



Contents lists available at ScienceDirect

Medical Image Analysis

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

Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis

Dirk Padfield^{a,b,*}, Jens Rittscher^a, Badrinath Roysam^b^a GE Global Research, One Research Circle, Niskayuna, NY 12309, United States^b Rensselaer Polytechnic Institute, 110 8th St., Troy, NY 12180, United States

ARTICLE INFO

Article history:

Available online 13 August 2010

Keywords:

Cell analysis
Tracking
Segmentation
Minimum-cost flow
Wavelets

ABSTRACT

A growing number of screening applications require the automated monitoring of cell populations in a high-throughput, high-content environment. These applications depend on accurate cell tracking of individual cells that display various behaviors including mitosis, merging, rapid movement, and entering and leaving the field of view. Many approaches to cell tracking have been developed in the past, but most are quite complex, require extensive post-processing, and are parameter intensive. To overcome such issues, we present a general, consistent, and extensible tracking approach that explicitly models cell behaviors in a graph-theoretic framework. We introduce a way of extending the standard minimum-cost flow algorithm to account for mitosis and merging events through a coupling operation on particular edges. We then show how the resulting graph can be efficiently solved using algorithms such as linear programming to choose the edges of the graph that observe the constraints while leading to the lowest overall cost. This tracking algorithm relies on accurate denoising and segmentation steps for which we use a wavelet-based approach that is able to accurately segment cells even in images with very low contrast-to-noise. In addition, the framework is able to measure and correct for microscope defocusing and stage shift. We applied the algorithms on nearly 6000 images of 400,000 cells representing 32,000 tracks taken from five separate datasets, each composed of multiple wells. Our algorithm was able to segment and track cells and detect different cell behaviors with an accuracy of over 99%. This overall framework enables accurate quantitative analysis of cell events and provides a valuable tool for high-throughput biological studies.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

High-throughput automated screening of *in vitro* systems is of great importance to biological research. For example, the measurement of cell motility and mitosis are useful for fluorescence microscopy applications in areas such as cancer research, immunology, and developmental biology. To enable such studies, researchers use automated microscopes to run large numbers of experiments in a high-throughput and high-content fashion over extended periods of time. For example, to study the effect of various cell treatments, biologists often use 96-well plates, where each well represents a different experiment and contains several hundred cells that are imaged over several hours or days resulting in the image acquisition of several hundred thousand cells. While such experiments are not focused on high-resolution imaging of internal structures, all the cells need to be tracked accurately over time to determine the effect of the cell treatments. Quantitative time-lapse analysis on the single-cell level provided by segmentation and

tracking algorithms can offer detailed insight into such experiments.

Although each biological application has specific requirements, they share several core challenges. Each dataset typically contains a large number of cells to be tracked, and the cells have similar appearance, making them difficult to distinguish. Events such as mitosis (cell division), cell death, occlusion, growth, sudden rapid movement, and cells moving into and out of the field of view need to be accurately captured. Limiting the potential for cytotoxicity induced by fluorescent labels is another important factor that influences the design of imaging protocols, and low concentrations of fluorescent dyes result in low signal-to-noise making the segmentation of the image data challenging. Also, in order to reduce photobleaching, the experiments are only imaged occasionally; while some experiments may be imaged every 3 or 5 min, others that last for several days may only be imaged every 30 min. Consequently, there is often no spatial overlap of the cells between adjacent frames. Further challenges include errors in the movement of the automated stage that need to be corrected for accurate tracking, and it is also necessary to remove defocused images resulting from the failure of the autofocus step during high-throughput screens.

The goal of this work is the design of a tracking and segmentation framework that addresses these challenges. The problem of

* Corresponding author at: GE Global Research, One Research Circle, Niskayuna, NY 12309, United States. Tel.: +1 518 387 4149; fax: +1 518 387 6981.

E-mail addresses: padfield@research.ge.com (D. Padfield), rittsc@crd.ge.com (J. Rittscher), roysam@ecse.rpi.edu (B. Roysam).

cell tracking has been extensively studied in the past, and many effective approaches have been developed recently (see Section 2). These approaches generally adapt some well established algorithmic framework, such as contour evolution or stochastic filtering, to the cell tracking problem and include extensions or post-processing steps to fit those frameworks to the challenges specific to cell tracking. These extensions and combinations of approaches often become complex, computationally intensive, and require many parameters to fine tune. Furthermore, many are built on assumptions and specific models of the data that can limit their generalizability; for example, approaches that model the duration of the cell cycle tend to fail on pathological datasets where the cells exhibit altered cell cycle patterns. All of these considerations indicate the need for a consistent cell tracking framework.

In this paper, we develop a mathematical framework that formulates the cell tracking problem using a single model that is consistent, extensible, general, efficient, and requires few parameters. It is consistent in that it models many cell behaviors such as cell splitting, merging, entering, leaving, and moving in one framework. It is extensible since new behavior models can be added using appropriate weighting functions. It is general in that it handles control and pathological cases implicitly without the need for parameter adjustment. It also does not depend on the imaging modality or image dimension (for example, 2D or 3D) since it is based on the comparison of features of segmented objects. It efficiently processes each image on the order of seconds, and it requires only one parameter that we set empirically and keep constant throughout all experiments.

Although the main focus of this paper is on our new tracking algorithm, this algorithm depends on several other algorithms we introduce including wavelet-based cell segmentation, stage shift measurement, and defocused image detection. All of these algorithms together make up our automated segmentation and tracking framework.

This paper is organized as follows. A survey of previous approaches to cell tracking and segmentation is given in Section 2. The details of our tracking and segmentation framework are given in Sections 3 and 4. In Section 5, experimental results on five different datasets with multiple wells and different imaging conditions representing nearly 400,000 cells and nearly 32,000 tracks are presented. Our conclusions and future work are discussed in Section 6.

2. State of the art

Two core components of effective time-lapse cellular analysis are tracking and segmentation. These components enable such measurements as cell motility, migration, proliferation, arrest,

death, deformation, and population dynamics, among others. In this section, we present the state of the art of tracking and segmentation algorithms as applied to biological data.

2.1. Tracking

Extensive research in the field of computer vision has resulted in powerful and versatile algorithms for visual tracking. We found that the existing methods can roughly be categorized into three groups: model-based contour evolution, stochastic filtering, and independent segmentation of individual frames followed by data association. This is a good way to organize the various approaches, although it is possible that methods exist that fall outside of these categories. Each of these general categories has certain advantages and disadvantages with regard to the challenges of cell tracking, and we list our understanding of these in Table 1. In this table, we also list example references that utilize these categories for the problem of live cell tracking. These references employ additional processing to address the limitations of the various categories.

2.1.1. Contour evolution

In the contour evolution approach, the contour from a previous image is propagated to the next image to serve as the segmentation initialization. Using this initialization, the contour is evolved to segment the object in the next image. Contour evolution approaches, and in particular level sets (Osher and Sethian, 1988; Sethian, 1996; Osher and Fedkiw, 2002), offer some advantages as they can easily handle changes in topology. Yang et al. (2005) exploit this property by evolving level sets in a spatio-temporal volume. However, the algorithm has difficulty with closely packed cells, and the authors admit that the algorithm only works on healthy cells since it expects a standard model for mitosis events. Padfield et al. (2009, 2008, 2006) also make use of the topological flexibility property by evolving level sets on the spatio-temporal volume; this is followed by linking with shortest paths using the fast marching algorithm to associate detections over time. In (Dzyubachyk et al., 2007), Dzyubachyk et al. present a fusion of the standard variational method described in (Rousson and Deriche, 2003) with the Chan and Vese model (Chan and Vese, 2001) for segmenting and tracking cells in microscopy images using level sets. The contours are evolved from frame to frame, and the cost functions are designed so that the level sets will latch onto the cell boundaries in the next frame. In (Bunyak et al., 2006), Bunyak et al. use level set segmentation to segment cells on each frame, and the tracking relies on a detection-based multiple-object tracking method with delayed decision enabled by multi-hypothesis

Table 1

Comparison of different categories of approaches to cell tracking. This table summarizes how well three main visual tracking categories address challenges particular to cell tracking. The “+” indicates that the method is effective at handling a particular challenge, and “–” indicates the method has difficulties. Level sets (Osher and Sethian, 1988) are an example of a common approach to contour evolution. Kalman filters (Kalman, 1960), particle filters (Blake and Isard, 1998), and mean shift tracking (Comaniciu et al., 2000) are typical examples of stochastic filters. The Hungarian algorithm (Kuhn, 1955) is a well known example of data association. Several example references of the application of these approaches to cell tracking are also given. These references employ additional processing to address the limitations of the tracking approaches.

Challenge	Contour evolution	Stochastic filtering	Segment & associate
Mitosis and merging	+	–	–
Appearing and disappearing	–	–	–
Low temporal resolution	–	+	+
Fast motion	–	+	+
Tracking drift	–	–	+
Accurate segmentation	+	–	+
Selected references	Yang et al. (2005) Padfield et al. (2009) Dufour et al. (2005) Dzyubachyk et al. (2007)	Debeir et al. (2005) Li et al. (2008) Genovesio et al. (2006) Kachouie et al. (2006)	Al-Kofahi et al. (2006) Padfield et al. (2009) Zhang et al. (2008) Sbalzarini and Koumoutsakos (2005)

testing. This framework is extended in (Nath et al., 2006) by using a vertex coloring algorithm to constrain the number of level sets needed to keep cell regions from merging. A 3D model-propagation approach is presented by Dufour et al. in (Dufour et al., 2005). They use multiple level set functions, one for each cell, and introduce an overlap penalty to inhibit merging of neighboring active surfaces. Since one level set is used for each cell, this can be computationally intensive, and the authors also mention that the method is ill-adapted to tight cell clusters because it relies on the existence of an image background. While all these contour evolution approaches have been successfully applied to many problems, one underlying requirement restricts their use: the objects generally need to be overlapping from frame to frame. If the image acquisition frame rate is low or the cells move rapidly, approaches that depend on such overlap will fail unless heuristic rules are included for re-initialization and post-processing. Additionally, sudden changes in topology caused by the appearance and disappearance of cells require the re-initialization of the level sets and additional heuristics.

2.1.2. Stochastic filters

Stochastic filters (Gelb, 1979; Blake and Isard, 1998) can be extremely powerful if a object motion can be modeled, and, in general, probabilistic approaches rely on strong model assumptions. The necessary model parameters are typically learned from training datasets. Examples of this category include mean-shift (Comaniciu et al., 2000), Kalman filters (Kalman, 1960; Simon, 2006; Welch and Bishop, 2004; Sorenson, 1970; Bar-Shalom and Fortmann, 1988), particle filters (Blake and Isard, 1998; van de Merwe et al., 2000; Arulampalam et al., 2002), and their variants. Mean-shift tracking can, for example, be used to track objects using a basic appearance model. The Kalman filter provides a recursive solution to estimating the state of a process by minimizing the mean of the squared error. Particle filters provide a means for estimating the state of highly non-linear systems, but require significant amounts of computation. In (Reid, 1979), Reid defines multiple hypothesis tracking using prediction and assignment, and tracking multiple targets is described in (MacCormick and Blake, 1999; Okuma et al., 2004; Rui and Chen, 2001; Vermaak et al., 2003; Isard and MacCormick, 2001). Debeir et al. in (Debeir et al., 2005) track cell centroids without segmentation by updating the centroid locations using the mean-shift algorithm; this algorithm depends heavily on the accuracy of the centroid localization since the tracking results degrade as the centroids start to drift. Multiple Kalman filters are used for cell tracking by Li et al. in (Li et al., 2008, 2006, 2007, 2008). They segment each frame independently and use model propagation with level sets and a stochastic motion filter to track large numbers of cells. Genovesio et al. (2006) detect cells by combining wavelet coefficients and use a particle filter to associate cells over time. In (Kachouie et al., 2006), Kachouie et al. introduce a maximum *a posteriori* based probabilistic framework for tracking cells across all images at once. Because of the large number of required hypotheses, the number of cells that the system can track is limited; the results are demonstrated on images with less than 10 cells. Despite the firm theoretical foundation of this category of algorithms, the fact that cells undergo mitosis and enter and leave the image frame make the application of stochastic filters less effective.

2.1.3. Segmentation and association

Another effective approach to tracking is first segmenting the objects on each frame and then associating the segmented objects across adjacent frames. For cell analysis, Al-Kofahi et al. (2006), use linear programming on various matching hypotheses generated from the features of segmented cells. Their approach models mitosis events but does not capture cell occlusion and cells entering

and leaving the field-of-view. This approach is also used by Chen et al. in (Chen et al., 2009, 2008) for tracking cells in 5D (x, y, z, t, color). In (De Hauwer et al., 1999), Dehauwer et al. use markers derived from the previous frame to segment cells in the next frame, and cells are matched across frames using the Euclidean distance. Padfield et al. in (Padfield et al., 2008) associate cells in a spatio-temporal volume by training features from tracks of single cells and using them to split tracks consisting of multiple-cells. In (Padfield et al., 2009), Padfield et al. first introduced the concept of “coupled minimum-cost flow”, which enables the implicit modeling of different cell behaviors including moving, splitting, merging, moving into the image, and moving out of the image in one consistent framework. Bonneau et al. (2005) process the 2D+t data as a spatio-temporal volume in which perceptual grouping and minimal paths are used to connect tracks of detected objects. Xie et al. (2008) track cells that change in appearance over time by using a matching function based on appearance and motion to associate cells detected on each image. Tracking of a large numbers of segmented cells is accomplished by Wong et al. in (Chen et al., 2006,) by matching cells using a greedy approach. Zhang et al. (2008) use a minimum-cost flow network for globally assigning a small number of detections over time in which the optimization is run multiple times as different hypotheses are tested; this is computationally expensive, and the large number of hypotheses limits its application to problems with a small number of tracks. Sbalzarini and Koumoutsakos (2005) solve the matching problem using a particle matching algorithm that minimizes a cost functional that also includes “dummy particles” for objects entering and leaving. These algorithms indicate that standard matching techniques have difficulty associating fast moving cells and that mitosis and occlusion events need to be handled carefully.

2.2. Segmentation

Many of the cell tracking approaches described above require some simple cell detection step rather than accurate cell segmentations. The segmentation itself, however, can yield information about the change in cell size, shape, and morphology that can lead to important biological insights such as the type of cell death induced by a particular treatment. A large array of cell segmentation algorithms has been developed over time. In particular, many approaches have been developed for segmentation of nuclei-stained cells in fluorescence images (Adiga and Chaudhuri, 2001; Lin et al., 2004; Chen et al., 2006; Wahlby et al., 2004; He et al., 2006; Harder et al., 2006). A common approach is to first binarize the image using some thresholding technique and then use the watershed algorithm on either the intensities, gradients, shapes as derived from the binary mask, or other measure or combination of measures. Usually, some merging and splitting is performed to refine the segmentations (Ancin et al., 1995; Adiga and Chaudhuri, 2001; Lin et al., 2003, 2004; Chen and Wong, 2005; Chen et al., 2006). In (Wahlby et al., 2004), Wahlby et al. use marker-controlled watershed for segmenting nuclei, where the seeds are defined using the h-maxima operation on the smoothed intensity image. The watershed algorithm is then run on the gradient image, and cells are merged by considering the strength of edges. In (He et al., 2006), He et al. describe a method for foreground/background segmentation in cell images using a manual training step and a classifier along with boundary refinement; this approach enables the segmentation of foreground from background but not individual cells from each other. In (Harder et al., 2006; Harder et al., 2006; Kovalev et al., 2006) Harder et al. classify cells into four phases of the cell life cycle using adaptive thresholding, feature extraction, and feature classification.

Level set methods (Osher and Sethian, 1988; Sethian, 1996; Osher and Fedkiw, 2002) are another effective method for

segmentation. In (de Solorzano et al., 2001), Solorzano et al. use competing 2D level sets to segment multiple cells. In (Mukherjee et al., 2004), Mukherjee, Ray, and Acton use 2D level sets to track leukocyte cells in transillumination images over time, and, in (Ray and Acton, 2005), for validating these tracks, Ray and Acton use 2D level sets in 2D spatio-temporal images. 3D level sets are used by Sarti et al. in (Sarti et al., 2000), in this case applied to 3D volumes of nuclei in confocal microscopy images. The ability of level sets to adapt to changes in topology makes them valuable for events such as mitosis where one cell splits into two, but this only works when there is significant overlap of the cells across frames.

Wavelets (Mallat, 1989, 1997) are an effective tool for decomposing an image in both the frequency and spatial domain. Wavelet frames are a wavelet variant that avoids the sub-sampling at each decomposition level. Research on biomedical data demonstrates that they can be made robust to local noise variations and can discard low frequency objects in the background. In (Genovesio et al., 2006; Olivo-Marin, 1996), Olivo-Marin et al. use such an approach to combine coefficients at different decomposition levels to effectively segment blob-like objects. While these methods may have difficulty segmenting internal cell structures in very high resolution microscopy, our experiments have indicated that they work well for whole cell segmentation.

3. Coupled minimum-cost flow tracking

In this section, we present our cell tracking approach that we first introduced in (Padfield et al., 2009), which follows the model of independent frame segmentation followed by association. To avoid unnecessary complexity and specific models, our approach embeds all of the cell behaviors into one consistent framework that is general, extensible, simple, and fast. This is accomplished by formulating the association problem in a graph-theoretic framework as a flow network that can be solved efficiently using the minimum-cost flow algorithm. Such flow networks are generally limited to one-to-one matches, but we show how to set up the graph to also handle entering and leaving cells. Furthermore, we introduce a method we call “coupled minimum-cost flow”, which enforces a coupling of the flow of certain edges that enables it to handle mitosis and merging events that require one-to-many and many-to-one associations, respectively. This extended framework then enables the implicit modeling of different cell behaviors including moving, splitting, merging, moving into the image, and moving out of the image. We demonstrate that, after simple operations on the associated graph, a solution can be efficiently found using linear programming.

In Section 3.1 we provide an overview of the standard minimum-cost flow algorithm. In Sections 3.2 and 3.3 we show how to extend this algorithm to our “coupled minimum-cost flow” algorithm. In Section 3.4, we show how to convert the graph into a vertex-edge matrix that can be solved using linear programming. Finally, in Section 3.5, we give the precise formulation of the cost functions that are based on the features of the segmented objects.

3.1. Bipartite graphs and minimum-cost flow

To describe bipartite graphs, we first provide some intuition in the form of an analogy. Suppose we have a set of workers and a set of machines, and we know which worker can handle which machines. We are given the task of assigning workers to machines in such a way that as many machines as possible are operated by a worker that can handle it. One worker can be assigned to at most one machine, and we can assign at most one worker to one machine.

Bipartite graphs can be used to represent such objects that need to be matched between two adjacent sets (Cormen et al., 2001). A bipartite graph is an undirected graph $G = (V, E)$ whose vertices V can be divided into two disjoint sets L and R such that every edge E in the graph connects a vertex in L to a vertex in R . In the worker-machine analogy, the workers and machines are the two types of vertices L and R in the bipartite graph, and the information of which worker can handle which machines defines the edges of the bipartite graph. In our problem, the L vertices represent the cells in the previous image, the R vertices represent the cells in the current image, and the edges connecting the vertices represents the matching of cells across images. Given such a graph, a matching is a subset of edges $M \subseteq E$ such that, for all vertices $v \in V$, at most one edge of M is incident on v . We say that a vertex $v \in V$ is matched by matching M if some edge in M is incident on v ; otherwise, v is unmatched by matching M . A maximum matching is a matching M of maximum cardinality such that for any matching M' , $|M| \geq |M'|$. The notation $|\cdot|$ indicates the cardinality (number of elements) of the set. There may be many maximum matchings.

A perfect matching is a matching which matches all vertices of the graph. That is, every vertex of the graph is incident to exactly one edge of the matching. For unweighted graphs, graphs whose edges do not have associated costs, every perfect matching is maximum. In our problem, if any vertex is unmatched, it means that the cell corresponding to that vertex is not accounted for, which is unacceptable since a suitable behavior must be defined for each cell on the previous and current frames. We will show in the next section how to ensure a perfect matching for any number of cells on the previous and current frame. Fig. 1 shows a simple bipartite graph, where the left and right vertices are the black vertices and there are edges

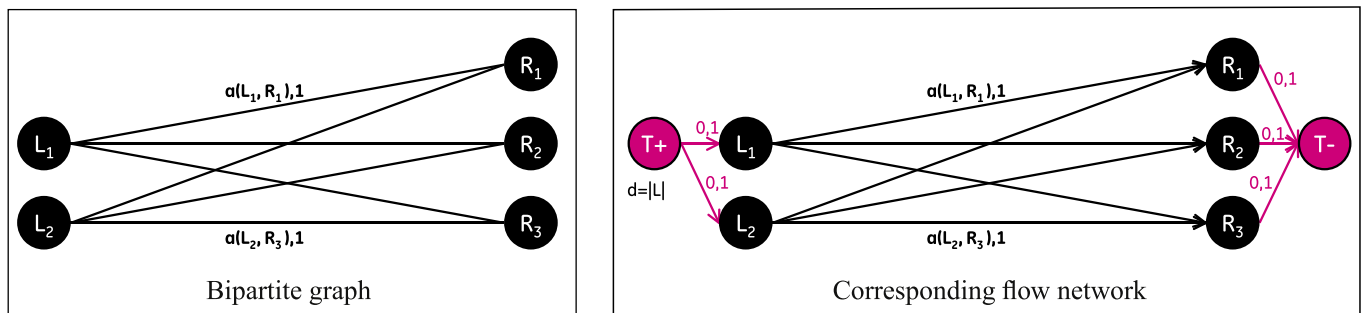


Fig. 1. Simple bipartite graph. The black vertices on the left represent the cells on the previous frame, and the black vertices on the right represent the cells on the current frame. The left vertices are fully connected to the right vertices. The source (T^+) and target (T^-) vertices are added to convert the bipartite graph to a minimum-cost flow problem. The value pair of each edge is the cost and capacity of the edge, respectively. The cost of the weighted edge is represented by a , and its calculation is described in Section 3.5.

between each of the left and right vertices. The purple vertices in the right graph are the source and target vertices needed for converting the problem to a minimum-cost flow problem as described next.

The unweighted bipartite graph is limited to finding matchings of maximum cardinality where all edges have equal cost. For matching cells across two images, weights can be defined for each pair of cells based on some measurement; for example, cells far away across two images are less likely to be matched than cells near each other. Each edge can thus be weighted by this cost rather than by unity. This can be represented as a weighted bipartite graph, which is a generalized graph where each edge has an associated cost. A minimum weighted bipartite matching is defined as a matching where the sum of the values of the edges in the matching have a minimal value. Finding such a matching is known as the assignment problem.

Maximum matching and assignment problems can be represented as a flow network problem. Before we describe mathematically the meaning of a flow network, we first provide an analogy to offer some intuition. Consider a series of water pipes that are part of a network. Each pipe has a particular diameter, and the amount of water that can flow through the pipe is limited by this diameter. Suppose that each pipe is also heated so that each one heats the water going through it by a particular amount. Wherever the pipes meet, the total amount of water coming into that junction must be equal to the amount going out. There is a source where the water originates and a sink where it leaves. We require that a certain amount of water must flow from the source to the sink. In this setup, a network flow is defined as one possible way for water to flow from the source to the sink, and the minimum-cost flow problem could be described as finding the path that sends the required amount of water through the network while heating the water as little as possible.

Converting from a maximum matching or assignment problem to a minimum-cost flow problem is accomplished by defining a flow network $G = (V, E)$. A flow network $G = (V, E)$ is a directed graph in which each edge $(u, v) \in E$ has a non-negative capacity $c(u, v) \geq 0$ (the pipe diameter in the analogy). Two special vertices that are added in the flow network are the source $T+$ with edges to all vertices in L and the sink $T-$ with edges from all vertices in R . Every vertex lies on some path from the source to the sink so that, for every vertex $v \in V$, there is a path $T+ \rightsquigarrow v \rightsquigarrow T-$. The graph is therefore connected, and $|E| \geq |V| - 1$.

A flow network by definition has the following three properties for all vertices u and v . *Capacity constraint*: The amount of flow on an edge cannot exceed the capacity of the edge. *Skew symmetry*: The net flow from u to v must be the opposite of the net flow from v to u . *Flow conservation*: The amount of flow into a vertex equals

the amount of flow out of it, which is true for all vertices except for the source, which “produces” flow, and the sink, which “consumes” flow.

Suppose that each edge (u, v) has, in addition to a capacity $c(u, v)$, a real-valued cost $a(u, v)$ (the heating of the pipe in the analogy). If we send $f(u, v)$ units of flow over edge (u, v) , we incur a cost of $a(u, v)f(u, v)$. For the minimum-cost flow problem, we are also given a *flow requirement* d with the goal of sending d units of flow from $T+$ to $T-$ in such a way that the total cost incurred by the flow is minimized. The minimum-cost flow problem can then be fully described as [Cormen et al. \(2001\)](#)

$$\text{minimize the flow cost } \sum_{(u,v) \in E} a(u, v)f(u, v)$$

subject to

$$\text{Capacity constraint : } f(u, v) \leq c(u, v) \quad \text{for each } u, v \in V$$

$$\text{Skew symmetry : } f(u, v) = -f(v, u) \quad \text{for each } u, v \in V$$

$$\text{Flow conservation : } \sum_{v \in V} f(u, v) = 0 \quad \text{for each } u \in V - \{T+, T-\}$$

$$\text{Flow requirement : } \sum_{v \in V} f(T+, v) = d$$

Given the minimum-cost flow solution, the set of all edges with flow from L to R then constitute a maximum matching with lowest cost. Minimum-cost flow can be solved by linear programming since we optimize a linear function and all constraints are simple. Matching of cells between two frames can be directly represented as a minimum-cost flow problem, but additional generalizations are required for more complex cell behaviors as described in the following sections.

3.2. Extending the graph for modeling appearing and disappearing cells

The solution to the minimum-cost flow problem on the weighted graph described in Section 3.1 may lead to unmatched vertices, which may occur when cells enter or leave the image from the border. Therefore, we here introduce an extension to the graph to enable the modeling of appearing and disappearing cells. The inclusion of models for cells moving into and out of the frame is especially important for fast moving cells or experiments where the microscope stage shifts over time. Without this extension, any cells that come into or move out of the frame during the experiment will be lost, and this could include all cells in an experiment if they move too quickly. We introduce models for cells entering and leaving by adding A (appear) and D (disappear) vertices to the graph as in [Fig. 2](#), which shows the example graph from [Fig. 1](#).

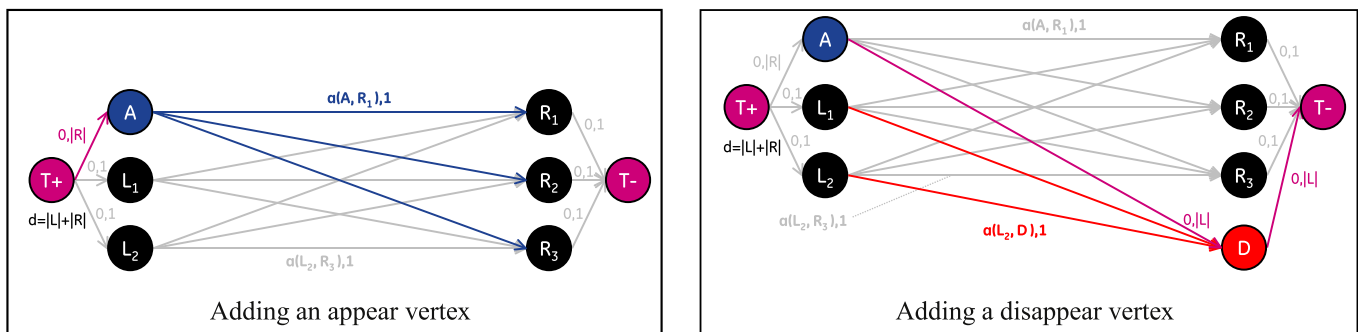


Fig. 2. Graphs incorporating matching, appearing, and disappearing cells. The appear vertex models the situation that any cells on the right image could have entered, and the disappear vertex models the situation that any cells on the left image could have exited. The cost of the edges are defined in Section 3.5. The flow requirement d is set to $|L| + |R|$ to ensure that all edges entering the left vertices and appear vertex are saturated (the flow equals the capacity) and all of the edges leaving the right vertices and disappear vertex are saturated.

Assume that there are $M = |L|$ cells in the previous image and $N = |R|$ cells in the current image, where $|\cdot|$ indicates the cardinality of the set. The appear vertex must have edges to all of the N right vertices meaning that each cell in the current frame could potentially have moved into the frame. All N edges entering the appear vertex come from the source vertex (which can be simplified as one edge with a capacity of N units). Analogously, the disappear vertex needs an edge from each of the M left vertices meaning that each cell in the previous frame could potentially have moved out of the frame. The edge leaving the disappear vertex goes to the target vertex, and the capacity of this edge is M .

For the flow requirement constraint, we found that the required flow d should be $M + N = |L| + |R|$. This is true because exactly M units need to flow to the left vertices and N units need to flow to the appear vertex. Analogously, a total of N units need to flow from the right vertices and M units need to flow from the disappear vertex. The flow requirement thus ensures that all of the left and right

vertices have exactly one entering and one leaving edge so that all of the cells on the previous and current image are accounted for. The determination of the cost of the edges is described in Section 3.5.

3.3. Extending the graph for modeling splitting and merging cells

To fully generalize the graph to handle complex cell behaviors, models for cell splitting (mitosis) and merging (occlusion) are needed. Whereas cell mitosis, where cells split into two daughter cells, is a natural phenomenon of cell growth, cell merging is generally not a biologically feasible behavior. Merge models are still important, however, because when cells occlude each other they appear to merge, so they need to be automatically split.

Although accounting for cells entering and leaving the image could be modeled in the graph by adding vertices with connections to the source and sink vertices of the graph, splitting and merging

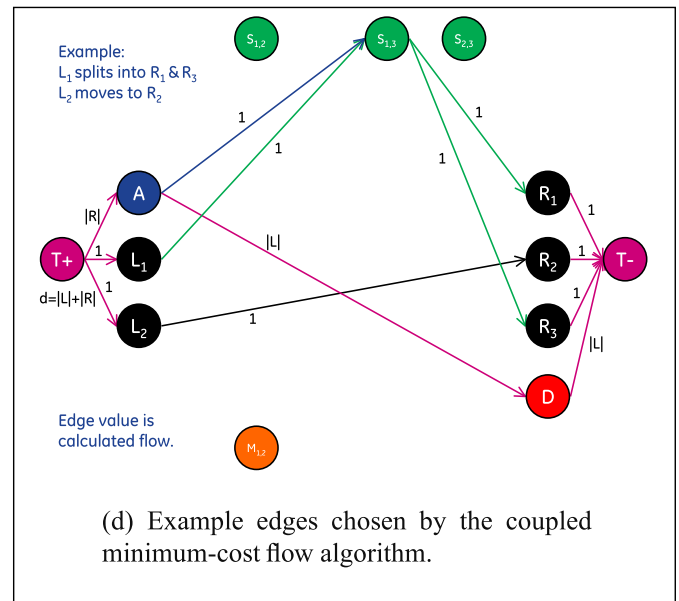
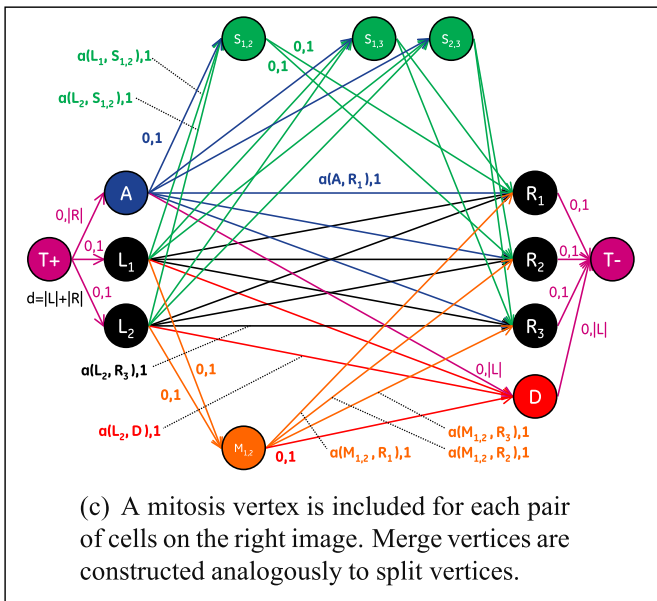
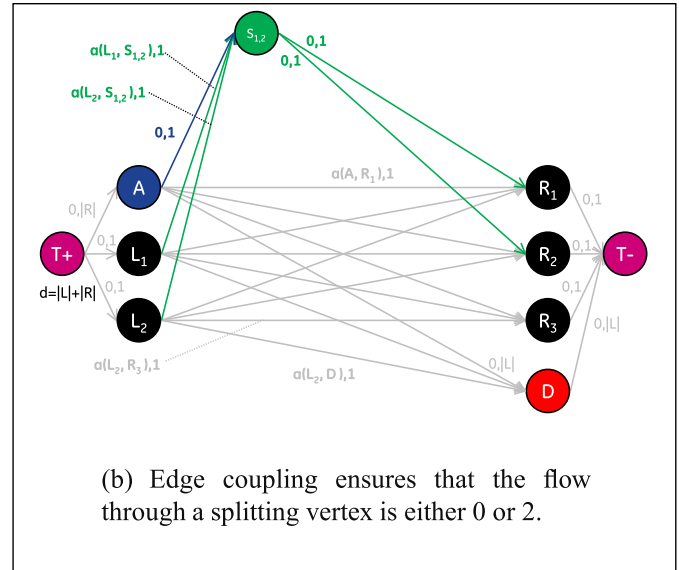
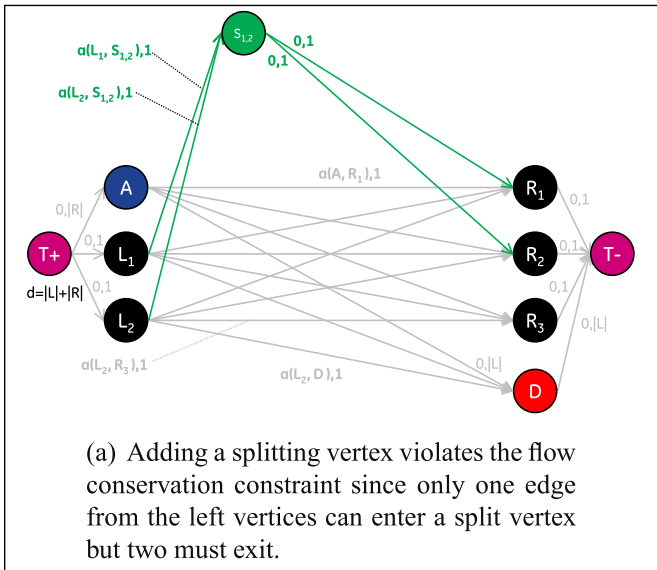


Fig. 3. Illustration of the process of adding splitting and merging vertices while satisfying the flow conservation constraint. The final graph in Fig. 3c incorporates matching, appearing, disappearing, splitting, and merging cells. In this example, two cells are present on the previous image ($M = 2$) and three on the current image ($N = 3$).

cells cannot be represented directly. This is due to the fact that the number of edges entering a split (merge) vertex is different from the number of edges leaving, which violates the flow conservation constraint of the graph. To satisfy the flow conservation constraint for splitting cells, we found that an edge can be included from the appear vertex to each of the splitting vertices as shown in Fig. 3b. This enables the appear vertex to feed the split vertex with an additional unit so that the number of edges entering and leaving a split vertex is equal. Merging vertices are handled analogously as in Fig. 3c.

However, one problem remains: a constraint is needed to ensure that exactly two units flow to and from splitting and merging vertices when those vertices are part of the solution. Without such a constraint, it is possible that one unit would flow into and out of a splitting (merging) vertex, leading to an incoherent representation of cell behavior. For example, in Fig. 3c, assume that in the solution of the graph one unit flows from L_1 to $S_{1,2}$ and then from $S_{1,2}$ to R_1 . Thus, although the vertex $S_{1,2}$ by definition means that a cell in the previous image split into two daughters R_1 and R_2 in the current image, R_2 is still unmatched and may be matched to another cell on the left. This is a violation of feasible cell behavior, since it is not clear whether R_2 is a daughter of a splitting cell or simply a moving cell.

To enforce the constraint of exactly two (or zero if the vertex is not part of the solution) units of flow entering and exiting split and merge vertices, we introduce the concept of edge coupling. This coupling ensures that if the edge from a left vertex to a split vertex is chosen, then the corresponding edge from the appear vertex to the split vertex is also chosen. Thus, exactly two units enter the splitting vertex, and this forces exactly two units to exit the splitting vertex as well. On the other hand, if the edge from a left to a split vertex is not chosen, the edge from the appear vertex to that split vertex is also not chosen, so the total flow through the split vertex is zero. Also, since the capacity of the edge from the appear vertex to the split vertex is exactly 1, only one of the edges coming from the left vertices can enter each split vertex, meaning that no more than one cell in the previous image can be matched to a given pair of daughter cells in the current image. The coupling for the merging vertices is analogous: if an edge is chosen from a merge vertex to a right vertex, then the edge from that merge vertex to the disappear vertex is also chosen. We named this algorithm the “coupled minimum-cost flow” algorithm, and we will show in Section 3.4 a simple way of realizing this coupling.

Given the cost and capacity constraints and flow requirement d , the flow equations can be determined as shown in the group of equations below. The colors in the equations match the colors of the corresponding edges in Fig. 3. In each equation, the terms on the left represent the flow entering the vertices, and the terms on the right represent the flow leaving the vertices. Setting the flow requirement d to exactly $M + N$ ensures that the flow through all edges leaving the source vertex (and all those entering the target vertex) is equal to the capacity of those edges. Thus, the sum of the flows is equal to the capacities u , as shown in the second line of the “Source” and “Target” equations. This requirement means that all of the edges entering the left vertices and all of the edges leaving the right vertices are part of the solution (meaning that all cells on the previous and current image will be associated with some behavior). Therefore, the flow into each of the left vertices (and out of each of the right vertices) is exactly 1, which is used to define the flow equations for the “Left” and “Right” vertices. The flow requirement also causes the flow entering the appear vertex to be exactly N , and the flow exiting the appear vertex to be exactly M , which are used to define the “Appear” and “Disappear” flow equations. The “Split” and “Merge” flow equations state that the flow entering the vertex must be equal to the

flow exiting the vertex. This flow is either 2 if the vertex is part of the solution, or 0 if not.

Source:	$\sum_{i=1}^M f(T+, L_i) + N \cdot f(T+, A)$ $= M + N = d = \sum_{i=1}^M u(T+, L_i) + N \cdot u(T+, A)$
Target:	$\sum_{j=1}^N f(R_j, T-) + M \cdot f(D, T-)$ $= M + N = d = \sum_{j=1}^N u(R_j, T-) + M \cdot u(D, T-)$
Left:	$f(T+, L_i) = 1 = \sum_{j=1}^N f(L_i, R_j) + f(L_i, D)$ $+ \sum_{j=1}^N \sum_{k=j+1}^N f(L_i, S_{j,k}) + \sum_{k=i+1}^M f(L_i, M_{i,k}) + \sum_{l=1}^{i-1} f(L_i, M_{l,j})$ $(i = 1 \dots M)$
Right:	$\sum_{i=1}^M f(L_i, R_j) + f(A, R_j) + \sum_{k=j+1}^N f(S_{j,k}, R_j) + \sum_{l=1}^{j-1} f(S_{l,j}, R_j)$ $+ \sum_{i=1}^M \sum_{k=i+1}^M f(M_{i,k}, R_j) = f(R_j, T-) = 1$ $(j = 1 \dots N)$
Appear:	$f(T+, A) = N = \sum_{j=1}^N f(A, R_j) + f(A, D) + \sum_{j=1}^N \sum_{k=j+1}^N f(A, S_{j,k})$
Disappear:	$\sum_{i=1}^M f(L_i, D) + f(A, D) + \sum_{i=1}^M \sum_{k=i+1}^M f(M_{i,k}, D) = f(D, T-) = M$
Split:	$\sum_{i=1}^M f(L_i, S_{j,k}) + f(A, S_{j,k}) = f(S_{j,k}, R_j) + f(S_{j,k}, R_k)$ $(j = 1 \dots N, k = j + 1 \dots N)$
Merge:	$f(L_i, M_{i,k}) + f(L_k, M_{i,k}) = \sum_{j=1}^N f(M_{i,k}, R_j) + f(M_{i,k}, D)$ $(i = 1 \dots M, k = i + 1 \dots M)$

Given the flow equations, the coupling constraints can be defined. They affect only the edges entering and exiting the split and merge vertices. We found that the coupling can be enforced by simply setting equal all terms in both the “Split” and “Merge” equations. This ensures that if any of the edges entering a split vertex from a left vertex is chosen, the appear vertex is also chosen, and so are both edges exiting the split vertex. The case is analogous for the merge vertices. We will show in Section 3.4 an efficient method for enforcing the coupling by a simple combination of edges.

Split:	$\sum_{i=1}^M f(L_i, S_{j,k}) = f(A, S_{j,k}) = f(S_{j,k}, R_j) = f(S_{j,k}, R_k)$ $(j = 1 \dots N, k = j + 1 \dots N)$
Merge:	$f(L_i, M_{i,k}) = f(L_k, M_{i,k}) = \sum_{j=1}^N f(M_{i,k}, R_j) = f(M_{i,k}, D)$ $(i = 1 \dots M, k = i + 1 \dots M)$

Fig. 3d shows an example solution of the resulting graph incorporating cells moving, entering, leaving, splitting, and merging. In this figure, $S_{1,3}$ is connected to L_1 , A , R_1 , and R_3 , indicating that cells 1 and 3 are daughters of cell 1 in the previous image. Also, L_2 is connected to R_2 meaning that cell 2 in the previous frame moved to cell 2 in the current frame. It can easily be verified that the flow through the graph is exactly $M + N = |L| + |R| = 5$.

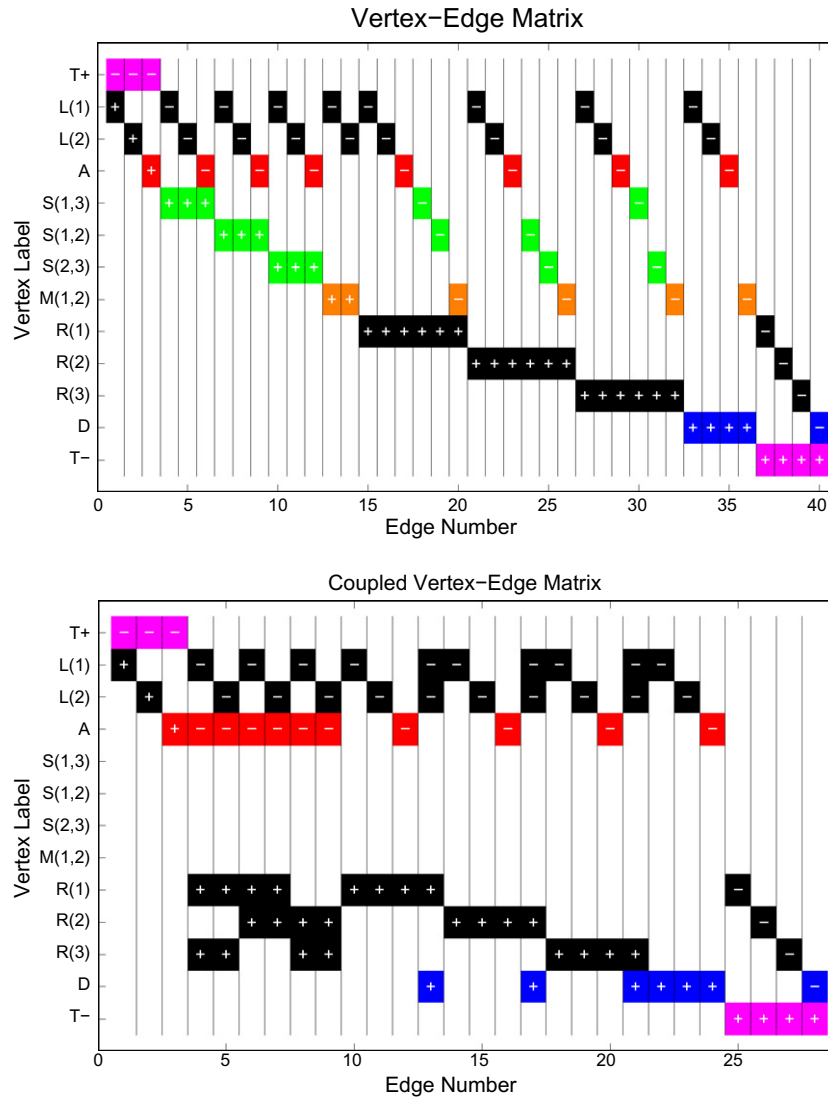


Fig. 4. Incidence matrices corresponding to the graph in Fig. 3c. In the incidence matrix on the top, each column corresponds to an edge in Fig. 3c and is color-coded as in that figure. Each column has only two entries since each edge has a unique source and destination. However, in the coupled incidence matrix on the bottom, the coupled matrix allows columns to account for more than one edge and thus have more than two entries. The coupling of the edges entering and exiting splitting and merging vertices enables the removal of these vertices, so there are no split (green) or merge (orange) entries in this matrix. The vertex labels are indicated on the y-axis, and the columns correspond to the edges of the graph. For each column, the entries with negative signs indicate the source vertex of an edge and those with positive signs indicate the destination vertex of an edge. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.4. Solving the coupled minimum-cost flow problem

To solve the coupled minimum-cost flow problem for the graph shown in Fig. 3c, an incidence matrix of size $|V| \times |E|$ can be used, where each column corresponds to an edge, and each row corresponds to a vertex. The top matrix in Fig. 4 shows the incidence matrix corresponding to the graph in Fig. 3c. In this matrix, all entries in each column are zero except those corresponding to the vertices that are connected by the edge corresponding to the column. The entry in a column corresponding to the start of the edge is set to -1 , and the entry corresponding to the end of the edge is set to 1 . The E columns consist of various sets of associations: those corresponding to moving cells, appearing cells, disappearing cells, splitting cells, and merging cells.

We developed a straightforward method of enforcing the coupling constraint using the incidence matrix representation. For each split (and merge) vertex, the columns representing the two edges entering and the two edges leaving the vertex (for a total of four columns) are added together to form a new column, and the four col-

umns are removed. For each split (and merge) vertex, this yields a column that has more than two nonzero entries so that, in order for this column to be included in the solution, all edges included in this combined column must be chosen together. Combining the columns in this manner has the added benefit of reducing the number of edges and vertices in the incidence matrix, reducing the computational complexity of the optimization. The bottom matrix in Fig. 4 shows the coupled incidence matrix corresponding to the incidence matrix on the top in Fig. 4. Because the edges entering and exiting the split and merge vertices are summed, the split and merge vertices are no longer needed and are not present in this matrix. Several columns of this matrix have four entries rather than two, indicating that edges of the appear (blue) or disappear (red) vertices are coupled with corresponding left (black) or right (black) edges, respectively.

Using the coupled incidence matrix, finding the optimal matches corresponds to finding a subset of columns in this matrix such that the sum of the cost of these columns is minimized, under the constraint that no two columns share common nonzero entries. This can be represented by the linear programming problem

$$\min_x (c^T x) \quad \text{subject to: } Ax \leq b \quad (1)$$

where c is the vector of costs, b is a vector of size $|V| \times 1$ representing the sum of the flow for each vertex (0 for all vertices except source and target), A is the coupled incidence matrix, and x is a binary $|E| \times 1$ solution vector that is 1 if the row is in the solution and 0 otherwise. Here $|V|$ and $|E|$ are the number of vertices and edges, respectively, in the coupled incidence matrix. The solution x is then found using linear programming (Cormen et al., 2001).

3.5. Calculation of association costs

Given the coupled minimum-cost flow approach to optimal cell association, it remains to define the association costs for the different kinds of cell behaviors. These association costs provide the data-driven component of the algorithm. Because the algorithm minimizes the cost of the graph, we desire the association costs to be small for correct associations. Fortunately, regardless of the sign of the costs, the algorithm will not revert to the trivial solution where none of the edges are chosen because of the flow requirement d of the coupled minimum-cost flow framework. Given the segmentations that will be described in Section 4.1, we calculate a set of features ζ on each cell and calculate feature difference vectors between cells on adjacent images $\vartheta(u, v) = |\zeta(u) - \zeta(v)|$. Any features of the cells can be used for this step such as location, area, eccentricity, and intensity (Padfield and Miller, 2008) as well as features calculated from the wavelet coefficients.

The association costs can be found by some weighted combination of each of the cell difference features. We have found that a cost that sums the weighted elements of the feature difference vector works well, although more complicated methods could be used. The weight vector w defines the approximate difference in scale of the various feature difference elements. This cost function is given in the following equation:

$$a(u, v) = \sum_{i=0}^K w_i |\zeta_i(u) - \zeta_i(v)| \quad (2)$$

where K is the number of features. The calculation of the feature difference vectors is handled separately for the different cell behaviors of moving, splitting, merging, appearing, and disappearing as described next.

3.5.1. Cell moving

For the cell movement cost, the features $\zeta(v)$ with $v = 1:N$ of all cells on the next slice are compared to the features of each of the cells on the current slice $\zeta(u)$ with $u = 1:M$ so that $\vartheta(u, v) = |\zeta(u) - \zeta(v)|$ is the absolute difference between the feature vectors. The resulting cost calculated using Eq. (2) is small for cells that closely match.

3.5.2. Cell splitting and merging

For the cell splitting cost, a cell on frame t will be associated with two cells on frame $t + 1$. In our formulation, $\zeta(v_1, v_2)$ represents the feature vector calculated from the combination of cells v_1 and v_2 in frame $t + 1$. In other words, rather than calculating the features on the segmented region of one cell, the features are calculated on the segmented region of both cells together as if they were one object. Thus, the absolute difference vector for splitting is $\vartheta(u, (v_1, v_2)) = |\zeta(u) - \zeta(v_1, v_2)|$. For the hypothesis of cell merging, which models occlusion since cells cannot naturally merge, the absolute difference vector is $\vartheta((u_1, u_2), v) = |\zeta(u_1, u_2) - \zeta(v)|$.

3.5.3. Cell appearing and disappearing

We calculate the cost of cell appearance and disappearance based on how far the cell is from an image border. When a cell appears, it is present on frame $t + 1$ and absent on frame t . In our formulation, $\zeta(\emptyset) = 0$ represents the features of a missing object, where $\zeta_i(\emptyset)$ is the distance to the image border if i is the distance feature and otherwise it is the mean of the features of the cells on the other image. This leads to a feature difference vector defined as $\vartheta(\emptyset, v) = |\zeta(\emptyset) - \zeta(v)|$. Analogously, when a cell disappears, it is present on frame t and absent on frame $t + 1$. This leads to a feature difference vector defined as $\vartheta(u, \emptyset) = |\zeta(u) - \zeta(\emptyset)|$.

3.6. Coupled minimum-cost flow algorithm

Our coupled minimum-cost flow algorithm is presented in Algorithm 1. An analysis of the computational complexity of the coupled minimum-cost flow solution is given in Appendix A.

Algorithm 1. Summary of coupled minimum-cost flow method.

- Input:** All $|L|$ cells from one image and all $|R|$ cells from another image.
Output: Calculated behaviors (move, appear, disappear, split, merge) for each cell.
- 1 Create a graph $G = (V, E)$ with the L vertices on the left, and the R vertices on the right;
 - 2 Add source, sink, appearing, disappearing, splitting, and merging vertices with corresponding edges as in Figs. 1–3;
 - 3 Set all edge weights based on the features according to Fig. 3(c) and Section 3.5;
 - 4 Convert the graph to an incidence matrix as in the top of Fig. 4;
 - 5 Couple the matrix columns of the splitting (merging) edges with the columns of the appearing (disappearing) edges as in the bottom of Fig. 4;
 - 6 Solve for the minimum-cost flow using linear programming on the coupled incidence matrix as in Eq. (1);

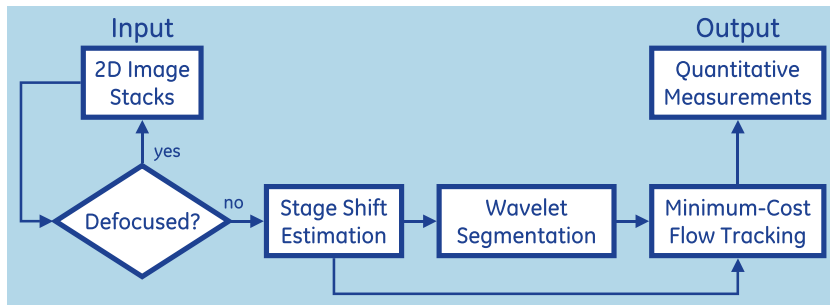


Fig. 5. Flowchart showing the relationship of the components of the segmentation and tracking framework.

4. Overall framework for cell tracking and segmentation

In this section, we present additional components required for a generalized tracking framework that enables accurate tracking and segmentation of cells. We describe our segmentation algorithm that is based on combinations of wavelet coefficient. We then describe our algorithms for stage shift correction and defocus detection, which comprise integral parts of the framework that enable accurate tracking and correct cell measurements. The overall framework is shown in the flowchart of Fig. 5.

4.1. Wavelet-based cell segmentation

Accurate cell segmentation is a crucial step for measuring the costs in Section 3.5 and for accurate quantitation. For this task, we choose to use wavelets (Mallat, 1989) for several reasons: wavelets decompose the image into both the spatial and frequency domain enabling effective scale-space analysis, the calculation of wavelets across multiple scales is fast and computationally efficient, and the number of parameters can be limited.

To denoise the images and segment the cells, we use an algorithm based on the shift-invariant wavelet frames transformation of the image as well as the filtering of non-salient wavelet coefficients (Padfield et al., 2008). Wavelet frames are identical to the standard wavelet transform except that the decimation operation at each level is omitted. Prior research on biomedical data (Genovesio et al., 2006; Olivo-Marin, 1996) demonstrates that this wavelet transform is robust to local noise variations and discards low frequency objects in the background. The decomposition is represented as

$$I_i(x, y) = \sum_{m, n} h(m, n) I_{i-1}(x - 2^{i-1}m, y - 2^{i-1}n) \quad (3)$$

$$W_i(x, y) = I_i(x, y) - I_{i+1}(x, y) \quad (4)$$

where I_i and W_i represent the approximation and detail images, respectively, at each scale, i , and $h(m, n)$ denotes the scaling function (we use a B_3 spline), and m and n are x and y indices.

The recursive definition in Eq. (3) is initialized by setting $I_0(x, y)$ to the original discrete image. The convolution can be implemented efficiently by multiplying the Fourier transform of the image with that of the scaling function with zero taps inserted at each scale.

From this decomposition, the reconstruction of the original image can be computed by summing all detail images $W_i(x, y)$ with the approximation image at the final scale $I_s(x, y)$.

$$I_0(x, y) = I_s(x, y) + \sum_{i=1}^s W_i(x, y) \quad (5)$$

Using the decomposition, the images can be directly denoised in the wavelet coefficient space. Assuming that the image noise is additive, the corresponding wavelet transformation results in coefficients generated by the underlying signal W^d and those that correspond to image noise W^N . To approximate the signal term, we threshold the image stack with an Amplitude-scale-invariant Bayes Estimator (ABE) using Jeffreys' non-informative prior (Figueiredo and Nowak, 2001; Donoho and Johnstone, 1994; Donoho, 1995) as an estimate of the significance of wavelet coefficient $W_i^d(x, y)$ at a given scale i and position (x, y)

$$W_i^d(x, y) \approx \delta^{ABE}(W_i(x, y)) = \frac{(W_i(x, y)^2 - 3\sigma_i^2)_+}{W_i(x, y)} \quad (6)$$

where σ_i^2 is the estimated noise variance at a given scale i calculated in a background patch. The background patch can be calculated in a variety of ways. In our case, we are dealing with fluorescent images, so the background is dark since it represents a lack of signal. Therefore, our background mask is the 5% of pixels that are the darkest in the original image, and the σ_i^2 is calculated at each scale based on these pixels. In order to further reduce noise and enhance objects that extend across multiple resolutions, we compute a correlation stack $C_s(x, y)$, which is the multiplication of a subset of the denoised wavelet coefficients corresponding to the selected scales

$$C_s(x, y) = \prod_{i=j_l}^{j_u} W_i^d(x, y)_+ \quad (7)$$

One of the advantages of the wavelet-based method for denoising and segmentation is how it naturally represents objects at different scales. We observed that the selection of which scales to combine in Eq. (7) can be determined from the resolution of the images and the biological variation in cell size. In Section 5, we indicate how to choose the scales in a way that is robust across multiple datasets. The subset of the channels used for the correlation stack are chosen to correspond with the scales at which cells appear, thus ignoring scales containing only noise and those containing only low-frequency background information. The segmentation is then obtained as the extraction of connected regions in the correlation stack for coefficients greater than zero. Since the correlation stack empha-

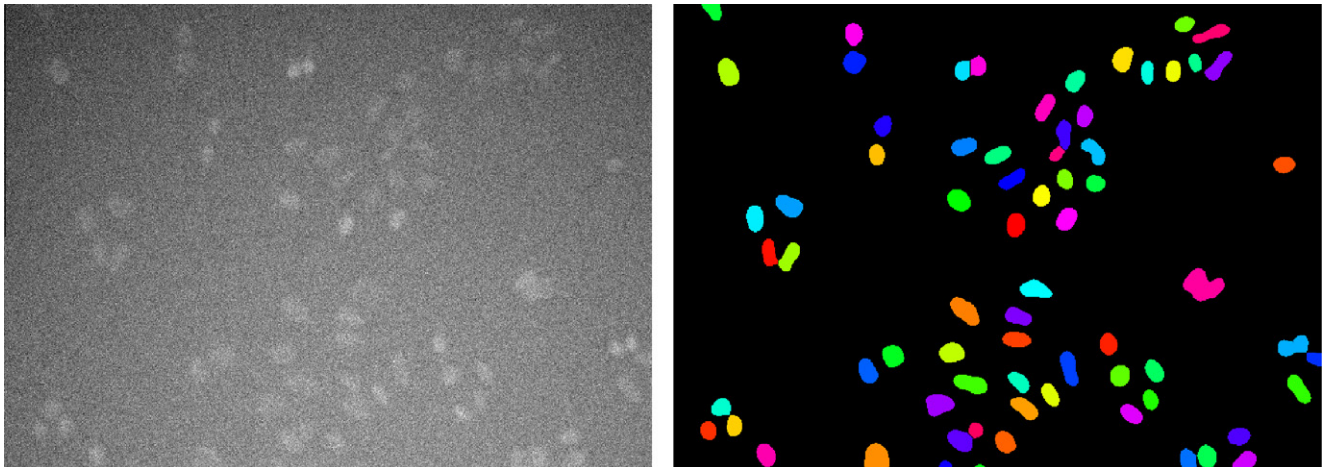
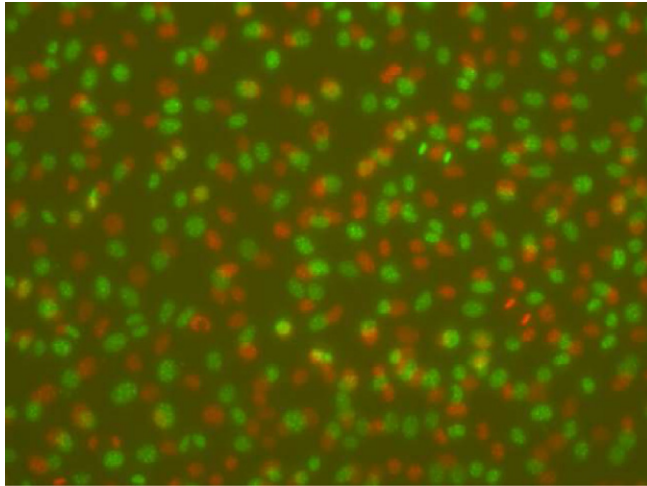
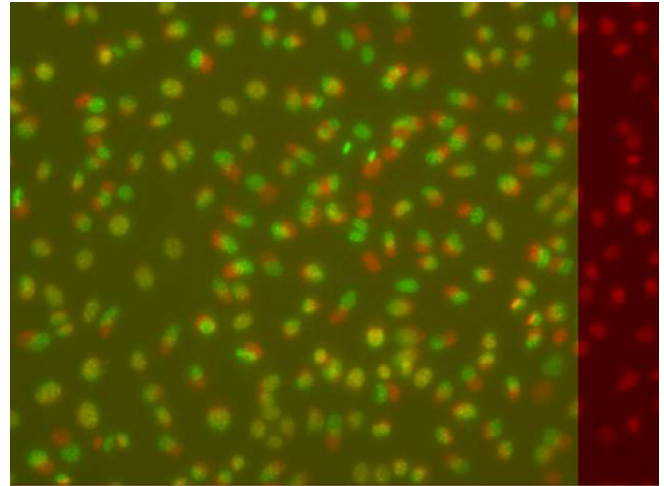


Fig. 6. Dataset with low contrast-to-noise. The figure on the left shows a cropped view of a typical low-dose Hoechst-stained image, and the figure on the right shows the segmentation using the wavelet approach. Contrast-to-noise (CNR) is defined as $\frac{\mu_S - \mu_N}{\sigma_N}$, where μ_S is the signal mean, μ_N is the noise mean, and σ_N is the noise standard deviation. For these images, $\text{CNR} = 0.5$, indicating that the average intensity difference between the signal and the noise is only half of the standard deviation of the noise. The intensity range is only 20 gray-scale levels.



(a) Original overlay of two frames showing stage shift.



(b) Two frames with stage shift removed.

Fig. 7. Removing the stage shift effect. Fig. 7a shows the original overlap between two successive frames, with the previous frame shown in red and the current in green. Fig. 7b shows the same figure corrected for stage shift. The only remaining shift between cells on successive frames is the result of actual cell movement. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

sizes the frequencies corresponding to the cells, no post-processing is necessary to yield consistent filled shapes.

We also utilize biological insight of the cell cycle to guide the segmentation process. A cell begins its life in the G1 phase with a given cell size. As it proceeds through the cell cycle, it duplicates its DNA in the S phase and, by the end of the G2 phase, it has duplicated its internal organelles and structures and has doubled in size. This doubling is essential since the cell breaks into two equal parts in mitosis, each of which has the same size as the original cell. This insight guides the construction of our scale selection in that it specifies that as cells proceed through the cell cycle they appear at two different scales.

To accomplish this, we apply watershed segmentation in the wavelet coefficient space. The coefficients of combined lower scales $C_a(\bar{x})$ (where \bar{x} is a vector indicating the (x,y) location of a pixel) are then used as seeds in a seeded watershed algorithm (Soille, 2003), denoted by ws , to separate the clustered cells resulting from the combined coefficients of the higher scale $C_b(\bar{x})$.

$$O(\bar{x}) = ws[(\max_{\bar{x}}(D(\bar{x})) - D(\bar{x}) + 1) \wedge (C_a(\bar{x}) > T)] \quad (8)$$

$$D(\bar{x}) = \min_{\bar{y} \in V^-} \|\bar{x} - \bar{y}\|$$

$$V^- := \{\bar{y} \in C_b(\bar{y}) | C_b(\bar{y}) < T\}$$

Here V^- designates all background pixels, T is a threshold set to 0, and \wedge is the point-wise minimum of the arguments. The distance map $D(\bar{x})$ is zero for all background pixels and records for all foreground pixels the closest distance to a background pixel. The distance map $D(\bar{x})$ is zero for all background pixels and records for all foreground pixels the closest distance to a background pixel.

Our wavelet segmentation approach can segment images even in the presence of relatively low contrast-to-noise and in the presence of flat field effects that manifest as slowly-varying backgrounds. Fig. 6 shows an example segmentation demonstrating the accuracy of our wavelet segmentation approach even with a very low contrast-to-noise ratio. Under such conditions,

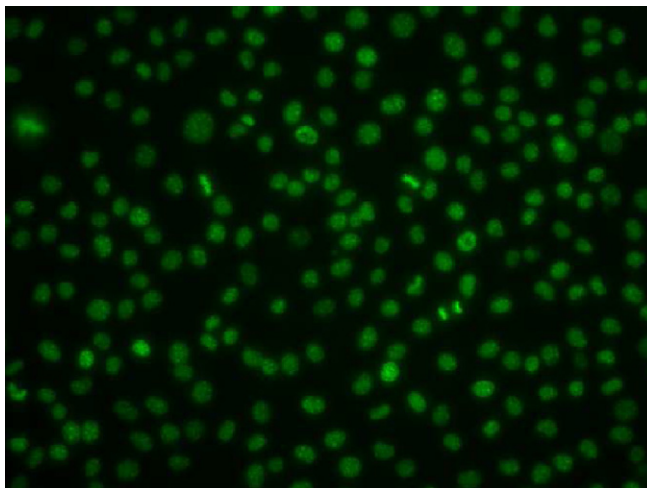
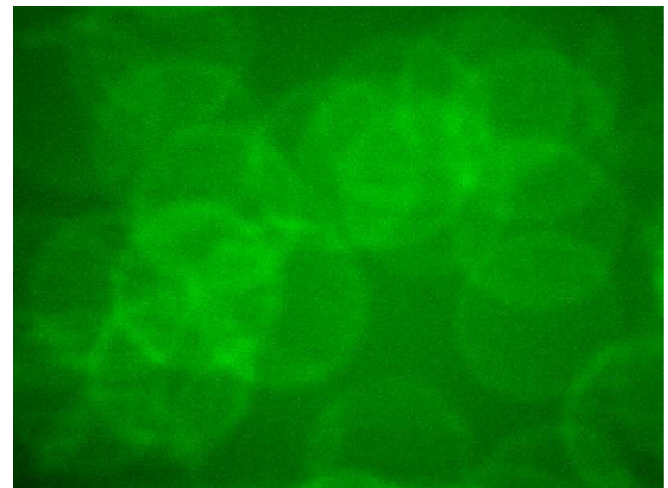
(a) Normal image. $R_d = 1.0733$ (b) Defocused image. $R_d = 3.6868$

Fig. 8. Comparison of normal and defocused images. The R_d number listed for each figure is the defocused ratio measured by our algorithm that indicates how defocused an image is. This defocused ratio is defined in the text.

approaches such as thresholding will have great difficulty and will likely require additional heuristics, and approaches such as active contours will have difficulty evolving only inside the object without spreading into the background. Under-segmentation errors are corrected implicitly by the tracking algorithm by splitting objects detected as having merged, whereas over-segmentation errors will result in detection of false mitosis events by the tracker.

4.2. Global alignment for stage-shift removal

In time-lapse high-throughput experiments, the microscope may drift over time. Experimenters occasionally desire to remove the cell plate from the microscope, for example to add some compound to the experiment, and then return it to continue the time-course. When the effect of the stage shift from these conditions is ignored in cell tracking experiments, two conditions can arise: (1) either the algorithms fail to track the cells, or (2) even if the algorithms are robust enough to correctly track the cells, the global stage shift movement will be added to the cell movement and will bias the cell velocity and direction calculations. It is therefore necessary to automatically correct for this stage shift.

An example of the stage shift effect is shown in the Fig. 7a. This figure shows the overlap of the previous frame in red and the current frame in green. Fig. 7b shows the same images with the stage shift corrected by the method presented in this section. In this image, most of the cells are yellow, indicating overlap of the cells. The remaining green and red cell regions represent the true local cell movements.

We approach this as a registration problem. While standard registration approaches could potentially be used to solve this problem, few constraints can be placed on the transformation between images since they could be significantly misaligned. With large transforms between images, registration methods based on local search can become stuck in a local minimum, and many such approaches require careful setting of the parameters. We therefore employ an algorithm that calculates the translation, rotation, and scale using the Fourier transform and that finds a global minimum, requires no parameters, and is fast.

The registration is based on the normalized cross correlation metric (Lewis, 1995), which is appropriate for several reasons. It is insensitive to multiplicative factors between the two images and produces a cost function with sharp peaks and well-defined minima. Furthermore, since the images are acquired using the same imaging modality, more complex metrics such as mutual information are not needed. And, importantly, this metric can be calculated using the Fourier transform, which leads to significant increase in speed and enables the calculation of the metric globally for all correlation locations of the two images. Our approach is similar to that of Reddy and Chatterji in (Reddy and Chatterji, 1996), and we refer the reader to that reference for details. Briefly, the method uses the Fourier translation, rotation, and scale properties to perform the registration. It is founded on the principles that the

Fourier transform is translation invariant and its conversion to log-polar coordinates converts the scale and rotation differences to vertical and horizontal offsets that can be measured. Our full algorithm for finding the registration transform (translation, rotation, and scale) is given in Algorithm 2.

Algorithm 2. Algorithm for image registration to compensate for translation, rotation, and scale.

Input: Fixed image $f_1(x,y)$ and moving image $f_2(x,y)$
Output: Registered moving image $\hat{f}_2(x,y)$ and transform parameters x_0, y_0, θ_0, τ

- 1 Calculate the Fourier transforms of the images F_1 and F_2 ;
- 2 Calculate the log-polar images $N_1(\log \rho, \theta)$ and $N_2(\log \rho, \theta)$;
- 3 Correlate the log-polar images using normalized cross-correlation (NCC) to find θ_0 and τ ;
- 4 Transform the moving image with θ_0 and τ to find $\tilde{f}_2(x,y)$;
- 5 Correlate $f_1(x,y)$ with $\tilde{f}_2(x,y)$ using NCC to find x_0 and y_0 ;
- 6 Transform $\tilde{f}_2(x,y)$ with x_0 and y_0 to find $\hat{f}_2(x,y)$;

The result of this processing measures the global stage shift and focuses the tracking algorithm on the local cell movements. Once the transformations between successive images has been calculated, these transformations are passed to the tracking step to correct the cell location.

4.3. Defocus detection

In high-throughput applications, the microscope acquires each image quickly and then moves on to the next, leaving little time for refocusing in cases where the specimen was not fully in focus. The application of the segmentation algorithm on such defocused images yields meaningless results that can also adversely affect the tracking approach. To address this, we have developed an algorithm to detect such defocused images automatically so they can be removed from analysis (Padfield et al., 2008). Fig. 8 shows an example of a normal and a defocused image.

The algorithm is based on the principle that a defocused image acquires a smoothed appearance, losing its texture. Such effects can be effectively captured using wavelet transforms that decompose an image into 1 approximation channel (ϕ) and 3 detail channels (ψ_0, ψ_1, ψ_2) that encode both the frequency and spatial information of the image. At successive iterations of the decomposition, the approximation (lowpass) image is further decomposed into 1 approximation channel and 3 detail channels, and the resulting images at each level are decimated by 2. Using the wavelet decomposition, the energy in each channel provides insight into the texturedness of the image. Smooth images tend to concentrate most of the energy in the low frequency channels, whereas the wavelet decomposition of textured images show energy spread throughout the channels (Porter and Canagarajah, 1996). For the application of defocused image detection, the high-frequency noise

Table 2

Dataset descriptions. This table gives a brief overview of the different datasets. The acquisition time spacing for datasets 1–4 is 15 min; for dataset 5 it is 6 min.

Dataset	Wells	Time points	Images	Total cells	Total tracks	Mitosis	Appear	Disappear
1	96	41	3936	168,589	10,151	467	5499	5037
2	24	41	984	116,041	13,359	521	9487	9699
3	4	112	448	28,570	558	71	168	211
4	1	88	88	12,301	258	20	83	89
5	36	12	432	68,270	7517	205	1501	1518
Total	161	294	5888	393,771	31,843	1284	16,738	16,554

is still present and thus provides little discriminative information. However, the middle frequencies are attenuated relative to in-focus images. Using this intuition, we calculate the defocus ratio R_d of the lower frequency channel to the middle frequency channels as

$$R_d = \frac{E(\phi_n)}{\sum_{j=n-1}^n \sum_{i=0}^2 E(\psi_{ij})} \quad (9)$$

where $E(\cdot)$ represents the energy operator, the first subscript of ψ represents the indices of the three detail channels, and the second subscript represents the level of the decomposition, with higher numbers signifying lower frequencies, and n corresponds to the lowest frequency of the given decomposition (we typically set $n = 5$). The energy is defined as the mean magnitude of the wavelet coefficients for that sub-band

$$E(x) = \frac{1}{J} \sum_{i=0}^{J-1} \sum_{j=0}^{J-1} |x(i,j)| \quad (10)$$

where J is the number of coefficients in channel x , and $x(i,j)$ is the value of the coefficient. Using this defocus ratio, it is possible to distinguish between focused and defocused images: if $R_d > \tau_d$, where τ_d is a threshold, the image is classified as defocused and discarded; otherwise the image is considered in focus and is retained.

5. Results

In this section, we present results of the tracking and segmentation algorithms on five time-lapse datasets with varying conditions and biological hypotheses and multiple wells as outlined in Table 2. For all of the experiments, the wells were imaged on a GE IN Cell Analyzer 1000 with environmental control (37 °C, 5% CO₂). The cell nuclei were stained with a nuclear DNA dye Hoechst, which is part of a family of fluorescent stains for labeling DNA in fluorescence microscopy.

Each of the datasets consists of images taken from different wells with many of the wells containing different doses of various treatments. In the first and second experiments, the movement of L929 mouse fibroblasts was monitored at 20× magnification (pixel spacing of 0.323 μm). This dataset includes both control wells and wells containing 10% horse serum, which is a source of growth factors and other agents promoting cell motility. The purpose of the third experiment is the study of the effect of the cell-cycle inhibitor compound Roscovitine in HeLa cells by tracking the cells at 20× magnification (pixel spacing of 0.323 μm). In the fourth experiment, the goal is to track a large number of cells at the low magnification of 10× (pixel spacing of 0.645 μm). The purpose of the fifth experiment is to measure the cytoplasm to nucleus translocation of proteins, which is a key step in signal transduction from the cell cyto-

plasm to the nucleus in the control of gene expression. To accomplish this, the distribution of fluorescence throughout the cell reported by a STAT3-GFP fusion protein is measured. The magnification was 20× (pixel spacing of 0.323 μm).

In the following, our results on these datasets are concerned with the evaluation of the segmentation and tracking algorithms since these algorithms are the focus of this paper. In total, nearly 400,000 cells were segmented and tracked representing nearly 32,000 tracks.

For each of the datasets, 25% of the wells were chosen at random and validated for segmentation and tracking accuracy, representing a validation of over 104,000 cells and nearly 8,000 tracks. To enable such large-scale validation, rather than requiring the user to manually mark all cells from scratch, the automatic results were presented to the user whose corrections were recorded and tabulated. The details of how this validation was carried out for tracking and segmentation are listed in Sections 5.1 and 5.2, respectively. The results were validated by two expert biologists: one from the Ordway Research Institute, and one from GE Healthcare.

The time required by each of the algorithms is as follows. Tracking: 0.17 seconds/cell; segmentation: 11.9 s/image; stage shift correction: 9.7 s/image; defocus detection: 1.1 s/image. For an average image with 50 cells, this leads to a total processing time of 31.2 s. This timing was calculated on a dual-core 2.6 GHz Dell laptop with 3.5 GB of RAM. The algorithms were implemented in Matlab, and it is expected that the processing time will decrease significantly for a C/C++ implementation.

In total, there are only two parameters, and their settings are held constant for all experiments. The only defocus detection parameter τ_d is the defocus detection threshold; it is set at $\tau_d = 1.9$ to avoid any missed detections of defocused images in a large training set (see (Padfield et al., 2008) for more details). There are no parameters for the stage shift correction. The segmentation parameters for the lower j_l and upper j_u scales of Eq. (7) depend only on the pixel size (image spacing) ζ since the magnification factor of the image determines the segmentation scale; we set these as $j_l = 5 - (3*\zeta)$ and $j_u = j_l + 1$. Setting j_u to one more than j_l indicates that two scales are being combined. While these parameters are correlated with the size of the cell nuclei, they span a large range of cell sizes since the wavelet decomposition captures features across several scales. Since they depend on the image spacing, no parameter setting by the user is needed. The only tracking parameter is the feature weights w . We found that only using the two features of distance moved and change in area yielded sufficiently accurate results. We set the weights empirically to $w = [110]$, where the elements are distance and area, and we observed that the setting of these weights was not very sensitive. Since these weights are relative to each other, this only represents one parameter.

Table 3

Summary of tracking results for all datasets. Overall tracks, mitosis events, move events, and appear/disappear events are shown separately. A mitosis error occurs if a cell splitting event is missed. Two move errors occur if two cell tracks switch identities. Two appear/disappear errors occur if a cell moves close to the boundary of the image and is detected as leaving and then entering as a new cell. Any of these errors will invalidate the track so that the number of correct tracks in the first column includes only tracks with no errors.

	Tracks		Mitosis		Move		Appear/disappear	
	Ratio	%	Ratio	%	Ratio	%	Ratio	%
1	2157/2177	99.1	95/96	99.0	35,170/35,183	100.0	2182/2188	99.7
2	3454/3471	99.5	132/135	97.8	26,607/26,619	100.0	4949/4951	100.0
3	166/171	97.1	33/33	100.0	8375/8379	100.0	72/73	98.6
4	244/258	94.6	20/20	100.0	12,032/12,043	99.9	169/172	98.3
5	1733/1737	99.8	34/37	91.9	14,008/14,008	100.0	747/748	99.9
Total	7754/7814	99.2	314/321	97.8	96,192/96,232	100.0	8119/8132	99.8

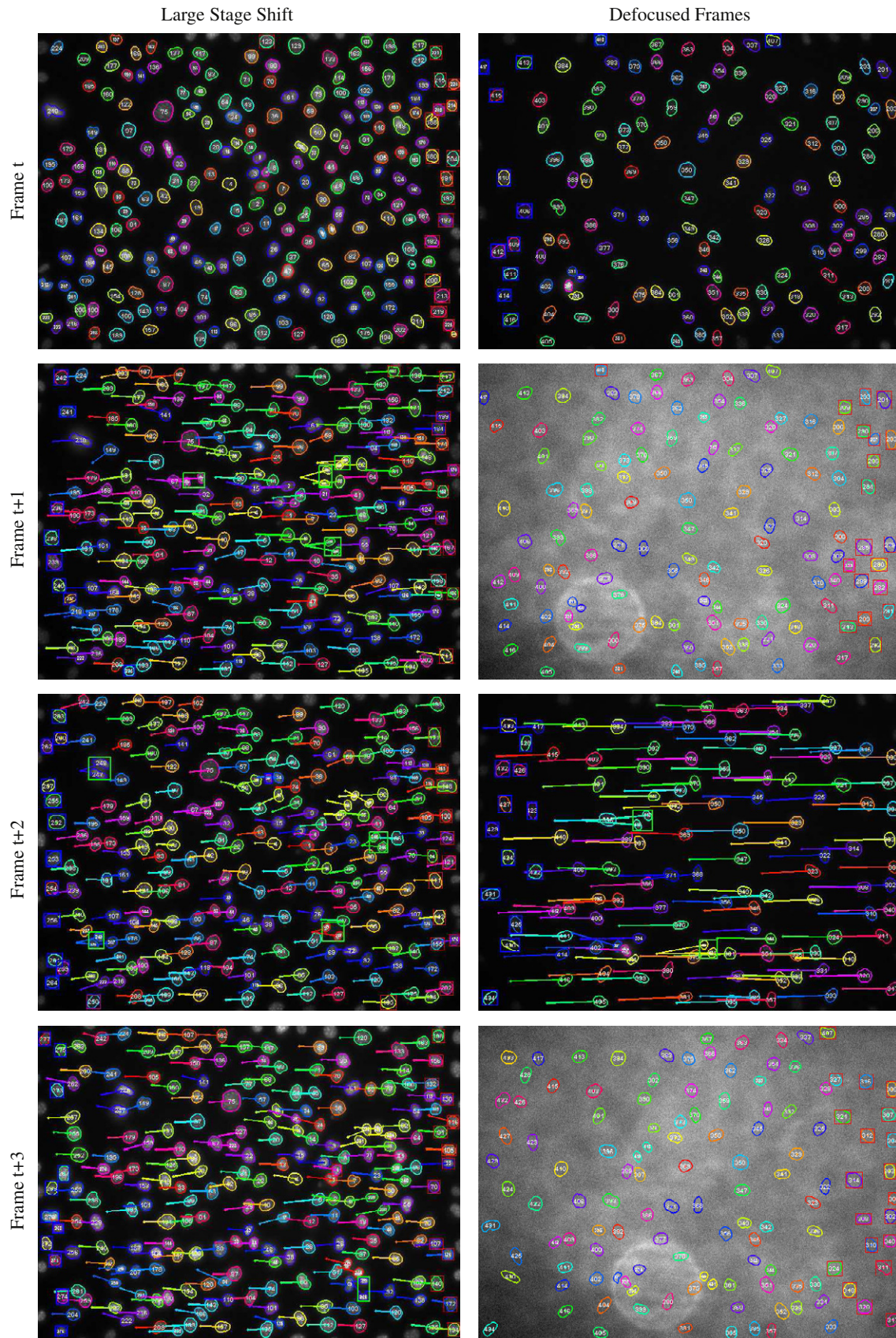


Fig. 9. Tracking results in the presence of large stage shift and defocused frames. On the left are four consecutive frames from one sequence that include large stage shift. On the right are consecutive frames from a sequence that includes defocused frames. The motion vectors show the previous location of the cells, and the segmented contours are overlaid. Blue boxes indicate cells entering, red boxes indicate cells leaving, and green boxes indicate cells splitting. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4

Summary segmentation validation results for all datasets. This table represents validation of over 104,000 cells. The total values for each of the five datasets are shown on the five rows. The true positives (TP) are the number of cells correctly segmented out of the total given by cell count. The false positives (FP) are the number of cells that were over-segmented (broken into pieces) and segmented objects that were actually noise. The false negatives (FN) are the number of cells that were under-segmented and cells that were missed. The precision is defined as $TP/(TP+FP)$. The recall is defined as $TP/(TP+FN)$. The last column reports the Kappa Index region overlap score mean μ_{KI} and standard deviation σ_{KI} for a random sample of cells in each of the five datasets. The Kappa Index is defined as $KI = (2 \times |A \cap M|)/(|A|+|M|)$, where A is the mask generated manually, and M is the mask generated automatically. The KI score demonstrates that the automatic segmentation masks have high overlap with the manual segmentations for all of the datasets, ranging from 0.81 to 0.96.

Dataset	Cells	TP	FP	FN	Precision (%)	Recall (%)	Area overlap ($\mu_{KI} \pm \sigma_{KI}$)
1	37,360	37,360	10	0	99.97	100.00	0.90 ± 0.05
2	30,090	30,087	4	3	99.99	99.99	0.94 ± 0.01
3	8550	8542	5	8	99.94	99.91	0.81 ± 0.04
4	12,301	12,285	5	16	99.96	99.87	0.88 ± 0.03
5	15,745	15,739	0	6	100.00	99.96	0.96 ± 0.01
Total	104,046	104,013	24	33	99.98	99.97	N/A

5.1. Tracking results

The nearly 400,000 cells resulted in nearly 32,000 tracks, of which nearly 8000 were validated as demonstrated in Table 3. In this table, the number of correctly detected mitosis, move, and appear/disappear events is also reported. For tracking validation, the user looked through the images and marked three types of er-

rors: (1) a mitosis error occurs if a cell splitting event is missed, (2) two move errors occur if two cell tracks switch identities, (3) two appear/disappear errors occur if a cell moves close to the boundary of the image and is detected as leaving and then entering as a new cell. Any of these errors will invalidate the track so that the number of correct tracks in the first column includes only tracks with no errors.

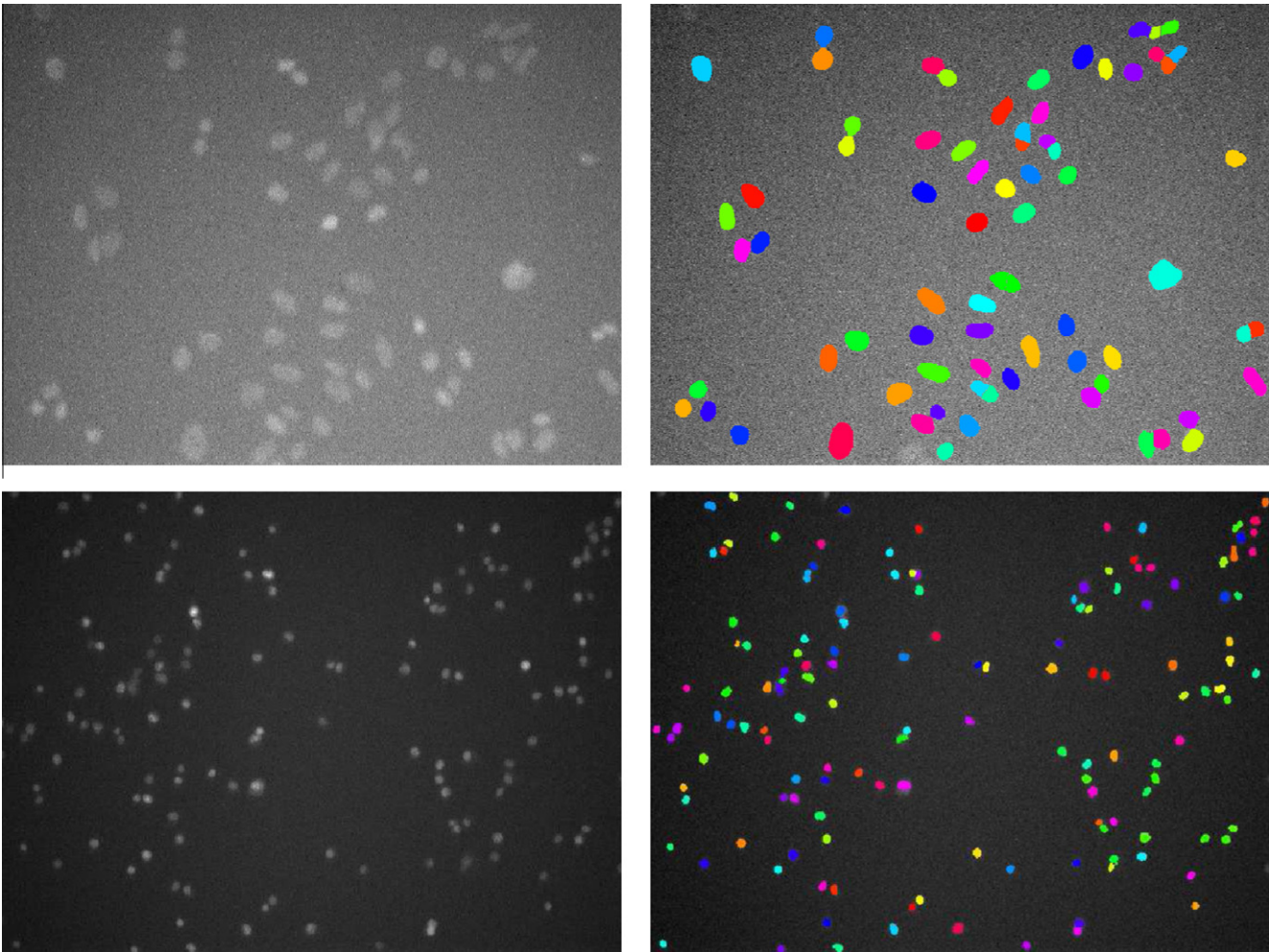


Fig. 10. Sample segmentation results. The two left images were acquired at different resolutions, and the right images show the corresponding segmentation results. The images in these experiments have low contrast-to-noise because of a low dose concentration. In addition, the flat-field effect from the lamp causes the interior to look brighter than the edges of the image. Although the cells vary in intensity and size and overlap partially, the algorithm is able to accurately segment all of the cells and does not require any post-processing.

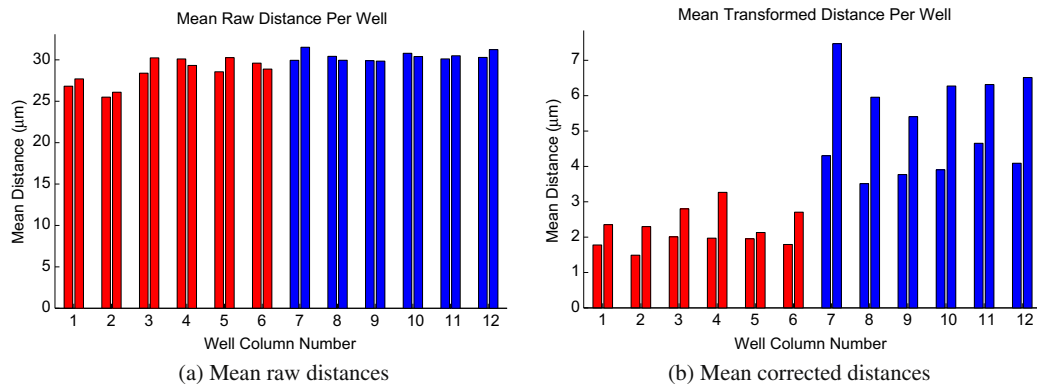


Fig. 11. Mean raw and transformed distances for dataset 2. This experiment consisted of 24 wells arranged in a matrix with 2 rows and 12 columns. The first 6 columns of wells were control wells (shown in red), and the last six columns contain a motility-promoting serum that causes cells to move faster (shown in blue). In the plots, each group of two wells corresponds to a column of the setup matrix. Each bar shows the mean distance moved for all cells in a well. In the left plot, the large stage shift obscures the actual cell movement, leading to statistically insignificant differences in the means of the wells. The right plot shows the corrected distances with the stage shift removed, leading to accurate cell distance measurements, and the two classes of cell distances can be clearly distinguished and are statistically different. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Our results indicate that the accuracy of the algorithms to correctly detect mitosis, move, and appear/disappear events is 97.8%, 100%, and 99.8% overall, respectively. For each track, if any event is incorrect, the track is considered invalid. Despite this stringent requirement, the percentage of correct tracks is 99.2%, indicating the accuracy and robustness of the algorithms. The robustness of the algorithms is further demonstrated by their ability to track through defocused images and large stage shift. Datasets 1, 2, and 5 contained wells with defocused images, all of which were detected by the defocus detection algorithm resulting in a total of 0 missed detections and 5 false positives out of nearly 6,000 images. Furthermore, the stage shift was correctly estimated for all of the images, leading to correct tracking results even when the actual cell movement was many times smaller than the movement of the stage.

Several example tracking results are given in Fig. 9. The track tails indicate the location of the cell in the previous image, demonstrating that there is often no overlap of the cells between adjacent images. The blue boxes indicate cells entering the image, red boxes indicate cells leaving, and green boxes indicate mitosis events. Although the cells move in clusters, touch one another, and undergo mitosis, the algorithms are able to successfully separate the cells into individual tracks. Because of the accurate cell segmentation, the algorithms can extract not only the tracks, but also the shape of the cells over time as shown in the figure.

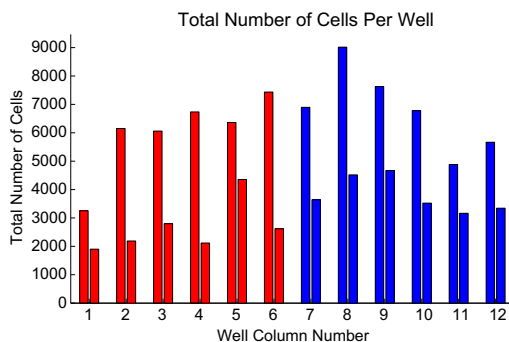


Fig. 12. Number of cells per well for dataset 2. The number of cells in the first bar of each pair are consistently greater than those in the second bar, by a factor of approximately 2. This explains the difference in cell movement in each pair of Fig. 11b: the more cells present in a well, the more the movement is restricted.

In summary, since the algorithms solve a global association problem with models for the different types of cell behavior, they are able to track cells that undergo mitosis, merge, move in and out of the image, and move quickly and erratically.

5.2. Segmentation results

Nearly 400,000 cells were segmented overall, and over 104,000 were validated as demonstrated in Table 4. Fig. 10 shows some example segmentation results.

Our main concern for segmentation is avoiding over-segmentation or under-segmentation meaning the avoidance of the following: missing cells, finding non-existent cells, breaking a cell into smaller pieces, and detecting multiple cells as one cell. Therefore, for segmentation validation, the user looked through the images and marked four types of errors: (1) over-segmented cells where a cell was broken into pieces by the algorithm, (2) segmented cells that were actually noise, (3) under-segmented cells where multiple cells were segmented as one, and (4) cells that were missed. The false positives (FP) are the number of cells that were over-segmented and segmented objects that were actually noise. The false negatives (FN) are the number of cells that were under-segmented and cells that were missed. The true positives (TP) are the number of cells correctly segmented out of the total given by cell count. With these numbers, the precision is defined as $TP/(TP + FP)$, and the recall is defined as $TP/(TP + FN)$.

For these validated cells, there was a total of only 24 false positives and 33 false negatives, leading to an overall precision and recall are 99.98% and 99.97%, respectively. These high accuracy rates and indicate the accuracy of the wavelet segmentation approach and its robustness across different kinds of datasets with varying levels of contrast-to-noise.

The segmented contours were also validated for a sample of randomly selected cells. Our segmentations yielded shapes very close to the correct segmentations, and we found that small errors in the borders of the segmented cells do not affect the statistics of the biological measurements. The cells were manually segmented, and the resulting masks were compared to the automatic segmentations using the Kappa Index (Zijdenbos et al., 1994) overlap score

$$KI = 2 \times \frac{|A \cap M|}{|A| + |M|} \quad (11)$$

where A is the automatic segmentation mask, M is the manual segmentation mask, and the score ranges from 0 (no overlap) to 1 (perfect overlap). The manual segmentation masks were generated using the livewire algorithm (Mortensen and Barrett, 1995) in which the user sequentially chooses points along the contour to segment and consecutive points are joined by the minimal cost path based on some function of the image.

The mean μ_{KI} and standard deviation σ_{KI} of KI are presented for a random sample of cells in each of the five datasets as the last column of Table 4. μ_{KI} for all of the datasets ranges from 0.81 to 0.96, which roughly correlates with the contrast-to-noise (CNR) ratios of the datasets: the dataset 3 has the lowest CNR, followed by dataset 4, with dataset 5 having the highest CNR. This correlation is due to the fact that both the manual and automatic segmentation are poorer for low-CNR images, resulting in less overlap of the segmentation masks.

5.3. Example biological results

The biological goal of one of the datasets is to automatically measure the difference in motility of cells in control wells from those containing a serum that promotes cell motility. Fig. 11a shows the plots of the mean cell movement per well without correcting for stage shift, and Fig. 11b shows the result of correcting for stage shift. The large stage shift in this experiment overwhelms the actual cell movement in Fig. 11a, leading to statistically insignificant differences in the means of the datasets based on a two-tailed t-test of whether the mean of two populations are statistically significant. Thus, without taking into account the stage shift, this dataset is not useful for measuring changes in cell velocity in the presence of the serum. However, Fig. 11b, which shows the cell distance measured with the stage shift removed, clearly shows the discrimination of the two classes of wells by a factor of approximately 3, and a two-tailed t-test reports a statistically significant difference.

It is evident in Fig. 11b that for each pair of bars the movement is nearly twice as large for cells in the second bar despite the fact that the treatment of the cells is the same. Fig. 12, showing the number of cells per well, helps to explain this phenomenon. This figure shows that the number of cells in the first bar is consistently about twice that of the number of cells in the second bar. The increased number of cells restricts the movement of the cells much like increased traffic on roads restricts the movement of individual cars. This demonstrates that this effect should be taken into account during experimental setup so as not to inadvertently bias the resulting measurements.

6. Conclusions and future work

Automated analysis of high-throughput time-lapse data can provide statistically meaningful measures that are difficult to achieve by manual analysis. This paper presented a generalized tracking framework that enables the automatic tracking, segmentation, and measurement of cells under various image conditions. Whereas many previous approaches to cell tracking are complex and require many parameters and significant post-processing, our approach is general, consistent, and extensible and operates in a graph-theoretic framework. This general approach avoids the need for specific models, parameter tuning, and training and is therefore not limited to tracking cells of a particular type or stained in a particular way. Indeed it can even be used for wholly different applications in various medical imaging modalities, surveillance applications, and industrial inspection.

In this framework, we introduced a tracking approach based on the minimum-cost flow algorithm that embeds the association costs in a weighted directed graph. We presented an efficient edge

coupling approach for handling splitting and merging events, and we have shown how to appropriately set the costs and capacities to represent cell behaviors such as splitting, merging, moving, and entering and leaving the field of view. Our framework includes a segmentation module that uses wavelets to effectively segment cells even in images with low contrast-to-noise. Our stage shift correction algorithm automatically detects microscope stage shift to assist accurate tracking and reduce the bias introduced by such effects in the measurement of cell velocity. The introduced defocus algorithm effectively separates images into the categories of normal and defocused, enabling the tracking algorithms to discard the defocused images.

These algorithms have been applied to nearly 6000 images representing 400,000 cells and 32,000 tracks. A large portion of the results (25%) were validated manually by expert biologists at two different institutions, and their edits were tabulated and stored to enable reproducibility of their assessment. The validation indicated that the algorithms are able to track the cells with an accuracy of 99.2% under the strict requirement that a track is considered failed if any errors are present in the track. The segmentation was found to have an overall precision and recall of 99.98% and 99.97%, respectively. The results demonstrate quantitatively the effectiveness of each of the modules and provide quantitative measurements of biological hypotheses.

For future work, we plan to apply the algorithms to a larger number of datasets to enable the large-scale study of biological hypotheses that are impractical to study through manual analysis. We also plan to expand the tracking model to incorporate more than two time points in the association, which should improve results when cells merge and then separate later in the sequence. While theoretically attractive, such an approach may not be necessary because of the already high accuracy of the system and may be intractable due to a significant increase in computational complexity.

Acknowledgments

We would like to thank Elizabeth Roquemore, Angela Williams, and Alla Zaltsman for generating the image sets and assisting with result validation; Paulo Mendonça for valuable technical discussions; and Gary Schools from the Ordway Research Institute for assisting with the result validation.

Appendix A. Computational complexity and complexity reduction

In this section, we analyze the complexity of solving the minimum-cost flow problem. In practice, there are many variants of solutions to this problem, each of which has different computational complexity. A well-known and common approach is based on successive flow augmentations along a minimal-cost path determined with Dijkstra's method, and we will use this method to describe the complexity of the algorithm.

Let $|V|$ be the number of vertices, $|E|$ be the number of edges with $|V| < |E| < |V|^2$, and d be the required flow. To solve the minimum-cost flow problem, Dijkstra's algorithm must be applied d times to augment the flow. A simple implementation of Dijkstra's algorithm has a complexity of $O(|V|^2)$, but using a binary heap reduces the complexity to $O((|E| + |V|)\log|V|) = O(|E|\log|V|)$. Thus, the complexity of the minimum-cost flow algorithm is

$$O(d|E|\log|V|) \quad (\text{A.1})$$

Next, we describe the complexity in more detail for including the different cell behaviors in the model. In all of the analysis that follows, let M be the number of left vertices (cells on the previous

frame) and N be the number of right vertices (cells on the next frame). For complexity analysis, these two numbers are of the same order, so we can arbitrarily use M to represent both.

For simple bipartite matching (see Fig. 1), the M left and N right vertices are fully connected leading to $M \times N$ edges. There are two additional vertices, the source and sink, and these add $M + N$ edges to the graph. Therefore, $|V| = (M + N) + 2 = O(M)$ and $|E| = (M \times N) + (M + N) = O(M \times N)$. The required flow is $d = \min(M, N) = O(M)$. Therefore, the complexity in this case is

$$O(M(M \times N) \log(M)) = O(M^3 \log(M)) \quad (\text{A.2})$$

When the appear and disappear vertices are added, the required flow becomes $d = M + N = O(M)$ as described in Section 3.2. Here, we add two vertices and $M + N + 3$ edges (see Fig. 2). Therefore, $|V| = (M + N) + 2 + 2 = O(M)$ and $|E| = (M \times N) + (M + N) + (M + N + 3) = O(M \times N)$. The complexity becomes

$$O(M(M \times N) \log(M)) = O(M^3 \log(M)) \quad (\text{A.3})$$

Thus, adding these vertices does not affect the overall complexity of the algorithm.

When splitting and merging behaviors are modeled, many more vertices and edges are added. For splitting, $\binom{N}{2} = \frac{N!}{(N-2)!(2)!} = \frac{(N)(N-1)}{2} = O(N^2)$ vertices are added, each of which has $(M + 1) + 2 = O(M)$ edges for a total of $O(N^2 \times M)$ edges (see Fig. 3(c)). Analogously, for merging $O(M^2)$ vertices and $O(M^2 \times N)$ edges are added. The total number of vertices in the graph is then $O(M + N^2 + M^2) = O(M^2)$ and the total number of edges is $O(M \times N + N^2 \times M + M^2 \times N) = O(M^3)$. The required flow d is still $M + N = O(M)$. Thus, the resulting complexity becomes

$$O(M(M^3) \log(M^2)) = O(M^4 \log(M^2)) \quad (\text{A.4})$$

It is clear that this adds significant complexity to the problem, so we next describe methods of pruning the edges and vertices that reduces this complexity.

To reduce the complexity of the coupled minimum-cost flow algorithm without reducing accuracy, we can prune both the vertices and edges of the graph by removing edges with the lowest cost. For the movement behavior, for each cell i in frame t , we calculate the cost of matching to all N cells j in frame $t + 1$ and then retain only the αN edges with lowest match cost calculated using Eq. (2). The number of mitotic candidates (graph vertices) $\binom{n}{2}$

in frame $t + 1$ is first limited to a fraction β of the N cells with lowest cost, where the cost of a pair of cells corresponding to a mitosis event is defined by the distance of the cells from each other normalized by their size and radius. The justification is that no more than βN cells at any given time will be splitting, which is $O(N)$ instead of $O(N^2)$ if all candidates were considered. Then, for each cell i in frame t , we calculate the cost of matching to each of the mitotic candidate pairs $j_1 j_2$ in frame $t + 1$ and retain only a fraction α of the edges with lowest cost as defined by Eq. (2). The pruning is analogous for merging cells. No pruning is done for the appear and disappear edges since the number is already small. It is also possible to completely remove the source and target vertices and all corresponding edges by making each of the left and appear vertices sources and each of the right and disappear vertices targets; although this reduces the size of the problem, it does not affect the overall computational complexity.

Thus, for moving cell candidates, the number of edges is limited to $\alpha(M \times N)$; for splitting cell candidates, the number of edges is limited to $\alpha(M \times \beta N)$; for merging cell candidates, the number of edges is limited to $\alpha(\beta M \times N)$. Although the use of the α and β result in significant reductions in the size of the problem, they do not affect the overall computational complexity. However, the pruning results in

the reduction of the number of splitting and merging edges from $O(M^3)$ to $O(M^2)$ and the number of vertices from $O(M^2)$ to $O(M)$. Thus, the computational complexity of the pruned graph is

$$O(M(M^2) \log(M)) = O(M^3 \log(M)) \quad (\text{A.5})$$

which is equivalent to the computational complexity of the simple bipartite graph. The value of M , representing the number of cells in an image, changes depending on the dataset. As defaults, we set $\alpha = 0.5$ and $\beta = 0.2$.

Appendix B. Supplementary data

Supplementary data associated with this article can be found, in the online version, at [doi:10.1016/j.media.2010.07.006](https://doi.org/10.1016/j.media.2010.07.006).

References

- Adiga, P.S.U., Chaudhuri, B.B., 2001. An efficient method based on watershed and rule-based merging for segmentation of 3-D histo-pathological images. *Pattern Recognition* 34 (7), 1449–1458.
- Al-Kofahi, O., Radke, R.J., Goderie, S.K., Shen, Q., Temple, S., Roysam, B., 2006. Automated cell lineage construction: a rapid method to analyze clonal development established with murine neural progenitor cells. *Cell Cycle* 5 (3), 327–335.
- Al-Kofahi, O., Radke, R., Roysam, B., Banker, G., 2006. Automated semantic analysis of changes in image sequences of neurons in culture. *IEEE Transactions on Biomedical Engineering* 53 (6), 1109–1123.
- Ancin, H., Dufresne, T.E., Ridder, G.M., Turner, J.N., Roysam, B., 1995. An improved watershed algorithm for counting objects in noisy, anisotropic 3-D biological images. *Proceedings of International Conference on Image Processing*, vol. 3. IEEE Computer Society, p. 3172.
- Arulampalam, S., Maskel, S., Gordon, L., Clapp, T., 2002. A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing* 50 (2), 174–188.
- Bar-Shalom, Y., Fortmann, T.E., 1988. *Tracking and Data Association*. Academic Press.
- Blake, A., Isard, M., 1998. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision* 28 (1), 5–28.
- Bonneau, S., Dahan, M., Cohen, L.D., 2005. Single quantum dot tracking based on perceptual grouping using minimal paths in a spatiotemporal volume. *IEEE Transactions on Image Processing* 14 (9), 1384–1395.
- Bunyak, F., Palaniappan, K., Nath, S., Baskin, T., Dong, G., 2006. Quantitative cell motility for in vitro wound healing using level set-based active contour tracking. In: *Proceedings of IEEE International Symposium on Biomedical Imaging*, pp. 1040–1043.
- Chan, T., Vese, L., 2001. Active contours without edges. *IEEE Transactions on Image Processing* 10 (2), 266–277.
- Chen, X., Wong, S., 2005. Automated dynamic cellular analysis in high throughput drug screens. In: *IEEE International Symposium on Circuits and Systems*, vol. 5, pp. 4229–4232.
- Chen, X., Zhou, X., Wong, S.T., 2006. Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy. *IEEE Transactions on Biomedical Engineering* 53 (4), 762–766.
- Chen, Y., Ladi, E., Herzmark, P., Robey, E., Roysam, B., 2009. Automated 5-D analysis of cell migration and interaction in the thymic cortex from time-lapse sequences of 3-D multi-channel multi-photon images. *Journal of Immunological Methods* 340 (1), 65–80.
- Comaniciu, D., Ramesh, V., Meer, P., 2000. Real-time tracking of non-rigid objects using mean shift. *Proceedings of IEEE Computer Vision and Pattern Recognition*, vol. 2. Hilton Head Island, South Carolina, pp. 142–149.
- Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., 2001. *Introduction to Algorithms*, second ed. The MIT Press.
- Debeir, O., Van Ham, P., Kiss, R., Decaestecker, C., 2005. Tracking of migrating cells under phase-contrast video microscopy with combined mean-shift processes. *IEEE Transactions on Medical Imaging* 24 (6), 697–711.
- De Hauwer, C., Darro, F., Camby, I., Kiss, R., Van Ham, P., Decaestecker, C., 1999. In vitro motility evaluation of aggregated cancer cells by means of automatic image processing. *Cytometry* 36 (1), 1–10.
- de Solorzano, C.O., Malladi, R., Lelievre, S., Lockett, S., 2001. Segmentation of cell and nuclei using membrane related proteins. *Journal of Microscopy* 201, 1–13.
- Donoho, D.L., 1995. De-noising by soft-thresholding. *IEEE Transactions on Information Theory* 41 (3), 613–627.
- Donoho, D., Johnstone, I., 1994. Ideal adaptation via wavelet shrinkage. *Biometrika* 81, 425–455.
- Dufour, A., Shinin, V., Tajbakhsh, S., Guillen-Aghion, N., Olivo-Marin, J., Zimmer, C., 2005. Segmenting and tracking fluorescent cells in dynamic 3-D microscopy with coupled active surfaces. *IEEE Transactions on Image Processing* 14 (9), 1396–1410.
- Dzyubachyk, O., Niessen, W., Meijering, E., 2007. A variational model for level-set based cell tracking in time-lapse fluorescence microscopy images. In: *Proceedings of IEEE International Symposium on Biomedical Imaging*, pp. 97–100.

- Figueiredo, M., Nowak, R., 2001. Wavelet-based image estimation: an empirical Bayes approach using Jeffreys' noninformative prior. *IEEE Transactions on Image Processing* 10 (9), 1322–1331.
- Gelb, A. (Ed.), 1979. *Applied Optimal Estimation*. MIT Press.
- Genovesio, A., Liedl, T., Emiliani, V., Parak, W.J., Coppey-Moisano, M., Olivo-Marín, J., 2006. Multiple particle tracking in 3-D+t microscopy: method and application to the tracking of endocytosed quantum dots. *IEEE Transactions on Image Processing* 15 (5), 1062–1070.
- Harder, N., Bermúdez, F., Godínez, W., Ellenberg, J., Eils, R., Rohr, K., 2006. Automated analysis of the mitotic phases of human cells in 3D fluorescence microscopy image sequences. In: *Proceedings of Medical Image Computing and Computer Assisted Intervention*, pp. 840–848.
- Harder, N., Neumann, B., Held, M., Liebel, U., Erfle, H., Ellenberg, J., Eils, R., Rohr, K., 2006. Automated recognition of mitotic patterns in fluorescence microscopy images of human cells. In: *Proceedings of IEEE International Symposium on Biomedical Imaging*, pp. 1016–1019.
- He, W., Wang, X., Metaxas, D., Mathew, R., White, E., 2006. Cell segmentation for division rate estimation in computerized video time-lapse microscopy. In: *Proceedings of Microscopic Image Analysis with Applications in Biology Workshop*, pp. 55–59.
- Isard, M., MacCormick, J., 2001. BraMBLe: a Bayesian multiple-blob tracker. In: *Proceedings of International Conference on Computer Vision*, vol. 2, pp. 34–41.
- Kachouie, N., Fieguth, P., Ramunas, J., Jervis, E., 2006. Probabilistic model-based cell tracking. *International Journal of Biomedical Imaging*, 1–10.
- Kalman, R., 1960. A new approach to linear filtering and prediction problems. *Transactions of the ASME* 82 (D), 35–45.
- Kovalev, V., Harder, N., Neumann, B., Held, M., Liebel, U., Erfle, H., Ellenberg, J., Eils, R., Rohr, K., 2006. Feature selection for evaluating fluorescence microscopy images in genome-wide cell screens. In: *Proceedings of IEEE Computer Vision and Pattern Recognition*, pp. 276–283.
- Kuhn, H., 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2, 83–97.
- Ladi, E., Schwickert, T., Chtanova, T., Chen, Y., Herzmark, P., Yin, X., Aaron, H., Chan, S.W., Lipp, M., Roysam, B., Robey, E., 2008. Thymocyte–dendritic cell interactions near sources of ccr7 ligands in the thymic cortex. *The Journal of Immunology* 181 (10), 7014–7023.
- Lewis, J.P., 1995. Fast normalized cross-correlation. In: *Vision Interface*. Canadian Image Processing and Pattern Recognition Society, pp. 120–123.
- Li, K., Miller, E., Weiss, L., Campbell, P., Kanade, T., 2006. Online tracking of migrating and proliferating cells imaged with phase-contrast microscopy. In: *Proceedings of IEEE Computer Vision and Pattern Recognition Workshop*, pp. 65–72.
- Li, K., Chen, M., Kanade, T., 2007. Cell population tracking and lineage construction with spatiotemporal context. In: *Proceedings of Medical Image Computing and Computer Assisted Intervention*, pp. 295–302.
- Li, K., Miller, E.D., Chen, M., Kanade, T., Weiss, L.E., Campbell, P.G., 2008. Cell population tracking and lineage construction with spatiotemporal context. *Medical Image Analysis* 12 (5), 546–566.
- Li, K., Miller, E., Chen, M., Kanade, T., Weiss, L., Campbell, P., 2008. Computer vision tracking of stemness. In: *Proceedings of IEEE International Symposium on Biomedical Imaging*, pp. 847–850.
- Lin, G., Adiga, U., Olson, K., Guzowski, J., Barnes, C., Roysam, B., 2003. A hybrid 3-D watershed algorithm incorporating gradient cues and object models for automatic segmentation of nuclei in confocal image stacks. *Cytometry Part A* 56A (1), 23–36.
- Lin, G., Chawla, M., Olson, K., Guzowski, J., Barnes, C., Roysam, B., 2004. Hierarchical, model-based merging of multiple fragments for improved 3-D segmentation of nuclei. *Cytometry Part A* 63A, 20–33.
- MacCormick, J., Blake, A., 1999. A probabilistic exclusion principle for tracking multiple objects. In: *Proceedings of International Conference on Computer Vision*, vol. 2, pp. 572–578.
- Mallat, S., 1989. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11, 674–693.
- Mallat, S., 1997. *A Wavelet Tour of Signal Processing*. Academic Press.
- Mortensen, E.N., Barrett, W.A., 1995. Intelligent scissors for image composition. In: *SIGGRAPH '95*, pp. 191–198.
- Mukherjee, D., Ray, N., Acton, S., 2004. Level set analysis for leukocyte detection and tracking. *IEEE Transactions on Image Processing* 13 (4), 562–572.
- Nath, S., Palaniappan, K., Bunyak, F., 2006. Cell segmentation using coupled level sets and graph-vertex coloring. In: *Proceedings of Medical Image Computing and Computer Assisted Intervention*, pp. 101–108.
- Okuma, K., Taleghani, A., de Freitas, N., Little, J., Lowe, D., 2004. A boosted particle filter: multitarget detection and tracking. In: *Proceedings of European Conference on Computer Vision*, vol. 1, pp. 28–39.
- Olivo-Marín, J., 1996. Automatic detection of spots in biological images by a wavelet-based selective filtering technique. In: *Proceedings of International Conference on Image Processing*, pp. 1:311–1:314.
- Osher, S., Fedkiw, R., 2002. *Level Set Methods and Dynamic Implicit Surfaces*. Springer.
- Osher, S., Sethian, J., 1988. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics* 79, 12–49.
- Padfield, D., Miller, J., 2008. A label geometry image filter for multiple object measurement. *Insight Journal*, 1–13.
- Padfield, D., Rittscher, J., Sebastian, T., Thomas, N., Roysam, B., 2006. Spatio-temporal cell cycle analysis using 3D level set segmentation of unstained nuclei in line scan confocal fluorescence images. In: *Proceedings of IEEE International Symposium on Biomedical Imaging*, pp. 1036–1039.
- Padfield, D., Rittscher, J., Thomas, N., Roysam, B., 2008. Automated spatio-temporal cell cycle phase analysis based on covert GFP sensors. In: *Microscopic Image Analysis for Life Science Applications*. Artech Publishing House, pp. 295–316.
- Padfield, D., Rittscher, J., Roysam, B., 2008. Spatio-temporal cell segmentation and tracking for automated screening. In: *Proceedings of IEEE International Symposium on Biomedical Imaging*, pp. 376–379.
- Padfield, D., Rittscher, J., Roysam, B., 2008. Defocus and low CNR detection for cell tracking applications. In: *Proceedings of Microscopic Image Analysis with Applications in Biology Workshop*.
- Padfield, D., Rittscher, J., Roysam, B., 2009. Coupled minimum-cost flow cell tracking. In: *Proceedings of Information Processing in Medical Imaging*, pp. 374–385.
- Padfield, D., Rittscher, J., Thomas, N., Roysam, B., 2009. Spatio-temporal cell cycle phase analysis using level sets and fast marching methods. *Medical Image Analysis* 13 (1), 143–155.
- Porter, R., Canagarajah, C., 1996. A robust automatic clustering scheme for image segmentation using wavelets. *IEEE Transactions on Image Processing* 5 (4), 662–665.
- Ray, N., Acton, S., 2005. Data acceptance for automated leukocyte tracking through segmentation of spatiotemporal images. *IEEE Transactions on Biomedical Engineering* 52 (10), 1702–1712.
- Reddy, B.S., Chatterji, B.N., 1996. An FFT-based technique for translation, rotation, and scale-invariant image registration. *IEEE Transactions on Image Processing* 5 (8), 1266–1271.
- Reid, D., 1979. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control* 24 (6), 843.
- Rousson, M., Deriche, R., 2003. A variational framework for active and adaptive segmentation of vector valued images. In: *Proceedings of IEEE Workshop on Motion and Video Computing*, pp. 56–61.
- Rui, Y., Chen, Y., 2001. Better proposal distributions: object tracking using unscented particle filter. In: *Proceedings of IEEE Computer Vision and Pattern Recognition*, vol. 2, pp. 786–794.
- Sarti, A., de Solorzano, C.O., Lockett, S., Malladi, R., 2000. A geometric model for 3D confocal image analysis. *IEEE Transactions on Biomedical Engineering* 47, 1600–1609.
- Sbalzarini, I.F., Koumoutsakos, P., 2005. Feature point tracking and trajectory analysis for video imaging in cell biology. *Journal of Structural Biology* 151 (2), 182–195.
- Sethian, J., 1996. *Level Set Methods and Fast Marching Methods*. Cambridge University Press.
- Simon, D., 2006. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. John Wiley & Sons.
- Soille, P., 2003. *Morphological Image Analysis*, second ed. Springer-Verlag, Heidelberg.
- Sorenson, H.W., 1970. Least-squares estimation: from Gauss to Kalman. *IEEE Spectrum* 7, 63–68.
- van de Merwe, R., Doucet, A., de Freitas, N., Wan, E., 2000. The unscented particle filter. Technical Report CUED-F-INFENG/TR 380, Cambridge University Engineering Department, August 2000.
- Vermaak, J., Doucet, A., Perez, P., 2003. Maintaining multi-modality through mixture tracking. In: *Proceedings of International Conference on Computer Vision*, pp. 1110–1116.
- Wahlby, C., Sintorn, I.-M., Erlandsson, F., Borgefors, G., Bengtsson, E., 2004. Combining intensity, edge, and shape information for 2d and 3d segmentation of cell nuclei in tissue sections. *Journal of Microscopy* 215, 67–76.
- Welch, G., Bishop, G., 2004. An introduction to the Kalman filter, Technical Report, University of North Carolina at Chapel Hill, Department of Computer Science.
- Xie, J., Khan, S., Shah, M., 2008. Automatic tracking of escherichia coli bacteria. In: *Proceedings of Medical Image Computing and Computer Assisted Intervention*, pp. 824–832.
- Yang, F., Mackey, M., Ianzini, F., Gallardo, G., Sonka, M., 2005. Cell segmentation, tracking, and mitosis detection using temporal context. In: *Proceedings of Medical Image Computing and Computer Assisted Intervention*, pp. 302–309.
- Zhang, L., Li, Y., Nevatia, R., 2008. Global data association for multi-object tracking using network flows. In: *Proceedings of IEEE Computer Vision and Pattern Recognition*, pp. 1–8.
- Zhou, X., Chen, X., Liu, K., Lyman, S., King, R., Wong, S., 2006. Time-lapse cell cycle quantitative data analysis using Gaussian mixture models. *Life Science Data Mining*, vol. 2. World Scientific, pp. 17–46.
- Zijdenbos, A.P., Dawant, B.M., Margolin, R.A., Palmer, A.C., 1994. Morphometric analysis of white matter lesions in mr images: method and validation. *IEEE Transactions on Medical Imaging* 13 (4), 716–724.