CELL TRACKING USING PARTICLE FILTERS WITH IMPLICIT CONVEX SHAPE MODEL IN 4D CONFOCAL MICROSCOPY IMAGES

Nisha Ramesh[†⋆] *Tolga Tasdizen* [†⋆]

[†] Department of Electrical and Computer Engineering, University of Utah, United States
[⋆] Scientific Computing and Imaging Institute, University of Utah, United States

ABSTRACT

Bayesian frameworks are commonly used in tracking algorithms. An important example is the particle filter, where a stochastic motion model describes the evolution of the state, and the observation model relates the noisy measurements to the state. Particle filters have been used to track the lineage of cells. Propagating the shape model of the cell through the particle filter is beneficial for tracking. We approximate arbitrary shapes of cells with a novel implicit convex function. The importance sampling step of the particle filter is defined using the cost associated with fitting our implicit convex shape model to the observations. Our technique is capable of tracking the lineage of cells for nonmitotic stages. We validate our algorithm by tracking the lineage of retinal and lens cells in zebrafish embryos.

Index Terms— Bayesian methods, particle filter, zebrafish, implicit functions.

1. INTRODUCTION

Studying the evolution of cells is necessary to improve our understanding of their growth and behavior. Therefore, tracking cells in time has become an important problem in biology. Existing methods for 3D tracking are elucidated in [1], but there is a need for automatic methods with increased robustness for arbitrary cell shapes and careful validation. Many cell tracking algorithms have been proposed and reported in the literature, [2]. Specific tracking tasks require task-specific algorithms, or combinations of algorithms, for efficacy [3].

Bayesian methods are ubiquitous for tracking various objects. Among bayesian methods, we focus on the particle filter for our application. Particle filters infer the hidden states of a dynamic system $x_{1:t} = \{x_1, x_2....x_t\}$, using a sequence of noisy observations $z_{1:t} = \{z_1, z_2....z_t\}$ [4]. Particle filter methods, along with variational methods, have been used to track spherical and elliptical objects [5, 6, 7, 8]. In variational methods, the hidden states are estimated by optimizing a cost function that depends on the observation using standard optimization algorithms [9]. There is a wide variation in cell shapes; hence, having a robust and generic model that can adapt to the shape of the cell is essential.

In this paper, we illustrate a novel shape model that can track cells with varying shapes when used in the particle filter framework. Our proposed model is an implicit convex function that can represent generic convex shapes, hence providing more flexibility in comparison to the elliptical and spherical models. We start with a convex polytope that is an intersection of N half-spaces. We convert the implicit convex function into a differentiable model by: 1) replacing the intersection of half spaces as a product of perceptron equations and 2) relaxing the perceptrons used in representing the half-spaces to logistic sigmoid functions. We then define a cost function that aligns the gradient of this function with the image gradient. Since the model is differentiable, gradient-based optimization algorithms can be used to find the model parameters. This shape model is embedded in the particle filter. The importance sampling step of the particle filter uses the fitting cost associated with our implicit convex shape model. The weights are derived from the optimized energy values of the cost function.

The paper is organized as follows: The methods section begins with a description of the implicit convex shape model in Section 2.1. Following that, we give a general introduction to particle filters and our tracking algorithm in Section 2.3 and Section 2.4, respectively. The algorithm has been used to track retinal and lens cells in zebrafish embryos, and the results are discussed in Section 3. We also compare our tracking algorithm with a basic spherical model to show the importance of modeling arbitrary cell shapes. We finally conclude and discuss future work in Section 4.

2. METHOD

2.1. Implicit Convex Shape Model

Shapes can be represented by their characteristic function $f : \mathbf{R}^n \rightarrow B$ where $B = \{0, 1\}$. For convex shapes, the function $f(\mathbf{x})$ can be approximated as the intersection $f(\mathbf{x}) = \cap_{i=1}^{N} H_i$

This work is supported by NIH grant 1R01-GM098151-01,"Fluorender: An Imaging Tool for Visualization and Analysis of Confocal Data as Applied to Zebrafish Research". We would like to thank Kristen Kwan from the Department of Human Genetics, University of Utah, for providing the dataset and ground truth for tracking.

of N half-spaces $H_i = \{x \in R^n : h_i(\mathbf{x}) > 0\}$ [10]. The half-space, in arbitrary dimensions, is defined using the perceptron equation

$$h_i(\mathbf{x}) = \begin{cases} 1, & \sum_{k=1}^{n} w_{ik} x_k \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where w_{ik} are the weights. To obtain a differentiable model, we first note that the conjunction of binary variables $\bigwedge_{i=1}^{N} h_i(\mathbf{x})$ is equivalent to the product $\prod_{i=1}^{N} h_i(\mathbf{x})$. We then approximate the binary perceptrons $h_i(\mathbf{x})$ with logistic sigmoid functions

$$\sigma_i(\mathbf{x}) = \frac{1}{1 + e^{-\sum_{k=0}^{n} w_{ik} x_k}} \quad (2)$$

The resulting approximation to the shape function is then given as

$$f(\mathbf{x}) = \prod_{i=1}^{N} \sigma_i(\mathbf{x}), \quad (3)$$

where $\mathbf{x} = \{x, y, z, 1\}$ for 3D shapes. Note that the input vector is appended by one to represent the bias. The level set $f(x) = 0.5$ is taken to represent the interface between the foreground and background regions, i.e., the shape of interest.

2.2. Parameter initialization

The parameters are initialized interactively using inputs from the user. The user defines the initial center $\{c_x, c_y, c_z\}$ by clicking the approximate centroid of the cell to be tracked. Using this centroid, the cells are approximated as spheres with a fixed radius. We fix the radius r at 5 pixels for initialization. This approximation is obtained by choosing the parameters as

$$w_{ik} = \begin{cases} \cos(\theta_p)\sin(\phi_q), & k = 0 \\ \sin(\theta_p)\sin(\phi_q), & k = 1 \\ \cos(\phi_q), & k = 2 \\ -(r + c_x\cos(\theta_p)\sin(\phi_q) + c_y\sin(\theta_p)\sin(\phi_q) \\ +c_z\cos(\phi_q)), & k = 3 \end{cases}$$

for varying values of θ_p, ϕ_q. We choose $\theta_p = \frac{\pi}{4}p$ and $\phi_q = \frac{\pi}{4}q$ for $p = [1 \cdots P]$ and $q = [1 \cdots Q]$. By using different combinations of θ_p and ϕ_q, we get parameters representing different planes. This initialization creates a model (3) with $N(P \times Q) = 18$ sigmoid functions.

2.2.1. Energy Minimization

To match the model to the underlying image, the gradient of the function ∇f should be in the direction of the normal vector to the image. This constraint can be formulated as

$$E_g(W) = \int_{\Omega} \left[(\nabla f \cdot N - 1)^2 + (\nabla f \cdot T)^2\right] |\nabla I|^2 d\mathbf{x} \quad (4)$$

where $N = \nabla I / |\nabla I|$ represents the unit normal vector computed from the image gradient, T represents the unit tangent vectors perpendicular to the normal, and Ω represents the image domain. The second energy term ensures that the object of interest is specifically represented by the level set $f(\mathbf{x}) = 0.5$:

$$E_b(W) = \int_{\Omega} (f - 0.5)^2 |\nabla I|^2 d\mathbf{x} \quad (5)$$

Note that the constraints given by (4) and (5) cannot be satisfied everywhere in Ω due to the limited representation capacity of the model given in (3); however, the multiplication with $|\nabla I|^2$ weights the penalty functions heavier where there is a strong image gradient and weakens the penalty functions where there is no strong image gradient. Hence, the model eliminates the need for explicit boundary detection before fitting the model. The same type of energy function was used in [11] to fit implicit polynomials to image data. Finally, the combined energy is given as

$$E = E_g + \alpha E_b \quad (6)$$

We fit the model to the data by minimizing this energy using gradient descent. First we need to compute the gradient of f:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{bmatrix} = \begin{bmatrix} -f(\mathbf{x}) \sum_{i=1}^{N} (1 - \sigma_i(\mathbf{x})) w_{i0} \\ -f(\mathbf{x}) \sum_{i=1}^{N} (1 - \sigma_i(\mathbf{x})) w_{i1} \\ -f(\mathbf{x}) \sum_{i=1}^{N} (1 - \sigma_i(\mathbf{x})) w_{i2} \end{bmatrix} \equiv \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} \quad (7)$$

We demonstrate the computation of the derivative of the energy with respect to the parameter w_{ik} :

$$\frac{\partial E}{\partial w_{ik}} = \Big[-(g'_x n_x + g'_y n_y + g'_z n_z) + (g_x g'_x + g_y g'_y + g_z g'_z) + 2(f(\mathbf{x}) - 0.5)\Big(- x_k f(\mathbf{x})(1 - \sigma_i(\mathbf{x}))\Big)\Big] |\nabla I|^2, \quad (8)$$

where $g'_x = \frac{\partial g_x}{\partial w_{ik}}$, $g'_y = \frac{\partial g_y}{\partial w_{ik}}$ and $g'_z = \frac{\partial g_z}{\partial w_{ik}}$. The partial derivative of g_x with respect to w_{ik} is found as

$$g'_x(\mathbf{x}) = \frac{\partial}{\partial w_{ik}}[-f(\mathbf{x})h(\mathbf{x})] = -[f'(\mathbf{x})h(\mathbf{x}) + h'(\mathbf{x})f(\mathbf{x})], \quad (9)$$

where we define $h(\mathbf{x}) = \sum_{i=1}^{N} (1 - \sigma_i(\mathbf{x})) w_{i0}$ to match (7). Solving for $f'(\mathbf{x})$ and $h'(\mathbf{x})$ we get

$$f'(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial w_{ik}} = \frac{\partial}{\partial w_{ik}} \prod_{i=1}^{N} \sigma_i(\mathbf{x}) = -x_k f(\mathbf{x})(1 - \sigma_i(\mathbf{x}))$$

$$h'(\mathbf{x}) = \frac{\partial h(\mathbf{x})}{\partial w_{ik}} = \frac{\partial}{\partial w_{ik}} \sum_{i=1}^{N} (1 - \sigma_i(\mathbf{x})) w_{i0} = \begin{cases} (1 - \sigma_i(\mathbf{x}))(1 + w_{ik} x_k \sigma_i(\mathbf{x})), k = 0, \\ w_{ik} x_k \sigma_i(\mathbf{x})(1 - \sigma_i(\mathbf{x})), k \neq 0 \end{cases}$$

These results can be substituted into (9) to compute g'_x. The partial derivatives of g_y and g_z with respect to w_{ik} are calculated in a similar manner by replacing the function $h(\mathbf{x})$ with

$\sum_{i=1}^{N}(1-\sigma_i(\mathbf{x}))w_{i1}$ and $\sum_{i=1}^{N}(1-\sigma_i(\mathbf{x}))w_{i2}$, respectively. Then g'_x, g'_y and g'_z are substituted into (8) to compute $\frac{\partial E'}{\partial w_{ik}}$. Four implicit convex shape models fitted to image data in this manner are shown in Figure (1). We observe that cell shape can be significantly different from a basic spherical or elliptical model.

(a) (b) (c) (d)

Fig. 1: 3D representation of cells.

2.3. Particle Filter - Generic Framework

In this section, we give an overview of the particle filter framework. Particle filters are used to estimate unknown states $x_{1:t} = \{x_1, x_2....x_t\}$ given observations $z_{1:t} = \{z_1, z_2....z_t\}$. The prior distribution $p(x_t|z_{1:t-1})$ for the unknown states along with the likelihood $p(z_t|x_t)$ helps us in estimating the posterior distribution $p(x_t|z_{1:t})$ using Bayes' rule. The sequential observations update the posterior distribution. The posterior distribution can be obtained recursively in two steps, prediction and update. For the prediction step, the state evolution $p(x_t|x_{t-1})$ is needed:

$$p(x_t|z_{1:t-1}) = p(x_t|x_{t-1})p(x_{t-1}|z_{1:t-1}) \tag{10}$$

Next, in the update step, Bayes' rule is used to modify the prior density and obtain the desired posterior distribution:

$$p(x_t|z_{1:t}) = \frac{p(x_t|z_{1:t-1})p(z_t|x_t)}{\int p(x_t|z_{1:t-1})p(z_t|x_t)dx_t} \tag{11}$$

The solution of the problem defined by (10) and (11) is analytically tractable in only a limited number of cases. When a large number of samples are drawn from the posterior distribution, intractable integrals can be numerically computed. Using Monte Carlo methods, N_s independent and identically distributed random weighted samples (particles) are simulated, $\{x_t^{(i)}, wt_t^{(i)}\}_{i=1}^{N_s}$ according to $p(x_t|z_{1:t})$

$$p(x_t|z_{1:t}) \sim \sum_{i=1}^{N_s} wt_t^{(i)}\delta(x_t - x_t^{(i)}), \tag{12}$$

where $\delta(.)$ is the Dirac delta and the sum of the weights is normalized to one. The particles are chosen using the principle of importance sampling.

2.4. Particle Filter with the Implicit Convex Shape Model

We use our implicit convex shape model to get a precise observation model that represents the shape of the cell. The state vector is comprised of the parameters $\{w_{ik}\}_{i=1}^{N}$ that characterize the shape of the cell. At a given frame, we propagate each particle of the previous frame by perturbing the location of the shape model, which generates the predicted set of particles $\tilde{x}_t^{(i)}$. First we pick location perturbations $(\Delta x, \Delta y, \Delta z)$ from a normal distribution with standard deviation 2 pixels. Then the model can be translated by adding $\Delta w_{i3} = w_{i0}\Delta x + w_{i1}\Delta y + w_{i2}\Delta z$, see equation (2), to the bias term. Each particle has an energy associated with it, which indicates how well the image gradients are aligned with the function boundaries. This energy is optimized using gradient descent to give the optimized particles $\tilde{x}_{t,opt}^{(i)}$ with associated energies $E_{opt}^{(i)}$. Particles with lower energy need to be given larger weights. Hence,

$$wt_t^{(i)} \leftarrow e^{-\lambda E_{opt}^{(i)}} \tag{13}$$

Finally, a resampling step is added to eliminate particles that have small weights and to focus on particles with large weights. At each frame the estimation of the shape of the target is given by a MAP estimator as follows:

$$\hat{x}_t \leftarrow \arg\max_i \{wt_t^{(i)}\}$$

The complete method is summarized in Algorithm 1.

input: Particle set $\{x_{t-1}^{(i)}\}_{i=1}^{N_s}$ from the implicit convex shape model and current image I_t
output: MAP estimation $\hat{x}_t$ and particle set $\{x_t^{(i)}\}_{i=1}^{N_s}$
for $i \leftarrow 1$ *to* N_s **do**
  $\tilde{x}_t^{(i)} \leftarrow$ Propagate $x_{t-1}^{(i)}$ with the motion model by perturbing the implicit convex shape model
  $\{\tilde{x}_{t,opt}^{(i)}, E_{opt}^{(i)}\} \leftarrow$ Adjust the implicit convex shape model to I_t
  $wt_t^{(i)} \leftarrow e^{-\lambda E_{opt}^{(i)}}$
end
for $i \leftarrow 1$ *to* N_s **do**
  $wt_t^{(i)} \leftarrow wt_t^{(i)} / \sum_j wt_t^{(j)}$
end
$\hat{x}_t \leftarrow \arg\max_i \{wt_t^{(i)}\}$
$\{x_t^{(i)}\}_{i=1}^{N_s} \leftarrow$ Resampling$(\{x_{t,opt}^{(i)}, wt_t^{(i)}\}_{i=1}^{N_s})$

Algorithm 1: Particle filter with the implicit convex shape model.

3. RESULTS

We applied our algorithm to a time series of confocal images to track the retinal and lens cells in zebrafish embryos. We address tracking during the nonmitotic stages of cells. Mitosis detection is a separate complex problem [5, 6] that is not addressed in this paper. The ground truth consists of a rough

estimate of the centroid of 14 lens cells and 19 retinal cells across 227 time frames. We used 50 particles per cell. The values for λ in (13) and α in (6) were chosen empirically as $\lambda = 0.01$ and $\alpha = 0.1$. Figures (2) and (3) illustrate the average distance (in pixels) between the true and predicted centroids for the lens cells and the retinal cells for 227 time frames, respectively. We observe that the distances vary by a maximum of 3-4 pixels. The average radius of these cells is approximately 6 pixels; however, one has to remember that the ground truth centroid positions are not precise.

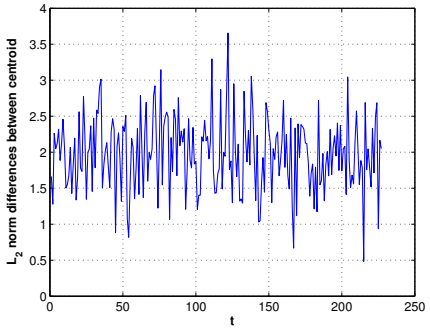

Fig. 2: Average distance between predicted and user provided centroids for 14 lens cells.

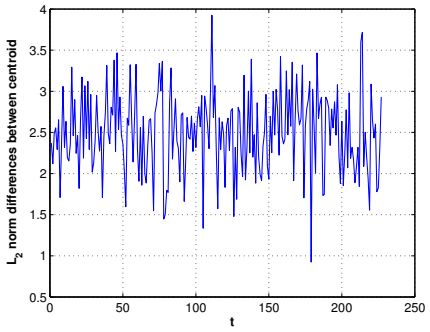

Fig. 3: Average distance between predicted and user provided centroids for 19 retinal cells.

As a second way to assess accuracy, the percentages of cells tracked correctly for varying time frames are shown in Figures (4) and (5) for the lens cells and retinal cells, respectively. A cell is considered correctly tracked in a given time frame if the ground truth centroid point is inside the model associated with that cell and no other ground truth centroids are inside the same model. For better assessment of the importance of our implicit convex shape model, we tested our tracking algorithm by replacing our shape model with a basic sphere model

$$f(\mathbf{x}) = (x - c_x)^2 + (y - c_y)^2 + (z - c_z)^2 - r^2 \qquad (14)$$

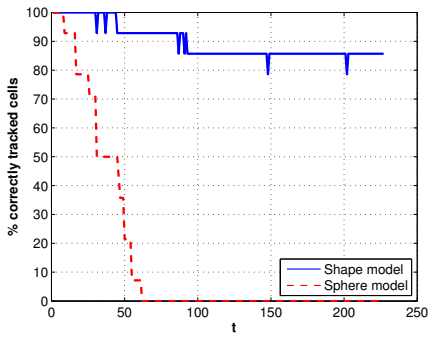

Fig. 4: % correct tracking vs time curves for lens cells.

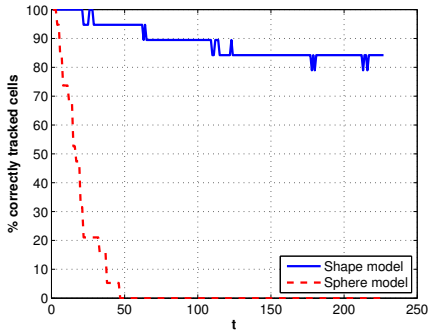

Fig. 5: % correct tracking vs time curves for retinal cells.

This spherical model was also embedded in the particle filter framework and similar experiments were conducted. The percentages of lens cells and retinal cells tracked correctly for varying time frames using the spherical model are shown in Figures (4) and (5), respectively. Notice that the percentage of accurately tracked cells drops much faster with the spherical model, which is likely due to the failure of the spherical model to precisely capture arbitrary cell shapes leading to unreliable energies (after minimization) and hence inaccurate importance sampling in the particle filter.

4. CONCLUSION

We demonstrated an algorithm that incorporates a novel implicit convex shape model in the particle filter. Experimental results show that, by using an implicit convex shape model, we are able to segment and track the cells more accurately in comparison to a spherical shape model. For future work, we will extend the shape model to include cells with concavities. This can be achieved by using a union of convex polytopes where each convex polytope is represented as in this paper. Finally, this more generic model can also make it possible to extend our tracking algorithm to mitotic stages by being able to capture cell shape during mitosis events.

5. REFERENCES

[1] P.J. Keller, A.D. Schmidt, J. Wittbrodt, and E.H.K. Stelzer, "Reconstruction of zebrafish early embryonic development by scanned light sheet microscopy," *Science*, vol. 322, no. 5904, pp. 1065–1069, 2008.

[2] A.J. Hand, T. Sun, D.C. Barber, D.R. Hose, and S. MacNeil, "Automated tracking of migrating cells in phase-contrast video microscopy sequences using image registration," *Journal of microscopy*, vol. 234, no. 1, pp. 62–79, 2009.

[3] E. Meijering, O. Dzyubachyk, I. Smal, and W.A. van Cappellen, "Tracking in cell and developmental biology," *Seminars in cell & developmental biology*, vol. 20, no. 8, pp. 894–902, 2009.

[4] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[5] R. Delgado-Gonzalo, N. Chenouard, and M. Unser, "A new hybrid bayesian-variational particle filter with application to mitotic cell tracking," in *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2011, pp. 1917–1920.

[6] N. Carranza, I. Smal, O. Dzyubachyk, W. Niessen, and E. Meijering, "Automated lineage tree reconstruction from caenorhabditis elegans image data using particle filtering based cell tracking," in *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2011, pp. 1921–1924.

[7] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi, "Tracking deforming objects using particle filtering for geometric active contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1470–1475, 2007.

[8] P. Thevenaz, R. Delgado-Gonzalo, and M. Unser, "The ovuscule," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 382–393, 2011.

[9] K. Miura, "Tracking movement in cell biology," *Microscopy Techniques*, pp. 267–295, 2005.

[10] M. Seyedhosseini, M. Sajjadi, and T. Tasdizen, "Image segmentation with cascaded hierarchical models and logistic disjunctive normal networks," in *IEEE International Conference on Computer Vison*, 2013.

[11] T. Tasdizen and D.B. Cooper, "Boundary estimation from intensity/color images with algebraic curve models," in *15th International Conference on Pattern Recognition*, 2000, vol. 1, pp. 225–228.