_\psi, \psi = 1,\ldots,N$, the expected number of transitions before reaching $s_e$ is $\alpha^{-1}$ and on every frame, $\alpha^{-1}$ is set as the number of normals on $\hat{\mathcal{C}}$.

### H. Decoding Complexity

In the decoding process, we are interested in finding the path, $\mathbf{q} \in \mathbf{Q}$ that maximizes the posterior $P(\mathbf{q}|\mathbf{O}_t,\lambda)$ (1). In our setting, the number of elements in the set $\mathbf{Q}$ is $(N+2)^M$ and grows exponentially with the number of normals, $M$. However, the optimal path that maximizes $P(\mathbf{q}|\mathbf{O}_t,\lambda)$, can be efficiently found with linear complexity in $M$ using Viterbi algorithm [24]. In the Viterbi algorithm, $P(\mathbf{q}|\mathbf{O}_t,\lambda)$ can be re-written as follows:

$$P(\mathbf{q}|\mathbf{O}_t,\lambda) = \frac{P(\mathbf{O}_t|\mathbf{q},\lambda)P(\mathbf{q}|\lambda)}{P(\mathbf{O}_t|\lambda)}. \tag{19}$$

We can further decompose the probabilities $P(\mathbf{O}_t|\mathbf{q},\lambda)$ and $P(\mathbf{q}|\lambda)$ as product of observation and transition probabilities

$$P(\mathbf{q}|\lambda) = P(s_{q_1}|\lambda)\prod_{\phi=1}^{M-1} P\left(s_{q_{\phi+1}}|s_{q_\phi},\lambda\right). \tag{20}$$

Observation probabilities for state-emission HMM case

$$P(\mathbf{O}_t|\mathbf{q},\lambda) = \prod_{\phi=1}^{M} P\left(\mathbf{o}_\phi|s_{q_\phi},\lambda\right). \tag{21}$$

Observation probabilities for arc-emission HMM case

$$P(\mathbf{O}_t|\mathbf{q},\lambda) = \prod_{\phi=1}^{M-1} P\left(\mathbf{o}_\phi|s_{q_{\phi+1}},s_{q_\phi},\lambda\right). \tag{22}$$

The complexity of the Viterbi algorithm in both state-emission and arc-emission HMM case is $O(MN^2)$. The complexity of feature extraction for state-emission HMM is $O(MN)$, whereas for arc-emission HMM it is $O(MN^2)$.

## III. Experimental Results

In this section, we demonstrate the performance of the proposed tracking algorithm on synthetic and biological datasets that exhibit severe noise, clutter and illumination variations. We show the effectiveness of the HMM-based algorithm in comparison to existing open-contour tracking methods.

### A. Open-Contour Adjustment on the Synthetic Dataset

In this section, we illustrate a critical advantage of arc-emission HMM over the traditional state-emission HMM as well as the active contour models in four scenarios. In the first scenario, the open-contour is adjusted on an image with a single sinusoidal open-contour. Note that in practice, one should expect similar objects cluttering the vicinity and where the "ackground" no longer can be modeled by a distinct background noise model. This is addressed in the other scenarios by including an interfering open-contour in the vicinity. We illustrate the output of different algorithms in Fig. 11.

Active contours provide a systematic way to adjust an initial contour such that its energy is minimized. The energy minimization is performed by gradient descent and the energy is defined based on the local ridge strength and contour smoothness for the open-contour adjustment [3], [31]. Low signal to noise ratio makes the algorithm prone to be trapped in local minima, which is the main drawback of the active contours. As illustrated on the first column of Fig. 11, active contour attaches to either a noisy region or a nearby contour. However, HMM-based contour adjustment uses Viterbi algorithm [24] which provides the globally optimal solution within the trellis.

Fast Marching methods [1] efficiently compute the distances on weighted domains with the initial conditions on the starting points. The distance map can be further used to find the shortest paths from any point on the domain to the closest starting point. The Fast Marching method can be adopted to solve the

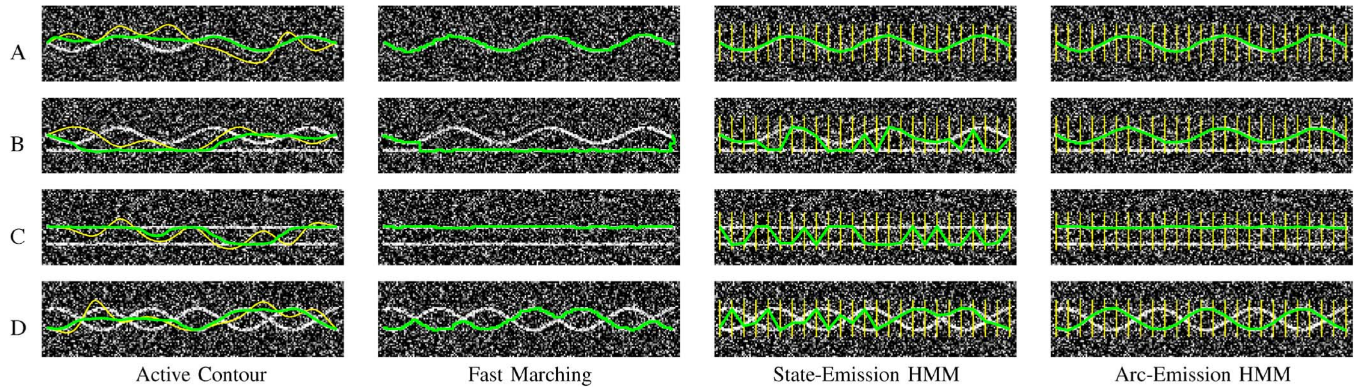



Fig. 11. Open-Contour adjustment results on synthetic dataset with four scenarios. Green lines on the columns illustrate the results associated with active contour (with ridge features), fast-marching, state-emission HMM and arc-emission HMM respectively. Yellow lines are the initialization of the active contour. Yellow dots represent the positions of the states in the state-emission and arc-emission HMM.

contour adjustment problem by using the intensity information to construct the weighted domain. Then the Fast Marching algorithm can be used to extract the optimal path between the beginning and ending points. The performance of this method for open-contour adjustment is illustrated on the second column of Fig. 11. The main disadvantage is that interfering curvilinear structures in the vicinity makes the fast marching algorithm favor undesired "shortcuts" as observed on the second and fourth scenarios from the top. In these cases, shortest path jumps to the straight line since it results in shorter path and there is nothing to prevent such behavior.

The last two columns of Fig. 11 illustrate the results of state-emission and arc-emission HMM-based open-contour adjustment. Arc-emission HMM does not show a significant improvement over state-emission HMM in the first scenario without an interfering curvilinear structure. However, on the other scenarios, states that are located on either contour will show significant evidence for a possible passage of the contour. If the emission probabilities are only conditioned on the state, there isn't sufficient penalty for jumping from one contour to the other, which the traditional HMM will do depending on which contour's local evidence is somewhat more pronounced at each point. (The only penalty for such skipping is provided by the state transition probabilities, but "tightening" those to solve this problem will severely compromise the flexibility of the trellis and its ability to model deformations). Arc-emission HMM, on the other hand, allows evaluation of the local evidence for consistency with the *direction of the transition* and it can easily eliminate such confusion. This is demonstrated on the last column of Fig. 11.

Quantitative performance measures are obtained by running the algorithms 100 times on the images corrupted with additive white Gaussian noise with zero mean and 0.25 variance. The average distances between the output and the ground truth are computed along the normals. The root mean squared distances are provided in Table I.

TABLE I
PERFORMANCE OF THE OPEN-CONTOUR ADJUSTMENT ALGORITHMS ON THE SYNTHETIC IMAGES

| Method | Scenario | | | |
|---|---|---|---|---|
| | A | B | C | D |
| Arc-Emission | **0.70** | **0.68** | **0.65** | **4.34** |
| State-Emission | 0.83 | 7.29 | 7.30 | 6.48 |
| Active Contour | 2.90 | 5.54 | 6.18 | 6.15 |
| Fast Marching | 0.89 | 7.77 | 2.12 | 6.64 |

## *B. Tracking of Live Cell Microtubules*

In this scenario, we consider tracking of microtubules on live cell florescence images. The dataset consists of 8 image sequences from [32] with approximately 40 frames per sequence. These images are extremely noisy and close proximity of similar microtubules make it difficult even for manual tracking. To represent this uncertainty on the ground truth, 4 experts manually tracked the same 30 microtubules by approximating their bodies with polylines in each frame.

In these image sequences, microtubules grow and shorten on the outer periphery of the cell. While moving towards nucleus, microtubule concentration increases and makes the center region a big, typically overexposed, blob (See Fig. 1). Therefore, in this application, we are interested in tracking the body of the microtubule extending from a fixed end-point close to the nucleus. These fixed points can be detected automatically during the initialization of the tracking, which is beyond the scope of this work. Instead, we let the user define the object of interest by these fixed points and use the tracing algorithm described in [25] to extract the body on the initial frame.

We measure the accuracy of the tracking by the distance between the output and the ground truth open-contours. We define the distance between two open-contours, $\mathcal{C}_1$ and $\mathcal{C}_2$, based on the distances between the points on $\mathcal{C}_1$ and the line segments on $\mathcal{C}_2$. Specifically, for each point on $\mathcal{C}_1$ we find a line segment on $\mathcal{C}_2$ such that the summation of the distances between these points and corresponding line segments is minimized. We consider the following constraints on matching a point to a line segment. First and last point of $\mathcal{C}_1$ is matched to the first and last point of $\mathcal{C}_2$. A point *can not* be matched to a line segment *before* the previous point's match, in other words the matching is ordered. This constrained matching problem can be effectively solved using dynamic programming to obtain *body distance* which is the optimal matching distance divided by the number of points

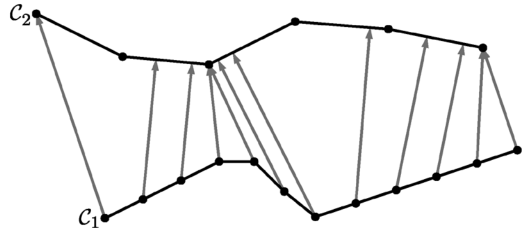



Fig. 12. Calculation of distance between two open contours.

TABLE II
PERFORMANCE OF THE OPEN-CONTOUR TRACKING ALGORITHMS ON THE LIVE CELL MICROTUBULE IMAGE SEQUENCES

| Method | Body Distance |
|---|---|
| Arc-Emission with Tilted-Gibbs Trans. | **4.94** |
| Arc-Emission with Gibbs Trans. | 6.28 |
| Arc-Emission with Viterbi Learning Trans. | 5.25 |
| State-Emission with Tilted-Gibbs Trans. | 9.16 |
| State-Emission with Gibbs Trans. | 13.04 |
| State-Emission with Viterbi Learning Trans. | 8.77 |
| Discrete Dynamic Contour Model | 9.34 |
| Tip Matching with Fast Marching | 11.78 |

on $\mathcal{C}_1$. An example matching problem and the corresponding solution are depicted in Fig. 12.

The body distance results on the living cell microtubule tracking application is given in Table II. The arc-emission HMM outperforms state-emission HMM with the same transition model since the image observations between the state pairs make the system more robust to interfering curvilinear structures. This is also supported by the results obtained in the synthetic data-set (See Section III-A). In the same table, we further compare the accuracy of the HMM-based approaches with three representations for the transition matrix. In the Gibbs and tilted-Gibbs representations, the transition probabilities are modeled with a Gibbs distribution

$$\mathbf{T} = [t_{\psi',\psi}] = P(s_\psi | s_{\psi'}) \propto \mathcal{N}\left(d_\perp(\psi - \psi'); \mu_k, \sigma_k^2\right) \quad (23)$$

and a tilted-Gibbs distribution (15), respectively. Finally, in the Viterbi learning-based representation, the transition probabilities are estimated by counting the occurrences of the state transitions on the training data. As shown in Table II, the best results are obtained when the transition matrix of the arc-emission HMM is modeled with the tilted-Gibbs distribution. The tilted-Gibbs representation improves the performance over the plain Gibbs since it can effectively model asymmetric transitions with reasonable model complexity. However, employing a more complex model would not necessarily yield better accuracy over a simpler model since the former would require a larger amount of training data than the latter for reliable parameter estimation. In fact, in our experiments, for the arc-emission HMM, a more complex model learned by the Viterbi learning approach yields less accurate results when compared to a simpler tilted-Gibbs distribution based model. This is primarily due to the limited amount of training data for estimating $N^2$ parameters of the transition matrix.

The performance of the HMM-based open-contour tracking methods are also compared with two earlier approaches based on tip point matching [3] and discrete dynamic contour model [33]. To the best of our knowledge, these are the closest approaches which address the variable length open-contour deformations. We briefly summarize these two methods and discuss the advantages of the HMM-based tracking over them.

The tip point matching method makes use of a graph where the nodes are associated with the pre-detected candidate tip points. The edge weights in the graph are set considering the minimum geodesic distance on both consecutive frames. The track associated with a user-specified tip point is determined by the shortest path on the graph starting from that point. The performance of the tip point matching algorithm is relatively poor when compared to the HMM-based tracking method since the morphological tip detection is very sensitive to noise. Moreover, the track can easily jump to a nearby curvilinear structure in the presence of clutter. However, the proposed scheme simultaneously adjusts the tip and body position hence yields better results. Sample results on live cell microtubule tracking are illustrated in Fig. 13 and Fig. 15. Additional tracking videos are available at [34].

In the discrete dynamic contour model (DDCM), the points on the curvilinear structure are iteratively moved by the external and internal forces to minimize a cost function based on the contour curvature and image features. This method performs relatively better than the tip matching method since it evolves whole body during the tracking. However, the evolution is performed using gradient descent which can get trapped in local minima. On the other hand, HMM-based methods provide the globally optimum solution within the trellis and yield better performance. Moreover, in DDCM, the points are added and removed by evaluating simple conditions based on image observations. Whereas HMM-based methods perform hypothesis testing over the entire trellis to make decisions about addition or removal of points.

The HMM-based tracking algorithm is implemented in `C++`. Experiments are performed on a standard computer with 2.33 GHz CPU and 2 GB of RAM. The number of states are set as 7 with 1 pixel spacing along the normals. The number of nodes are selected such that the normals are separated by 2 pixels. The running times to track a single microtubule on 40 frames with frame size 560$\times$ 452 using state-emission and arc-emission HMMs are 15.80 and 17.55 seconds, respectively. The DDCM and the tip matching methods are implemented in `Matlab`. The running times to track a single microtubule using the DDCM and tip matching methods are 20.97 and 24.23 seconds, respectively.

### C. Tracking of Gliding Microtubules

The effectiveness of the proposed algorithm is also demonstrated on tracking of gliding microtubules. In this case, both of the starting and ending points are actively moving so that the microtubules are gliding on the surface. The same tubes are also tracked using the discrete dynamic contour model [33] and the results are presented in Fig. 14. The tracking videos are available at [34].

It is clear that DDCM is sensitive to local illumination variations and performs poorly in the intersections. Joint adjust-

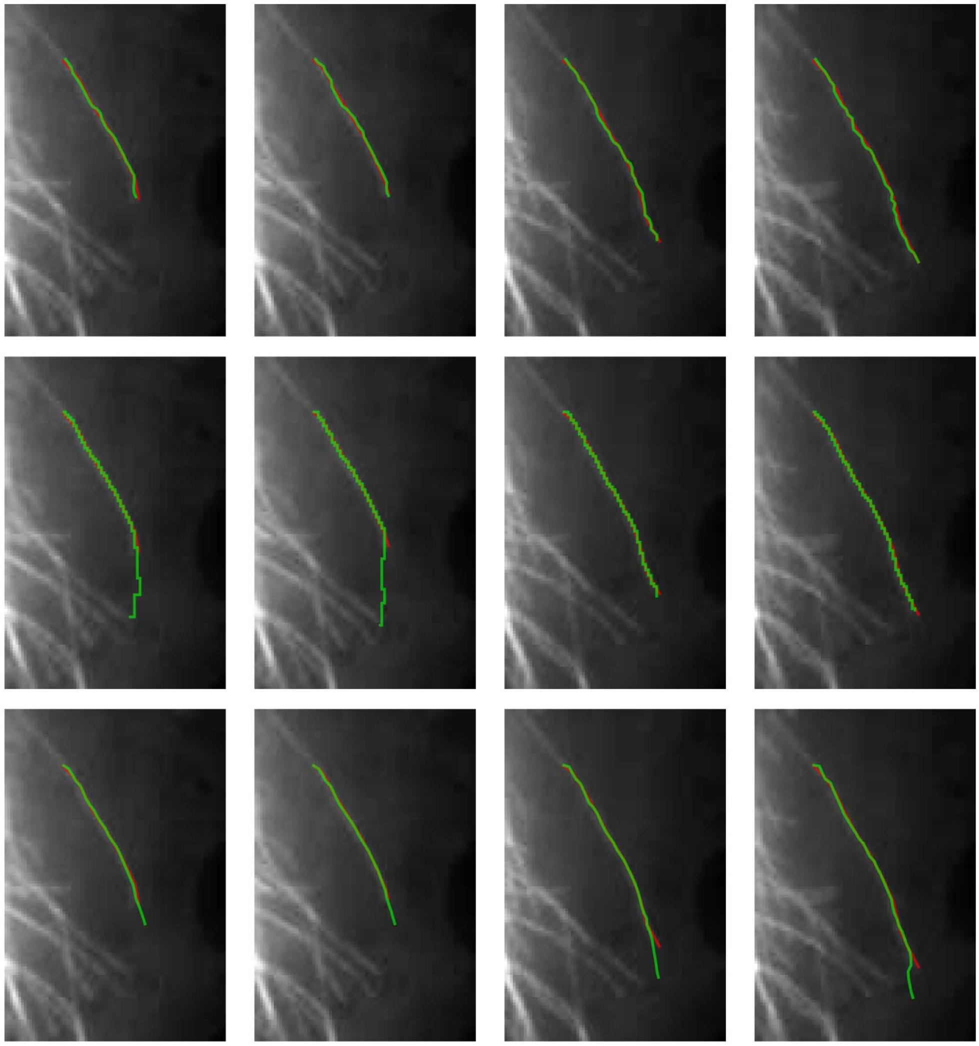


Fig. 13. Output (green) of arc-emission HMM (first row), tip matching (second row) and DDCM (third row) together with the ground truth (red).

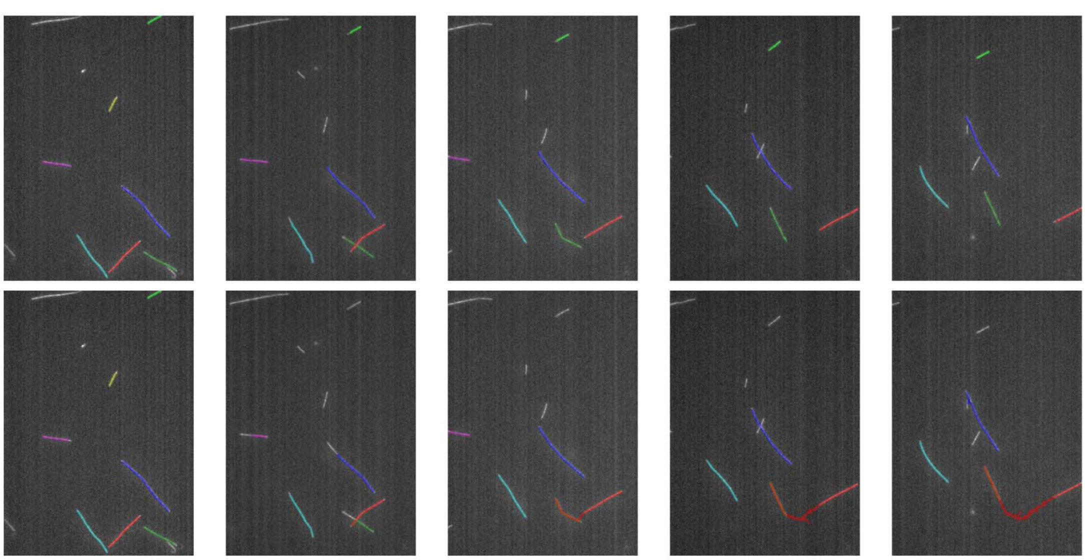


Fig. 14. Output of the proposed algorithm (first row) and the discrete dynamic contour model (second row) on tracking multiple gliding microtubules.

ment for body deformations and tip positions makes the proposed approach robust. In this case, the tip matching method does not apply as both ends of microtubules need to be tracked simultaneously.

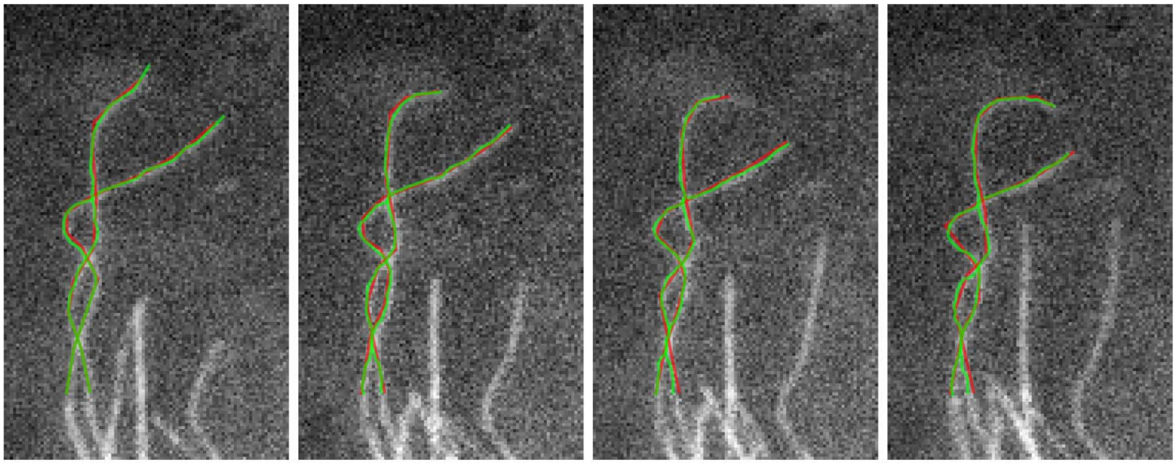


Fig. 15. Output of the proposed method (green) on multiple intersections with relatively small intersection angles. The ground truth is shown in red.

## IV. Conclusion

This paper presents an algorithm for variable length open-contour tracking. There are three main contributions. A framework based on arc-emission HMM is proposed for open-contour tracking, and is shown to offer substantial benefits when properly optimized. The arc-emission HMM framework is complemented with auxiliary states so as to simultaneously capture lateral deformations and time varying lengths of open contours. As there are various biological structures that are curvilinear and dynamic in nature, automated quantification methods can benefit considerably from the proposed approach. Finally, we introduce a learning method for parameter estimation in arc-emission HMMs. The experimental results on both synthetic and real-life image sequences evidence the effectiveness of the proposed scheme and substantial performance gains over earlier approaches.

In this paper, our focus is on tracking curvilinear structures without branches. We note that curvilinear structures with branches can be represented with a tree-graph and the deformable trellis-based approach is extendible to identify the most probable paths within such a graph. However, such extensions would require non-trivial expansion of the scope and volume of the paper, which will be considered in a future work.

## References


[1] J. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge, U.K.: Cambridge Univ. Press, 1999.

[2] J. Melonakos, E. Pichon, S. Angenent, and A. Tannenbaum, "Finsler active contours," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 3, pp. 412–423, Mar. 2008.

[3] A. Altinok, M. El-Saban, A. Peck, L. Wilson, S. Feinstein, B. Manjunath, and K. Rose, "Activity analysis in microtubule videos by mixture of hidden markov models," in *Proc. 2006 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2006, vol. 2, pp. 1662–1669.

[4] T. Koller, G. Gerig, G. Szekely, and D. Dettwiler, "Multiscale detection of curvilinear structures in 2-d and 3-d image data," in *Proc. 5th Int. Conf. Comput. Vis.*, Jun. 20–23, 1995, vol. 1, pp. 864–869.

[5] C. Steger, "An unbiased detector of curvilinear structures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 2, pp. 113–125, Feb. 1998.

[6] D. Geman and B. Jedynak, "An active testing model for tracking roads in satellite images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 1, pp. 1–14, Jan. 1996.

[7] A. Can, H. Shen, J. Turner, H. Tanenbaum, and B. Roysam, "Rapid automated tracing and feature extraction from retinal fundus images using direct exploratory algorithms," *IEEE Trans. Inf. Tech. Biomed.*, vol. 3, no. 2, pp. 125–138, Jun. 1999.

[8] M. Martinez-Perez, A. Hughes, A. Stanton, S. Thom, A. Bharath, and K. Parker, "Segmentation of retinal blood vessels based on the second directional derivative and region growing," in *Proc. ICIP*, 1999, vol. 2, pp. 173–176.

[9] C. Kirbas and F. Quek, "A review of vessel extraction techniques and algorithms," *ACM Computing Surveys*, vol. 36, no. 2, pp. 81–121, 2004.

[10] D. Terzopoulos and R. Szeliski, "Tracking with kalman snakes," *Active Vis.*, vol. 1, pp. 3–20, 1993.

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

[12] R. Toledo, X. Orriols, X. Binefa, P. Radeva, J. Vitria, and J. Villanueva, "Tracking elongated structures using statistical snakes," in *Proc. IEEE Conf. Comp. Vis. Pattern Recognit.*, 2000, vol. 1, pp. 157–162.

[13] N. Paragios and R. Deriche, "Geodesic active contours and level sets for the detection and tracking of moving objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 3, pp. 266–280, Mar. 2000.

[14] A.-R. Mansouri, "Region tracking via level set pdes without motion computation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 947–961, 2002.

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

[16] N. Vaswani, Y. Rathi, A. Yezzi, and A. Tannenbaum, "Deform pf-mt: Particle filter with mode tracker for tracking nonaffine contour deformations," *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 841–857, Apr. 2010.

[17] 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, 2007.

[18] D. Geiger, A. Gupta, L. Costa, and J. Vlontzos, "Dynamic programming for detecting, tracking, and matching deformable contours," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 3, pp. 294–302, Mar. 1995.

[19] A. Amini, T. Weymouth, and R. Jain, "Using dynamic programming for solving variational problems in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 9, pp. 855–867, Sep. 1990.

[20] M.-P. Dubuisson-Jolly, C.-C. Liang, and A. Gupta, "Optimal polyline tracking for artery motion compensation in coronary angiography," in *Proc. 6th Int. Conf. Comput. Vis.*, Jan. 4–7, 1998, vol. 1, pp. 414–419.

[21] Y. Chen, Y. Rui, and T. Huang, "Jpdaf based hmm for real-time contour tracking," in *Proc. IEEE CVPR*, 2001, vol. 1, pp. I-543–I-550.

[22] Y. Chen, Y. Rui, and T. Huang, "Multicue hmm-ukf for real-time contour tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1525–1529, Sep. 2006.

[23] M. Sargin, A. Altinok, K. Rose, and B. Manjunath, "Deformable trellis: Open contour tracking in bio-image sequences," in *Proc, IEEE ICASSP*, Mar. 31–Apr. 4 2008, vol. 1, pp. 561–564.

[24] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[25] M. Sargin, A. Altinok, K. Rose, and B. Manjunath, "Tracing curvilinear structures in live cell images," in *Proc. ICIP*, Sept. 16–Oct. 19 2007, vol. 6, pp. VI-285–VI-288.

[26] M. Jacob and M. Unser, "Design of steerable filters for feature detection using canny-like criteria," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 1007–1019, Aug. 2004.

[27] W. Freeman and E. Adelson, "The design and use of steerable filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 9, pp. 891–906, Sep. 1991.

[28] M. Heath, *Scientific Computing: An Introductory Survey*. New York: McGraw-Hill Science/Engineering/Math, 2002.

[29] T. Cover and J. Thomas, *Elements of Information Theory*. Hoboken, NJ: Wiley-Interscience, 2006.

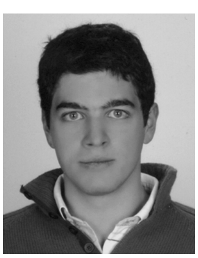
[30] S. Ross, *Introduction to Probability Models*. New York: Academic, 2000.

[31] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *Int. J. Comput. Vis.*, vol. 22, no. 1, pp. 61–79, 1997.

[32] K. Kamath, L. Wilson, F. Cabral, and M. A. Jordan, "Betaiii-tubulin induces paclitaxel resistance in association with reduced effects on microtubule dynamic instability," *J Biol Chem*, vol. 280, no. 13, pp. 12 902–12 907, 2005.

[33] S. Lobregt and M. Viergever, "A discrete dynamic contour model," *IEEE Trans. Med. Imag.*, vol. 14, no. 1, pp. 12–24, Mar. 1995.

[34] M. Sargin [Online]. Available: http://vision.ece.ucsb.edu/~msargin/open_contour_tracking



**Mehmet Emre Sargin** (S'06–M'10) received the B.Sc. degree in electrical and electronics engineering from Middle East Technical University, Turkey, in 2004, the M.Sc. degree in electrical and computer engineering from Koc University, Turkey, in 2006, and the Ph.D. degree in electrical and computer engineering from University of California, Santa Barbara, in 2010.

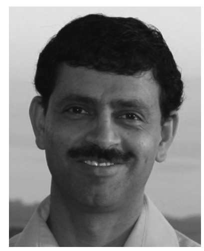
He is now with Google Inc., Mountainview, CA. His research interests include computer vision, pattern recognition, machine learning.

Dr. Sargin is the second prize winner in the ICASSP'07 student paper contest.

**Alphan Altinok** (M'06) received the Ph.D. in computer science from the University of California at Santa Barbara in 2006.

As a postdoctoral scholar with the Department of Electrical and Computer Engineering at UCSB, he worked on algorithms for tracking curvilinear structures in live cell images. He joined California Institute of Technology, Pasadena, in 2009 as a Research Scientist in computational biology. His research interests include computer vision and image analysis techniques for exploring biological processes depicted in image data.

Dr. Altinok a member of ASCB and IEEE.



**Bangalore S. Manjunath** (M'91–SM'04–F'05) received the B.E. degree from the Bangalore University, Bangalore, India, the M.E. degree from the Indian Institute of Science, India, and the Ph.D. degree in electrical engineering from the University of Southern California, Santa Barbara, in 1991.

He is now a Professor of electrical and computer engineering and Director of the Center for Bio-Image Informatics at the University of California, Santa Barbara. His research interests include large scale image/video databases, camera networks, information security and bio-image informatics. He has published over 200 peer reviewed articles and is a co-inventor on 22 US patents.

Dr. Manjunath has served as an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, PAMI, Multimedia and the IEEE SP Magazine, and is currently an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY.

**Kenneth Rose** (S'85–M'91–SM'01–F'03) received the Ph.D. degree in 1991 from the California Institute of Technology, Pasadena.

He then joined the Department of Electrical and Computer Engineering, University of California at Santa Barbara, where he is currently a Professor. His main research activities are in the areas of information theory and signal processing, and include rate-distortion theory, source and source-channel coding, audio and video coding and networking, pattern recognition, and nonconvex optimization. He is interested in the relations between information theory, estimation theory, and statistical physics, and their potential impact on fundamental and practical problems in diverse disciplines.

Dr. Rose was co-recipient of the 1990 William R. Bennett Prize Paper Award of the IEEE Communications Society, as well as the 2004 and 2007 IEEE Signal Processing Society Best Paper Awards.