Robust Estimation of Stem Cell Lineages Using Local Graph Matching

Min Liu, Amit K. Roy-Chowdhury *
Department of Electrical Engineering
University of California, Riverside

G. Venugopala Reddy †
Department of Botany and Plant Sciences
University of California, Riverside

Abstract

In this paper, we present a local graph matching based method for tracking cells and cell divisions. This will allow us to estimate the lineages of the cells in a 4D spatio-temporal image stack obtained using fluorescence imaging techniques. We work with plant cells, where the cells are tightly clustered in space and computing correspondences in space and time can be very challenging. The local graph matching method is able to compute the lineages even when significant portions of the images are corrupted due to sensor noise in the imaging process or segmentation errors. The geometric structure and topology of the cells' relative positions are efficiently exploited to solve the tracking problem using the local graph matching technique. The process not only computes the correspondences of cells across spatial and temporal image slices, but is also able to find out where and when cells divide, identify new cells and detect missing ones. Using this method we show experimental results to track the properly segmented cells, and compute cell lineages from images captured over 72 hours, even when some of those images are highly noisy (e.g., missing cells).

1. Introduction

Local spatio-temporal co-ordination of cell growth and cell division is important for proper development of organs in both plant and animal systems. The subject of this study, the shoot apical meristems (SAMs) also referred to as the stem-cell niche, is the most important part of the plant body plan because it supplies cells for all the above ground plant parts such as leaves, branches and stem, and at the same time maintains its stable size. Therefore, a tight spatio-temporal co-ordination between cell division and differentiation of progeny cells into organs is critical to maintain the stability of SAMs. However, the causal link between cell growth and cell division and how they, in turn, affect organ formation is not well understood. This is because of the lack of quantitative measurement of cell growth patterns. Therefore, the development of computational platforms capable of automatically tracking cells and cell division patterns from fluorescent 4D (3D spatial slices + time) images acquired by using laser scanning confocal microscopy is very important. It will lead to computational models to explain the causal relationships between cell deformation dynamics, cell growth and cell division patterns. This is a computational challenge that has universal application to all developmental fields, including both animals and plants.

In this work, we present a method to automatically compute cell lineages (i.e., cell tracks through cell divisions) from the 4D image stacks obtained over days of continu-

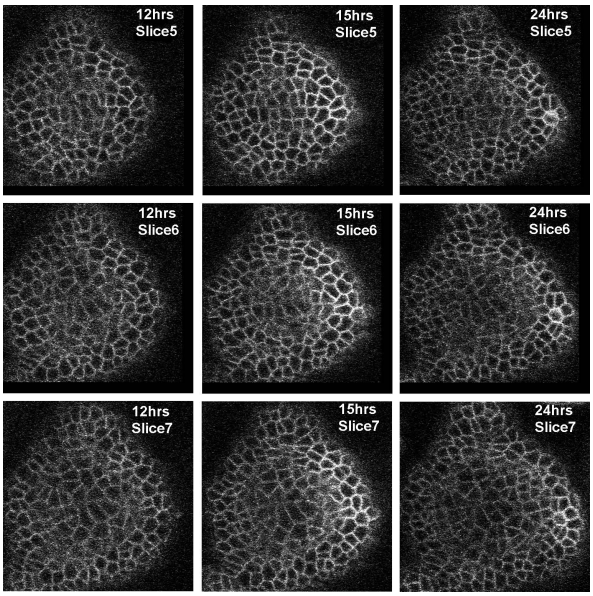

Figure 1. Examples of three image slices (vertical) at three time instants (horizontal). Cells in the central part of some images are missing. Images along the vertical axis are separated by 1.5 um, while the size of a single cell is about 5 um in diameter; therefore certain cell could be detected in 3 consecutive slices, which means that we could integrate the tracking results in 3 such slices.

*This work was partially supported by NSF Grant IIS-0712253.
†This work was also partially supported by NSF Grant IOS-0718046.

ous observations of the growth of the plant stem cell niche. The SAMs of model plant *Arabidopsis thaliana* consist of approximately 500 cells locating in multiple layers. We observed the same set of SAMs repeatedly for 3 days at 3 hour intervals. A few examples of images used in this study are given in Figure 1. The images obtained from live-imaging study can be noisy (due to inherent limitations of imaging physics); thus, the tracking method has to be robust enough to account for this noise. Also, the cells are highly clustered together and it is very challenging to maintain their individual correspondences over extended periods of time.

1.1. Proposed Approach and Relation to Existing Work

Given sets of segmented cells at different time instants, cell tracking is essentially a kind of point matching problem, which has been widely studied. One of the most popular solutions is the Softassign Procrustes algorithm [2, 7], which has been applied to compute cell lineages, and has been improved to take care of cell divisions [5]. The Softassign method uses the information of point location, and solves the global correspondence problem as well as the affine transformation problem between two point sets iteratively at the same time. It however does not take into account the information about the topological inter-relationship between the points, especially the cells' local neighborhood structures [11], which are very important for tracking similarly shaped cells in close proximity and in noisy images, like those in Figure 1. By considering this inter-relationship, we are able to handle high level of noise in the imaging and segmentation process.

In this paper, we propose a graph matching method, where instead of finding the correspondences of cells globally, we would like to solve the matching problem *locally* cell by cell, through local graph matching techniques [3, 4]. We will exploit the local geometrical and topological structure of the relative position of the cells. By matching the relative position information of cells with respect to their nearest neighbors, such as the length and orientation of the edges in the local graph, we find the most similar cell pair. Starting from this seed pair, we grow the number of matched cells by computing the similarities between local regions in the graph [8]. The proposed method has greater robustness over the global matching methods [5], especially when there are missing cells due to noise in imaging, as shown in Figure 1, or poor segmentation, as shown in Figure 8. The local graph matching method can be very flexible also. In situations, where it is hard to find the first corresponding cell pair (seed) automatically, we can identify the first corresponding pair, and then continue the rest of the tracking process. (In our application, the seeds were chosen automatically by the algorithm.) In fact, the local graph matching technique is motivated by object recognition approaches

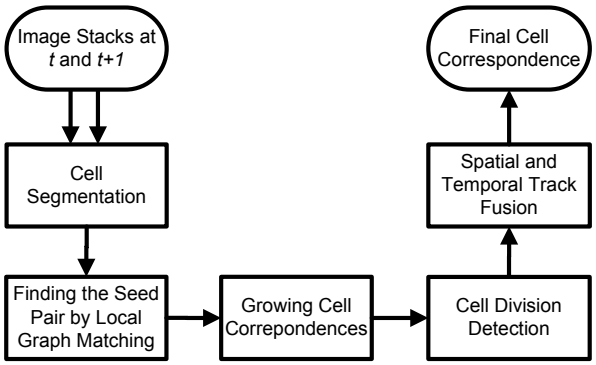

Figure 2. Diagrammatic representation of local graph-matching based cell tracking for two time-consecutive image stacks.

that rely on local feature matching to overcome the problems of noise and partial occlusion [3, 10].

In the context of tracking through cell divisions, level set approach is another possible method. However, it is difficult to apply when the cells are in close contact and it is hard to distinguish a cell from its neighboring cells because they have similar shapes and almost the same features, and when large parts of the image may be missing in one instance due to sensor noise. This is why method like [6] can not be adapted to our problem. Although the level set method is not suitable for cell tracking in our case, we can adopt it for cell segmentation where the objective is not to find correspondences [1].

Another unique feature of our approach is that we can obtain the tracks by integrating the tracking results over the entire 4D image stack, unlike methods which do single-layer tracking or surface-layer analysis only [5] [8]. This allows us to get better tracking results since each cell is typically imaged in more than one slice at each time point.

1.2. Broad Overview of Solution Strategy

The level set segmentation will provide information like the centroids of the cells, their area, orientation and so on to the tracker. The local graph matching based tracking for any two time-consecutive cell image stacks can now be described as follows, a diagrammatic representation of which is given in Figure 2.

1. Find the seed cell pair by local graph matching.
2. Starting from this seed pair, find the correspondences of the neighboring cells by computing the similarity of local regions in the graph.
3. Detect cell divisions by detecting changes in the topology of the graph.
4. Fuse correspondence information across both time and slices to obtain cells' tracks in the 4D image stack.

2. Detailed Methodology

2.1. From Cells to Graphs

We describe here how to create the graphical abstraction given a collection of cells in an image. Every cell is represented by a vertex in the graph and neighboring vertices are connected by an edge. We regard those cells that are within a certain distance around cell c as its neighboring cells, denoted by the neighborhood cell set $N(c)$, which is ordered in the counter-clockwise direction. An example of the graph superimposed on the cells is shown in Figure 3.

2.2. Local Graph Matching Method

The graph structure automatically includes the relative position information of the cells, such as the relative distance between two neighboring cells (the edge length) and the edge orientation. Usually, if there is no cell division and the images are not corrupted, the local graph topology should not change. When this condition is satisfied, our tracking method can find the corresponding cell pairs by matching their related local graphs. The case where the graph structure changes due to cell divisions is described in the later section.

Given two local graphs G_1 and G_2, corresponding to cell c_i at time t and c_j at time $t+1$ respectively, the distance measure $D_L(c_i, c_j)$ for these two graphs is composed of three parts: the normalized difference of the edge lengths at consecutive time instants t and $t+1$, the difference of the orientation angles between the edges, and the cells' normalized location difference information. This distance measure can then be mathematically expressed as

$$
\begin{aligned}
D_L(c_i, c_j) = & \\
& \sum_{c_{k_i} \in N(c_i), c_{k_j} \in N(c_j)} \frac{\left(l_{c_{k_i}, c_i}(t) - l_{c_{k_j}, c_j}(t+1) \right)^2}{l_{c_{k_i}, c_i}(t)^2} \\
& + \lambda \cdot \sum_{c_{k_i} \in N(c_i), c_{k_j} \in N(c_j)} \left(\theta_{c_{k_i}, c_i}(t) - \theta_{c_{k_j}, c_j}(t+1) \right)^2 \\
& + \frac{\| \underline{\mathrm{P}}_{c_i}(t) - \underline{\mathrm{P}}_{c_j}(t+1) \|^2}{\Delta^2} ,
\end{aligned}
\tag{1}
$$

where c_{k_i} is a neighboring cell of c_i, and c_{k_j} is a neighboring cell of c_j, $l_{c_{k_i}, c_i}(t)$ and $l_{c_{k_j}, c_j}(t+1)$ are the edge lengths, $\theta_{c_{k_i}, c_i}(t)$ and $\theta_{c_{k_j}, c_j}(t+1)$ are the orientation angles in radians of the edges measured relative to a horizontal axis, $\underline{\mathrm{P}}_{c_i}(t)$ and $\underline{\mathrm{P}}_{c_j}(t+1)$ are the cell position vectors, and Δ is the average distance between two neighboring cells. If two local graphs match (i.e., the distance measure is small), we can say that the central cells (i.e., c_i and c_j) in those two local graphs are a corresponding cell pair. This process is depicted diagrammatically in Figure 3.

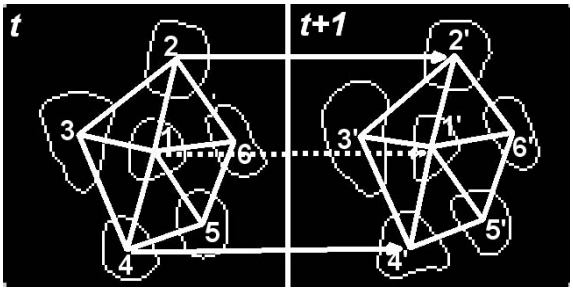

Figure 3. The matched local graphs G_1 at t and G_2 at $t+1$, the correspondence of the seed cell pair (1,1'), as well as the correspondences of the neighboring cells, such as (2,2') and (4,4').

2.3. Finding the seed (first) cell pair

Although the local graph matching method is effective in finding some corresponding cell pairs, it is based on the assumption that the local graph topology does not change along time, and the geometry of the local graph changes little. However, this is not always true, since the cells are growing and every cell may not be imaged properly, so our tracking strategy is to first find the most similar cell pair, and then starting from it, to grow the matching process spatially. We now describe how to find the first cell pair.

Let us assume that the first image has N cells, and the second image has M cells. According to the distance function in Equation (1), we can get the distance value $D_L(c_i, c_j)$ of every cell pair (c_i, c_j), and pick the most similar cell pair (c_a, c_b) that satisfies

$$
D_L(c_a, c_b) = \min_{c_i, c_j} \{ D_L(c_i, c_j), i = 1, \cdots N, j = 1, \cdots M \} .
\tag{2}
$$

Alternatively, we could find the top few most similar cell pairs instead. Then the tracking process will be faster, because we can start from more corresponding cell pairs (more seeds). This procedure is used in this paper. However, in some situations, it may be better to find the seed manually, which can be easily integrated into our approach.

2.4. Growing cell correspondences from a seed pair

Given a matching cell pair, we now describe the process to compute matches for the neighboring cells. This distance function is composed of two parts: the normalized difference of the edge lengths and the difference of the orientation angles between the edges. Let c_i and c_j be two correctly matched cells respectively at time instants t and $t+1$, and c_{k_i} and c_{k_j} be two neighboring cells around them respectively. Then the distance measure function is

$$
\begin{aligned}
D_N(c_i, c_j; c_{k_i}, c_{k_j}) = & \frac{\left| l_{c_{k_i}, c_i}(t) - l_{c_{k_j}, c_j}(t+1) \right|^2}{l_{c_{k_i}, c_i}(t)^2} \\
& + \lambda \cdot \left| \theta_{c_{k_i}, c_i}(t) - \theta_{c_{k_j}, c_j}(t+1) \right|^2 .
\end{aligned}
\tag{3}
$$

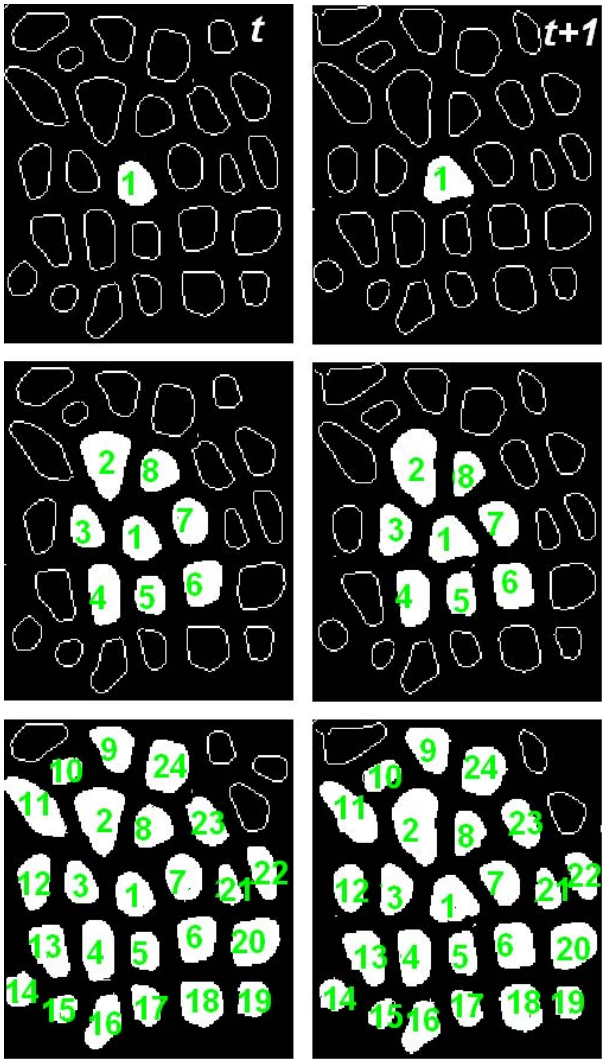

Figure 4. Growing cell correspondences from a seed to its neighbors. The top row denotes the seed cell pair, the middle row denotes the tracking results after the first step of the recursion of growing the correspondences, while the bottom row denotes the tracking results after the second step.

If the distance is small, c_{k_i} and c_{k_j} are identified as matching cells, and the process continues. This leads to a recursive procedure for matching cells from a seed to the neighbors. The process is illustrated in Figure 4 where the seeds identified in the top row are used to grow the tracking spatially using local graph matching.

2.5. Cell division detection

The topology of a certain local graph will change if a cell divides. We now explain how cell divisions are detected in the local graph matching approach. Take the local graphs illustrated in Figure 5 for example. Here a cell M at time t is divided into two sibling cells A and B at time $t+1$.

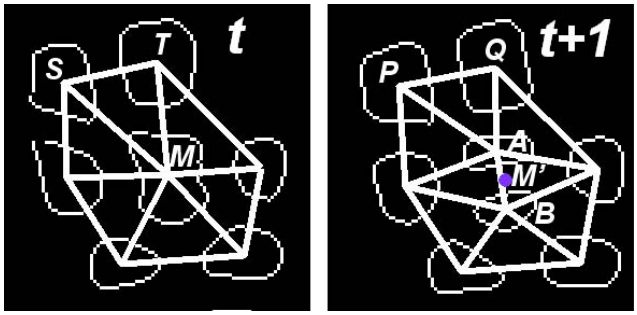

Figure 5. A diagram to describe the relative position of the mother cell and the daughter cells.

We can first find the cell pair (M, A) through the correspondence growing procedure starting from cell pair (T, Q) and/or (S, P). Then we check the area difference of cells M and A. If the difference is over about half of the area of an adult cell, then we could say that there maybe a cell division event, and cell M may be the mother cell and cell A may be one of the daughter cells. If there is really a cell division, we should be able to find the other sibling cell in the neighboring cells of A. Specifically, we can search among the neighborhood around the original location M' of the mother cell at time $t+1$, and the cell B with the shortest distance from M' should be the candidate for the other sibling cell. Furthermore, this distance $|BM'|$ should be within about half of the average distance between two neighboring adult cells.

Formally, the procedure for detecting the cell division is as follows:

1. Find the corresponding cell pair (M, A) using the local graph matching technique described in Section 2.4;
2. Check whether the area difference of (M, A) exceeds half the area size of an adult cell; if true, then (M, A) is a candidate pair of mother cell and sibling cell;
3. Search in the neighborhood of the daughter candidate A to find the other daughter candidate (sibling) cell B;
4. Compare the sibling cell candidates A and B; if the sum of their areas is about the same as the mother cell M, then we can say that M divides into two daughter cells A and B. We can also check the distance between A and B, which should be within about half the average distance between two neighboring adult cells.

Tracking results with cell divisions are shown in Figure 7.

2.6. Correspondence across image slices

Using the local graph matching methods described above, we can compute correspondences across slices S_k

and S_{k+1} at the same time instants, i.e., correspondences within the same stack. The distance measure function is

$$D_S(c_i, c_j) = \sum_{c_{r_i} \in N(c_i), c_{r_j} \in N(c_j)} \frac{\left(l_{c_{r_i},c_i}(S_k) - l_{c_{r_j},c_j}(S_{k+1})\right)^2}{l_{c_{r_i},c_i}(S_k)^2} + \lambda \cdot \sum_{c_{r_i} \in N(c_i), c_{r_j} \in N(c_j)} \left(\theta_{c_{r_i},c_i}(S_k) - \theta_{c_{r_j},c_j}(S_{k+1})\right)^2 + \frac{\|\underline{\mathrm{P}}_{c_i}(S_k) - \underline{\mathrm{P}}_{c_j}(S_{k+1})\|^2}{\Delta^2} + \left|\frac{A_{c_i}(S_k) - A_{c_j}(S_{k+1})}{A_{c_i}(S_k)}\right|^2, \quad (4)$$

where $A_{c_i}(S_k)$ and $A_{c_j}(S_{k+1})$ are the area size of cells c_i and c_j respectively. Because in this case there should be no cell divisions, we add the cell area difference into the distance function to make it more accurate in finding the correspondences across slices. Tracking results across slices, with high sensor noise are shown in Figure 8.

2.7. Fusion of spatial and temporal tracks

We now show how to fuse the spatial and temporal correspondences in order to get the final cell tracks. As explained in Figure 1, every cell usually occupies 3 slices, so we need to integrate the tracking results in every 3 consecutive slices. The fusion process is described in Figure 6 and below.

In the tracking process, we build a correspondence matrix C to denote whether two cells c_i, c_j in two images (may be at different times or different slices at the same time) are the same cell or not. If the two cells are the same, we set the correspondence value $C(i, j)$ to 1, otherwise we set it to 0. If cell c_i splits into cells c_{j_1} and c_{j_2}, we will set $C(i, j_1)$ and $C(i, j_2)$ to 2.

Let us denote the correspondence across time instants t and $t + 1$ at slice S_k as $C_k(t, t + 1)$, and the correspondence across slices S_k and S_{k+1} at time t as $C_t(k, k + 1)$. The fusion process is composed of the upward fusion matrix $C_{k-1,k}(t, t+1)$ and downward fusion matrix $C_{k,k+1}(t, t+1)$, which can be easily proved to be products of the individual correspondence matrices as follows:

$$C_{k-1,k}(t, t+1) = C_t(k, k-1)C_{k-1}(t, t+1)C_{t+1}(k-1, k);$$
$$C_{k,k+1}(t, t+1) = C_t(k, k+1)C_{k+1}(t, t+1)C_{t+1}(k+1, k). \quad (5)$$

The final set of the correspondences at slice S_k, denoted by $C^*_k(t, t + 1)$, is obtained as follows:

1. $C^*_k(t, t + 1) = C_k(t, t + 1)$;
2. For any new cell pair identified by $C_{k-1,k}(t, t + 1)$ or $C_{k,k+1}(t, t + 1)$, but not in $C_k(t, t + 1)$, add this pair to $C^*_k(t, t + 1)$;
3. For any cell pairs identified in $C_{k-1,k}(t, t + 1)$ and $C_{k,k+1}(t, t + 1)$, but with conflicting results (i.e., the matching cell pairs have one cell in common but the other cell is different in the upward fusion and downward fusion matrices), do not add to $C^*_k(t, t + 1)$.

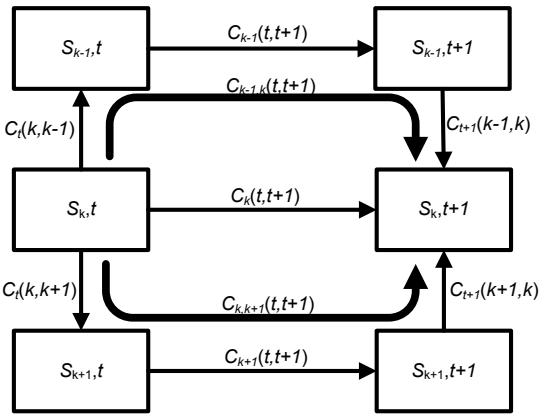

Figure 6. The diagram showing integration of the tracking results in 3 consecutive slices by fusing the spatial and temporal tracks to compute the final cell correspondence.

With this fusion process, we can find new cell pairs, which are not identified by single slice tracking, but can be found through the adjacent slices. With this process, we can track most of the properly segmented cells. The effect of the fusion in identifying additional cells is shown in Figure 9.

3. Experimental Results

We have tested our method on three data sets of SAMs. For limitation of space, we show detailed results on only one of them. The experimental results are shown on 4D cell image stacks obtained from plant cells and observed along 24 consecutive time instants, with the time interval of 3 hours between two consecutive instants. The cells were segmented by existing level set segmentation methods [1]. There are about 500 cells located in different layers. Registration is done by the existing method such as the alignment method of Maximization of Mutual Information [9]. The code for the tracker will be released publicly with this paper.

Choice of the parameters in the tracking system: The average distance between two neighboring cells is about 50 pixels, so if the relative distance of two cells is within 50 pixels, they are regarded as neighbors in the implementation. On the average, if two cells are the same cell in different images, the normalized length difference of the same edge is less than 0.1, the orientation difference of the same edge is within 0.17 radians (10°), the cells' normalized location difference is within 0.2, and the normalized cell area difference is within 0.2. Based on these and the fact that one cell usually has 5 to 8 neighbors, we set $\lambda = 0.33$ to normalize the distance measure for different physical parameters.

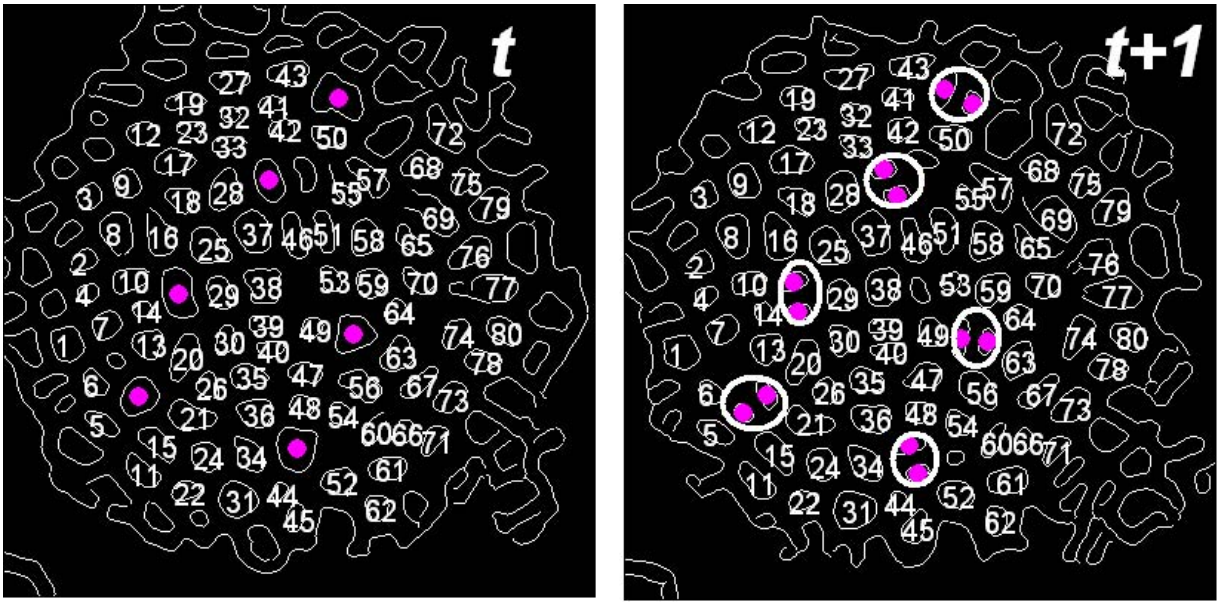

Figure 7. Tracking results with cell division, where the mother cells are denoted by purple dots in the first frame, while the sibling cells are in white circles in the second frame.

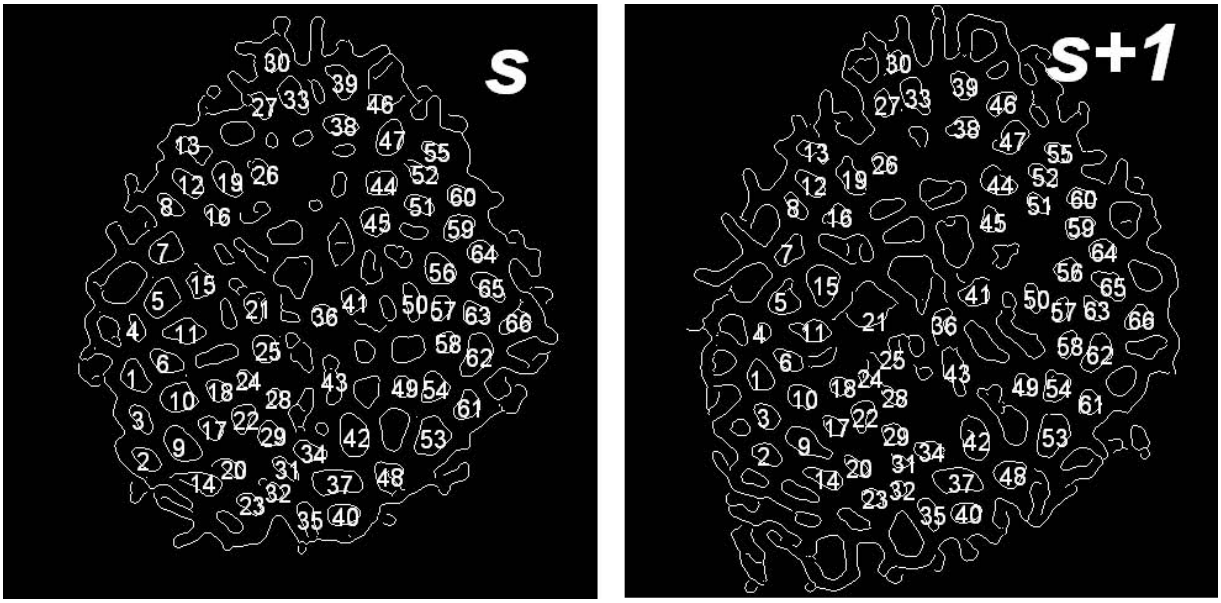

Figure 8. Tracking results across slices in noisy cell images, where some cells in the central part are totally missing.

If two cells are considered to be the same one, the distance measure value D_L should be less than 0.2 in Equation (1), D_N less than 0.02 in Equation (3), and D_S less than 0.24 in Equation (4). The average cell area is 400 pixel^2, so the area of the divided daughter cell is about 200 pixel^2. If two matched cell pair's area difference is beyond 200 pixel^2, we consider that there may be a cell division. And if there really is a cell division, the distance between the two daughter cells should be about half of the average distance between two neighboring adult cells, i.e., 25 pixels.

Performance Evaluation: Figure 7 shows the tracking results across time instants with cell division. The same cells are denoted by the same number while for the cell division, the mother cell and daughter cells are denoted by circles. Table 1 shows the number of cell divisions being detected manually and automatically by the proposed method

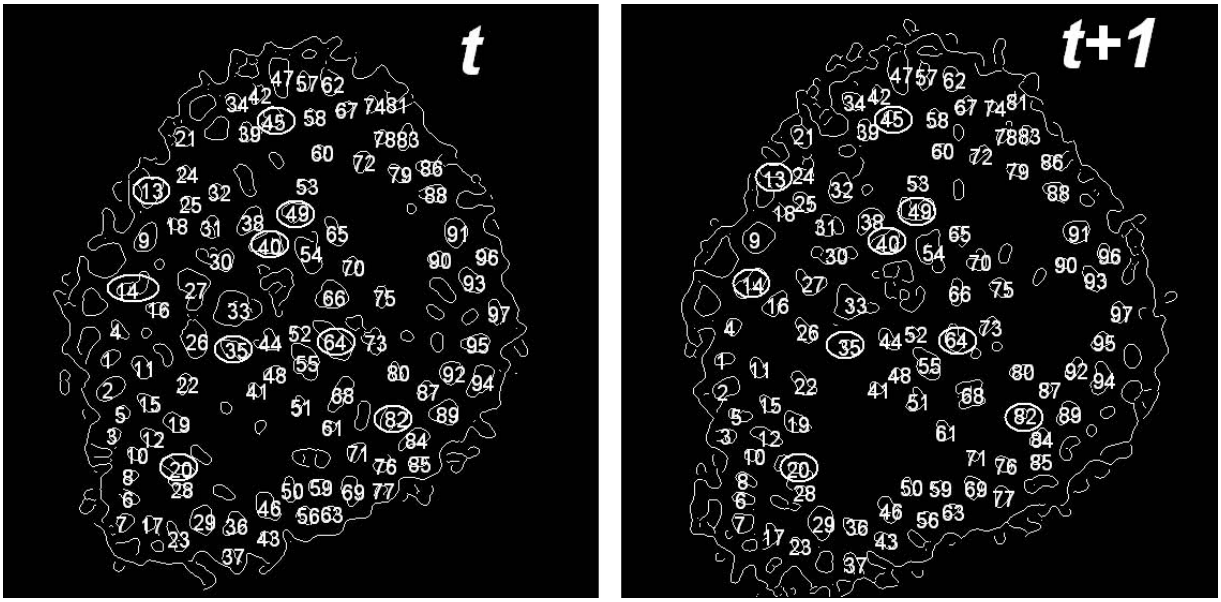

Figure 9. The tracking results after the spatial and temporal track fusion process, where 9 newly tracked cells are denoted by white circles.

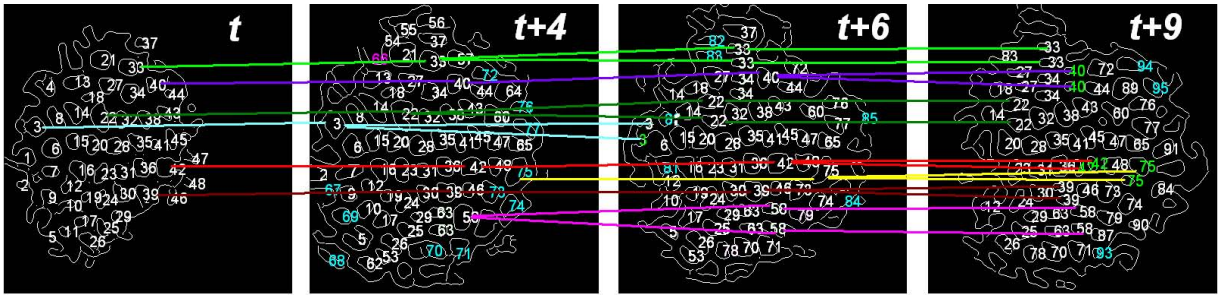

Figure 10. The cell lineages along 4 time instants are denoted by lines of different colors. Here the same number denotes the same cell at different time instants, and the divided daughter cells are denoted by the same number at the same time instant.

Slice	Time (hours)	Manual	Proposed Method
3	39	21 (20)	20
5	57	48 (49)	49
7	69	55 (53)	53

Table 1. The comparison of cell divisions tracked manually (both in the unsegmented data and the segmented data, the latter is in the brackets), and those detected automatically with the proposed local graph matching method in different slices.

in different slices. Here the manually detected results act as the ground-truth. We can see that this local-graph matching method can detect 100% of cell divisions in the segmented data, and about 96% of the unsegmented data. Sometimes, imperfect segmentation will cause pseudo-divisions, as in the second row in Table 1.

Figure 8 is the case of tracking across image slices, with noise in the central part of the cell images, where local graph matching method demonstrates its robustness to find the corresponding cell pairs. From the tracking results, we can notice that when the cells in the central part are almost totally corrupted, the correctly segmented cells are effectively tracked. The robustness of this local graph matching method comes from the ability to grow the correspondences to the neighborhood from a seed pair. For example, if we assume the 10th cell pair as the seed pair, we can find the 34th cell pair by different correspondence growing paths, such as (10, 18, 22, 29, 34), (10, 17, 20, 32, 31, 34), (10, 9, 14, 23, 32, 31, 34) and so on. As long as we the right seed pairs are chosen, the tracking algorithm will automatically pick up a right path from the seed pair to any other cell pairs. Even if some of these paths are corrupted by noise, cell pairs can be found as long as there is one feasible path.

Figure 9 depicts the tracking results before the fusion of the spatial and temporal tracks and after the fusion process, respectively. We can notice some newly tracked cells (de-

Slice	T (hours)	Manual	No fusion	With fusion
2	3-6	45 (37)	37	43
3	6-9	72 (64)	64	69
5	3-6	84 (74)	74	80
5	18-21	85 (72)	72	77
5	21-24	52 (41)	41	48
6	6-9	99 (77)	77	90

Table 2. Cells being tracked manually (in the unsegmented data and in the segmented data, the latter is in the brackets) and automatically (before the fusion process and after the fusion process). We have chosen some representative examples from the data set.

noted by white circles) identified by the fusion process. Table 2 gives a comparison on how many cells are tracked before the fusion process in two consecutive time instants, and how many cells are tracked after the fusion process. Also, the number of the cells can be tracked manually both in the unsegmented data and the segmented data are treated as ground truth. We can track all those cells that can be tracked manually in the segmented data in one single slice. By fusion of the cells correspondences across time and slices, we can identify more cell pairs, improving the tracking accuracy up to 90% of the manually tracking cells in the unsegmented data.

Finally, we show the tracking results on computing the cell lineages by repeating the two-frame tracking process over the entire data collection of 72 hours. Based on the correspondence results of cell tracking and cell division tracking, we can get the cell lineages, as illustrated in Figure 11. Figure 10 shows the example of cell lineages superimposed on the images.

4. Conclusions

In this paper, we presented a local graph matching based method to track cells and cell divisions from 4D (3D space + time) confocal microscopy images with significant sensor noise. The main challenge comes from the special structure of the SAMs where the cells adhere to each other. We address this problem by exploiting the geometric structure and topology of the cells' relative positions. By matching the cells' related local graphs we get the cells' correspondences both spatially and temporally, and locate the cell divisions. Finally, we fuse the temporal and spatial tracking results together to get the cell lineage. In the experiment, we can track up to 90% of all the cells and can identify 96% of cell divisions.

References

[1] T. Chan and L. Vese. Active contours without edges. *Image Processing, IEEE Transactions on*, 10(2):266–277, Feb 2001.

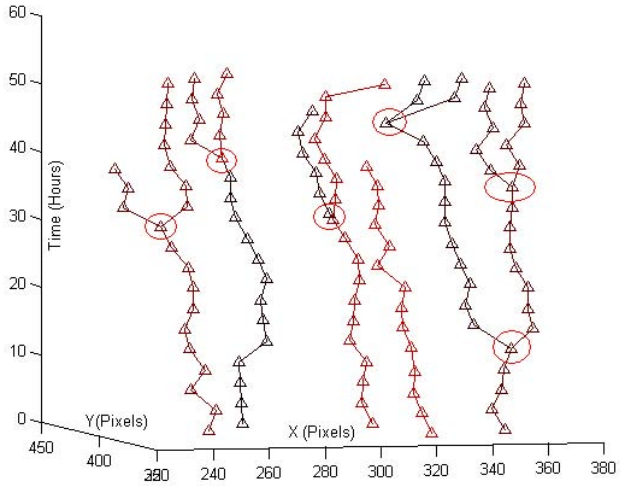

Figure 11. The tracks of 5 cells along 18 time instants (i.e., 54 hours), with 6 cell divisions and one cell dividing twice, denoted by red circles. Because of bad image quality in this slice, the cell lineages are not over 72 hours in this example.

[2] H. Chui. A new algorithm for non-rigid point matching. In *Proc. CVPR*, volume 2, pages 44–51, 2000.

[3] E. Fazl-Ersi, J. S. Zelek, and J. K. Tsotsos. Robust face recognition through local graph matching. *Journal Of Multimedia*, 2(5):31–37, 2007.

[4] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(4):377–388, 1996.

[5] V. Gor, M. Elowitz, T. Bacarian, and E. Mjolsness. Tracking cell signals in fluorescent images. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, page 142, Washington, DC, USA, 2005.

[6] K. Li and T. Kanade. Cell population tracking and lineage construction using multiple-model dynamics filters and spatiotemporal optimization. In *Proceedings of the 2nd International Workshop on Microscopic Image Analysis with Applications in Biology (MIAAB)*, September 2007.

[7] A. Rangarajan, H. Chui, and F. L. Bookstein. The softassign procrustes matching algorithm. In *Information Processing in Medical Imaging*, pages 29–42. Springer, 1997.

[8] P. B. Reuille, I. Bohn-Courseau, C. Godin, and J. Traas. A protocol to analyse cellular dynamics during plant development. *The Plant Journal*, 44:1045–1053, 2005.

[9] P. Viola and I. Wells, W.M. Alignment by maximization of mutual information. *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 16–23, Jun 1995.

[10] L. Wiskott, J.-M. Fellous, N. Krger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):775–779, 1997.

[11] Y. Zheng and D. Doermann. Robust point matching for nonrigid shapes by preserving local neighborhood structures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(4):643, 2006.