

SCOTT: Shape-Location Combined Tracking with Optimal Transport*

Xinye Zheng[†], Jianbo Ye[‡], James Z. Wang[§], and Jia Li[¶]

Abstract. Optimal transport (OT) is a prominent framework for point set registration, that is, to align points in two sets. Point set registration becomes particularly difficult when points are organized into objects and the correspondence among the objects is to be established. The registration of pixels must maintain consistency at the object level despite the possibility of object division, merging, and substantial alteration in size and shape over time. Existing approaches in OT exploit either similarity in shape for the entire set of points or spatial closeness of individual points, but not the two simultaneously. We propose a new weighted Gromov–Wasserstein distance (WGWD) to combine both sources of information. Importantly, we use a bipartite graph partitioning strategy to regularize OT in order to achieve object-level consistency and to enhance computational efficiency. We apply the method to cell tracking, specifically, the task of associating biological cells in consecutive image frames from time-lapse image sequences. We call the system SCOTT (Shape-Location COmbeded Tracking with Optimal Transport). By establishing a pixel-to-pixel correspondence, our method can effectively detect intricate scenarios including cell division and merging (overlapping). Experiments show that our method achieves high accuracy in tracking the movements of cells and outperforms existing methods in the detection of cell division and merging. Location information is shown to be more useful than shape information, while the combination of the two achieves optimal results.

Key words. optimal transport, point set registration, cell tracking, biological imaging

AMS subject classifications. 92C55, 68U10, 82C70

DOI. 10.1137/19M1253976

1. Introduction. In order to understand cell activities, such as division (mitosis), migration, and death (apoptosis), biologists must track individual cells in time-lapse microscopy images. This tedious process incentivizes automatic tracking of cells and is highly valuable for research in many areas, including genomics, bacteriology, and medical science. There are two major categories of approaches to cell tracking, namely, *tracking based on cell detection* [2, 21, 22, 27] and *tracking by evolving models* [14, 16, 40, 55]. The approach of tracking based on cell detection contains two steps: (1) cells are detected in each image frame, and (2)

*Received by the editors April 4, 2019; accepted for publication (in revised form) February 3, 2020; published electronically April 8, 2020.

<https://doi.org/10.1137/19M1253976>

Funding: This material is based upon work supported by the National Science Foundation (NSF) under grant 1521092. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by NSF grant 1548562. The work was also supported by the Amazon AWS Cloud Credits for Research Program and the NVIDIA Corporation's GPU Grant Program.

[†]College of Information Sciences and Technology, The Pennsylvania State University, University Park, PA 16802 (xvz5220@psu.edu).

[‡]Amazon.com, Inc., Sunnyvale, CA 94089 (jianboye.ai@gmail.com). This author contributed to this work when he was with The Pennsylvania State University.

[§]College of Information Sciences and Technology and Huck Institutes of the Life Sciences, The Pennsylvania State University, University Park, PA 16802 (jwang@ist.psu.edu).

[¶]Department of Statistics, The Pennsylvania State University, University Park, PA 16802 (jiali@stat.psu.edu).

the detected cells are aligned across the image frames. For tracking by evolving models, cell detection and tracking are tackled simultaneously based on models with parameters evolving with time. One main challenge in cell tracking for the existing approaches is aligning cells in consecutive frames. A cell is specified either by its centroid location in the image plane or by an expanded feature vector containing its location and other features, e.g., intensity and volume. A common idea is to match cells from two frames to their nearest neighbors in the feature space. This approach requires that cells are well separated in at least one dimension of the feature space [30].

Although the cells are described by the group of pixels in the detection stage, during the cell alignment phase, their specification is simplified into a certain parametric form, e.g., its centroid location [1, 7, 12], or its Gaussian distribution fitted onto pixel coordinates [41, 46]. However, these simplified specifications may lose considerable information about the shapes of the cells. Figure 1 shows an example from pancreatic stem cells (PSCs) [37]. The PSCs have rather irregular shapes, which cannot be captured by the aforementioned descriptors. Cell A evolves into cell C, as indicated by similar shapes, despite the fact that cell B is closer to cell A spatially.

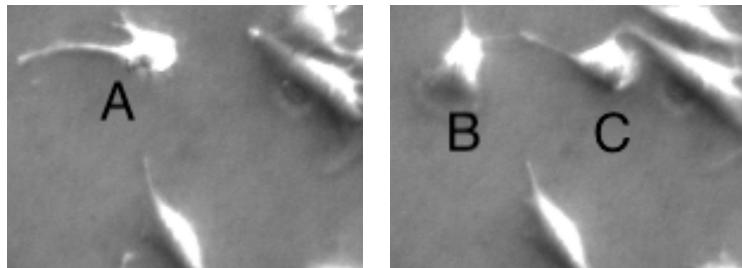


Figure 1. An example of cell alignment in two consecutive PSC microscopic images. Although cell B is closer to cell A, cell A actually moved and became cell C, which is more similar to cell A in shape.

To address this challenge, we propose a new approach called SCOTT (Shape-Location COmbined Tracking with Optimal Transport). SCOTT exploits cell detection, but in the alignment phase, pixel-level tracking is performed instead of cell-level tracking. The detected cells impose constraints on the alignment of pixels, and conversely, the result of pixel alignment is used to adjust the detected cells. The motivation for pixel-level tracking is to better preserve the shape of the cells during alignment because the cells will be specified in their original forms, that is, a group of pixels. However, in order to be sensitive to shape, the choice of the optimization criterion for pixel alignment matters. This issue has been addressed in the literature of optimal transport (OT) [35, 48].

We specify a cell by a set of 2D pixel coordinates. The problem of *pixel-level tracking* is transformed into *point-set registration* between two consecutive frames, which is a widely studied problem in OT. A commonly used criterion for registering two point sets is to minimize the total transportation cost, often called the Wasserstein distance (WD) [13], which is a linear cost in terms of the matching weights to be optimized. Point set registration based on WD has been used to solve object tracking [9, 39]. However, if the goal is to best preserve the shape of the point set after the transport, then a quadratic cost should be used, which results

in the Gromov–Wasserstein distance (GWD) [35]. GWD aims at keeping the fidelity of the similarity matrix between points in one set instead of the absolute locations of the points themselves. As a result, GWD tends to keep the geometric relationship between the points in one set. See [35] for the detailed mathematical justification.

In SCOTT, because we need to consider both spatial location similarity and shape similarity between cells, we employ a new transportation cost which is a weighted linear combination of the WD cost and the GWD cost, referred to as weighted Gromov–Wasserstein distance (WGWD). In addition, our problem has another layer of complexity because the pixels are grouped into multiple cells in one image. When the pixels of one cell in an image are mapped to pixels in another image, it is only sensible if the mapped pixels also form a cell. The original WD or GWD problems have no such concern. We have developed a bipartite graph partitioning (BGP) strategy to ensure that the pixels are transported consistently with respect to their memberships to the cells. In addition, BGP adopts a two-stage process, in which OT based on WD is used in the first stage and the computationally more intensive OT based on WGWD is evoked in the second stage only for small groups of cells that are difficult to track.

By establishing a pixel-to-pixel correspondence in microscopy image frames, SCOTT performs better in revealing complex situations, such as division of cells or the appearance of multiple cells merging into one when they are adjacent to or overlapping with each other. For brevity, we use the term “merger” to mean that multiple cells blend into one blob in the image. This technical “artifact” in imaging can last over a widely varying length of time. Merging is not a biological process, while cell division is. As a result, when cells merge, to precisely track them, their individual identities should be kept. Pixel-level registration enables us to detect the case of merging as well as division.

In summary, the major *contribution* of our work is a new OT-based framework for cell tracking. We have proposed WGWD to combine location and shape information. In addition, we have developed a new algorithm that integrates BGP with OT so that pixel-level registration can be solved more accurately and efficiently for high-resolution microscopic images. Based on the pixel-level information, we can better track cells.

The remainder of this paper is organized as follows. We begin by reviewing previous work in cell tracking in [section 2](#). The pixel-level tracking setup with OT is discussed in [section 3](#), with the problem formulation and OT background introduced in [subsection 3.1](#), followed by the definition and computation method of our WGWD in [subsection 3.2](#). Our SCOTT framework is detailed in [section 4](#), with the cell-level transportation defined in [subsection 4.1](#), the cell alignment method in [subsection 4.2](#), and the cell division detection in [subsection 4.3](#). The experimental setup and analyses are described in [section 5](#). We discuss some limitations in [section 6](#) and conclude in [section 7](#).

2. Related work. Managing cell alignment between consecutive frames is a challenging task in cell tracking. Most existing methods align the cells according to an optimization criterion based on distances between cells in two image frames. A cell is typically described by a feature vector which is used to determine the cell-level distance. Such features include center coordinates in the image plane [12, 37], histogram of pixel intensities [24], and contour-based features [26]. Various techniques have been developed to optimize the alignment, e.g., multi-assignment [22], integer programming [2, 24, 46], minimal cost flow [54], dynamic pro-

gramming [27], OT [9], graphical models [21, 41], and minimization of contour changes [14, 16].

Cell shape is an important cue to distinguish multiple cells when they merge into one blob in the image [45]. Some contour-based features of the cell, such as the concave degree [51] and the parametric active contour models [8, 23], are proposed to assist cell detection. However, contour-based methods rely heavily on the accuracy of cell detection [46] as well as prior knowledge about the cell shape [23]. Instead of using boundary solely, we use the full set of pixels included in a cell to better capture shape characteristics.

Mitosis (cell division) is an import process during which the genetic material of a parental cell is equally distributed between its descendants through nuclear division. We can compute valuable features for a population after identifying mitosis, e.g., the proliferation rate. Capturing mitosis timely and accurately is challenging because two daughter cells may adhere to each other for a stretch of time [45]. Various methods have been developed for detecting mitosis. For example, [52] computes multiple cell features for each blob region; [25] uses Haar-like filters to extract features of the cell; and [51] measures the extent of the concavity of the boundary (roughly, how severe a convex contour is dented). These methods are usually designed for specific types of cells and are not generalizable. We attempt to develop a general framework for broad cell tracking applications.

3. OT for cell tracking. In this section, we first cast the cell tracking problem into the OT framework in subsection 3.1, identifying the problem as essentially point set registration. We solve the pixel-level alignment and use it to infer cell alignment from the pixels. Importantly, the shape of a cell is captured by all the pairwise distances between pixels in a cell, that is, a pixel-level similarity matrix, which contains richer information than a few hand-crafted features. We introduce the new distance between two point sets, WGWD, in subsection 3.2. As will be elaborated on later, WGWD takes into account both the location and shape difference between cells.

3.1. Mathematical formulation with OT. OT solves the minimal transportation plan between two discrete distributions. Denote the two distributions by X_i , $i = 0, 1$. X_i is specified by a total of n_i support points, each assigned with a probability. Denote a support point of X_i by $x_{i,j} \in \mathbb{R}^d$, where \mathbb{R}^d is a d -dimensional Euclidean space. In our case, $d = 2$ for pixels specified by their coordinates in the 2D image plane. Let the probability assigned to $x_{i,j}$ be $w_{i,j}$. Then $X_i = \{(x_{i,1}, w_{i,1}), (x_{i,2}, w_{i,2}), \dots, (x_{i,n_i}, w_{i,n_i})\}$, $i = 0, 1$. We have $\sum_j w_{i,j} = 1$ for $i = 0, 1$. Let $a_i = (w_{i,1}, \dots, w_{i,n_i}) \in \mathbb{R}_+^{n_i}$ be the weight simplex for $i = 0, 1$. Denote the Euclidean distance matrix between the support points in X_0 and X_1 by $M = (m_{j,j'}) \in \mathbb{R}_+^{n_0 \times n_1}$, where $m_{j,j'}$ is the distance between $x_{0,j}$ and $x_{1,j'}$. A transportation plan, or a point registration plan, specifies how the mass of simplex a_0 is redistributed to simplex a_1 . The transportation plan can be represented by a matrix $T = (t_{j,j'}) \in \mathbb{R}^{n_0 \times n_1}$, where element $t_{j,j'}$ is the weight assigned from $x_{0,j}$ to $x_{1,j'}$. Denote $\mathbf{1}_n$ as an n -dimensional vector with all the elements equal to one and $\mathbf{1}_{n_0 \times n_1}$ as an $(n_0 \times n_1)$ -dimensional matrix with all the elements equal to one. The set of transportation plans from a_0 to a_1 is defined as

$$(3.1) \quad U(a_0, a_1) \stackrel{\text{def}}{=} \left\{ T \in \mathbb{R}_+^{n_0 \times n_1} \mid T\mathbf{1}_{n_1} = a_0, T'\mathbf{1}_{n_0} = a_1 \right\}.$$

Intuitively, we can view the weight $t_{j,j'}$ as the mass transported from $x_{0,j}$ to $x_{1,j'}$, and

$m_{j,j'}$ is the cost to transport a unit mass between them. Then OT aims to find the optimal transportation plan $T \in U(a_0, a_1)$ that has the minimal total transportation cost.

Wasserstein distance (WD) [38, 50] is the minimal total *spatial distance* to transport masses located at the support points of X_0 to the support points of X_1 subject to the mass weights specified by a_0 and a_1 . Denote the Frobenius dot-product of two matrices by $\langle A, B \rangle = \text{tr}(AB')$. The 1-Wasserstein distance is defined by

$$(3.2) \quad \text{WD}(M, a_0, a_1) \stackrel{\text{def}}{=} \min_{T \in U(a_0, a_1)} \langle T, M \rangle .$$

Note that if matrix M is defined by the p th power of the distance between the points, the above equation defines the p th power of the p -Wasserstein distance. In our context, we restrict ourselves to 1-Wasserstein distance with Euclidean distance in M as the baseline distance between points.

One of the state-of-the-art methods to solve the point registration is to solve its entropic regularized function with Sinkhorn iterations [11]. With the entropy function $h(T) \stackrel{\text{def}}{=} -\sum_{j,j'} t_{j,j'} \log t_{j,j'}$ and a small number ε , the approximated point registration $\mathcal{T}_{\text{WD},\varepsilon}$ is defined as

$$(3.3) \quad \mathcal{T}_{\text{WD},\varepsilon}(M, a_0, a_1) \stackrel{\text{def}}{=} \arg \min_{T \in U(a_0, a_1)} \langle T, M \rangle - \varepsilon h(T) .$$

According to the analysis of the MNIST dataset [11] and our experiments, the difference between $\mathcal{T}_{\text{WD},\varepsilon}$ and the accurate registration in WD can be ignored when $\varepsilon < 1e-2$.

Gromov–Wasserstein distance (GWD), on the other hand, minimizes the total *intrinsic geometry dissimilarity* to transport X_0 to X_1 [31, 35]. For some specific data such as networks, the internal structure of data is much more meaningful than the spatial locations. Besides a_i , we use the pairwise distance matrix $C_i \in \mathbb{R}_+^{n_i \times n_i}$ to represent shapes of the data X_i because the C_i is more robust in coordination translation, scaling, and rotation. Let $\mathcal{L}(a, b) = \frac{1}{2}|a - b|^2$ be the quadratic distance and $L \otimes T \stackrel{\text{def}}{=} (\sum_{k,l} L_{i,j,k,l} T_{k,l})_{i,j}$ be the tensor-matrix multiplication.

$$(3.4) \quad L(C_0, C_1) \stackrel{\text{def}}{=} \left(\mathcal{L}((C_0)_{i,k}, (C_1)_{j,l})_{i,j,k,l} \right)$$

is a 4-way tensor. The GWD between (C_0, a_0) and (C_1, a_1) is defined as

$$(3.5) \quad \text{GWD}(C_0, C_1, a_0, a_1) \stackrel{\text{def}}{=} \min_{T \in U(a_0, a_1)} \langle T, L(C_0, C_1) \otimes T \rangle .$$

The point registration computed with GWD aims to preserve the similarity matrix for the points in one set after the transportation instead of preserving the locations of individual points.

The formula of GWD defines a nonconvex problem. Similar to WD, GWD can be approximated by its entropic regularization. The point registration $\mathcal{T}_{\text{GWD},\varepsilon}$ deducted from Kullback–Leibler (KL) projection iterations in [35] is defined as

$$(3.6) \quad \mathcal{T}_{\text{GWD},\varepsilon}(C_0, C_1, a_0, a_1) \stackrel{\text{def}}{=} \arg \min_{T \in U(a_0, a_1)} \langle T, L(C_0, C_1) \otimes T \rangle - \varepsilon h(T) .$$

3.2. Weighted Gromov–Wasserstein distance. For cell tracking, similarities in both location and shape are valuable cues to establish the correspondence between cells in multiple image frames. As explained in the previous section, WD minimizes a weighted sum of location differences, while GWD aims at retaining the geometric characteristics of the point set and the geometry is captured in a rich manner by a similarity matrix between the points. We are thus motivated to combine the WD and GWD and propose the weighted Gromov–Wasserstein distance (WGWD). Letting $\lambda \in [0, 1]$ be the weight to balance the influence of shape versus location, WGWD is defined as

$$(3.7) \quad \text{WGWD} (C_0, C_1, M, a_0, a_1, \lambda) \stackrel{\text{def}}{=} \min_{T \in U(a_0, a_1)} \langle T, \lambda L(C_0, C_1) \otimes T + (1 - \lambda) M \rangle.$$

Here WD and GWD can be viewed as special cases of WGWD at $\lambda = 0$ and $\lambda = 1$, respectively.

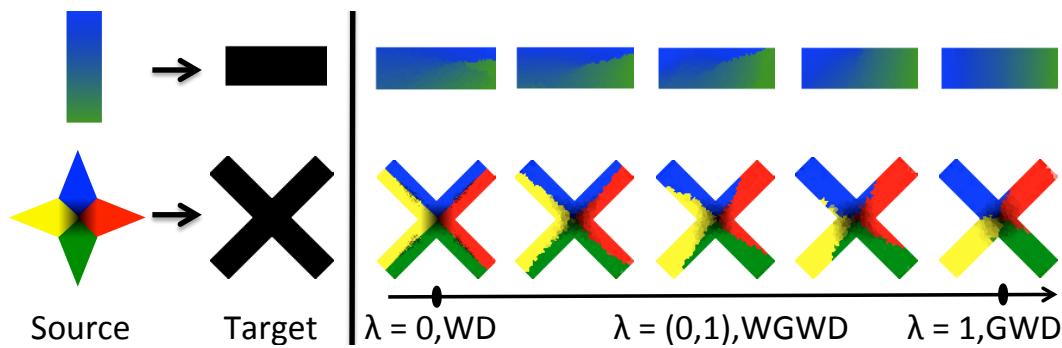


Figure 2. Point registration evolution with the increase of weight parameter λ . Left: Two pairs of point sets aligned by OT. For the two rectangles in the first row, the gravity center of the source shape has the same position in the image plane as the midpoint on the left side border of the target shape. In the second row, the gravity centers of the source shape and the target shape locate at the same position. Right: Visualization of point registration with the increase of λ .

An example of how the point registration changes with the weight parameter λ is shown in Figure 2, where we aim to register points in the source to those in the target in the 2D coordinate. We use the gradually changed color in the source shape to represent the registered locations in the target. As shown, points are registered according to the minimized total spatial distance when $\lambda = 0$, and then the points are registered according to the most geometric similarity when $\lambda = 1$. When λ changes from 0 to 1, the shape is becoming increasingly important compared with the location.

Similar to the process to compute WD and GWD, WGWD can be approximated by its entropy regularization, i.e.,

$$(3.8) \quad \mathcal{T}_{\text{WGWD}, \varepsilon} (C_0, C_1, M, a_0, a_1, \lambda) \stackrel{\text{def}}{=} \arg \min_{T \in U(a_0, a_1)} \langle T, \lambda L(C_0, C_1) \otimes T + (1 - \lambda) M \rangle - \varepsilon h(T).$$

We use the projected gradient descent in the KL metric space similar to equation (8) in [35] to solve the problem. The iterations are given by

$$(3.9) \quad T \leftarrow \text{proj}_{U(a_0, a_1)}^{\text{KL}} \left(T \odot e^{-\tau \nabla K} \right),$$

where $a \odot b = (a_i b_i)$ is the elementwise product, $K = \langle T, \lambda L(C_0, C_1) \otimes T + (1 - \lambda) M \rangle - \varepsilon h(T)$, and the KL projector on any matrix N is defined as

$$(3.10) \quad \text{proj}_{U(a_0, a_1)}^{\text{KL}}(N) \stackrel{\text{def}}{=} \arg \min_{T' \in U(a_0, a_1)} \text{KL}(T'|N) .$$

According to [5], the KL projection gives the solution to the entropy-regularized Wasserstein distance $\mathcal{T}_{\text{WD}, \varepsilon}$, i.e.,

$$(3.11) \quad \text{proj}_{U(a_0, a_1)}^{\text{KL}}(N) = \mathcal{T}_{\text{WD}, \varepsilon}(-\varepsilon \log N, a_0, a_1) .$$

Take the gradient descent on T , we have

$$(3.12a) \quad \nabla K = \nabla \langle T, \lambda L(C_0, C_1) \otimes T + (1 - \lambda) M \rangle - \nabla \varepsilon h(T)$$

$$(3.12b) \quad = \lambda L(C_0, C_1) \otimes T + (1 - \lambda) M + \varepsilon \log(T) .$$

Let $\tau \varepsilon = 1$, and substitute (3.11) and (3.12b) into (3.9). We get the final iteration:

$$(3.13) \quad T \leftarrow \mathcal{T}_{\text{WD}, \varepsilon}(\lambda L(C_0, C_1) \otimes T + (1 - \lambda) M, a_0, a_1) ,$$

which solves the entropy-regularized WGWD in (3.8).

4. SCOTT cell tracking system. Ideally, we can view cell tracking as a pixel matching problem formulated by OT with hard matching (each pixel is mapped entirely to another pixel) under the coupling constraint that all pixels in one cell are mapped to pixels in another cell. To have feasible solutions, a cell is allowed to drop or expand its pixels in a new image frame. This constrained OT problem is NP-hard [36]. We solve an approximation by combining the following ideas:

- The *basic OT* problem allows soft matching between pixels and does not assume an equal number of pixels in the cells of the two image frames. In order to obtain hard matching between pixels, that is, each pixel matching entirely with another pixel, virtual pixels/points are added so that the mapping between virtual and actual pixels indicates changes in cell mass.
- To speed up the basic OT, we use multiscale OT as an approximation so that the complexity of matching is reduced to $O(n \log n)$, where n is the number of cell pixels in one image frame.
- The basic OT does not ensure that all the pixels in one cell match with pixels in another cell. In order to achieve the cellwise hard matching, that is, to satisfy the cellwise coupling constraint, we define an approximate cellwise transportation plan $\tilde{T}^{(Y)}$. First, we compute a cellwise transportation plan $T^{(Y)}$ by aggregating the pixelwise transportation plan $T^{(X)}$ (see subsection 4.1), which is in general not hard matching. Second, via thresholding, we obtain hard-matching cellwise transportation plan $\tilde{T}^{(Y)}$ from $T^{(Y)}$ (see subsection 4.2).

A schematic diagram for the process of cell tracking by SCOTT is shown in Figure 3. SCOTT has the following major steps. First, pixels belonging to cells are segmented in each image frame. These pixels are then divided into connected components, which are treated as

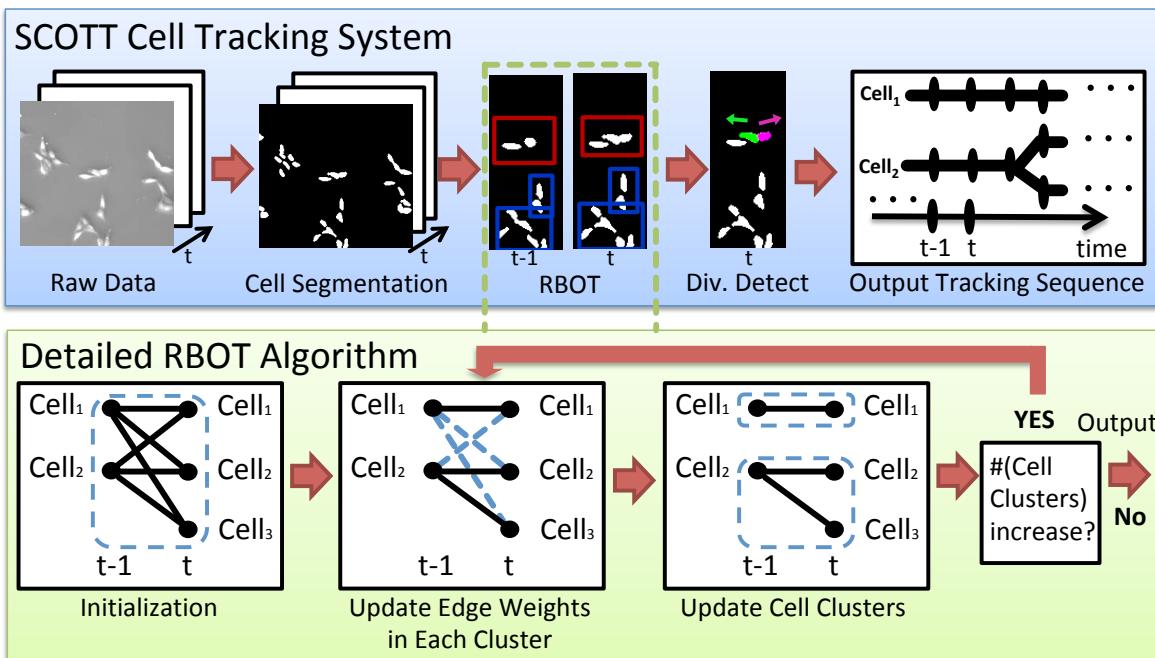


Figure 3. The proposed cell tracking framework based on pixel-level point registration via OT. The top panel illustrates the five steps of cell tracking: cell segmentation, cell alignment by the RBOT algorithm, division detection, refined cell segmentation, and the determination of the entire tracking sequence for each cell. The bottom panel shows the four steps of the RBOT algorithm on two consecutive image frames: initialization, updating edge weights in each cell cluster, updating cell clusters, and computing the total number of cell clusters. The cells and their transportation weights are coded in the bipartite graph with each vertex corresponding to one cell. The RBOT algorithm stops when the number of cell clusters remains unchanged.

segmented cells, but only initially. Precise identification of cells depends on the feedback from later steps. For instance, a connected component may be partitioned into multiple cells. To clarify the terminology, we will call a connected component extracted by initial segmentation a “segment” and a cell identified by SCOTT a “detected cell.” Second, we align cell pixels by OT and apply bipartite graph partitioning (BGP) to find correspondence between groups of clusters. The cascade of OT and BGP is applied recursively until the cell clusters cannot be further divided. Due to intricate scenarios such as cell division or cell merging, some of the cell clusters are subject to another round of OT-based analysis. We refer to this algorithm as *Recursive BGP with OT* (RBOT). Finally, based on the detected divisions and pixel-level point registration, we refine cell segmentation and obtain the tracked sequence of every cell.

4.1. Cell-level transportation. Although we are motivated to solve the pixel-level registration, as discussed in section 1, we must convert pixel-level registration to cell-level alignment so as to ultimately track cells. Ideally, pixels of the same cell in one image frame are mapped to pixels which also belong to one cell. We call this requirement *cellwise consistency*, which is not guaranteed by pixel-level transportation. Frequently, a small portion of one cell is mismatched with part of another cell. Sometimes, the mismatched part comes from a cell located quite far away. This phenomenon seems to be counterintuitive, given that OT attempts to

minimize the total transportation cost. This mismatching is a potential consequence of the marginal constraints of OT forcing every pixel to be matched with some pixel in another frame. Therefore, if the size of a cell varies slightly in two frames, the extra pixels in one cell are matched with distant pixels in other cells because there is no nearby pixel left for matching. As a result, cell alignment cannot be easily achieved by mapping the transportation plan from pixel-level to cell-level. We explain our approach to overcoming this difficulty in the next section. Nevertheless, mapping the transportation plan across the two levels is a basic technical component in our system, which will be described next.

We adopt the notations introduced in subsection 3.1. For image $i = 0, 1$, let the set of all the cell pixels be X_i . Let $n_i = |X_i|$, i.e., the cardinality of set X_i . Suppose there are N_i cells in X_i . Denote cell k , $k = 1, \dots, N_i$, by $Y_{i,k}$, which is the set of pixels contained in cell k . Thus, $X_i = \cup_{k=1}^{N_i} Y_{i,k}$. Let the set of indices $\mathcal{I}_{i,k} = \{j \mid x_{i,j} \in Y_{i,k}\}$. Equivalently, $Y_{i,k} = \{x_{i,j} \mid j \in \mathcal{I}_{i,k}\}$. Denote the pixel-level transportation plan by $T^{(X)} = (t_{j,j'}^{(X)}) \in \mathbb{R}^{n_0 \times n_1}$ and the cell-level transportation plan by $T^{(Y)} = (t_{k,k'}^{(Y)}) \in \mathbb{R}^{N_0 \times N_1}$. The intuition is that the matching weight between two cells is the sum of all the matching weights between pixels contained in the two cells. Specifically, $T^{(Y)}$ is defined by

$$(4.1) \quad t_{k,k'}^{(Y)} \stackrel{\text{def}}{=} \sum_{j \in \mathcal{I}_{0,k}, j' \in \mathcal{I}_{1,k'}} t_{j,j'}^{(X)}.$$

For the convenience of describing the RBOT algorithm, we represent the cell-level alignment by a graph, denoted by G , in which every node corresponds to one cell $Y_{i,k}$, $i = 0, 1$, $k = 1, \dots, N_i$, and the edge weight between $Y_{0,k}$ and $Y_{1,k'}$ is $t_{k,k'}^{(Y)}$. Graph G is bipartite since an edge only exists between cells in different frames.

4.2. Cell alignment by RBOT. To achieve cellwise consistency in the pixel-level transportation plan, our strategy is to gradually shrink the region of pixels involved in OT, guided by the pixel-level transportation itself. Specifically, we start from the transportation of all the cell pixels. Once we obtain the bipartite graph G that encodes the alignment between cells, we perform subgraph extraction by setting edge weights that are below a threshold to zeros. Each subgraph contains a group of cells in one image frame and another group of cells in the other frame. We call the group of cells in each frame a *cell cluster*. The two cell clusters in every subgraph are treated as “aligned,” and the decision is final. In contrast, the cell-level transportation plan $T^{(Y)}$ is only an intermediate result that leads to the extraction of these subgraphs. Since we need the alignment at the granularity level of cells, each pair of matched cell clusters are subject to further alignment for the cells they contain. We thus apply to the pixels contained in each subgraph the same steps of aligning pixels, assigning edge weights in the bipartite graph for cells, and partitioning the graph to obtain even smaller subgraphs. During this recursive process, the pixel-level transportation plan is solved by OT based on WD. However, after the recursive graph partitioning is completed, for certain difficult types of cell clusters, such as those caused by cell division, OT based on WGWD is applied to each cell cluster. Exploiting OT based on both WD and WGWD allows us to balance complexity and accuracy.

We call the above algorithmic design the BGP strategy. This method, inspired by Zha

et al. [53], aims to cluster data represented by a bipartite graph. By this recursive top-down approach, we shrink the region of pixels involved in transportation step by step. This strategy effectively boosts the consistency in the transportation plan. The approach can also avoid the forced matching of cells that are far apart and improve computational efficiency since OT is conducted on smaller and smaller sets of pixels. We now elaborate on the main components in the RBOT algorithm, namely, **Initialization**, **Recursion**, and **Interpretation of graph partitioning results**. After introducing these components, we will analyze the complexity of the algorithm. At the end of this section, a toy example is provided in Figure 4 to illustrate the various aspects of RBOT.

Initialization. We initialize graph G by a bipartite graph where the edge weights are derived from the pixel-level transportation plan with (4.1), preserving only the edges with weights greater than a predetermined threshold. Because the scale to compute the pixel-level transportation plan is very large, we adopt the multiscale OT approximation proposed by Gerber and Maggioni [18] as the initialized pixel-level transportation. The idea of this algorithm is to use a hierarchical approach so that point transportation is restricted within a neighborhood. We denote the transportation plan yield from the multiscale OT approximation method by $T_0^{(X)}$. In addition to fast computation, $T_0^{(X)}$ solved by the multiscale OT restricts the transportation within a neighborhood of a pixel and hence yields a simpler graph G for the cells. Compared with a fully connected bipartite graph G with $O(N_0N_1)$ edges, the graph G constructed from $T_0^{(X)}$ only has $O(N_0 + N_1)$ weight-bearing edges, which is a substantial reduction.

Recursion. In RBOT, cells are clustered recursively by extracting disconnected subgraphs with more edges in G removed via thresholding. Initially, G is the only existing subgraph. Each round of recursion aims at dividing every existing subgraph into even smaller subgraphs. In one round, sweeping through every existing subgraph, the following steps are applied:

- **Step 1:** Compute pixel-level transportation plan $T^{(X)}$ for pixels in the subgraph.
- **Step 2:** Compute edge weights in the graph, i.e., the cell-level transportation plan $T^{(Y)}$, from $T^{(X)}$ by (4.1).
- **Step 3:** Update cellwise transportation plan to $\tilde{T}^{(Y)}$ by removing edges with weights below a predetermined threshold t_0 . Identify the newly formed disconnected subgraphs. Here the element of the matrix is computed by $\tilde{t}_{ij}^{(Y)} = (t_{ij}^{(Y)} > t_0 ? t_{ij}^{(Y)} : 0)$ and t_0 is the minimal proportion of divided cytoplasm.

The recursion terminates when no subgraph can be further partitioned.

One technical issue in OT as applied to the pixels in any subgraph is that the numbers of cell pixels in the two images are in general different. If we assign the same weight to each pixel in every image, then the total weights for the two sets are not equal. In this case, the problem of point set transportation is called the *unbalanced OT* or *Kantorovich distance* (KD) problem [19, 20]. On the contrary, if the total weights are equal, we have *balanced OT*. Note that when the two sets contain different numbers of points, we can construct a balanced OT problem by maintaining the same total weight while allocating different weights to individual points. With this setup, however, we no longer obtain one-to-one mapping between the points. A point in the smaller set is split and transported to multiple points in the other set. We

adopt the method by Gramfort, Peyré, and Cuturi [19] to solve unbalanced OT. The main idea is to add virtual points to the smaller set and define a distance between any actual point and a virtual one.

For example, for a subgraph with n_0 and n_1 pixels in each image frame, assume without loss of generality that $n_0 < n_1$. We add $n_1 - n_0$ virtual points to the source. We need to extend the distance matrix $M \in \mathbb{R}_+^{n_0 \times n_1}$ to $\tilde{M} \in \mathbb{R}_+^{n_1 \times n_1}$. Let $\Delta \in \mathbb{R}_+^{(n_1-n_0) \times n_1}$ be the distance matrix between the virtual points and the actual ones. In order to properly define Δ to satisfy the requirement of KD, we first solve the balanced OT problem for the subgraph and obtain the transportation plan $T^{(X)} = (t_{j,j'}^{(X)})$. Let $\mathcal{I} = \{(j, j') \mid t_{j,j'}^{(X)} > 0\}$. We set $\Delta = \max_{(j,j') \in \mathcal{I}} (m_{j,j'}) \cdot \mathbf{1}_{(n_1-n_0) \times n_1}$ to satisfy the definition of KD (Definition 1 of [19]). Then

$$\tilde{M} = \begin{bmatrix} M \\ \Delta \end{bmatrix}.$$

Let $\tilde{a}_0 = \frac{1}{n_1} \mathbf{1}_{n_1}$. Then we have

$$(4.2) \quad W(\tilde{M}, \tilde{a}_0, a_1) = W\left(\begin{bmatrix} M \\ \Delta \end{bmatrix}, \frac{1}{n_1} \mathbf{1}_{n_1}, \frac{1}{n_1} \mathbf{1}_{n_1}\right).$$

We then apply Sinkhorn iterations to solve transportation plan $\mathcal{T}_{W,\epsilon}(\tilde{M}, \tilde{a}_0, a_1) \in \mathbb{R}_+^{n_1 \times n_1}$. Since the last $n_1 - n_0$ rows specify transportation between virtual points and actual points, we only output the first n_0 rows of $\mathcal{T}_{W,\epsilon}$, which is denoted by $\mathcal{T}_{W,\epsilon}[1 : n_0]$.

Interpretation of graph partitioning results. We categorize the cell clusters according to the number of segmented cells in the source and target images and then code these categories by the format of **#Source-to-#Target**. We describe the meaning of each category for the sake of obtaining the tracking sequence of each cell. The term “multiple” here means “two or more”:

- **One-to-One** can be the result of two cases. The first is the simplest case when a cell moved without touching other cells. The second is the difficult case when a cell has divided but the new cells do not appear separated in the new frame.
- **One-to-Multiple** is the case when one cell divides into multiple distinct cells.
- **Zero-to-One** is the case when a new cell appears in the observation range. All the pixels of this cell are registered to virtual pixels in the source image.
- **One-to-Zero** is the reverse case of **Zero-to-One** when an existing cell moves out of the observation range. All the pixels of this cell are registered to virtual pixels in the target image.
- **Multiple-to-One** is the case when multiple cells are merged into one segment. In this case, we need to align the pixels with the corresponding cells in the source frame.
- **Multiple-to-Multiple** is the case when the cells are difficult to align. When this case occurs, we iteratively remove the edge with the smallest weight and normalize the other edge weights again until the graph can be separated into two subgraphs. Applying this operation recursively, **Multiple-to-Multiple** will eventually be reduced to one of the previous cases.

If the cell cluster is of type **One-to-One** or **Multiple-to-One**, we apply OT based on WGWD to the pixels contained in this cluster. For **One-to-One**, this extra point registration

is needed to detect cell division, which will be elaborated on in the next subsection. For **Multiple-to-One**, OT based on WGWD provides more accurate registration by combining shape and location information. The complete RBOT algorithm is presented formally in Algorithm 4.1.

Algorithm 4.1. Recursive BGP with OT (RBOT).

Input: two sets of cell pixels X_0, X_1 , the cells Y_0, Y_1 .

Output: point-level registration $T^{(X)}$ and cell-level registration $T^{(Y)}$.

- 1: construct a bipartite graph G . Compute distance matrix M and similarity matrix C_0, C_1 .
 - 2: initialize edge weight with multiscale OT approximation $T^{(X)} \leftarrow T_0^{(X)}$ and (4.1).
 - 3: **repeat**
 - 4: add virtual points and extend the boundary condition (M, a_0, a_1) to $(\tilde{M}, \tilde{a}_0, \tilde{a}_1)$.
 - 5: $T^{(X)} \leftarrow \mathcal{T}_{\text{WD},\epsilon}(\tilde{M}, \tilde{a}_0, \tilde{a}_1)[1:n_0]$
 - 6: compute $T^{(Y)}$ with (4.1).
 - 7: compute $\tilde{T}^{(Y)}$ by thresholding $T^{(Y)}$. Remove the corresponding edges in G .
 - 8: partition G into disconnected subgraphs $G = \bigcup_l G_l$.
 - 9: For each subgraph G_l , let $G \leftarrow G_l$ and apply RBOT on each G .
 - 10: **until** G cannot be partitioned.
 - 11: For subgraphs with cell merging and division, $T^{(X)} \leftarrow \mathcal{T}_{\text{WGWD},\epsilon}(C_0, C_1, M, a_0, a_1, \lambda)$.
 - 12: update the matrix $T^{(X)}$ and $T^{(Y)}$ from every subgraph.
 - 13: **return** $T^{(X)}, T^{(Y)}$.
-

Complexity. High computational complexity is a major concern of OT [3, 11, 15, 35]. Assume there are N cells in each frame and c pixels in each cell. Suppose RBOT terminates within a constant number of recursions (five in our experiments). Then the time complexity of the whole process is $O(N^2c^2)$. The computation comprises three parts. First, the multiscale OT approximation has complexity of $O(Nc \log(Nc))$ [18]. Second, Sinkhorn iterations with virtual points have complexity $O(N^2c^2)$ when cells are not fully separate and $O(Nc^2)$ when cells are fully separate [11]. Third, if cell division is detected, the complexity of computing WGWD within cells that are eventually divided is $O(Nc^3)$ [35]. We note that if WGWD is applied to all the pixels in the frame, the complexity is $O(N^3c^3)$, much higher than that of our algorithm.

Example. Figure 4 shows an example of cell alignment by the RBOT algorithm. In each subfigure, the picture on the left shows the segmented cells and the cell-level transportation plan, while to the right of the picture, the bipartite graph for cell alignment is shown. The cell clusters extracted so far are indicated by the text below the picture. We use the blue letters to list the cells in the source, the red letters for cells in the target, and the green lines to show the cell-level transportation plan. In this simple example, we can easily align the cells according to their spatial closeness. We use letters A–H to mark each pair of correctly aligned cell clusters, as shown in the top-left figure. Initially, we compute the pixel-level alignment with multiscale OT approximation and compute the cell-level OT alignment with (4.1). Without imposing cellwise consistency (see subsection 4.1), we obtain long-distance mapping between some cells, e.g., F–G in the upper-left figure. This kind of error in mapping is caused by the marginal

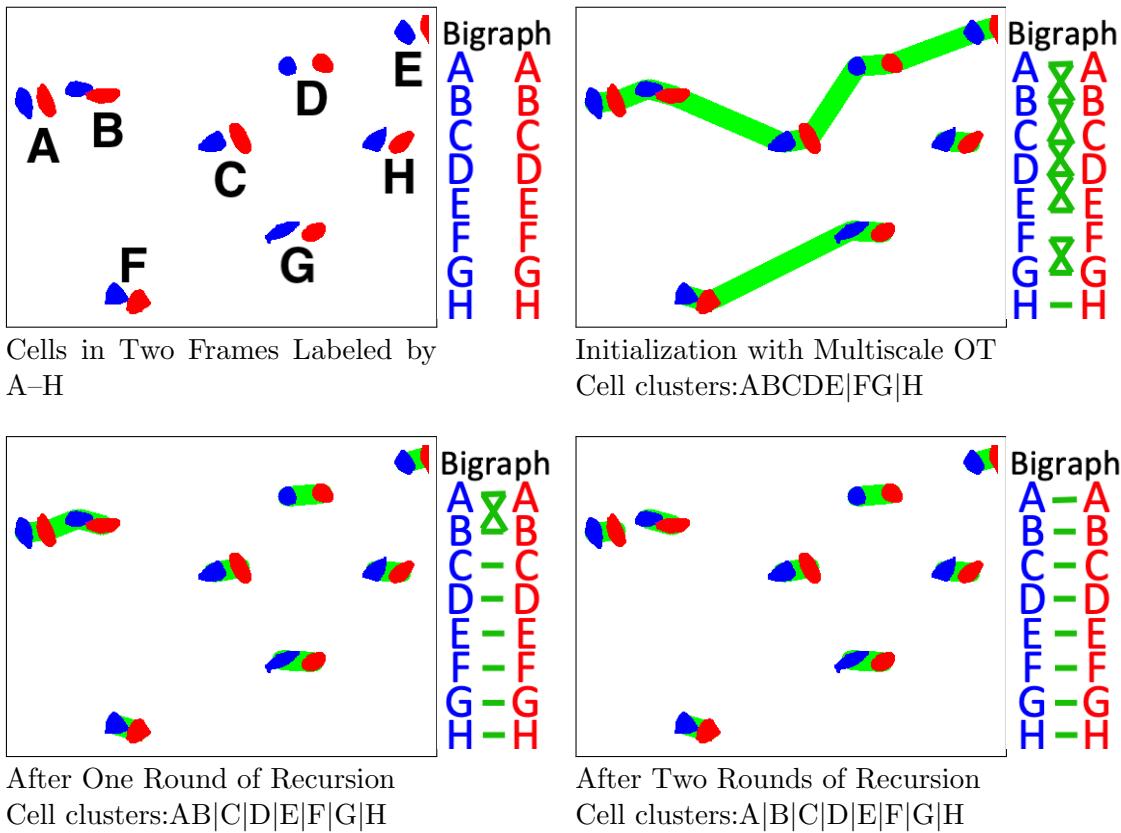


Figure 4. Visualization of cell clusters generated by RBOT. These cell clusters are binary cell masks of two consecutive frames of dataset PSC-Passage7 in [37]. Cells in the same color are from the same frame. Every subplot corresponds to the result obtained up to a certain step. In each subplot, the bipartite graph formed from the cell-level transportation plan is shown on the right, while the cell clusters are described below the figure. Top-left: Two consecutive frames with segmented cells. Top-right: The initialization of the cell-level transportation plan yielded from multiscale OT. Bottom-left: The cell-level transportation plan after one round of recursion in RBOT. Bottom-right: The cell-level transportation plan after two rounds of recursion in RBOT.

constraints of traditional OT, while our RBOT algorithm is designed to address this issue. After one recursion, as shown in the upper-right figure, many edges have been removed based on the pixel-level registration computed by multiscale OT and (4.1). Cells are now clustered into three disconnected subgraphs: the cell pairs A–E, the cell pairs F–G, and the cell pair H, as shown in the top-right figure. In two figures of the bottom, after more recursions have been completed, the cells are grouped into smaller clusters. Finally, as shown in the bottom-right figure, every disconnected cell cluster now contains only a pair of cells from the source and the target and hence cannot be further partitioned. The RBOT algorithm terminates and produces the cell-level transportation plan, as shown in the bottom-right figure. We provide more cell alignment examples computed with the RBOT algorithm, which can be found in section SM2 of the supplementary materials, linked from the main article webpage.

4.3. Cell division detection. In cell biology, cell division usually refers to two processes: mitosis and cytokinesis. At the mitosis stage, the chromosomes of a single cell replicate and separate into two new nuclei. At the cytokinesis stage which follows mitosis, the cytoplasm of a single cell separates into two parts. In some literature [44], cell division is considered to be marked by the end of mitosis or the beginning of cytokinesis, which is also the notion of cell division adopted in the Cell Tracking Challenges where we obtained the data. For comparison with existing work, we also take the beginning of cytokinesis as the biological definition of cell division in this paper.

SCOTT can detect cell division by finding the cell division trend from the pixelwise alignment in WGWD. First, we assume that all the cells are alive and there is no cell fusion phenomenon where multiple cells appear to merge into one component in the image. In addition, SCOTT is only effective under the following conditions: (1) The cell did not have a contraction in shape during division, as often happens with DNA-stained cells. (2) The chromosome and cytoplasm of the two daughter cells move in opposite directions during mitosis. (3) The sizes of the separated cytoplasm of the two daughter cells are almost equal; and the separated cells do not have cell walls.

Detecting cell division from 2D images is challenging because cell division is a continuous process with a time span varying between different types of cells or even individual cells. Sometimes cytokinesis finishes so quickly that we can observe one single cell at time t dividing into two disconnected cells at time $t + 1$. In this case, RBOT can capture the division by recognizing that a cell cluster belongs to the **One-to-Two** category. In another case, cytokinesis may take a long stretch of time, and the cytoplasm of the two daughter cells appears to be a single segment over many frames. It is then difficult to quickly detect the division without adequate prior knowledge about the shape of the cells.

We now explain how to detect cell division by computing WGWD. Based on the pixel-level transportation plan, we propose a novel approach to detect cell division before the spatial separation of the daughter cells is shown in the image. Importantly, our approach is not based on cell-level geometric features such as contours, which often vary with the types of cells and are not necessarily straightforward to quantify. As a result, our approach is general and easy to apply. Our basic idea is to compute the motion of individual pixels based on their transportation plan and then examine the uniformity of the motion vectors. When a cell divides, the pixels are likely to move in significantly different directions. On the contrary, pixels in the same cell normally move in similar directions. As shown in Figure 5, pixels in one cell exhibit a strong pattern of moving in two opposite directions at the moment of division. We found that this pattern of pixel movement is more evident when the transportation is solved by WGWD than that by WD or GWD. This observation reflects that both location and shape information are valuable for establishing the correspondence between pixels.

Let $x_{0,k}$ be any pixel in the source image, and denote its registered pixel in the target by $\pi(x_{0,k})$. Then the motion vector of $x_{0,k}$ is defined by $\pi(x_{0,k}) - x_{0,k}$, which is represented by its polar coordinates (r_k, θ_k) : $r_k e^{i\theta_k} = \pi(x_{0,k}) - x_{0,k}$. Since a cell usually divides into exactly two new cells [49], we are motivated to estimate two main movement directions $\hat{\theta}_0$ and $\hat{\theta}_1$ by fitting a two-component Gaussian mixture model on all the angles θ_k . Denote the density of the i th, $i = 0, 1$, Gaussian component by $\hat{\varphi}_i = \varphi(\hat{\theta}_i, \hat{\sigma}_i^2)$, where $\hat{\theta}_i$ is the mean and $\hat{\sigma}_i^2$ is

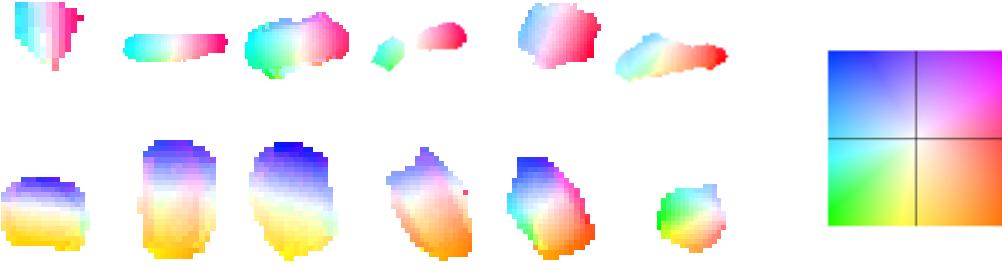


Figure 5. Pixel motion vectors at cell division. To illustrate the motion vectors, their directions are coded by the color hue, and their magnitudes are coded by saturation. A more saturated color means a higher magnitude. The motion vector is the difference between the coordinates of the registered pixels in the target and source images. Top-left: Cells in division with pixels moving in opposite horizontal directions. Bottom-left: Cells in division with pixels moving in opposite vertical directions. The motion vectors are computed based on the pixel-level transportation plan yielded from WGWD at $\lambda = 0.90$. Right: The color palette to code the motion vectors in [4].

the variance. Using the typical mixture model-based clustering, the angles θ_k 's are clustered into two groups. This result determines the grouping of the pixels $x_{0,k}$. Denote the two pixel groups by $Z_0^{(i)}$, $i = 0, 1$.

For pixels $x_{0,k}$ with nonzero r_k (motion vector magnitude), we compute the proportion \hat{q}_i of each pixel group $Z_0^{(i)}$, $i = 0, 1$. We have $\hat{q}_0 + \hat{q}_1 = 1$. There are three conditions to satisfy in order to declare cell division:

1. The difference between the two main movement angles, denoted by $\Delta\theta$, is larger than a threshold. Note that

$$(4.3) \quad \Delta\theta \stackrel{\text{def}}{=} \min \left(\left| \hat{\theta}_0 - \hat{\theta}_1 \right|, 2\pi - \left| \hat{\theta}_0 - \hat{\theta}_1 \right| \right).$$

2. The standard deviation of the two fitted Gaussian components $\hat{\sigma}_i$ is less than a threshold.
3. The proportions \hat{q}_i of $Z_0^{(i)}$, $i = 0, 1$, are both larger than a threshold. $i = 0, 1$ are smaller than a threshold.

Conditions 1 and 2 ensure that there are two sufficiently distinct moving directions relative to the variation within each group. Condition 3 ensures that there are a sufficient number of pixels moving in each direction. As shown by Table 2 (see the column Div.), our method captures cell division more accurately when the two daughter cells have not appeared as separate segments in the image.

5. Experimental results. We have implemented the whole pipeline shown in Figure 3 in Python 3; part of the codes are provided by [17, 18], specifically, the implementation of multiscale OT and the WD/GWD.¹ We conducted the experiments on Amazon Web Service instance EC2-C5.12Xlarge, which has 48 virtual Intel Xeon Scalable Processors CPUs with

¹The code for our experiments are available online from <https://github.com/RachelZheng/scott>.

96GiB memory. In the current implementation, approximately 30% of the computation is parallelized, mainly in the computation of OT. It takes 10–800 seconds for our algorithm to process a single frame, and the exact speed depends on the image size and the average number of pixels in one cell. Our code can be improved in computation efficiency by a more effective implementation, e.g., in C++ or by further parallelization. The processing time can also be reduced substantially by reducing the resolution of the images (that is, to shrink the image size), but this reduced resolution also compromises accuracy. The processing time of each dataset and the effect of reducing image size can be found in section SM3 of the supplementary materials. In our experiments, we set the minimal proportion of cell division $t_0 = 0.1$. In cell division detection, we set the cell division criteria to be $\Delta\theta \geq 0.60$, $\hat{\sigma}_i \leq 0.12$, and $\hat{q}_i \geq 0.30$ for $i = 0, 1$. The thresholds we used for $\Delta\theta$, $\hat{\sigma}_i$, and \hat{q}_i were chosen by grid search. We applied SCOTT using different combinations of threshold values to the first 10 frames in the sequences and evaluated the results using the ground truth. The values yielding the best accuracy are used in later experiments. For new datasets, we recommend searching within these ranges: $\Delta\theta_{\min} \in [30^\circ, 120^\circ]$, $\hat{\sigma}_{\max} \in [0.1, 0.5]$, and $\hat{q}_{\min} \in [0.2, 0.4]$.

5.1. Test sequences. We use seven sequences with annotations from four open-source datasets to test our algorithm. They are the following:

DIC-HeLa consists of two videos of HeLa cells on a flat glass (denote as DIC-HeLa-01 and DIC-HeLa-02). The initial segmentation yields 15 segments on average over 84 image frames of size 512×512 . The dataset has a high cell density and a low signal-to-noise ratio of 0.74 [47].

PhC-U373 consists of two videos of Glioblastoma-astrocytoma U373 cells on a polyacrylamide substrate (denote as PhC-U373-01 and PhC-U373-02). The initial segmentation yields about eight segments on average over 115 image frames of size 520×696 .

PhC-PSC consists of two time series observations of PSCs on a polystyrene substrate (denote as PhC-PSC-01 and PhC-PSC-02). Only the 150th–250th frames are used in the experiment because they have ground-truth labeling. The initial segmentation yields about 300 segments on average over the 101 image frames of size 576×720 . The dataset also has a high cell density and a low signal-to-noise ratio. All three aforementioned datasets are collected from Cell Tracking Challenges [29].

PSC-Passage7 is one image time series of PSC observations collected in [37]. They are proliferating PSC observations in different shapes during a time-lapse experiment with oblique illumination. The dataset contains images taken every 10 minutes with a total of 399 frames of size 1038×1376 , and the human-labeled ground truth is available.

5.2. Baselines. SCOTT is compared with the following two state-of-the-art automatic cell tracking methods.

Validation with tracking (VT) [37]. The code is publicly available online. We only use the tracking function of the codes. We use the default parameters `minPathLength=30` and `minOverlap=0.70` for PhC-PSC and PSC-Passage7, and `minPathLength=10` and `minOverlap=0.50` for DIC-HeLa and PhC-U373 to get the best results.

Conservation tracking (CT) [41]. We ran Ilastik V1.3.2 [43], which implements the method by Schiegg et al. [41]. This method applies the probabilistic graphical model to track the cells, which can effectively handle the oversegmentation and undersegmentation of the cells. Since

the CT algorithm requires the initial segmentation, we manually labeled the cells in the first frame. In addition, to train the cell division detectors, we manually labeled the first five cell division cases in each image time series.

Because segmenting cells from the background is not the focus of this paper, we adopt the cell segmentation methods proposed by Sixta et al. [42] for datasets DIC-HeLa, PhC-U373, and PhC-PSC and Rapoport et al. [37] for dataset PSC-Passage7. Cell segmentation is performed independently on every image frame. Each connected component of the cell pixels is treated as one segmented cell. For multiple cells that overlap in the image, they will be marked as one cell. Starting from this initial cell segmentation, we then apply the SCOTT tracking system.

5.3. Evaluation measures. We use precision, recall, and F-measure (the harmonic mean of precision and recall) to evaluate each algorithm’s ability to detect cell migration, division, and merger. The cell activities labeled in the ground truth are called *correct activities*, and the cell activities output by the tracking algorithm are called *captured activities*. Then the precision and recall are defined as

$$(5.1) \quad \text{precision} \stackrel{\text{def}}{=} \frac{\#(\text{correct activities} \cap \text{captured activities})}{\#(\text{correct activities})},$$

$$(5.2) \quad \text{recall} \stackrel{\text{def}}{=} \frac{\#(\text{correct activities} \cap \text{captured activities})}{\#(\text{captured activities})}.$$

The activities are identified based on every two consecutive image frames on initial cell segments. This evaluation method is also adopted in the literature [41, 46]. Here the term “merger” means multiple cells are in one connected component in the segmentation map, one of the undersegmentation cases discussed by Schiegg et al. [41]. Because VT does not attempt to separate merged cells, the ground-truth labels in dataset PSC-Passage7 do not separate merged cells. We thus cannot evaluate the detection of this type of activity. In [Table 2](#), the measures for the “merger” case are marked N/A. The cell is correctly tracked if a value larger than 0.5 is obtained for the ratio between the intersection and the union of the detected cell and the ground-truth cell.

In addition to precision, recall, and F-measurement, we also evaluate the results by the multiple object tracking accuracy (MOTA) [6] and the tracking accuracy (TRA) based on acyclic oriented graph matching (AOGM) measure [29]. MOTA is used for multiple objects tracking challenges [32]. To compute MOTA, we first use the nearest neighbor rule to map a detected object to a labeled object in the ground truth. The center pixel location of an object is taken as the position of the whole object. MOTA combines the ratios of objects that are undetected, incorrectly detected, and mismatched. TRA is a scaled version of AOGM. The idea of AOGM is to represent the tracking result as a graph, in which every node represents a detected cell and every edge represents tracking of the same cell in two consecutive frames. The misalignment, including undetected cell division, incorrectly aligned cells, and errors in cell segmentation, are assigned with different error weights and penalized in the final accuracy scores. For some datasets like PhC-PSC, whose ground truth is labeled as dots rather than the whole segment, we annotate cell segments with the nearest annotation as the ground truth.

Table 1

Tracking accuracy for cell migration based on every two consecutive frames in datasets DIC-HeLa, PhC-U373, and PSC-Passage7.

Accuracy DIC-HeLa-01	Move			MOTA	TRA
	Prec.	Rec.	F-M.		
VT	0.70	0.11	0.20	0.44	0.34
CT	0.79	0.24	0.37	0.33	0.49
Ours(NoDiv)	0.91	0.48	0.63	0.89	0.90
DIC-HeLa-02					
VT	0.44	0.06	0.11	0.18	0.12
CT	0.68	0.26	0.38	0.32	0.59
Ours(NoDiv)	0.75	0.77	0.76	0.61	0.78
PhC-U373-01					
VT	0.79	0.80	0.79	0.71	0.88
CT	0.80	0.78	0.79	0.86	0.94
Ours(NoDiv)	0.93	0.83	0.88	0.86	0.91
PhC-U373-02					
VT	0.66	0.70	0.68	0.60	0.89
CT	0.77	0.75	0.76	0.80	0.90
Ours(NoDiv)	0.87	0.85	0.86	0.80	0.91
PSC-Passage7					
VT	0.98	0.92	0.95	0.92	0.94
CT	0.97	0.88	0.92	0.97	0.95
Ours(NoDiv)	0.98	0.88	0.93	0.97	0.96

All the aforementioned accuracy measures are within the range [0,1], and a higher value of any measurement indicates better performance.

5.4. Comparison with baselines. Datasets DIC-HeLa and PhC-U373 do not follow our assumptions about cell division and have a negligible number of cell merger cases. In dataset PSC-Passage7, cells that merge and appear as one segment are not labeled as separate cells in the manually generated ground truth. We thus apply SCOTT without division detection to these three datasets. The result is provided in [Table 1](#). For dataset PhC-PSC, we apply SCOTT with cell division detection, and the result is provided in [Table 2](#).

[Table 1](#) shows that SCOTT performs the best for aligning cells that only have the migration activity. Compared with the two baseline methods, for DIC-HeLa, SCOTT achieves higher values for MOTA and TRA, framewise tracking recall, and precision. Cells in DIC-HeLa vary substantially in size and have a high density, while the two baseline methods assume that the cell sizes are similar and cells separate with enough in-between space. As a result, SCOTT outperforms the baseline methods. For PhC-U373 and PSC-Passage7, SCOTT is only slightly better than the two baseline methods.

[Table 2](#) shows that SCOTT performs better in detecting cell division and solving cell merger cases. On the other hand, SCOTT achieves comparable accuracy in cell movement. To be more specific, in terms of the F-measurement for cell division, the accuracy of SCOTT is about 20% higher than VT and about 5% higher than CT. SCOTT significantly increases the recall of cell division and cell merger, which is the reason for the increase of the F-measurement.

In terms of MOTA and TRA, SCOTT achieves higher accuracy than the baseline methods,

Table 2

Tracking accuracy for different cell activities based on every two consecutive frames in dataset PhC-PSC.

Accuracy	Move			Merger			Div.			MOTA	TRA	
	Prec.	Rec.	F-M.	Prec.	Rec.	F-M.	Prec.	Rec.	F-M.			
PhC-PSC-01	VT	0.85	0.95	0.90	N/A	N/A	N/A	0.23	0.49	0.31	0.58	0.62
	CT	0.82	0.90	0.85	0.78	0.89	0.83	0.17	0.15	0.16	0.70	0.78
	Ours($\lambda = 0$)	0.84	0.94	0.89	0.79	0.99	0.88	0.30	0.47	0.37	0.71	0.82
	Ours($\lambda = 0.1$)	0.84	0.95	0.89	0.80	0.99	0.89	0.30	0.46	0.37	0.71	0.83
	Ours($\lambda = 0.3$)	0.84	0.95	0.89	0.81	0.99	0.89	0.31	0.48	0.38	0.72	0.83
	Ours($\lambda = 0.5$)	0.84	0.94	0.89	0.80	0.95	0.87	0.30	0.46	0.36	0.71	0.82
	Ours($\lambda = 1$)	0.83	0.90	0.87	0.78	0.90	0.86	0.26	0.47	0.33	0.66	0.81
	PhC-PSC-02											
PhC-PSC-02	VT	0.87	0.96	0.92	N/A	N/A	N/A	0.18	0.49	0.27	0.69	0.66
	CT	0.86	0.99	0.92	0.65	0.80	0.72	0.12	0.37	0.18	0.82	0.76
	Ours($\lambda = 0$)	0.88	0.93	0.90	0.85	0.99	0.92	0.39	0.46	0.43	0.81	0.83
	Ours($\lambda = 0.1$)	0.89	0.93	0.91	0.85	0.99	0.92	0.39	0.46	0.43	0.82	0.84
	Ours($\lambda = 0.3$)	0.88	0.93	0.90	0.85	0.99	0.92	0.37	0.44	0.40	0.82	0.84
	Ours($\lambda = 0.5$)	0.88	0.92	0.89	0.84	0.97	0.91	0.37	0.44	0.40	0.82	0.84
	Ours($\lambda = 1$)	0.88	0.90	0.89	0.83	0.95	0.90	0.28	0.40	0.33	0.75	0.82

especially for the more complicated dataset PhC-PSC. When parameter λ varies from 0 to 1, MOTA and TRA obtained by SCOTT first increase and then decrease. This observation indicates that WGWD has advantages over WD by taking into account shape information. However, the improvement demonstrated by PhC-PSC is moderate. Specifically, when $0.1 < \lambda < 0.3$, the best MOTA and TRA are obtained, but only slightly better than that at $\lambda = 0$. When $\lambda > 0.5$, SCOTT overdetects cell divisions, resulting in a drop in MOTA and TRA. The overall gain from shape information is limited because the datasets we have experimented with do not contain many cell divisions. Whether we can obtain more impressive results from WGWD in comparison with WD on other datasets is an interesting question for future investigation.

Figure 6 shows an example demonstrating that our SCOTT method works better to detect cell division and merging. Here one random color marks one cell. As we can see, our method separates merged cells well, in a way consistent with both the ground truth and our visual perception. In addition, as shown in the red bounding box at $T = 193$, row 1, one cell divides into two which are still merged. Only our method detects the division right away. In comparison, the two baseline methods fail to recognize the emergence of the two cells until they are completely separated.

Figure 7 shows a few failure cases of SCOTT. The first row contains three original images. Cells that are not accurately identified in the image below are highlighted by a red box. The second row shows the corresponding cell tracking results. For the image in the left column, SCOTT detects two cells for a group of three cells; for the image in the middle column, SCOTT does not separate two cells with the correct boundary; for the image in the right column, part of a cell is incorrectly labeled as a portion of another cell located right above it. It is possible to improve the algorithm by exploiting the characteristics of cell boundaries in more sophisticated ways.

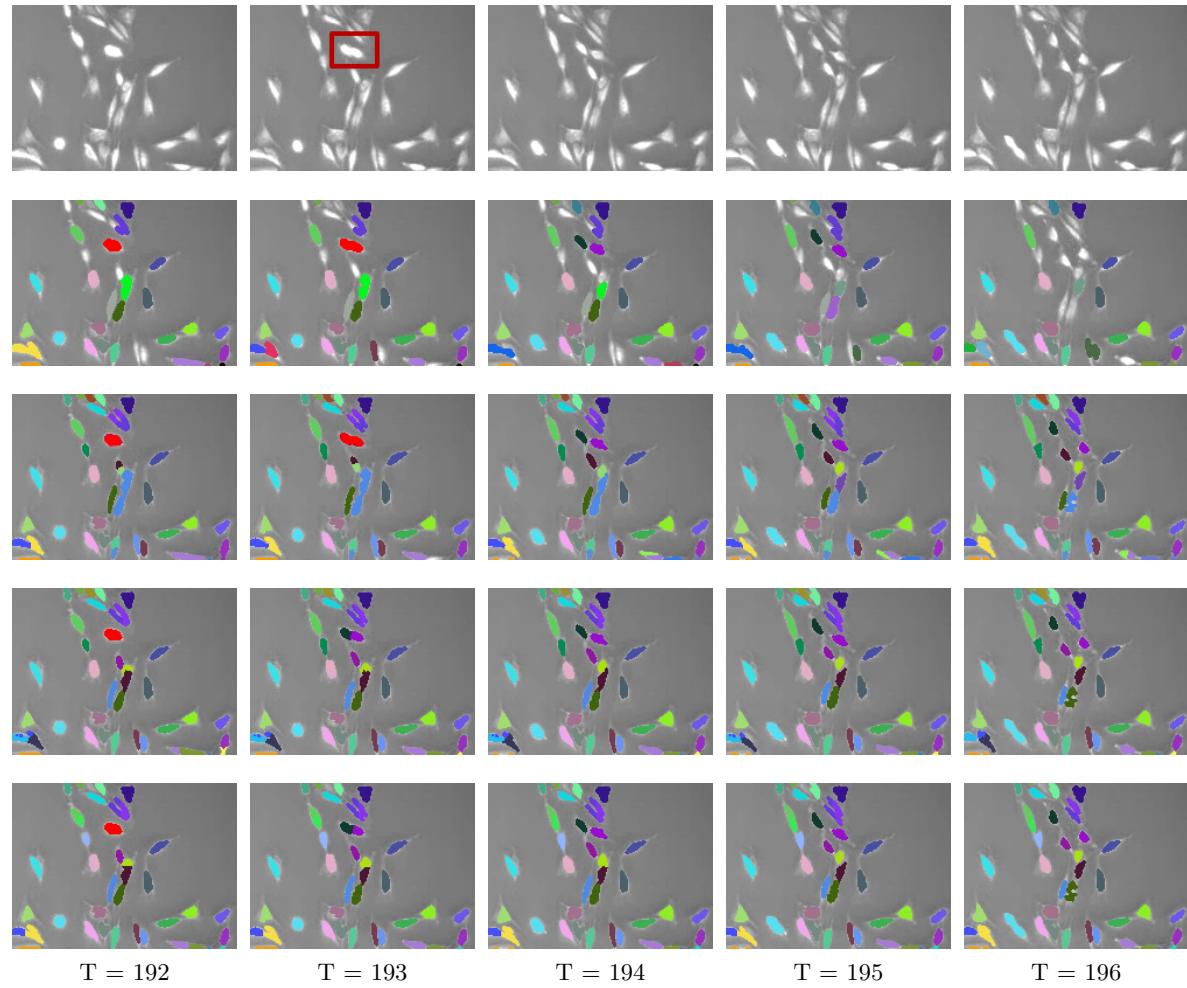


Figure 6. Comparison of tracking results on dataset PhC-PSC-01 at time $T = 192\text{--}196$. From top to bottom. Row 1: Microscopy frames. The red bounding box at $T = 193$ shows a cell division. Row 2: Cell tracking results with VT. Row 3: Cell tracking results with CT. Row 4: Cell tracking results with SCOTT at $\lambda = 0.1$. Row 5: Ground-truth labeling. One random color marks one cell in the time series.

6. Discussions. Our method based on a novel OT framework has demonstrated great potential for cell tracking. Next, we discuss some limitations of the method and future work.

6.1. Simultaneous matching over the whole video. Considering only pairs of consecutive frames [16, 34] and considering the entire video together [10, 28, 41] are both widely adopted approaches in cell tracking literature. Compared with methods that track cells based on pairs of frames, the methods that make use of the whole video simultaneously can achieve higher accuracy at higher computational and memory costs [47] because a larger spatial-temporal context is exploited. However, the latter methods usually require carefully crafted features to characterize a cell. Thus, whole video methods are impractical for general datasets.

In contrast, there are two main limitations if we only consider pairs of image frames. First

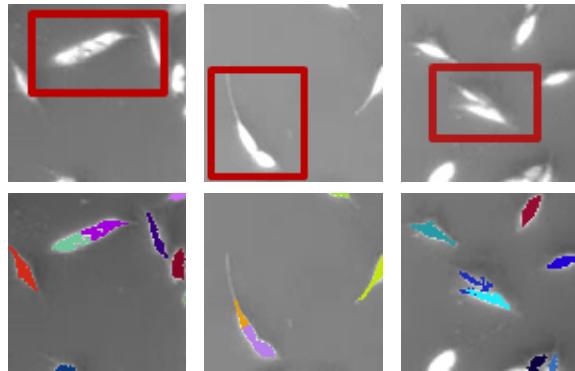


Figure 7. Failure cases of cell tracking by SCOTT. Left: Three cells are detected as two cells. Middle: Two cells are separated with inaccurate boundary. Right: Part of a cell is incorrectly labeled as a portion of another cell nearby.

is the difficulty of observing long-lasting activities. If activities like cell division take a longer time than the span of two frames, it is hard to detect them promptly. Second is the sensitivity to random segmentation errors. In cell segmentation, changes in lighting conditions may cause false or missing segmentation of a cell. Such errors can be easily noticed if we observe the whole video, but they are harder to spot if we inspect only two frames. One remedy to overcome these limitations to some extent is by postprocessing.

We, however, propose a more principled approach that performs pixel mapping by optimization over the entire video sequence. We can adopt the so-called multi-marginal optimal transport problem [33], an extended OT problem. In particular, we want to align the pixels in all the frames simultaneously. Index the image frames by $s \in [0, S]$. Using the same math notations of [33], let the point set at time s be X_s with the weight simplex a_s . The set of couplings with the boundary constraints $\{a_s\}_{s=1}^S$ is given by

$$(6.1) \quad U(\{a_s\}) \stackrel{\text{def}}{=} \left\{ T \in \mathbb{R}_+^{n_1 \times \dots \times n_S} \mid \forall s, \forall i_s, \sum_{l \neq s} \sum_{i_l=1}^{n_l} t_{i_1, \dots, i_S} = (a_s)_{i_s} \right\}.$$

Here $(a_s)_{i_s}$ is the i_s th element of the simplex a_s . If tensor matrix $M \in \mathbb{R}^{n_1 \times \dots \times n_S}$ represents the transportation cost between multiple point sets $\{X_s\}$, the optimization problem can be represented as $\min_{T \in U(\{a_s\})} \langle T, M \rangle$, which is the dot product of tensor T and M .

The solution space of multi-marginal OT is $O(n_1 n_2, \dots, n_S)$, exponential in the number of frames. Currently, the problem is computationally intractable. Our method to compute a series of OT problems between every two frames is an approximation to the global optimum in the whole video.

6.2. Effect of initial segmentation. Tracking results by our system rely heavily on the initial cell segment. To measure the tracking accuracy, AOGM contains terms corresponding directly to the number of cells that are not detected or that of nonexistent cells that are falsely detected. Furthermore, segmentation can affect the calculation of changes in cell mass, a quantity used for detecting cell division. Consequently, we need to start with sufficiently

accurate initial cell segmentation, which we assume is achievable by an existing method from the literature.

To evaluate the quality of segmentation, we use reference sequences (sequences with cells and the tracking of them manually labeled) as the gold standard and compute a level of agreement between the reference and the segmentation. The segmentation method we used can capture on average $\sim 95\%$ of cells in the reference sequences with which we experimented.

6.3. Extension to 3D space. The point alignment method can be easily adapted to three dimensions if we replace 2D coordinates by 3D coordinates. In practice, however, 3D images pose a few additional challenges. First, the increased amount of pixels in 3D images (that is, a stack of 2D images) require more computation for our algorithm. Second, to fully take advantage of the 3D information, the segmentation algorithm will become more complex. Third, as more cell activities are observed in the 3D space, we need to consider movement and division along the direction orthogonal to the 2D image plane.

7. Conclusions and future work. We present a novel cell tracking system called SCOTT, which is based on pixel-level registration via OT. Experiments show that SCOTT performs better in the detection of cell division and cell merging than two other cell tracking algorithms can. In future work, we will extend our approach to 3D+t microscopy images. Applying machine learning to determine “cell division or not” could yield a more accurate classification than manually selecting the thresholds. When more labeled data is available, this can be an interesting direction to explore.

Acknowledgments. We thank the Cell Tracking Challenges for providing the data and Jianyu Mao for helpful discussions.

REFERENCES

- [1] I. ADANJA, O. DEBEIR, V. MÉGALIZZI, R. KISS, N. WARZÉE, AND C. DECAESTECKER, *Automated tracking of unmarked cells migrating in three-dimensional matrices applied to anti-cancer drug screening*, Exp. Cell Res., 316 (2010), pp. 181–193.
- [2] O. AL-KOFAHI, R. J. RADKE, S. K. GODERIE, Q. SHEN, S. TEMPLE, AND B. ROYSAM, *Automated cell lineage construction: A rapid method to analyze clonal development established with murine neural progenitor cells*, Cell Cycle, 5 (2006), pp. 327–335.
- [3] J. ALTSCHULER, J. WEED, AND P. RIGOLLET, *Near-linear time approximation algorithms for optimal transport via sinkhorn iteration*, in Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2017, pp. 1964–1974.
- [4] S. BAKER, D. SCHARSTEIN, J. LEWIS, S. ROTH, M. J. BLACK, AND R. SZELISKI, *A database and evaluation methodology for optical flow*, Int. J. Comput. Vis., 92 (2011), pp. 1–31.
- [5] J.-D. BENAMOU, G. CARLIER, M. CUTURI, L. NENNA, AND G. PEYRÉ, *Iterative Bregman projections for regularized transportation problems*, SIAM J. Sci. Comput., 37 (2015), pp. A1111–A1138, <https://doi.org/10.1137/141000439>.
- [6] K. BERNARDIN AND R. STIEFELHAGEN, *Evaluating multiple object tracking performance: The Clear Mot metrics*, EURASIP J. Image Vide., 2008 (2008), 246309.
- [7] S. BONNEAU, M. DAHAN, AND L. D. COHEN, *Single quantum dot tracking based on perceptual grouping using minimal paths in a spatiotemporal volume*, IEEE Trans. Image Process., 14 (2005), pp. 1384–1395.
- [8] F. BUNYAK, A. HAFIANE, AND K. PALANIAPPAN, *Histopathology tissue segmentation by combining fuzzy clustering with multiphase vector level sets*, in Software Tools and Algorithms for Biological Systems, Springer, New York, 2011, pp. 413–424.

- [9] J. CHEN, C. W. HARVEY, M. S. ALBER, AND D. Z. CHEN, *A matching model based on earth mover's distance for tracking *Myxococcus xanthus**, in Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, Berlin, Heidelberg, 2014, pp. 113–120.
- [10] S. CORALUPPI AND C. CARTHEL, *Modified scoring in multiple-hypothesis tracking*, *J. Adv. Inf. Fusion*, 7 (2012), pp. 153–164.
- [11] M. CUTURI, *Sinkhorn distances: Lightspeed computation of optimal transport*, in Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2013, pp. 2292–2300.
- [12] O. DEBEIR, P. VAN HAM, R. KISS, AND C. DECAESTECKER, *Tracking of migrating cells under phase-contrast video microscopy with combined mean-shift processes*, *IEEE Trans. Med. Imaging*, 24 (2005), pp. 697–711.
- [13] R. L. DOBRUSHIN, *Prescribing a system of random variables by conditional distributions*, *Theory Probab. Appl.*, 15 (1970), pp. 458–486, <https://doi.org/10.1137/1115049>.
- [14] A. DUFOUR, R. THIBEAUX, E. LABRUYERE, N. GUILLEN, AND J.-C. OLIVO-MARIN, *3-D active meshes: Fast discrete deformable models for cell tracking in 3-D time-lapse microscopy*, *IEEE Trans. Image Process.*, 20 (2011), pp. 1925–1937.
- [15] P. DVURECHENSKY, A. GASNIKOV, AND A. KROSHNIN, *Computational optimal transport: Complexity by accelerated gradient descent is better than by Sinkhorn's algorithm*, in Proceedings of the 35th International Conference on Machine Learning, 2018.
- [16] O. DZYUBACHYK, W. A. VAN CAPPELLEN, J. ESSERS, W. J. NIJSEN, AND E. MEIJERING, *Advanced level-set-based cell tracking in time-lapse fluorescence microscopy*, *IEEE Trans. Med. Imaging*, 29 (2010), pp. 852–867.
- [17] R. FLAMARY AND N. COURTY, *POT Python Optimal Transport Library*, <https://github.com/rflamary/POT> (accessed 2019/04/01).
- [18] S. GERBER AND M. MAGGIONI, *Multiscale strategies for computing optimal transport*, *J. Mach. Learn. Res.*, 18 (2017), pp. 2440–2471.
- [19] A. GRAMFORT, G. PEYRÉ, AND M. CUTURI, *Fast optimal transport averaging of neuroimaging data*, in Proceedings of the International Conference on Information Processing in Medical Imaging, Springer, Cham, 2015, pp. 261–272.
- [20] L. G. HANIN, *Kantorovich-Rubinstein norm and its application in the theory of Lipschitz spaces*, *Proc. Amer. Math. Soc.*, 115 (1992), pp. 345–352.
- [21] B. X. KAUSLER, M. SCHIEGG, B. ANDRES, M. LINDNER, U. KOETHE, H. LEITTE, J. WITTBRODT, L. HUFNAGEL, AND F. A. HAMPRECHT, *A discrete chain graph model for 3d+ t cell tracking with high misdetection robustness*, in Proceedings of the European Conference on Computer Vision, Springer, Berlin, Heidelberg, 2012, pp. 144–157.
- [22] T. KIRUBARAJAN, Y. BAR-SHALOM, AND K. R. PATTIPATI, *Multiassignment for tracking a large number of overlapping objects [and application to fibroblast cells]*, *IEEE Trans. Aerosp. Electron. Syst.*, 37 (2001), pp. 2–21.
- [23] B. C. KO, J.-W. GIM, AND J.-Y. NAM, *Automatic white blood cell segmentation using stepwise merging rules and gradient vector flow snake*, *Micron*, 42 (2011), pp. 695–705.
- [24] F. LI, X. ZHOU, J. MA, AND S. T. WONG, *Multiple nuclei tracking using integer programming for quantitative cancer cell cycle analysis*, *IEEE Trans. Med. Imaging*, 29 (2010), pp. 96–105.
- [25] K. LI, E. D. MILLER, M. CHEN, T. KANADE, L. E. WEISS, AND P. G. CAMPBELL, *Computer vision tracking of stemness*, in Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2008, pp. 847–850.
- [26] X. LOU AND F. A. HAMPRECHT, *Structured learning for cell tracking*, in Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2011, pp. 1296–1304.
- [27] K. E. MAGNUSSON AND J. JALDÉN, *A batch algorithm using iterative application of the Viterbi algorithm to track cells and construct cell lineages*, in Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI), 2012, pp. 382–385.
- [28] K. E. MAGNUSSON, J. JALDÉN, P. M. GILBERT, AND H. M. BLAU, *Global linking of cell tracks using the Viterbi algorithm*, *IEEE Trans. Med. Imaging*, 34 (2014), pp. 911–929.
- [29] M. MAŠKA, V. ULMAN, D. SVOBODA, P. MATULA, P. MATULA, C. EDERRA, A. URBIOLA, T. ESPAÑA, S. VENKATESAN, D. M. BALAK, P. KARAS, T. BOLCKOVÁ, M. ŠTREITOVA, C. CARTHEL, S.

- CORALUPPI, N. HARDER, K. ROHR, K. E. G. MAGNUSSON, J. JALDÉN, H. M. BLAU, O. DZYUBACHYK, P. KŘÍŽEK, G. M. HAGEN, D. PASTOR-ESCUREDO, D. JIMENEZ-CARRETERO, M. J. LEDESMA-CARBAYO, A. MUÑOZ-BARRUTIA, E. MEIJERING, M. KOZUBEK, AND C. ORTIZ-DE-SOLORZANO, *A benchmark for comparison of cell tracking algorithms*, Bioinformatics, 30 (2014), pp. 1609–1617.
- [30] E. MEIJERING, O. DZYUBACHYK, AND I. SMAL, *Methods for Cell and Particle Tracking*, Method Enzymol. 504, Elsevier, Amsterdam, 2012, pp. 183–200.
- [31] F. MÉMOLI, *Gromov-Wasserstein distances and the metric approach to object matching*, Found. Comput. Math., 11 (2011), pp. 417–487.
- [32] A. MILAN, L. LEAL-TAIXÉ, I. REID, S. ROTH, AND K. SCHINDLER, *Mot16: A Benchmark for Multi-object Tracking*, preprint, <https://arxiv.org/abs/1603.00831>, 2016.
- [33] B. PASS, *Multi-marginal optimal transport: Theory and applications*, ESAIM Math. Model. Numer. Anal., 49 (2015), pp. 1771–1790.
- [34] D. PASTOR-ESCUREDO, M. A. LUENGO-OROZ, L. DULOQUIN, B. LOMBARDOT, M. LEDESMA-CARBAYO, P. BOURGINE, N. PEYRIERAS, AND A. SANTOS, *Spatio-temporal filtering with morphological operators for robust cell migration estimation in “in-vivo” images*, in Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI), 2012, pp. 1312–1315.
- [35] G. PEYRÉ, M. CUTURI, AND J. SOLOMON, *Gromov-Wasserstein averaging of kernel and distance matrices*, in Proceedings of the International Conference on Machine Learning, 2016, pp. 2664–2672.
- [36] U. PFERSCHY AND J. SCHAUER, *The maximum flow problem with disjunctive constraints*, J. Comb. Optim., 26 (2013), pp. 109–119.
- [37] D. H. RAPOPORT, T. BECKER, A. M. MAMLOUK, S. SCHICKTANZ, AND C. KRUSE, *A novel validation algorithm allows for automated cell tracking and the extraction of biologically meaningful parameters*, PloS One, 6 (2011), e27315.
- [38] Y. RUBNER, C. TOMASI, AND L. J. GUIBAS, *The earth mover’s distance as a metric for image retrieval*, Int. J. Comput. Vis., 40 (2000), pp. 99–121.
- [39] L.-P. SAUMIER, B. KHOUIDER, AND M. AGUEH, *Optimal transport for particle image velocimetry: Real data and postprocessing algorithms*, SIAM J. Appl. Math., 75 (2015), pp. 2495–2514, <https://doi.org/10.1137/140988814>.
- [40] M. SCHIEGG, P. HANSLOVSKY, C. HAUBOLD, U. KOETHE, L. HUFNAGEL, AND F. A. HAMPRECHT, *Graphical model for joint segmentation and tracking of multiple dividing cells*, Bioinformatics, 31 (2014), pp. 948–956.
- [41] M. SCHIEGG, P. HANSLOVSKY, B. X. KAUSLER, L. HUFNAGEL, AND F. A. HAMPRECHT, *Conservation tracking*, in Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 2928–2935.
- [42] T. SIXTA, J. CAO, J. SEEBAKH, H. SCHNITTNER, AND B. FLACH, *Coupling Cell Detection and Tracking by Temporal Feedback Software*, <http://celltrackingchallenge.net/participants/CVUT-CZ/> (accessed 2019/04/01).
- [43] C. SOMMER, C. N. STRAEHLE, U. KOETHE, AND F. A. HAMPRECHT, *Ilastik: Interactive learning and segmentation toolkit*, in Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2011, pp. 230–233.
- [44] T. TAKAKI, K. TRENZ, V. COSTANZO, AND M. PETRONCZKI, *Polo-like kinase 1 reaches beyond mitosis—cytokinesis, DNA damage response, and development*, Curr. Opin. Cell Biol., 20 (2008), pp. 650–660.
- [45] M. THERY AND M. BORNENS, *Cell shape and cell division*, Curr. Opin. Cell Biol., 18 (2006), pp. 648–657.
- [46] E. TÜRETKEN, X. WANG, C. J. BECKER, C. HAUBOLD, AND P. FUJ, *Network flow integer programming to track elliptical cells in time-lapse sequences*, IEEE Trans. Med. Imaging, 36 (2017), pp. 942–951.
- [47] V. ULMAN, M. MAŠKA, K. E. MAGNUSSON, O. RONNEBERGER, C. HAUBOLD, N. HARDER, P. MATULA, P. MATULA, D. SVOBODA, M. RADOJEVIC, I. SMAL, K. ROHR, J. JALDÉN, H. M. BLAU, O. DZYUBACHYK, B. LELIEVELDT, P. XIAO, Y. LI, S. Y. CHO, A. C. DUFOUR, J. C. OLIVO-MARIN, C. C. REYES-ALDASORO, J. A. SOLIS-LEMUS, R. BENNSCH, T. BROX, J. STEGMAIER, R. MIKUT, S. WOLF, F. A. HAMPRECHT, T. ESTEVES, P. QUELHAS, Ö. DEMIREL, L. MALMSTRÖM, F. JUG, P. TOMANCAK, E. MEIJERING, A. MUÑOZ-BARRUTIA, M. KOZUBEK, AND C. ORTIZ-DE-SOLORZANO, *An objective comparison of cell-tracking algorithms*, Nat. Methods, 14 (2017), pp. 1141–1152.
- [48] C. VILLANI, *Topics in Optimal Transportation*, Grad. Stud. Math. 58, American Mathematical Society,

- Providence, RI, 2003.
- [49] C. M. WATERMAN-STORER, A. DESAI, J. C. BULINSKI, AND E. SALMON, *Fluorescent speckle microscopy, a method to visualize the dynamics of protein assemblies in living cells*, Curr. Biol., 8 (1998), pp. 1227–1230.
- [50] M. WERMAN, S. PELEG, AND A. ROSENFELD, *A distance metric for multidimensional histograms*, Comput. Vis. Graph. Image Process., 32 (1985), pp. 328–336.
- [51] S. WIENERT, D. HEIM, K. SAEGER, A. STENZINGER, M. BEIL, P. HUFNAGL, M. DIETEL, C. DENKERT, AND F. KLAUSCHEN, *Detection and segmentation of cell nuclei in virtual microscopy images: A minimum-model approach*, Sci. Rep., 2 (2012), 503.
- [52] F. YANG, M. A. MACKEY, F. IANZINI, G. GALLARDO, AND M. SONKA, *Cell segmentation, tracking, and mitosis detection using temporal context*, in Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, Berlin, Heidelberg, 2005, pp. 302–309.
- [53] H. ZHA, X. HE, C. DING, H. SIMON, AND M. GU, *Bipartite graph partitioning and data clustering*, in Proceedings of the Tenth ACM International Conference on Information and Knowledge Management, 2001, pp. 25–32.
- [54] L. ZHANG, Y. LI, AND R. NEVATIA, *Global data association for multi-object tracking using network flows*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
- [55] C. ZIMMER, E. LABRUYERE, V. MEAS-YEDID, N. GUILLÉN, AND J.-C. OLIVO-MARIN, *Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: A tool for cell-based drug testing*, IEEE Trans. Med. Imaging, 21 (2002), pp. 1212–1221.