A BATCH ALGORITHM USING ITERATIVE APPLICATION OF THE VITERBI ALGORITHM TO TRACK CELLS AND CONSTRUCT CELL LINEAGES

Klas E. G. Magnusson, and Joakim Jaldén

ACCESS Linnaeus Center, Signal Processing Lab,
KTH Royal Institute of Technology, Stockholm, Sweden
klasma@kth.se, jalden@kth.se

ABSTRACT

Advances in microscope hardware in the last couple of decades have made it possible to acquire large data sets with image sequences of living cells grown in cell culture. This has led to a demand for automated ways of analyzing the acquired images. This article presents a new algorithm for tracking cells and constructing cell lineages in such image sequences. The algorithm uses information from the entire sequence to make local decisions about cell tracks and can therefore make more robust decisions than algorithms that process the data sequentially. It also incorporates image-based likelihoods of cell division and cell death into the tracking, without having to resort to separate detection algorithms or post processing of tracks. The algorithm consists of a scoring function to rank tracks and an iterative algorithm that searches for the highest scoring tracks, in a computationally efficient way, using the Viterbi algorithm.

Index Terms— Cell Tracking, Multiple Target Tracking, Data Association, Viterbi Algorithm, Dynamic Programming

1. INTRODUCTION

In cell biology, microscopy has always been an important tool to analyze the behavior of cells in response to different culture conditions and treatments. Proliferation and apoptosis (cell death) is often studied using microscopy, but more complex statistics, such as motion patterns and morphology, can also be analyzed [1]. It is particularly hard to analyze primary stem cell populations, as these are heterogeneous and one therefore needs to track a large number of cells to be able to draw conclusions about the populations. It is not until recently that advances in microscope hardware, software, and data storage have made it simple and feasible to acquire data sets large enough to analyze these parameters in a systematic fashion. Now, the main bottleneck is instead analyzing the images once they have been acquired [1]. In many cases, image sequences of cells have been analyzed successfully using manual approaches, often aided by a computer program to input and store information. This type of manual analysis is however very time consuming, the number of parameters that can be analyzed is limited, and the analysis procedures are hard to reproduce exactly. It is therefore of interest to do all of the analysis in a fully automated fashion using computer software. This requires automatically finding the cells in the images using image segmentation and finding their trajectories using target tracking. There are a number of prior publications that discuss both problems and describe complete systems for performing automated cell tracking, e.g., [2, 3, 4]. The most advanced system at this point in time is arguably that described in [3].

In this article we present a new algorithm for the tracking problem. The algorithm works independently of the segmentation algorithm and is therefore applicable in cell tracking problems for which adequate segmentation algorithms exist. The algorithm consists of a scoring function and an algorithm that searches for a local maximum of the scoring function by adding cells to a tentative cell lineage tree, using the Viterbi algorithm. The Viterbi algorithm has previously been used to add tracks in multiple target tracking problems [5]. However, we extend the approach so that our algorithm handles mitosis (cell division) and apoptosis (cell death), and we introduce a novel way of altering previously added tracks to prevent such tracks from blocking the creation of new ones. The algorithm also handles situations where multiple cells have merged into a single detection, an important problem seldom treated rigorously in cell tracking algorithms.

A precursor of the proposed algorithm was previously applied in a full scale biological study [6]. However, the technical description of the algorithm has not been published before. Although [6] already speaks for the applicability of the proposed algorithm, we include a brief experimental validation of the algorithm in Section 3.

2. THE PROPOSED CELL TRACKING ALGORITHM

Given a sequence of T microscope images indexed by t, each image with a set of N_t *detections* potentially containing zero, one, or multiple cells, the goal of the proposed cell tracking algorithm is to link these detections temporally and to create a cell linage tree that is consistent with the detections. A detection is here simply a set of pixels obtained by the *segmentation* algorithm (cf. Fig. 2 - 4). We will refer to the set of detections in image t as $\mathcal{D}_t = \{D_{t,1}, D_{t,2}, \ldots, D_{t,N_t}\}$, and the set of all detections as $\mathcal{D}$.

Ideally, each detection $D_{t,i}$ would only contain the pixels of a single cell, but due to the difficulty of the segmentation problem this is seldom possible to achieve in practice. Multiple cells can merge into a single detection, and debris or features in the background can give rise to false detections. Many of these errors can however be resolved during the tracking step, when information from other images in the sequence is considered. It is therefore important not to make hard decisions about how many cells each detection contains, in the segmentation step.

A set of cell tracks in a sequence of images can be represented as a directed, layered, graph F with labeled *nodes*. Every node in F represents a single cell in one of the images, and the label of the node specifies the detection that the cell is associated with in that image.

We would like to thank Penney Gilbert, Karen Havenstrite and Helen Blau at the Blau laboratory, Stanford University, for providing us with the image data used to evaluate the performance of the algorithm, and acknowledge funding from the Swedish Research Council (VR).

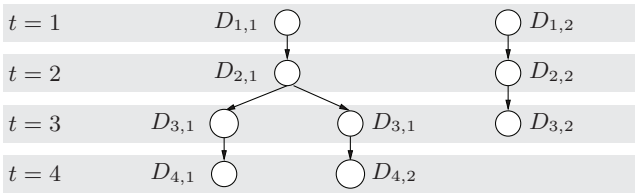


Fig. 1. Example graph F for cells in a sequence of length $T = 4$. Two cells are present in image $t = 1$, and they occupy detections $D_{1,1}$ and $D_{1,2}$ respectively. The cell in $D_{1,1}$ occupies $D_{2,1}$ in image $t = 2$ and divides between image $t = 2$ and $t = 3$, where the two daughter cells are segmented as one in $D_{3,1}$ and then occupy separate detections, $D_{4,1}$ and $D_{4,2}$, in image $t = 4$. The cell starting in $D_{1,2}$ dies in image $t = 3$.

The nodes of a cell are linked by directed edges between nodes in consecutive images. The last nodes of cells that divide are linked by two edges, to the first nodes of the two daughter cells. This results in an acyclic graph with one or more connected components, a type of graph normally referred to as a forest (i.e., a disjoint union of trees), as exemplified in Fig. 1.

For every image sequence with an associated set of detections $\mathcal{D}$ there will, as several cells can occupy a single detection, be an infinite set $\mathcal{F}$ of admissible forests. The objective of the cell tracking algorithm is to, among the admissible forests $\mathcal{F}$, find the forest $F^\star$ that is most consistent with $\mathcal{D}$. Given a *scoring function* $g(\mathcal{D}, F)$ that quantifies how well a hypothetical forest $F \in \mathcal{F}$ matches the set of detections $\mathcal{D}$, we can pose the cell tracking problem as the maximization problem

$$F^\star = \arg\max_{F\in\mathcal{F}} g(\mathcal{D}, F)\,. \tag{1}$$

As $g(\mathcal{D}, F)$ depends on the same $\mathcal{D}$ throughout, we will from now on (notationally) exclude this dependence and write $g(F)$.

The proposed cell tracking algorithm consists of the selection of $g(F)$ and the proposal of a computationally efficient maximization procedure that approximately solves (1). The function $g(F)$ is chosen so that ground truth forests created by cell biologists score highly, and so that the function satisfies a certain Markov property that facilitates efficient approximate solutions to (1).

2.1. The scoring function

If a cell is present in image t, there are a number of things that can happen to the cell in image $t + 1$. We assume that the possibilities are migration, mitosis, and apoptosis. In some applications, cells can also leave the field of view but in the data sets that we are interested in, the cells are confined in microwells (cf. Fig. 4), so this is not considered. The possible events motivate a scoring function that consists of 4 separate contributions, $g^{\mathrm{M}}(F)$, $g^{\mathrm{S}}(F)$, $g^{\mathrm{A}}(F)$ and $g^{\mathrm{C}}(F)$, connected to migration, mitosis, apoptosis, and the number of cells in every detection. The different contributions are added together to form the scoring function

$$g(F) = g^{\mathrm{M}}(F) + g^{\mathrm{S}}(F) + g^{\mathrm{A}}(F) + g^{\mathrm{C}}(F)\,. \tag{2}$$

The term in $g(F)$ connected to migration is given by

$$g^{\mathrm{M}}(F) = \sum_{t=1}^{T}\sum_{i=1}^{N_t}\sum_{j=1}^{N_{t+1}} g^{\mathrm{M}}_{t,i,j}(F) \tag{3}$$

where $g^{\mathrm{M}}_{t,i,j}(F) = \log p^{\mathrm{M}}_{t,i,j}$ if $M_{t,i,j}(F) = 1$ and $g^{\mathrm{M}}_{t,i,j}(F) = \log(1 - p^{\mathrm{M}}_{t,i,j})$ if $M_{t,i,j}(F) = 0$, where $p^{\mathrm{M}}_{t,i,j} \in (0, 1)$ models the *intrinsic belief* (probability) that a cell migration occurred between $D_{t,i}$ and $D_{t+1,j}$, and where $M_{t,i,j}(F) = 1$ indicates that there is a cell migration from $D_{t,i}$ to $D_{t+1,j}$ in F. In the example of Fig. 1, it follows that $M_{3,1,2}(F) = 1$ as a cell migrates from detection $D_{3,1}$ to $D_{4,2}$, but $M_{1,1,2}(F) = 0$ as no cell migrates from $D_{1,1}$ to $D_{2,2}$. Note also that by the term intrinsic belief, we refer to the belief that a cell migrates between $D_{t,i}$ and $D_{t+1,j}$, based only on the characteristics of $D_{t,i}$ and $D_{t+1,j}$ and no additional information from other detections or images.

The term $g^{\mathrm{S}}(F)$ connected to mitosis is defined analogously to $g^{\mathrm{M}}(F)$, with S in the place of M and $p^{\mathrm{S}}_{t,i,j} \in (0, 1)$ modeling the intrinsic belief that a parent cell in $D_{t,i}$ divides, so that the two daughter cells end up in $D_{t+1,j}$. Note that we by this construction implicitly assume that the daughter cells are always segmented as one in the image following a mitotic event. While the scoring function can be extended to avoid such an assumption, it simplifies the exposition and we find that the assumption is true for the data sets that we have been interested in. For apoptosis we have

$$g^{\mathrm{A}}(F) = \sum_{t=1}^{T}\sum_{i=1}^{N_t} g^{\mathrm{A}}_{t,i}(F) \tag{4}$$

where $g^{\mathrm{A}}_{t,i}(F) = \log p^{\mathrm{A}}_{t,i}$ if $A_{t,i}(F) = 1$ and $g^{\mathrm{A}}_{t,i}(F) = \log(1 - p^{\mathrm{A}}_{t,i})$ if $A_{t,i}(F) = 0$, where $p^{\mathrm{A}}_{t,i}$ models the intrinsic belief that there is a cell undergoing apoptosis in detection i of image t, and where $A_{t,i}(F) = 1$ indicates that such an event occurs in F. The term $g^{\mathrm{C}}(F)$ related to the cell counts is defined slightly differently from the above, and is given by

$$g^{\mathrm{C}}(F) = \sum_{t=1}^{T}\sum_{i=1}^{N_t} g^{\mathrm{C}}_{t,i}(F) \tag{5}$$

where $g^{\mathrm{C}}_{t,i}(F) = \log p^{\mathrm{C}}_{t,i,C_{t,i}}$. Here, $p^{\mathrm{C}}_{t,i,c}$ models the intrinsic belief that there are c cells in detection $D_{t,i}$, and $C_{t,i} = C_{t,i}(F)$ denotes the number of cells in F that occupy detection $D_{t,i}$.

In our implementation of the scoring function, and in particular in the experiment in Section 3, we compute the intrinsic beliefs of mitosis, apoptosis and cell counts using a multinomial logistic regression model [7] applied to a large number of detection features, e.g., detection area, circumference, and variation in pixel intensities. The intrinsic beliefs of migrations are based on a Brownian motion assumption and computed from a symmetrical Gaussian distribution centered around the position of the detection in image t. This motion model (and the scoring function as a whole) satisfies the Markov property that the score associated with the way in which a cell leaves a detection does not depend on how the cell got there.

2.2. The search algorithm

For simplicity of exposition, we first describe a version of the search algorithm where chains of nodes (i.e., cell tracks) are added to a tentative forest F without changing any of the preexisting nodes or edges in F. Then we briefly describe how to incorporate changes to the existing forest F into the search. At each iteration, the algorithm starts with a given forest $F \in \mathcal{F}$ and extends F by adding a single cell. This can be done either by adding the cell track as a branch from an existing cell track, corresponding to a mitotic event giving rise to two new cells, or as a new track not linked to any existing cells, corresponding to a new cell being introduced in the first image. The

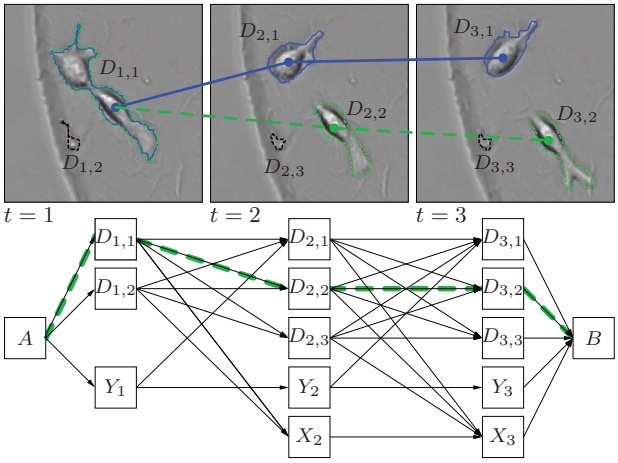


Fig. 2. Trellis diagram for the addition of a cell to a forest F with one existing cell (in blue), in an image sequence of length $T = 3$. States associated with detections are labeled $D_{t,i}$, "born later" is labeled Y_t, "dead" is labeled X_t, and the starting and stopping states are labeled A and B respectively. Structural constraints disallow certain states and arcs, e.g., the cell is not included if it is dead already in the first image, which explains the absence of X_1, and cannot appear by mitosis in a detection that currently contains no cells, explaining the absence of arcs from Y_2 to $D_{3,2}$ and $D_{3,3}$. Note however that even with these restrictions, relatively few detections, and with the modest sequence length of $T = 3$, there are a total of 32 different paths from A to B and thus 32 different ways of adding a single cell. The (arguably) best way of adding a second cell, as having been segmented jointly with the first cell in detection $D_{1,1}$ at $t = 1$, and then separately in detections $D_{2,2}$ and $D_{3,2}$ is illustrated by the dashed green line in the image sequence and in the corresponding path through the trellis diagram.

algorithm is started with an empty forest, and terminated once the scoring function cannot be increased by the addition of a single cell. Although not immediately obvious, it turns out that a single cell can be both efficiently and optimally added to an existing forest F. The operation is optimal in the sense that no other cell added to F would lead to a greater increase in the scoring function $g(F)$. The key to accomplishing this is to represent all possible ways of adding a cell in terms of a state space diagram known as a *trellis*.

In image t, the added cell can either be associated with one of the N_t detections, it can be dead, or it can be born later (implying that it will appear later due to mitosis). These possibilities are represented as $N_t + 2$ different states that the cell can be in in image t. A starting state A before $t = 1$, and a stopping state B after $t = T$, are also introduced. All possible ways of adding a new cell are now compactly described by linking the states of consecutive images with arcs representing allowed transitions between the states. To exemplify the concept, Fig. 2 shows the trellis associated with adding a cell to an image sequence with $T = 3$ images and one preexisting cell.

Each arc in the trellis diagram has a score reflecting the change in $g(F)$, associated with choosing a path though the arc. Finding the highest scoring track to add to F is equivalent to finding the highest scoring path from A to B, where the score of a path is the sum of its arc scores. For example, following an arc from the starting state A to a detection $D_{1,i}$ in the first image is equivalent to proposing an

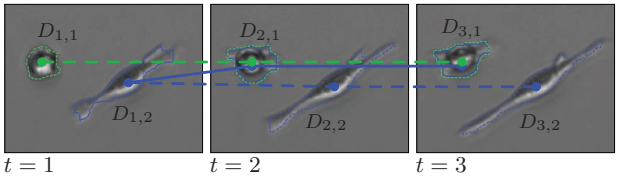


Fig. 3. Swap operation between $t = 1$ and $t = 2$. The dashed green track is first added to $D_{1,1}$ and then a swap is made so that the track takes over the states $D_{2,1}$ and $D_{3,1}$ from the preexisting blue, solid track. After the swap, the Viterbi algorithm switches to finding a new optimal track for the blue cell (dashed blue line), from $t = 1$.

increase in the cell count for $D_{1,i}$ by 1, and the arc score is consequently $\Delta g = \log p^{\mathrm{C}}_{1,i,C_{1,i}+1} - \log p^{\mathrm{C}}_{1,i,C_{1,i}}$ (cf. (5)). The scores for other arcs, involving migration, mitosis, and apoptosis, are slightly more complicated but similar in principle and we omit the details for brevity. However, we hasten to add that the ability to decompose the scoring function into additive arc scores is critically dependent on the Markov property of the scoring function introduced in Section 2.1. The Markov property, along with the trellis structure of the state space diagram, allows a dynamic programming algorithm – known as the Viterbi algorithm [8] in the communication literature – to find the optimal way of adding a cell, with a complexity that is linear in T, as opposed to trying all possibilities which would be exponential in T and infeasible for longer sequences. The principal observation behind the Viterbi algorithm is that of all incoming paths to a state at some time t, only the currently highest scoring path can be extended to an optimal path through the trellis. In our application, the trellis changes as a function of the forest F for each added cell, which is unusual in that most applications have a fixed trellis, but this does not hinder the application of the Viterbi algorithm. We emphasize however that this approach is only guaranteed to find a local maximum of (1), even though each iteration is optimal in the sense described above. Nevertheless, the ability to incrementally build the final forrest and iteratively update the scoring function greatly reduces the computational complexity of the algorithm.

A problem with the iterative algorithm as described above is that sometimes a cell added in an earlier iteration will occupy a detection that we would like to add a new cell track to. This type of situation, where a preexisting track blocks the creation of good tracks later, is a well known problem in iterative multiple target tracking algorithms [5], and is not fully solved by allowing several cells to occupy a single detection. To address this problem, we propose to add the option of moving an existing track somewhere else and let the new cell take over the old track of the cell that is moved, along with any daughter cells that this cell might have (cf. Fig. 3). After all, this path through the state space was optimal when the old track was added. The next state in the path through the trellis will be the new state in image $t+1$ of the cell that is being moved and all of the following states will also be associated with this cell and not the cell that was first added. If there are M_t cells that can be moved to any of the detections in image $t + 1$, this swap operation can be directly incorporated in the trellis by adding M_t additional arcs between every pair of detections $D_{t,i}$ and $D_{t+1,j}$. Traversing the trellis through such an arc implies that a swap is made. This mechanism is a crucial part of the algorithm and one of the extensions necessary for the ideas in [5] to be applicable in cell tracking problems. One could easily allow swapping of more than 2 tracks by instead solving a combinatorial assignment problem in each step, but that would require more computation.

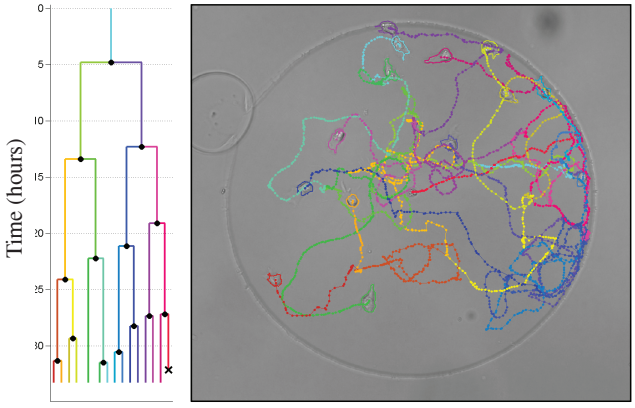


Fig. 4. Cell tracks plotted on top of the last image in a sequence, cell outlines in this image, and the cell lineage tree for the full sequence.

3. APPLICATION EXAMPLE

We evaluated the performance of the proposed tracking algorithm on a data set with 115 image sequences of Muscle Stem Cells (MuSCs). The cells were imaged every 3 minutes for 33 hours with a Zeiss PALM/AxioObserverZ1 microscope using phase contrast microscopy at 10x magnification with a resolution of 1388x1040 pixels. The cells were spatially segregated in 600 μm microwells, so that no cells could enter or leave the field of view. The same data set was used in [6] to study the effects of different substrate rigidities on MuSCs, and more details about the cells and the culturing conditions can be found in [6]. To train the classifiers used to produce the intrinsic beliefs in the scoring function we computed 89 features for all detections in a separate, manually corrected data set with 55 image sequences acquired during 23 hours, with the same imaging conditions.

The image sequences were stabilized, using the image stabilization plugin for ImageJ, to remove camera shake introduced by errors in the positioning of the stage. Stationary background objects were then removed by the subtraction of background images, computed as the pixelwise median through the time dimension of the sequences. Finally, the segmentation was done by thresholding the local variation in pixel intensity, as advocated in [9]. The tracking algorithm described in Section 2 was applied to the segmented image sequences, producing a set of tracks and a lineage tree for every sequence (cf. Fig. 4).

To evaluate the performance of the proposed tracking algorithm we compared the computer generated tracks, for the 115 image sequences described above, to manually created ground truth trajectories. To investigate how well the computed tracks agree with the true cell trajectories, we computed the track purity, which is a measure of how well computed tracks follow cells, and the target effectiveness, which measures how well cells follow computed tracks. Both measures are properly defined in [10]. The track purity and target effectiveness were found to be 0.81 and 0.88, which is almost identical to the values 0.81 and 0.87 found in [10]. This shows that the proposed algorithm tracks the cells accurately, although the results do not rank our algorithm with respect to the algorithm in [10], as different data sets were used.

To evaluate how accurately the tracking algorithm detects mitosis and apoptosis, we computed the precision and recall with which the events were detected. The precision and recall were computed to 0.78 and 0.75 for mitosis 0.74 and 0.62 for apoptosis. In [10] the mitosis recall (called mitosis branching correctness therein) was found to be 0.65. As in the computation of the mitosis recall in [10], mitotic and apoptotic events were considered correct if there was a corresponding ground truth event within 50 minutes of the detected event.

4. CONCLUSION

We have proposed a multiple target tracking algorithm for tracking biological cells in microscope images. The algorithm performs global data association by iterative application of the Viterbi algorithm and is able to use information from the entire image sequence to make local decisions about cell tracks. The algorithm was shown, through evaluation on a test data set with MuSCs imaged using phase contrast microscopy, to have good tracking performance and a good ability to detect mitosis and apoptosis.

5. REFERENCES

[1] C. Zimmer, B. Zhang, A. Dufour, A. Thébaud, S. Berlemont, V. Meas-Yedid, and J.-C. O. Marin, "On the digital trail of mobile cells," *IEEE Signal Proc. Magazine*, vol. 23, no. 3, pp. 54–62, May 2006.

[2] O. Al-Kofahi, R.J. Radke, S.K. Goderie, Q. Shen, S. Temple, and B. Roysam, "Automated cell lineage construction: A rapid method to analyze clonal development established with murine neural progenitor cells," *Cell Cycle*, vol. 5, no. 3, pp. 327–335, Feb. 2006.

[3] T. Kanade, Z. Yin, R. Bise, S. Huh, S. Eom, M. F. Sandbothe, and M. Chen, "Cell image analysis: Algorithms, system and applications," in *Proc. IEEE Workshop on Applications of Computer Vision (WACV)*, 2011, pp. 374–381.

[4] D. House, M. L. Walker, Z. Wu, J. Y. Wong, and M. Betke, "Tracking of cell populations to understand their spatio-temporal behavior in response to physical stimuli," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 186–193.

[5] J. Berclaz, F. Fleuret, and P. Fua, "Robust people tracking with trajectory optimization," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2006, pp. 744–750.

[6] P. M. Gilbert, K. L. Havenstrite, K. E. G. Magnusson, A. Sacco, N. A. Leonardi, P. Kraft, N. K. Nguyen, S. Thrun, M. P. Lutolf, and H. M. Blau, "Substrate elasticity regulates skeletal muscle stem cell self-renewal in culture," *Science*, vol. 329, no. 5995, pp. 1078–1081, 2010.

[7] D. W. Hosmer and S. Lemeshow, *Applied logistic regression*, Wiley Series in Probability and Statistics, 2 edition, 2000.

[8] G. D. Forney, Jr., "The Viterbi algorithm," *Proceeding of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[9] K. Wu, D. Gauthier, and M.D. Levine, "Live cell image segmentation," *Biomedical Engineering, IEEE Transactions on*, vol. 42, no. 1, pp. 1–12, 1995.

[10] R. Bise, Z. Yin, and T. Kanade, "Reliable cell tracking by global data association," in *Proc. IEEE Int. Symp. on Biomedical Imaging (ISBI): From Nano to Macro*, 2011.