Published in final edited form as:
IEEE Trans Med Imaging. 2012 May ; 31(5): 1008–1020. doi:10.1109/TMI.2011.2178122.

Tracking monotonically advancing boundaries in image sequences using graph cuts and recursive kernel shape priors

Joshua Chang,
UCLA Department of Biomathematics

KC Brennan, and
University of Utah Department of Neurology

Tom Chou
UCLA Departments of Biomathematics and Mathematics

Abstract

We introduce a probabilistic computer vision technique to track monotonically advancing boundaries of objects within image sequences. Our method incorporates a novel technique for including statistical prior shape information into graph-cut based segmentation, with the aid of a majorization-minimization algorithm. Extension of segmentation from single images to image sequences then follows naturally using sequential Bayesian estimation. Our methodology is applied to two unrelated sets of real biomedical imaging data, and a set of synthetic images. Our results are shown to be superior to manual segmentation.

Index Terms

Contour tracking; Level set method; Particle filter; Gaussian Process; Bayesian Vision; Graph cut; Segmentation; Shape Prior; Shape Statistics; Cortical Spreading Depression; Wound healing assay; Optical Intrinsic Signal Imaging; Gaussian Markov Random Fields

I. Introduction

BOUNDARY tracking is a common problem in many imaging applications, particularly in biomedical imaging where the dynamics of a wavefront or the boundary of a region of interest provide insights into the underlying physiological processes. Many processes are characterized by monotonic boundary movement, where a boundary crosses each point at most once. In biology and medicine, examples of such moving boundaries include cardiac impulse waves [1], tumors [2], extent of cerebral infarct [3], glial cell signal waves [4], and spreading depression waves [5].

The motion of monotonically advancing boundaries is described by the eikonal equation [6, 7]

$$V(s)|\nabla T(s)|=1, \quad (1)$$

where $V(s):\mathbb{R}^d \to \mathbb{R}^+$ is the speed of the interface along its normal when it crosses a point $s \in \mathbb{R}^d$, and $T(s):\mathbb{R}^d \to \mathbb{R}^+$ is the time when the boundary first reaches s. For collapsing boundaries, $V(s)$ is strictly negative, and $-V(s)|\nabla T(s)|=1$. Computationally, several algorithms for approximating solutions to the eikonal equation exist, notably the fast marching method [6], a quick $\mathscr{O}(N \log N)$ solver. This method has an interpretation as a special *boundary value* case of the more general *initial value* level set method [7] for describing boundary motion.

The level set method, as proposed by Osher and Sethian [8], is another approach to computing boundary motion. It describes the evolution in time t of a closed hyper-surface $\Gamma(t)$ of dimension $d - 1$ that bounds an evolving region $\Omega(t) \subset \mathbb{R}^d$. Instead of directly tracking the hyper-surface $\Gamma(t)$, one embeds $\Gamma(t)$ in an object of higher dimension known as a level set. The interface is embedded as the zero-level set of a function $\phi_{\Omega(t)}(s):\mathbb{R}^d \to \mathbb{R}$, which for every $s \in \mathbb{R}^d$ is the signed Euclidean distance from s to the boundary of $\Omega(t).\phi_{\Omega(t)}(s)$ is by convention negative if $s \in \Omega(t)$, and positive if $s \in \mathbb{R}^d \backslash \Omega(t)$. The level set equation

$$\frac{\partial \phi_\Omega}{\partial t}+V(s)|\nabla \phi_\Omega|=0 \qquad (2)$$

then describes the interface undergoing movement with speed $V(s)$ along its normal vectors. Computationally, one typically discretizes the level set function (Fig 1) and solves Eq 2 using finite differences. The main advantages of this approach are its freedom from parameters, and its ease of handling topological changes, especially in comparison to Lagrangian methods of curve evolution.

Both fast marching and level set methods have been useful for computationally describing moving interfaces in a variety of physical applications; the speed of the interface encodes the interesting physics of the observed process, and can be a function of any number of observed or unobserved covariates. For example, a curvature-dependent speed has been used to model flame propagation [8]. In biomedical applications, Malladi et al. [9] and others [2, 10, 11] have used level sets to model tumor growth. Sermesant et al. [12] used level sets to model cardiac electrophysiology. Recently, Wolgemuth and Zajac [13] used level sets to model cell motility.

Here, we develop a theory for the segmentation and tracking of boundaries that move according to the eikonal equation. We regularize our tracking by recursively estimating the position of the boundary, using a statistical model for the boundary speed based upon its observed history. The recursive estimation is weighted against evidence of the boundary in the image sequence within a Bayesian filtering framework.

We provide three applications of our method. First, we identify a boundary in a synthetic image sequence, validating our method against a “ground truth.” Second, we track cortical spreading depression waves in *in-vivo* image sequences. Cortical spreading depression

(CSD) arises in many pathologies such as stroke, brain trauma, epilepsy, and migraine. It is characterized by a slow-moving, concentric, traveling wave of runaway excitation propagating through contiguous regions of brain gray-matter. Lastly, we demonstrate the ability of our method to detect the boundary of a collapsing unhealed wound region. Here, a "wound" is scratched into a monolayer of cultured cells and the closing wound edge formed by the leading cells is tracked.

A. Related prior work in literature

There has been much work in tracking boundary motion in image sequences, using a variety of methods. One method is the use of level set-based active contours on the gradient field of image sequences [14, 15, 16]. In these methods, the minimum of a phenomenologically defined energy defines the contour positions. Other successful methods have utilized statistical models in order to define energies. Mansouri [17] developed a Bayesian tracking method where motion information is not computed. Many other authors have developed variants of Kalman filtering [18, 19, 20], and more generally, Bayesian filtering [21, 22, 23, 23].

Bayesian filtering is a sequential technique for estimating an unknown probability density as it evolves in time [24]. When the underlying state space is first-order Markovian, Bayesian filtering amounts to recursively predicting the next state given the observations up to the current state, and updating the prediction of the next state as the next observation comes in. The prediction of the next state acts as a prior. This prediction step entails propagating the probability distribution for the current state through the system dynamics (generally nonlinearly) to generate a predictive distribution for the next state. A popular variant of Bayesian filtering is the particle filter [25]. In the particle filter, one draws a weighted sample of states from the current state posterior, and then propagates each sample state according to the system dynamics to construct the prediction distribution.

We now take a moment to provide some background about image segmentation, the labeling of regions in images. Bayesian filtering for boundary detection can be thought of as recursive segmentation regularized by a motion model. To segment a closed subset of a region into a foreground set, Mumford and Shah [26] proposed an optimization problem. Suppose that an image $I:S \subset \mathbb{R}^2 \to \mathbb{R}$ contains pixels that represent two regions, foreground (Ω), and background ($\Delta = S \setminus \Omega$), separated by a closed boundary $\Gamma = \partial\Omega$. The optimal segmentation is the construction of a piecewise-smooth image I_0 that is found by minimizing an energy functional involving pixel-mismatches and a boundary-length penalty. Motivated by the idea of minimizing this functional, a class of methods known loosely as active contour methods were born [27]. Within this class of methods, level set approaches have achieved possibly the most success. Chan and Vese [28] were among the first to employ the level set method to minimize a Mumford-Shah like energy, where they used level sets to describe a gradient descent minimization. These methods, however, are plagued by slow convergence, and by reliance on the placement of an initial labeling.

Recently, computer vision researchers have employed graph-cuts based optimization to solve the segmentation problem. In this method, a segmentation corresponds to a two-coloring of a graph, where pixels constitute nodes with coloring representing membership to either the foreground or background sets. This coloring is typically found in low order polynomial time using a maximum-flow/minimum-cut algorithm [29]. However, graph cut methods can be applied only on a restricted class of energies [30, 31]. Despite much work in the field, it is challenging to incorporate prior knowledge about shapes into an energy that is minimizable within the graphcuts framework. For static image segmentation of general shapes, popular priors in use are the star-shaped prior [32], elliptical shape prior [33], and the compact shape prior [34]. For more application-specific segmentation, where the shape is known, researchers have expressed shapes in kernel principal component space [35, 36], where parameterized shapes are reduced to their eigenspace, as indicator functions [37], or implicitly embedded in levelsets [38, 39, 40].

Shape priors are useful for tracking applications since they can represent models for the motion of objects in an image sequence. Dynamical and statistical shape priors are particularly useful since they can account for uncertainty of an object's motion. There are many strategies for modeling this motion and generating associated shape priors. One approach does not seek to model the motion of the object, but instead relies on propagation of contours by gradient flow [14, 15]. Such methods however are incapable of detecting large changes in a boundary that correspond to shape changes, and are computationally expensive due to the reliance on solving a PDE.

Another common approach is to identify features of an object to track from a set of training images. For example, PCA-based methods [41] model deformation of features learned from a set of training templates. Other studies directly parameterize a particular target object [19], or contour [21, 22]. All of the aforementioned tracking methods have had success in many tracking applications, but they are not suitable for use in our desired applications without major modifications. These methods are restrictive in that they are intended to track objects that are expected to retain their overall structure in an image sequence.

B. Motivation for our method

In identifying the motion of a boundary evolving according to the eikonal equation, we seek to solve a subtly different tracking problem. As a boundary evolves, the region enclosed by the boundary need not retain any particular shape (Fig 3). Furthermore, it is not sufficient to only know the approximate location of the object, we seek precise discrimination of the object's boundaries. Finally, we are also motivated by the goal of online estimation, where we do not wish to rely on relatively slow techniques such as PDE-based energy minimization.

To solve these problems, our method introduces the concept of using recursive statistical shape priors in graph cuts segmentation. With the aid of a majorization-minimization step, we show how one can iteratively use graph cuts to arrive at a segmentation that takes into consideration an ensemble of predicted interface positions. This ensemble is constructed by modeling the speed of the interface using its observed history. The predicted ensemble is weighted against evidence of the interface position in the image, providing regularization that makes our method robust to real-world noise encountered in biomedical imaging.

II. Mathematical Approach

Our overall approach is to develop a recursive Bayesian filter to regularize segmentation of the evolving boundary. Our filter stochastically samples the possible motion of the boundary by evolving it against speeds sampled from a stochastic autoregressive model, whose parameters we recursively infer. The boundary is embedded implicitly as level sets in the arrival time function of Eq 1, and is propagated by the fast marching method. For the motion of the interface, we make only the loose assumption that its speed is locally correlated in space.

Given a set of interface positions $\{\Omega_k\}_{k=0}^K$, measured at times $\{t_k\}_{k=0}^K$, we calculate the past speed of the interface by bilinearly interpolating its past arrival times over the discrete image lattice, and then solving the eikonal equation (Eq 1) using second order upwind finite differences [42]. Denoting a vector containing these past speeds $\mathbf{V}_0$, we predict future speeds $\mathbf{V}$ using our stochastic model. Propagation of the interface against stochastic samples of $\mathbf{V}$ provides us with estimates of the future position of the interface.

In this section we will restrict our discussion to strictly positive speeds (advancing fronts). The same method applies for strictly negative speeds (collapsing fronts).

A. Motion prediction by Gaussian Process modeling

We use a Gaussian Markov Random Field (GMRF) to model the speed. With it, we can represent the spatial correlation of the speed of the boundary, where we expect speeds of the interface at neighboring locations to be close. In essence, we use a GMRF to penalize large spatial variations in our predicted speeds.

A GMRF is a stochastic field that follows a multivariate Gaussian distribution, and is Markovian with respect to a graph. The random variates of a GMRF constitute the nodes of a graph $\mathscr{G}$, with covariance matrix Σ such that an entry in the precision matrix $\mathbf{Q} = \Sigma^{-1}$ is nonzero if and only if there is an edge connecting the two corresponding random variates. A GMRF has a sparse precision matrix in that its corresponding graph is not complete – we assume that we can predict the speed at a position given only the speeds at nearby locations. GMRFs have been shown to approximate arbitrary covariance structures well in practice [43, 44], even when the correlation range is much larger than the size of the Markovian neighborhoods. We say that the speed $V:\mathbb{R}^2 \to \mathbb{R}^+$ of the interface as it crosses over $s \in \mathbb{R}^d$ is

$$V(s)=\mathbf{X}(s)\boldsymbol{\beta}+\eta(s),$$

where $\eta:\mathbb{R}^2 \to \mathbb{R}$ is a GMRF. Of perhaps most interest, $\mathbf{X}(s)$ is a $1 \times p$ vector of any *known* linear predictors whose influence is unknown. We determine the influence of these predictors by inferring the $p \times 1$ vector of coefficients $\boldsymbol{\beta}$. For example, if one has an anatomical labeling distinguishing two different tissue types in an imaging field, $X_1(s)$ could be an indicator of one of the tissue types. The speed of the interface is then $V(s) = \beta_0 + X_1(s)\beta_1 + \eta(s)$, so that the expected speed of the interface is $\beta_0 + \beta_1$ inside the region, and outside. Under this construct, it is possible to infer the coefficient β_1 and ascertain if it is distinct from zero, thereby statistically testing whether the interface travels through the two regions at different speeds. It is worthy to note that the predictors X_i are also allowed to be nonlinear functions of any auxiliary variables. It is convenient to infer the GMRF using data from all locations simultaneously, so we define a matrix **X** to be an $n \times p$ matrix, where each of the n rows corresponds to an observation point (in our case, speed at a particular location), and each row has p entries, corresponding to known predictor variables. Such a matrix is commonly known as a covariate matrix. **X**(s) refers to the row in **X** corresponding to point s.

It may be illuminating to write the model in the following hierarchical form

$$V(s)=\mathbf{X}(s)\boldsymbol{\beta}+\eta(s) \quad (3)$$

$$\eta|\tau_\eta^2 \sim \mathcal{N}(0,\mathbf{Q}) \quad (4)$$

$$\boldsymbol{\beta},\tau_\eta^2 \sim \mathcal{N}(\boldsymbol{\beta};\boldsymbol{\beta}_0,\tau_\eta^2\mathbf{P})G(\tau_\eta^2;a_\eta,b_\eta). \quad (5)$$

$\mathcal{N}(x;\mu,\tau^2)$ represents a Gaussian distribution on random variable x with mean μ and precision τ^2. $G(y; a, b)$ represents a statistician's gamma distribution on random variable y with shape a and rate b. For convenience, throughout this paper we parametrize Gaussian distributions with precisions $\tau_{(\cdot)}^2=1/\sigma_{(\cdot)}^2$ rather than the more commonly encountered variances. $\sigma_{(\cdot)}^2$ Eq 3 states that the speed is a linear composition of the covariates and their associated coefficients, plus a normally distributed spatial noise term with distribution given in Eq 4. **Q** has entries Q_{su} that are a function of the distance between two points s, and u, hence it is an $n \times n$ matrix. We set $\mathbf{Q}=\tau_\eta^2\mathbf{R}$, where **R** encodes spatial correlation between locations, and τ_η^2 is a scalar parameter independent of η. For simplicity we will assume we know **R**; we use the 5 × 5 Markovian Gaussian process given by Rue and Tjelmeland [43], to yield a sparse banded **R**.

In Eq 5, we place the normal-gamma conjugate prior distribution on the parameters $\boldsymbol{\beta}$, and τ_η^2. The hyperparameters a_η and b_η are set small (0.001) to be weakly informative [45]. **P** represents the precision of the prior knowledge of $\boldsymbol{\beta}$, scaled by τ_η^2. This model bears more than superficial resemblance to common spatial interpolation methods used in geostatistics. In particular, this model can be considered a Bayesian variant of universal kriging [46]. Inference on this model can be performed exactly and analytically as a generalized least squares problem [47]. It is important to note that we desire positive values for speeds $V(s)$. In this model we do not explicitly enforce positivity of V; however, if μ is large enough, as in the applications we discuss, fields with negative values of V have negligible measure. We simply discard any samples of V that have negative values. In this section, we provide the maximum a-posterior estimates for $\boldsymbol{\beta}$, and $\mathbf{V}$. Please see the supplementary document for details of the derivations.

Suppose one has a $n \times 1$ vector $\mathbf{V}_0$ of observed speeds, at locations with associated $n \times n$ spatial correlation encoding matrix $\mathbf{R}_0$, and $n \times p$ covariate matrix $\mathbf{X}_0$. Then one may perform maximum-a-posterior inference to predict speeds at unobserved locations. The model given in Eqs 3–5 yields a posterior t-distribution on $\boldsymbol{\beta}$ with location

$$\hat{\boldsymbol{\beta}}=(\mathbf{X}_0^T\mathbf{R}_0\mathbf{X}_0+\mathbf{P})^{-1}(\mathbf{X}_0^T\mathbf{R}_0\mathbf{V}_0+\mathbf{P}\boldsymbol{\beta}_0), \quad (6)$$

and scale

$$\begin{aligned}\mathbf{A}_{\boldsymbol{\beta}}=&\frac{1}{n+2a_\eta}(\mathbf{X}_0^T\mathbf{R}_0\mathbf{X}_0+\mathbf{P})^{-1}\\ &\times\Big(2b_\eta+\boldsymbol{\beta}_0^T\mathbf{P}\boldsymbol{\beta}_0+\mathbf{V}_0^T\mathbf{R}_0\mathbf{V}_0\\ &-\hat{\boldsymbol{\beta}}^T(\mathbf{X}_0^T\mathbf{R}_0\mathbf{X}_0+\mathbf{P})\hat{\boldsymbol{\beta}}\Big).\end{aligned} \quad (7)$$

Let $\mathbf{V}$ be a vector of unknown speeds that we would like to estimate at m new locations. Suppose that these locations are associated with the ***known*** $m \times p$ covariate matrix $\mathbf{X}$. Denoting the $m \times m$ spatial correlation encoding matrix for these new points $\mathbf{R}$, and a $m \times n$ matrix that encodes correlation between these points and the original n points as U, one finds that the posterior $\mathbf{V}$ is t-distributed with location

$$\hat{\mathbf{V}}=\overbrace{\mathbf{X}\hat{\beta}}^{\text{mean}}-\overbrace{\mathbf{R}^{-1}\mathbf{U}\left(\mathbf{V}_0-\mathbf{X}_0\hat{\beta}\right)}^{\text{spatially correlated noise}} \quad (8)$$

and scale

$$\begin{aligned}\mathbf{A}=&(\mathbf{X}+\mathbf{R}^{-1}\mathbf{U}\mathbf{X}_0)\mathbf{A}_{\boldsymbol{\beta}}(\mathbf{X}+\mathbf{R}^{-1}\mathbf{U}\mathbf{X}_0)^T\\ &+\frac{1}{n+2a_\eta}\mathbf{R}^{-1}\Big(2b_\eta+\boldsymbol{\beta}_0^T\mathbf{V}\boldsymbol{\beta}_0+\mathbf{V}_0^T\mathbf{R}_0\mathbf{V}_0\\ &-\hat{\boldsymbol{\beta}}^T(\mathbf{X}_0^T\mathbf{R}_0\mathbf{X}_0+\mathbf{P})\hat{\boldsymbol{\beta}}\Big).\end{aligned} \quad (9)$$

One sees from Eq 8 that the mean predicted speed is a sum of a mean term and a spatially correlated term that appears as a convolution. Entries in the convolution term correspond to values of η(s) in Eq 3.

We sample from the posterior speed field by using the normal approximation to the t-distribution. Such an approximation is justified since the degrees of freedom in the model is quite large. To sample the m desired speeds we first calculate the Cholesky decomposition $\mathbf{A}$

= $\mathbf{LL}^T$[48]. Then we sample a $m \times 1$ vector of independent standard normal values $\mathbf{Z}$. Finally, the vector $\hat{\mathbf{V}}+\mathbf{L}Z$ constitutes a sample from the desired distribution.

Finally, we sample predictions of future interface positions at t_{k+1} by solving the eikonal equation with Dirichlet boundary condition $T(s \in \partial\Omega_k)=t_k$, by fast marching the interface with the sampled speed fields for times $t_k<t \leq t_{k+1}$.

B. Generative model for static image segmentation

With the ability to predict the future position of an interface from its past history, we now turn our attention to extracting information about the interface position from images. At the core of this exercise is a generative computer vision model. We model an image probabilistically with normal distributions of intensity values conditional on region:

$$I(s) \sim \begin{cases} \mathscr{N}\left(\mu_\Omega, \tau_\Omega^2\right) & s \in \Omega \quad \text{(foreground);} \\ \mathscr{N}\left(\mu_\Delta, \tau_\Delta^2\right) & s \in \Delta \quad \text{(background).} \end{cases} \qquad (10)$$

Additionally, we incorporate information about Ω as a prior distribution $p(\Omega)$, and perform inference on the joint posterior $p(\Omega, \theta|I) \propto p(I|\Omega, \theta)p(\theta|\Omega)p(\Omega)=e^{-U(\Omega,\theta)}$, where $U(\Omega,\theta)$ is an energy, and $\theta=[\mu_\Omega, \mu_\Delta, \tau_\Omega^2, \tau_\Delta^2]$ is a vector of the Gaussian image intensity parameters.

In this formulation, $p(\theta|\Omega)$ represents the prior knowledge of the regional image intensities, and $p(\Omega)$ represents the prior knowledge of the underlying foreground shape and location. For $p(\theta|\Omega)$, we use the normal-gamma conjugate prior distribution

$$\begin{aligned} p(\theta|\Omega)=&\mathscr{N}\left(\mu_\Omega;\overline{\mu}_\Omega, \tau_\Omega^2\right)\mathscr{N}\left(\mu_\Delta;\overline{\mu}_\Delta, \tau_\Delta^2\right) \\ &\times G\left(\tau_\Omega^2;a_\Omega, b_\Omega\right) G\left(\tau_\Delta^2;a_\Delta, b_\Delta\right) \end{aligned} \qquad (11)$$

with gamma hyper-prior [45] over the image-intensity precisions. The parameters $\overline{\mu}_\Omega, \overline{\mu}_\Delta$ are the prior regional mean image intensities, and a_Ω, b_Ω, a_Δ, b_Δ are hyperparameters which are set to be weakly informative (all equal to 0.001) unless otherwise stated.

To represent shape-knowledge, we use a kernel density estimate of the distribution of possible shapes embedded as discrete level sets. Let us denote χ_Ω the characteristic function for a region Ω,

$$\chi_\Omega(s)=\begin{cases} 1, & s \in \Omega; \\ 0, & s \notin \Omega. \end{cases} \qquad (12)$$

Then, for two shapes Ω and Λ, embedded as discrete signed distance functions ϕ_Ω and ϕ_Λ we introduce an asymmetric shape divergence

$$
\begin{aligned}
d(\Omega,\Lambda)= &\sum_{s} \overbrace{|\chi_{\Omega}(s)-\chi_{\Lambda}(s)|}^{\text{indicator of pixel mismatch}} \quad |\phi_{\Lambda}(s)|^{\alpha} \\
&+\sum_{s,u\in\mathbf{N}} \overbrace{|(\chi_{\Omega}(s)(1-\chi_{\Omega}(u))+\chi_{\Omega}(u)(1-\chi_{\Omega}(s))|}^{1\text{indicates } s,u \text{ lie on opposite sides of the boundary of } \Omega_1} \\
&\times w_{su}\left|\phi_{\Lambda}\left(\tfrac{s+u}{2}\right)\right|^{\alpha}, \qquad (13)
\end{aligned}
$$

where the parameter α represents how severely we penalize shape mismatch. The parameter w_{su} is one divided by the length of the edge (Euclidean distance between s and u), and **N** denotes the set of neighboring grid points. We take the grid points s to lie in the center of each grid cell, and use the eight neighbor system detailed in El Zehiry et al. [49], and represented pictorially in Fig 2. The term $\phi_{\Lambda}\left(\frac{s+u}{2}\right)$ refers to the signed distance between the midpoint of the $s - u$ segment and the boundary of Λ. This expression penalizes mismatches between the two shapes, and particularly penalizes protrusions in Ω not represented in Λ. Given this distance measure, we can define a Gaussian kernel of the form

$$K(\Omega,\Lambda)=\sqrt{\frac{\tau^2}{2\pi}e^{-\frac{\tau^2}{2}d(\Omega,\Lambda)}}. \qquad (14)$$

Now, using some available reference shapes $\{\Omega_j\}_{j=1}^{J}$ and carefully-chosen weighting coefficients w_j, we can represent a distribution over shapes, $p_S(\Omega)$ as follows:

$$p_S(\Omega) \propto \sum_{j=1}^{J} w_j K(\Omega,\Omega_j).$$

This representation of the prior is the kernel density estimate (KDE) of the distribution of shapes. Like in Cremers [41] and Cremers et al. [50], we empirically set **τ**^{2} to the following value:

$$\tau^2=\left[\sum_{j=1}^{J} w_j \min_{k\neq j} d(\Omega_k,\Omega_j)\right]^{-1}. \qquad (15)$$

To impose smoothness on the boundary of Ω, we additionally assume a prior on the boundary length, $p_l(\Omega) \propto e^{-v\mathscr{H}(d\Omega)}$, where $\mathscr{H}(d\Omega)$ is the Hausdorff measure or length of the curve. We use the discrete approximation of this measure found in El Zehiry et al. [49]. Finally, we can write our complete prior over Ω as follows:

$$p(\Omega) \propto e^{-v\mathscr{H}(d\Omega)}\sum_{j=1}^{J} w_j K(\Omega,\Omega_j).$$

We wish to infer the segmentation Ω by maximizing the posterior probability relative to Ω. To this end, we will maximize the logarithm of the posterior, or equivalently, minimize the following energy:

$$\begin{aligned} U(\Omega,\theta) &= -\,log\, \overbrace{[p(I|\Omega,\theta)}^{\text{likelihood}}\overbrace{p(\theta|\Omega)p(\Omega)}^{\text{prior}} \\ &= \frac{1}{2}\sum_{s\in\Omega}\left[log\left(\frac{2\pi}{\tau_\Omega^2}\right)+\tau_\Omega^2(I(s)-\mu_\Omega)^2\right] \\ &\quad+\frac{1}{2}\sum_{s\in\Delta}\left[log\left(\frac{2\pi}{\tau_\Delta^2}\right)+\tau_\Delta^2(I(s)-\mu_\Delta)^2\right] \\ &\quad+\underbrace{v\mathscr{H}(d\Omega)}_{\text{length penalty}}\;\underbrace{-\,log\sum_{j=1}^{J}w_jK(\Omega,\Omega_j)}_{KDE\text{ of shapes}} \\ &\quad-log\,p(\theta|\Omega)+\text{const.} \end{aligned} \qquad (16)$$

We take an iterative two-step approach to minimizing this energy. Given Ω, we minimize the energy with respect to *θ* directly by setting the gradient of the energy with respect to *θ* to zero, and solving for *θ* (see O'Hagan et al. [45]). Then, given *θ*, we find the optimal Ω using the majorization- minimization algorithm described in the following section. We repeat this two-step procedure until a stable energy is reached.

C. Majorization-minimization (MM) algorithm

The shape contribution $log\sum w_jK(\Omega,\Omega_j)$ can make minimization of the energy difficult, since its formulation involves a sum within a logarithm. To separate the contributions from the reference shapes Ω_j, we will derive a surrogate function with separated terms. A function $f(x|x_k)$ is said to majorize a function $g(x)$ at x_k if $g(x) \le f(x|x_k)$, $\forall x$, and if $f(x_k) = g(x_k|x_k)$ [51]. We perform inference by iteratively computing $\Omega^{(n-1)} = \arg\min_\Omega Q(\Omega|\Omega^{(n)})$, where $Q(\Omega|\Omega^{(n)})$ majorizes Eq 16. Noting that $-\log(\cdot)$ is convex, we will use a definition for convexity $f(\sum_i\alpha_it_i) \le \sum_i\alpha_if(t_i)$, to show that for any segmentation $\Omega^{(n)}$, the following holds:

$$\begin{aligned} & -log\overbrace{\sum_{j=1}^{J} w_j K(\Omega, \Omega_j)}^{\text{shape kernel density}} \leq \\ & -\sum_{j=1}^{J} \frac{w_j K(\Omega^{(n)}, \Omega_j)}{\sum_{k=1}^{J} w_k K(\Omega^{(n)}, \Omega_k)} \\ & \times log\left[\frac{\sum_{k=1}^{j} w_k K(\Omega^{(n)}, \Omega_k)}{w_j K(\Omega^{(n)}, \Omega_j)} w_j K(\Omega, \Omega_j) \right] \\ & = -\sum_{j=1}^{J} \frac{w_j K(\Omega^{(n)}, \Omega_j)}{\sum_{k=1}^{J} w_k K(\Omega^{(n)}, \Omega_k)} log\, K(\Omega, \Omega_j) \\ & +\text{const} \end{aligned}$$

substituting Eq 14

$$\begin{aligned} & = \underbrace{\frac{\tau^2}{2} \sum_{j=1}^{J} \frac{w_j K(\Omega^{(n)}, \Omega_j)}{\sum_{k=1}^{J} w_k K(\Omega^{(n)}, \Omega_k)} d(\Omega, \Omega_j)}_{\text{separated log shape kernel density}} \\ & +\text{const}. \end{aligned} \qquad (17)$$

Since Eq 17 majorizes the log-kernel density, we can minimize our original energy by iteratively minimizing

$$\begin{aligned} & Q(\Omega|\Omega^{(n)}) = \\ & \tfrac{1}{2}\sum_{s\in\Omega}\left[log\left(\frac{2\pi}{\tau_\Omega^2}\right) + \tau_\Omega^2 (I(s) - \mu_\Omega)^2 \right] \\ & + \tfrac{1}{2}\sum_{s\in\Delta}\left[log\left(\frac{2\pi}{\tau_\Delta^2}\right) + \tau_\Delta^2 (I(s) - \mu_\Delta)^2 \right] \\ & + v\mathscr{H}(d\Omega) - logp(\theta|\Omega) \\ & + \tfrac{\tau^2}{2} \sum_{j=1}^{J} \frac{w_j K(\Omega^{(n)}, \Omega_j)}{\sum_{k=1}^{J} w_k K(\Omega^{(n)}, \Omega_k)} d(\Omega, \Omega_j). \end{aligned} \qquad (18)$$

Since the distance function can be written as a sum over the vertices, so can Eq 18. As a result, it is possible to minimize Eq 18 within the graph cuts framework described in the next section.

D. Graph cuts for segmentation

Here, we describe minimization of the surrogate energy in Eq 18 using graph cuts, which quickly finds a global minimum of a restricted set of energies. Graph cut methods have their grounding in combinatorial optimization theory, and are concerned with finding the minimum cut in an undirected graph. A cut is a partition of a connected graph into two disconnected sets. The cost of a cut is the sum of the edge weights along a cut, and a max-flow min-cut algorithm finds the cut with the lowest cost. To use graph cuts for image segmentation, we must express our energy function in terms of edge-weights on a graph. We will describe an image as a connected graph, where each pixel represents a node, and edges exist between neighboring nodes (Fig 2). Note that edges in this context refer to connections between nodes in a graph, and not to edges in an image. We want to infer an unknown two-coloring on the nodes of the graph that represents inclusion of a node s into either the foreground set Ω, or the background set Δ. Following El Zehiry et al. [49], we begin by expressing the energy given in Eq 18 as a function of the vertices V and edges $\mathscr{E}$ of a graph $\mathscr{G}=(V,\mathscr{E})$:

$$U(\mathscr{G})=\sum_{s\in V}U_V(s)+\sum_{(s,u)\in\mathscr{E}}U_{\mathscr{E}}(s,u).$$

From equation Eq 18 we find that

$$\begin{aligned}U_V(s)=&\tfrac{1}{2}\left[log\left(\tfrac{2\pi}{\tau_\Omega^2}\right)+\tau_\Omega^2(I(s)-\mu_\Omega)^2\right]\chi_\Omega(s)\\&+\tfrac{1}{2}\left[log\left(\tfrac{2\pi}{\tau_\Delta^2}\right)+\tau_\Delta^2(I(s)-\mu_\Delta)^2\right](1-\chi_\Omega(s))\\&+\tfrac{\tau^2}{2}\sum_{j=1}^{J}\frac{w_jK(\Omega^{(n)},\Omega_j)|\phi_{\Omega_j}(s)|^\alpha|\chi_\Omega(s)-\chi_{\Omega_j}(s)|}{\sum_{k=1}^{J}w_kK(\Omega^{(n)},\Omega_k)}\end{aligned}\qquad(19)$$

and

$$\begin{aligned}&U_{\mathscr{E}}(s,u)=\\&\tfrac{\pi\tau^2}{16w_l(s,u)}\left[\sum_{j=1}^{J}\frac{w_jK(\Omega^{(n)},\Omega_j)|\phi_{\Omega_j}(\tfrac{s+u}{2})|^\alpha}{\sum_{k=1}^{J}w_kK(\Omega^{(n)},\Omega_k)}\right.\\&\left.\times|\chi_{\Omega_j}(s)(1-\chi_{\Omega_j}(u))+\chi_{\Omega_j}(u)(1-\chi_{\Omega_j}(s))|\right]\\&+vw_l(s,u)|\chi_\Omega(s)(1-\chi_\Omega(u))+\chi_\Omega(u)(1-\chi_\Omega(s))|.\end{aligned}\qquad(20)$$

In Eq 20, $w_l(s, u)$ is an edge weighting that approximates the Hausdorff measure of the boundary [49]. In the eight-neighbor system we use, it takes values of $w_{su}\pi/8$. Weighting of edges in this manner helps enforce homogeneity of labeling between neighboring spatial points. It is of note that the energy depends upon the distance functions for the kernel density reference shapes, and not the evolving segmentation. If it were to depend on the signed distance function of the segmentation, it would not be possible to write the energy in a form minimizable in a graph cuts framework, thus necessitating the use of an asymmetric shape distance like the one in Eq 13.

To minimize the energy, we augment our pixel lattice graph with two special nodes. In the language of graph cuts, these nodes are known as the source and sink. For our purposes, the labeling of source and sink are arbitrary since we are dealing with an undirected graph. We will call one of these vertices v_Ω, and the other one v_Δ. The existence of an edge between a pixel s and v_Ω will represent the segmentation of s into Ω. We then connect each pixel node directly to both v_Ω and v_Δ, and weight these new edges as follows:

$$w(s,v_\Omega)=\frac{1}{2}\left[log(\frac{2\pi}{\tau_\Delta^2})+\tau_\Delta^2(I(s)-\mu_\Delta)^2\right] + \frac{\tau^2}{2}\sum_{j=1}^{J}\frac{w_jK(\Omega^{(n)},\Omega_j)\chi_{\Omega_j}(s)}{\sum_{k=1}^{J}w_kK(\Omega^{(n)},\Omega_k)}|\phi_{\Omega_j}(s)|^\alpha, \tag{21}$$

$$w(s,v_\Delta)=\frac{1}{2}\left[log(\frac{2\pi}{\tau_\Omega^2})+\tau_\Omega^2(I(s)-\mu_\Omega)^2\right] + \frac{\tau^2}{2}\sum_{j=1}^{J}\frac{w_jK(\Omega^{(n)},\Omega_j)|\phi_{\Omega_j}(s)|^\alpha(1-\chi_{\Omega_j}(s))}{\sum_{k=1}^{J}w_kK(\Omega^{(n)},\Omega_k)}. \tag{22}$$

The cutting of the edge from an s to v_Ω implies that $s \in \Delta$, so it adds to the cost of the cut by the contribution of s into the total energy as if $s \in \Delta$. In other words, these weights can be interpreted as a pixel's strength of belonging to each region. If s is in a particular region, then neighbors of s are more likely to be in the same region. This fact is represented by edge weights between neighboring pixels. Those weights are

$$\frac{\pi\tau^2}{16w_l(s,u)}\sum_{j=1}^{J}\frac{w_jK(\Omega^{(n)},\Omega_j)|\phi_{\Omega_j}\left(\frac{s+u}{2}\right)|^\alpha}{\sum_{k=1}^{J}w_kK(\Omega^{(n)},\Omega_k)} + vw_l(s,u) \tag{23}$$

where u is a grid-neighbor of s. Our surrogate energy is now minimized by finding the minimum cut of the graph. For details on how to perform this optimization, we refer the reader to Boykov and Kolmogorov [52]. To minimize the original energy function, one iteratively computes the graph-cut minimum within the MM algorithm described in section II-C. It is of note that θ does not significantly change if the segmentation labels do not significantly change. Therefore, it is computationally beneficial to recompute θ after each of the first few MM iterations, and only recompute it if the labels undergo further large changes.

E. Bayesian filtering

We now have all the pieces needed to perform sequential Bayesian estimation of the interface positions. In section II-A, we described our model for predicting the speed and hence position of the interface. In section II-B, we showed how one can use a kernel density estimate of the position of the boundary in an image to craft an energy functional that when minimized yields the position of the boundary. In sections II-C and II-D, we provided an iterative method of minimizing the energy functional. We now describe how to combine these components together into a Bayesian filter. Let $\Omega_{k|k-1}$ be the random variable Ω_k conditional on all images up to and including time $t_{k\text{-}1}$. Then for each frame I_k in an image sequence, after initialization, our Bayesian filter iterates between two steps, *predict* and *update*.

Initialization—To initialize our segmentation method, one needs an initial segmentation at the first image frame. Possessing prior shape knowledge, one may initialize the segmentation with a set of prior shape templates [41], and then minimize Eq 16 directly. In the absence of shape information, one may create a non-informational shape prior by defining a single arbitrary reference shape, and setting $\tau^2 \rightarrow 0$. This procedure reduces the energy function to pure intensity-based graph cuts.

Predict—We draw samples of segmentations $\Omega_{k|k}$ from its posterior, and infer the GMRF speed field associated with each segmentation. Then for each segmentation sample, propagate the associated interface through samples from its speed field to generate an ensemble of predictive positions for $\Omega_{k+1|k}$.

The posterior of $\Omega_{k/k}$ is defined by an energy

$$\begin{aligned} &U(\Omega_{k|k}, \theta_k)= \\ &\frac{1}{2}\sum_{s\in\Omega_{k|k}}\left[\tau^2_{\Omega_{k|k}}(I_k(s)-\mu_{\Omega_{k|k}})^2 - log\left(\tau^2_{\Omega_{k|k}}\right)\right] \\ &+\frac{1}{2}\sum_{s\in\Delta_{k|k}}\left[\tau^2_{\Delta_{k|k}}(I_k(s)-\mu_{\Delta_{k|k}})^2 - log\left(\tau^2_{\Delta_{k|k}}\right)\right] \\ &+\upsilon\mathscr{H}(\partial\Omega_{k|k}) - log\, p(\theta_k|\Omega_{k|k}) \\ &-log\sum_{j=1}^{J}\sum_{l=1}^{L}W_jK\left(\Omega_{k|k}, \Omega^{(j,l)}_{k|k-1}\right), \end{aligned} \tag{24}$$

for some known weighting coefficients W_j and reference shapes $\Omega^{(j,l)}_{k|k-1}$.

We sample segmentations from the posterior by using importance sampling [53]. Importance sampling obtains samples from a difficult target distribution $f(x)$ by sampling from an easier importance distribution $g(x)$. With samples $\mathbf{X}_i \sim g(\mathbf{x})$, one approximates expectations of a function $h(\mathbf{x})$ under the distribution $f(\mathbf{x})$ by first calculating the weights $W_i = f(\mathbf{X}_i)/g(\mathbf{X}_i)$, and then approximating the expectation by $\mathbf{E}[h(\mathbf{x})] = \Sigma\ W_i h(\mathbf{X}_i)/\Sigma\ W_i$. To approximate the target distribution itself, one can combine kernel density estimation and importance sampling to deduce an approximating distribution of the form $\hat{f}(\mathrm{x})=\sum W_i K(\mathrm{x}, \mathbf{X}_i)/\sum W_i$, where $K(\cdot,\cdot)$ is a kernel function.

We sample a set of current interface positions $\left\{\Omega^{(j)}_{k|k}\right\}^{J}_{j=1}$. as the conditional maximum-a-posterior segmentations under modified length penalties $v^{(j)}$, where each $v^{(j)}$ follows an exponential distribution with rate parameter $1/v$. With each $v^{(j)}$, we modify the length penalty in Eq 24 by setting $v \rightarrow v^{(j)}$ and minimize the resulting energy to obtain a sample interface position $\Omega^{(j)}_{k|k}$ with associated image intensity parameters $\theta^{(j)}_{k}$. Conditional on each $\Omega^{(j)}_{k|k}$, we infer the associated stochastic speed field $\mathbf{V}^{(j)}_{k|k}|\Omega^{(j)}_{k|k}$, which follows a multivariate *t*-

distribution with scale matrix $\mathbf{A}_{k|k}^{(j)}$. To infer this field we first linearly interpolate arrival times of the interface from the set of wave positions. Then we calculate the past speeds according to the eikonal equation using second-order upwind finite differences [42], giving us a vector of known speeds $\mathbf{V}_0$. Finally, we input the resulting speeds into the GMRF model of section II-A, where Eqs 6, 7, 8, and 9 provide us with the parameters of the resulting multivariate t-distribution. Then, from each GMRF, we draw a fixed number of samples (we used 16). Thus, for each sampled interface position $\Omega_{k|k}^{(j)}$, we have a collection of samples of the interface speed $\left\{\mathbf{V}_{k|k}^{(j,l)}\right\}_{l=1}^{L}$.

With our states $\left\{\Omega_{k|k}^{(j)},\left\{\mathbf{V}_{k|k}^{(j,l)}\right\}_{l=1}^{L}\right\}_{j=1}^{J}$ at t_k, and the eikonal equation, we may now predict the location of the boundary at t_{k+1} by propagation of each pair of interface and speed field through Eq 1 to calculate $\Omega_{k+1|k}^{(j,l)}$. This procedure is accomplished by using the fast marching method [6], starting with an initial position $\Omega_{k|k}^{(j)}$ at t_k, and solving with speeds $\mathbf{V}_{k|k}^{(j,l)}$ until $t = t_{k+1}$. The result is a set of predictive samples $\left\{\Omega_{k+1|k}^{(j,l)}\right\}_{j=1,l=1}^{J,L}$ of the interface location at time t_{k+1}, given the information up to time t_k.

The weights in Eq 24 are then given by the importance sampling weights

$$W_j = exp\left[\frac{v^{(j)}}{v} - U\left(\Omega_{k|k}^{(j)},\theta_k^{(j)}\right)\right], \qquad (25)$$

normalized to sum to 1. Here, U is the energy function given in Eq 24 with the original shape penalty v. In Eq 16, we took the prior on Ω to be the kernel density estimate of reference shapes. We can compute this representation of the prior by setting $P\left(\Omega_{k+1|k}\right)\propto\sum_j\sum_l W_j K\left(\Omega_{k+1|k},\Omega_{k+1|k}^{(j,l)}\right)$. With our prior constructed, we now have all the components necessary to specify the posterior distribution for time t_{k+1}.

Update—When the new observation at time t_{k+1} comes in, one updates his prediction of the state at t_{k+1} by minimizing the posterior energy $U\left(\Omega_{k+1|k+1},\theta_{k+1}\right)$. This update is as simple as relaxing the energy (Eq 24 with k incremented by one) by the MM procedure described in section II-C, iterated with estimation of the image intensity statistics. We initialize the MM algorithm to start at the state obtained by propagating the maximum-a-posterior $\Omega_{k|k}$ according to its maximum-a- posterior speed field (Eq 8), and initially inferring the image intensity statistics conditional on this state.

III. Application to Biomedical Images

We implemented the segmentation and speed interpolation method given in previous sections as a Java-based plug-in for ImageJ, the image manipulation and analysis package from the NIH [54]. We chose UJMP [55], a LGPL licensed fast matrix library, to perform all matrix operations. For graphcuts optimization, we modified Fiji's [56] implementation of the max-flow min-cut algorithm by Boykov and Kolmogorov [52] for our needs. In these examples, the shape mismatch penalty was set to $\alpha = 2$, and the length penalty was set to $\nu = 20$, unless otherwise stated. Convergence to a minimum energy typically occurred within five to nine MM iterations. It is worthy to note that one need not recompute the image intensity statistics θ at each step if only a small number of labels have changed. We only recalculated the image intensity statistics in the first three iterations, where convergence was most rapid. All segmentations were performed on sequences of 320 × 240 images.

A. Synthetic image sequence

To test the ability of our method to recover a known interface from images, we applied our method to a sequence of synthetic images (Fig 4). We defined a "ground truth" speed field within a 320 × 240 pixel spatial field using the UCLA logo (Fig 5, left), with speeds of 10 pixels/frame and 6 pixels/frame inside and outside of the letters, respectively. Then, we advected a boundary traveling from the lower right of the field until it crossed the entire field (total of 90 frames). From this set of interface positions, we generated a series of images, with mean intensities of 3 inside the region traversed by the boundary, and 0 outside, with a variance of 9 (Fig 4). We used weakly informative image intensity and uninformative initial shape priors to segment the resulting images. To estimate the speed, we used a weakly informational prior speed centered at 8 pixels/frame (weakly informative implies that all Gamma hyperparameters are set to 0.001).

The boundary that we found is in good accord with the ground truth. Our regularization is strong enough to provide robustness to noise, yet, not so strong that relatively large deformations are not detected. The ground truth image undergoes robust changes in topology that are faithfully reproduced in our segmentation results. Fig 5 shows our reconstruction of the ground truth interface speed, performed using second-order upwind finite differences. The original UCLA logo is clearly visible in the reconstructed speed (right). There is noisiness in the reconstruction due to the indeterminacy of the inversion of the eikonal equation (Eq 1), however we obtain an accurate reconstruction. The mean reconstructed speed is 9.8 ± 1.5 pixels per frame within the UCLA lettering, and 6.0 ± 0.7 pixels per frame outside of the UCLA lettering; these values are within 5% of the original speed field.

In Fig 6, we show that even large deformations far from the predicted mean shape are detected, typically in very few iterations. Here, protrusions in the front are developing such that the shape of the front is significantly different from the mean predicted shape. After a single MM iteration, the algorithm snaps to the protrusion. This behavior is a result of the smoothness and large support of the shape divergence measure (Eq 13). Even protrusions that are improbable under the motion model can be stabilized by the likelihood.

B. Cortical Spreading Depression (CSD)

Optical intrinsic signal (OIS) imaging is a simple method of visualizing physiological processes without the use of dyes or tracers. Instead, OIS captures changes in the sample's intrinsic optical reflectance. Because of its simplicity and versatility, OIS is used for in vivo imaging applications in neuroscience. During CSD, there are large hemodynamic or blood-related, changes in cortical tissue [57, 58]. Conveniently, in cortical brain tissue under visible light, changes in blood volume and blood oxygen saturation constitute the majority of the OIS signal. The CSD wave appears as a *low contrast* brightening of the tissue, visible in inter-frame difference images, though due to the diffuse signal, is time-consuming for even a trained practitioner to trace.

Using the notation of section II-B, we treated the spatial extents of the CSD wave as foreground regions Ω_k in an image sequence $\{O_k\}_{k=0}^{K}$. We modeled each inter-frame difference $I_k = O_{k+1} - O_k$ image as a conditional Gaussian mixture like in section II-B. As prior information, the wavefront is known to travel at approximately 1 – 5 millimeters per minute. We incorporated this information by setting a prior mean wave speed of 3 millimeters per minute, with variance of 1 millimeters^2 per minute^2 (we set $a_\eta = b_\eta = 1$). As in the synthetic image sequence application, we used uninformative initial shape priors.

We tested our method on a real set of CSD images acquired from two separate experiments on separate C57Bl/6J mice. The mice were approved for experiments in accordance with University of California, Los Angeles Animal Research Committee Guidelines. The mice had their skulls exposed under anesthesia, and a rectangular section of the parietal bone (1 mm from the sagittal suture, temporal ridge, lambdoidal suture and coronal suture) was thinned to transparency. Burrholes were drilled proximal to the imaging field to allow for placement of stimulating electrodes. After allowing the animal to rest, the experimenter induced CSD by passing electric current through the stimulating electrodes. VGA-resolution (640 × 480) images were collected at a frequency of 1Hz, in 8-bit greyscale, under white light. The imaging field was approximately 3.2mm × 2.4mm, with each pixel representing approximately a 5μm × 5μm square. Before analysis, to save computation time, we rescaled each image to a quarter of its original size by bilinear interpolation (to 320 × 240).

CSD segmentation results are shown in Fig 7. For comparison, we also provide the results of segmentation done with graph cuts in the absence of shape prior [49]. Our tracking method is able to regularize against the noise and random heterogeneity that is typical of these in-vivo experiments. In the absence of a prior, the interface location is not as well defined, and graph cuts fails to select the entire CSD region.

In Fig 8, we examine the results of our model under non-ideal conditions. In this experiment, biological movement has caused the presence of a vascular distraction in the difference images. The images in the top row show frame-byframe segmentation in the absence of shape priors, where the vascular system is causing formidable interference. Using our method, we achieve increasingly better segmentations when increasing the shape mismatch penalty α, though $\alpha = 2$ still works reasonably well. Increasing α increases the weight of the speed-based regularization relative to the likelihood. This parameter offers flexibility for tuning the method for data sets with increasing random heterogeneity.

C. Wound healing assays

Another application we explore is the tracking of a collapsing boundary. One specific example of this type of system arises in *in vitro* assays of wound healing. In the typical wound healing assay, a layer of cells is grown to confluency on a substrate. A portion of the cell layer is then removed, either by scratching, lifting off a localized region of cells, or removing a constraint confining the monolayer [59]. The dynamics of how the cells refill the bare substrate is then studied typically with bright or dark-field light microscopy under a variety of physical and chemical conditions. While the biological process is complex, involving chemical signaling pathways, mediated by mechanical interactions with the elastic substrate and neighboring cells [60], the main observable is the moving wound edge.

Both cell migration and cell proliferation can occur, [61], and as more quantitative studies of these types of assays emerge [62], tracking the spatio-temporal dynamics of cells and their proliferation will become critical. In response to a wound or free space in which to migrate, the cells near the edge increase their motility to attempt to cover the wound. The motion of the wound boundary appears to be largely monotonic. However, without additional experimental imaging modalities such as fluorescent labeling of membranes, accurate tracking of the wound edge can be difficult due to low contrast, boundaries with other cells, and extraneous material (such as floating dead cells) in the image field.

Our statistical approach for tracking the moving boundary of a cell monolayer was tested on an *in vitro* wound healing assay. Fig 9 shows segmentations on a series of sobel filtered [54] images of a shrinking circular wound that was induced in a monolayer of epithelial cells. Pre-wounding, the epithelial cells were grown to confluency on extracellular matrix substrate. Bright-field images were then taken of the epithelial cells migrating to fill the circular wound region. In these images, the wound region and the healed region are distinguished by the lack and presence of cells. Cells in the field appear dark at their boundaries, and bright in their bodies. In this application, we treat the wounded region as Ω. As the wound heals, it undergoes robust changes in shape. Our segmentation method accurately tracked the boundary of the wound. For these segmentations, where we knew the location of the initial wound circle, we used an informative shape prior with the known reference shape, and set $\tau^2 = 1$. We used a weakly informative prior mean speed centered at −4 pixels/frame. In this application, the speed of the front V is strictly negative.

D. Validation

We evaluated the accuracy of our segmentation method by comparing results from our method against the results of human-assisted segmentations for the synthetic image sequence (Fig 4) and the CSD image sequence (Fig 7). In both cases, we compute the mismatch between two segmentations

$$\text{Error}=\frac{\text{number of mismatching pixels}}{\text{mean boundary lenght of segmentations}}. \quad (26)$$

The resulting quantity has an interpretation as the average thickness in pixels of the mismatched region.

For the synthetic image sequence, where the ground truth is available, we compared the error of segmentations made using our method to error from human-assisted segmentations. Fig 10 depicts the mismatch from ground truth for segmentations on the synthetic image sequence of Fig 4. For this data set, it is clear that our segmentation method outperforms human segmentation, by approximately a factor of two.

For the cortical spreading depression images of Fig 7, where the ground truth is not available, we compared human segmentation to our automated segmentations (Fig 11). Our results are in good agreement with manually-segmented results, good to within 1.7 ± 0.5 pixels. By comparison, the humans segmentations disagreed with each other by 1.4 ± 0.4 pixels. These data demonstrate the ability of our method to perform as-well-as or better than manual segmentation.

IV. Discussion

We have demonstrated a framework for simultaneous segmentation and inference of the dynamics of monotonically traveling boundaries in image sequences. Our method is simple, generalizable, and easily extended to a wide variety of applications. The novelty of our method is in bringing level-set based interface modeling into a statistical estimation framework, where inference can occur using graph cuts. In the process, we have developed a theory for including shape priors into the graph cuts method using an MM algorithm.We have demonstrated the efficacy of our method in solving the boundary tracking problem for two unrelated biomedical applications, and is able to recover a ground truth speed pattern for a synthetically generated image sequence.

Since we developed our method from the ground up with statistical theory in mind, it can be easily extended or modified to suit a wide range of applications. Being modular in design, it is easy to alter particular components of our method while leaving others unchanged. For instance, one may choose to modify the shape distance we introduce in Eq 13, to suit a particular application. There is much prior work on shape distances in the level set literature, and the graph cut segmentation community would greatly benefit from more research on how to incorporate shape knowledge.

Since our method is model-based, it is fairly easy to predict when performance suffers. In particular, when the image pixel intensities are poorly described by the likelihood model of section II-B, or when the speed of the interface is poorly modeled as the smooth Gaussian process of section II-A. In many cases, success of the method depends on the choice of shape penalty exponent a. In some sense, a controls the degree of regularization done by the recursive shape priors. If a is set too large, the segmenter will tend to favor the predicted interface positions over the data, ignoring real deformations in the interface. conversely, if a is set too small, the regularization may not be sufficient, and the segmenter is more likely to pick up noise in the image. For example, in Fig 8, increasing the shape penalty parameter to $a = 4$ improved the tracking of the interface in the presence of distractors. In general, $a = 2$ worked well for our examples.

Perhaps the aspect of our method that would benefit the most from future research would be the spatial interpolation method we use to sample fluctuations in interface speed. The Gaussian process model, while theoretically clean, is computationally complex. Yet, the Bayesian nature of this method allows for the application of natural Bayesian model evaluation and inference methods. In particular, future development will focus on Bayesian model selection of competing speed models. Finally, we would like to note that extension of our method to non-monotonic movement is theoretically straightforward. One need only implement the time-dependent Gaussian process interpolation of Sahu et al. [46] to predict speeds, and use the level-set equation rather than the eikonal equation to describe the motion.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

We thank Lydia L. Shook for performing the CSD experiments, Berta sandberg for her input in the area of segmentation, and Tanye Y. Tang for helping to adapt the Vaswani et al. [23] Matlab code for use on the image sequences in this manuscript. We are grateful to Prof. c.-L. Guo, caltech Bioengineering for the wound healing assay images. JC acknowledges support by grant mumber T32GM008185 from the National Institute of General Medical Sciences. JC and TC also acknowledge support from the the National Science Foundation through grants DMS-1032131 and DMS-1021818, and from the Army Research Office through grant 58386MA. KCB acknowledges support from the National Institutes of Health (NINDS NS059072 and NS070084) and the Department of Defense (CDMRP PR100060).

References

1. Fast V, Kléber A. Role of wavefront curvature in propagation of cardiac impulse. Cardiovascular research. 1997; 33(2):258. [PubMed: 9074688]
2. Hogea C, Murray B, Sethian J. Simulating complex tumor dynamics from avascular to vascular growth using a general level-set method. journal of mathematical biology. 2006; 53(1):86–134. [PubMed: 16791651]
3. Chien D, Kwong K, Gress D, Buonanno F, Buxton R, Rosen B. MR diffusion imaging of cerebral infarction in humans. American journal of neuroradiology. 1992; 13(4):1097. [PubMed: 1636519]
4. Cornell-Bell A, Finkbeiner S, Cooper M, Smith S. Glutamate induces calcium waves in cultured astrocytes: long-range glial signaling. Science. 1990; 247(4941):470. [PubMed: 1967852]
5. Leao A. Spreading depression of activity in the cerebral cortex. journal of Neurophysiology. 1944; 7(6):359.
6. Sethian J. A fast marching level set method for monotonically advancing fronts. Proceedings of the National Academy of Sciences of the United States of America. 1996; 93(4):1591. [PubMed: 11607632]
7. Sethian, JA. Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science. Cambridge Univ Pr; 1999.
8. Osher S, Sethian J. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. journal of computational physics. 1988; 79(1):12–49.
9. Malladi R, Sethian J, Vemuri B. Shape modeling with front propagation: A level set approach. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 1995; 17(2):158–175.
10. Macklin P, Lowengrub J. An improved geometry- aware curvature discretization for level set methods: application to tumor growth. journal of Computational Physics. 2006; 215(2):392–401.
11. Zheng X, Wise S, Cristini V. Nonlinear simulation of tumor necrosis, neo-vascularization and tissue invasion via an adaptive finite-element/level-set method. Bulletin of mathematical biology. 2005; 67(2):211–259. [PubMed: 15710180]

12. Sermesant M, Konukoglu E, Delingette H, Coudière Y, Chinchapatnam P, Rhode K, Razavi R, Ayache N. An anisotropic multi-front fast marching method for real-time simulation of cardiac electrophysiology. Functional Imaging and Modeling of the Heart. 2007:160–169.
13. Wolgemuth C, Zajac M. The moving boundary node method: A level set-based, finite volume algorithm with applications to cell motility. journal of computational physics. 2010; 229(19): 7287–7308. [PubMed: 20689723]
14. Paragios N, Deriche R. Geodesic active contours and level sets for the detection and tracking of moving objects. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 2000; 22(3): 266–280.
15. Tang J, Acton S. Vessel boundary tracking for intravital microscopy via multiscale gradient vector flow snakes. Biomedical Engineering, IEEE Transactions on. 2004; 51(2):316–324.
16. Zhong Y, Jain A, Dubuisson-Jolly M. Object tracking using deformable templates. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 2000; 22(5):544–549.
17. Mansouri A. Region tracking via level set pdes without motion computation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2002:947–961.
18. Baumberg, A., Hogg, D. Motion of Non-Rigid and Articulated Objects, 1994., Proceedings of the 1994 IEEE Workshop on. IEEE; 1994. An efficient method for contour tracking using active shape models; p. 194-199.
19. Chen Y, Huang T, Rui Y. Parametric contour tracking using unscented kalman filter. Image Processing 2002 Proceedings 2002 International Conference on. 2002; 3:613–616. Ieee.
20. Peterfreund N. Robust tracking of position and velocity with kalman snakes. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 1999; 21(6):564–569.
21. Isard M, Blake A. Contour tracking by stochastic propagation of conditional density. Computer VisionECCV'96. 1996:343–356.
22. Li P, Zhang T, Pece A. Visual contour tracking based on particle filters. Image and Vision Computing. 2003; 21(1):111–123.
23. Vaswani N, Rathi Y, Yezzi A, Tannenbaum A. Deform pf-mt: particle filter with mode tracker for tracking nonaffine contour deformations. Image Processing, IEEE Transactions on. 2010; 19(4): 841–857.
24. Doucet A, Godsill S, Andrieu C. On sequential Monte Carlo sampling methods for Bayesian filtering. Statistics and computing. 2000; 10(3):197–208.
25. Liu, J. Monte Carlo strategies in scientific computing. Springer Verlag; 2008.
26. Mumford D, Shah J. Boundary detection by minimizing functionals. Image understanding. 1988
27. Zhu S, Yuille A. Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 2002; 18(9):884–900.
28. Chan T, Vese L. Active contours without edges. IEEE Transactions on image processing. 2001; 10(2):266–277. [PubMed: 18249617]
29. Ford L, Fulkerson D. Flows in networks. 1962
30. Freedman D, Drineas P. Energy minimization via graph cuts: Settling what is possible. 2005
31. Kolmogorov V, Zabih R. What energy functions can be minimized via graph cuts? IEEE Transactions on Pattern Analysis and Machine Intelligence. 2004:147–159. [PubMed: 15376891]
32. Veksler O. Star shape prior for graph-cut image segmentation. Computer Vision–ECCV 2008. 2008:454–467.
33. Slabaugh G, Unal G. Graph cuts segmentation using an elliptical shape prior. Image Processing, 2005 ICIP 2005 IEEE International Conference on. 2005; 2 IEEE.
34. Das P, Veksler O, Zavadsky V, Boykov Y. Semiautomatic segmentation with compact shape prior. Image and Vision Computing. 2009; 27(1–2):206–219.
35. Malcolm J, Rathi Y, Tannenbaum A. Graph cut segmentation with nonlinear shape priors. Image Processing, 2007 ICIP 2007 IEEE International Conference on. 2007; 4 IEEE.
36. Dambreville S, Rathi Y, Tannenbaum A. A framework for image segmentation using shape models and kernel space shape priors. IEEE transactions on pattern analysis and machine intelligence. 2008:1385–1399. [PubMed: 18566493]

37. Freedman D, Zhang T. Interactive graph cut based segmentation with shape priors. Computer Vision and Pattern Recognition, 2005 CVPR 2005 IEEE Computer Society Conference on. 2005; 1:755–762. IEEE.

38. Vu, N., Manjunath, B. Computer Vision and Pattern Recognition, 2008 CVPR 2008 IEEE Conference on. IEEE; 2008. Shape prior segmentation of multiple objects with graph cuts; p. 1-8.

39. Chang, H., Yang, Q., Parvin, B. Computer Vision and Pattern Recognition, 2008 CVPR 2008 IEEE Conference on. IEEE; 2008. A bayesian approach for image segmentation with shape priors; p. 1-8.

40. Zhu-Jacquot, J., Zabih, R. Signal-Image Technologies and Internet-Based System, 2007 SITIS'07 Third International IEEE Conference on. IEEE; 2007. Graph cuts segmentation with statistical shape priors for medical images; p. 631-635.

41. Cremers D. Dynamical statistical shape priors for level set-based tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2006:1262–1273. [PubMed: 16886862]

42. Aldridge D, Oldenburg D. Two-dimensional tomographic inversion with finite-difference traveltimes. Journal of Seismic Exploration. 1993; 2(25):7–274.

43. Rue H, Tjelmeland H. Fitting Gaussian Markov random fields to Gaussian fields. Scandinavian journal of Statistics. 2002; 29(1):31–49.

44. Hartman L, Hössjer O. Fast kriging of large data sets with Gaussian Markov random fields. Computational Statistics & Data Analysis. 2008; 52(5):2331–2349.

45. O'Hagan, A., Forster, J., Kendall, M. Bayesian inference. Arnold; 2004.

46. Sahu S, Yip S, Holland D. A fast Bayesian method for updating and forecasting hourly ozone levels. Environmental and Ecological Statistics. 2009:1–23.

47. Cressie N. Fitting variogram models by weighted least squares. Mathematical geology. 1985; 17(5):563–586.

48. Genz A. Numerical computation of multivariate normal probabilities. journal of computational and graphical statistics. 1992; 1(2):141–149.

49. El Zehiry, N., Xu, S., Sahoo, P., Elmaghraby, A. The Seventh IASTED International Conference on Visualization, Imaging and Image Processing. ACTA Press; 2007. Graph cut optimization for the Mumford-Shah model; p. 182-187.

50. Cremers D, Osher S, Soatto S. Kernel density estimation and intrinsic alignment for shape priors in level set segmentation. International journal of Computer Vision. 2006; 69(3):335–351.

51. Hunter D, Lange K. A tutorial on MM algorithms. The American Statistician. 2004; 58(1):30–37.

52. Boykov Y, Kolmogorov V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 2004; 26(9):1124–1137.

53. Ripley, B. Stochastic simulation. Vol. 21. Wiley Online Library; 1987.

54. Abramoff M, Magelhaes P, Ram S. Image processing with ImageJ. Biophotonics international. 2004; 11(7):36–42.

55. Arndt, H., Bundschus, M., Naegele, A. 2009 33rd Annual IEEE International Computer Software and Applications Conference. IEEE; 2009. Towards a next-generation matrix library for Java; p. 460-467.

56. Schindelin J. Fiji is just ImageJbatteries included. Proceedings of the ImageJ User and Developer Conference, Luxembourg. 2008

57. Charles A, Brennan K. Cortical spreading depression new insights and persistent questions. Cephalalgia. 2009; 29(10):1115. [PubMed: 19735537]

58. Chang J, Shook L, Biag J, Nguyen E, Toga A, Charles A, Brennan K. Biphasic direct current shift, haemoglobin desaturation and neurovascular uncoupling in cortical spreading depression. Brain. 2010; 133(4):996. [PubMed: 20348134]

59. Nikolic DL, Boettiger AN, Bar-Sagi D, Carbeck JD, Shvartsman SY. Role of boundary conditions in an experimental model of epithelial wound healing. Am J Physiol Cell Physiol. 2006; 291:C68–C75. [PubMed: 16495370]

60. DiMilla PA, Barbee K, Lauffenburger DA. Mathematical model for the effects of adhesion and mechanics on cell migration speed. Biophys J. 1991; 84:2907–2918.

61. Poujade M, Grasland-Mongrain E, Hertzog A, Jouanneau J, Chavrier P, Ladoux B, Buguin A, Silberzan P. Collective migration of an epithelial monolayer in response to a model wound. Proc Natl Acad Sci USA. 2007; 104:15988–15993. [PubMed: 17905871]

62. Posta F, Chou T. A mathematical model of intercellular signaling during epithelial wound healing. journal of Theoretical Biology. 2010; 266(1):70–78. [PubMed: 20685318]

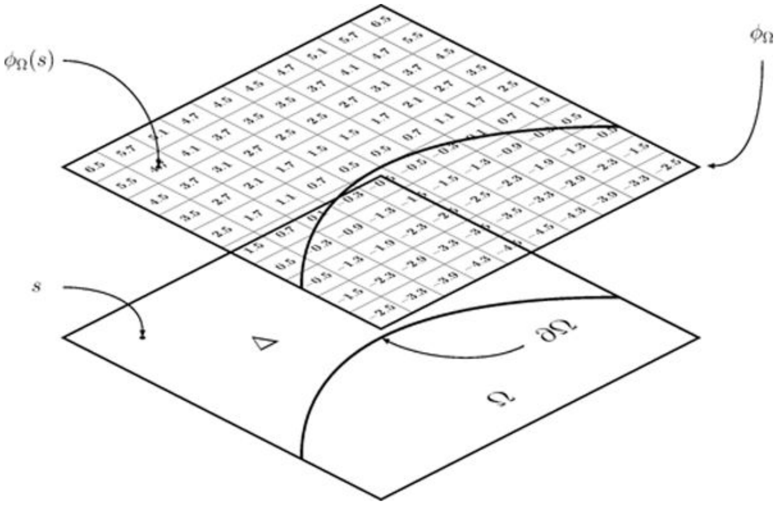

Fig. 1. Level-set embedding of shapes

Embedding of a region Ω into a signed distance function ϕ_Ω defined on a discrete lattice. ϕ_Ω takes values according to the signed Euclidean distance from the boundary ∂Ω, with negative values inside Ω. The boundary ∂Ω is implicitly embedded as the zero-level set of ϕ_Ω. Shown is a 2-d lattice, however, level set method works for $\mathbb{R}^d$ with any arbitrary *d*.

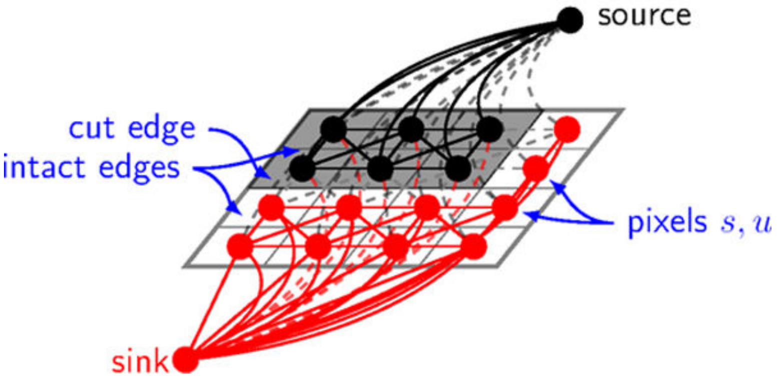

Fig. 2. Embedding of an image into a graph

In the graph cuts framework, pixels are nodes in a graph. Connections between neighboring pixels are made, as well as connections between each pixel and two special nodes called the source (foreground) and sink (background). Depicted is an eight-neighbor system, where s and u are neighbors. These connections are weighted according to the strength of the association between two nodes to the same class (either foreground or background). A segmentation is found by cutting the graph into two parts separating the source and sink such that the edge weights along the cut are minimal. The segmented foreground then consists of the pixels that have intact edges with the source (depicted as solid black lines).

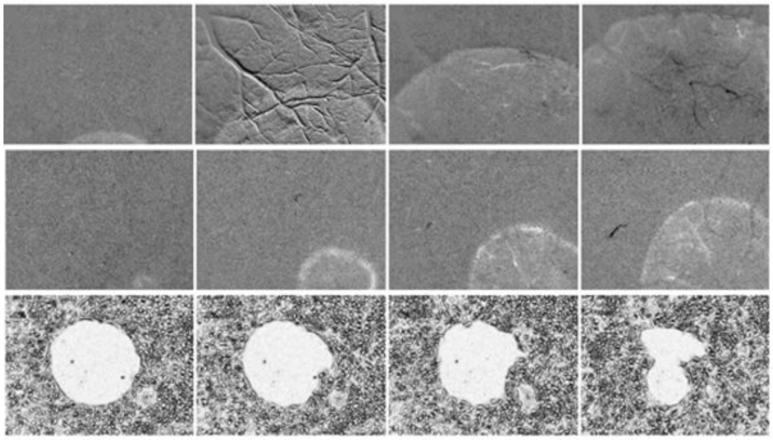

Fig. 3. Examples of monotonic boundary movement

Shown are stills from three image sequences depicting monotonic boundary motion. Top two rows are cortical spreading depression hemodynamical waves. Bottom row is a wound undergoing healing. The noise characteristics in these images, and the shapes that develop, differ markedly.

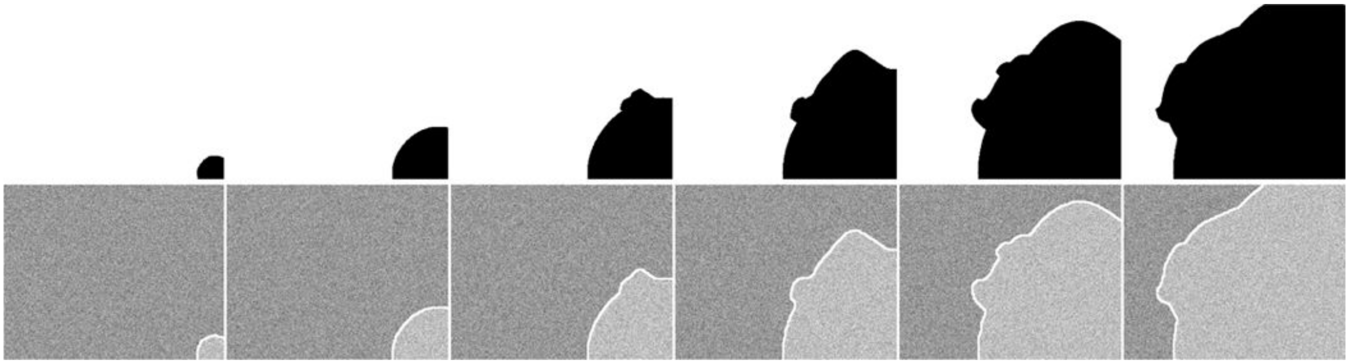

Fig. 4. Segmentation of a synthetic image sequence

(top) Mask of ground truth wave sequence, with a growing interior region shown **in black. The spatial field is of size** 320 × 240 pixels. The speed of the front varies between 6 and 10 pixels/frame (shown in fig 5), producing topological changes **in the interface.** ***(bottom)*** **Segmentation** of images where noise has been added to the ground truth. The image intensity was 0 ± 3 in the exterior and 3 ± **3 in the interior.** ***δ*****t = 7 frames.**

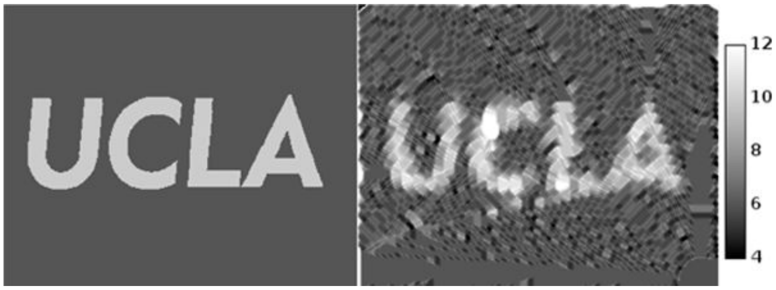

Fig. 5. Recovery of interface speed

(left) Ground truth speed field. *(right)* Reconstructed speed field. Reconstruction of synthetic wave speed using the segmentations shown in figure 4, and second-order upwind finite differences. The ground-truth speed of the interface is 10 pixels/frame as it passes through the UCLA letters, and 6 pixels/frame outside of the letters. The reconstructed speed field has an estimated speed of 9.8 ± 1.5 pixels/frame inside the letters, and 6.0 ± 0.7 pixels/frame outside. Velocity scale shown at right in pixels/frame.

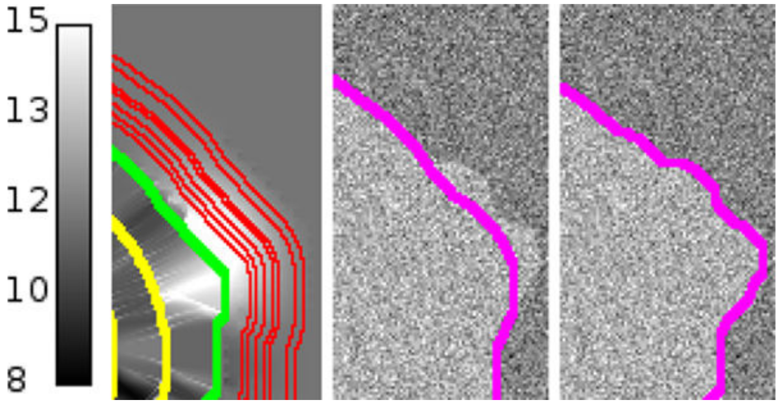

Fig. 6. Updating predictions with new data

(left) Past (yellow), current (green), and predicted future (red) interface positions drawn over mean predicted speed field. Predictions of future interface positions, which act as shape priors, are made using samples from our stochastic speed model. To the left of the green boundary are speeds interpolated from the collection of past interface positions. To the right are interface positions found by propagating the green contour with speeds sampled from the GMRF model. Speed scale shown at left is pixels/frame. *(middle)* When a new noisy image is acquired, the MM algorithm is initialized at the position obtained by propagating the previous interface position according to its estimated mean speed field. The true boundary deviates from the mean predicted boundary because of developing protrusions. The mean predicted boundary is calculated by propagating the green boundary against the mean speeds given in Eq 8. *(right)* After a single MM iteration, the protrusions are found. Due to the continuous nature of the kernel density shape prior, our method is able to account for large deformations.

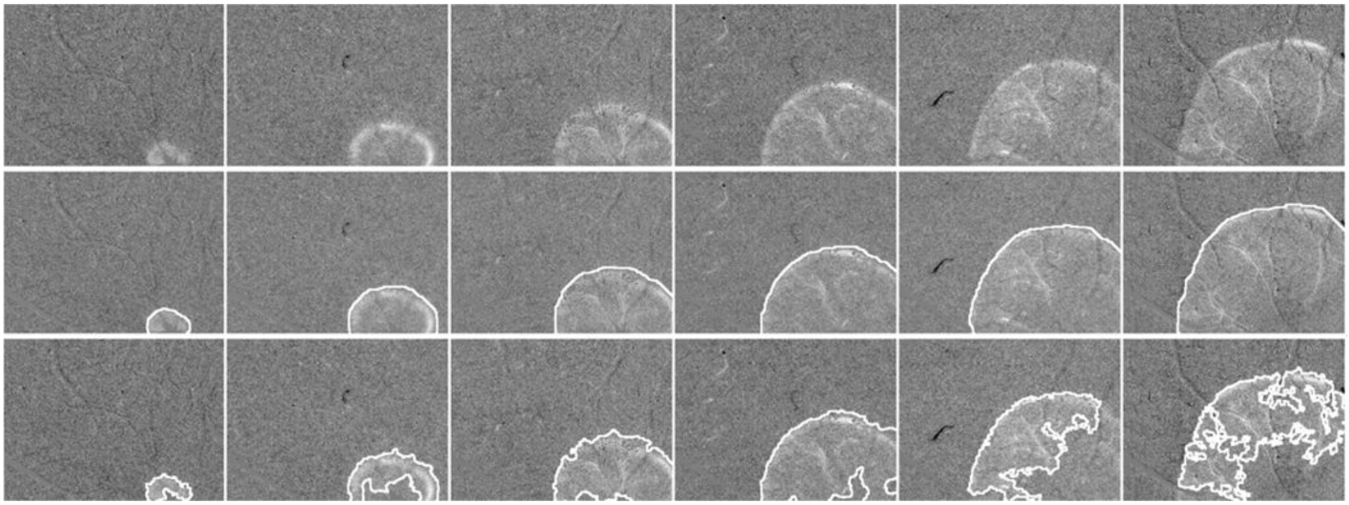

Fig. 7. Segmentation of real *in-vivo* CSD data

(Left→ Right) CSD shown propagating, $\delta t = 7$s. *Top:* Original unsegmented inter-frame differences showing CSD-related changes in blood signal. *Second row:* Results from our segmentation method, where we track the moving front probabilistically. *Third row:* Segmentation of the spreading region done without shape priors.

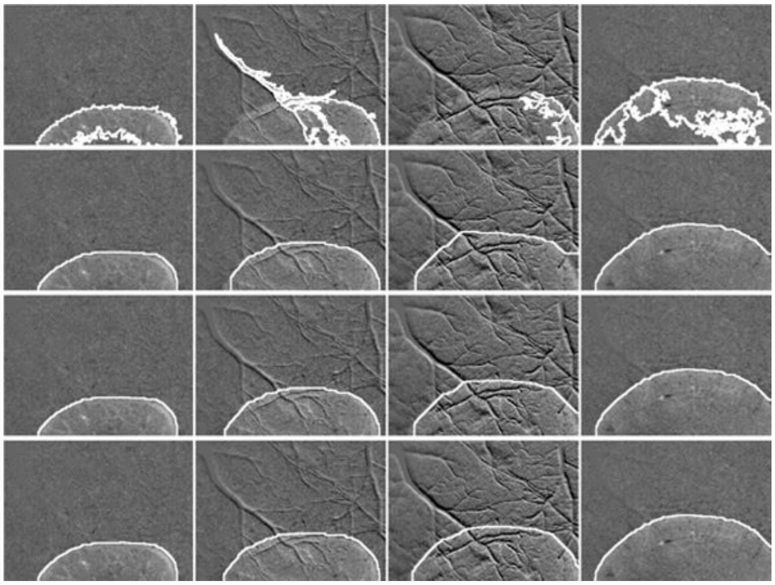

Fig. 8. Adjusting regularization by adjusting the shape penalty parameter

(Left→Right) and (Top→ Bottom) Biological movement during imaging causes artifacts in the difference image. Failure to adjust for the movement results in less than ideal data. Increasing α, the shape mismatch penalty, can compensate for poorly acquired image data. Our method is able to track the moving front even as it is partially occluded. *Top:* segmentation without shape prior. *Second row* $\alpha = 2$. *Third row:* $\alpha = 3$. *Fourth row:* $\alpha = 4$. $\delta t = 2$s.

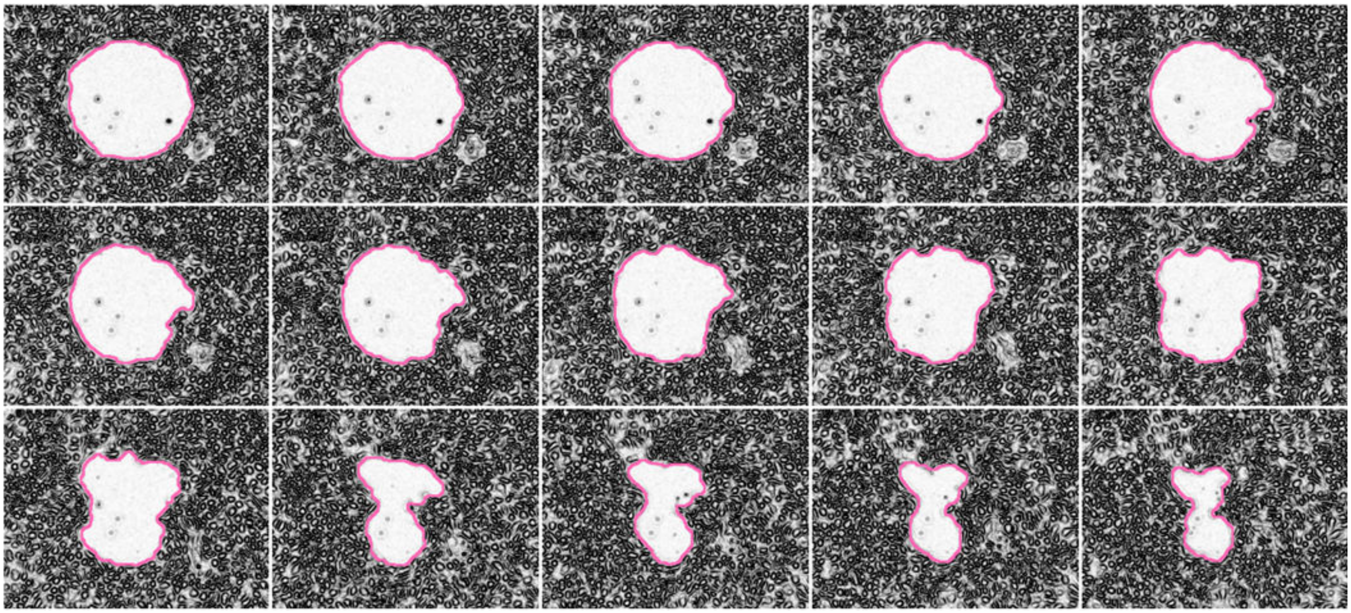

Fig. 9. Segmentation of wound healing assays featuring robust shape changes

(Left→Right) and (Top→ Bottom) Wound-healing time stills. Segmentation performed on the sobel filter (shown) of the original image sequence, which is well-modeled by the Gaussian mixture of section II-B. In this application, the boundary is moving inward, while the shape of the inner region is undergoing large changes. The segmentation method works well even for non-convex shapes. $\delta t = 30$ min. We are grateful to Prof. C.-L. Guo, Caltech Bioengineering for these images. Resolution: 320 × 240.

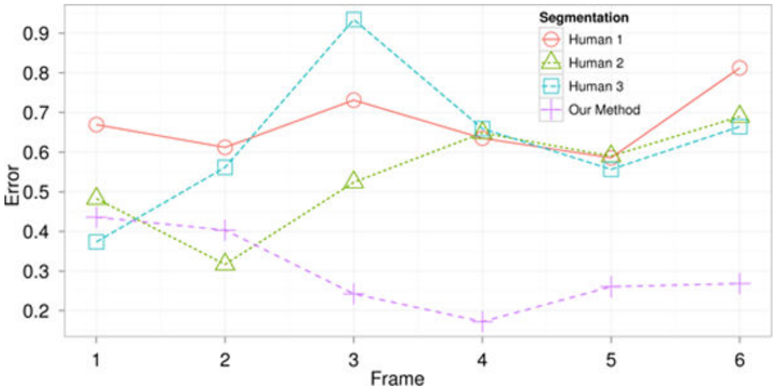

Fig. 10. Comparison of results against laborious manual human segmentation of synthetic images

The accuracy of our method's segmentation of the synthetic image sequence (for the six frames shown in Fig 4) compared to the accuracy of three humans. The *error* plotted in the y-axis is the average number of misclassified pixels per boundary-length, where boundary-length is the average of the lengths of the ground truth and segmented boundary. Our method differed from the ground truth by 0.30 ± 0.10 pixels. The humans performed significantly worse with error of 0.61 ± 0.14 pixels.

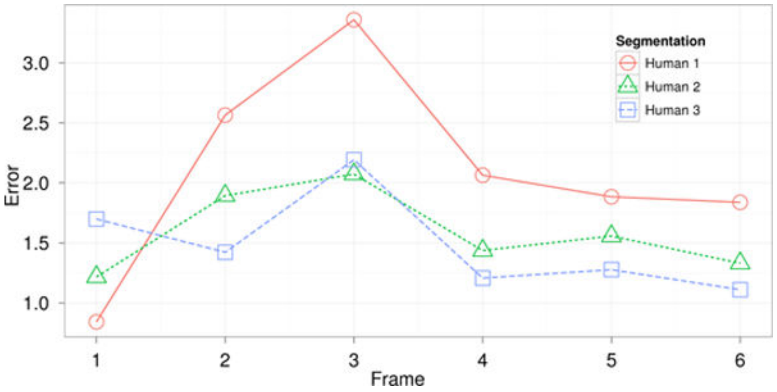

Fig. 11. Comparison of results against laborious manual human segmentation of in-vivo CSD images

Deviation of human segmentations from the results of our automated approach for the image sequence shown in Fig 7. The results from our segmentation method agreed with the results from the manual segmentations to within 1.7 ± 0.6 pixels.