

Cell Population Tracking and Lineage Construction with Spatiotemporal Context

Kang Li¹, Mei Chen², and Takeo Kanade¹

¹ Carnegie Mellon University

² Intel Research Pittsburgh

Abstract. Automated visual-tracking of cell populations *in vitro* using phase contrast time-lapse microscopy is vital for quantitative, systematic and high-throughput measurements of cell behaviors. These measurements include the spatiotemporal quantification of migration, mitosis, apoptosis, and cell lineage. This paper presents an automated cell tracking system that can simultaneously track and analyze thousands of cells. The system performs tracking by cycling through frame-by-frame track compilation and spatiotemporal track linking, combining the power of two tracking paradigms. We applied the system to a range of cell populations including adult stem cells. The system achieved tracking accuracies in the range of 83.8%–92.5%, outperforming previous work by up to 8%.

1 Introduction

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

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

However, several limitations are inherent in the aforementioned tracking system. First, all of its modules operate in a frame-by-frame manner. Hence, only very limited spatiotemporal context is considered, hindering the capability in handling complete or long term occlusions. Secondly, the track arbitrator module makes immediate, hard decisions for each frame, precluding the possibility for retrospective error detection and correction. Thirdly, the Kalman filter used for

motion filtering is bound to use only one dynamic model, which is problematic as the dynamics of cells vary frequently with time.

We propose an improved tracking system to address the above issues. Specifically, we divide the track arbitrator into two submodules: *track compiler* and *track linker*. Track compiler operates in a frame-by-frame manner and produces intermediate tracking results called *track segments*. Track linker oversees the entire tracking history and establishes final cell trajectories and lineages only when enough information is available. We also adopt the interacting multiple models (IMM) filter [2], which allows multiple dynamics models in parallel, and was shown to be more biologically relevant than a Kalman filter [3]. We focus on reliable long-term tracking of cell centroid locations and lineages. Accurate segmentation of cell boundaries is a plus, but not the emphasis of this paper.

2 Methods

The proposed tracking system has five major modules (Fig. 1).

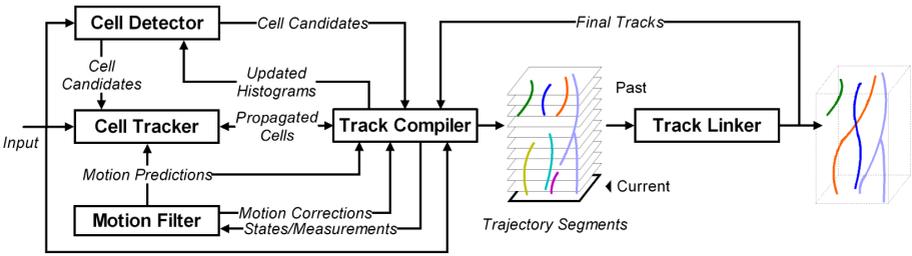


Fig. 1. System Overview

The trajectory of one cell may have multiple track segments. The system associates each cell track segment with a unique positive-integer label n . We identify each cell using the label of its first track segment. Let $k = 0, \dots, K - 1$ be the frame index. The cell regions in an image frame $I_k(x, y)$ are represented using a region labeling function $\psi_k(x, y)$. Wherein $\psi_k(x, y) = n$ if pixel (x, y) is part of cell n , and $\psi_k(x, y) = 0$ if pixel (x, y) belongs to the background. To initialize tracking, the system generates initial cell labeling $\psi_0(x, y)$ by running the cell detector on the first frame $I_0(x, y)$. For each subsequent frame $I_k(x, y)$:

Step 1: The cell detector classifies image pixels into cell (C) and background (B) classes based on histograms learned off-line and updated online [1]. The output is a binary map of cell regions, denoted $\zeta_k(x, y)$. Each connected foreground component is considered a *cell candidate* in frame k .

Step 2: The cell tracker propagates cell region labeling $\psi_{k-1}(x, y)$ from frame $k - 1$ to frame k , denoted $\psi_k^*(x, y)$, using a real-time level set method [4].

Step 3: The track compiler compares the output of the cell detector and cell tracker, and take one of the following actions: to create a new or daughter track

segment, to update an existing track, or to terminate a track. Meanwhile, the *motion filter* updates the cell motion state in frame k , and predicts its state for frame $k + 1$. The output includes the track segments, an updated region labeling $\psi_k(x, y)$, and updated cell and background histograms.

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

The following sections elaborate on the motion filter, track compiler, and track linker. We refer readers to [1] for details on the other modules.

2.1 IMM Motion Filter

Suppose cell motion consists of a finite number of modes, each of which can be described by a linear model with additive Gaussian noise. The IMM filter [2] 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, where $i, j \in \{1, \dots, M\}$ is the model index.

We define the state vector $\mathbf{s}_{n,k}$ to be a concatenation of centroid locations and mean intensities of cell n in frames $k, k - 1$ and $k - 2$, and the measurement vector $\mathbf{z}_{n,k}$ to consist of the measured centroid and mean intensity. We adopted the random walk, first-order, and second-order linear extrapolation models given in [3], and estimated the process and noise covariance matrices \mathbf{Q}^i and \mathbf{R}^i by the expectation maximization (EM) algorithm [5] utilizing three manually-tracked sequences. The filtering cycle has two recursive stages: *prediction* and *correction*.

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

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

$$\mathbf{P}_{n,k-1}^{0j} = \sum_i \rho_{n,k-1}^{i|j} \left[\mathbf{P}_{n,k-1}^i + \left(\hat{\mathbf{s}}_{n,k-1}^i - \hat{\mathbf{s}}_{n,k-1}^{0j} \right) \left(\hat{\mathbf{s}}_{n,k-1}^i - \hat{\mathbf{s}}_{n,k-1}^{0j} \right)^T \right], \tag{2}$$

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

$$\hat{\mathbf{s}}_{n,k|k-1} = \sum_j \rho_{k|k-1}^j \hat{\mathbf{s}}_{n,k|k-1}^j, \tag{3}$$

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

These are fed to the cell tracker to guide the level set evolution in frame k [1].

Correction: Given the predicted states, covariances, and measurement $\mathbf{z}_{n,k}$ (Fig. 1), we use the Kalman filters to obtain the updated state $\hat{\mathbf{s}}_{n,k}^j$ and covariance $\mathbf{P}_{n,k}^j$. The likelihood that model j is activated in frame k is

$$\lambda_{n,k}^j = \exp \left[-\frac{1}{2} (\mathbf{y}_{n,k}^j)^T (\mathbf{S}_{n,k}^j)^{-1} \mathbf{y}_{n,k}^j \right] / \sqrt{2\pi \det(\mathbf{S}_{n,k}^j)}, \quad (5)$$

where $\mathbf{y}_{n,k}^j = (\mathbf{z}_{n,k} - \hat{\mathbf{z}}_{n,k|k-1}^j)$ is the innovation of Kalman filter j , and $\mathbf{S}_{n,k}^j$ is the associated covariance. Then, the combined state $\hat{\mathbf{s}}_{n,k}$ and covariance $\mathbf{P}_{n,k}$ estimates can be computed by Equations (3) and (4), with $\rho_{n,k|k-1}^j$ replaced by $\rho_{n,k}^j = \rho_{n,k|k-1}^j \lambda_{n,k}^j / (\sum_i \rho_{n,k|k-1}^i \lambda_{n,k}^i)$.

To initialize the motion filter, the system tracks each cell without motion filtering in the first three frames that it appears, and use the concatenation of measurements in these frames as its initial state $\hat{\mathbf{s}}_{n,0}$. We set the initial covariance $\mathbf{P}_{n,0}^i$ of model i to be the Kronecker product of a 3×3 identity matrix and \mathbf{R}^i .

2.2 Track Compilation

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

Algorithm 1. TrackCompilation

```

 $\Omega_0 \leftarrow \{(x, y) | \psi_k^*(x, y) = 0\}$ 
1 foreach cell candidate  $\omega \subset \zeta_k$  do
  | if  $\omega \subset \Omega_0$  then  $AddTrack(n_{new}, k, \omega)$ 
2 foreach active track  $n \in \mathbf{N}_{k-1}$  do
  |  $\Omega_n \leftarrow \{(x, y) | \psi_k^*(x, y) = n\}$ 
3   if  $\Omega_n = \emptyset$  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   |   |    $\omega^* \leftarrow SelectBestMatch(n, k, \Omega_n)$ 
  |   |   |    $UpdateTrack(n, k, \omega^*)$ 
  |   |   |   foreach connected component  $\omega \subset \Omega_n \setminus \omega^*$  do  $AddTrack(n_{new}, k, \omega)$ 
6   |   else  $UpdateTrack(n, k, \Omega_n)$ 

```

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

Next, the algorithm scans through all active track segments, and deactivates track segments whose labels are not found in the propagated region labeling $\psi_k^*(x, y)$. A track segment whose corresponding propagated cell region contains only one connected component will be updated directly. If a cell region consists of more than one well-separated connected components, the track compiler will judge between two possibilities: 1) the cell divided into daughter cells; or 2) one or more of these components are from occluded cells or close-by newly-entered cells. The algorithm will either create daughter tracks or continue tracking using the component that best matches the cell trajectory, depending on whether the cell is previously detected to be mitotic.

Details of several key operations are as follow.

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

SelectBestMatch(n, k, Ω_n) selects component $\omega^* \in \Omega_n$ that best matches the dynamics of cell n , i.e., the one which maximizes the innovation likelihood given by Equation (5) among all dynamic models.

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

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

2.3 Track Linking

The track linker detects potential problems among all track segments up to frame k based on two physical constraints: 1) a cell does not vanish unless it leaves the field-of-view, dies and releases into the media, or is occluded; and 2) a cell does not appear unless it enters from outside, is produced as a daughter cell, or moves out of occlusion. The linker attempts to correct these problems by linking track segments into complete cell trajectories using spatiotemporal context.

Algorithm 2 outlines the track linking algorithm. Wherein, \mathbf{N}_{lost} denotes the label set of track segments that ended before frame k , and $\mathbf{N}_{\text{found}}$ is the label set of track segments whose starting point is after frame 0. Most operations in the algorithm are self-explanatory. One vital step of the algorithm is the matching between lost and appeared track segments, *MatchTracks* (Line 5).

In *MatchTracks*, we first create a bipartite graph G , whose nodes correspond to the labels in \mathbf{N}_{lost} and $\mathbf{N}_{\text{found}}$. We construct the arcs of G as follows. We create an arc $\langle n_l, n_f \rangle$ between node n_l and node n_f if the last centroid (x_l, y_l, k_l) of track n_l is related to the first centroid (x_f, y_f, k_f) of track n_f by $\sqrt{(x_l - x_f)^2 + (y_l - y_f)^2} \leq R$, and $|k_l - k_f| \leq H/2$, where H and R are user-defined parameters. Each arc $\langle n_l, n_f \rangle$ is assigned a weight $w_{lf} = \lambda_{n_l, k_f}^{\max}(n_f)$, which is the maximum innovation likelihood of track n_l on the measurement of track n_f in frame k_f (Eq. 5). Intuitively, w_{lf} indicates how likely track n_f is a continuation of track n_l based on the dynamics of n_l .

Algorithm 2. TrackLinking

```

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

```

Next, we obtain a maximum-likelihood matching between tracks n_l and n_f . This correspond to selecting the subset of arcs in graph G with maximum total weight, subject to the constraint that no two arcs share a common node. We adopt the efficient Jonker-Volgenant (JV) algorithm [6]. Specifically, we define a square cost matrix \mathbf{C} with dimension $d = \max(|\mathbf{N}_{\text{lost}}|, |\mathbf{N}_{\text{found}}|)$, where $|\cdot|$ denotes the size of a set. The entry of the cost matrix $\mathbf{C}(l, f)$ equals $1 - w_{lf}$ if $\langle n_l, n_f \rangle \in G$, or 1 otherwise. Taking \mathbf{C} as input, the JV algorithm outputs a minimum-cost column-to-row (or vice versa) assignment. Track segments n_l and n_f are *matched* if column f is assigned to row l and $\mathbf{C}(l, f) < 1$.

3 Experiments and Results

We quantitatively analyzed the performance of our system on two image sequences (A and B) of MG-63 osteosarcoma cells used previously in [1], and two sequences (C and D) of proprietary amnion epithelial (AE) stem cells.

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

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

A human operator manually tracked the cell centroids in Sequences A and B, and two randomly-selected 256×256 -pixel subregions in Sequences C and D, respectively (Fig. 2). Only those cells that appear in the initial frame of each sequence and their children were tracked. A cell trajectory is *valid* only if it followed the *same* cell through all frames that the cell is visible. The operator also manually identified all mitosis events. We compared the tracking results produced by the current and the previous systems, as shown in Table 1.

We visually compared the current tracking results with those produced by the previous system [1] for more than 30 sequences of AE stem cells. The new system showed superior robustness in handling long-term occlusion and against cell detection error. Fig. 3 shows an example where cell 116 is occluded by cell 47 in frame 36 and reappeared in frame 46. The new system (top row) correctly

Table 1. Tracking Accuracy Comparison

Sequences	Trajectory Validity				Division Tracking Correctness			
	Current		Previous		Current		Previous	
A	74/81	(91.4%)	70/81	(86.4%)	1/1	(100%)	1/1	(100%)
B	86/93	(92.5%)	82/93	(88.2%)	0/0	(N/A)	0/0	(N/A)
C	78/92	(84.8%)	70/92	(76.1%)	45/55	(81.8%)	43/55	(78.2%)
D	98/117	(83.8%)	90/117	(76.9%)	44/52	(84.6%)	41/52	(78.8%)

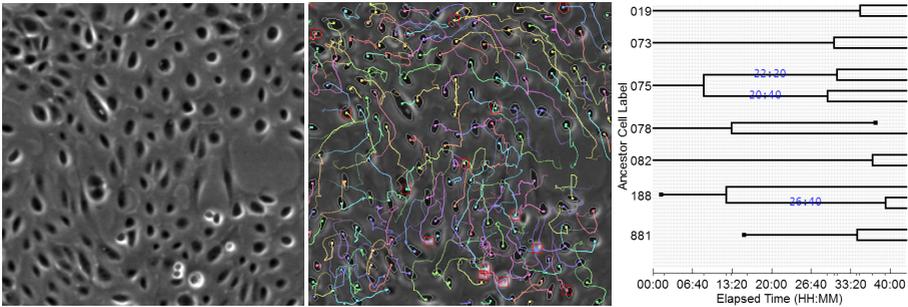


Fig. 2. Tracking results of AE cells in the subregion used for quantitative validation. **Left:** original image. **Middle:** the image with cell trajectories overlaid. Red rectangles indicate cells that were detected mitotic in the past $T = 10$ frames. **Right:** lineage map for selected cells. Black squares indicate cell entrance or departure. Blue text shows division time.

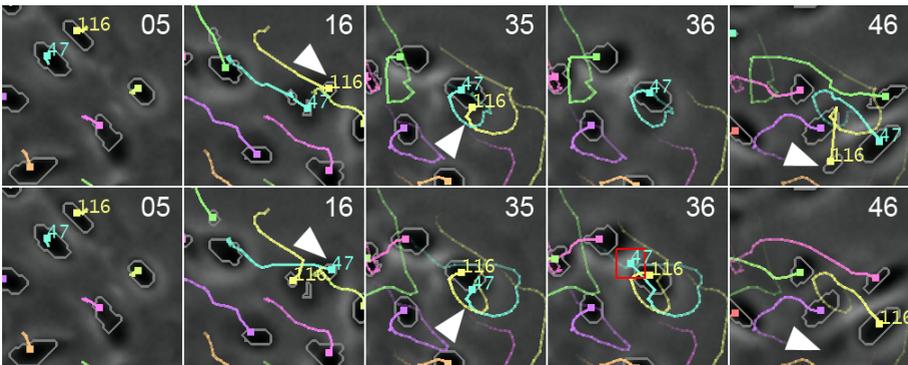


Fig. 3. Tracking AE cells through occlusion. **Top row:** The new system correctly tracked cell 116, which was completely occluded by cell 47 and reappeared later. **Bottom row:** Incorrect result was produced by the previous system, where cell 116 switched with cell 47 and was lost eventually. The numbers at the top-right corner are the frame indices. The trailing curves represent cell trajectories. Different colors represent different cell lineages.

recovered the trajectory of cell 116 after occlusion, whereas the previous system (bottom row) switched the identities of cells 47 and 116 in frame 16, detected a false mitosis in frame 36, and eventually lost cell 47 after frame 36.

Another application of the tracking system that is valuable to stem cell research is to automatically reconstruct cell lineage maps. We used the system to construct the lineages for the whole population of AE cells. Fig. 2(c) shows a sample set of the lineage trees with cells undergoing multiple divisions.

Our system (implemented in ISO C++) runs at an average speed of 90 frames/hour for tracking approximately 3000 cells in a 1280×1024 pixels/frame image sequence on an Intel Xeon 2.66GHz workstation.

4 Conclusion and Future Work

We developed and validated an automated system capable of tracking thousands of individual cells in dense cell populations in phase contrast microscopy image sequences. The system incorporated spatiotemporal track linking and a biologically relevant motion filter, and achieved performance boosts of up to 8% compared to its predecessor with nominal computational overhead. We plan to incorporate more effective segmentation algorithms and graphical models to cope with more complex intercellular interactions.

Acknowledgements

We would like to thank Eric Miller, Dr. Phil Campbell and Dr. Lee Weiss for their help. This work was supported partially by NIH Grant R01 EB007369, and the Pennsylvania Infrastructure Technology Alliance Grant 1C76 HF 00381-01.

References

1. Li, K., Miller, E.D., Weiss, L.E., Campbell, P.G., Kanade, T.: Online tracking of migrating and proliferating cells imaged with phase-contrast microscopy. In: Proc. IEEE Conf. Comp. Vision and Patt. Recog. Workshop, p. 65. IEEE Computer Society Press, Los Alamitos (2006)
2. Blom, H.A.P.: An efficient filter for abruptly changing systems. In: Proc. 23rd IEEE Conference on Decision and Control, pp. 656–658. IEEE Computer Society Press, Los Alamitos (1984)
3. Genovesio, A., Liedl, T., Emiliani, V., Parak, W.J., Coppey-Moisan, M., Olivo-Marin, J.C.: Multiple particle tracking in 3-D+t microscopy: Method and application to the tracking of endocytosed quantum dots. *IEEE Transactions of Medical Imaging* 15, 1062–1070 (2006)
4. Shi, Y., Karl, W.C.: Real-time tracking using level sets. In: Proc. IEEE Conf. Comp. Vision and Patt. Recog., vol. 2, pp. 34–41. IEEE Computer Society Press, Los Alamitos (2005)
5. Shumway, R., Stoffer, D.: An approach to time series smoothing and forecasting using the EM algorithm. *Journal of Time Series Analysis* 3, 253–264 (1982)
6. Jonker, R., Volgenant, A.: A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing* 38, 325–340 (1987)