Medical Image Analysis 12 (2008) 546–566

Contents lists available at ScienceDirect

Medical Image Analysis

journal homepage: www.elsevier.com/locate/media

Cell population tracking and lineage construction with spatiotemporal context

Kang Li [a,*], Eric D. Miller [a], Mei Chen [b], Takeo Kanade [a], Lee E. Weiss [a], Phil G. Campbell [a]

[a] Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, United States
[b] Intel Research Pittsburgh, 4720 Forbes Avenue, Suite 410, CM2, Pittsburgh, PA 15213, United States

ARTICLE INFO

Article history:
Received 5 February 2008
Received in revised form 16 May 2008
Accepted 10 June 2008
Available online 18 June 2008

Keywords:
Cell tracking
Level set
Jump Markov systems
IMM filter
Quasi-Bayes estimation
Linear programming
Phase contrast
Time-lapse microscopy
Stem cell

ABSTRACT

Automated visual-tracking of cell populations *in vitro* using time-lapse phase contrast microscopy enables quantitative, systematic, and high-throughput measurements of cell behaviors. These measurements include the spatiotemporal quantification of cell migration, mitosis, apoptosis, and the reconstruction of cell lineages. The combination of low signal-to-noise ratio of phase contrast microscopy images, high and varying densities of the cell cultures, topological complexities of cell shapes, and wide range of cell behaviors poses many challenges to existing tracking techniques. This paper presents a fully automated multi-target tracking system that can efficiently cope with these challenges while simultaneously tracking and analyzing thousands of cells observed using time-lapse phase contrast microscopy. The system combines bottom-up and top-down image analysis by integrating multiple collaborative modules, which exploit a fast geometric active contour tracker in conjunction with adaptive interacting multiple models (IMM) motion filtering and spatiotemporal trajectory optimization. The system, which was tested using a variety of cell populations, achieved tracking accuracy in the range of 86.9–92.5%.

1. Introduction

Biological discovery and its translation into new clinical therapies are rapidly advancing through the use of combinatorial, high-throughput experimental approaches. Automated tracking of cell populations *in vitro* in time-lapse microscopy images enables high-throughput spatiotemporal measurements of a range of cell behaviors, including the quantification of *migration* (translocation), *mitosis* (division), *apoptosis* (death), as well as the reconstruction of cell *lineages* (mother–daughter relations). This capability is valuable for several areas including stem cell research, tissue engineering, drug discovery, genomics, and proteomics (Huang et al., 1999; Patrick and Wu, 2003; Braun et al., 2003; Al-Kofahi et al., 2006; Bao et al., 2006).

The automation of cell tracking faces many challenges. These challenges include: varying cell population densities due to cells dividing/dying and leaving/entering the field-of-view; complex cellular topologies (shape deformation, close contact, and partial overlap); and in particular, massive amounts of image data. As an example, we have been using computer-aided *bioprinting* to create complex patterned arrays of growth factors for inducing and directing the fates of whole cell populations (Weiss et al., 2005; Campbell et al., 2005; Miller et al., 2006; Phillippia et al., 2008). To quantify how these patterns regulate cell behaviors over time and space requires time-lapse phase contrast microscopy to continuously record the cellular responses over extended periods (e.g., 5–10 days), while monitoring multiple experiments in parallel. This process routinely produces large datasets with low signal-to-noise ratios (Fig. 1). Typical experiments produce over 100 gigabytes (GB) of image data consisting of about 40,000 frames, with up to thousands of cells in each frame. Manual cell tracking in these images by an experienced microscopist can routinely take weeks of tedious work, while the results can be imprecise and subject to interobserver variability. Therefore, for efficiency and accuracy, automated cell tracking and analysis are required. A robust computer vision based system can address the automated tracking requirements. Previously reported cell tracking systems, however, do not address all the challenges, and are typically validated on short-term and/or small-scale experiments only.

In this paper, we present a fully automated multi-target tracking system that can successfully cope with the aforementioned challenges, and can simultaneously track hundreds to thousands of cells over the duration of a biological experiment. The system exploits a two-level design, integrating multiple collaborative modules. The lower level consists of a cell detector, a fast geometric active contour tracker, and an interacting multiple models (IMM) motion filter adapted for biological behaviors. The higher level is comprised of two trajectory management modules called the track compiler and the track linker.

* Corresponding author. Tel.: +1 412 268 4864; fax: +1 412 268 5570.
E-mail address: kangl@cmu.edu (K. Li).

1361-8415/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.media.2008.06.001

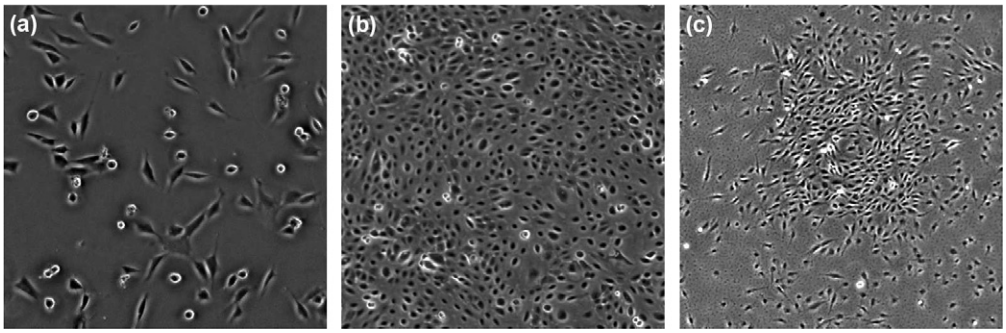

Fig. 1. Examples of phase contrast microscopy images of cell populations. (a) and (c) MG-63 human osteosarcoma cells. (b) Human amnion epithelial (AE) stem cell population. The images are cropped to 512 × 512 pixels.

The system has several advantages. First, the geometric active contour tracker simultaneously performs segmentation and data association by integrating image intensity, edge, motion, and shape information with a fast level set framework. Second, the IMM filter with online parameter adaptation enhances the tracking of varying cell dynamics, and provides the additional capability of motion pattern identification. Third, the spatiotemporal track linking approach makes the system capable of resolving complete or long-term occlusions. Finally, although multiple algorithms are integrated in our system, many of its parameters are estimated automatically, while the remaining ones are intuitive to set.

As an example application for the tracking system, we demonstrate its use to automatically measure stem cell lineages. This task requires long-term tracking of cell locations. The accurate segmentation of cell boundaries is an added benefit of our system for other applications, but it is not the emphasis of the results reported here.

2. Related work

An overview is presented below on the methods currently used for automated tracking of cells in time-series images. These methods can be classified as either *tracking by detection* or *tracking by model evolution*.

2.1. Tracking by detection

In the tracking by detection approach, cells are first detected in each frame based on intensity, texture, or gradient features (Al-Kofahi et al., 2006), and then the detected cells are associated between two or more consecutive frames, typically by optimizing certain probabilistic objective functions. This approach is computationally efficient and robust when cell density is low. However, tracking mitosis can be problematic (Kirubarajan et al., 2001), and segmentation errors generally increase with increasing cell density as a result of the inability to discriminate between multiple touching cells.

For one example, Bhanson et al. report on an automated system for measuring cell motility and proliferation over time, but the system is unable to distinguish between cells that are not well-separated (Bahnson et al., 2005). As another example, Al-Kofahi et al. used a seeded watershed method (Vincent and Soille, 1991) to detect cells, which can, to some degree, distinguish touching cells. They then perform feature-based cell matching between two frames to determine cell trajectories and lineage (Al-Kofahi et al., 2006). They acknowledged that tracking becomes difficult as multiple cells merge into a dense blob, and they did not address cells leaving or entering the image. They also suggest that their methodology could be implemented in real-time since tracking by detection in general requires low computational overhead. In yet another example, Yang et al. used watershed and mean shift (Cheng, 1995) to segment fluorescence-labeled nuclei to track cell cycle progression (Yang et al., 2005b), but did not address cell lineage construction.

Another popular set of techniques (Smal et al., 2006; Smal et al., 2007; Godinez et al., 2007) is based on particle filtering (Doucet and Ristic, 2002), which elegantly integrates detection and data association in a Bayesian probabilistic framework. While these techniques are well-suited for particle tracking in fluorescence microscopy image sequences, their extension to cell tracking in phase contrast microscopy images is not straightforward.

2.2. Tracking by model evolution

In the tracking by model evolution approach, parametric and non-parametric model-based representations of cell appearances or shapes are evolved from frame to frame (Debeir et al., 2005; Zimmer et al., 2002; Zimmer and Olivo-Marin, 2005; Mukherjee et al., 2004) or in spatiotemporal volumes (Padfield et al., 2006a,b, 2008) in order to keep track of moving cells over time.

Techniques based on parametric active contour models have the potential to produce better estimates of cell morphologies, but must be adapted to handle cell–cell contacts and mitosis at the cost of reduced computational efficiency. For example, Zimmer et al. adapted the classic "snake" model to track cells by adding repulsive forces between snakes to handle close contact of cells and incorporating "topological operators" to handle cell division (Zimmer et al., 2002; Zimmer and Olivo-Marin, 2005). However, the computational overhead can be prohibitively expensive for tracking a large number of cells. Debeir et al. considered a simplified problem of tracking only the centroid positions, but not the boundaries of the cells (Debeir et al., 2005), which permits a mean shift based model (Cheng, 1995) to be used. However, similar to the snakes model, this model cannot handle cell divisions. As a remedy, the authors proposed to track backwards (from the last frame to the first), which simplified the problem but made the tracking unsuitable for real-time processing during image acquisition. Moreover, this method requires manual identification of cell centroids for initialization, and cannot automatically incorporate new cells entering the field-of-view.

Geometric active contour models implemented via the level set method (Osher and Sethian, 1988) have recently been investigated for cell tracking applications (Mukherjee et al., 2004; Yang et al., 2005a; Dufour et al., 2005; Padfield et al., 2006a,b, 2008). Geometric models are generally deemed to be more powerful representations than parametric models. However, the use of level sets for cell tracking had been dismissed before because, in its classic form, it does not prevent two contacting boundaries from merging (i.e., it will fuse multiple cells that move into close contact as one object), and it is computationally expensive. Most previous studies on level set cell tracking either did not consider contacting cells (Mukherjee

et al., 2004), or resorted to off-line post-processing to correct cell fusions (Yang et al., 2005a; Bunyak et al., 2006). These methods made little use of temporal contextual information. Padfield et al. approached tracking as a spatiotemporal segmentation task (Padfield et al., 2006a,b, 2008). This method can potentially yield more accurate cell segmentation than frame-by-frame processing. However, it requires additional post-processing to separate cell clusters and to produce cell trajectories (Padfield et al., 2008). It is also more computational and memory intensive than frame-by-frame sequential processing.

To partially address the problem of cell fusion, Zhang et al. proposed a "coupled geometric active contours" model (Zhang et al., 2004), which represents each cell by a separate level set function, and enforces a coupling constraint that prevents different contours from overlapping. Dufour et al. further extended this approach to 3D for tracking fluorescent cells (Dufour et al., 2005). This approach is constrained both by computer memory and computing power, which makes it unsuitable to handle a large number of cells.

Another approach is to incorporate topological constraints, which explicitly prohibit cell merging while allowing cell division. Although the topological control of level sets has been studied extensively for image segmentation problems (Han et al., 2003; Ségonne, 2005), its potential for tracking has yet to be fully exploited. The application of topology-constrained level set methods to cell tracking was first reported by our group (Kanade and Li, 2005; Li et al., 2006) and more recently, by Nath et al. (2006). The key idea shared by Nath et al.'s method and ours is to label the contours of different cells using different "colors", and to prohibit the contours with distinct colors from merging. The distinction between the two approaches, however, lies in the way this coloring mechanism is implemented. Nath et al.'s method relies on the *four-color theorem* (Appel and Haken, 1977a,b), which states that no more than four colors are required to paint a set of disjoint regions on a plane such that no two adjacent regions share the same color. Based on this theorem, their approach applies planar graph-vertex coloring to distinguish cell contours, and requires four level set functions to handle an arbitrary number of cells while preventing cell merging. On the other hand, our approach only requires one level set function, along with a *region labeling map* that evolves together with the level set function. Besides being more memory-efficient, the region labeling map is conveniently utilized to store the identity of each cell, which facilitates cell tracking.

3. Methods

Our tracking system integrates five modules (Fig. 2), including: (1) *cell detector*, which detects and labels candidate cell regions in the input image utilizing region, edge, and shape information; (2) *cell tracker*, which propagates candidate cell regions and identities across frames using a fast topologically constrained geometric active contour algorithm; (3) *motion filter*, which performs prediction and filtering of the cell motion dynamics using a biologically relevant adaptive interacting multiple models (IMM) filter; (4) *track compiler*, which generates intermediate result called *track segments* by fusing the output from the above modules, and judging on what is and what is not physically possible; and (5) *track linker*, which oversees the entire tracking history and establishes the complete cell trajectories and lineages. To achieve robust and versatile cell tracking, our system combines the advantages of both tracking by detection and tracking by model evolution approaches (Section 2), while mitigating against their disadvantages. Before dwelling on each module, we provide an overview of the system workflow and establish notations.

3.1. System workflow

Our system starts by processing the input images sequentially. The system output is a complete spatiotemporal history of the cell trajectories, including cell centroid positions, cell migration velocities, shape, and intensity parameters for every cell, as well as the parent–child relations between cells. For each cell, the system may generate multiple *track segments* as intermediate output. Each track segment is associated with a unique positive-integer label n. Each cell is identified using the label of its first track segment.

To initialize tracking, the cell detector detects all candidate cells in the first frame $I_0(x,y)$ and generates an initial cell region labeling map $\psi_0(x,y)$, where $\psi_0(x,y) = n$ if pixel (x,y) is part of cell n, and $\psi_0(x,y) = 0$ if (x,y) belongs to the background. Subsequently, for each frame $I_k(x,y)$, $k = 1,2,\ldots,K$:

Step 1: The cell detector segments cell regions in the image using a combination of region-based and edge-based approaches. The output is a binary map of cell regions, denoted $\chi_k(x,y)$. Each connected foreground component in $\chi_k(x,y)$ is considered 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 extended a fast geometric active contour algorithm (Shi and Karl, 2005b) to segment cell regions and to propagate the corresponding cell labels. First, a level set function $\phi_k(x,y)$ is initialized using $\psi_{k-1}(x,y)$. Then, ϕ and ψ are evolved together to minimize an "energy" functional that combines a region competition term (Zhu and Yuille, 1996), a geodesic edge term (Caselles et al., 1997), and a motion term based on the distribution of the predicted cell position

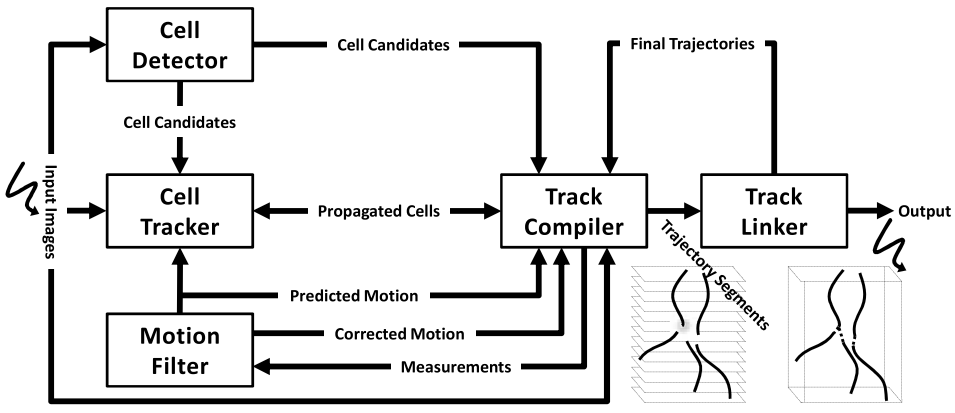

Fig. 2. System overview.

from the motion filter. Topological constraints are incorporated to the level set evolution to prevent contours that represent different cells from merging. The output is the propagated cell labeling map for frame k, denoted $\hat{\psi}_k(x,y)$.

Step 3: The track compiler compares the outputs of the cell detector and the cell tracker, and takes one of the following actions: creates a new or daughter track segment, or updates an existing track, or terminates a track. For continuing track segments, the track compiler calls on the motion filter to update the cell motion state in frame k, and to predict its state for frame $k+1$. The predictions will be useful for the track linking process (step 4), as well as for the level set evolution for the subsequent frame (step 5). For new track segments, initial motion states are initialized based on quantities measured from the corresponding cell regions. The output of this step includes the track segments and an updated region labeling map $\psi_k(x,y)$.

Step 4: The track linker examines all track segments up to frame k, and detects whether two or more track segments 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 each module of this system.

3.2. Cell detection

Cells in phase contrast microscopy normally appear as dark regions surrounded by bright halo artifacts, except for mitotic (dividing) or apoptotic (dying) cells, which appear rounder and brighter than the other cells. Consequently, the cell detector combines two approaches: (1) *region-based* detection, which employs a grayscale morphological filter and the level set method to extract non-mitotic and non-apoptotic cells; and (2) *edge-based* detection, which detects mitotic and apoptotic cells based on image edges, as well as a set of shape and appearance criteria. The outputs of the two approaches are combined to yield a binary image $\chi(x,y)$: $\Omega \to \{0,1\}$, in which each non-zero connected component is considered a cell candidate. The steps of cell detection are illustrated in Fig. 3, and detailed in the next two subsections.

3.2.1. Region-based cell detection

The region-based cell detection approach consists of two steps: (1) morphological pre-segmentation, the result of which is used to estimate the intensity distributions of cells and background; and (2) level set segmentation, which achieves more robust cell localization.

The *rolling-ball* filter (Sternberg, 1983) is applied to pre-segment the non-mitotic and non-apoptotic cells. The rolling-ball filter simulates rolling a ball beneath the intensity profile of an image, removing the peaks that are untouchable by the ball surface. It is a grayscale morphological filter that is related to the classical top-hat transformation (Meyer, 1979) by

$$I - \text{rollball}(I,r) = \text{tophat}(I, \text{ball}_\text{r}),$$

where r is the radius of the rolling-ball, and ball_r is a non-flat half ball-shaped structuring element with radius r. The parameter r is set roughly equal to the average radius of cells to be detected. To apply the rolling-ball filter, the input image is first inverted such that the cell interior appears brighter than the surrounding halo. The operation $\hat{I}_\text{r} = \bar{I} - \text{rollball}(\bar{I},r)$ on the inverted image $\bar{I}$ will produce an image $\hat{I}_\text{r}$ with cell regions solidified and highlighted. Otsu thresholding (Otsu, 1979) is then applied on $\hat{I}_\text{r}$ to obtain a binary mask $\hat{\chi}_\text{r}$ of the cell regions.

The binary mask $\hat{\chi}_r$ constitutes a rough pre-segmentation of the image, which enables us to obtain two histograms: a cell histogram h_C, and a background histogram h_B. With these histograms, a Bayesian maximum *a posteriori* probability (MAP) classifier can be implemented via the following test:

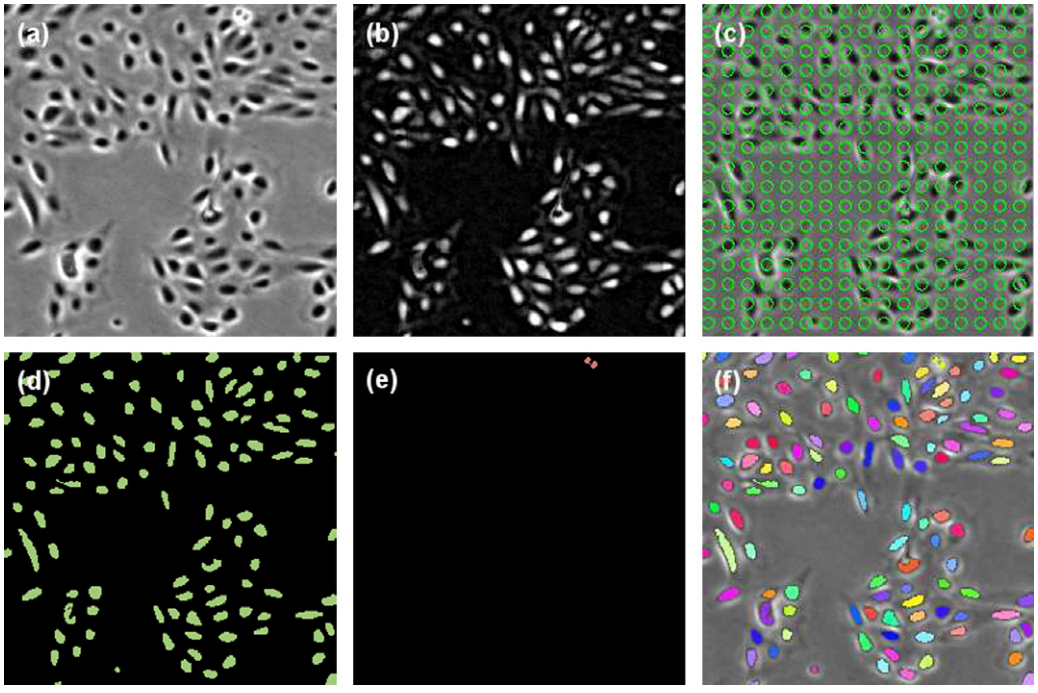

Fig. 3. Illustration of cell detection steps. Shown are the human amnion epithelial (AE) stem cells. (a) Original image I. (b) Result of rolling-ball filtering $\hat{I}_\text{r}$. (c) Green circles represent initial level set contour for region-based cell detection. (d) Region-based cell detection output χ_r. (e) Edge-based cell detection output χ_e. (f) Combined output with each cell region shown in a different color.

$$\text{Classify a pixel } I(x,y) \text{ as } \begin{cases} \text{cell}, & \text{if } h_C(I(x,y)) > h_B(I(x,y)), \\ \text{background}, & \text{otherwise}. \end{cases} \tag{1}$$

To understand this, recall that a MAP classifier can be expressed via the Bayes rule as: $\arg\max_c p(c|I(x,y)) = \arg\max_c p(I(x,y)|c)p(c)$, where $c \in \{C,B\}$ is the class label. The following relation holds:

$$\frac{p(I(x,y)|C)p(C)}{p(I(x,y)|B)p(B)} = \frac{h_C(I(x,y))/m_C \cdot m_C/m}{h_B(I(x,y))/m_B \cdot m_B/m} = \frac{h_C(I(x,y))}{h_B(I(x,y))},$$

where $m_C = \text{sum}(h_C)$ is the number of pixels in the cell regions, $m_B = \text{sum}(h_B)$ is the number of pixels in the background, and $m = m_C + m_B$ is the total number of pixels in the image.

Instead of a direct application of Eq. (1), the MAP classifier is implemented via the level set method (Osher and Sethian, 1988), which is less sensitive to noise and yields robust segmentation. The method will be elaborated in Section 3.3. After level set segmentation, an *a priori* size constraint is imposed by removing the connected components with sizes smaller than $s_{\min}$ pixels or larger than $s_{\max}$ pixels. The output is a binary map of segmented cell regions χ_r.

3.2.2. Edge-based cell detection

The edge-based cell detection approach aims to detect mitotic and apoptotic cells, which appear rounder and brighter than the other cells. This approach consists of three steps. First, the Canny edge detector (Canny, 1986) is applied to compute an *edge map* of the image. Then, the regions that are enclosed by edges are located and filled. The regions whose sizes fall outside the valid range of $[s_{\min}, s_{\max}]$ (Section 3.2.1) are discarded. Then, for each remaining region, the mean pixel intensity μ_o in a w-pixel-wide rim of the region and the *eccentricity* are computed. The eccentricity is measured by fitting an ellipse to the region using second-moment matching and computing the ratio of the distance between the foci of the ellipse and its major axis length. Finally, the regions with eccentricities smaller than 0.95 and $\mu_o > \mu_N + \sigma_N$ are selected as cell regions, where μ_N and σ_N are, respectively, the mean and standard deviation of the pixel intensities in a neighborhood of radius r_N surrounding the region. The parameters w and r_N are set to $w = \max(1, r/3)$ and $r_N = 4r$ in our implementation, where r is defined in Section 3.2.1. The output is a binary map χ_e of mitotic and apoptotic cells.

3.3. Geometric active contour cell tracker

Because cells are highly deformable objects and may divide over time, we choose to represent cell boundaries using an implicit contour model, commonly known as the *geometric active contour* model. In this model, the boundary of each cell is considered as a closed contour C in the image domain $\Omega \in \mathbb{R}^2$. Its contour is represented as the zero level line of a time-dependent embedding function $\phi : \Omega \times [0,T] \to \mathbb{R}$, where

$$C(t) = \{(x,y) \in \Omega | \phi(x,y,t) = 0\},$$

such that $\phi(C(t),t) = 0$ at any time. Evolving the embedding function ϕ over time is an elegant method to keep track of the motion of the boundary, including its topological changes such as splitting and merging.

Among various approaches to evolve a geometric active contour, the most popular one is the level set method (Osher and Sethian, 1988), in which one evolves the embedding function (or the *level set function*) ϕ according to an appropriate partial differential equation (PDE). The PDE is usually derived as the Euler–Lagrange equation:

$$\frac{\partial \phi}{\partial t} = -\frac{\partial E(\phi)}{\partial \phi}, \tag{2}$$

which minimizes an application-specific "energy" functional $E(\phi)$.

For cell tracking, the energy functional is constructed such that its minimization leads to the propagation of cell boundaries from frame $k-1$ to frame k. The propagated cell boundaries should not only match the cell appearances in frame k, but also be consistent with the cell motion pattern. The energy consists of a weighted sum of three terms, which are derived from: (1) the image region statistics (E_{region}); (2) the image edges (E_{edge}); and (3) the prediction of cell motion (E_{motion}):

$$E = E_{\text{region}} + w_{\text{edge}} E_{\text{edge}} + w_{\text{motion}} E_{\text{motion}}, \tag{3}$$

where $w_{\text{edge}} \geqslant 0$ and $w_{\text{motion}} \geqslant 0$ are weighting coefficients. The dependency on ϕ is omitted in the notation for simplicity. The energy terms implicitly depend on ϕ, which will be further explained in Section 3.3.3.

$\mathbf{N}_{k-1}$ is the set of cells to be propagated from frame $k-1$ to frame k. Each cell $n \in \mathbf{N}_{k-1}$ occupies the region $\Omega_n \subset \Omega$, enclosed by its boundary $C_n = \{(x,y) | (x,y) \in \partial\Omega_n\}$. The region that is not occupied by cells is the background, denoted by Ω_0.

The region energy E_{region} is based on the Bayesian region-competition framework (Zhu and Yuille, 1996; Cremers et al., 2007):

$$E_{\text{region}} = -\sum_{n \in \mathbf{N}_{k-1}} \iint_{\Omega_n} \log p(\Omega_n | I_k(x,y)) \mathrm{d}x\,\mathrm{d}y - \iint_{\Omega_0} \log p(\Omega_0 | I_k(x,y)) \mathrm{d}x\,\mathrm{d}y + \frac{\nu}{2} \sum_{n \in \mathbf{N}_{k-1}} \int_{C_n} \mathrm{d}l. \tag{4}$$

It represents the joint posterior probability that each pixel in frame k belongs to a certain propagated region (first two terms), subject to a penalty on the total length of the region boundaries (third term). The parameter ν specifies the strength of the penalty.

The edge energy E_{edge} measures the *edgeness* along the region boundaries. It is formulated following the approach of geodesic active contours (Caselles et al., 1997; Goldenberg et al., 2001), which can be interpreted as the length of a curve in a Riemannian space whose metric is induced by the image edges:

$$E_{\text{edge}} = \sum_{n \in \mathbf{N}_{k-1}} \int_{C_n} e(C_n) \mathrm{d}l. \tag{5}$$

The function $e(\cdot)$ is the edgeness metric, which is ideally zero at the locations of image edges, and takes on larger values elsewhere.

The motion energy E_{motion} represents the joint probability that the cell regions reside at the locations predicted by their respective motion filters:

$$E_{\text{motion}} = -\sum_{n \in \mathbf{N}_{k-1}} \iint_{\Omega_n} \tau + \log \hat{p}_{k|k-1}(x,y|\Omega_n) \mathrm{d}x\,\mathrm{d}y. \tag{6}$$

Here, $\tau > 0$ is a size-constraint parameter, which is necessary because $\log \hat{p}_{k|k-1}$ is non-positive everywhere, and because the contour that minimizes E_{motion} will enclose the entire image if $\tau = 0$. In addition to providing motion context, the distribution $\hat{p}_{k|k-1}(x,y|\Omega_n)$ serves as an implicit shape prior. The definition of $\hat{p}_{k|k-1}(x,y|\Omega_n)$ will be further discussed in Section 3.3.3.

For tracking $N > 1$ cells in parallel, one key issue is how to uniquely identify each cell region using the implicit contour representation. A straightforward solution is to utilize N level set functions (Brox and Weickert, 2006; Mansouri et al., 2006), each of which represents one cell. This solution, however, is highly inefficient for simultaneously tracking thousands of cells. Inspired by the approaches in Feng et al. (2001) and Shi and Karl (2005b), we chose to represent all regions using one level set function ϕ, and to keep track of the identities of cell regions by evolving the region labeling function ψ (Section 3.1) simultaneously with the level set function ϕ. The implementation of our approach will be detailed in the next three sections.

3.3.1. Euler–Lagrange equations

The first step towards tracking is to rewrite the energy terms such that they explicitly depend on ϕ. We introduce three auxiliary functions: the region indicator function $R_n(\cdot)$, the Heaviside function $H(\cdot)$, and the one-dimensional Dirac measure $\delta(\cdot)$, defined as follow:

$$R_n(\psi)=\begin{cases}1, & \text{if } \psi=n,\\ 0, & \text{if } \psi\neq n,\end{cases}\quad H(\phi)=\begin{cases}1, & \text{if } \phi\geqslant 0,\\ 0, & \text{if } \phi<0,\end{cases}\quad \delta(\phi)=\frac{\mathrm{d}}{\mathrm{d}\phi}H(\phi),$$

where $\phi<0$ applies to points inside the cell regions, and $\phi>0$ for points in the background. The energy terms can now be rewritten as

$$\begin{aligned}E_{\text{region}}=&-\sum_{n\in\mathbf{N}_{k-1}}\iint_{\Omega}R_n(\psi)(1-H(\phi))\log p(\Omega_n|I_k)\mathrm{d}x\mathrm{d}y\\&-\iint_{\Omega}H(\phi)\log p(\Omega_0|I_k)\mathrm{d}x\mathrm{d}y+\frac{\nu}{2}\sum_{n\in\mathbf{N}_{k-1}}\iint_{\Omega}\delta(\phi)|\nabla\phi|\mathrm{d}x\mathrm{d}y,\end{aligned}\tag{7}$$

$$E_{\text{edge}}=\int_{\Omega}\delta(\phi)|\nabla\phi|e\,\mathrm{d}x\mathrm{d}y,\tag{8}$$

$$E_{\text{motion}}=\sum_{n\in\mathbf{N}_{k-1}}\iint_{\Omega}R_n(\psi)(1-H(\phi))(\tau+\log\hat{p}_{k|k-1}(x,y|\Omega_n))\mathrm{d}x\mathrm{d}y.\tag{9}$$

Herein, the dependency on (x,y) is omitted in the notations for simplicity.

Then, by computing the first variation $\partial E(\phi)/\partial\phi$ and by substituting it into Eq. (2), the Euler–Lagrange equation for minimizing the energy can be obtained. The equation can be written in the following standard form:

$$\frac{\partial\phi}{\partial t}=-F\delta(\phi),\quad \text{where } F=F_{\text{region}}+w_{\text{edge}}F_{\text{edge}}+w_{\text{motion}}F_{\text{motion}}.\tag{10}$$

The *speed* functions F_{region}, F_{edge}, and F_{motion} are defined as

$$F_{\text{region}}=\sum_{n\in\mathbf{N}_{k-1}}\log p(\Omega_n|I_k)R_n(\psi)-\log p(\Omega_0|I_k)+\frac{\nu}{2}\kappa,\tag{11}$$

$$F_{\text{edge}}=\nabla e\cdot\nabla\phi/|\nabla\phi|+e\kappa,\tag{12}$$

$$F_{\text{motion}}=\sum_{n\in\mathbf{N}_{k-1}}(\log\hat{p}_{k|k-1}(x,y|\Omega_n)-\tau)R_n(\psi),\tag{13}$$

where

$$\kappa=\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}\tag{14}$$

is the mean curvature.

3.3.2. Contour merging avoidance by topology constraints

The topological flexibility of the implicit contour representation not only facilitates the tracking of cell divisions, but also permits the merging of contacting objects. This may cause two adjacent objects in one frame to falsely merge into one object in the next frame.

In the context of cell tracking, the merging of multiple cells would signify cell fusion. While cell fusion occurs in specific cell types (e.g., activated macrophages and osteoclasts), it does not normally occur for the cell types studied in this paper, nor in many other studies (Zimmer et al., 2002; Mukherjee et al., 2004; Zhang et al., 2004; Yang et al., 2005a; Zimmer and Olivo-Marin, 2005; Debeir et al., 2005; Al-Kofahi et al., 2006; Bunyak et al., 2006; Nath et al., 2006). To prevent false cell fusion, it is important to incorporate a topological constraint that permits division but prohibits merging.

To introduce the topological constraint, we borrow the concept of *topological numbers* from digital topology (Han et al., 2003). Let $N_8(x,y)$ be the set of eight neighbors of pixel (x,y). The topological number of (x,y) with respect to the cell region Ω_n ($n>0$), denoted $T_n(x,y)$, is the number of four-connected components in the set $\Omega_n\cap N_8(x,y)$. Similarly, the topological number of (x,y) with respect to the background Ω_0, denoted $T_0(x,y)$, is the number of eight-connected components in the set $\Omega_0\cap N_8(x,y)$. Let $o(x,y)$ denote the number of cell regions that overlap with $N_8(x,y)$. Then, the relaxed topological number (Shi and Karl, 2005b) for pixel (x,y) is defined as

$$T_{\text{r}}(x,y)=\min[o(x,y),\max(T_n(x,y),T_0(x,y))].$$

The boundaries of two different cell regions can merge only if the level set function changes sign from positive to negative at a point (x,y) with $T_{\text{r}}(x,y)>1$. By detecting the points at which $T_{\text{r}}>1$, and preventing the level set function from changing sign at these points during the contour evolution, merging of different cell regions can be effectively prevented.

3.3.3. Fast implementation

Traditional implementations of the level set method require evaluating PDEs (e.g., Eq. (10)) using numerical methods (e.g., finite difference), which is computationally expensive. Among various approaches to speed up the computation (Cates et al., 2004; Lefohn et al., 2004; Pan et al., 2006), the fast two-cycle algorithm proposed in Shi and Karl (2005a,b) is chosen, since it achieves near real-time tracking speed, allows straightforward incorporation of topological constraints, and is easy to implement.

The algorithm evolves a contour iteratively by operations as simple as switching elements between two linked lists, L_{in} and L_{out}, which keep track of the points adjacent to the contour. This approach can be viewed as an extreme case of the narrow-band scheme with a two-pixel bandwidth (Chopp, 1993; Sethian, 1999). For tracking N contours, $2N$ linked lists are initialized from the region labeling map $\psi(x,y)$ according to:

$$\begin{aligned}L_{\text{out}}(n)&=\{\mathbf{x}|\psi(\mathbf{x})=n,\exists\mathbf{x}^*\in N_4(\mathbf{x})\text{ where }\psi(\mathbf{x}^*)\neq n\},\\L_{\text{in}}(n)&=\{\mathbf{x}|\psi(\mathbf{x})=n,\exists\mathbf{x}^*\in N_4(\mathbf{x})\text{ where }\mathbf{x}^*\in L_{\text{out}}(n)\},\end{aligned}\tag{15}$$

where $\mathbf{x}\equiv(x,y)$. Accordingly, the level set function is defined as

$$\phi(x,y)=\begin{cases}3, & \text{if } (x,y)\text{ is an exterior pixel},\\ 1, & \text{if } (x,y)\in L_{\text{out}}(n),\ \forall n,\\ -1, & \text{if } (x,y)\in L_{\text{in}}(n),\ \forall n,\\ -3, & \text{if } (x,y)\text{ is an interior pixel},\end{cases}\tag{16}$$

which approximates a signed distance function.

Each contour-evolution iteration is performed in two cycles: an *update* cycle and a *regulation* cycle.

3.3.3.1. Update cycle. The update cycle evolves the contour according to the sign of a speed function $\widehat{F}$, which approximates F given in Eq. (10) with all curvature-dependent terms removed (i.e., the terms $\frac{\nu}{2}\kappa$ in (11) and $e\kappa$ in (12) are no longer necessary):

$$\widehat{F}(x,y)=\widehat{F}_{\text{region}}(x,y)+w_{\text{edge}}\widehat{F}_{\text{edge}}(x,y)+w_{\text{motion}}\widehat{F}_{\text{motion}}(x,y),\tag{17}$$

where

$$\widehat{F}_{\text{region}}(x,y)=\begin{cases}1, & \text{if } \chi(x,y)>0,\\ -1, & \text{otherwise},\end{cases}\tag{18}$$

$$\begin{aligned}\widehat{F}_{\text{edge}}(x,y)=&-(e(x+1,y)-e(x-1,y))(\phi(x+1,y)-\phi(x-1,y))\\&+(e(x,y+1)-e(x,y-1))(\phi(x,y+1)-\phi(x,y-1)),\end{aligned}\tag{19}$$

$$\widehat{F}_{\text{motion}}(x,y)=\sum_{n\in\mathbf{N}_{k-1}}(\log\mathcal{N}(x,y|\hat{\mathbf{z}}_{n,k|k-1},\mathbf{S}_{n,k-1})-\tau)R_n(\psi(x,y)).\tag{20}$$

The region speed $\widehat{F}_{\text{region}}$ requires the cell candidate map $\chi(x,y)$ output from the cell detector. Recall from Section 3.2 that $\chi(x,y)$ is computed by combining two approaches: region-based detection and edge-based detection. In region-based detection, the level set algorithm is executed using a uniform lattice-of-circles initialization and the following speed function:

$$\widehat{F}^*(x,y) = h_{\text{C}}(I_k(x,y)) - h_{\text{B}}(I_k(x,y)), \tag{21}$$

which implements the MAP classifier given in Eq. (1). The output segmentation $\chi_{\text{r}}(x,y)$ is combined with the output from edge-based detection $\chi_{\text{e}}(x,y)$ by a binary *OR* operation to obtain $\chi(x,y)$.

The edge speed $\widehat{F}_{\text{edge}}$ is a central-difference approximation of the first term of Eq. (12). Inspired by Huang et al. (2004), we define the edgeness function $e(x,y)$ to be the Euclidean distance transform of the edge map of $I_k(x,y)$, which is produced by the Canny edge detector. This definition induces fewer local minima as opposed to the gradient-based definition in Caselles et al. (1997). The edge map is also utilized for edge-based cell detection (Section 3.2.2), hence this computation can be reused.

The function $\mathcal{N}(\cdot|\mathbf{z},\mathbf{S})$ in Eq. (20) denotes a bivariate normal distribution with mean $\mathbf{z}$ and covariance $\mathbf{S}$. The vector $\hat{\mathbf{z}}_{n,k|k-1}$ is the centroid position of cell n in frame k predicted by the motion filter, which will be explained further in Section 3.4. $\mathbf{S}_{n,k-1}$ is the *shape* matrix, computed as

$$\mathbf{S}_{n,k-1} = \text{cov}\{(x,y)|\psi_{k-1}(x,y) = n\}. \tag{22}$$

It can be considered as an elliptical approximation of the cell shape in frame $k-1$ by second-moment matching.

3.3.3.2. Regulation cycle. The regulation cycle provides smoothness regulation to the contour using local Gaussian filtering. This regulation has a similar effect as the curvature-dependent terms in Eqs. (11) and (12), but avoids the expensive computation of the curvature. This is because the curvature equals $\nabla^2\phi$ (i.e., the Laplacian of ϕ) when ϕ is a signed distance function ($|\nabla\phi| = 1$); and based on the theory of heat diffusion (Perona and Malik, 1990), evolving a function according to its Laplacian is equivalent to Gaussian filtering.

More detail of the implementation is provided in Appendix A with pseudocode. This algorithm is limited to a pixel-level accuracy unless the input image is interpolated. A pixel-level accuracy is adequate for our study, since our primary goal is to construct the cell trajectories over time, rather than to delineate the cell boundaries at a sub-pixel precision.

3.4. Interacting multiple models motion filter

A motion filter is the fundamental building block of many tracking systems (Ristic et al., 2004). It provides recursive estimations of the target states (such as position, speed, and acceleration) based on noisy measurements. Essential to any motion filter is a motion model that describes the target dynamics, and a measurement model that relates states to measurements. Traditional motion filters, such as the Kalman filter (Kalman, 1960) and the standard particle filter (Gordon et al., 1993), are bound to use only one motion model, which is inadequate for tracking biological cells because cell dynamics vary frequently with time. The interacting multiple models (IMM) filter (Blom, 1984), instead, is capable of incorporating multiple motion models in parallel, and it has been shown to be well-suited for biological object tracking (Genovesio et al., 2006).

Cell motions are assumed to consist of a finite number of modes. Each mode can be described by a linear model with additive Gaussian noise. The motion models and the measurement model are defined as

$$\text{Motion models}: \quad \mathbf{s}_k = \mathbf{F}^i\mathbf{s}_{k-1} + \mathbf{v}^i_{k-1}, \quad i \in \{1,\ldots,M\},$$

$$\text{Measurement model}: \quad \mathbf{z}_k = \mathbf{H}\mathbf{s}_k + \mathbf{w}_k.$$

Here, $\mathbf{s}_k$ is the state vector of a cell in frame k, which consists of the centroid position, velocity, and acceleration of the cell, i.e., $\mathbf{s}_k \equiv (x_k, \dot{x}_k, \ddot{x}_k, y_k, \dot{y}_k, \ddot{y}_k)'$. Note that the "prime" sign ($'$) denotes vector or matrix transposition. The corresponding measurement vector $\mathbf{z}_k \equiv (x_k, y_k)'$ contains the measured centroid position. $\mathbf{F}^i$ is the state transition matrix of model i, and $\mathbf{H}$ is the measurement matrix that relates states to measurements. $\mathbf{v}^i_{k-1}$ and $\mathbf{w}^i_k$ are the process and measurement noise vectors, which are uncorrelated zero-mean Gaussian processes with covariances $\mathbf{Q}^i$ and $\mathbf{R}$, respectively.

The IMM filter operates M Kalman filters in parallel, each of which is matched to a distinct motion model. It assumes that the transition between models is regulated by a finite-state Markov chain, with probability p_{ij} of switching from model i to model j in successive frames. However, rather than making hard commitments to any single model, it maintains a weighting among the models, which is determined as the probability of each model being correct given the current measurement. Hence, the optimal state estimate at any time instant is a mixture of Gaussian distributions. Each mixture component is the estimate from a Kalman filter, weighted by the posterior probability of the corresponding motion model. This leads to a mixture with exponentially growing number of components in time because of the branching of model switching hypotheses. To avoid the combinatorial explosion and make the computation tractable, the IMM filter approximates the mixture of Gaussians with a single Gaussian with equal mean and covariances.

The filtering recursion consists of two stages: *prediction* and *correction*. The prediction stage predicts the state $\hat{\mathbf{s}}_{k|k-1}$ at time k based on the state history up to time $k-1$; the correction stage generates a refined estimate $\hat{\mathbf{s}}_k$ by incorporating the newly arrived measurement $\mathbf{z}_k$. The mechanisms of the two stages are detailed below.

Prediction: Starting from M weights ρ^i_{k-1}, states $\hat{\mathbf{s}}^i_{k-1}$ and covariances $\mathbf{\Sigma}^i_{k-1}$ from the previous iteration, the mixed initial condition is computed:

$$\hat{\mathbf{s}}^{0j}_{k-1} = \sum_i \rho^{i|j}_{k-1}\hat{\mathbf{s}}^i_{k-1}, \tag{23}$$

$$\mathbf{\Sigma}^{0j}_{k-1} = \sum_i \rho^{i|j}_{k-1}[\mathbf{\Sigma}^i_{k-1} + (\hat{\mathbf{s}}^i_{k-1} - \hat{\mathbf{s}}^{0j}_{k-1})(\hat{\mathbf{s}}^i_{k-1} - \hat{\mathbf{s}}^{0j}_{k-1})'], \tag{24}$$

where $\rho^{i|j}_{k-1} = p_{ij}\rho^i_{k-1}/\rho^j_{k|k-1}$, and $\rho^j_{k|k-1} = \sum_i p_{ij}\rho^i_{k-1}$.

These are input to M Kalman filters to compute the state prediction $\hat{\mathbf{s}}^j_{k|k-1}$ and covariance $\mathbf{\Sigma}^j_{k|k-1}$:

$$\hat{\mathbf{s}}^j_{k|k-1} = \mathbf{F}^j\hat{\mathbf{s}}^{0j}_{k-1}, \tag{25}$$

$$\mathbf{\Sigma}^j_{k|k-1} = \mathbf{F}^j\mathbf{\Sigma}^{0j}_{k-1}(\mathbf{F}^j)' + \mathbf{Q}^j. \tag{26}$$

The combined state and covariance predictions can be determined by

$$\hat{\mathbf{s}}_{k|k-1} = \sum_j \rho^j_{k|k-1}\hat{\mathbf{s}}^j_{k|k-1}, \tag{27}$$

$$\mathbf{\Sigma}_{k|k-1} = \sum_j \rho^j_{k|k-1}[\mathbf{\Sigma}^j_{k|k-1} + (\hat{\mathbf{s}}^j_{k|k-1} - \hat{\mathbf{s}}_{k|k-1})(\hat{\mathbf{s}}^j_{k|k-1} - \hat{\mathbf{s}}_{k|k-1})']. \tag{28}$$

The predicted centroid positions $\hat{\mathbf{z}}_{k|k-1} = \mathbf{H}\hat{\mathbf{s}}_{k|k-1}$ of all cells are fed to the cell tracker to guide the level set evolution in frame k (see Section 3.3.3).

Correction: Given the predicted states, covariances, and measurement $\mathbf{z}_k$, the Kalman filters are used to obtain the updated state $\hat{\mathbf{s}}^j_k$ and covariance $\mathbf{\Sigma}^j_k$:

$$\hat{\mathbf{s}}_k^j = \hat{\mathbf{s}}_{k|k-1}^j + \mathbf{K}_k^j(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{s}}_{k|k-1}^j), \tag{29}$$

$$\mathbf{\Sigma}_k^j = \widehat{\mathbf{\Sigma}}_{k|k-1}^j - \mathbf{K}_k^j\mathbf{H}\widehat{\mathbf{\Sigma}}_{k|k-1}^j, \tag{30}$$

where $\mathbf{K}_k^j = \mathbf{\Sigma}_{k|k-1}^j\mathbf{H}'(\mathbf{H}\mathbf{\Sigma}_{k|k-1}^j\mathbf{H}' + \mathbf{R})^{-1}$ is the Kalman gain.

The likelihood that model j is activated in frame k is

$$\lambda_k^j = \exp\left[-\frac{1}{2}(\mathbf{y}_k^j)'(\mathbf{S}_k^j)^{-1}\mathbf{y}_k^j\right] \Big/ \sqrt{2\pi\det(\mathbf{S}_k^j)}, \tag{31}$$

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 $\mathbf{\Sigma}_k$ estimates can be computed by Eqs. (27) and (28), with $\rho_{k|k-1}^j$ replaced by $\rho_k^j = \rho_{k|k-1}^j\lambda_k^j \Big/ \left(\sum_i \rho_{k|k-1}^i\lambda_k^i\right)$.

To initialize the IMM filter, the system tracks each cell without motion filtering in the first three frames in which it appears. The measured cell centroid positions in these frames are used to initialize the cell state $\hat{\mathbf{s}}_0$. The initial model weights ρ_0^i are set to equal $1/M$ ($i \in \{1,\ldots,M\}$), indicating the initial complete uncertainty as to which model is more correct. The definitions of the remaining filter parameters will be discussed in Sections 3.4.1 and 3.4.2.

3.4.1. Motion models

To adapt the IMM filter for cell tracking, cell motion models need to be defined by specifying the system matrices $\mathbf{F}^i$ and $\mathbf{H}$. Inspired by Genovesio et al. (2006), we define four motion models ($M = 4$): random walk (RW), constant-velocity (CV), constant-acceleration (CA), and constant-speed circular turn (CT). They represent four typical modes of cellular motion: Brownian motion, constant-velocity migration, constant-acceleration migration, and turning. Compared to the approach of Genovesio et al., the circular turn model is a novel addition, since the amnion epithelial stem cells that we experimented with perform an interesting turning motion. Moreover, instead of interpreting the motion models as the extrapolation of cell positions (Genovesio et al., 2006), we explicitly incorporate velocity and acceleration components into the state vector, and derive the models based on state-space differential equations. The state-transition matrices corresponding to the RW, CV, CA, and CT models are, respectively,

$$\mathbf{F}^1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{F}^2 = \begin{bmatrix} 1 & T_s & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T_s & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{F}^3 = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} & 0 & 0 & 0 \\ 0 & 1 & T_s & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T_s & \frac{T_s^2}{2} \\ 0 & 0 & 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{F}_k^4 = \begin{bmatrix} 1 & \frac{\sin(\theta_k T_s)}{\theta_k} & \frac{1-\cos(\theta_k T_s)}{\theta_k^2} & 0 & 0 & 0 \\ 0 & \cos(\theta_k T_s) & \frac{\sin(\theta_k T_s)}{\theta_k} & 0 & 0 & 0 \\ 0 & -\theta_k \sin(\theta_k T_s) & \cos(\theta_k T_s) & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{\sin(\theta_k T_s)}{\theta_k} & \frac{1-\cos(\theta_k T_s)}{\theta_k^2} \\ 0 & 0 & 0 & 0 & \cos(\theta_k T_s) & \frac{\sin(\theta_k T_s)}{\theta_k} \\ 0 & 0 & 0 & 0 & -\theta_k \sin(\theta_k T_s) & \cos(\theta_k T_s) \end{bmatrix},$$

where T_s is the time between the measurements (i.e., the frame interval). The subscript k in the coordinated turn transition matrix $\mathbf{F}_k^4$ indicates that it is time varying. It depends on the angular turning rate θ_k, which can be computed from the velocity and acceleration vectors as $\theta_k = \sqrt{\ddot{x}_k^2 + \ddot{y}_k^2}/\sqrt{\dot{x}_k^2 + \dot{y}_k^2}$. We refer the reader to Zarchan and Musoff (2005); Herman, 2002 for detailed derivations of the state-transition matrices. The proposed motion models share a common measurement matrix:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

3.4.2. Parameter estimation and adaptation for IMM

With the system matrices defined, the noise covariances $\mathbf{Q}^i$, $\mathbf{R}$, and the initial error covariance matrix $\mathbf{\Sigma}_0^i$ can be estimated from training sequences using the expectation–maximization (EM) algorithm (Bishop, 2007). The details of the EM–IMM parameter estimation procedure are presented in Appendix B. While EM also permits the estimation of $\mathbf{F}^i$ and $\mathbf{H}$, the resulting matrices may be in arbitrary forms and are difficult to interpret. With the predefined system matrices, we gain additional insight into the typical motion patterns of each cell. Namely, we can identify if the cell motion is predominated by Brownian motion, constant-velocity migration, accelerating migration, or turning motion based on the corresponding model weights ρ_k^i computed by the IMM filter.

One important parameter yet to be specified is the Markovian model transition probabilities p_{ij}. By convention, p_{ij} can be arranged in an $M \times M$ transition probability matrix (TPM) $\mathbf{P}$, with $\mathbf{p}_i$ denoting the ith row of $\mathbf{P}$. Traditionally, the TPM is almost always treated as a fixed design parameter chosen empirically. For many biological applications, however, *a priori* information about the TPM may be inadequate or lacking. Cellular motion could vary considerably or become unpredictable due to changes of experimental procedures, extracellular environments, cell densities, and/or cell types. Moreover, imposing an empirical TPM would contradict the very goal of biological discovery, i.e., to discover *unknown* cell behavioral variations.

With the above considerations, we chose to perform online minimum mean-square error estimation of the TPM. Various algorithms exist for our purpose, and we adopt the *quasi-Bayesian* algorithm (Jilkov and Li, 2004), which is simple to implement, numerically stable, and requires negligible computational overhead. The quasi-Bayesian estimation assumes that each row $\mathbf{p}_i$ of the TPM follows a *Dirichlet* prior distribution. The Dirichlet distribution is defined by

$$p(\mathbf{p}_i|a_{i1},\ldots,a_{iM}) = \frac{\Gamma(a_{i1}+\cdots+a_{iM})}{\Gamma(a_{i1})\cdots\Gamma(a_{iM})}\prod_{j=1}^{M} p_{ij}^{a_{ij}-1}, \tag{32}$$

with *hyperparameters* $a_{ij} \geqslant 0$. The Dirichlet distribution naturally satisfies the unit simplex requirement $\sum_{j=1}^{M} p_{ij} = 1$ and $p_{ij} \in [0,1]$, for all i. The parameters a_{ij} represent the unnormalized *a priori* TPM. If they are chosen as $a_{i1} = \cdots = a_{iM} = 1$ for any i, the corresponding Dirichlet distribution of $\mathbf{p}_i$ coincides with the uniform distribution. Therefore, if *a priori* knowledge about the TPM is unavailable, the quasi-Bayesian estimator can naturally be initialized with the non-informative (uniform) prior $p_{ij} = 1/M$ using parameters $a_{ij} = 1$ ($i,j = 1,\ldots,M$).

The quasi-Bayesian algorithm proceeds as follows. Upon receiving the first measurement $\mathbf{z}_1$, a posterior probability $p(\mathbf{p}_i|\mathbf{z}_1)$ can be obtained based on the Dirichlet prior $p(\mathbf{p}_i)$ for each model i, which is a weighted sum of M Dirichlet distributions. The posteriors over

Algorithm 1: Quasi-Bayesian TPM Estimation

Input: A hyperparameter matrix $\mathbf{A}_0$ with entries $a_{ij,0} \geq 0$, $i, j = 1, \ldots, M$.

initialize

$\hat{\mathbf{p}}_{i,0} = (a_{i1,0}, \ldots, a_{iM,0})/a_{i,0}$, where $a_{i,0} = \sum_{j=1}^{M} a_{ij,0}$, $i = 1, \ldots, M$

end

for $k \leftarrow 1, 2, \ldots$ **do**

  for $i \leftarrow 1, \ldots, M$ **do**

    for $j \leftarrow 1, \ldots, M$ **do**

$$g_{ij,k} = 1 + \frac{\rho_{k-1}^{i}}{\boldsymbol{\rho}_{k-1}\hat{\mathbf{P}}_{k-1}\boldsymbol{\lambda}_k}(\lambda_k^j - \hat{\mathbf{p}}'_{i,k-1}\boldsymbol{\lambda}_k),$$

where $\boldsymbol{\rho}_k \equiv (\rho_k^1, \ldots, \rho_k^M)$ and $\boldsymbol{\lambda}_k \equiv (\lambda_k^1, \ldots, \lambda_k^M)'$

$$a_{ij,k} = a_{ij,k-1} + (a_{ij,k-1}g_{ij,k})/(\textstyle\sum_{j=1}^{M} a_{ij,k-1}g_{ij,k})$$

$$\hat{p}_{ij,k} = a_{ij,k}/(k + a_{i,0})$$

the subsequent measurements will be mixtures of exponentially more Dirichlet distributions. The quasi-Bayesian approach utilizes a similar approximation as in IMM to obtain a *quasi-posterior* distribution. At each time step, it approximates the posterior mixture of M Dirichlet distributions by a single Dirichlet distribution, then computes the quasi-posterior estimation $\hat{\mathbf{p}}_i$ as the mean of this approximated distribution. This process is elaborated in Jilkov and Li (2004) and Smith and Makov (1978), and can be summarized as a recursive algorithm (Algorithm 1). The quasi-Bayesian algorithm integrates seamlessly with IMM, enabling us to update the TPM after the correction step in each filtering cycle (see Section 3.4). A diagram of the TPM-adaptive IMM filter with two models is given in Fig. 4.

3.5. Track compilation

The track compiler coordinates the 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 2.

The compiler first compares the output of the cell detector and cell tracker, $\chi_k(x,y)$ and $\hat{\psi}_k(x,y)$. Each cell candidate in $\chi_k(x,y)$ that does not overlap with any propagated cell region in $\hat{\psi}_k(x,y)$ is considered a *new* cell. A new track segment will be initialized, and $\hat{\psi}_k(x,y)$ 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 $\hat{\psi}_k(x,y)$. A track segment whose corresponding propagated cell region contains only one connected component will be updated directly. If a cell region consists of more than one connected components separated by a minimum distance $d_{\min}$, 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 follows:

AddTrack$(\omega, n_{\text{new}}, k)$ 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 the centroid location, mean inten-

Algorithm 2: Track Compilation

$\Omega_0 \leftarrow \{(x,y) | \hat{\psi}_k(x,y) = 0\}$

1 **foreach** *cell candidate* $\omega \subset \chi_k$ **do**
  if $\omega \subset \Omega_0$ **then** $AddTrack(n_{new}, k, \omega)$

2 **foreach** *active track* $n \in \mathbf{N}_{k-1}$ **do**
  $\Omega_n \leftarrow \{(x,y) | \hat{\psi}_k(x,y) = n\}$
3 **if** $\Omega_n = \varnothing$ **then** $DeactivateTrack(n)$
4 **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
5      $\hat{\omega} \leftarrow SelectBestMatch(n, k, \Omega_n)$
      $UpdateTrack(n, k, \hat{\omega})$
      foreach *connected component* $\omega \subset \Omega_n \setminus \hat{\omega}$ **do** $AddTrack(n_{new}, k, \omega)$
6 **else** $UpdateTrack(n, k, \Omega_n)$

sity, area, and eccentricity. The centroid and the mean intensity are fed to the motion filter to obtain a filtered state of cell n in frame k. The last three features are used to classify a cell as normal, mitotic, or apoptotic, using three-nearest-neighbor (3NN) matching with the Mahalanobis distance to a set of training samples obtained off-line.

AddDaughterTrack($n_{\text{daughter}}, n, k, \omega$) 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 $\hat{\omega} \in \Omega_n$ that best matches the dynamics of cell n, i.e., the one which maximizes the innovation likelihood given by Eq. (31) 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 $d_{\min}$; otherwise, it returns `false`.

IsMitotic(n, k) returns `true` if cell n is classified as mitotic during the past T frames using the approach described in *UpdateTrack* above.

The parameters $d_{\min}$ and T involved in the algorithm need to be adjusted for specific datasets. Their values will be provided in Section 4.2.

3.6. Track linking

The track linker module provides the global view. It oversees the entire tracking history, and it 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 is released 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 violations of these constraints by linking track segments into complete cell trajectories, utilizing spatiotemporal context.

The track linking procedure is outlined in Algorithm 3. Here, $\mathbf{N}_{\text{lost}} \equiv \{n_l | l = 1, \ldots, L\}$ denotes the label set of track segments that end before frame k, and $\mathbf{N}_{\text{found}} \equiv \{n_f | f = 1, \ldots, F\}$ denotes the label set of track segments that start after the first frame. Most operations in the algorithm are self-explanatory. One vital step is the matching between lost and appearing track segments: *MatchTracks* (Line 4).

In *MatchTracks*, a bipartite graph G is created, whose nodes correspond to the labels in $\mathbf{N}_{\text{lost}}$ and $\mathbf{N}_{\text{found}}$.

For each node pair (n_l, n_f), an arc $\langle n_l, n_f \rangle$ is created 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.,

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

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. (31)). Intuitively, w_{lf} indicates how likely track segment n_f is a continuation of track segment n_l based on the dynamics of track n_l.

Next, a maximum-likelihood matching is computed between tracks n_l and n_f. The approach we reported previously (Li et al., 2007) only considered one-to-one matches. Hence, it could not handle the case where a cell is lost during mitosis, and whose daughter cells are re-detected in later frames. To handle this case, we improved the algorithm to consider both one-to-one and one-to-two matches. The algorithm relies on two inputs: an $H \times (L + F)$ constraint matrix $\mathbf{C}$ and an $H \times 1$ likelihood vector $\mathbf{d}$. Here, H is the total number of one-to-one and one-to-two matching hypotheses. L and F are, respectively, the numbers of track segments in $\mathbf{N}_{\text{lost}}$ and $\mathbf{N}_{\text{found}}$. Matrix $\mathbf{C}$ and vector $\mathbf{d}$ are constructed as follows:

For each arc $\langle n_l, n_f \rangle$ in G, a new row is appended 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 $n_{f_1}, \ldots, n_{f_m} \in \mathbf{N}_{\text{found}}$ $(m \geqslant 2)$, all possible one-to-two matchings are enumerated, 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 $n_l \to (n_{f_1}, n_{f_2})$, a new row with index h' to $\mathbf{C}$ and a corresponding new element is appended to $\mathbf{d}$. The value of $\mathbf{d}(h')$ is set to be the *maximum* innovation likelihood of track n_l for the spatiotemporal mean of the starting points of tracks n_{f_1} and n_{f_2}, with the constraint:

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

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 non-zero entries. This can be posed as the following integer programming problem:

Algorithm 3: Track Linking

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

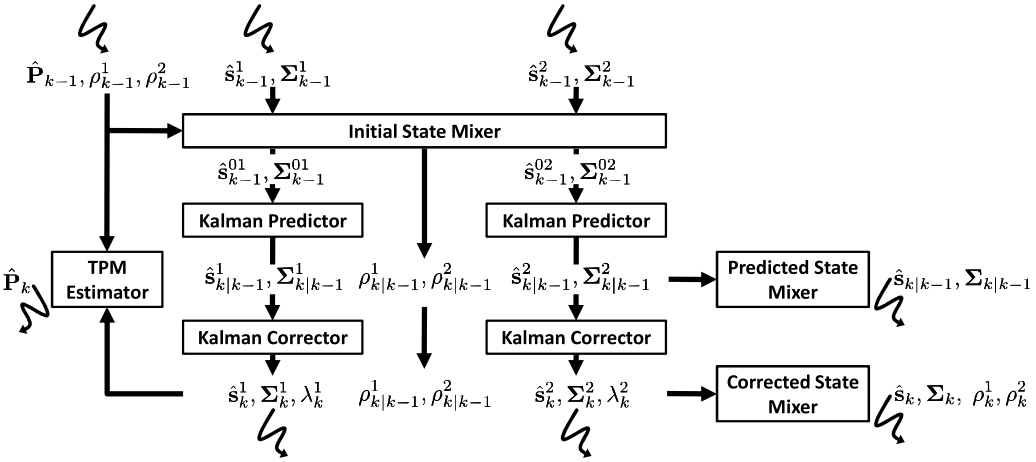

Fig. 4. Block diagram of the TPM-adaptive IMM filter framework for two models.

$$\max_{\mathbf{x}} \mathbf{d}'\mathbf{x}, \quad \text{such that } \mathbf{C}'\mathbf{x} \leqslant \mathbf{1}, \tag{33}$$

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. (33) 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 (Papadimitriou and Steiglitz, 1998). In our implementation, the open-source software package *lpsolve* (Berkelaar et al., 2007) is used to solve the above integer programming problem. A similar optimization approach was used by Al-Kofahi et al. (2006) for inter-frame cell matching. As an optional step after the completion of the track linking procedure, all cell trajectories that terminate in the field-of-view with lengths shorter than a preset threshold will be regarded as noise and removed (Line 7).

4. Experimental methods

The tracking system is implemented in ISO C++. The inputs to the system are grayscale image sequences generated by the imaging software QED Image (Media Cybernetics Inc.). Unprocessed microscopy images are often distorted by spatial illumination inhomogeneity. The distortion is especially severe when low-magnification objectives are used. To normalize illumination, a flat-field correction filter is applied to the input images. This filter divides each input frame by a preacquired light field image, and then it scales the output pixel values to a fixed range. In our experiments, a light field image is unavailable. Therefore, a pseudolight field is generated for each sequence by applying a Gaussian filter with standard deviation of 50 to the first frame.

4.1. Data

The performance of our system was quantitatively evaluated on eight phase contrast microscopy image sequences. They were categorized into three data sets (*A*, *B* and *C*) according to the cell type, imaging protocol and cell seeding method.

Dataset A includes two image sequences of MG-63 human osteosarcoma cells acquired with a 12-bit Qimaging Retiga EXi Fast 1394 CCD camera mounted on a Zeiss Axiovert 135 TV microscope, at a time-lapse interval of 4 min for 10 h. Each sequence consists of 150 frames, with a frame dimension of 1280 × 1024 pixels, and a resolution of 1.9 μm/pixel at 4.9× magnification. The cells were seeded randomly on a polystyrene dish. The images are cropped to a size of 512 × 512 pixels (Fig. 1a) to speed up processing and evaluation. The cell populations in the cropped sequences are in the range of 80–110 cells/frame. An independent sequence of the same cell type was utilized for training.

Dataset B includes four image sequences of proprietary amnion epithelial (AE) stem cells (Fig. 1b), acquired using the same imaging protocol as dataset A, except that the acquisition rate is 1 frame/10 min. The AE cells are extracted from the placenta following live birth, and are potentially a non-controversial source of stem cells for cell transplantation and regenerative medicine (Miki et al., 2005). The sequences were acquired over a duration of 42.5 h, each consisting of 256 frames with 1280 × 1024 pixels/frame. The cell population density in each sequence is roughly 2000–5000 cells/frame, and is nearly confluent towards the end of the sequence. An independent sequence of the same cell type was utilized for training.

Dataset C includes one sequence of MG-63 cells (Fig. 1c) recorded by an 8-bit CCD camera on a Zeiss IM35 microscope. The sequence lasts for 43.5 h and has a frame interval of 15 min, corresponding to 174 frames/sequence. The frame dimension is 512 × 512 pixels with a resolution of 3.9 μm/pixel at 5:1 magnification. The cells are seeded randomly on a fibrin-coated slide, on which a 0.75 × 0.75 mm^2 uniformly concentrated square pattern of FGF-2 was created using our bioprinter (Weiss et al., 2005). The cell population in the sequence is in the range of 350–750 cells/frame. The first 40 frames of the sequence are reserved for training, the rest is used for testing.

In addition to the above datasets, 35 sequences of AE cells were utilized to qualitatively assess the tracking performance of the system.

4.2. Parameters

Many of the parameters used in our system can be learned automatically from training data. These trainable parameters include the process noise covariances $\mathbf{Q}^i$ ($i = 1,\ldots,4$), measurement noise covariance $\mathbf{R}$, initial estimation error covariance $\mathbf{\Sigma}_0^i$ ($i = 1,\ldots,4$), and the model transition probability matrix (TPM) $\mathbf{P}$, all of which are required by the IMM motion filter. For each dataset, the parameters are learned using a set of manually tracked cell trajectories. The training set include 71 trajectories from dataset A, 101 trajectories from dataset B, and 232 trajectories from dataset C.

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

The training procedure alternates between two steps. First, $\mathbf{Q}^i$, $\mathbf{R}$, and $\boldsymbol{\Sigma}_0^i$ are trained using the EM–IMM algorithm (Appendix B) with an initial TPM $\mathbf{P}_0$. This TPM has diagonal entries $p_{ii,0} = 0.85$ and off-diagonal entries $p_{ij,0} = 0.05$, $(i \neq j)$, which encodes the assumption that a cell tends to stay in a motion mode rather than to switch to the other modes in successive time steps. It is chosen instead of a uniform (uninformative) initialization as suggested in Section 3.4.2 because it leads to better capability of model identification and faster convergence of the parameters. Then, once $\mathbf{Q}^i$, $\mathbf{R}$, and $\boldsymbol{\Sigma}_0^i$ are learned, the TPM is re-estimated using the quasi-Bayesian algorithm (Algorithm 1) with the hyperparameter matrix $\mathbf{A}_0$ set to equal the previous $\mathbf{P}$. The procedure iterates until the TPM converges.

The above procedure converges to near-identical TPMs for all datasets. One representative TPM approximately equals:

$$\mathbf{P} = \begin{bmatrix} 0.9793 & 0.0074 & 0.0064 & 0.0070 \\ 0.0072 & 0.9834 & 0.0037 & 0.0056 \\ 0.0272 & 0.0281 & 0.9156 & 0.0290 \\ 0.0276 & 0.0296 & 0.0279 & 0.9148 \end{bmatrix}. \quad (34)$$

In addition to the tendency for cells to stay unchanged in a motion mode, the TPM indicates that the random walk and constant-velocity motions are more persistent, whereas the acceleration and turning motions are relatively transient. It is used to initialize the hyperparameter matrix $\mathbf{A}_0$ in the subsequent experiments.

In contrast to the TPM, the learned values of $\mathbf{Q}^i$, $\mathbf{R}$, and $\boldsymbol{\Sigma}_0^i$ vary between different datasets. Their specific values are less informative than the TPM, and are omitted here.

The settings of the additional parameters are summarized in Table 1. These parameters can be intuitively determined based on direct observation. For example, the cell detector parameter r is set to roughly equal to the average cell radius. The size constraints $s_{\min}$ and $s_{\max}$ loosely correspond to the expected cell size range. The cell tracker parameters w_{edge}, w_{motion}, and τ are determined empirically, and they are mostly held constant for different datasets. The parameter T is related to the maximum duration of mitosis events. The track linker parameters D, R, and R_0 are constrained by the maximum cell migration speed.

4.3. Cell detection accuracy assessment

To quantitatively evaluate cell detection accuracy, the centroid positions of all visible cells in five randomly selected images in each dataset were manually identified using an interactive program. The human operators can navigate through contextual frames to better identify overlapping cells, and to distinguish cells from background and other objects (e.g., glass scraps, air bubbles, etc.) that may exist in the field.

Table 1
Summary of parameter settings for each dataset

Dataset	r	$s_{\min}$	$s_{\max}$	w_{edge}	w_{motion}	τ	$d_{\min}$	T	D	R	R_0
A	10	16	4000	0.1	0.1	0.001	10	10	10	5	10
B	5	5	2500	0.1	0.2	0.001	8	10	10	5	20
C	4	4	1000	0.1	0.1	0.001	3	5	6	3	5

The detection accuracy is gauged using two metrics: (1) *precision*, which is the ratio of the number of detected cells to the total number of detected objects, and (2) *recall* (*a.k.a.* sensitivity), which is the ratio of the number of detected cells to the total number of cells actually in the image, visually determined by the human observer. In terms of *true positives* (TPs), *false positives* (FPs), *true negatives* (TNs) and *false negatives* (FNs), the metrics can be computed as: precision = TP/(TP + FP) and recall = TP/(TP + FN).

4.4. Cell tracking accuracy assessment

The cells that appear in the initial frame of each sequence and their progeny were manually tracked. The manual tracking results were obtained after weeks of expert scrutiny and consensus was gained among multiple observers. The manually and automatically tracked trajectories (branches) were paired in the initial frame of each sequence, and they were compared in the remaining frames. An automatically tracked cell trajectory is considered *valid* only if it follows the *same* cell through all the frames that the cell appears. Any swapping of identities between two nearby cells will invalidate the trajectories of both cells and their progeny.

The sequences in datasets B and C contain hundreds to thousands of cells per frame, making it unrealistic to manually track all cells in the entire sequences. To make quantitative validations feasible, a 256×256-pixel region of interest (ROI) is defined in each image sequence of dataset B, and a 192×192-pixel ROI in dataset C. The automated tracking results are manually examined only within the ROI volume. In addition to the tracking validity defined previously, the ratio of cell divisions that were correctly tracked by the tracking system was also evaluated. This ratio is referred to as the *division tracking ratio*. A division is considered to be correctly tracked if the daughter cells are correctly located and the cell lineage is successfully established.

5. Results

5.1. Tracking examples

Before presenting quantitative results, we provide several explanative examples to demonstrate the key features of our system.

Fig. 5 demonstrates that the topology-constrained level set can effectively prevent merging of closely contacting cells and maintain cell identities. In addition, cells (e.g., cell 25) are automatically initialized when they enter, and they (e.g., cells 16 and 20) are removed when they exit the field of view. This example is cropped from one of the sequences in dataset B.

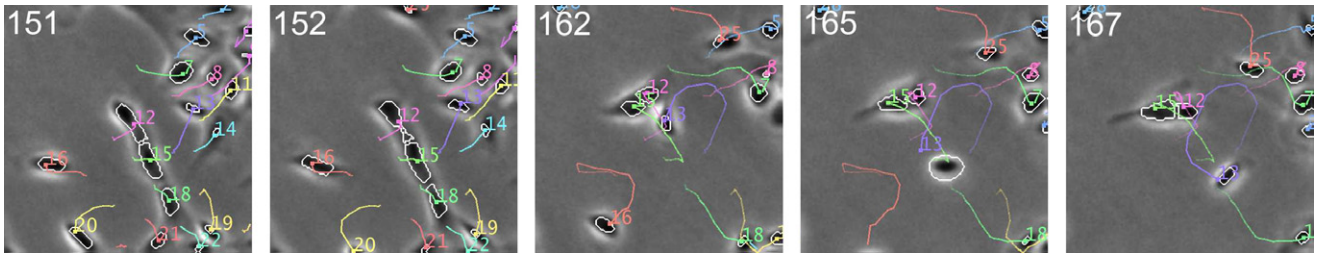

Fig. 5. Tracking contacting and partially overlapping AE cells. The numbers at the top-left corner are the frame indices. Cells 12, 15 and 18 are partially overlapping in frames 151–152. Cells 15 and 12 are closely passing each other in frames 162–167.

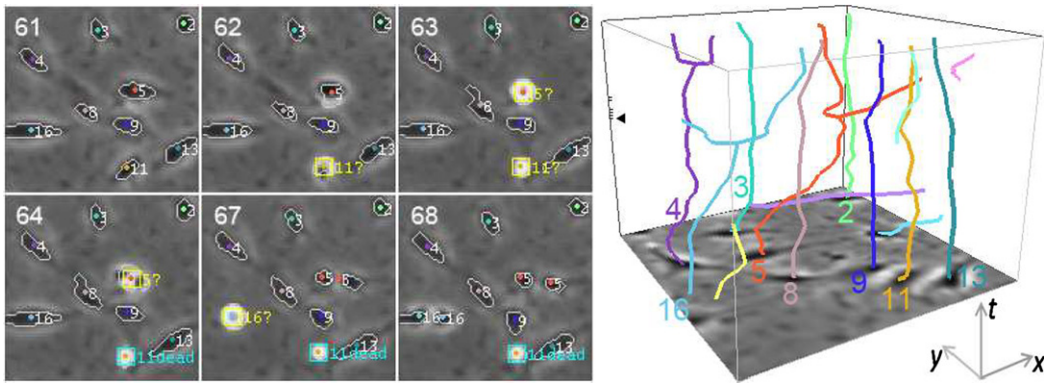

Fig. 6. Tracking mitotic and apoptotic MG-63 cells. Left: Six frames with cell boundaries and centroids overlaid. Question marks indicate cells in intermediate stages (either mitotic or apoptotic). For daughter cells, the label of their parent is shown. Right: A spatiotemporal plot of the corresponding cell trajectories. The tick marks on the bounding box indicate the time instants of the six frames shown in the left panel, and the triangle indicates frame 61.

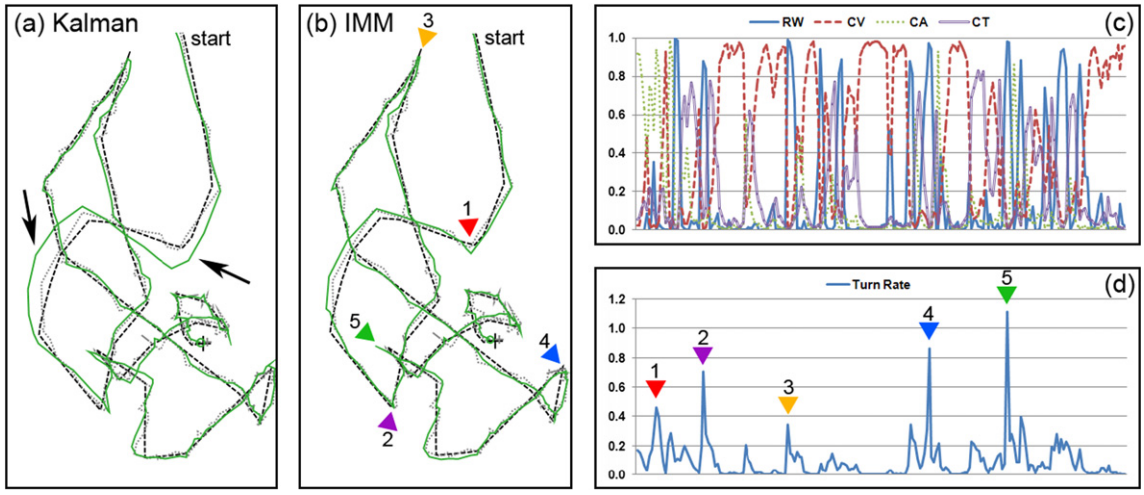

Fig. 7. IMM filter versus Kalman filter. Black dashed curves in (a) and (b) represent the true trajectory of a cell. Gray dotted curves show the noisy trajectories after superimposing additive Gaussian noise. Solid green curves are the estimated trajectories by the Kalman and IMM filers using the noisy trajectories as measurements. Plots (c) and (d) show the model weights ρ_k^j ($j = 1,\ldots,4$) and the turn rates θ_k estimated by the IMM filter during its operation. Colored triangles in (b) and (d) indicate major turning points of the trajectory.

Fig. 6 shows an example of tracking mitotic and apoptotic cells. The images are taken from dataset C. Since the appearances of mitotic and apoptotic cells are almost identical during a certain period (frames 62 and 63), they can only be distinguished with sufficient temporal information (frames 64–68). This example illustrates that our tracking system can effectively detect mitosis and apoptosis, and distinguish between them by using the temporal context.

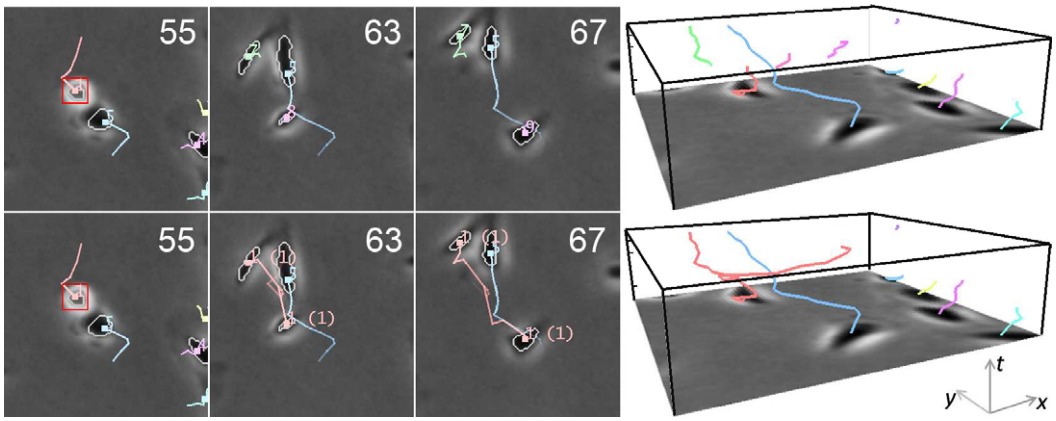

Fig. 8. Example of spatiotemporal track linking. Top: Track segments output by the track compiler. Bottom: Completed cell trajectories after track linking. The numbers in the parentheses indicate cell generations.

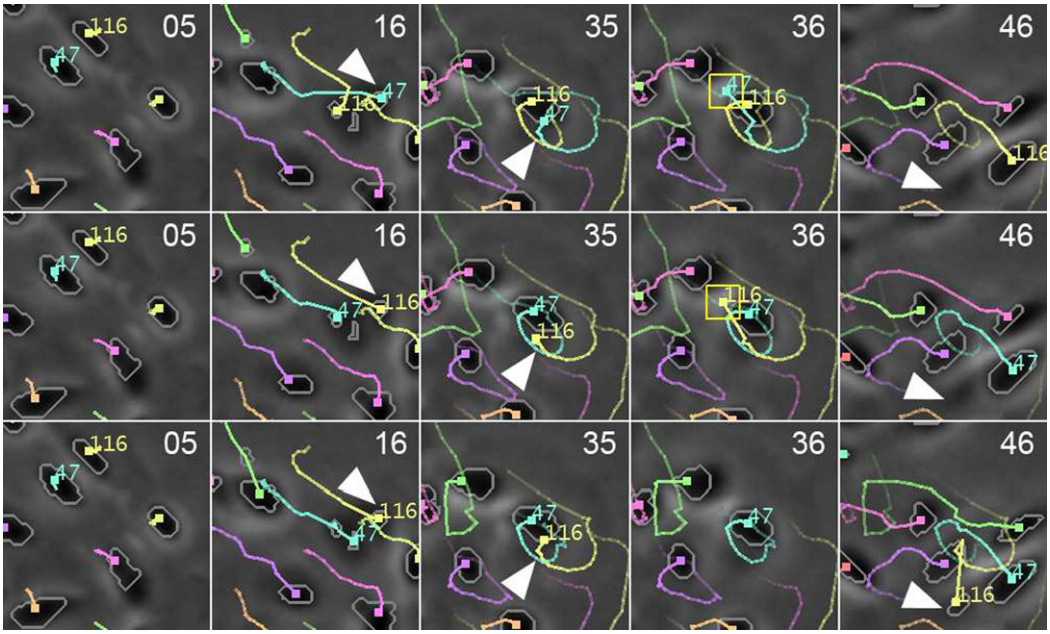

Fig. 9. Tracking AE cells through occlusion. Cell 116 was completely occluded by cell 47 from frame 35 to 45 and reappeared in frame 46. The numbers at the top-right corner are frame indices. The trailing curves represent cell trajectories. Different colors represent different cell lineages. Top: With standard Kalman motion filter and no spatiotemporal track linking, the system switched the identities of cell 116 and cell 47 and loses track of cell 116 eventually. Middle: By replacing the Kalman filter with an IMM filter, the system correctly maintained the identities of cells 47 and 116. Bottom: By incorporating both IMM motion filter and spatiotemporal track linking, the system successfully tracked all cells.

To illustrate the operation of the IMM filter (Section 3.4) and demonstrate its superiority to Kalman filter, an artificial example was devised as shown in Fig. 7. To reflect realistic cell motion and to serve as the ground truth, the trajectory of a cell in one of the sequences in dataset B was manually tracked. Gaussian noise of covariance $25\mathbf{I}$ is added to the trajectory to simulate the measured cell positions during tracking, where $\mathbf{I}$ is a 2×2 identity matrix. The IMM filter with the four motion models described in Section 3.4.1, as well as a standard Kalman filter using only the constant-velocity (CV) model, is then executed to estimate the cell trajectory based on the noisy measurements. Both filters utilized equivalent parameter settings.

As shown in Fig. 7a, the trajectory estimated by the Kalman filter (green curve) diverges from the true trajectory (black dashed curve) at the arrow-indicated positions, indicating that the CV model is no longer adequate to represent the turning motion at these locations. In comparison, the trajectory estimated by the IMM filter stays close to the ground truth, and exhibits appreciably smaller deviation from the true trajectory.

To provide additional insights into the IMM filter, we plotted the model weights ρ_k^j ($j = 1,\ldots,4$) (Fig. 7c) and the turn rate θ_k (Fig. 7d) estimated by the filter during its operation. As shown in these plots, the major turning points of the trajectory are indicated by peaks in the estimated turn rates, with higher peaks indicating tighter turns. An interesting exception is at the location indicated by triangle 3, where a near 180° turn is spotted in the trajectory, but the corresponding peak in the turn rate is relatively small. By examining the ground truth, we found that the cell stopped at the aforementioned location for a short period (approx. 20 min) before heading toward a different direction, resulting in a smaller turn rate. The stopping motion is captured by an increase in the random-walk model weight at the corresponding location in Fig. 7c.

The effect of spatiotemporal track linking is illustrated in Fig. 8. The top row of the figure shows the track segments, which are intermediate outputs of the track compiler. As shown, the trajectory of cell 1 and its daughters are broken in to multiple segments due to abrupt jumping motions of the cells. The bottom row shows the result after track linking, where the identities of the daughter cells and their lineage with cell 1 were successfully recovered.

Table 2
Summary of cell detection accuracy for all datasets

Dataset	Cell count	Detected	FP	FN	Precision (%)	Recall (%)
A	673	662	9	20	98.6	97.0
B	9879	9987	271	163	97.3	98.4
C	2126	2107	44	63	97.9	97.0
Overall	12,678	12,756	324	246	97.5	98.1

Table 3
Summary of tracking validity and division tracking accuracy of the automated tracking results as compared with manual tracking

Sequences	Kalman		IMM		IMM + track linking	
Tracking validity						
A1	70/81	(86.4%)	72/81	(88.9%)	75/81	(92.6%)
A2	82/93	(88.2%)	82/93	(88.2%)	86/93	(92.5%)
B1	70/92	(76.1%)	73/92	(79.3%)	81/92	(88.0%)
B2	90/117	(76.9%)	94/117	(80.3%)	101/117	(86.3%)
B3	81/104	(77.8%)	83/104	(79.8%)	91/104	(87.5%)
B4	80/108	(74.1%)	84/108	(77.8%)	93/108	(86.1%)
C1	98/121	(81.0%)	102/121	(84.3%)	110/121	(90.9%)
Division tracking ratio						
A1	1/1	(100%)	1/1	(100%)	1/1	(100%)
A2	0/0	(N/A)	0/0	(N/A)	0/0	(N/A)
B1	43/55	(78.2%)	43/55	(78.2%)	47/55	(85.5%)
B2	41/52	(78.8%)	42/52	(80.8%)	45/52	(86.5%)
B3	36/48	(75.0%)	37/48	(77.1%)	41/48	(85.4%)
B4	44/57	(77.2%)	47/57	(82.5%)	50/57	(87.7%)
C1	32/42	(76.2%)	33/42	(78.6%)	37/42	(88.1%)

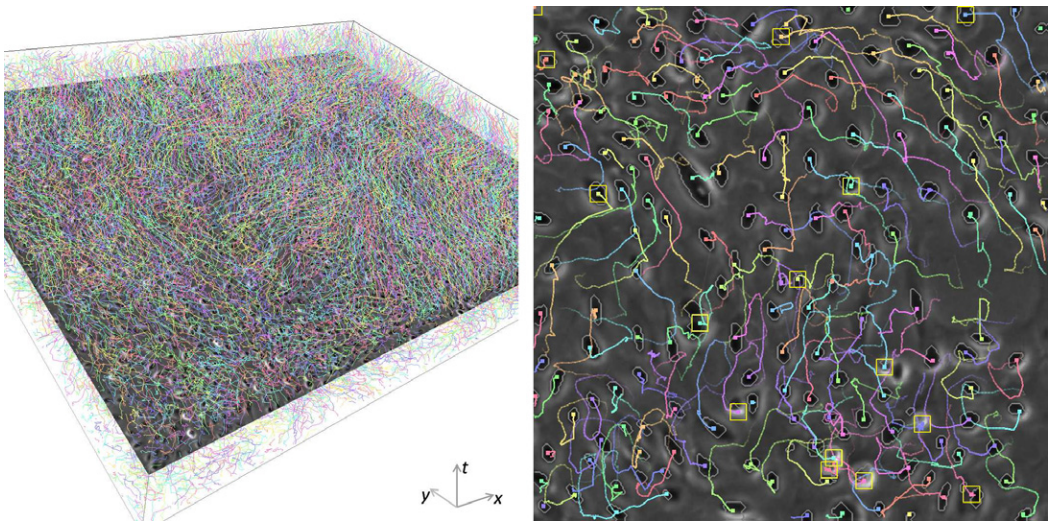

Fig. 10. Visualization of the tracking result of AE cells. Left: Spatiotemporal visualization of cell trajectories. Right: One of the selected regions used for quantitative validation with cell trajectories overlaid. Yellow rectangles indicate occurrences of mitosis in the past $T = 10$ frames.

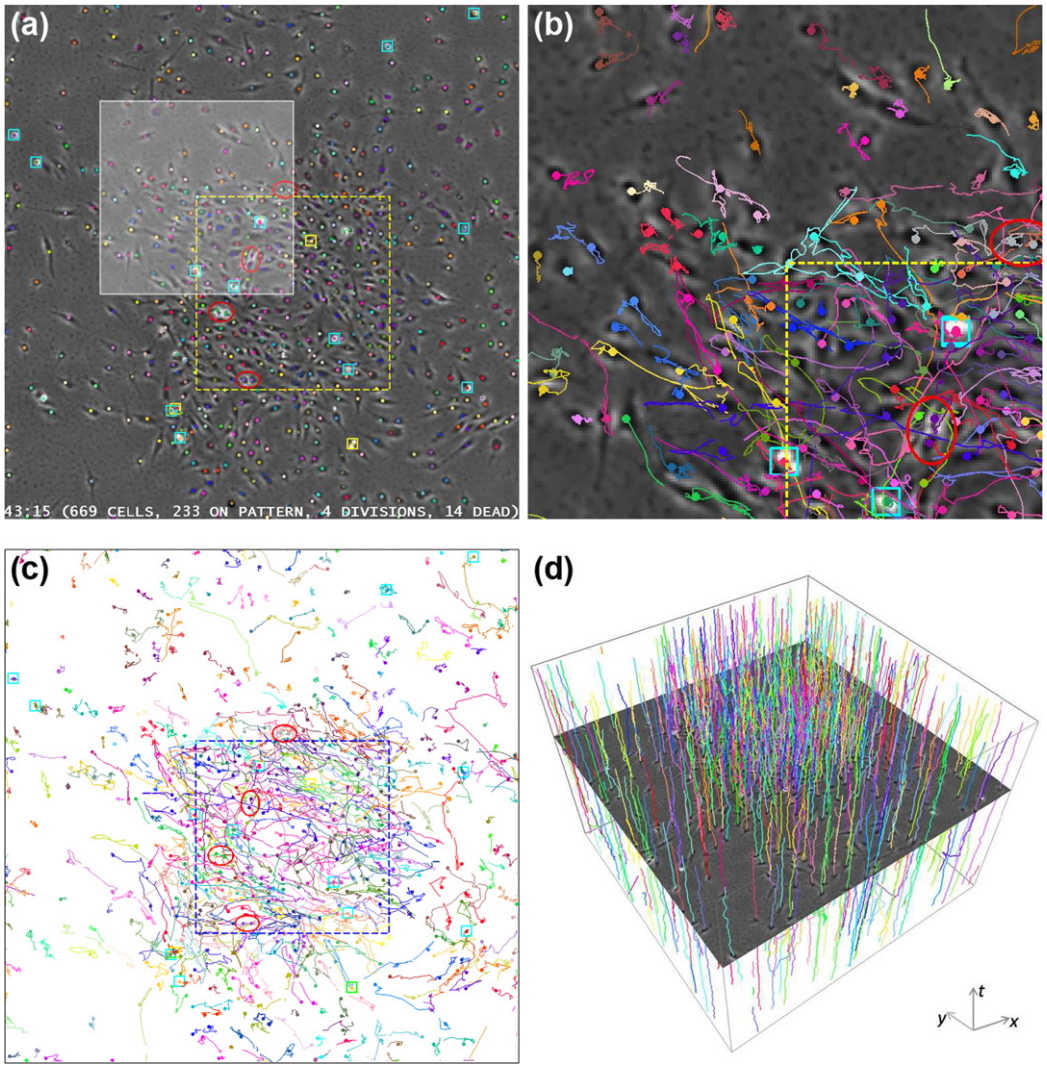

Fig. 11. Automatically tracked MG-63 cell trajectories. The yellow and blue dashed lines indicate the location of the printed growth-factor pattern. Red ellipses indicate cell division. Cyan squares indicate dead cells. Yellow squares indicate mitotic or apoptotic cells. Cells inherit the colors of their farthest ancestors. (a) A sample frame with cell centroids overlaid. (b) Magnified view of the highlighted region in (a) with automatically tracked cell trajectories overlaid. (c) and (d) 2D and spatiotemporal rendering of the automatically tracked cell trajectories.

With the incorporation of IMM motion filter and spatiotemporal track linking (Section 3.6), our system achieves superior robustness in handling varying cell motions and long-term occlusions. Fig. 9 shows a 20×20-pixel portion of the tracking result for a sequence in dataset B. In this example, cell 116 is occluded by cell 47 in frame 36 and reappeared in frame 46. With a standard Kalman motion filter and no spatiotemporal linking, the system (top 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. By replacing the Kalman filter with an IMM filter, the system correctly maintained the identities of cell 47 and 116, but it still lost track of cell 116 due to occlusion (middle row). Finally, by incorporating spatiotemporal optimization, the system correctly recovered the trajectory of cell 116 after occlusion (bottom row).

5.2. Detection and tracking accuracy

The selected experimental datasets represent high and varying densities of cell populations as well as a variety of complex cell behaviors, which pose significant challenges for the tracking system. For example, the cell populations in datasets B and C are nearly confluent towards the end of each sequence, with densities as high as approximately 30 cells/100^2 pixels. In addition, while the typical diameter of AE stem cells in the experimental image sequence is around 5–12 pixels, some cells migrate more than 20 pixels between frames, which is a distance much larger than the cell diameter. The active contour tracker alone is insufficient to handle such large displacements. Moreover, some of the cells were frequently occluded in some frames and reemerged in other frames. Under these challenging conditions, our system achieved high detection and tracking accuracy as summarized in Tables 2 and 3, respectively.

The first column of Table 3 summarizes the tracking performance of our system with a standard Kalman motion filter and a constant-velocity motion model. The second column shows the performance with an adaptive IMM motion filter and four motion models as described in Section 3.4.1. The third column shows the performance after the incorporation of spatiotemporal track linking. As the statistics suggests, the incorporation of spatiotemporal track linking leads to significant performance boosts (as much as a 12% difference) compared to the Kalman filter-based system. In comparison, the IMM filter provides relative small performance improvements. However, the IMM filter still always outperforms the Kalman filter, and helps to resolve certain challenging tracking scenarios. One example is shown in Fig. 9. With the incorporation of IMM filter and spatiotemporal track linking, our system achieved overall tracking validity of 92.5% for dataset A, 86.9% for

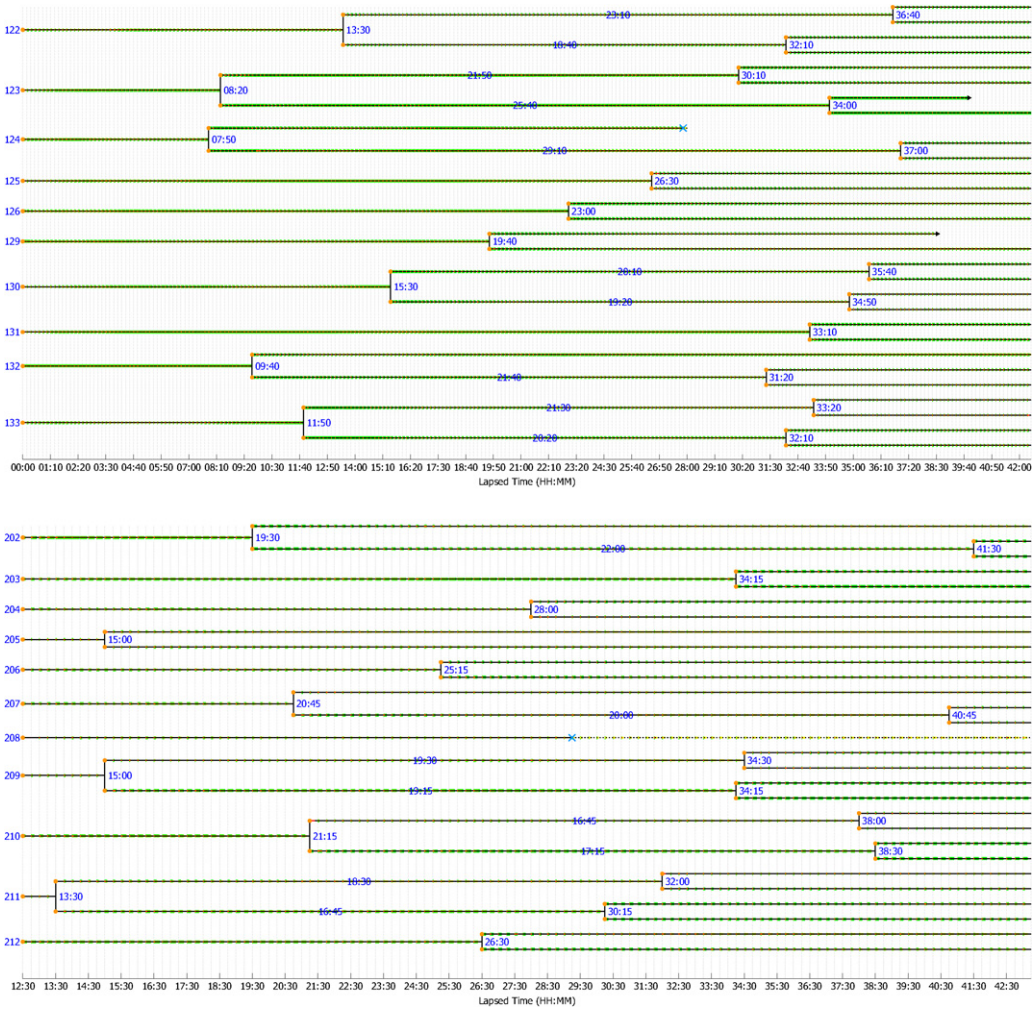

Fig. 12. Portions of color-coded lineage maps of AE cells (top) and MG-63 cells (bottom) constructed based on automated tracking results. Green line segments indicate the relative migration distances, where a longer line segment suggests a longer migration distance up to the current time point. Black arrows indicate cell departure. Blue crosses indicate cell death. Texts on the lineage lines show the division times.

dataset B, and 90.9% for dataset C. It achieved overall division tracking ratios of 100% for dataset A, 86.3% for dataset B, and 88.1% for dataset C.

A visualization of the automatically tracked trajectories of more than 4000 AE stem cells across 256 frames is provided in Fig. 10, and a visualization of the tracking result for MG-63 cells is shown in Fig. 11.

5.3. Lineage construction

One application of the tracking system is to automatically reconstruct cell lineage maps, which is especially important for stem cell research. In addition to revealing the mother–daughter relations between cells, metrics such as symmetry and division times can also be derived from lineage maps. Symmetry, which is defined as the mitotic fraction, is a measure of the capability of a stem cell to divide and produce daughter cells that are essentially identical to the mother, thus representing self-renewal (Deasy and Huard, 2002). And, the division time, which is defined as the lapsed time between cytokinetic events, is a key parameter in determining the expansion rate of stem cell populations. For example, these metrics can be used in predictive models of stem cell population growth during cell culture expansions (Deasy and Huard, 2002), as well as the design and optimization of subculturing strategies.

We utilized the system to construct the lineage maps for the entire populations of AE stem cells in dataset B and of MG-63 cells in dataset C. Our system correctly constructed 62.4% of the lineage trees for AE stem cells and 68.3% for MG-63 cells as measured in the selected regions of interest (see Section 3.4); Fig. 12 shows samples of the correctly constructed lineage trees with cells undergoing multiple divisions. In general, achieving higher accuracy is challenging since a single tracking error will invalidate the entire lineage tree that the cell belongs to. However, alternative approaches for mitosis/apoptosis event detection (Li et al., 2008) and spatiotemporal image processing techniques (Padfield et al., 2006a) could potentially improve lineage tracking accuracy. Increasing the image acquisition rate would be another possibility to reduce ambiguities and hence increase tracking accuracy. We also emphasize that acceptable accuracy levels required in various experiments, using large and dense cell populations, will likely vary depending on the specific biological question being proposed.

5.4. Computation time

All of the above experiments were conducted on a computer with a 2.66 GHz Intel Xeon processor and 4 GB memory, running 64-bit Gentoo Linux operating system. Our system runs at an average speed of 90 frames/h for tracking approximately 3000 cells in a 1280×1024 pixels/frame image sequence. The most time consuming computation step is the level set evolution.

6. Conclusion

We developed and validated an automated system capable of simultaneously tracking thousands of individual cells in dense cell populations in phase contrast microscopy image sequences. The system employs a modular design, which integrates an efficient cell detector, a topology-constrained geometric active contour tracker, a biologically relevant IMM motion filter, and spatiotemporal trajectory optimization. Our system enables automatic quantification of cell migration, proliferation, apoptosis, and construction of cell lineage maps, which facilitates the analysis of massive biological datasets. For future work, we will focus on further improving tracking accuracy for automated cell lineage construction, and applying the system to tackle challenging biological problems.

Acknowledgements

This work was supported partially by the National Institute of Health Grants R01 EB007369-01 and R01 EB0004343-01, the Pennsylvania Infrastructure Technology Alliance Grant 1C76 HF 00381-01, and an equipment grant from Intel Corporation.

Appendix A. Two-cycle fast level set algorithm

Algorithm 4 provides the pseudocode of the fast dual-cycle level set algorithm described in Section 3.3.3. Fig. A.1 illustrates the key steps of the algorithm.

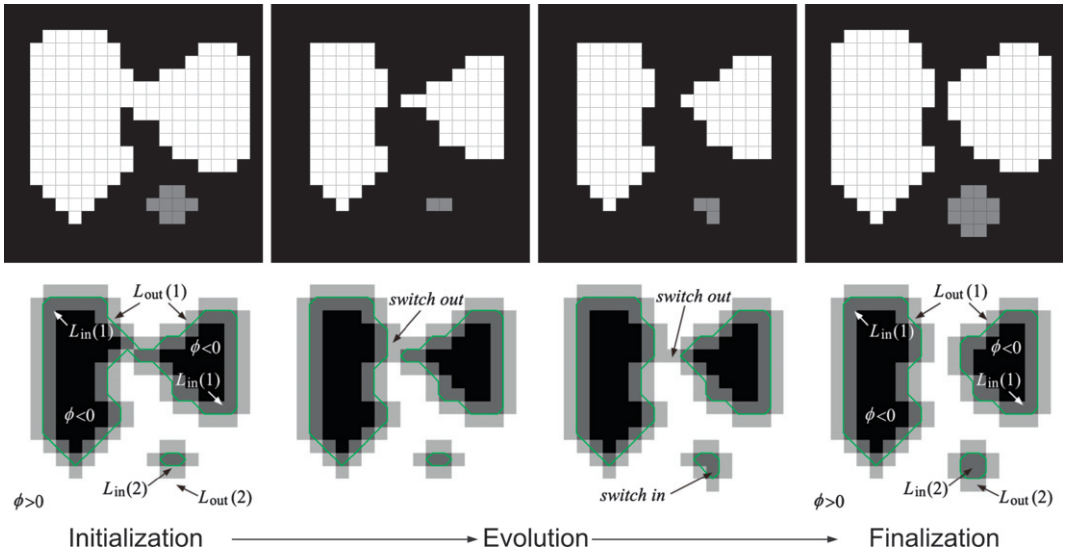

Fig. A.1. Illustration of fast level set initialization, evolution, and finalization.

Algorithm 4: Fast Dual-Cycle Level Set Evolution

Input: region labeling map $\psi_{k-1}(x,y)$; parameters $T_{\max}$, U, V

initialize

  $\hat{\psi}_k(x,y) \leftarrow \psi_{k-1}(x,y)$; $\mathbf{N}_{k-1} \leftarrow \varnothing$

  foreach *labeled region* $\Omega_n \subset \hat{\psi}_k(x,y)$ **do**

    Initialize $\phi(x,y)$, $L_{\text{in}}(n)$, and $L_{\text{out}}(n)$ according to (15) and (16)

    foreach $(x,y) \in L_{out}(n)$ **do** $\hat{\psi}_k(x,y) \leftarrow 0$

    $\mathbf{N}_{k-1} \leftarrow \mathbf{N}_{k-1} \cup \{n\}$

end

`// main loop`

for $t \leftarrow 1, \ldots, T_{max}$ **do**

  stop ←`false`

  for $u \leftarrow 1, \ldots, U$ **do** `// update cycle`

    foreach $n \in \mathbf{N}_{k-1}$ **do**

      foreach $(x,y) \in L_{in}(n) \cup L_{out}(n)$ **do** *compute* $\hat{F}(x,y)$

      foreach $(x,y) \in L_{out}(n)$ **do**

        if $\hat{F}(x,y) > 0$ **then** *SwitchIn*(x,y)

      UpdateInterior$(L_{\text{in}}(n))$

      foreach $(x,y) \in L_{in}(n)$ **do**

        if $\hat{F}(x,y) < 0$ **then** *SwitchOut*(x,y)

      UpdateExterior$(L_{\text{out}}(n))$

1   **if** *stopping condition is met* **then** *stop* ←`true` and *exit cycle*

  for $v \leftarrow 1, \ldots, V$ **do** `// regulation cycle`

    foreach $n \in \mathbf{N}_{k-1}$ **do**

      foreach $(x,y) \in L_{in}(n)$ **do**

2   **if** $(G * \phi)(x,y) < 0$ **then** *SwitchIn*(x,y)

      UpdateInterior$(L_{\text{in}}(n))$

      foreach $(x,y) \in L_{out}(n)$ **do**

3   **if** $(G * \phi)(x,y) > 0$ **then** *SwitchOut*(x,y)

      UpdateExterior$(L_{\text{out}}(n))$

  if *stop* **then** *exit main loop*

finalize

  foreach $n \in \mathbf{N}_{k-1}$ **do**

    foreach $(x,y) \in L_{out}(n)$ **do**

4       $\hat{\psi}_k(x,y) \leftarrow \arg\min_{n^* \in \mathbf{N}^*} \text{distance}(n^*, n|x,y)$

end

The input to the algorithm is the region labeling map $\psi_{k-1}(x,y)$ for frame $k-1$; the output is the evolved labeling map $\hat{\psi}_k(x,y)$ for frame k. There are three parameters: the maximum iterations of the main loop ($T_{\max}$), the number of iterations for the update cycle (U), and the number of iterations for the regulation cycle (V). U and V controls the relative strength of regulation, which replace the parameter v in Eq. (11). We set $T_{\max} = 500$, $U = 3$, and $V = 1$ in all the experiments.

The stopping condition (Line 1) is satisfied if $F(x,y) \leqslant 0$, $\forall (x,y) \in L_{\text{out}}(n)$, $\forall n$ and $F(x,y) \geqslant 0$, $\forall (x,y) \in L_{\text{in}}(n)$, $\forall n$. The discrete Gaussian kernel G (Lines 2 and 3) is approximated by

$$G = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}. \tag{A.1}$$

$\mathbf{N}^*$ (Line 4) denotes the set $\{n \in \mathbf{N}_{k-1} | \Omega_n \cap N_4(x,y) \neq \varnothing\}$, and the distance function is defined as

$$\text{distance}(n^*, n|x,y) = |I_k(x^*,y^*) - I_k(x,y)|, \tag{A.2}$$

where $(x^*,y^*) \in N_4(x,y)$, $\psi(x^*,y^*) = n^*$, and $\psi(x,y) = n$.

The psuedocode of the procedures *SwitchIn*, *UpdateInterior*, *SwitchOut*, and *UpdateExterior*, which are required by the dual-cycle algorithm, are listed in Algorithms 5–8.

Appendix B. EM–IMM parameter estimation

The unknown parameters are denoted by $\mathbf{\Theta} \equiv \{\mathbf{Q}^i, \mathbf{R}, \mathbf{\Sigma}_0^i\}$. Our objective is to estimate the parameters $\mathbf{\Theta}$ and the hidden states $\mathbf{S} = \{\mathbf{s}_0, \ldots, \mathbf{s}_K\}$ based on the measurements $\mathbf{Z} = \{\mathbf{z}_{-2}, \ldots, \mathbf{z}_K\}$. We make the index of the measurement sequence to start from -2 for the convenience of notation. The initial state $\mathbf{s}_0$ is obtained using the first three measurements $\{\mathbf{z}_{-2}, \mathbf{z}_{-1}, \mathbf{z}_0\}$.

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

Algorithm 5: Procedure *SwitchIn*

foreach $(x, y) \in L_{out}(n)$ **do**
  if $T_r(x, y) \neq 1$ **then** *exit*
  remove (x, y) from $L_{\text{out}}(n)$ and add it to $L_{\text{in}}(n)$
  foreach $(x^*, y^*) \in N_4(x, y)$ *with* $\phi(x^*, y^*) = 3$ **do**
    add (x^*, y^*) to $L_{\text{out}}(n)$; $\phi(x^*, y^*) \leftarrow 1$

Algorithm 6: Procedure *UpdateInterior*

foreach $(x, y) \in L_{in}(n)$ **do**
  if $\phi(x^*, y^*) < 0$, $\forall (x^*, y^*) \in N_4(x, y)$ **then**
    remove (x, y) from $L_{\text{in}}(n)$; $\phi(x, y) \leftarrow -3$

Algorithm 7: Procedure *SwitchOut*

foreach $(x, y) \in L_{in}(n)$ **do**
  remove (x, y) from $L_{\text{in}}(n)$ and add it to $L_{\text{out}}(n)$
  foreach $(x^*, y^*) \in N_4(x, y)$ *with* $\phi(x^*, y^*) = -3$ **do**
    add (x^*, y^*) to $L_{\text{in}}(n)$; $\phi(x^*, y^*) \leftarrow -1$

Algorithm 8: Procedure *UpdateExterior*

foreach $(x, y) \in L_{out}(n)$ **do**
  if $\phi(x^*, y^*) > 0$, $\forall (x^*, y^*) \in N_4(x, y)$ **then**
    remove (x, y) from $L_{\text{out}}(n)$; $\phi(x, y) \leftarrow 3$

$$\begin{aligned}\log P(\mathbf{S}, \mathbf{Z}|\boldsymbol{\Theta}) = &-\frac{K}{2}\log|\mathbf{Q}^i| - \frac{K}{2}\log|\mathbf{R}| - \frac{1}{2}\log|\boldsymbol{\Sigma}_0^i| \\ &-\frac{1}{2}\sum_{k=1}^{K}(\mathbf{z}_k - \mathbf{H}\mathbf{s}_k)'\mathbf{R}^{-1}(\mathbf{z}_k - \mathbf{H}\mathbf{s}_k) \\ &-\frac{1}{2}\sum_{k=1}^{K}(\mathbf{s}_k - \mathbf{F}^i\mathbf{s}_{k-1})'(\mathbf{Q}^i)^{-1}(\mathbf{s}_k - \mathbf{F}^i\mathbf{s}_{k-1}) + \text{constant},\end{aligned} \tag{B.1}$$

where $|\cdot|$ denotes matrix determinant. 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: *expectation* (E) and *maximization* (M).

B.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 $\boldsymbol{\Theta}^{\text{old}}$:

$$\mathcal{Q}(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{\text{old}}) = E[\log P(\mathbf{S}, \mathbf{Z}, \boldsymbol{\Theta}|\mathbf{Z}, \boldsymbol{\Theta}^{\text{old}})]. \tag{B.2}$$

This quantity depends on three expectations:

$$\hat{\mathbf{s}}_{k|K} \equiv E[(\mathbf{s}_k|\mathbf{Z}, \boldsymbol{\Theta}^{\text{old}})], \tag{B.3}$$

$$\boldsymbol{\Sigma}_{k|K} \equiv E[(\mathbf{s}_k\mathbf{s}_k'|\mathbf{Z}, \boldsymbol{\Theta}^{\text{old}})], \tag{B.4}$$

$$\boldsymbol{\Sigma}_{k,k-1|K} \equiv E[(\mathbf{s}_k\mathbf{s}_{k-1}'|\mathbf{Z}, \boldsymbol{\Theta}^{\text{old}})]. \tag{B.5}$$

Note that the estimates $\hat{\mathbf{s}}_{k|K}$ and $\hat{\boldsymbol{\Sigma}}_{k|K}$ 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, the fixed-interval IMM smoother in Helmick et al. (1995) is utilized. 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 $\boldsymbol{\Sigma}_k^j$ as given previously. The other filter propagates in the backward-time direction, and produces estimates $\hat{\mathbf{s}}_k^{b,i}$, $\boldsymbol{\Sigma}_k^{b,i}$, $\hat{\mathbf{s}}_{k|k+1}^{b,i}$ and $\boldsymbol{\Sigma}_{k|k+1}^{b,i}$. We refer the reader to Helmick et al. (1995) for details on the backward-time IMM filter. The fixed-interval IMM smoother combines the forward and backward filtered outputs to obtain smoothed estimates according to the following procedure.

Step 1: Compute combined estimates:

$$\begin{aligned}\hat{\mathbf{s}}_{k|K}^{ji} &= \boldsymbol{\Sigma}_{k|K}^{ji}[(\boldsymbol{\Sigma}_k^j)^{-1}\hat{\mathbf{s}}_k^j + (\boldsymbol{\Sigma}_{k|k+1}^{b,i})^{-1}\hat{\mathbf{s}}_{k|k+1}^{b,i}], \\ \boldsymbol{\Sigma}_{k|K}^{ji} &= [(\boldsymbol{\Sigma}_k^j)^{-1} + (\boldsymbol{\Sigma}_{k|k+1}^{b,i})^{-1}]^{-1}, \\ \boldsymbol{\Sigma}_{k+1,k|K}^{ji} &= [(\boldsymbol{\Sigma}_{k+1,k|k}^{j})^{-1} + (\boldsymbol{\Sigma}_{k+1,k|k+1}^{b,i})^{-1}]^{-1},\end{aligned} \tag{B.6}$$

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

Step 2: Compute model-conditioned smoothed estimates:

$$\begin{aligned}\hat{\mathbf{s}}_{k|K}^j &= \sum_{i=1}^{r}\rho_{k+1|K}^{ij}\hat{\mathbf{s}}_{k|K}^{ji}, \\ \boldsymbol{\Sigma}_{k|K}^j &= \sum_{i=1}^{r}\rho_{k+1|K}^{ij}[\boldsymbol{\Sigma}_{k|K}^{ji} + (\hat{\mathbf{s}}_{k|K}^{ji} - \hat{\mathbf{s}}_{k|K}^j)(\hat{\mathbf{s}}_{k|K}^{ji} - \hat{\mathbf{s}}_{k|K}^j)'], \\ \boldsymbol{\Sigma}_{k+1,k|K}^j &= \sum_{i=1}^{r}\rho_{k+1|K}^{ij}[\boldsymbol{\Sigma}_{k+1,k|K}^{ji} + (\hat{\mathbf{s}}_{k+1|K}^{ji} - \hat{\mathbf{s}}_{k+1|K}^j)(\hat{\mathbf{s}}_{k|K}^{ji} - \hat{\mathbf{s}}_{k|K}^j)'].\end{aligned} \tag{B.7}$$

The conditional probability $\rho_{k+1|K}^{ij}$ is obtained by

$$\rho_{k+1|K}^{ij} = p_{ji}\lambda_k^{ji}/\gamma_j \quad \text{with } \gamma_j = \sum_{i=1}^{r}p_{ji}\lambda_k^{ji}. \tag{B.8}$$

The likelihood λ_k^{ji} is given by

$$\lambda_k^{ji} = \mathcal{N}(\hat{\mathbf{s}}_{k|k+1}^{b,i} - \hat{\mathbf{s}}_k^j; 0, \boldsymbol{\Sigma}_{k|k+1}^{b,i} + \boldsymbol{\Sigma}_k^j), \tag{B.9}$$

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

Step 3: Compute the overall smoothed estimates:

$$\hat{\mathbf{s}}_{k|K} = \sum_{j=1}^{r} \rho_k^{s,j} \hat{\mathbf{s}}_{k|K}^{j},$$

$$\boldsymbol{\Sigma}_{k|K} = \sum_{j=1}^{r} \rho_k^{s,j} [\boldsymbol{\Sigma}_{k|K}^{j} + (\hat{\mathbf{s}}_{k|K}^{j} - \hat{\mathbf{s}}_{k|K})(\hat{\mathbf{s}}_{k|K}^{j} - \hat{\mathbf{s}}_{k|K})'], \tag{B.10}$$

$$\boldsymbol{\Sigma}_{k+1,k|K} = \sum_{j=1}^{r} \rho_k^{s,j} [\boldsymbol{\Sigma}_{k+1,k|K}^{j} + (\hat{\mathbf{s}}_{k+1|K}^{j} - \hat{\mathbf{s}}_{k+1|K})(\hat{\mathbf{s}}_{k|K}^{j} - \hat{\mathbf{s}}_{k|K})'].$$

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

$$\rho_k^{s,j} = \frac{\gamma_j \rho_k^{j}}{\sum_{j=1}^{r} \gamma_j \rho_k^{j}}, \tag{B.11}$$

where ρ_k^{j} is the forward-time filtered model probability.

B.2. M step

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

$$\boldsymbol{\Theta}^{\text{new}} = \max_{\boldsymbol{\Theta}} \mathcal{Q}(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{\text{old}}). \tag{B.12}$$

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:

$$(\mathbf{Q}^{i})^{\text{new}} = \frac{1}{K} \sum_{k=1}^{K} [\boldsymbol{\Sigma}_{k|K} - \mathbf{F}^{i} \boldsymbol{\Sigma}_{k-1,k|K} - \boldsymbol{\Sigma}_{k,k-1|K} (\mathbf{F}^{i})' + \mathbf{F}^{i} \boldsymbol{\Sigma}_{k-1|K} (\mathbf{F}^{i})'], \tag{B.13}$$

$$\mathbf{R}^{\text{new}} = \frac{1}{K} \sum_{k=1}^{K} (\mathbf{z}_k \mathbf{z}_k' - 2\mathbf{H}\hat{\mathbf{s}}_{k|K} \mathbf{z}_k' + \mathbf{H}\boldsymbol{\Sigma}_{k|K} \mathbf{H}'), \tag{B.14}$$

$$(\boldsymbol{\Sigma}_0^{i})^{\text{new}} = \boldsymbol{\Sigma}_{0|K}. \tag{B.15}$$

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.

References

Al-Kofahi, O., Radke, R.J., Goderie, S.K., Shen, Q., Temple, S., Roysam, B., 2006. Automated cell lineage construction: a rapid method to analyze clonal development established with murine neural progenitor cells. Cell Cycle 5 (3), 327–335.

Appel, K., Haken, W., 1977a. Every planar map is four colorable. Part I: Discharging. Illinois J. Math 21, 429–490.

Appel, K., Haken, W., 1977b. Every planar map is four colorable. Part II: Reducibility. Illinois J. Math. 21, 491–567.

Bahnson, A., Athanassiou, C., Koebler, D., Qian, L., Shun, T., Shields, D., Yu, H., Wang, H., Goff, J., Cheng, T., Houck, R., Cowsert, L., 2005. Automated measurement of cell motility and proliferation. BMC Cell Biol. 6 (19).

Bao, Z., Murray, J.I., Boyle, T., Ooi, S.L., Sandel, M.J., Waterston, R.H., 2006. Automated cell lineage tracing in caenorhabditis elegans. Proc. Natl. Acad. Sci. 103 (8), 2707–2712.

Berkelaar, M., Dirks, J., Eikland, K., Notebaert, P., 2007. lpsolve: a mixed integer linear programming (MILP) solver. <http://sourceforge.net/projects/lpsolve>.

Bishop, C.M., 2007. Pattern Recognition and Machine Learning. Springer.

Blom, H.A.P., 1984. An efficient filter for abruptly changing systems. In: Proceedings of the 23rd IEEE Conference on Decision and Control, pp. 656–658.

Braun, V., Azevedo, R.B.R., Gumbel1, M., Agapow, P.-M., Leroi, A.M., Meinzer, H.-P., 2003. ALES: cell lineage analysis and mapping of developmental events. Bioinformatics 19 (7), 851–858.

Brox, T., Weickert, J., 2006. Level set segmentation with multiple-regions. IEEE Trans. Image Process. 15 (10), 3213–3218.

Bunyak, F., Palaniappan, K., Nath, S.K., Baskin, T.I., Dong, G., 2006. Quantitative cell motility for in vitro wound healing using level set-based active contour tracking. In: Proceedings of the Third IEEE International Symposium Biomedical Imaging (ISBI), pp. 1040–1043.

Campbell, P.G., Miller, E.D., Fisher, G.W., Walker, L.M., Weiss, L.E., 2005. Engineered spatial patterns of FGF-2 immobilized on fibrin direct cell organization. Biomaterials 26, 6762–6770.

Canny, J., 1986. A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. 8, 679–698.

Caselles, V., Kimmel, R., Sapiro, G., 1997. Geodesic active contours. Int. J. Comput. Vis. 22 (1), 61–79.

Cates, J.E., Lefohn, A.E., Whitaker, R.T., 2004. GIST: an interactive, GPU-based level set segmentation tool for 3D medical images. Med. Image Anal. 8, 217–231.

Cheng, Y., 1995. Mean shift, mode seeking, and clustering. IEEE Trans. Pattern Anal. Mach. Intell. 17 (8), 790–799.

Chopp, D.L., 1993. Computing minimal surfaces via level set curvature flow. J. Comput. Phys. 106, 77–91.

Cremers, D., Rousson, M., Deriche, R., 2007. A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. Int. J. Comput. Vis. 72 (2), 195–215.

Deasy, B., Huard, J., 2002. Gene therapy and tissue engineering based on muscle-derived stem cells. Curr. Opin. Mol. Ther. 4, 382–389.

Debeir, O., Ham, P.V., Kiss, R., Decaestecker, C., 2005. Tracking of migrating cells under phase-contrast video microscopy with combined mean-shift processes. IEEE Trans. Med. Imag. 24, 697–711.

Doucet, A., Ristic, B., 2002. Recursive state estimation for multiple switching models with unknown transition probabilities. IEEE Trans. Aerosp. Electron. Syst. 38 (3), 1098–1104.

Dufour, A., Shinin, V., Tajbakhsh, S., Guillen-Aghion, N., Olivo-Marin, J.-C., Zimmer, C., 2005. Segmenting and tracking fluorescent cells in dynamic 3D microscopy with coupled active surfaces. IEEE Trans. Image Process. 14 (9), 1396–1410.

Feng, H., Castanon, D.A., Karl, W.C., 2001. A curve evolution approach for image segmentation using adaptive flows. In: Proceedings of the IEEE International Conference on Computer Vision, vol. 2, pp. 494–499.

Genovesio, A., Liedl, T., Emiliani, V., Parak, W.J., Coppey-Moisan, M., Olivo-Marin, J.-C., 2006. Multiple particle tracking in $3D+t$ microscopy: method and application to the tracking of endocytosed quantum dots. IEEE Trans. Image Process. 15 (5), 1062–1070.

Godinez, W.J., Lampe, M., Wörz, S., Müller, B., Eils, R., Rohr, K., 2007. Tracking of virus particles in time-lapse fluorescence microscopy image sequences. In: Proceedings of the IEEE International Symposium on Biomedical Imaging, pp. 256–259.

Goldenberg, R., Kimmel, R., Rivlin, E., Rudzsky, M., 2001. Fast geodesic active contours. IEEE Trans. Med. Imag. 10 (10), 1467–1475.

Gordon, N.J., Salmond, D.J., Smith, A.F.M., 1993. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proc. F 140 (2), 107–113.

Han, X., Xu, C., Prince, J.L., 2003. A topology preserving level set method for geometric deformable models. IEEE Trans. Pattern Anal. Mach. Intell. 25 (6), 755–768.

Helmick, R.E., Blair, W.D., Hoffman, S.A., 1995. Fixed-interval smoothing for Markovian switching systems. IEEE Trans. Inform. Theory 41 (6), 1845–1855.

Herman, S.M., 2002. A particle filtering approach to joint passive radar tracking and target classification. Ph.D. Thesis.

Huang, S., Law, P., Francis, K., Palsson, B.O., Ho, A.D., 1999. Symmetry of initial cell divisions among primitive hematopoietic progenitors is independent of ontogenic age and regulatory molecules. Blood 94 (8), 2595–2604.

Huang, X., Metaxas, D., Chen, T., 2004. Metamorphs: deformable shape and texture models. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. I, pp. 496–503.

Jilkov, V.P., Li, X.R., 2004. Online Bayesian estimation of transition probabilities for Markovian jump systems. IEEE Trans. Signal Process. 52 (6), 1620–1630.

Kalman, Rudolph E., 1960. A new approach to linear filtering and prediction problems. Trans. ASME: J. Basic Eng. Ser. D 82, 35–45.

Kanade, T., Li, K., 2005. Tracking of migrating and proliferating cells in phase-contrast microscopy imagery for tissue engineering. In: Proceedings of the Computer Vision for Biomedical Image Applications Workshop, p. 24.

Kirubarajan, T., Bar-Shalom, Y., Pattipati, K.R., 2001. Multiassignment for tracking a large number of overlapping objects [and application to fibroblast cells]. IEEE Trans. Aerosp. Electron. Syst. 37 (1), 2–21.

Lefohn, A., Kniss, J., Hansen, C., Whitaker, R., 2004. A streaming narrow-band algorithm: interactive computation and visualization of level sets. IEEE Trans. Visual. Comput. Graph. 10, 422–433.

Li, K., Chen, M., Kanade, T., 2007. Cell population tracking and lineage construction with spatiotemporal context. In: Proceedings of the Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 295–302.

Li, K., Kanade, T., Chen, M., Miller, E.D., Weiss, L.E., Campbell, P.G., 2008. Computer vision tracking of stemness. In: Proceedings of the IEEE International Symposium on Biomedical Imaging, pp. 847–850.

Li, K., Miller, E.D., Weiss, L.E., Campbell, P.G., Kanade, T., 2006. Online tracking of migrating and proliferating cells imaged with phase-contrast microscopy. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop, pp. 65–72.

Mansouri, A.-R., Mitiche, A., Vazquez, C., 2006. Multiregion competition: a level set extension of region competition to multiple region image partitioning. Comput. Vis. Image Understand. 101, 137–150.

Meyer, F., 1979. A threshold selection method from gray level histograms. J. Histochem. Cytochem. 27 (1), 128–135.

Miki, T., Lehmann, T., Cai, H., Stolz, D.B., Strom, S.C., 2005. Stem cell characteristics of amniotic epithelial cells. Stem Cells.

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

Mukherjee, D.P., Ray, N., Acton, S.T., 2004. Level set analysis for leukocyte detection and tracking. IEEE Trans. Image Process. 13, 562–572.

Nath, S., Palaniappan, K., Bunyak, F., 2006. Wound cell segmentation using coupled level sets and graph-vertex coloring. In: Proceedings of the Medical Image Computing and Computer-Assisted Intervention, vol. LNCS 4190, pp. 101–108.

Osher, S., Sethian, J.A., 1988. Fronts propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations. J. Comput. Phys. 79, 12–49.

Otsu, N., 1979. A threshold selection method from gray level histograms. IEEE Trans. Syst. Man Cybern. 9, 62–66.

Padfield, D., Rittscher, J., Roysam, B., 2008. Spatio-temporal cell segmentation and tracking for automated screening. In: Proceedings of the IEEE International Symposium on Biomedical Imaging, pp. 376–379.

Padfield, D., Rittscher, J., Sebastian, T., Thomas, N., Roysam, B., 2006a. Spatio-temporal cell cycle analysis using 3D level set segmentation of unstained nuclei in line scan confocal fluorescence images. In: Proceedings of the IEEE International Symposium on Biomedical Imaging, pp. 1036–1039.

Padfield, D., Rittscher, J., Thomas, N., Roysam, B., 2006b. Spatio-temporal cell cycle phase analysis using level sets and fast marching methods. In: Proceedings of the Workshop on Microscopic Image Analysis with Applications in Biology, pp. 2–9. <http://www.miaab.org/miaab-2006-papers.html>.

Pan, Y., Birdwell, J.D., Djouadi, S.M., 2006. Efficient implementation of the Chan-Vese models without solving PDEs. In: Proceedings of the IEEE International Workshop on Multimedia and Signal Processing.

Papadimitriou, C.H., Steiglitz, K., 1998. Combinatorial Optimization: Algorithms and Complexity. Dover Publications.

Patrick, C.W., Wu, X., 2003. Integrin-mediated preadipocyte adhesion and migration on laminin-1. Ann. Biomed. Eng. 31, 505–514.

Perona, P., Malik, J., 1990. Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. 12 (7), 629–639.

Phillippia, J.A., Miller, E.D., Weiss, L.E., Huard, J., Waggoner, A.S., Campbell, P.G., 2008. Microenvironments engineered by inkjet bioprinting spatially direct adult stem cells toward muscle- and bone-like subpopulations. Stem Cells 26 (1), 127–134.

Ristic, B., Arulampalam, S., Gordon, N.J., 2004. Beyond the Kalman Filter: Particle Filters for Tracking Applications. Artech House Publishers.

Ségonne, F., 2005. Segmentation of medical images under topological constraints. Ph.D. Thesis.

Sethian, J.A. (Ed.), 1999. Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science, second ed. Cambridge University Press.

Shi, Y., Karl, W.C., 2005a. A fast level set method without solving PDEs. In: Proceedings of the IEEE International Conference Acoustics Speech and Signal Processing, vol. II, Philadelphia, PA, pp. 97–100.

Shi, Y., Karl, W.C., 2005b. Real-time tracking using level sets. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 34–41.

Smal, I., Draegestein, K., Galjart, N., Niessen, W., Meijering, E., 2007. Rao-blackwellized marginal particle filtering for multiple object tracking in molecular bioimaging. In: Proceedings of the International Conference on Information Processing in Medical Imaging, pp. 110–121.

Smal, I., Niessen, W., Meijering, E., 2006. Bayesian tracking for fluorescence microscopic imaging. In: Proceedings of the IEEE International Symposium on Biomedical Imaging, pp. 550–553.

Smith, A.F.M., Makov, U.E., 1978. A quasi-Bayes sequential procedure for mixtures. J. R. Stat. Soc. Ser. B 40 (1), 106–112.

Sternberg, S., 1983. Biomedical image processing. IEEE Comput. 16 (1), 22–34.

Vincent, L., Soille, P., 1991. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Trans. Pattern Anal. Mach. Intell. 13 (6), 583–598.

Weiss, L.E., Amon, C., Finger, S., Miller, E., Romero, D., Verdinelli, I., Walker, L., Campbell, P., 2005. Bayesian computer-aided experimental design of heterogeneous scaffolds for tissue engineering. Comput. Aided Des. 37, 1127–1139.

Yang, F., Mackey, M.A., Ianzini, F., Gallardo, G., Sonka, M., 2005a. Cell segmentation, tracking, and mitosis detection using temporal context. In: Duncan, J.S., Gerig, G. (Eds.), Proceedings of the Medical Image Computing and Computer-Assisted Intervention, vol. I, pp. 302–309.

Yang, X., Li, H., Zhou, X., Wong, S., 2005b. Automated segmentation and tracking of cells in time-lapse microscopy using watershed and mean shift. In: Proceedings of the International Symposium on Intelligent Signal Processing and Communication Systems, pp. 533–536.

Zarchan, P., Musoff, H., 2005. Fundamentals of Kalman Filtering: A Practical Approach, second ed. American Institute of Aeronautics and Astronautics Inc..

Zhang, B., Zimmer, C., Olivo-Marin, J.-C., 2004. Tracking fluorescent cells with coupled geometric active contours. In: Proceedings of the Second IEEE International Symposium on Biomedical Imaging (ISBI), Arlington, VA, pp. 476–479.

Zhu, S.C., Yuille, A., 1996. Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 18 (9), 884–900.

Zimmer, C., Labruy'ere, E., Meas-Yedid, V., Guill'en, N., Olivo-Marin, J.-C., 2002. Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: a tool for cell-based drug testing. IEEE Trans. Med. Imag. 21 (10), 1212–1221.

Zimmer, C., Olivo-Marin, J.-C., 2005. Coupled parametric active contours. IEEE Trans. Pattern Anal. Mach. Intell. 27 (11), 1838–1842.