Cell Population Tracking and Lineage Construction Using Multiple-Model Dynamics Filters and Spatiotemporal Optimization

Kang Li, Takeo Kanade

Department of Electrical and Computer Engineering
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213

Abstract— **Automated visual-tracking of cell populations *in vitro* using phase contrast time-lapse microscopy is vital for quantitative, systematic and high-throughput measurements of cell behaviors. These measurements include the spatiotemporal quantification of migration, mitosis, apoptosis, and cell lineage. This paper presents an automated cell tracking system that can simultaneously track and analyze thousands of cells. The system performs tracking by cycling through frame-by-frame track compilation and spatiotemporal track linking, combining the power of two tracking paradigms. We applied the system to a range of cell populations including adult stem cells. The system achieved tracking accuracies in the range of 85.9%–92.5%, outperforming previous work by up to 9%. The proposed tracking methodology is valuable for tissue engineering, stem cell research, drug discovery and development, and related areas.**

I. Introduction

Automated tracking of cell populations *in vitro* in time-lapse microscopy images [6], [14], [18] can provide high-throughput spatiotemporal measurements of a range of cell behaviors, including *migration* (translocation), *mitosis* (division), *apoptosis* (death), and *lineage* (parent-daughter relations). This capability is valuable to research in genomics, proteomics, stem cell biology, and tissue engineering.

Traditional approaches for tracking include *tracking by detection* and *tracking by model-evolution*, each with its advantages and disadvantages. Recently, efforts were made to combine the strengths of both approaches and mitigating against their weaknesses [12]. The solution was to integrate four collaborative modules, including: 1) *cell detector*, which detects and labels candidate cell regions in the input image; 2) *cell tracker*, which propagates cell regions and identities across frames; 3) *dynamics filter*, which performs prediction and filtering of the cell motion dynamics using a Kalman filter; and 4) *track arbitrator*, which manages the tracking task by incorporating newly-entered cells, removing departed/dead cells, establishing cell lineages, and recovering lost tracks. The system can simultaneously track thousands of living cells imaged with phase-contrast microscopy in realtime [12].

However, several limitations are inherent in the aforementioned tracking system. First, all of its modules operate in a frame-by-frame manner. Hence, only very limited spatiotemporal context is considered, hindering the capability in resolving complete or long term occlusions. Secondly, the track arbitrator module makes immediate, hard decisions for each frame, precluding the possibility for retrospective error detection and correction. Thirdly, the Kalman filter used for dynamics filtering is bound to use one dynamics model only, which can be problematic as the dynamics of cells may vary frequently with time.

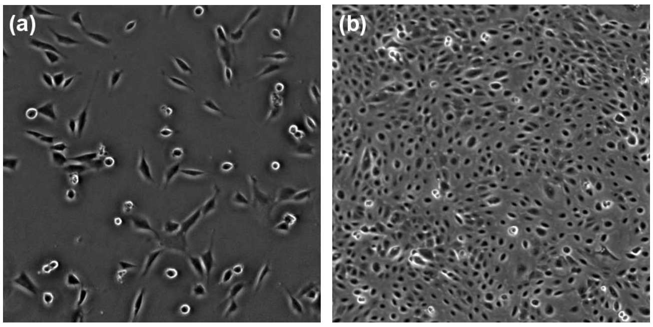

Fig. 1. Examples of phase contrast microscopy images of cell populations. (a) shows MG-63 human osteosarcoma cells; and (b) shows human amnion epithelial (AE) cell population. The images are contrast-enhanced for the purpose of display and publication.

We propose an improved tracking system to address the above issues. Specifically, we divide the track arbitrator into two submodules: *track compiler* and *track linker*. Track compiler operates in a frame-by-frame manner and produces intermediate tracking results called *track segments*. Track linker oversees the entire tracking history and establishes final cell trajectories and lineages only when enough information is available. We also adopt the *interacting multiple models* (IMM) filter [4], which allows multiple dynamics models in parallel, and was shown to be more biologically relevant than a Kalman filter [8]. We focus on reliable long-term tracking of cell centroid locations and lineages. Accurate segmentation of cell boundaries is a plus, but not the emphasis of this paper.

II. Methods

The proposed tracking system has five major modules (Fig. 2). The trajectory of one cell may consist of multiple track segments. The system associates each cell track segment with a unique positive-integer label n. We identify each cell using the label of its first track segment. Let $k = 0, ..., K$ be the frame index. The cell regions in an image frame $I_k(x, y)$ are represented using a region labeling function $\psi_k(x, y)$. Wherein $\psi_k(x, y) = n$ if pixel (x, y) is part of cell n, and $\psi_k(x, y) = 0$ if pixel (x, y) belongs to the background. To

Corresponding author: Kang Li, email: kangl@cmu.edu.

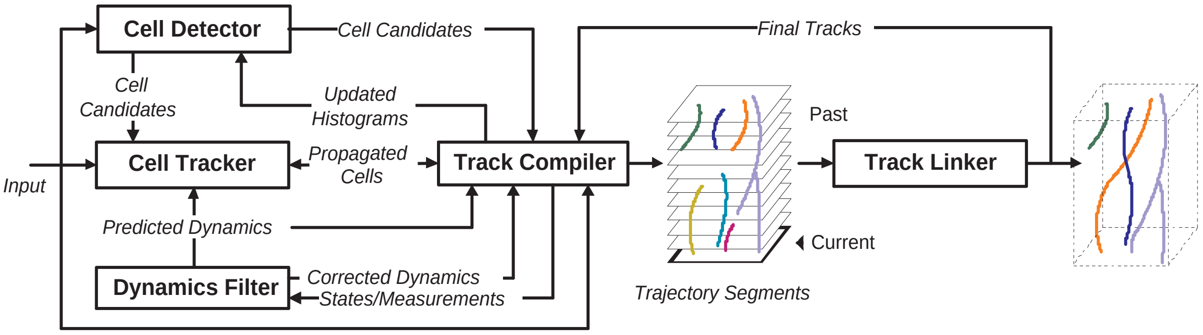

Fig. 2. System Overview

initialize tracking, the system generates initial cell labeling $\psi_0(x, y)$ by running the cell detector on the first frame $I_0(x, y)$. For each subsequent frame $I_k(x, y)$:

Step 1: The cell detector classifies the image pixels into cell (C) and background (B) regions by a combined use of morphorlogical rolling-ball filtering [17] and a Bayesian classifier based on estimated cell and background color histograms. The histograms are initially learned off-line and incrementally updated on-the-fly using the tracking result as feedback. The output is a binary map of cell regions, denoted $\zeta_k(x, y)$. Each connected foreground component in $\zeta_k(x, y)$ is considered as a *cell candidate* in frame k.

Step 2: The cell tracker propagates the cell region labeling $\psi_{k-1}(x, y)$ from frame $k - 1$ to frame k. We use a fast level set algorithm proposed in [16] to segment cell regions and to propagate the corresponding cell labels. We first initialize a level set function $\phi_k(x, y)$ using $\psi_{k-1}(x, y)$. Then ϕ and ψ are evolved together to minimize an "energy" functional that combines a region-competition term [19], a geodesic edge term [5], and a dynamics term based on the cell dynamics predicted by the dynamics filter. To prevent contours that represent different cells from merging, we incorporate topological constraints to the level set evolution. The output is the propagated cell labeling for frame k, denoted $\psi_k^*(x, y)$.

Step 3: The track compiler compares the output of the cell detector and cell tracker, and take one of the following actions: to create a new or daughter track segment, to update an existing track, or to terminate a track. Meanwhile, the *dynamics filter* updates the cell dynamics state in frame k, and predicts its state for frame $k + 1$. The output includes the track segments, an updated region labeling $\psi_k(x, y)$, and updated cell and background histograms.

Step 4: The track linker examines all track segments up to frame k, and detects whether two or more track segments may correspond to one cell. It attempts to link track segments in the spatiotemporal image volume, and to form more complete cell trajectories. The updated cell trajectories are fed back to the track compiler for subsequent tracking in frame $k + 1$.

The following sections elaborate on the dynamics filter, track compiler, and track linker. We refer readers to [12] for details on the other modules.

A. Interacting Multiple Models Dynamics Filter

The mechanism underlying cell locomotion involves a complicated set of factors. Mathematical modeling of this mechanism relies on the solution of an inverse problem [2], which is, unfortunately, ill-posed given limited image data. To reach a realistic model of cell dynamics for the purpose of tracking, several simplifying assumptions must be made. Our assumptions include: 1) the cell shape can be approximated by an ellipse at appropriate scales of observation, which can be parameterized by its centroid and a covariance matrix; 2) the cell centroid translocation and intensity change are governed by two independent dynamical processes; and 3) the dynamical processes governing different cells are independent. These seemingly-restrictive assumptions allow us to employ state-space dynamical filtering to estimate unknown cell states from noisy measurements in real-time. With the collected strength of multiple modules and spatiotemporal inference, this approach allows us to achieve satisfactory tracking performance at realistic computational overhead.

We define the state of each cell in frame k to consist of two subcomponents: a (centroid) motion state $\mathbf{s}_k^O$ and an intensity state $\mathbf{s}_k^I$. The motion state is defined as a concatenation of the cell centroid coordinates in the last three frames: $\mathbf{s}_k^O = (x_k, y_k, x_{k-1}, y_{k-1}, x_{k-2}, y_{k-2})^T$, where the superscript T denotes matrix transposition. Analogously, the intensity state is a concatenation of the mean cell intensities in the last three frames, i.e., $\mathbf{s}_k^I = (i_k, i_{k-1}, i_{k-2})^T$. The corresponding motion measurement vector $\mathbf{z}_k^O$ and intensity measurement vector $\mathbf{z}_k^I$ consist of the measured cell centroid coordinates and mean intensity in frame k, respectively. In the following text, we focus on the dynamical filtering of cell centroid motion. A similar discussion applies to the filtering of cell intensities. For brevity, we will omit the superscripts, O and I, on the state and measurement vectors.

Suppose cellular motion consists of a finite number of modes, each of which can be described by a linear state-space model with additive Gaussian noise:

$$\mathbf{s}_k = \mathbf{F}^i \mathbf{s}_{k-1} + \mathbf{v}_{k-1}^i, \quad (1)$$

$$\mathbf{z}_k = \mathbf{H} \mathbf{s}_k + \mathbf{w}_k. \quad (2)$$

Here, matrix $\mathbf{F}^i$ is the state transition matrix of model $i \in \{1, ..., r\}$, where r is the number of models. $\mathbf{H}$ is the measurement matrix that relates states to measurements. $\mathbf{v}_{k-1}^i$ and

$\mathbf{w}_k^i$ are the process and measurement noise vectors, which are uncorrelated zero-mean Gaussian processes with covariances $\mathbf{Q}^i$ and $\mathbf{R}$, respectively. The determination of these model parameters will be discussed in the next section.

The interacting multiple models (IMM) filter [4] operates several Kalman filters in parallel. Each filter is matched to a distinct mode of target motion. A finite state Markov chain governs the transition between models, with probability p_{ij} of switching from model i to model j in successive frames. The (forward-time) filtering recursion consists of two stages: *prediction* and *correction*.

Prediction: Starting from r weights ρ_{k-1}^i, states $\hat{\mathbf{s}}_{k-1}^i$ and covariances $\mathbf{P}_{k-1}^i$ from the previous iteration, we compute the mixed initial condition:

$$\hat{\mathbf{s}}_{k-1}^{0j} = \sum_i \rho_{k-1}^{i|j} \hat{\mathbf{s}}_{k-1}^i, \tag{3}$$

$$\mathbf{P}_{k-1}^{0j} = \sum_i \rho_{k-1}^{i|j} \left[\mathbf{P}_{k-1}^i + \left(\hat{\mathbf{s}}_{k-1}^i - \hat{\mathbf{s}}_{k-1}^{0j}\right)(...)^T\right], \tag{4}$$

where $\rho_{k-1}^{i|j} = p_{ij}\rho_{k-1}^i/\rho_{k|k-1}^j$, $\rho_{k|k-1}^j = \sum_i p_{ij}\rho_{k-1}^i$, and $(...)$ denotes a replication of the previous term. These are input to r Kalman filters to compute the state prediction $\hat{\mathbf{s}}_{k|k-1}^j$ and covariance $\mathbf{P}_{k|k-1}^j$. The combined state and covariance predictions can be determined by:

$$\hat{\mathbf{s}}_{k|k-1} = \sum_j \rho_{k|k-1}^j \hat{\mathbf{s}}_{k|k-1}^j, \tag{5}$$

$$\mathbf{P}_{k|k-1} = \sum_j \rho_{k|k-1}^j \left[\mathbf{P}_{k|k-1}^j + (\hat{\mathbf{s}}_{k|k-1}^j - \hat{\mathbf{s}}_{k|k-1})(\cdots)^T\right]. \tag{6}$$

These are fed to the cell tracker to guide the level set evolution in frame k [12].

Correction: Given the predicted states, covariances, and measurement $\mathbf{z}_k$ (Fig. 2), we use the Kalman filters to obtain the updated state $\hat{\mathbf{s}}_k^j$ and covariance $\mathbf{P}_k^j$. The likelihood that model j is activated in frame k is

$$\lambda_k^j = \exp\left[-\frac{1}{2}(\mathbf{y}_k^j)^T(\mathbf{S}_k^j)^{-1}\mathbf{y}_k^j\right] / \sqrt{2\pi\det(\mathbf{S}_k^j)}, \tag{7}$$

where $\mathbf{y}_k^j = (\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})$ is the innovation of Kalman filter j, and $\mathbf{S}_k^j$ is the associated covariance. Then, the combined state $\hat{\mathbf{s}}_k$ and covariance $\mathbf{P}_k$ estimates can be computed by Equations (5) and (6), with $\rho_{k|k-1}^j$ replaced by $\rho_k^j = \rho_{k|k-1}^j\lambda_k^j/(\sum_i \rho_{k|k-1}^i\lambda_k^i)$.

To initialize the IMM filter, the system tracks each cell without dynamical filtering in the first three frames that it appears. It uses the concatenation of measurements in these frames as the initial cell state $\hat{\mathbf{s}}_0$, and sets the initial error covariances $\mathbf{P}_0^i$ (for all i) to be the Kronecker product of a 3×3 identity matrix and $\mathbf{R}$. The initial model weights ρ_0^i are taken as $1/r$, indicating complete uncertainty as to which model is initially correct.

B. Maximum-Likelihood Parameter Estimation for IMM

The efficacy of the IMM filter hinges on the parameters $\mathbf{F}^i$, $\mathbf{H}$, $\mathbf{Q}^i$, $\mathbf{R}$, and the model transition probabilities p_{ij} $(i, j \in \{1, ..., r\})$. Inspired by [8], we define three models of cell dynamics: random walk, first-order, and second-order linear extrapolation. They represent three typical modes of cellular motion: Brownian motion, migration with constant speed, and migration with constant acceleration. The corresponding system matrices are:

$$\mathbf{F}^1 = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix}, \quad \mathbf{F}^2 = \begin{bmatrix} 2\mathbf{I} & -\mathbf{I} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix},$$
$$\mathbf{F}^3 = \begin{bmatrix} 3\mathbf{I} & -3\mathbf{I} & \mathbf{I} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{8}$$

where $\mathbf{I}$ is an 2×2 identity matrix, and $\mathbf{0}$ denotes a 2×2 zero matrix. We chose the transition probabilities as $p_{ij} = 1/3$ $(\forall i, j \in \{1, 2, 3\})$, allowing equally probable switching between any two models.

With these parameters set, we can estimate the remaining parameters $\mathbf{Q}^i$ and $\mathbf{R}$ from off-line training sequences using the expectation-maximization (EM) algorithm. With the unknown parameters denoted by $\boldsymbol{\Theta} \equiv \{\mathbf{Q}^i, \mathbf{R}\}$, our objective is to estimate the parameters $\boldsymbol{\Theta}$ and the hidden states $\mathbf{S} = \{\mathbf{s}_0, ..., \mathbf{s}_K\}$ based on the measurements $\mathbf{Z} = \{\mathbf{z}_{-2}, ..., \mathbf{z}_K\}$. The index of the measurement sequence starts from -2 purely for the convenience of notation. We use the first three measurements $\{\mathbf{z}_{-2}, \mathbf{z}_{-1}, \mathbf{z}_0\}$ to obtain the initial state $\mathbf{s}_0$.

The EM algorithm maximizes the complete-data log likelihood, defined by

$$\begin{aligned} &\log P(\mathbf{S}, \mathbf{Z}|\boldsymbol{\Theta}) = \\ &\quad -\frac{K+3}{2}\log|\mathbf{R}| - \frac{K}{2}\log|\mathbf{Q}^i| \\ &\quad -\frac{1}{2}\sum_{k=1}^{K}(\mathbf{z}_k - \mathbf{H}\mathbf{s}_k)^T\mathbf{R}^{-1}(\mathbf{z}_k - \mathbf{H}\mathbf{s}_k) \\ &\quad -\frac{1}{2}\sum_{k=1}^{K}(\mathbf{s}_k - \mathbf{F}^i\mathbf{s}_{k-1})^T(\mathbf{Q}^i)^{-1}(\mathbf{s}_k - \mathbf{F}^i\mathbf{s}_{k-1}) \\ &\quad + \text{constant}, \end{aligned} \tag{9}$$

where $|\cdot|$ denotes matrix determinant. Note that to obtain Eq. (9), we utilized the fact that the initial error covariance $\mathbf{P}_0^i$ equals the Kronecker product of a 3×3 identity matrix and $\mathbf{R}$, and hence $\log|\mathbf{P}_0| = 3\log|\mathbf{R}|$.

The maximization of $\log P(\mathbf{S}, \mathbf{Z}|\boldsymbol{\Theta})$ with respect to the unknown parameters $\boldsymbol{\Theta}$ is a chicken-and-egg problem since the system states $\mathbf{S}$ are also unknown. The EM algorithm solves this problem by iterating between two steps: an *expectation* (E) step and a *maximization* (M) step.

1) E Step: The E step finds the expected value of the complete-data log likelihood with respect to the unknown states $\mathbf{S}$, given the observed data $\mathbf{Z}$ and the current parameter estimates $\boldsymbol{\Theta}^{\text{old}}$, i.e.,:

$$\mathcal{Q}(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{\text{old}}) = E\left[\log P(\mathbf{S}, \mathbf{Z}, \boldsymbol{\Theta}|\mathbf{Z}, \boldsymbol{\Theta}^{\text{old}})\right]. \tag{10}$$

This quantity depends on three expectations:

$$\hat{\mathbf{s}}_{k|K} \equiv E\left[\mathbf{s}_k|\mathbf{Z},\mathbf{\Theta}^{\text{old}})\right] \tag{11}$$

$$\mathbf{P}_{k|K} \equiv E\left[\mathbf{s}_k\mathbf{s}_k^T|\mathbf{Z},\mathbf{\Theta}^{\text{old}})\right] \tag{12}$$

$$\mathbf{P}_{k,k-1|K} \equiv E\left[\mathbf{s}_k\mathbf{s}_{k-1}^T|\mathbf{Z},\mathbf{\Theta}^{\text{old}})\right] \tag{13}$$

Note that the estimates $\hat{\mathbf{s}}_{k|K}$ and $\hat{\mathbf{P}}_{k|K}$ differ from the ones computed by the forward-time IMM filter in that they depend on past as well as *future* observations. To obtain these estimates we utilize the fixed-interval IMM smoother proposed by Helmick *et al* [9]. The algorithm uses two IMM filters. One of the filters propagates in the forward-time direction, and produces estimates $\hat{\mathbf{s}}_k^j$ and $\mathbf{P}_k^j$ as given previously. The other filter propagates in the backward-time direction, and produces estimates $\hat{\mathbf{s}}_k^{b,i}$, $\mathbf{P}_k^{b,i}$, $\hat{\mathbf{s}}_{k|k+1}^{b,i}$ and $\mathbf{P}_{k|k+1}^{b,i}$. We refer the reader to [9] for details on the backward-time IMM filter. The IMM smoother combines the forward and backward filtered outputs to obtain smoothed estimates following the procedure below.

Step 1: Compute combined estimates:

$$\begin{aligned}
\hat{\mathbf{s}}_{k|K}^{ji} &= \mathbf{P}_{k|K}^{ji}\left[(\mathbf{P}_k^j)^{-1}\hat{\mathbf{s}}_k^j + (\mathbf{P}_{k|k+1}^{b,i})^{-1}\hat{\mathbf{s}}_{k|k+1}^{b,i})\right],\\
\mathbf{P}_{k|K}^{ji} &= \left[(\mathbf{P}_k^j)^{-1} + (\mathbf{P}_{k|k+1}^{b,i})^{-1}\right]^{-1},\\
\mathbf{P}_{k+1,k|K}^{ji} &= \left[(\mathbf{P}_{k+1,k|k}^j)^{-1} + (\mathbf{P}_{k+1,k|k+1}^{b,i})^{-1}\right]^{-1},
\end{aligned} \tag{14}$$

with $\mathbf{P}_{k+1,k|k}^j = \mathbf{F}^j\mathbf{P}_k^j$, and $\mathbf{P}_{k+1,k|k+1}^{b,i} = \mathbf{P}_{k+1}^{b,i}[(\mathbf{F}^i)^{-1}]^T$.

Step 2: Compute model-conditioned smoothed estimates:

$$\begin{aligned}
\hat{\mathbf{s}}_{k|K}^j &= \sum_{i=1}^r \mu_{k+1|K}^{i|j}\hat{\mathbf{s}}_{k|K}^{ji},\\
\mathbf{P}_{k|K}^j &= \sum_{i=1}^r \mu_{k+1|K}^{i|j}\left[\mathbf{P}_{k|K}^{ji} + (\hat{\mathbf{s}}_{k|K}^{ji} - \hat{\mathbf{s}}_{k|K}^j)(...)^T\right],\\
\mathbf{P}_{k+1,k|K}^j &= \sum_{i=1}^r \mu_{k+1|K}^{i|j}\left[\mathbf{P}_{k+1,k|K}^{ji}\right.\\
&\quad\left. + (\hat{\mathbf{s}}_{k+1|K}^{ji} - \hat{\mathbf{s}}_{k+1|K}^j)(\hat{\mathbf{s}}_{k|K}^{ji} - \hat{\mathbf{s}}_{k|K}^j)^T\right].
\end{aligned} \tag{15}$$

The conditional probability $\mu_{k+1|K}^{i|j}$ is obtained by

$$\mu_{k+1|K}^{i|j} = p_{ji}\lambda_k^{ji}/\gamma_j, \text{ with } \gamma_j = \sum_{i=1}^r p_{ji}\lambda_k^{ji}. \tag{16}$$

The likelihood λ_k^{ji} is given by

$$\lambda_k^{ji} = \mathcal{N}(\hat{\mathbf{s}}_{k|k+1}^{b,i} - \hat{\mathbf{s}}_k^j; 0, \mathbf{P}_{k|k+1}^{b,i} + \mathbf{P}_k^j), \tag{17}$$

where $\mathcal{N}(\cdot)$ denotes a multivariate normal distribution.

Step 3: Compute the overall smoothed estimates:

$$\begin{aligned}
\hat{\mathbf{s}}_{k|K} &= \sum_{j=1}^r \mu_k^{s,j}\hat{\mathbf{s}}_{k|K}^j,\\
\mathbf{P}_{k|K} &= \sum_{j=1}^r \mu_k^{s,j}\left[\mathbf{P}_{k|K}^j + (\hat{\mathbf{s}}_{k|K}^j - \hat{\mathbf{s}}_{k|K})(...)^T\right],\\
\mathbf{P}_{k+1,k|K} &= \sum_{j=1}^r \mu_k^{s,j}\left[\mathbf{P}_{k+1,k|K}^j\right.\\
&\quad\left. + (\hat{\mathbf{s}}_{k+1|K}^j - \hat{\mathbf{s}}_{k+1|K})(\hat{\mathbf{s}}_{k|K}^j - \hat{\mathbf{s}}_{k|K})^T\right].
\end{aligned} \tag{18}$$

The smoothed model probability $\mu^{s,j}$ can be computed as

$$\mu_k^{s,j} = \frac{\gamma_j\mu_k^j}{\sum_{j=1}^r \gamma_j\mu_k^j}, \tag{19}$$

where μ_k^j is the forward-time filtered model probability.

2) M Step: The M-step of the EM algorithm re-estimates the unknown parameters by maximizing the expectation we computed in the first step, i.e.,

$$\mathbf{\Theta}^{\text{new}} = \max_{\mathbf{\Theta}} \mathcal{Q}(\mathbf{\Theta}|\mathbf{\Theta}^{\text{old}}). \tag{20}$$

By taking the partial derivatives of $\mathcal{Q}$ with respect to $(\mathbf{Q}^i)^{-1}$ and $\mathbf{R}^{-1}$, and setting the respective result to zero, we obtain

$$\begin{aligned}
(\mathbf{Q}^i)^{\text{new}} &= \frac{1}{K}\sum_{k=1}^K \left[\mathbf{P}_{k|K} - \mathbf{F}^i\mathbf{P}_{k-1,k|K}\right.\\
&\quad\left. - \mathbf{P}_{k,k-1|K}(\mathbf{F}^i)^T + \mathbf{F}^i\mathbf{P}_{k-1|K}(\mathbf{F}^i)^T\right],
\end{aligned} \tag{21}$$

$$\mathbf{R}^{\text{new}} = \frac{1}{K+3}\sum_{k=1}^K \left(\mathbf{z}_k\mathbf{z}_k^T - 2\mathbf{H}\hat{\mathbf{s}}_{k|K}\mathbf{z}_k^T + \mathbf{H}\mathbf{P}_{k|K}\mathbf{H}^T\right). \tag{22}$$

The expectation and maximization steps are computed repeatedly until the relative absolute change of the expected log likelihood is below a threshold. Each iteration is guaranteed to increase the log likelihood, and the algorithm will converge to a local maximum of the log likelihood function.

As an possible enhancement to the above approach, we can also treat the transition probabilities as unknowns and estimate them online using historical tracking data [7], [10]. This approach needs further experimental validation and will be our future work. In addition, while the EM algorithm permits the estimation of $\mathbf{F}^i$ and $\mathbf{H}$ from training data as well, the resulting matrices may be arbitrary and hard to interpret. With predefined system matrices, we gain additional insight as to identifying the typical motion mode of each cell. That is, we can identify whether the motion of a cell is predominated by random walk, migration with constant speed, or acceleration based on the model weights ρ_k^i output by the IMM filter.

C. *Track Compilation*

The track compiler coordinates cell detector, cell tracker and motion filter to produce track segments. We use $\mathbf{N}_k$ to denote the set of labels of all track segments created up to frame k. A track segment is *active* in frame k if it was successfully tracked in frame $k-1$, otherwise it becomes *inactive*. Let Ω_0 denote the background region, and Ω_n denote the cell region with label n. An outline of the track compilation algorithm is shown in Algorithm 1.

The compiler first compares the output of the cell detector and cell tracker, $\zeta_k(x,y)$ and $\psi_k^*(x,y)$. Each cell candidate in $\zeta_k(x,y)$ that does not overlap with any propagated cell region in $\psi_k^*(x,y)$ is considered a *new* cell. A new track segment will be initialized, and $\psi_k^*(x,y)$ will be updated accordingly.

Next, the algorithm scans through all active track segments, and deactivates track segments whose labels are not found in the propagated region labeling $\psi_k^*(x,y)$. A track segment whose corresponding propagated cell region contains only one

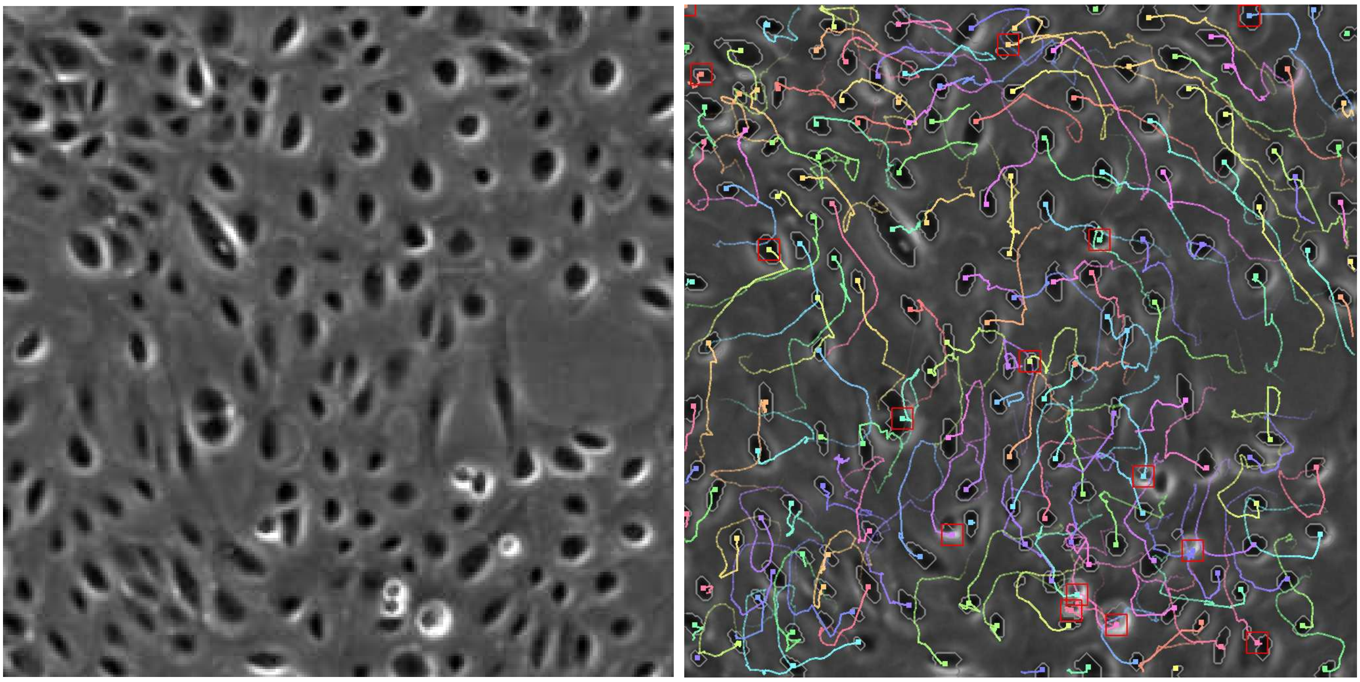

Fig. 3. Visualization of tracking results of AE cells in the subregion used for quantitative validation. **Left**: original image. **Right**: the image with cell trajectories overlaid. Red rectangles indicate cells that were detected to be mitotic in the past $T = 10$ frames.

```
Algorithm 1: TrackCompilation
$\Omega_0 \leftarrow \{(x,y)|\psi_k^*(x,y)=0\}$
foreach cell candidate $\omega \subset \zeta_k$ do
    if $\omega \subset \Omega_0$ then AddTrack($n_{new}, k, \omega$)
foreach active track $n \in \mathbf{N}_{k-1}$ do
    $\Omega_n \leftarrow \{(x,y)|\psi_k^*(x,y)=n\}$
    if $\Omega_n = \varnothing$ then DeactivateTrack($n$)
    else if IsDivided($\Omega_n$) then
        if IsMitotic($n, k$) then
            foreach connected component $\omega \subset \Omega_n$ do
                AddDaughterTrack($n_{daughter}, n, k, \omega$)
        else
            $\omega^* \leftarrow$SelectBestMatch($n, k, \Omega_n$)
            UpdateTrack($n, k, \omega^*$)
            foreach connected component $\omega \subset \Omega_n \setminus \omega^*$
            do AddTrack($n_{new}, k, \omega$)
    else UpdateTrack($n, k, \Omega_n$)
```

connected component will be updated directly. If a cell region consists of more than one well-separated connected components, the track compiler will judge between two possibilities: 1) the cell divided into daughter cells; or 2) one or more of these components are from occluded cells or close-by newly-entered cells. The algorithm will either create daughter tracks or continue tracking using the component that best matches the cell trajectory, depending on whether the cell is previously detected to be mitotic.

Details of several key operations are as follow.

AddTrack(ω, n_{new}, k) creates a new track segment labeled n_{new}; fills region ω with n_{new}; and initializes the cell state based on measurements of ω.

UpdateTrack(n, k, ω) updates the track segment n using the features of region ω, including centroid location, mean intensity, area, and eccentricity. We feed the centroid and mean intensity to the motion filter to obtain a filtered state of cell n in frame k. We use the last three features to classify a cell as normal, mitotic, or apoptotic, using nearest neighbor matching with Mahalanobis distance to a set of training samples obtained off-line.

AddDaughterTrack$(n_{daughter}, n, k, \omega)$ creates a daughter track of cell n with a unique label $n_{daughter}$, and fills the region ω with $n_{daughter}$. The state of the daughter cell will be computed based on the measured centroid location and mean intensity of ω, and the predicted state of cell n.

SelectBestMatch(n, k, Ω_n) selects component $\omega^* \in \Omega_n$ that best matches the dynamics of cell n, i.e., the one which maximizes the innovation likelihood given by Equation (7) among all dynamic models.

IsDivided(Ω_n) returns `true` if region Ω_n has multiple connected components, and the minimum distance between any two points in different components is greater than a preset threshold D. Otherwise, it returns `false`.

IsMitotic(n, k) determines if cell n is mitotic during the past T frames.

D. Track Linking

The track linker detects potential problems among all track segments up to frame k based on two physical constraints: 1) a cell does not vanish unless it leaves the field-of-view, dies and releases into the media, or is occluded; and 2) a cell does not appear unless it enters from outside, divides from another cell, or comes out of occlusion. The linker attempts to correct

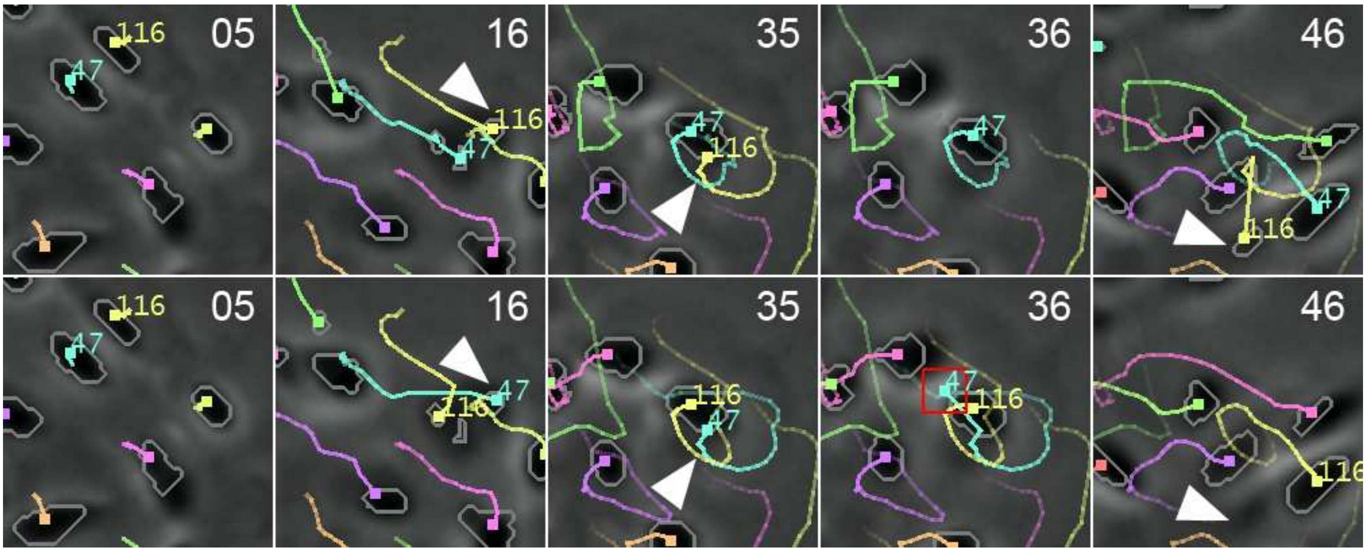

Fig. 4. Tracking AE cells through occlusion. **Top row**: The new system correcly tracked cell 116, which was completely occluded by cell 47 and reappeared later. **Bottom row**: Incorrect result was produced by the previous system, where cell 116 switched with cell 47 and was lost eventually. The numbers at the top-right corner are frame indices. The trailing curves represent cell trajectories. Different colors represent different cell lineages.

violations of these constraints by linking track segments into complete cell trajectories using spatiotemporal context.

Algorithm 2 outlines the track linking algorithm. Wherein, $\mathbf{N}_{\text{lost}} \equiv \{n_l | l = 1, ..., L\}$ denotes the label set of track segments that end before frame k, and $\mathbf{N}_{\text{found}} \equiv \{n_f | f = 1, ..., F\}$ denotes the label set of track segments that start after frame 0. Most operations in the algorithm are self-explanatory. One vital step is the matching between lost and appeared track segments: *MatchTracks* (Line 5).

Algorithm 2: TrackLinking

```
   $\mathbf{N}_{\text{lost}}, \mathbf{N}_{\text{found}} \leftarrow \varnothing$
1  foreach track $n \in \mathbf{N}_k$ do
2      if IsShort$(n, k)$ then DeleteTrack$(n)$
3      else if LostInField$(n, k)$ then Add $n$ to $\mathbf{N}_{\text{lost}}$
4      else if FoundInField$(n, k)$ then Add $n$ to $\mathbf{N}_{\text{found}}$
5  MatchTracks$(\mathbf{N}_{\text{lost}}, \mathbf{N}_{\text{found}})$
   foreach $n_l \in \mathbf{N}_{lost}$ do
6      if IsMatched$(n_l, n_f \in \mathbf{N}_{found})$ then
           LinkTracks$(n_l, n_f)$
7      else if IsMatched$(n_l; n_{f_1}, n_{f_2} \in \mathbf{N}_{found})$ then
           LinkTracks$(n_l; n_{f_1}, n_{f_2})$
```

In *MatchTracks*, we first create a bipartite graph G, whose nodes correspond to the labels in $\mathbf{N}_{\text{lost}}$ and $\mathbf{N}_{\text{found}}$. For each node pair (n_l, n_f), we create an arc $\langle n_l, n_f \rangle$ between node n_l and node n_f if the last centroid location (x_l, y_l, k_l) of track n_l is within a spatiotemporal double cone centered at the first centroid location (x_f, y_f, k_f) of track n_f, i.e.,

$$\sqrt{(x_l - x_f)^2 + (y_l - y_f)^2} \le |k_l - k_f| R + R_0, \quad \text{and}$$
$$|k_l - k_f| \le D/2,$$

where D, R and R_0 are user-defined parameters. Each arc $\langle n_l, n_f \rangle$ is assigned a weight $w_{lf} = \lambda^{\max}_{n_l, k_f}(n_f)$, which is the maximum innovation likelihood of track n_l on the measurement of track n_f in frame k_f (Eq. 7). Intuitively, w_{lf} indicates how likely track n_f is a continuation of n_l based on the dynamics of n_l.

Next, we compute a maximum-likelihood matching between tracks n_l and n_f. The approach we reported previously [11] only considered one-to-one matches. Hence, it could not handle the case where a cell is lost track during a mitosis, and whose daughter cells are re-detected in later frames. To handle this case, we propose an improved algorithm that considers both one-to-one and one-to-two matches. The algorithm rely on two inputs: an $H \times (L + F)$ constraint matrix $\mathbf{C}$ and an $H \times 1$ likelihood vector $\mathbf{d}$, where H is the total number of one-to-one and one-to-two matching hypotheses. We construct matrix $\mathbf{C}$ and vector $\mathbf{d}$ as follows.

For each arc $\langle n_l, n_f \rangle$ in G, we append a new row to $\mathbf{C}$ and a corresponding new element to $\mathbf{d}$. Let h be the index of this new row. We set $\mathbf{d}(h) = w_{lf}$, and

$$\mathbf{C}(h, i) = \begin{cases} 1, & \text{if } i = l \text{ or } i = L + f, \\ 0, & \text{otherwise.} \end{cases}$$

For each node n_l that is connected to multiple nodes $n_{f_1}, \cdots, n_{f_m} \in \mathbf{N}_{\text{found}}$ ($m \ge 2$), we enumerate all possible one-to-two matchings, e.g., $n_l \to (n_{f_1}, n_{f_2})$, $n_l \to (n_{f_1}, n_{f_3})$, and so on. For each of these hypotheses, say $n_l \to (n_{f_1}, n_{f_2})$, we append a new row with index h' to $\mathbf{C}$ and a corresponding new element to $\mathbf{d}$. We set $\mathbf{d}(h')$ to be the maximum innovation likelihood of track n_l on the spatiotemporal average of the starting points of tracks n_{f_1} and n_{f_2}, and

$$\mathbf{C}(h', i) = \begin{cases} 1, & \text{if } i = l,\ i = L + f_1, \text{ or } i = L + f_2, \\ 0, & \text{otherwise.} \end{cases}$$

With $\mathbf{C}$ and $\mathbf{d}$ constructed, the matching problem reduces to selecting a subset of rows of $\mathbf{C}$ such that the sum of corresponding elements in $\mathbf{d}$ is maximized, under the constraint that no two rows share common nonzero entries. This can be posed as the following integer programming problem:

$$\max_{\mathbf{x}} \mathbf{d}^T\mathbf{x}, \quad s.t. \;\; \mathbf{C}^T\mathbf{x} \leq \mathbf{1}, \tag{23}$$

where $\mathbf{1}$ is a $H \times 1$ vector of ones. $\mathbf{x}$ is a $H \times 1$ binary vector to be solved for, with $\mathbf{x}(h) = 1$ if row h is selected in the solution, or $\mathbf{x}(h) = 0$ otherwise. While integer programming problems are in general NP-hard, the problem given in Eq. (23) can be solved exactly using linear programming. This is due to the fact that the constraint matrix $\mathbf{C}$ is totally unimodular[1], and the right-hand sides of the constraints are all integers. In fact, if the above two conditions are satisfied, a linear programming problem will always have an integer-valued solution [15]. In practice, we use the open-source software package *lpsolve* [3] to solve the above integer programming problem. Note that a similar integer programming approach was used by Al-Kofahi *et al* [1] for inter-frame cell matching.

III. Experiments and Results

We quantitatively analyzed the performance of our system on two image sequences (A and B) of MG-63 human osteosarcoma cells used previously in [12], and two sequences (C and D) of proprietary amnion epithelial (AE) stem cells. Examples of phase-contrast microscopy images of these cell lines are shown in Fig. 1.

Sequences A and B were acquired with a 12-bit Qimaging Retiga EXi Fast 1394 CCD camera mounted on a Zeiss Axiovert 135 TV microscope, at an interval of 4 minutes/frame for 10 hours. Each sequence consists of 150 frames, with 512×512 pixels/frame, and 1.9 μm/pixel at 4.9x magnification. The cells were seeded randomly on a polystyrene dish.

Sequences C and D were acquired using the same protocol, aside from a frame interval of 10 minutes/frame. Each sequence spans 42.5 hours, and consists of 256 frames with 1280×1024 pixels/frame. The cell population is roughly 2000-5000 cells/frame, and is nearly confluent towards the end.

TABLE I. Tracking Accuracy Comparison

	Trajectory Validity			
Sequences	Current		Previous	
A	74/81	(91.4%)	70/81	(86.4%)
B	86/93	(92.5%)	82/93	(88.2%)
C	78/92	(84.8%)	70/92	(76.1%)
D	101/117	(86.3%)	90/117	(76.9%)
	Division Tracking Correctness			
Sequences	Current		Previous	
A	1/1	(100%)	1/1	(100%)
B	0/0	(N/A)	0/0	(N/A)
C	47/55	(85.5%)	43/55	(78.2%)
D	45/52	(86.5%)	41/52	(78.8%)

A human operator manually tracked the cell centroids in Sequences A and B, and two randomly-selected 256×256-pixel subregions in Sequences C and D, respectively (Fig. 3). Only those cells that appear in the initial frame of each sequence and their children were tracked. A cell trajectory is *valid* only if it followed the *same* cell through all frames that the cell is visible. The operator also manually identified all mitosis events. We compared the tracking results produced by the current and the previous systems, as shown in Table I.

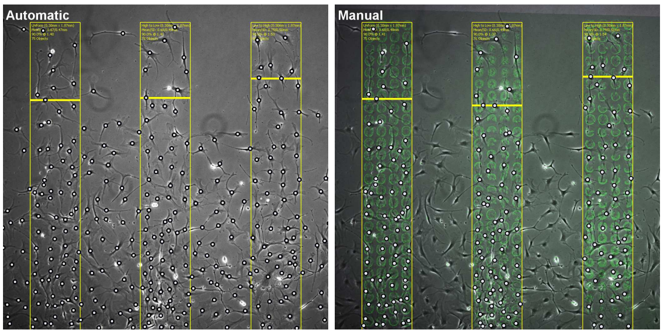

Fig. 5. Automatic versus manual detection of C3H10T1/2 stem cell populations in response to inkjet-printed patterns of HB-EGF growth factor. The cells are seeded near the bottom of the image. Yellow rectangles indicate the locations of grow factor patterns. The concentration distributions of the patterns are (from left to right, bottom to top): uniform, high-to-low, and low-to-high. The thick yellow lines indicate the migration front defined as the 90th percentile boundary of the cell population.

We visually compared the current tracking results with those produced by the previous system [12] for more than 30 sequences of AE stem cells. The new system showed superior robustness in handling long-term occlusion and against cell detection error. Fig. 4 shows an example where cell 116 is occluded by cell 47 in frame 36 and reappeared in frame 46. The new system (top row) correctly recovered the trajectory of cell 116 after occlusion, whereas the previous system (bottom row) switched the identities of cells 47 and 116 in frame 16, detected a false mitosis in frame 36, and eventually lost cell 47 after frame 36.

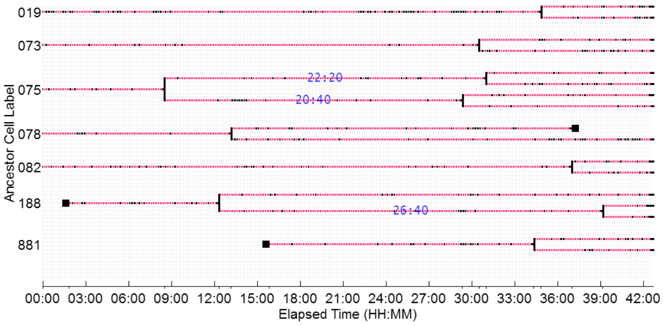

Fig. 6. Color-coded lineage map for selected AE cells. Red line segments indicate the relative migration distances between frames. Black squares indicate cell entrance/departure. Blue text shows division time.

Another application of the tracking system that is valuable to stem cell research is to automatically reconstruct cell lineage maps. We used the system to construct the lineages for the whole population of AE cells. Fig. 6 shows a sample set of the lineage trees with cells undergoing multiple divisions.

We also applied our system to study the migratory responses of C3H10T1/2 mouse embryonic fibroblast cell populations on various inkjet-printed patterns of HB-EGF growth factors. Three growth factor patterns of different concentration

[1] A matrix is totally unimodular if the determinant of any square submatrix takes one of the values in {-1, 0, 1}.

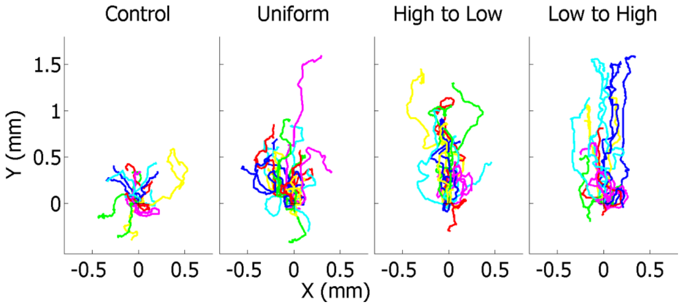

Fig. 7. Migration paths of C3H10T1/2 cells in response to differently distributed HB-EGF patterns traced every 10 minutes for 62 hours. The paths are registered such that they share common starting points.

distributions (uniform, high-to-low gradient, and low-to-high gradient) are printed in each image field using a bio-inkjet printer [13]. Example of automatic and manual cell detection results are shown in Fig. 5 for the purpose of qualitative comparison. In Fig. 7, we plotted the migration paths of 40 randomly selected cells on each pattern registered to share a common starting point. The plot reveals an obvious difference in the migration directivities in response to different patterns.

IV. Conclusion and Future Work

We developed and validated an automated system capable of tracking thousands of individual cells in dense cell populations in phase contrast microscopy image sequences. The system incorporated spatiotemporal track linking and a biologically relevant motion filter, and achieved performance boosts of up to 9% compared to its predecessor with nominal computational overhead. We plan to incorporate more effective segmentation algorithms and graphical models to cope with more complex cell shapes and intercellular interactions.

Acknowledgements

We would like to thank Eric D. Miller, Dr. Phil G. Campbell, and Dr. Lee E. Weiss for performing cell culture, bioprinting, and timelapse imaging. This work was supported partially by the National Institute of Health Grants R01 EB007369-01 and R01 EB0004343-01, and the Pennsylvania Infrastructure Technology Alliance Grant 1C76 HF 00381-01.

References

[1] O. Al-Kofahi, R. J. Radke, S. K. Goderie, Q. Shen, S. Temple, and B. Roysam, "Automated cell lineage construction: A rapid method to analyze clonal development established with murine neural progenitor cells," *Cell Cycle*, vol. 5, no. 3, pp. 327–335, Feb. 2006.

[2] D. Ambrosi, "Cellular traction as an inverse problem," *SIAM J. Appl. Math*, vol. 66, no. 6, pp. 2049–2060, Oct. 2006.

[3] M. Berkelaar, J. Dirks, K. Eikland, and P. Notebaert, "lpsolve: A mixed integer linear programming (MILP) solver," 2007. [Online]. Available: http://sourceforge.net/projects/lpsolve

[4] H. A. P. Blom, "An efficient filter for abruptly changing systems," in *Proc. 23rd IEEE Conference on Decision and Control*, Dec. 1984, pp. 656–658.

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

[6] O. Debeir, P. V. Ham, R. Kiss, and C. Decaestecker, "Tracking of migrating cells under phase-contrast video microscopy with combined mean-shift processes," *IEEE Transactions on Medical Imaging*, vol. 24, pp. 697–711, Jun. 2005.

[7] A. Doucet and B. Ristic, "Recursive state estimation for multiple switching models with unknown transition probabilities," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 3, pp. 1098–1104, Jul. 2002.

[8] A. Genovesio, T. Liedl, V. Emiliani, W. J. Parak, M. Coppey-Moisan, and J.-C. Olivo-Marin, "Multiple particle tracking in 3-D+t microscopy: Method and application to the tracking of endocytosed quantum dots," vol. 15, pp. 1062–1070, May 2006.

[9] R. E. Helmick, W. D. Blair, and S. A. Hoffman, "Fixed-interval smoothing for markovian switching systems," *IEEE Transactions on Information Theory*, vol. 41, no. 6, pp. 1845–1855, Nov. 1995.

[10] V. P. Jilkov and X. R. Li, "Online bayesian estimation of transition probabilities for markovian jump systems," *IEEE Transactions on Signal Processing*, vol. 52, no. 6, pp. 1620–1630, Jun. 2004.

[11] K. Li, M. Chen, and T. Kanade, "Cell population tracking and lineage construction with spatiotemporal context," in *Proc. 10th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*. Springer, Oct. 2007, in press.

[12] K. Li, E. D. Miller, L. E. Weiss, P. G. Campbell, and T. Kanade, "Online tracking of migrating and proliferating cells imaged with phase-contrast microscopy," in *Proc. IEEE Conf. Comp. Vision and Patt. Recog. Workshop*, Jun. 2006, p. 65.

[13] E. D. Miller, G. W. Fisher, L. E. Weiss, L. M. Walker, and P. G. Campbell, "Dose-dependent cell growth in response to concentration modulated patterns of FGF-2 printed on fibrin," *Biomaterials*, vol. 27, pp. 2213–2221, Apr. 2006.

[14] D. Padfield, J. Rittscher, T. Sebastian, N. Thomas, and B. Roysam, "Spatio-temporal cell cycle analysis using 3D level set segmentation of unstained nuclei in line scan confocal fluorescence images," in *Proc. IEEE Intl. Symp. Biomed. Imaging*, Apr. 2006, pp. 1036–1039.

[15] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications, Jan. 1998.

[16] Y. Shi and W. C. Karl, "Real-time tracking using level sets," in *Proc. IEEE Conf. Comp. Vision and Patt. Recog.*, vol. 2, Jun. 2005, pp. 34–41.

[17] S. Sternberg, "Biomedical image processing," *IEEE Computer*, vol. 16, no. 1, pp. 22–34, Jan. 1983.

[18] F. Yang, M. A. Mackey, F. Ianzini, G. Gallardo, and M. Sonka, "Cell segmentation, tracking, and mitosis detection using temporal context," in *Proc. Medical Image Computing and Comp.-Assisted Intervention*, J. S. Duncan and G. Gerig, Eds., vol. I, Oct. 2005, pp. 302–309.

[19] S. C. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 884–900, Sep. 1996.