

Automated and Semi-automated Cell Tracking: Addressing Portability Challenges

*Andrey Kan**, James Bailey, Christopher Leckie,
John Markham, Mark R. Dowling, Rajib Chakravorty

Abstract

Cell tracking is a key task in the high-throughput quantitative study of important biological processes, such as immune system regulation and neurogenesis. Variability in cell density and dynamics in different videos, hampers portability of existing trackers across videos. We address these portability challenges in order to develop a portable cell tracking algorithm. Our algorithm can handle noise in cell segmentation as well as divisions and deaths of cells. We also propose a parameter-free variation of our tracker. In the tracker, we employ a novel method for recovering the distribution of cell displacements. Further, we present a mathematically justified procedure for determining the gating distance in relation to tracking performance. For the range of real videos tested, our tracker correctly recovers on average 96% of cell moves, and outperforms an advanced probabilistic tracker when the cell detection quality is high. The scalability of our tracker was tested on synthetic videos with up to 200 cells per frame. For more challenging tracking conditions, we propose a novel semi-automated framework that can increase the ratio of correctly recovered tracks by 12%, through selective manual inspection of only 10% of all frames in a video.

*Published in Journal of Microscopy in 2011. The definitive version is available at www.blackwell-synergy.com

Andrey Kan, James Bailey, and Christopher Leckie are with Victoria Research Laboratory, National ICT Australia (NICTA), Department of Computer Science and Software Engineering, University of Melbourne, VIC, 3010, Australia.

Rajib Chakravorty and John Markham are with Victoria Research Laboratory, National ICT Australia (NICTA), Department of Electrical and Electronic Engineering, University of Melbourne, VIC, 3010, Australia.

Mark R. Dowling is with Immunology Division, The Walter and Eliza Hall Institute, 1G Royal Parade, Parkville, VIC, 3052, Australia.

Andrey Kan (akan@csse.unimelb.edu.au) is the corresponding author. His address is 3.25b, ICT Building, 111 Barry St, Carlton, VIC 3053, Australia.

1 Introduction

Automated cell tracking enables high throughput quantitative analysis of cell behavior (e.g., cell proliferation and motility), which can provide valuable insights into biological processes, such as immune system regulation (Nitschke et al., 2008) or neurogenesis (Al-Kofahi et al., 2006). Cell tracking involves segmenting cells from the video and associating observed cell occurrences between frames. These steps can be combined, but developing a dedicated association procedure usually supports a wider range of tracking conditions, such as abrupt cell motions (Padfield et al., 2010). In this paper, by cell tracking we mean the data association stage, which is the focus of our paper.

A major challenge for automated cell tracking is the substantial variation that cell videos can exhibit in terms of cell density and dynamics (Fig. 1). In practice, these variations can arise even within a single laboratory, due to different experimental conditions, e.g., a control group and a treated group. Even in the case of several videos with similar cell appearance, there can be a variation in cell density and dynamics. For such videos, one can use the same segmentation algorithm, but the variation still poses challenges for the tracking stage.



Fig. 1: Fragments of video frames used in this study. The images show B lymphocytes at different scales (left, middle), and neural progenitor cells (right, with permission from Al-Kofahi et al., 2006). There is a variation in cell dynamics: videos with B lymphocytes contain abrupt cell motions, while neural progenitor cells tend to have smooth motions. There is also a variation in cell density.

There has been substantial work in the field of cell tracking. However based on the previous approaches, Meijering et al. [2009] conclude that “the consensus arising from the literature seems to be that any specific tracking task requires dedicated (combinations of) algorithms to obtain optimal results”.

Indeed, it appears that a number of factors hamper the application of the previously proposed algorithms across videos: some algorithms rely on smooth cell motions, some algorithms use association costs specific to Brownian motion, some algorithms require the user to set parameter values, whereas the optimal values may differ across videos.

Our aim has been to minimize the amount of manual work needed to enable cell tracking across a variety of different videos. To this end, we propose two novel methods: an automated and a semi-automated cell tracking algorithm.

In the design of our automated tracker we addressed the factors that affect the portability of cell trackers. Specifically, we propose a parameter-free tracker.

Tab. 1: Common challenges that hamper applying tracking methods across different videos

Methods	Unaddressed Challenge
Evolving models (Adanja et al., 2010; Debeir et al., 2005).	abrupt cell motions
Interframe assignments (Al-Kofahi et al., 2006; Padfield et al., 2010).	cost function for a variety of conditions
Interframe assignments (Jaqaman et al., 2008); motion filters (Li et al., 2008).	avoid requiring prior knowledge
Most of the methods (Adanja et al., 2010; Chen et al., 2009).	manual post-processing

By parameter-free, we mean that our algorithm (in one of the versions) requires only the problem input (cell detection), and does not require any additional information from a user. Internally, our algorithm uses a parameter (gating distance), but this parameter is estimated automatically from the input data.

In this work, we found that using only cell centroid locations can be sufficient for achieving reasonably high tracking performance. However, as we argue in Section 4, other features, such as cell size or fluorescence can be seamlessly added to our tracker, while keeping the algorithm parameter-free.

Further, we note that even the best proposed algorithms make occasional tracking errors and may require a post-processing stage with manual error correction. Therefore we designed a semi-automated tracker for assisting this manual post-processing step, by selecting the frames for manual review.

1.1 Related work and our method

We can categorize previously proposed cell tracking methods into two groups. Methods in the first group try to fit models of cells tracked in previous frames to a new frame (Adanja et al., 2010; Debeir et al., 2005). Methods in the second group use assignments between cell detections found independently at each frame (Al-Kofahi et al., 2006; Padfield et al., 2010). Further, methods in both groups can use additional tools such as stochastic motion filters (Degerman et al., 2009; Li et al., 2008).

While there are many advantages of the proposed methods, we note that there is always an unaddressed challenge that affects the transferability of each particular method across videos. We summarize the challenges in Table 1.

We now introduce some of the methods from the first group. Adanja et al.

[2010] and Debeir et al. [2005] model cells with centroid locations and follow cell movements using variations of the mean-shift procedure. Dufour et al. [2005] and Dzyubachyk et al. [2010] use active contours and level-set based methods. Inglis et al. [2010] propose an adaptation of segmentation by weighted aggregation technique for clustering, while Padfield et al. [2008] propose clustering spatio-temporal “tubes”, where a “tube” is constructed from segmented cell contours in subsequent frames. One can argue for categorizing the latter two methods into a separate group, but we leave them in the first group for brevity.

Methods from the first group allow the combination of cell detection and tracking stages, but these methods rely on smooth cell motions (Debeir et al., 2005; Padfield et al., 2010). Methods from the second group focus on the assignment stage for previously detected cells. There are various approaches proposed for assigning cell locations between frames.

Al-Kofahi et al. [2006] use an integer programming framework; Bonneau et al. [2005] treat measurements from all frames as points in a spatio-temporal volume, and iteratively link the closest points in this volume; Chowdhury et al. [2010] employ a graph matching formulation; Jaqaman et al. [2008] state the task as a two-stage linear assignment problem; Kirubarajan et al. [2001] start with two frame assignment and iterate over several frames modifying the assignment costs; and Padfield et al. [2010] propose an extension of a minimum-cost flow algorithm.

Some of the methods (Chowdhury et al., 2010; Kirubarajan et al., 2001; Sbalzarini and Koumoutsakos, 2005) do not directly handle the cases of cell divisions and deaths, hence requiring an adaptation for the cell tracking problem. Other assignment methods (Bonneau et al., 2005; Jaqaman et al., 2008) either do not give an explicit form of their assignment cost function, or derive the costs assuming Brownian motion of cells. However, we found that in most of our real videos, cells are unlikely to follow a Brownian motion model (Section 3.1). We develop our cell tracking algorithm under a relaxed assumption on cell dynamics. Moreover, the method of Jaqaman et al. [2008], requires choosing two cost functions, one for divisions and one for moves, whereas in our method only one function is required.

Al-Kofahi et al. [2006] and Padfield et al. [2010] propose different cost functions for links corresponding to different events (e.g., cell moves, cell divisions), and give explicit forms of these functions (Gaussian functions and absolute differences). However, it is not obvious why these particular forms should be used and what is the set of tracking conditions for which these functions are effective. In particular, the choice of the cost function for division events is non-trivial. In contrast, in our paper, we provide a systematic argument for using only one cost function. We also state the assumptions that determine the possible choices of the cost function.

Some methods require prior knowledge in order to be effective. For example, a user can be asked to provide a gating threshold (Al-Kofahi et al., 2006; Jaqaman et al., 2008; Sbalzarini and Koumoutsakos, 2005). This affects the portability of the methods across different videos, because, in general, the opti-

mal threshold value needs to be estimated for each new video. In our work, we propose a technique for an *automated* estimation of the threshold. Providing the gating threshold is not required in the methods of Mori et al. [1992] and Padfield et al. [2010], but these methods do not directly handle both types of noise in cell detection: missing detections and spurious measurements.

The cell motion model is another kind of prior knowledge. This is typically required by stochastic motion filters. Examples of such works include the method of Degerman et al. [2009] which uses a modified auction algorithm combined with Hidden Markov Models for predicting cell locations; Godinez et al. [2009] which combines interframe assignments with particle filters; and Li et al. [2008] which employs an active contours technique combined with motion prediction using interacting multiple models. Stochastic filters usually require the user to be specific about the cell motion model, e.g., by providing the state transition matrix for Kalman filtering. However, it is difficult to be specific about cell motions prior to tracking. Therefore, we focus on developing a method that works under a general assumption about cell motions.

In summary, there are challenges that affect the portability of cell trackers across videos. We propose an automated cell tracker that addresses these challenges. Our automated tracker uses nearest neighbor based interframe assignments and is called NENIA.

Automated cell tracking is a difficult task, therefore it is common to use a post-processing stage where errors of the automated tracking are corrected manually (Chen et al., 2009; Hamzic et al., 2008). Here the challenge is to identify the likely errors in tracking.

Recently there has been some work addressing this problem. Adanja et al. [2010] propose to look at the situations when two tracks come too close to each other, because it is likely that such tracks will be confused. Chen et al. [2009] propose to sort automatically constructed links according to their costs. The links with unusually high costs are likely to be incorrect. However, both methods cannot spot missing correct links. There is also no estimation of the effectiveness of the proposed methods. In our work, we present a novel semi-automated tracking method called SAMTRA, that allows selection of frames that are likely to require manual review. Moreover, we provide an evaluation of our semi-automated tracker.

We present our automated cell tracker in Sections 2.1 — 2.4, our semi-automated tracker in Section 2.5, an evaluation of both algorithms in Section 3, and further discussion of our methods and results in Section 4.

2 Methods

Informally, our first goal has been to develop an automated cell tracker that is (i) portable, i.e., easily transferable across a variety of videos, (ii) capable of handling cell divisions and deaths, and (iii) able to perform reasonably well, i.e., resolve the majority of cell moves between frames correctly.

In order to meet the portability requirement, we address the previously dis-

Tab. 2: The key elements of our work that are designed to address the portability challenges

Challenge	Solution	Details
abrupt cell motions	using assignment based method	Sections 2.1 and 2.2
cost function for a variety of conditions	mathematical framework that shows that one can use a single cost function, and which function to use	Sections 2.1 and 2.2
avoid requiring prior knowledge	method for an automated estimation of the optimal gating distance	Section 2.3
manual post-processing	method for an automated selection of frames for manual review	Section 2.5

cussed challenges (Table 1). In Table 2, we repeat these challenges and comment on how we have addressed them.

As it was discussed in the previous section, tracking methods based on evolving models and clustering of spatio-temporal volumes tend to rely on smooth cell motions. Therefore we use an assignment based method.

A novel aspect of our work is a mathematical framework that allows us to (a) demonstrate that using a single cost function for both cell movement and division events is sufficient, (b) show which cost function can be used, and (c) treat gating distance as an optimization criterion. Preliminaries and formal definitions are given in Section 2.1 and the framework is presented in Section 2.2.

Another novel aspect is a method for an automated estimation of the optimal gating distance presented in Section 2.3. The gating distance is a key variable of our tracker, and the automated estimation of its value makes it easier to transfer the algorithm between videos.

An overview of our cell tracker is presented in Fig. 2A. In addition to an interframe assignment method, our automated tracker employs a track management module that uses the assignment results to maintain (e.g., update or split) the set of cell tracks.

Our second goal has been to develop a method for selecting frames that are likely to contain tracking errors. The overview of our method, called SAMTRA, is presented in Fig. 2B, and details are given in Section 2.5. SAMTRA enables us to further improve tracking accuracy with minimal manual intervention.

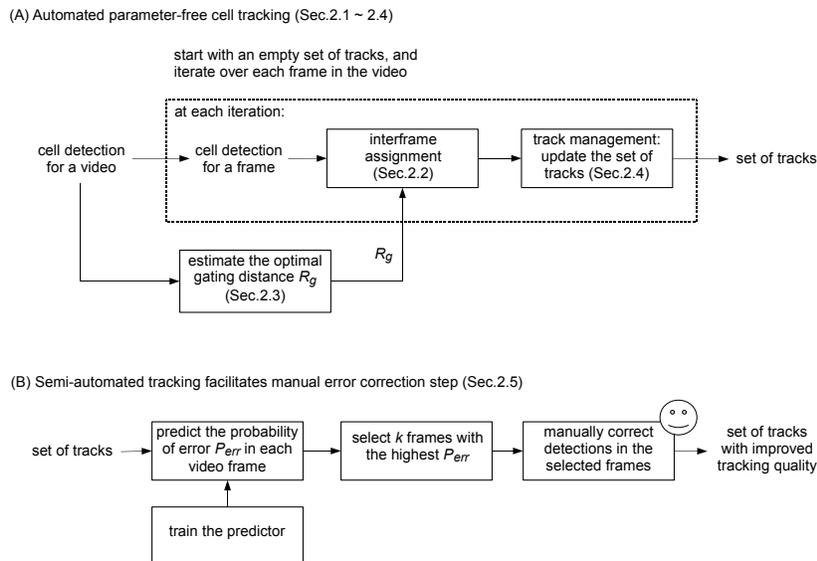


Fig. 2: Overview of our automated and semi-automated tracking methods. Details are given in the corresponding sections.

2.1 Cell Tracking Preliminaries

Consider a video of living cells. We characterize a cell by its location, i.e., its centroid position, in every video frame. A *cell occurrence* is the cell location in a certain frame. The video is first processed by a segmentation algorithm¹, which identifies the likely cell locations in each frame. We refer to each given estimate of a cell location as a *measurement* $m_{f,i} = \vec{x}_{f,i}$, where \vec{x} is a point in the Euclidean space, f is the frame number (when unambiguous, we will omit this subscript), and i is the measurement index within a frame. The cell occurrence may or may not have a corresponding measurement.

Between any two consecutive frames f and $f+1$, a cell can either *move*, *divide* or *die* (Fig. 3). In the case of a move, the cell persists in both frames (remaining still is characterized as a “move”), and $P(i \rightarrow j)$ denotes the probability that a cell moves from measurement $m_{f,i}$ to $m_{f+1,j}$. A *cell offset* is a vector pointing from the previous cell position to the next position, and a *cell displacement* is the magnitude of the cell offset.

In the cases of a division or a death, the cell ceases to exist in all frames

¹ For videos with lymphocyte cells we used the algorithm of Cheng et al. [2008] supplemented with manual post processing of cell detections (less than 30% of manual corrections). For the video with neural progenitor cells (that have significantly different appearance than lymphocytes) we did not have an implementation of a segmentation algorithm and did the segmentation manually. There is evidence that we could find a better automated segmentation algorithm for our videos (Al-Kofahi et al., 2006; Padfield et al., 2010). However, the segmentation is not the focus of our paper, we merely needed to collect the ground truth.

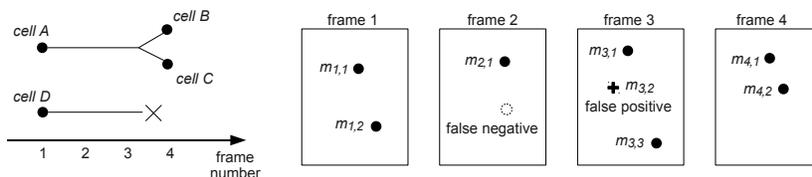


Fig. 3: Sample lineage trees (left) summarize cell events: cell A moves between frames 1, 2, and 3, and divides between frames 3 and 4; cell D moves between frames 1, 2, and 3, and dies between frames 3 and 4. In a sample detection resulting from these events (right), $m_{i,j}$ are measurements, where i is a frame index and j is a measurement index within a frame. In this example, the correct track for cell A is $\{m_{1,1}, m_{2,1}, m_{3,1}\}$, for cell B it is $\{m_{4,1}\}$, for cell C it is $\{m_{4,2}\}$, for cell D it is $\{m_{1,2}, ?, m_{3,3}\}$. There is a false negative error (cell D is not detected) in frame 2, and a false positive error (spurious measurement) in frame 3.

starting from $f + 1$. Also in the case of a division, the cell that ceases to exist is called *the mother cell*, and two new cells, called *daughter cells*, appear in frame $f + 1$.

For a given video, across all frames we define the following probabilities of a cell move P_m , cell division (split) P_s , and cell death P_d as

$$P_m = \frac{\# \text{occurrences when cell moves}}{\# \text{occurrences}}, \quad (1)$$

$$P_s = \frac{\# \text{occurrences when cell divides}}{\# \text{occurrences}}, \quad (2)$$

$$P_d = \frac{\# \text{occurrences when cell dies}}{\# \text{occurrences}}. \quad (3)$$

Here $\#$ occurrences is the total number of occurrences of all cells in the video (except the last frame), and $\#$ occurrences when cell moves (divides, dies) is the number of occurrences corresponding to cell moves (divisions, deaths). Note that $P_m + P_s + P_d = 1$.

The probabilities P_m , P_s , P_d , and $P(i \rightarrow j)$ characterize the cells. In general, these probabilities are unknown prior to tracking. Moreover, tracking is often used to derive these probabilities (Degerman et al., 2009; Hawkins et al., 2009).

In our tracker, we do not directly address cells entering and leaving the field of view. We observe that using small wells or microgrids [Day et al., 2008] is becoming increasingly common to address the problem of cells migrating into or out of the field of view. Further, in our tracker, entering the view is registered as a division (if a new cell enters somewhere close to another cell) or as a false positive (otherwise), and leaving the view is registered as a death. Therefore, if handling of entering and leaving cells is required, our tracker can be supplemented with an extra step to distinguish between cell divisions and

enterings, and cell deaths and leavings. This step can be based on the proximity to the frame border.

The cell segmentation algorithm extracts a set of measurements from all frames of the video. We refer to this set of measurements as a *detection*. Note that in practice, detections can contain errors: spurious measurements (*false positives*) and missed cell occurrences (*false negatives*). The quality of the cell detection can be measured by standard precision and recall metrics, or alternatively by the probability of false positives (P_{fp}) and the probability of false negatives (P_{fn}) defined as²

$$P_{fp} = \frac{\#fp}{\#tp + \#fp} = 1 - Precision, \quad (4)$$

$$P_{fn} = \frac{\#fn}{\#tp + \#fn} = 1 - Recall. \quad (5)$$

Here $\#fp$ and $\#fn$ are the numbers of false positive and false negative errors, and $\#tp$ is the number of cell occurrences that have the corresponding measurements in the detection.

The detection forms the input of the tracking problem. The output of the tracking problem is a set of tracks, where a *track* is a list of measurements from the subsequent frames. The track may contain *dummy measurements* that indicate that at a particular frame the track was not detected (there was a false negative error). The quality of the output is measured as explained below.

2.1.1 Cell Tracking Performance

There are two popular approaches to measuring the tracking performance. The first approach is based on the ratio of the correctly reconstructed links between consecutive frames [Degerman et al., 2009], while the second measures the ratio of the correctly reconstructed tracks [Li et al., 2008]. The two performance metrics are defined as follows.

Let a *correct track* be a track that includes all occurrences of the same cell and does not include any other measurements (the correct track may include dummy measurements). Each distinct cell in a video has a corresponding correct track.

We define the first performance measure based on the definition of a “mix-up” error given by Degerman et al. [2009]. A cell tracker produces a set of tracks where each track (not necessarily correct) has a unique non-zero identification number (ID). Then each measurement can be labeled by the ID of the containing track, or with 0, if the measurement is not included in any track (considered as

² Note that these are different from the standard false positive rate and false negative rate metrics. The standard metrics require the number of true negatives. A true negative in cell detection is a point in space that was not classified as a cell location and at which there is indeed no cell. However, there are infinitely many such points in space. If we limit ourselves to the pixel resolution, the number of such true negative points will depend on the resolution of a video and the definition of the frame borders. Therefore the concept of a true negative is impractical for cell detection.

a false positive). A *swap error* is a situation when the labels of two consecutive measurements in a correct track change. Let $\#all\ measurements$ be the number of measurements in all correct tracks. We then propose to measure the tracking performance by the ratio of correctly reconstructed links defined as

$$P_{links} = 1 - \frac{\#swap\ errors}{\#all\ measurements}. \quad (6)$$

Further, let $\#corr\ tracks$ be the number of correct tracks produced by a cell tracker, and $\#cells$ be the number of distinct cells in a video (we count all cells that have appeared in the video in at least one frame). We then use the second performance metric as proposed by Li et al. [2008],

$$P_{tracks} = \frac{\#corr\ tracks}{\#cells}. \quad (7)$$

Clearly, the two metrics are related: higher P_{links} generally implies a higher P_{tracks} and vice versa. P_{tracks} is harsher and less sensitive to tracking algorithm improvements, because, in a cell track, a single swap error leads to invalidation of the entire track. On the other hand, the P_{tracks} measure is more relevant in the case when correct reconstruction of the entire track is of paramount importance (e.g., in the analysis of Hawkins et al., 2009).

We use P_{links} in the development and evaluation of NENIA, because our cell tracker is designed to resolve as many correct moves as possible. By contrast, our SAMTRA algorithm can be used to manually improve the tracking results, when the subsequent data analysis is sensitive to the correct tracks recovery. Therefore, we use P_{tracks} for the evaluation of SAMTRA. Reporting both measures for both algorithms does not add much new information and is omitted in the interest of space.

Resolving cell divisions is critical for reconstructing lineage trees. A correctly resolved division is one for which the frame number when the division occurs, and both mother-daughter relations are established correctly. We measure the ability of NENIA to resolve cell divisions by using the ratio of correctly resolved divisions to the total number of divisions in a video. At the same time we emphasize that resolving cell moves is also critical for reconstructing lineage trees. For example, a swap error can result in a loss of two cell tracks, and further in an inability to construct lineage trees for these cells. Moreover, note that in the videos, the number of cell moves tends to be much larger (often by several orders of magnitude) than the number of divisions (Hawkins et al., 2009; Li et al., 2008; Padfield et al., 2010). Therefore P_{links} alone can give an estimation of how useful NENIA can be in practice, although we also report the ratio of resolved divisions for NENIA.

2.1.2 Assumptions

We conclude our cell tracking preliminaries with a list of assumptions. While the assumptions determine the scope of our work, we also provide comments explaining why our work is applicable to a wide range of real world situations.

Assumption 1. Probabilities of cell and detection events. *We assume that the cell video has a good temporal resolution, i.e., the time between successive frames is small in comparison to a cell lifetime. Specifically, we assume that $P_s + P_d \leq 0.1$ (here P_s and P_d are defined by equations 2 and 3). This assumption means that there are on average at least 10 frames per cell lifetime. The videos reported in the literature satisfy this property (Al-Kofahi et al., 2006; Hawkins et al., 2009).*

We also assume a reasonable cell detection quality, specifically $P_{fp} + P_{fn} \leq 0.1$. In practice, researchers have reported detections with $P_{fp}, P_{fn} < 0.03$ (Li et al., 2008; Padfield et al., 2010).

If the above assumption does not hold, it is hard to expect an accurate cell data analysis regardless of the quality of tracking. Poor temporal resolution implies a coarse estimation of cell lifetimes and speeds. Poor detection quality leads to poor tracking quality (Jaqaman et al., 2008; or our evaluation in Section 3.2).

Assumption 2. Independence. *In any given frame, the decisions whether to move, divide or die are made independently for different cells. In the case of a move, the direction and the cell displacement occur independently for different cells.*

Although there is likely to be a correlation in the lifetimes and fates of cells, under Assumption 1 the time between frames is much shorter than a cell lifetime. On this short time scale, the behavior of cells is more stochastic, and the independence assumption is reasonable.

Assumption 3. Cell dynamics. *(i) The probability $P(i \rightarrow j)$ of a cell move between locations \vec{x}_i and \vec{x}_j decreases according to the squared Euclidean distance $\|\vec{x}_i - \vec{x}_j\|^2$ between the locations. (ii) The spatial distribution of cells is similar across consecutive frames.*

The first part of this assumption is applicable to a number of distributions that are likely to describe the cell dynamics. In particular, normal-like distributions are prominent in nature. Gomez et al. [1998] introduce the multivariate power exponential distribution as a generalization of the normal distribution. This generalized normal distribution satisfies our assumption (see Appendix A for mathematical details).

Notice that Assumption 3 allows us to develop a parameter-free tracker, without an explicit specification of the distribution of cell offsets. Moreover, our experiments (Section 3.2) show that the assumption does not affect the applicability of NENIA across real cell videos with different distributions of cell displacements. Our cell tracker performs well on each of these videos.

The second part of Assumption 3 is related to cell density. If the directions of cell moves are random, we can expect cell density to remain approximately the same between consecutive frames. If there is a prevalent direction, we can expect the whole group of cells to shift in one direction, again, preserving the density. Finally, if there are subpopulations of cells with different prevalent directions,

we still expect them to keep the spatial distribution between consecutive frames, because under Assumption 1, the time between frames is relatively short.

The key step of NENIA is interframe assignments. This step is described in the next section.

2.2 Interframe Assignment Problem

In this section, we describe our novel approach to the assignment problem in the context of cell tracking. In this context, the assignment problem has already been described in the literature (Jaqaman et al., 2008; Kirubarajan et al., 2001). We first re-introduce some of the related definitions, and then provide the details of our new approach.

Suppose that at frame f we have a set of previously established tracks $T_f = \{t_i\}$, where a track t_i has the location $m_{f,i}$. Further, suppose that at frame $f + 1$ we have a set of measurements $M_{f+1} = \{m_{f+1,j}\}$.

Consider assigning measurements in M_{f+1} to tracks in T_f such that a measurement m can be assigned to 0 or 1 tracks, and 0, 1, or 2 measurements can be assigned to a track t . Fig. 4 explains the choice of this assignment setup with an example. We refer to an individual assignment of measurement j to track i as a *link* $i - j$. The link $i - j$ represents a potential cell offset from the location $m_{f,i}$ to the location $m_{f+1,j}$. The *length of a link* $i - j$ is the Euclidean distance $r_{ij} = \|\vec{x}_i - \vec{x}_j\|$ between the locations of m_i and m_j . Where unambiguous, we use the term link to denote the length of a link.

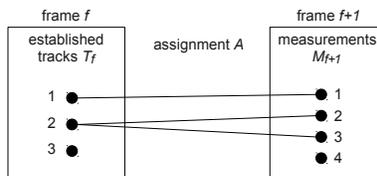


Fig. 4: A sample assignment between two consecutive frames. Measurements are assigned to previously established tracks. In the assignment, there are four possible combinations that represent different cell events, such as moves, divisions, deaths and noise in cell detections.

We refer to a set of tracks, measurements, and established links between them as an *assignment*. Our task is to find the assignment that corresponds to the true set of cell events that occur between frames f and $(f + 1)$. A popular approach for solving this kind of task is based on Bayesian inference. Here the likelihood of an assignment is expressed using the probabilities of individual events implied by the assignment (Kirubarajan et al., 2001; Mori et al., 1992).

Under the independence assumption 2, we can express the likelihood of assignment A as $L(A) = \prod_{e \in E(A)} P(e)$. Here $E(A)$ is the set of all events implied by assignment A . Event $e \in E(A)$ can be one of the following (see Fig. 4):

cell move, cell division, cell disappearance due to cell death or a false negative error, and appearance of a new measurement due to a false positive error. The probabilities of the events in $E(A)$ can be expressed with the parameters P_s , P_d , P_{fp} , and P_{fn} (see Appendix B). Now the problem is to find assignment $A' = \arg \max_A L(A)$.

The search space of all possible assignments is usually reduced by introducing gating. The *gating* involves filtering out assignments with invalid links. An *invalid link* is a link with length $r_{ij} \geq R_g$, where R_g is called a gating distance (or gating threshold). The exact definition of an invalid link varies in the literature (Jaqaman et al., 2008; Kirubarajan et al., 2001), but the intuition is that the links that are deemed to be unlikely are not considered. In the case of our method, gating plays an important role. As we show later in the paper, there is a dependency between the gating distance R_g and the tracking performance.

So far we introduced an assignment problem formulation that closely follows previous work (Jaqaman et al., 2008; Kirubarajan et al., 2001). In what follows we present our contribution, a novel approach to the assignment problem.

Note that a straightforward solution of the problem (e.g., using integer programming) requires values for parameters P_s , P_d , P_{fp} , and P_{fn} . We propose to replace the four parameters by a single parameter R_g , and then formulate the problem as a maximization of performance with respect to R_g .

We claim that an approximate solution to the assignment problem can be found using the two round assignment procedure outlined below. The formal argument for our claim is presented in Appendix B.

In both assignment rounds we impose the following constraints: (1: no splits) at most one measurement can be assigned to one track, (2: gating) assigning measurement j to track i is only allowed if $r_{ij} < R_g$, and (3: maximum) we consider only the assignments with the maximum possible number of links.

Constraint 1 implies that in a single assignment round we eliminate the possibility of splitting a track. Instead, we capture a cell division event when in both rounds a measurement is assigned to the same track. Constraint 2 introduces the gating, and constraint 3 is explained as follows.

Consider, for example, assigning 2 measurements to 2 tracks, and suppose that no links are filtered out by the gating. There are a number of possible assignments: (a) link 1 — 1, and unassigned measurement 2, (b) links 1 — 1 and 2 — 2, (c) links 1 — 2 and 2 — 1, and so on. Constraint 3 states that assignment (a) cannot be considered because it consists of one link, whereas making two links is possible.

Recall that our task is to assign the measurements M_{f+1} in the next frame to tracks T_f in the current frame. In the first assignment round, we find an assignment of measurements M_{f+1} to tracks T_f that minimizes the cost $C = \sum_{a \in links} r_{ij}^2$, subject to constraints 1 — 3. Let $M_{un} \in M_{f+1}$ be the set of measurements (possibly empty) that were not unassigned in the first round. In the second round, we find the assignment of M_{un} to T_f that minimizes the same cost $C = \sum_{a \in links} r_{ij}^2$ subject to the same constraints.

Cases when two measurements (one from each round) are assigned to the same track represent cell divisions. All other links represent cell moves. The measurements that were not unassigned in both rounds represent false positives, and the tracks that received no measurements after the both rounds represent disappearance due to cell death or a false negative error.

Our two round assignment approach allows us to find an approximate solution for an assignment problem (see Appendix B). At the same time, our approach addresses the possibility of cell divisions and deaths, yet it uses a single and simple expression for cost C . Essentially we have reduced our interframe association problem to the global nearest neighbor assignment task in each round. This problem can be solved using the polynomial time algorithm of Munkres [1957].

Moreover, our approach does not require values of P_s , P_d , P_{fp} , and P_{fn} . We replaced these parameters by a single parameter R_g . In the next section, we present a method for deriving the value of this remaining parameter.

2.3 The Choice of Gating Distance

In previous work, the choice of gating distance has not been considered from the perspective of tracking performance. Indeed, it is an open challenge to derive a principled procedure for determining gating distance. For example, Jaqaman et al. [2008] choose the distance “beyond which a link between two particles was deemed impossible”. Further, deriving the value for the gating distance has required prior knowledge about cell offsets, i.e., required training (Godinez et al., 2009; Kirubarajan et al., 2001). In contrast, we provide an analytical framework for the choice of gating distance that maximizes tracking performance for a given video. Moreover, we propose an automated method for estimating the value of the optimal gating distance.

2.3.1 The Optimal Gating Distance

Previously, we approached the cell tracking problem as a likelihood maximization task. Now we consider the same tracking problem from a different perspective. We analyze how the tracking performance depends on the choice of the gating distance R_g . The performance is measured by the ratio of correctly reconstructed links P_{links} (equation 6).

Let us divide all possible links between measurements m_i and m_j into two groups. If the two measurements correspond to occurrences of the same cell, the link is a *true link*, otherwise the link is a *false link* (Fig. 5). For a given distance R , let the number of true [false] links shorter than R be $n_t(R)$ [$n_f(R)$]. For brevity we also denote these values as n_t and n_f . The tracking performance can then be expressed as a function $P_{links}(R) = P_{links}(n_t, n_f)$. Further, let the first derivative of P_{links} with respect to n_t (n_f) be denoted as dP_{link}/dn_t (dP_{link}/dn_f).

Gating implies filtering out the links longer than some gating distance R_g . In turn, filtering out a true link implies a swap error. Moreover, including more

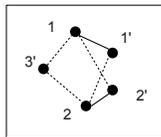


Fig. 5: A sample frame f superimposed on frame $f + 1$. Measurements in frame f are labeled with numbers (e.g., 1). Measurements in frame $f + 1$ are labeled with numbers with prime signs (e.g., 1'). Measurements with the same number correspond to the same cell, so 3' is a spurious measurement. Solid lines represent true links, and dotted lines represent false links.

true links increases the tracking performance, i.e., $dP_{link}/dn_t > 0$. On the other hand, including a false link, decreases the probability of correct tracking, thus $dP_{link}/dn_f < 0$. Notice, that the proportion of true links involved in tracking determines the upper bound of the tracking performance. Therefore, our hypothesis is that the proportion of true links is the dominating factor for tracking performance, i.e.,

$$\left| \frac{dP_{link}}{dn_t} \right| \gg \left| \frac{dP_{link}}{dn_f} \right|. \quad (8)$$

Consider an example that illustrates the dependency between the gating distance and the tracking performance (Fig. 6). Here R_{tmax} (R_{fmax}) is the maximal length of a true (false) link. If $0 \leq R_g < R_{tmax}$ then both n_t and n_f are increasing according to R_g . Under our hypothesis (equation 8), the increasing proportion of true links is the dominating factor, hence the performance rises.

If $R_{tmax} \leq R_g < R_{fmax}$ then we have all true links involved in the tracking process, i.e., n_t is fixed. Now the only factor that changes is the portion of false links that are passed in tracking, and we have that $dP_{link}/dn_f < 0$, hence the performance decreases. Under our hypothesis, the part of the curve between 0 and R_{tmax} is steeper than the curve between R_{tmax} and R_{fmax} . After the point R_{fmax} , both n_t and n_f are fixed, thus the choice of R_g has no effect on the performance.

The *optimal gating distance* is the distance when NENIA achieves the best performance. Under our hypothesis, the optimal gating distance is the distance $R_g = R_{tmax}$ (this suggestion is validated by our experiments). It is natural to assume $R_{fmax} > R_{tmax}$, however, the optimal gating distance is the same $R_g = R_{tmax}$ in the case when $R_{fmax} < R_{tmax}$. In the case when $R_{fmax} < R_{tmax}$ (diagram not shown), in the interval $0 \leq R_g < R_{fmax}$, the proportion of true links is increasing. Under our hypothesis, the proportion of true links is a dominating factor, hence the performance is increasing. In the interval $R_{fmax} \leq R_g < R_{tmax}$, n_f is fixed and we also have $dP_{link}/dn_t > 0$. Thus the maximum performance is achieved at $R_g = R_{tmax}$.

In order to find the point R_{tmax} , we reconstruct the cumulative density

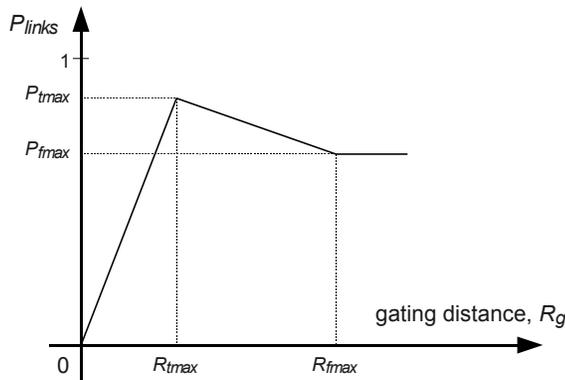


Fig. 6: Hypothetical curve of the cell tracker performance as a function of gating distance. R_{tmax} (R_{fmax}) is the maximum length of a true (false) link. Our hypothesis is that the portion of allowed true links is the dominant factor that affects the performance ($0 \leq R_g < R_{tmax}$). This factor is stronger than the portion of allowed false links ($R_{tmax} \leq R_g < R_{fmax}$). The choice of $R_g \geq R_{fmax}$ corresponds to a tracking without gating. The tracking performance is the same for all $R_g \geq R_{fmax}$.

function (CDF) for the true links (i.e., cell displacements) as described in the next section.

2.3.2 Distribution of Cell Displacements

Let $P_t(r \leq R)$, $P_f(r \leq R)$, and $P_{all}(r \leq R)$ be the probabilities that the length of a true, false, and any link is less than a certain distance R and let CDF_t , CDF_f , CDF_{all} be the cumulative distribution functions for true, false and all links respectively.

We can empirically estimate CDF_{all} by considering all possible links between any two measurements in consecutive frames in the video.

Now consider false links. These links connect two unrelated measurements in consecutive frames. Under Assumption 3 the spatial distribution of cells within a frame does not change greatly, hence connecting two unrelated measurements in consecutive frames is approximately the same as connecting unrelated measurements within a frame. This can be illustrated using two superimposed frames (Fig. 5). We therefore propose to approximate CDF_f with the distribution of links between measurements within a frame.

Now let N_t , N_f , N_{all} be the numbers of true, false and all links in a video. Here $N_t + N_f = N_{all}$. We can calculate N_{all} by counting the number of all potential links in the video. At the moment, assume that we know N_t .

At a given R let n_t , n_f , n_{all} be the numbers of true, false and all links that are less than R , here $n_t + n_f = n_{all}$. Then $P_t(r \leq R) = \frac{n_t}{N_t}$, $P_f(r \leq R) = \frac{n_f}{N_f}$

and $P_{all}(r \leq R) = \frac{n_{all}}{N_{all}}$, and for all R , the CDF_t can be reconstructed using the following equation

$$P_t(r \leq R) = \frac{N_{all}P_{all}(r \leq R) - (N_{all} - N_t)P_f(r \leq R)}{N_t}. \quad (9)$$

Finally, we show how to find N_t . Consider a frame i with n_i measurements (Fig. 7). Among these measurements there are on average $n_i P_{fp}$ false positives, hence we expect on average $n_i(1 - P_{fp})$ cells. However, on average $n_i(1 - P_{fp})P_{fn}$ of these cells are not detected in the previous frame $i - 1$. Therefore, there are on average $n_i(1 - P_{fp})(1 - P_{fn})$ true links between frames $i - 1$ and i , and we can approximate N_t as

$$N_t \approx (1 - P_{fp})(1 - P_{fn}) \sum_{i=2}^F n_i, \quad (10)$$

where F is the number of frames in the video.

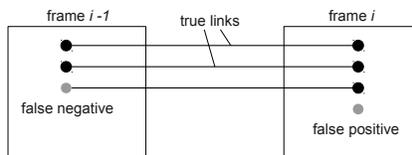


Fig. 7: Estimation of the number of true links. This number depends on the number of measurements in each frame and on the probabilities of a false positive and a false negative error.

Let $\hat{N}_t \in (0, N_{all})$ be an approximation of N_t . We then set the gating distance $R_g = \hat{R}_{tmax}$, where \hat{R}_{tmax} is the first point such that

$$\frac{N_{all}P_{all}(r \leq R) - (N_{all} - \hat{N}_t)P_f(r \leq R)}{\hat{N}_t} = 1. \quad (11)$$

We refer to the method of estimating N_t using equation 10 as method A. Here we require two parameters P_{fp} and P_{fn} for the estimation of N_t . Estimation of P_{fp} and P_{fn} generally involves manual verification of cell segmentation results. Therefore, in the rest of this section, we present method B — a rough estimation of N_t without these parameters.

Let denote $Q = (1 - P_{fp})(1 - P_{fn})$. Notice that, under Assumption 1 about cell detection quality, $Q \in [0.9; 1]$. Further, note the following property.

Lemma 1. *If $\hat{N}_t \geq N_t$ then $\hat{R}_{tmax} \geq R_{tmax}$.*

Recall that our hypothesis was that the part of the error curve (Fig. 6) that is on the left from R_{tmax} is steeper than the part on the right. Further, on the right hand side, there is an intercept value R_{fmax} , beyond which we do not have

performance degradation. Thus, we conclude that it is “safer” to overestimate R_{tmax} rather than underestimate. We, therefore, suggest to use

$$N_t \approx \hat{N}_t = \sum_{i=2}^F n_i \quad (12)$$

as an approximation for N_t .

Algorithm 1 summarizes our method for deriving the gating distance value. Method A and method B differ only in the way they set the value N_t .

Algorithm 1 Estimation of Gating Distance

- 1: **function** *EstimateGatingDistance*(M, P_{fp}, P_{fn})
 - 2: // M is the set of measurements in the entire video;
 - 3: // P_{fp}, P_{fn} are not required in method B
 - 4: method A: use P_{fp}, P_{fn} and equation 10 to estimate the number of true links N_t ;
 - 5: (or method B: use equation 12 to estimate the number of true links N_t);
 - 6: use equation 11 to estimate the distribution of true links;
 - 7: find the minimum R_g such that $P_t(r \leq R_g) = 1$;
 - 8: **return** R_g
-

We have now described the key parts of our tracking algorithm. In the following section, we summarize the algorithm.

2.4 Automated Cell Tracker NENIA

Previously, we formulated the cell tracking task as a problem of iterative assignment between two frames. However, in each iteration, this method does not take into consideration the information available outside the two frames. Iterative interframe assignment represents a greedy approach to cell tracking.

Cell tracking can be formulated as a global assignment problem, but then the solution becomes computationally infeasible even for relatively short videos. It is, therefore, common to use heuristic methods that represent trade-offs between the greedy and global approaches (Kirubarajan et al., 2001; Shafique and Shah, 2005).

Our cell tracking algorithm (see Algorithm 2) uses interframe assignments supplemented with a track management module. The track management module uses the information available outside the two frames to improve the interframe assignments.

The algorithm takes as input sets of measurements in each frame of a video and produces a set of established tracks with lineage relations between tracks.

Algorithm 2 Parameter-free tracker NENIA

Input: measurements M_1, \dots, M_F ; detection quality P_{fp}, P_{fn} (these two are not required in method B);

Output: set of tracks T ;

```

1:  $T \leftarrow \text{InitNewTracks}(M_1)$ ;
2:  $R_g \leftarrow \text{EstimateGatingDistance}(M_1, \dots, M_F, P_{fp}, P_{fn})$ ;
3: for each frame  $f = 2 \dots F$ 
4:    $\text{mapping}_1 \leftarrow \text{FirstAssignment}(T, M_f, R_g)$ ;
5:    $T \leftarrow \text{UpdateTracks}(T, \text{mapping}_1)$ ;
6:    $T \leftarrow \text{ManageLostTracks}(T, \text{mapping}_1)$ ;
7:    $T \leftarrow \text{SplitTracks}(T, \text{mapping}_1)$ ;
8:    $M_{un1} \leftarrow \text{UnassignedMeasurements}(\text{mapping}_1)$ ;
9:    $\text{mapping}_2 \leftarrow \text{SecondAssignment}(T, M_{un1}, R_g)$ ;
10:   $T \leftarrow \text{MarkDivCandidates}(T, \text{mapping}_2)$ ;
11: end
12: return  $T$ ;
```

At each iteration, an interframe assignment is performed in two rounds (lines 4 and 9). Cell move events are processed by appending a new measurement to the corresponding track (line 5).

NENIA uses an extra frame to supplement interframe assignments. The extra frame is implicitly used in functions *ManageLostTracks* (line 6 and *SplitTracks* (line 7). At iteration i , these functions use the result of the assignment between frames $i - 1$ and i to complete processing of tracks based on the assignment between frames $i - 2$ and $i - 1$. Specifically, the disappearance events in the first assignment round between frames $i - 2$ and $i - 1$ result in marking the corresponding tracks as lost. A track is then terminated only if it is lost in two consecutive frames (line 6).

Similarly, in the second round of assignment between frames $i - 2$ and $i - 1$, there can be division events, but then the mother track is only marked as a division candidate (line 10). The division is processed only if both daughter tracks persist in two consecutive frames (line 7).

Lemma 2. *The probability of encountering a sequence of two or more false positive errors linked together by NENIA is less than 0.05. Similarly the probability of encountering a sequence of two or more false negative errors missing from the same track is less than 0.05.*

Lemma 2 gives an upper bound estimation for the probabilities. In practice, we expect the sequences of related errors to be even more rare. Based on this lemma we decided to use only one look-ahead frame.

Our evaluation shows that NENIA achieves strong performance on a variety of videos. However, for some videos, our tracker still makes occasional tracking errors (this is also true for the other modern tracking algorithms). These errors can be resolved manually by an operator. In the next section, we present a novel method to assist this manual correction process.

2.5 Semi-automated Tracker SAMTRA

Tracking can be improved using human assistance, where an operator reviews the video superimposed with the tracking results. Once the operator detects a tracking error, he or she can intervene to correct those errors. However, the drawback of this approach is that the operator has to review all frames.

Evaluations of tracking algorithms (Jaqaman et al., 2008; or Section 3.2 in this paper) suggest that better cell detection leads to better tracking. One way to improve cell detections is by manual intervention, although this is time consuming. In order to speed up the process we propose a system called SAMTRA to identify those frames most likely to benefit from manual editing. SAMTRA (semi-automated tracker) predicts the probability of a detection error in a given frame. If human resources are limited, then only the frames with the highest error probability are selected for manual review.

SAMTRA predicts the probability of detection errors in a frame using a set of features that are collected during tracking. Let M_f be the number of measurements in frame f . Consider a transition from frame $f - 1$ to frame f . Suppose that at this point, according to some tracker, T_f existing tracks are updated, N_f new tracks are initiated, L_f lost tracks are terminated, R_f lost tracks are recovered, and S_f tracks are split.

For each updated track t_i , let a displacement between the track positions $x_{f,i}$ and $x_{f-1,i}$ at frames f and $f - 1$ be $r_{f,i} = \|x_{f,i} - x_{f-1,i}\|^2$, and the mean displacement over all updated tracks for frame f be

$$R_f = \frac{1}{T_f} \left(\sum_{i=1}^{T_f} r_{f,i} \right). \quad (13)$$

We use 11 tracker independent features $\mathcal{F} = \{M_{f-2}, M_{f-1}, M_f, M_{f+1}, N_f, L_f, R_f, S_f, R_{f-1}, R_f, \frac{R_f}{R_{f-1}}\}$ in a linear logistic regression with the logit

$$z = \omega_o + \sum_{i=1}^{11} \omega_i F_i, \quad (14)$$

where F_i enumerates the features in \mathcal{F} and ω_i are feature weights.

Let k_1 and k_2 be chosen such that $k_1 + k_2$ is the number of frames that is considered feasible for manual inspection, and k_1 is the number of instances that is sufficient to train the logistic regression classifier. We then take k_1 frames from a video and run a cell segmentation algorithm. We then manually generate the response vector. For each frame, the corresponding value in the vector equals

to 1 if there is a detection error, or 0 otherwise. Using this manually collected response vector, we learn the weights ω_i using the logit in equation 14 with features \mathcal{F} . Finally, for the rest of the video, or for other similar videos, we estimate the probability of detection error in frame f is as

$$P_{err} = \frac{1}{1 + e^{-z}}. \quad (15)$$

Finally, we take k_2 frames with the highest probabilities of error, and manually correct cell detection (not tracking) in these frames. This results in improved tracking quality, as shown by our evaluations, presented in the next section.

3 Evaluation

The goal of our experiments has been to evaluate the proposed automated and semi-automated cell tracking methods across a variety of tracking scenarios. To this end, we used five videos that differ in cell density, and motility. The videos are described in Section 3.1.

We implemented our cell tracking system in MATLAB, version *R2009a*. In our system, we used one of the assignment algorithm implementations freely available from the Web³. This implementation is based on the algorithm of Munkres [1957], and is capable of processing rectangular cost matrices with infinite costs representing prohibited links.

In the gating distance estimation, the CDF of true links was reconstructed as described in Section 2.3 (with both methods A and B). We also applied median filtering to reduce noise in the reconstructed CDF curve. See Appendix D for details of the filtering.

We evaluate the effect of the gating distance on the tracker’s performance in Section 3.2. In the same section, we compare the performance of NENIA with the performance of an advanced probabilistic cell tracker. We evaluate NENIA on synthetic videos in Section 3.3. Finally, we study the efficiency of SAMTRA in Section 3.4.

3.1 Cell Videos

The videos used in our evaluations are summarized in Table 3.

Video *ak* shows the development of neural progenitor cells. This video sequence consists of the first 300 frames (enough to observe 3 division rounds) of supplementary movie one⁴ from the work of Al-Kofahi et al. [2006].

The remaining four videos show the development of B lymphocytes. Primary B lymphocytes were purified from the spleen of C57/BL6 mice modified

³ Functions for the rectangular assignment problem by Markus Buehren. Available on-line at <http://www.mathworks.com/matlabcentral/fileexchange/6543>

⁴ Used with permission. Available on-line from the Cell Cycle journal website at <http://www.landesbioscience.com/journals/cc/supplement/alkofahi.zip>

Tab. 3: Videos used for tracker evaluation. These videos differ in cell density, and motility. The parameter β_n reflects the “difficulty of video” for cell tracking (defined in Section 3.3).

Name	Frames	Cells	$\beta_n \times 10^2$	Comments
<i>ak</i>	300	18	0.25	dense clutter of cells, low motility, 7 divisions
<i>hex.6</i>	100	6	5.94	starts with 2 cells, 2 divisions
<i>hex.16</i>	100	16	3.45	dense clutter of cells, no events, except cell moves
<i>hex.22</i>	100	22	3.18	dense clutter of cells, 4 divisions, 2 deaths
<i>square</i>	50	35	0.14	cells are uniformly distributed with low density, 2 divisions

to ubiquitously express GFP, prepared as described in the work of Hawkins et al. [2009]. These were placed in $250 \mu\text{m}$ hexagonal microwells (videos *hex**) and $125 \mu\text{m}$ square microgrids (video *square*), and photographed once every 2 minutes using an Axiovert 200m microscope with a $20\times$ objective at different scales for hexagonal and square microwells.

Fig. 1 shows the fragments of frames from videos *hex.16* (left), *square* (middle), and *ak* (right), see Supplementary Fig. 15 for sample frames from videos *hex.6* and *hex.22*.

We found that our videos represent a variety of tracking scenarios, such as different cell density and dynamics (see details in Appendix E). This is despite four of the videos showing the same type of cells. Further, three videos *hex** give an example of variation in cell density and dynamics, while having similar cell appearance. In this case, the same segmentation algorithm can be used, but the variability challenge still needs to be addressed by the tracker.

Some of the previously proposed methods made an assumption of the Brownian motion of cells (Bonneau et al., 2005; Jaqaman et al., 2008). However, in our videos, we found that at least in four of them, cells are unlikely to follow Brownian motion (see details in Appendix E).

For each video, we generated detections with different qualities specified by $\{P_{fp}, P_{fn}\}$. We refer to a combination $\{\text{video}, P_{fp}, P_{fn}\}$ as a *case*. For each case, we generated 5 random versions (see details in Appendix E) and report the average performance across 5 versions. Cases with $P_{fp} + P_{fn} \leq 0.06$ correspond to high detection quality.

We do not report the sizes of videos in pixels, because the sizes are not directly related to the tracking performance. Instead, we report a dimensionless

parameter β_n which is the normalized object density. This parameter is formally defined in Section 3.3.

The number of frames in our videos is not very large. In practice, videos can comprise thousands of frames. However, our videos include several instances of cell division and death events. Having short videos does not affect our evaluation, since NENIA is based on independent frame by frame assignments. Further, we compare our tracker with a competitor’s tracking algorithm called LJIPDA (introduced in the next section). Having short videos does not affect the evaluation of LJIPDA, because LJIPDA is also assignment based (however, LJIPDA uses a different approach for the assignment).

Despite the variety in cell density and dynamics, NENIA is capable of achieving high tracking accuracy on different detection qualities, as illustrated in the next section.

3.2 Evaluation of NENIA

In this section, we first analyze the performance of NENIA on different detections and compare it with the performance of a probabilistic tracker. We then compare the tracking performance for estimation methods A and B. For each video, all video frames were used for automated derivation of the gating distance and evaluation of performance. The derivation of the gating distance parameter is one of the crucial steps in NENIA. Therefore, we conclude with the measurement of the effect of gating distance on the performance of our cell tracker (measured with P_{links}).

Fig. 8 shows the performance of NENIA (with method A for selection of the gating distance) compared with the performance of the Linear Joint Integrated Probabilistic Data Association (LJIPDA) tracker. LJIPDA is one of the most advanced probabilistic trackers [Musicki and Evans, 2002].

Unfortunately at present, it is difficult to compare tracking results directly across different trackers. This is due to the absence of common baseline datasets, and the lack of open-source algorithm implementations. Most papers in the field tend to report results only for their method and on their datasets (Al-Kofahi et al., 2006; Li et al., 2008; Padfield et al., 2010). We managed to obtain an original implementation of the LJIPDA tracker, which allowed us to perform a direct comparison easily.

When the detection quality is high (cases when $P_{fp}/P_{fn} = 0/0, 1/0, 0/1, 3/3$, expressed in %), NENIA outperforms LJIPDA and the difference is statistically significant. On average across all cases, there is no statistically significant difference in performance (the statistical significance was measured with the Wilcoxon matched-pairs signed-rank test at the 5% significance level). At the same time NENIA does not require setting parameter values manually, whereas for LJIPDA we had to use the manually produced ground truth to tune the parameter values for LJIPDA.

We now compare the performance corresponding to methods A and B in Fig. 9. On average across all cases, method A is better and the difference is statistically significant as measured with the Wilcoxon matched-pairs signed-

rank test at the 5% significance level. On the other hand, in all observed cases, method B recovers more than 80% of cell moves correctly, thus it still performs well enough to be useful. The two methods provide a trade-off: if the detection quality can be estimated easily, then method A gives better tracking accuracy; if it is hard to estimate the detection quality, then method B can still be useful.

Finally, Fig. 10 shows empirical curves for the tracking performance as a function of gating distance. It can be observed that in all cases, empirical data supports our hypothetical curve (compare with Fig. 6). When the gating distance is too small, filtering out a part of the true links has a dramatic effect on the performance. When the gating distance lets all the true links in, the performance depends on the proportion of the true and false links. However, when the gating distance is too large and allows all true and false links, the performance does not depend on the gating.

Vertical strokes in Fig. 10 show the gating distances, estimated by method A. Both methods A and B estimate suboptimal gating distances (strokes are not shown for method B). The optimal performance, on average, is only 1.6% higher than the performance of method A (data not shown).

When the cell detection is perfect ($P_{fp} = P_{fn} = 0\%$), NENIA was able to recover all division events correctly. We further investigate the ability of NENIA to resolve cell divisions in our synthetic experiments in the next section.

We summarize our evaluation results as follows.

- For high cell detection quality, NENIA resolves almost 100% of moves correctly and can be used as a main tracking algorithm. In all other cases, NENIA (with method A or B) resolves more than 80% of cell moves correctly and could be used for collecting the ground truth for training of a more specialized tracking algorithm. The ground truth can be first automatically generated by NENIA. Then a human operator is required to label at most 20% of links.
- For high cell detection quality, NENIA (with method A) outperforms LJIPDA, despite NENIA being a simpler parameter-free algorithm.
- Method A for determining the gating distance outperforms method B. However, in the case when the estimation of P_{fp} and P_{fn} is hard, method B still gives reasonable performance, more than 80% in all cases.
- The tracking performance degrades with the detection performance. This suggests that in order to improve a tracker’s performance one can investigate improving detection quality.

The running time for our cell tracker was on average 65 frames per second (i.e., in total, 10 seconds for all 5 videos, which includes the estimation of gating distance) run on a desktop PC (Intel Core 2 Duo CPU *E8400* @ 3.00 GHz and 3.4 GB RAM, Ubuntu 9.10). The scalability of NENIA for larger numbers of cells is evaluated in the next section.

3.3 Evaluation on Synthetic Videos

In addition to our real videos, we used synthetic videos for evaluating different aspects of NENIA: scalability in terms of the tracking performance and running time, validity of our hypothesis about tracking performance as a function of gating distance, and the ability of NENIA to resolve cell divisions. In this section, we outline main findings on our synthetic videos. Details of the evaluation are presented in Appendix F.

The first type of synthetic video that we used consists of pairs of frames. The cells are randomly placed with uniform distribution in the first frame, and randomly moved with a bivariate normal distribution. In the synthetic videos we varied a number of parameters, such as the number of cells N , the variance of a random offset σ , and the side of a square video frame L . We also added random false positives and false negative errors with error rates P_{fp} , P_{fn} .

Mori et al. [1992] define a *normalized object density* as a dimensionless value $\beta_n = (N \cdot \sigma^2 \cdot \pi / L^2)$ and show that this density determines the “difficulty” of the tracking problem (a similar observation has been made by Jaqaman et al., 2008). An intuitive interpretation is that, given a fixed frame size, increasing the number of cells or cell speeds increases the probability of confusing their tracks, and hence makes the tracking more difficult.

For the evaluation of tracking scalability in terms of the performance and running time, we varied the number of cells N in the range 30 – 200 and used several different sets of other parameters. Overall we found that once the normalized density, β_n , is fixed, the tracking performance, P_{links} , does not depend on the number of cells. This is consistent with the result of Mori et al. [1992]. Further, we observed a polynomial relation between the running time and the number of cells. Empirical polynomial complexity of NENIA is consistent with theoretical complexity estimation for the assignment algorithm of Munkres [1957] which is a core step of NENIA.

For testing the relationship between the tracking performance and the gating distance, we varied β_n such that it covers a much wider range of values than the real videos. We used a few different sets of other parameters, and recorded the performance of NENIA P_{links} as a function of gating distance R_g . All obtained empirical curves (see examples in Fig. 11, not all curves shown) preserve the prominent features of our hypothesized curve (Fig. 6): sharp rise from zero to maximum, maximum around the point $R_g = R_{tmax}$, and gradual decrease, after this point (recall that R_{tmax} is the maximum length of a true link). We conclude that it is feasible to estimate the optimal gating distance as $R_g = R_{tmax}$, as proposed in our analysis (Section 2.3).

Synthetic videos cover both cases when $R_{tmax} < R_{fmax}$ and $R_{tmax} > R_{fmax}$ (recall that R_{fmax} is the maximum length of a false link). The cases when $R_{tmax} > R_{fmax}$ occur at high values of β_n ($\beta_n > 300$). These cases correspond to a situation when, between consecutive frames, cells travel distances that are large compared to the distances between cells in a frame. This results in a rapid confusion of cells and consequently in a poor tracking performance. We found that the empirical performance curve in this situation ($R_{tmax} > R_{fmax}$) still

resembles our analytical curve.

Finally, we used another type of synthetic video to evaluate the ability of NENIA to resolve cell divisions. These videos consist of 3 frames, and start with 100 cells out of which 30 cells divide. We generated the synthetic videos using different combinations of parameters β_n and P_{fp} . Please see the details of our evaluation in Appendix F.

We found that for every parameter combination, the majority of divisions are resolved correctly: the ratio was about 100% for the easiest cases and about 72% for the hardest cases. We cannot compare these numbers with the performance of the LJIPDA algorithm, because LJIPDA does not address cell divisions, but we note that our results are somewhat similar to the results of Jaqaman et al. [2008] and Padfield et al. [2010]. The evaluation of Padfield et al. [2010] roughly corresponds to our case when $P_{fp} = 0\%$. This comparison is indirect, as different authors use different videos.

Overall, we found NENIA to be scalable in terms of performance and running time. Further, we verified our hypothesis about the relation between the tracking performance and the gating distance. We also found that in our synthetic videos, NENIA correctly resolves most cell divisions.

Further, we note that incorporating a physically well-conserved observable quantity such as cell fluorescence into NENIA can improve its ability to resolve divisions (this is the subject of future work). Finally, recall that cell moves significantly outnumber cell divisions in real videos. Therefore the *accumulation* of errors from cell moves in between divisions needs to be taken into account when comparing a tracker’s ability to maintain cell identity *between* and *across* divisions.

3.4 Evaluation of SAMTRA

In the evaluation of SAMTRA, we used P_{track} (equation 7) as the tracking performance measure, because SAMTRA is designed for the situations when it is important to recover as many tracks as possible correctly. Evaluation of SAMTRA was conducted using five real cell videos described previously in Section 3.1.

We trained SAMTRA using only one of the videos (*hex.22*) with one detection quality level ($P_{fp} = P_{fn} \approx 2.5\%$). We then used all our videos at other detection levels (other values for P_{fp} and P_{fn}) to evaluate SAMTRA. For each video, we randomly chose a detection such that the tracking quality on that detection is around 75%. We then randomly selected 10% of the frames and corrected the cell detection, where necessary, in the selected frames. We then ran the tracking again and recorded the tracking performance. In the evaluation, we used NENIA with method B.

After that, we took the original detection and again selected 10% of frames. This time we selected frames with the highest probabilities of errors as predicted by SAMTRA. Where necessary, we corrected cell detection in these frames, ran tracking again, and recorded the tracking performance. For each video we ran the experiment 5 times (in Table 4, we show only the first experiment for each

Tab. 4: Tracking performance gain for NENIA by manual correction of cell detections in 10% of all frames in a video. SAMTRA is generally better than random selection.

Video	Frames total	Performance (original)	Performance (random recovery)	Performance (SAMTRA recovery)
ak	300	78	78	100
hex.6	100	67	67	67
hex.16	100	75	75	81
hex.22	100	82	82	95
square	50	83	83	100

video). In most runs, SAMTRA gave a higher tracking performance increase than the random frame selection. In none of the runs did SAMTRA give a smaller increase than the random selection.

The results show that using SAMTRA we were able to increase the ratio of correct tracks by 12% (recall that a correct track has all its links correct). This was achieved by inspecting only 10% of frames and correcting cell detections in those frames. Also note that in most experiments, selecting frames for review using SAMTRA led to more substantial performance gains than random selection. These results demonstrate that SAMTRA can be used for efficient semi-automated tracking correction.

We also studied how much training data is needed for SAMTRA to achieve a certain level of performance gain. We observed an increase when using as few as 20 frames. The results are presented in Table 6, and the details of the evaluation are presented in Appendix G.

In the next section, we further review the potential and applicability of our automated and semi-automated methods. We also comment on the prospects of our method for estimation of the CDF of the true links.

4 Discussion

In our evaluation, we only use cell centroid locations because it appears that this feature is available with any cell segmentation algorithm. We found that the performance of NENIA is reasonably high even with this minimum amount of information. In practice, there can be more information available, such as cell size or fluorescence. These extra features can be seamlessly incorporated in NENIA to improve the performance further.

New features can be added to the Euclidean distance expression after normalizing by the features variances. The segmentation algorithm produces sets of features, e.g., set of centroid locations and set of cells areas. Then the variance of each feature can be calculated automatically. Therefore, after incorporating new features, NENIA does not require extra human attention.

In NENIA we use the same gating threshold and division probabilities for

all cells. We note that in reality the lengths of cell moves and the division probabilities are cell and time dependent. Varying the parameters across cells might further improve the accuracy of tracking. However, estimating the division probability or the gating threshold for a particular cell requires tracking cell identities in the first place. It is possible to use tracking results for the past frames to estimate the gating threshold for a cell in the future frames [Jaqaman et al., 2008], but the past tracking results may contain errors and therefore the estimation can be inaccurate. To the best of our knowledge, there is no evidence yet in the literature that such an adaptive method is superior to one that uses a common gating threshold (comparative analysis can be a direction for future work). A similar remark can be applied to adaptive estimation of division probabilities. Further, we note that on the range of videos tested, NENIA (with a common gating threshold and division probability) performs sufficiently well to be useful in practice.

NENIA with method B is a parameter-free algorithm in the sense that for its operation it only requires the problem input (i.e., cell detection), and does not require user attention. Internally NENIA does use some variables, such as the gating distance R_g , but these are initialized automatically. Therefore the use of internal variables is immaterial for end-users. Note that this is not the case with many previously proposed algorithms, where a user is required to somehow determine the value for a gating threshold (Al-Kofahi et al., 2006; Jaqaman et al., 2008; Kirubarajan et al., 2001).

NENIA with method A requires the user to provide an estimation of the quality of cell detection (P_{fp} , P_{fn}). In some cases, one can assume that the detection quality remains roughly the same across a number of videos (e.g., the videos show the same cells at about the same density). In these cases, a manual estimation of the detection quality can be done for one video and then used for the other videos. In cases where the similarity assumption cannot be made, we suggest to use NENIA with method B. We emphasize that providing accurate estimations for the detection quality is not critical for our cell tracker as our experiments show that NENIA with method B can still achieve reasonable performance.

Some variables in NENIA are fixed to particular values (e.g., the form of the cost function and the number of look-ahead frames). However, for each such variable we provide an argument as to which particular values are reasonable under the assumptions 1 – 3.

NENIA was designed to be portable across a variety of videos. In particular cases, specialized algorithms can achieve higher performance. For example, there can be a tracker optimized to certain cell dynamics. The problem is, given a video, to find out which particular case it represents. Generally, the cell dynamics are not known a priori. To address this problem, NENIA can be used in a “bootstrap” procedure: it can assist manual collection of the ground truth for a part of the video. The ground truth can then be used to decide which specialized algorithm to use for the rest of the video.

Regarding the semi-automated method, notice that SAMTRA uses tracker-independent features, and thus is applicable for use with different tracking al-

gorithms. Moreover, SAMTRA can be used for tracking in the first place, by aiming to find a tracking that minimizes the probability of errors in all frames.

Our implementation of SAMTRA involves training from the tracking results. This approach is independent of the choice of segmentation algorithm. Moreover, it can be possible to avoid training when using a segmentation algorithm that yields the relative level of certainty in its detections as a part of its output. Here the estimation of the probabilities of errors in frames can be based on how certain the segmentation algorithm was in a particular frame.

Finally, we would like to highlight our method for the reconstruction of the CDF of true links. This method was presented here in the context of our cell tracking algorithm. However, this technique does not depend on the tracking algorithm and has further potential applications as follows.

- The method can be used for the estimation of the average speed of cells (without doing cell tracking). This, in turn, can be used for deriving an optimal frame acquisition rate for cell videos.
- The method can be used for automated detection of the changes in distribution of cell displacements. This can be used for searching for some key time points in a video.
- The method can be used for estimation of a cell tracker’s performance in the absence of a ground truth. Consider a cell tracker that produces a set of tracks. From this set of tracks, one can derive the CDF of the cell displacements. On the other hand, we can reconstruct the CDF of true links using our method (without tracking). We can then use the Kolmogorov-Smirnov test to compare two distributions. The closer the distributions are to each other (i.e., the smaller the p -value of the test), the better is the tracker.

5 Conclusions

We have developed and evaluated a novel cell tracking algorithm called NENIA. The algorithm can handle cell divisions and deaths, and is portable across a variety of videos. We also propose a parameter-free configuration of NENIA. Our algorithm is based on interframe assignments supplemented by a track management module. We first designed the algorithm in such a way that it depends on a single parameter, the gating distance. Further, we analyzed the effect of gating distance on the tracker’s performance, and proposed methods for automated estimation of the gating distance, hence removing this parameter.

For the range of videos tested, NENIA correctly resolves on average 96% of cell moves, and when the quality of cell detection is high, NENIA correctly resolves almost 100% of cell moves. This performance is achieved across a variety of real cell videos that differ both in cell density and dynamics. The scalability of our algorithm was tested on synthetic videos with up to 200 cells per frame. The empirical complexity of NENIA was found to be a polynomial of degree 3 on synthetic videos.

NENIA is based on a novel method for reconstruction of the CDF of cell displacements. This method has several potential applications, such as evaluation of tracking performance without the ground truth.

We also proposed a semi-automated tracking method called SAMTRA. This method uses logistic linear regression to predict the probability of detection errors in a video. Our evaluation shows that SAMTRA is capable of increasing the proportion of correctly reconstructed tracks on average by 12% through manually correcting detections in only 10% of all frames in a video.

Acknowledgments

We would like to thank Daniel Day from the Center for Micro-Photonics at Swinburne University of Technology for supply of the $125\ \mu\text{m}$ microgrids.

This work is partially supported by National ICT Australia. National ICT Australia is funded by the Australian Government's Backing Australia's Ability initiative, in part through the Australian Research Council.

References

- I. Adanja, O. Debeir, V. Mégalizzi, R. Kiss, N. Warzée, and C. Decaestecker. Automated tracking of unmarked cells migrating in three-dimensional matrices applied to anti-cancer drug screening. *Exp. Cell Res.*, 316(2):181–193, 2010.
- O. Al-Kofahi, R.J. Radke, S.K. Goderie, Q. Shen, S. Temple, and B. Roysam. Automated cell lineage construction: A rapid method to analyze clonal development established with murine neural progenitor cells. *Cell Cycle*, 5(3):327, 2006.
- S. Bonneau, M. Dahan, and L.D. Cohen. Single quantum dot tracking based on perceptual grouping using minimal paths in a spatiotemporal volume. *IEEE Trans. Image Process.*, 14(9):1384–1395, 2005.
- Y. Chen, E. Ladi, P. Herzmark, E. Robey, and B. Roysam. 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. *J. Immunol. Methods*, 340(1):65–80, 2009.
- E.D. Cheng, S. Challa, R. Chakravorty, and J. Markham. Microscopic Cell Segmentation by Parallel Detection and Fusion Algorithm. In *Proceedings of the 10th IEEE Workshop On Multimedia Signal Processing*, pages 94–100, 2008.
- A.S. Chowdhury, R. Chatterjee, M. Ghosh, and N. Ray. Cell Tracking in Video Microscopy using Bipartite Graph Matching. In *Proceedings of the 20th International Conference on Pattern Recognition*, pages 2456–2459, 2010.

- D. Day, K. Pham, M.J. Ludford-Menting, J. Oliaro, D. Izon, S.M. Russell, and M. Gu. A method for prolonged imaging of motile lymphocytes. *Immunol. and Cell Biol.*, 87(2):154–158, 2008.
- O. Debeir, P.V. Ham, R. Kiss, and C. Decaestecker. Tracking of migrating cells under phase-contrast video microscopy with combined mean-shift processes. *IEEE Trans. Med. Imaging*, 24(6):697–711, 2005.
- J. Degerman, T. Thorlin, J. Fajerson, K. Althoff, P.S. Eriksson, R.V.D. Put, and T. Gustavsson. An automatic system for in vitro cell migration studies. *J. Microsc.*, 233(1):178–191, 2009.
- A. Dufour, V. Shinin, S. Tajbakhsh, N. Guillen-Aghion, J.C. Olivo-Marin, and C. Zimmer. Segmenting and tracking fluorescent cells in dynamic 3-D microscopy with coupled active surfaces. *IEEE Trans. Image Process.*, 14(9):1396–410, 2005.
- O. Dzyubachyk, W.A. van Cappellen, J. Essers, W.J. Niessen, and E. Meijering. Advanced level-set-based cell tracking in time-lapse fluorescence microscopy. *IEEE Trans. Med. Imaging*, 29(3):852, 2010.
- W.J. Godinez, M. Lampe, S. Woerz, B. Mueller, R. Eils, and K. Rohr. Deterministic and probabilistic approaches for tracking virus particles in time-lapse fluorescence microscopy image sequences. *Med. Image Anal.*, 13(2):325–342, 2009.
- E. Gomez, M.A. Gomez-Vilegas, and J.M. Marin. A multivariate generalization of the power exponential family of distributions. *Commun. Stat. Theory Methods*, 27(3):589–600, 1998.
- E. Hamzic, J.E. Cartwright, R.J. Keogh, G.S.J. Whitley, D. Greenhill, and A. Hoppe. Live cell image analysis of cell–cell interactions reveals the specific targeting of vascular smooth muscle cells by fetal trophoblasts. *Exp. Cell Res.*, 314(7):1455–1464, 2008.
- E.D. Hawkins, J.F. Markham, L.P. McGuinness, and P.D. Hodgkin. A single-cell pedigree analysis of alternative stochastic lymphocyte fates. *Proc. Natl. Acad. Sci. U.S.A.*, 106(32):13457, 2009.
- T. Inglis, H. De Sterck, G. Sanders, H. Djambazian, R. Sladek, S. Sundararajan, and T. Hudson. Multilevel Space-Time Aggregation for Bright Field Cell Microscopy Segmentation and Tracking. *Int. J. Biomed. Imaging*, 2010, 2010.
- K. Jaqaman, D. Loerke, M. Mettlen, H. Kuwata, S. Grinstein, S.L. Schmid, and G. Danuser. Robust single particle tracking in live cell time-lapse sequences. *Nat. Methods*, 5(8):695, 2008.
- T. Kirubarajan, Y. Bar-Shalom, and K.R. Pattipati. Multiassignment for tracking a large number of overlapping objects. *IEEE Trans. Aerosp. Electron. Syst.*, 37(1):2–21, 2001.

- K. Li, E.D. Miller, M. Chen, T. Kanade, L.E. Weiss, and P. Campbell. Cell population tracking and lineage construction with spatiotemporal context. *Med. Image Anal.*, 12(5):546–566, 2008.
- E. Meijering, O. Dzyubachyk, I. Smal, and W.A. van Cappellen. Tracking in cell and developmental biology. *Semin. Cell Dev. Biol.*, 20(8):894–902, 2009.
- S. Mori, K.C. Chang, and C.Y. Chong. Performance analysis of optimal data association with applications to multiple target tracking. In Y. Bar-Shalom, editor, *Multitarget-Multisensor Tracking: Applications and Advances*, volume 2, pages 183–235. Artech House, Boston, 1992.
- J. Munkres. Algorithms for the assignment and transportation problems. *J. of the Soc. for Ind. and Appl. Math.*, 5(1):32–38, 1957.
- D. Musicki and R. Evans. Linear joint integrated probabilistic data association-LJIPDA. In *Proceedings of the 41st IEEE Conference on Decision and Control*, volume 3, pages 2415–2420, 2002.
- C. Nitschke, A. Garin, M. Kosco-Vilbois, and M. Gunzer. 3D and 4D imaging of immune cells in vitro and in vivo. *Histochem. Cell Biol.*, 130(6):1053–1062, 2008.
- D. Padfield, J. Rittscher, and B. Roysam. Spatio-temporal cell segmentation and tracking for automated screening. In *Proceedings of the 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 376–379, 2008.
- D. Padfield, J. Rittscher, and B. Roysam. Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis. *Med. Image Anal.*, 2010. ISSN 1361-8415. doi: 10.1016/j.media.2010.07.006.
- I.F. Sbalzarini and P. Koumoutsakos. Feature point tracking and trajectory analysis for video imaging in cell biology. *J. of Struct. Biol.*, 151(2):182–195, 2005. ISSN 1047-8477.
- K. Shafique and M. Shah. A noniterative greedy algorithm for multiframe point correspondence. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(1):51–65, 2005.

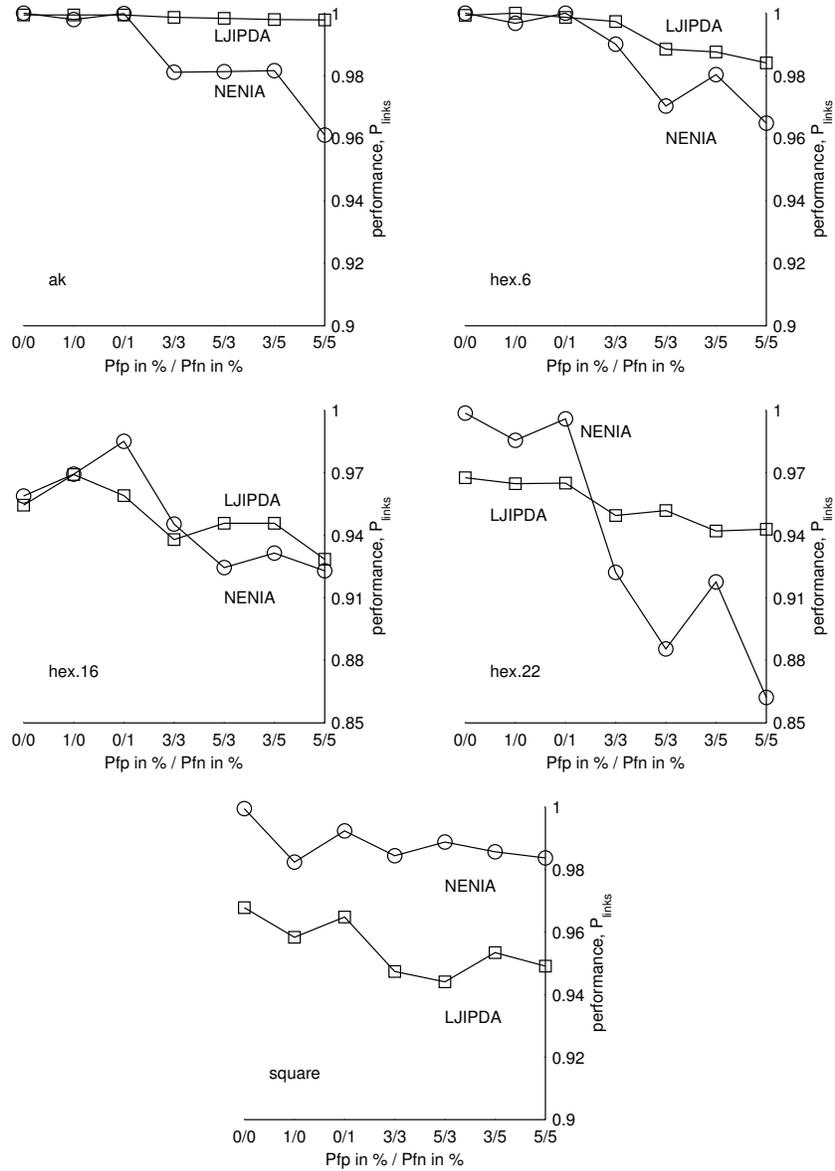


Fig. 8: Performance of NENIA compared with the performance of LJIPDA on different detection qualities. Over all cases, both algorithms have similar performance. NENIA outperforms LJIPDA on high detection qualities. The lines connecting points have no interpretation and are given for perceptual convenience. Note the different scales on the y-axes.

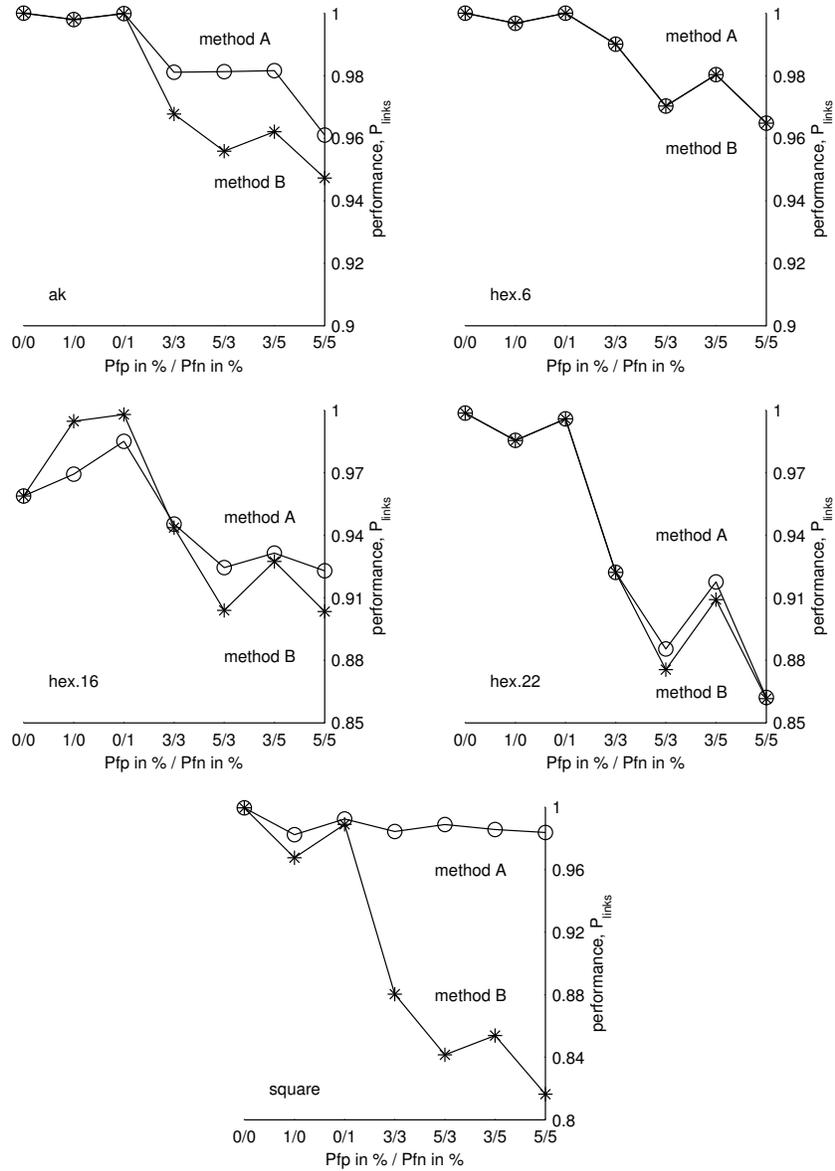


Fig. 9: Tracking performance of NENIA with method A compared with method B. On average, method A outperforms method B. Both methods recover more than 80% of cell moves correctly in all cases. The lines connecting points have no interpretation and are given for perceptual convenience. Note the different scales on the y-axes.

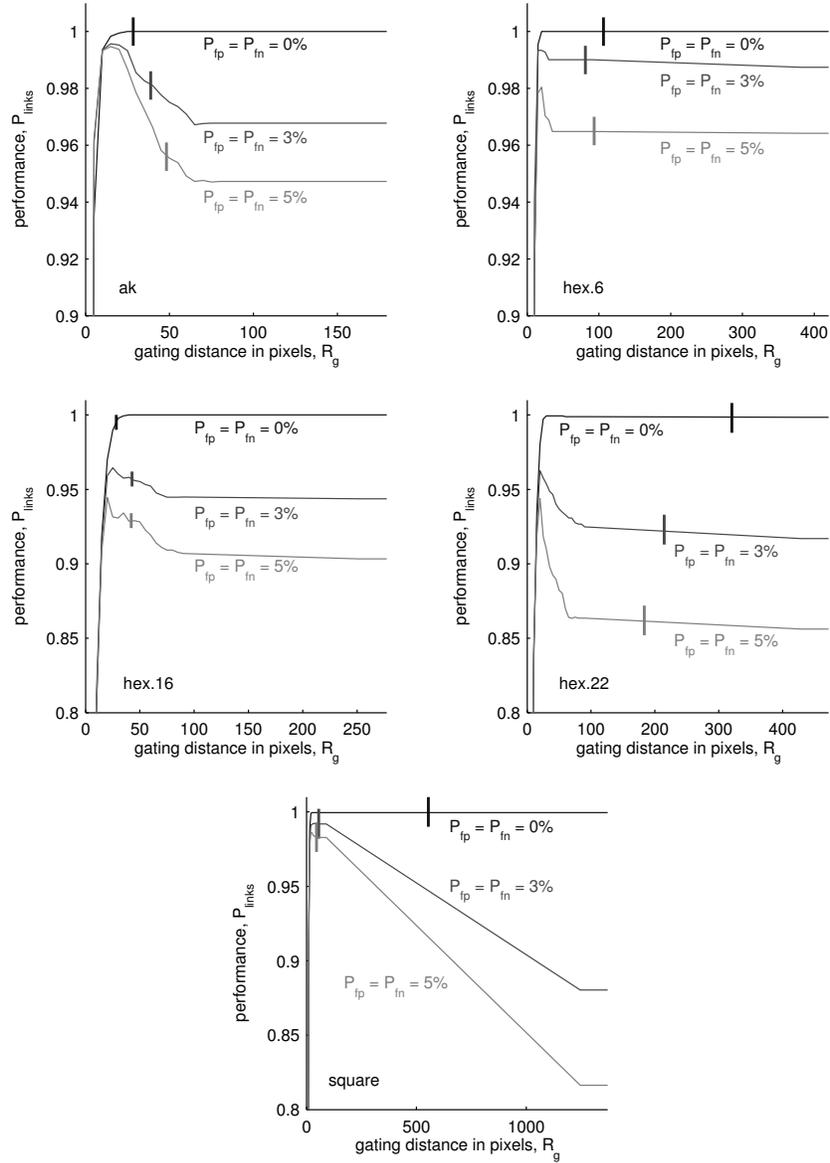


Fig. 10: Empirical cell tracking performance curve as a function of gating distance for detections with different qualities. Empirical performance curves support our hypothesis about the dependency of performance on the gating distance (compare with Fig. 6). Vertical strokes correspond to the gating distances estimated by a method A. Note the different scales in the top two diagrams (y-axes), and all diagrams (x-axes).

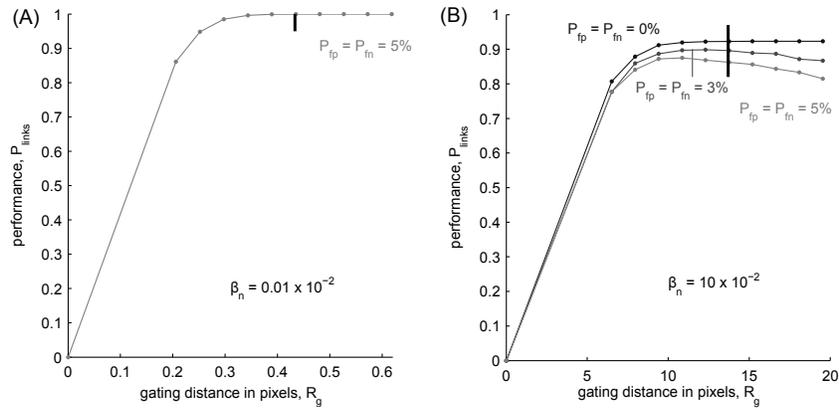


Fig. 11: Empirical performance curves as a function of gating distance obtained on a synthetic videos. When the normalized object density is low (left) the tracking performance remains high for different detection levels (the performance curves overlap, therefore only the case $P_{fp} = P_{fn} = 5\%$ is shown). When the object density is high (right), the performance depends on the detection quality. Dots correspond to actual data points. The vertical stroke marks the point $R_g = R_{tmax}$. At this point the performance is maximized or near the maximum. The empirical performance curves preserve prominent features of our analytical curve (compare with Fig. 8).