博士論文
(Ph.D. Thesis)

Cell Tracking
Under Dense Cell Culture Conditions
for Cell Behavior Analysis

(細胞挙動解析のための密な細胞画像における細胞トラッキング)

The University of Tokyo

Graduate School of Interdisciplinary Information Studies

Emerging Design and Informatics Course

49-127306　　備瀬竜馬 (Ryoma Bise)

Supervisor　　佐藤洋一 教授 (Professor Yoichi Sato)

Abstract

Analysis of cell behaviors in populations is important for research and discovery in biology and medicine. For example, in drug discovery, cell migration speed and cell death are analyzed to assess the effectiveness of anticancer drug candidates. In regenerative medicine, cell behaviors are considered important measures to assess the quality of cells in non-invasive images. To effectively obtain quantitative measurements of cell behaviors, automatic cell-tracking methods have been developed and successfully applied to low-to-middle density cell cultures. However, tracking cells in highly populated cultures remains difficult despite recent advances. The main difficulty arises from the following aspects of this problem in such a case. First, cells more often touch and partially overlap and form a cell cluster with blurry intercellular boundaries. Such conditions cause difficulty in segmenting cell regions from one image. Second, the number of cells may change due to cell division and cells entering or leaving the field of view. Third, neighboring cells often have similar appearances. This makes it difficult for appearance-based association methods to properly work. In addition, the cell movements between successive frames are often larger than the distances to nearby cells when a wide area is observed in a 3D volume. This makes it almost impossible to associate cells between frames based on their proximity.

In this thesis, I address these difficulties step-by-step. Before addressing the tracking of cells under dense conditions, in the first two chapters of this thesis, I propose cell-tracking methods for reliably tracking cells under low-to-middle density conditions in which several cells touch and make a cluster with blurry inter-cellular boundaries. I first propose a tracking method for identifying touching-cell clusters in the frame-by-frame association step, then re-segmenting the clusters to their member cells by partial contour matching to reliably track individual cells even when they partially overlap.

A disadvantage of frame-by-frame association methods is that they use only local temporal information to optimize an association in each successive frames. This often causes mistakes in which a cell trajectory is associated with a false positive detection error, such as tips of cells, when a false positive appears near the cell. If cells are observed for several frames, such mistakes can be easily corrected since false positives usually quickly disappear. Based on this observation, I proposed a global spatio-temporal data association method for tracking dividing objects. This method first generates reliable tracklets by frame-by-frame association, then the tracklets are linked using binary linear programming. This method exhibits higher accuracy compared with state-of-the-art methods.

Under high-density conditions, the performance of the above-mentioned methods drastically declines because cell detection cannot be done reliably in such conditions. For better cell detection in such high-density conditions, I propose a cell-detection method for selecting an optimal set of cell regions from redundant candidate regions. The method first detects redundant candidate regions by allowing candidates to overlap to avoid miss detections. Then, to avoid over detections, an optimal set of cell regions is selected from the redundant regions under non-overlapping constraints, where a selected region looks like a single cell and does not overlap with other cells. This cell-detection

method improves the performance of cell detection and tracking. This idea of selecting an optimal set from redundant candidates is expanded to tracking.

A disadvantage of typical detection-and-association tracking methods is that the tracking process heavily depends on detection results since the detection step is independent of the associate step, and the detection errors directly propagate to the association step. To solve this problem, I propose a cell-tracking method which uses the cell region information from the previous frame to help segment the blurry cells, rather than relying on only the image appearance at the current frame. This method concurrently solves both selection of an optimal set of cell regions from the redundant candidates and association of cell regions between successive frames for generating reliable tracklets. The generated tracklets are linked to obtain entire trajectories by using global spatio-temporal data association method proposed in Chapter 3. This method achieves better tracking performance compared with above methods proposed in Chapters 2 and 3. Moreover, I extend this method to solve the joint problem of optimal tracklet selection from redundant candidate tracklets and global association in order to track cells more robustly under dense conditions.

To address the large displacement problem, in which cell movements between successive frames are often larger than the distances to the nearby cells when a wide are is observed in 3D volume, I also propose a tracking method for exploiting an observation in which nearby cells under high-density conditions exhibit similar motion patterns. This is done by introducing global motion estimation and local pairwise spatial relationships of cell positions. Finally, I discuss the effectiveness of my methods in biological research by analyzing cell behavior in scratch wound healing assays.

Acknowledgements

First and foremost, I would like to thank my advisor, Yoichi Sato, for his guidance and advice over the years. His clear insight and bright personality continually encouraged me to complete this thesis. I would also like to thank my advisor at Carnegie Mellon University (CMU), Takeo Kanade for his valuable comments and suggestions when I was a visiting industrial scholar of Robotics Institute at CMU from 2008 to 2010. Besides my advisors, I would like to thank the members of my thesis committee, Kiyoharu Aizawa, Takeshi Naemura, Shunsuke Kamijo and Takeshi Oishi, for their helpful feedback on my research.

I am grateful to have the opportunity to work with my colleagues in the Cell Image Analysis Research Group and my collaborators. I would especially like to thank Phil G. Campbell, Lee E. Weiss, Dai Fei E. Ker, Mei Chen, Silvina N. Junkers, Zhaozheng Yin, Seung-il Hu, Sungeun Eom, An-an Liu, Kang Li, Katsuya Kondo and Jiyan Pan. I will never forget the rewarding time I spent with them.

I greatly appreciate the many insightful discussions I had with Akiko Kondow, Tetsuya J. Kobayashi, Kiyoshi Ohnuma and Shigenori Nonaka. A special thanks goes to Ryo Yonetani, Tetsu Matsukawa, Yusuke Sugano, Haruoki Maruyama, Sakie Suzuki, Yoko Imagawa and Chio Usui and all of my laboratory colleagues for helping me with my study.

I greatly appreciate Masahiro Kino-oka, Mee-hae Kim and all members of the Strategic Promotion of Innovative Research and Development (S-Innovation) Program of the Japan Science and Technology Agency. This work (Chapter 5) was partly supported by S-Innovation Program.

I am also grateful to Osamu Nakagawa, Hideo Fujimura, Yuji Osumi, Yoshitaka Maeda, Yuji Tsuzuki, Sho Sanami, Naoki Yokoyama, Hideshi Hattori, and all my superiors and colleagues at Dai Nippon Printing. I am also grateful to all my friends in Tokyo, Okinawa, Pittsburgh, and all over the world. This thesis would not have been possible without the great times I spent with them.

Lastly, my deepest gratitude goes to my wife, Mika and my son, Satoshi and all my family for their love and support. I am grateful for the time we have been together.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and motivation

Analysis of cell behaviors in populations is important for research and discovery in biology and medicine. For example, in biological research, Nikolic *et al.* [Nikolic06] manually tracked cell migration in wound healing assay to understand how multiple cells execute highly dynamic and coordinated movements during the healing process. In addition to individual cell trajectories, the analysis of cell lineage is important, in particular, to study cell differentiations. Ravin *et al.* [Ravin08] developed an in vitro system that allows analysis of the fate transitions from central nervous system (CNS) stem cells to differentiated neurons and glia cells. In their system, the cells are manually annotated for identifying mother-daughter relationships and when and which stem cells are differentiated to other types of cells. To effectively obtain quantitative measurements of cell behaviors, many automatic cell-tracking methods have been developed. However, in biology and medicine, cells are often cultured under a variety of cell culture conditions, such as low-to-high density, as shown in Figure 1.1. For example, in regenerative medicine, the cells are cultured until they densely fill in the dish in order to mature them. Cell behavior metrics, such as migration speed and cell shape information, in high density are important to assess the quality of cells in non-invasive images before transplantation. Cell tracking under such high-density conditions still remains a non-trivial task. The main difficulty arises from the following aspects of this problem.

- Touching and partially overlapping: When multiple cells touch or partially overlap, they form a cell cluster with blurry intercellular boundaries. Such touching and/or partially overlapping cells present a performance bottleneck with most current cell-tracking methods; they may either lose track of one or more of the cells or confuse their identities.
- Cell division: The number of cells may change due to cell division and cells entering

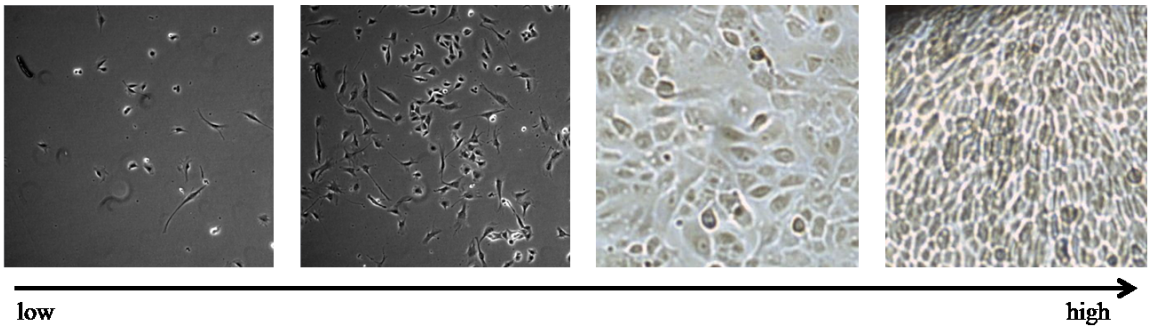


Figure 1.1: Example images in cell populations. Cell density increases from left to right.

or leaving the field of view.

- Similar appearance: Neighboring cells often have similar appearances. This makes it difficult for appearance-based association methods to properly work.
- Large displacement: Cell movements between successive frames are often larger than the distances to the nearby cells when it takes time to obtain 3D volume data for a wide range of specimens. This makes it almost impossible to associate cells between frames based on their proximity.

In this thesis, I propose several cell-tracking methods for addressing these challenges. I quantitatively evaluated these methods by applying them to actual biological research and discuss their effectiveness.

1.2 Related work

1.2.1 Cell detection

Automated cell detection in microscopy images is one of the most important tasks in cell behavior analysis in biological research. Robust cell detection and segmentation play a crucial role in developing cell-tracking methods. Even though many cell-detection methods have been proposed, cell detection under high-density conditions still remains a non-trivial task. There are several difficulties in cell detection. First, microscopy images often have inhomogeneous backgrounds and noise, and their contrast is usually very low. In such images, the pixel intensity of a cell is often lower than that of the background. Second, cells often touch other cells and form a cluster with blurry inter-cellular boundaries.

Many cell-detection methods have been developed for detecting individual cells in populations. Simple intensity thresholding methods, such as the Otsu method [Otsu79], set an intensity threshold on the basis of the statistical intensity histogram of all image pixels and classify these pixels into cells and background. However, such single-threshold

methods do not work well since microscopy images often have inhomogeneous backgrounds. Local adaptive thresholding methods have thus been developed for detecting cells under inhomogeneous background conditions [Peng09b][Praveen06]. They set the thresholds on the basis of a local spatial intensity histogram. They generally work well when the cell density is sufficiently sparse and the image contrast is sufficiently high. However, if the density is high and/or the contrast is low, they often fail to segment closely located cells since the intensities at the boundary between these cells tend to exceed the local threshold. In addition, morphological operations, such as dilate and erode that are used to remove noise [House09][Li08], are not sufficient to detect cells under high-density conditions. Two commonly used open-source detection methods are CellProfiler [Carpenter06] and FIJI [Schindelin12]. The IdentifyPrimaryObjects module in CellProfiler first segments the image foreground using binarization with a threshold. Then, the touching cells are separated using the intensity peaks and shapes. In FIJI detection, the contrast is first enhanced using the difference of Gaussian filter. The local maxima or minima that satisfy pre-defined conditions, e.g., intensity range and window size, are then found since local maxima often include false positives. This method usually depends on pre-defined conditions: if the conditions between the training data and test data differ, performance is degraded. Filter-based methods have also been proposed. Laplacian-of-Gaussian (LoG) filters are applied to detect cell blobs [Peng09a]. Eom *et. al.* proposed a cell-detection method for selecting an optimal solution from candidate points after applying a filter-bank [Eom10]. Watershed segmentation methods [Kachouie08][Yang05] have also been developed for segmenting multiple cells in one image. The watershed method of Vincent and Soille [Vincent91] isolates the catchment basin by flooding the gray value relief of the gradient of an image. A number of "water sources" are placed at various pixel locations, usually local maxima/minima, and the water is allowed to flow out and flood freely. The water flows eventually meet along the watershed lines. However, this method tends to result in over-segmentation since many false positives of water sources are usually detected. Active contour methods such as the level-set method [Tse09][Yu08] have also been widely used for cell segmentation. The level-set method represents closed curves by using the zero level-set of a function in which multiple closed curves are allowed to represent multiple cell regions. Optimal regions are found while the closed curves are being changed by maximizing the energy function, which is defined on the basis of several elements such as region statistics and image edges. Different energy functions for level-set and snakes have been proposed, such as edge-based [Zimmer02], region-based [Chan01], and texture-feature-based [Wang07]. However, this optimization is done locally; thus, it depends on the initial curves, which tends to result in miss detection when multiple cells touch and partially overlap.

A graphical-model-based method [Pan10] has been proposed for detecting cells in phase-contrast images. It first detects interest points, which are then used to form a graph. It

next solves the problem of whether the neighboring points are in the same cell by using a graphical model. It uses the characteristics of phase-contrast images, in which the intensities of a cell's boundaries are higher than those of its inner area, to compute the probability that neighboring interest points are in the same cell. However, this method works for only a certain microscope type and cannot be used for images from other types of microscopes. For fluorescent microscopy, Lou *et. al.* proposed a graph cut framework that incorporates a " blob "-like shape prior [Lou12]. Lesko incorporated a gradient-based score into the pair-wise term in the energy function [Lesko10] for phase-contrast microscopy images.

Suitable preconditioning methods have been developed [Li09][Yin10a] for segmenting cells in non-invasive microscopy images such as phase-contrast and differential interference contrast (DIC) images. They are based on a specific microscopy image formation model that is used to restore authentic images without artifacts such as halos and shading. With these methods, cells appear as regions of positive values against a uniform background with almost zero pixel intensities. Non-invasive microscopy images can be treated like fluorescent images by using these methods. However, although they segment the foreground and background very well, post processing is needed to detect individual cells. Yin *et. al.* [Yin12] presented a cell-detection method for use after preconditioning. Their method segments a preconditioned image into foreground and background regions by using a threshold value. It then connects bright neighbor pixel components into a blob as a cell candidate. Finally, it classifies the blobs into cells or non-cell regions by using a trained support vector machine (SVM) classifier that uses cell appearance features including the shape and corresponding intensity values in the original and preconditioned images. This method removes false positive regions such as chips of cells. However, when cells touch under high-density conditions, they often cause false negatives in which multiple cells are detected as a single cell.

1.2.2 Cell tracking

As recently reviewed in [Hand09][Meijering09][Rittscher10][Maska04], many cell-tracking methods have been proposed, which can be classified into three groups: those based on filtering-based sampling, those with model-based contour evolution, and those based on detection-and-association.

The first group uses sequential Bayesian filters, such as Kalman filters and particle filters, that are widely used in object tracking [Smal06][Okuma04]. Particle filter tracking maintains a probability distribution as a set of weighted particles. Each particle is an estimate representing one possible location of the object being tracked. This group requires other initialization methods to define the prior distributions of object states in the first frame. One of the disadvantages of this group is that cells may be difficult to track when they are closely located and exhibit a similar appearance.

The second group, in particular those using the level-set, is widely used in cell segmentation and tracking [Yang05][Xiaoxu07][Li08][Dzyubachyk10] [Zhang04][Dufour05][Maska04]. Segmented contours in the previous frame are given as initial contours to identify the cell contours in the subsequent frame by minimizing the energy function. The advantage of level-set tracking is that it can easily follow shapes that change the topology. Traditional level-set methods have several shortcomings, such as their sensitivity to the energy weights and how they handle (initially) touching cells and cells entering the observation frame. Li *et. al.* proposed a tracking method that involves using both level-set and detection-and-association methods to handle cell entering and exiting [Li08]. To handle contact between cells in a sequence, Zhang *et. al.* proposed a method for assigning a single level-set function per cell by minimizing the overlap region between these level-sets [Zhang04]. Dzyubachyk *et. al.* proposed a tracking method for overcoming all the shortcomings of traditional level-set methods [Dzyubachyk10]. The method uses the watershed transform to refine the results of a level-set method for closely located cells. The method also uses Radon transform, which "decouples" the active surfaces by means of separating planes, making it possible to apply the stopping criterion to each level-set function separately. For 3D volume data, Dufour *et. al.* [Dufour11] proposed a 3D cell-tracking method for expressing a surface as a discrete triangular mesh and minimizing the energy functional accordingly to reduce computational cost. A disadvantage of this group is that tracking cells would fail under densely populated conditions due to the problem of local minima in energy minimization, such as part of an active contour being incorrectly assigned to the boundary of a neighboring cell.

The third group, based on detection-and-association, has also been frequently used. Al-Kofahi *et. al.*[Kofahi06] proposed a method for first segmenting the cell regions using the adaptive thresholds then solving the association between the successive frames using integer linear programming. However, this method cannot handle cases in which multiple cells frequently overlap and split. Padfield *et. al.* [Padfield11] used the min-cost flow to solve the association problem. This method can handle various cell behaviors that include migration (cell movement), mitosis (cell division), overlapping, entering, and leaving. For 3D data, this third group can be straightforwardly extended using 3D cell segmentation methods [Lin05][Long07][Indhumathi11] for the detection step. However, such frame-by-frame association methods face a problem when a false positive appears near a cell because the false positive may be associated with a track.

Recently, global spatio-temporal data association methods have been proposed for tracking general objects that do not divide [Bonneau05][Zhang08]. Zhang *et. al.* [Zhang08] proposed a minimum-cost flow network to solve the global data association of multiple objects over time. Global data association methods for tracking dividing objects have been also proposed [Schiegg13][Kausler12]. Schiegg *et. al.*[Schiegg13] proposed a tracking method for identifying a cluster consisting of touching cells in the global association step then

re-segmenting the touching cells. These global association methods are known to exhibit higher accuracy of tracking general objects than frame-by-frame association methods since the method uses global spatio-temporal information for association. The disadvantage of this third group is that the detection results are independently produced with the association step, where the detection results are directly used. Thus, these methods are susceptible to errors in the detection step.

Identifying cell division events is an important task in cell tracking. I briefly summarize related work on detecting cell division, though it is not my main target in this thesis. Several mitosis detection methods, which use temporal shape dynamics in cell division without tracking, have been proposed. Li *et. al.* [Li08b] applied a fast cascade Adaboost method to volumetric Haar-like features to detect mitosis events. Liu *et. al.* [Liu10] applied the Hidden Conditional Random Field (HCRF) model to detect patch sequences containing mitosis. This method cannot localize division timing in the sequence, though it determines if the patch sequences contain cell division. Huh *et. al.* [Huh11] proposed a graphical model called 'EDCRF', which identifies the temporal localization of birth events as well. They also demonstrated that the functionality of this recent mitosis detection algorithm significantly improves state-of-the-art cell tracking systems [Huh11b]. In this research, I used EDCRF to detect cell division events.

1.3 Overview and contributions

The work presented in this thesis is based on the third group of cell-tracking methods , *i.e.*, detection and association, since these methods can easily handle various cell behaviors, including migration, mitosis, overlapping, entering, and exiting. Figure 1.2 shows the framework of all the proposed cell-tracking methods including the following five functions: detecting cell positions at each frame (cell detection); identifying cell division events in sequences (mitosis detection); associating detection results between successive frames to identify cell behaviors including migration, division, enter, exit, and overlapping (frame-by-frame association); segmenting cell cluster regions to their member cells (re-segmentation); and linking tracklets globally to obtain entire cell trajectories (global association). In particular, my methods contribute four functions, cell detection, frame-by-frame association, re-segmentation, and global association. For mitosis detection, I use a current method 'EDCRF' [Huh11]. I developed tracking methods step-by-step to address all the difficulties in tracking cells under dense conditions. Each proposed method and specific problems are presented in each chapter. I first present proposed cell-tracking methods for addressing difficulties, in which several cells touch and make a cluster under low-to-middle density conditions, in Chapters 2 and 3 before focusing on the difficulties under high-density conditions. Then, cell detection and tracking methods for addressing difficulties under high-density are

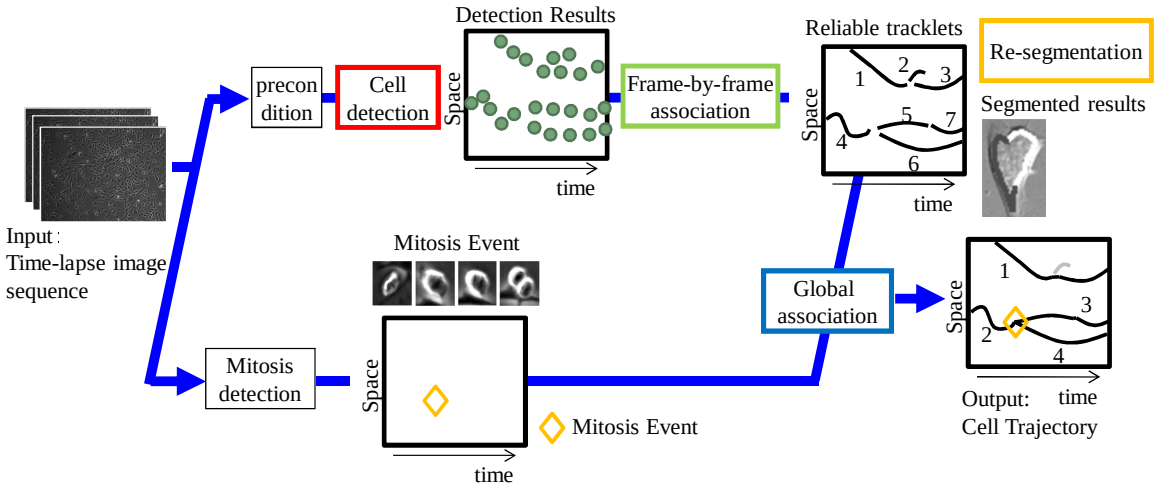


Figure 1.2: Overview of framework of proposed tracking methods. The proposed methods contribute to four functions: cell detection, frame-by-frame association, re-segmentation and global association.

presented in the remaining chapters.

Figure 1.3 shows the relationships among the proposed methods and tracking functions. The circles indicate the main contributions of the proposed methods. If a chapter number is in a cell of the table, that particular method uses the method proposed in that referenced chapter for that particular function. For example, the cell-tracking method by global association proposed in Chapter 3 uses the frame-by-frame association method proposed in Chapter 2 for generating tracklets. The description of "(Chapter X)" indicates that the method proposed in Chapter X could be applied for a function, though the method was not used in the experiments. The contributions of each chapter are summarized as follows:

- Chapter 2: When multiple cells touch or overlap, they appear to form a cell cluster with blurry intercellular boundaries. In this case, it is often difficult to identify individual cells in the cluster from one image even though a human manually annotated. I propose a tracking method that identifies touching cell clusters from detection results by frame-by-frame data association then re-segments the clusters to their member cells by partial contour matching between cells and the cluster. This method makes it possible to robustly track two or three partially overlapping cells while maintaining the identity information of individual cells throughout the process from their initial contact to eventual separation.

- Chapter 3: When a false positive segmentation, such as tips of cells, appears near a mitotic cell, local temporal association methods may cause a mother-daughter relation-

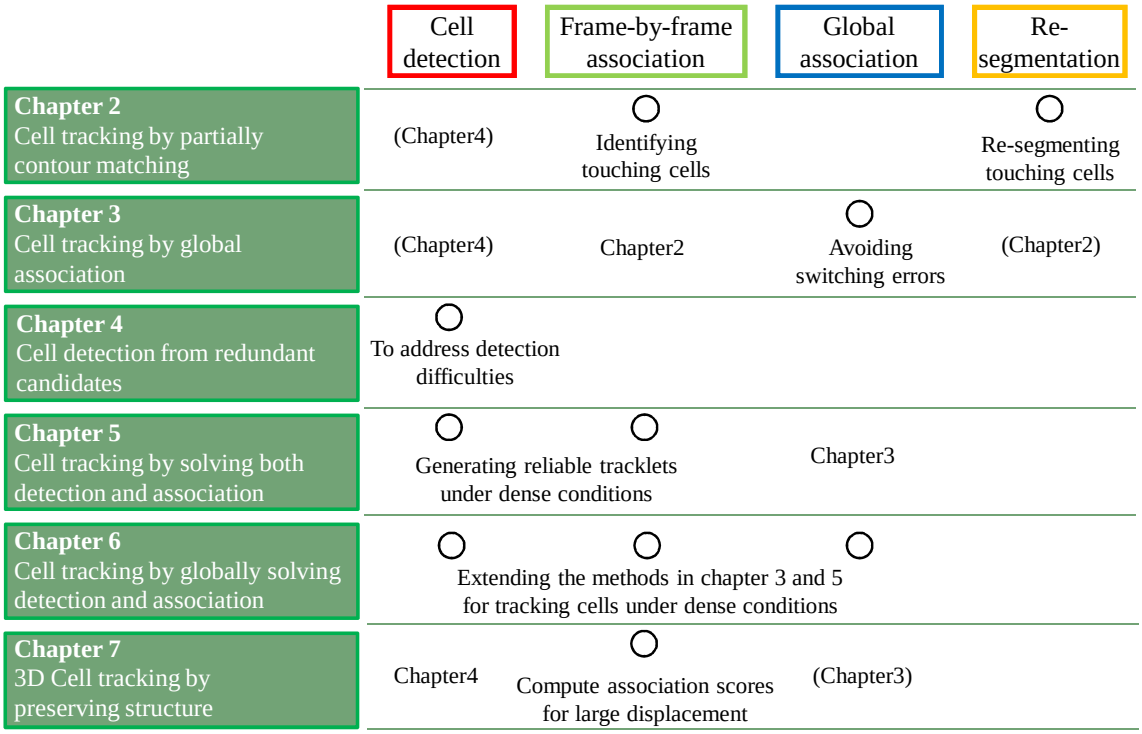


Figure 1.3: Relationships of proposed methods.

ship error. Global temporal information is important in solving this problem. If the cells are observed for several frames after the birth event, it can be easily determined that one of the children cells is a false positive since false positives usually quickly disappear. This allows to correct the relationship. Current global spatio-temporal data association methods for tracking non-dividing objects cannot be applied to cell tracking directly since they do not take into account cell division i.e., a mother cell divides into two daughter cells to form a tree structure in the trajectory. I propose a global spatio-temporal data association method for the tree structure to obtain cell trajectories and lineage trees. First, reliable tracklets (i.e., short trajectories) are generated by linking detection responses based on the frame-by-frame detection-and-association approach method in Chapter 2. Then, the global tracklet association for the tree structures is solved using linear programming. To the best of my knowledge, this is the first attempt at formulating tree structure global association to track dividing objects. By introducing global spatio-temporal information, we can easily determines the false positives of cell detection as cell or noise since false positives usually quickly disappear. This method was quantitatively evaluated on sequences with thousands of cells captured over several days.

- Chapter 4: Cell detection in each frame is obviously important for improving tracking performance since detection errors usually cause association errors. When cell density is high, the performances of the above two methods proposed in Chapters 2 and 3 drastically decline because of the increase in cell-detection errors. Thus, I propose a cell-detection method for addressing all the difficulties in detecting dense cells simultaneously: multiple cells are mistakenly merged, a single cell is divided into multiple regions, and low-intensity cells are miss detected. The method first detects redundant candidate regions by allowing candidates to overlap to avoid miss detections. Then, to avoid over detections, I select an optimal set of cell regions from the redundant regions under non-overlapping constraints, in which a selected region looks like a single cell and does not overlap with other cells. This method improves the performance of cell detection and tracking. This idea of selecting an optimal set from redundant candidates is expanded to tracking in the following chapters.

- Chapter 5: Under dense cell-culture conditions, cells more often touch other cells with blurry intercellular boundaries. Such conditions cause difficulty in generating reliable tracklets with the global spatio-temporal data association method proposed in Chapter 3 since the frame-by-frame detection-and-association tracking process heavily depends on the detection results. To mitigate this problem, I propose a tracking method for determining the detection results in the association step by using both image features in the current frame and the tracking results in the previous frame. This make it possible to make more reliable tracklets under high density conditions compared with typical detection-and-association methods. After generating reliable tracklets, the global data association method proposed in Chapter 3 is used to obtain all cell trajectories and lineage trees. This method was evaluated based on the challenging image sequences in which cells were cultured in high density and the boundaries of cells were blurring. The experimental results show that this method significantly improves the tracking performance comparing with the two other proposed methods introduced above.

- Chapter 6: The method proposed in Chapter 5 depends on initialization of cell detection since the joint problem of optimal region selection and association is solved at each frame independently. I propose a cell-tracking method that first generates redundant candidate tracklets, then solves the joint problem of optimal tracklets selection and association globally. This method generates redundant candidate tracklets, which include many false positives but in turn very few false negatives, by allowing tracklets to overlap. This is a similar idea with the detection step proposed in Chapter 5. Next, the problem of both selecting an optimal set of cell tracklets from the redundant tracklets and associating the tracklets over frames under non-overlapping constraints

is solved simultaneously. This methods achieved the best performance on the comparison with the above tracking methods proposed in Chapters 2, 3 and 5.

- Chapter 7: I propose a cell-tracking method that enables to track cells with large movements. The increment in time-lapse imaging cannot be shortened to monitor a wide area, which results in the problem in which the movements of cells between successive frames are often larger than the distances to the nearby cells. This makes it almost impossible to associate cells between frames based on their proximity. To mitigate this problem, this method exploits the observation in which nearby cells under high-density conditions exhibit similar motion patterns. This is done by introducing global motion estimation and local pairwise spatial relationships. This method was evaluated on synthetic point-sets and compared against the existing methods. The proposed method was evaluated on synthetic point-set and compared against current methods.

- Chapter 8: I show how easily and effectively automated cell-tracking systems can provide detailed spatio-temporal cell behavior measurements for biological analysis. The spatio-temporal measurements of cell behaviors are important for critical analysis, because the cell culture conditions vary with time and space on the dish. For example, the effectiveness of a medicine may change with time and space since cell density can differ in the different space. I present an application of automatic cell-tracking for wound healing assay *in vitro* under three different culture conditions to demonstrate how easily and effectively automated cell-tracking systems can provide detailed spatio-temporal cell behavior measurements for biological research.

Finally, I present a conclusion and future directions in Chapter 9.

1.4 Notations

I propose cell-detection and tracking methods individually in each chapter. Thus, I basically define the notations of functions, variables, sets and parameters independently in each chapter. For key notations, I use common notations for all chapters as following. A set of candidate cell regions at each frame is denoted as $\mathbf{A}^t = \{A_i^t\}$, where A_i represents the i-th candidate cell region at frame t. The set of detection results is denoted as $\mathbf{R} = \{R_i\}$, where R_i represents the ith cell blob. The term $\mathbf{X} = \{X_i\}$. $X_i = \{R_{i_j}\}$ is a tracklet consisting of an order list of detection results, where R_{i_j} indicates the jth detection result on tracklet X_i.

Chapter 2

Tracking Partially Overlapping Cells

2.1 Introduction

A reliable cell-tracking system should be capable of tracking not only well separated cells but also cells that are touching or partially overlapping. When multiple cells touch or overlap, they appear to form a cell cluster with blurry intercellular boundaries. Such touching and/or partially overlapping cells present a performance bottleneck with most current cell-tracking methods; they may either lose track of one or more of the cells, or confuse their identities. These errors, frequently compounded with the misclassification of the split and merge of cell clusters, as cellular division or fusion events, result in fatally erroneous cell lineages.

I propose a tracking method for identifying touching cell clusters from segmentation results by solving a data association problem, then re-segmenting the clusters to their member cells by using contour shapes of cells and cell clusters to reliably track individual cells even when they partially overlap.

This method first segments images into blobs for each frame, where each blob corresponds to either an individual cell or a cluster of overlapping cells. Next, the association step involves making hypotheses of all possible cell actions including migration, mitosis, touch or overlap, enter and exit, and computes corresponding an association score. Then an optimization problem that maximizes the sum of the scores of the hypotheses is solved by a binary linear programming problem.

When touching cell clusters are identified by data association, they are re-segmented to their member cells. The boundary contour of the resultant cluster is made of the partial contours of the cells that constitute the cluster when cells touch and overlap. As the contours of member cells deform and as the manner and degree of cell overlap change, the boundary contour of the cluster also continually deforms. This deformation provides the primary cue

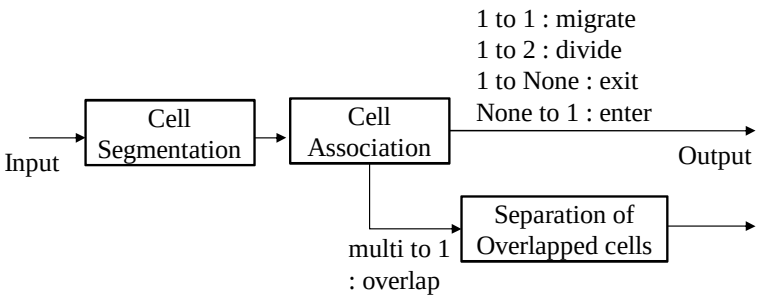


Figure 2.1: Method overview.

for distinguishing individual cells in a cluster.

My method proposed in this chapter utilizes this information by optimal matching between partial cell contours and the cluster boundary contour. For each cell, its contour shape is identified when it first appears, and is updated while being tracked. When two or more cells touch and overlap, and form a cluster, I compute the optimal combination of their partial contours such that they together comprise the cluster boundary. These identified partial contours are maintained as the updated contours of the respective member cells. This process is repeated until they separate and no longer form a cluster. Therefore, this proposed method can maintain each cell's identity information throughout the process from their initial contact to eventual separation.

The proposed method is applied to a task of tracking migrating and proliferating human CNS stem cells in image sequences of DIC microscopy and achieved 96% tracking accuracy despite the frequent formation of multiple-cell clusters. I also evaluated the accuracy of cell lineages, in which 94% of target effectiveness and 93% of mitosis branching correctness were achieved.

2.2 Cell tracking method for tracking partially overlapping cells

Figure 2.1 shows the overview of the proposed cell-tracking method in which I use for automated construction of cell lineage from a sequence of DIC images. Each image frame of a sequence is processed in three steps: 1) Step of cell segmentation: converting an input image to a binary image consisting of blobs, each of this corresponds to either an isolated cell or a cluster of overlapping cells; 2) Step of cell association: solving the data association problem between the cells tracked until the previous frame and the blobs detected in the current frame, and identifying cell clusters; and 3) Step of segmentation of overlapping cells: decomposing each cluster to its member cells by using partial contour matching [Bise09].

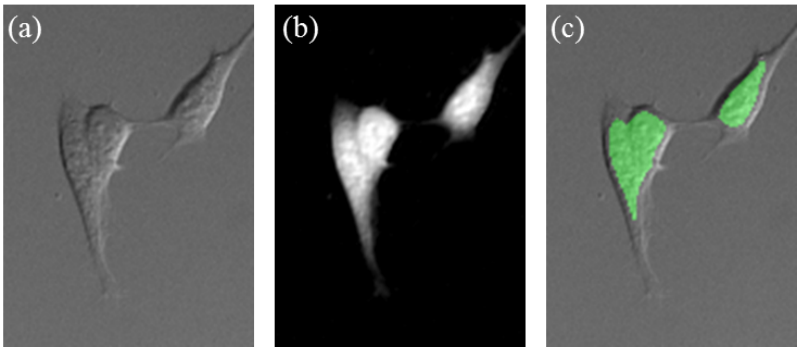


Figure 2.2: Example of cell segmentation results. (a) Original image. (b) Preconditioned image. (c) Detected cell blob regions overlaid on the original image.

2.2.1 Cell segmentation of difference interference images

A differential interference contrast (DIC) microscope is widely used for long term imaging of unstained, transparent specimens, such as living cells and micro-organisms. Due to the dual-beam interference optics of a DIC microscope, DIC images include non-uniform shadow-cast artifacts as shown in Figure 2.2 (a), making its direct segmentation difficult. To facilitate segmentation, I have adopted the image preconditioning method recently developed in [Li09]. The method utilizes the opto-physical principle of image formation by DIC microscopy, and transforms an input DIC image into an artifact-free image by minimizing a nonnegative-constrained convex objective function. In the resultant transformed image shown in Figure 2.2 (b), cells appear as regions of positive values against a uniformly-zero background. A simple thresholding method, such as Otsu thresholding, can easily segment out the cell regions (Figure 2.2 (c)). The set of such detection results is denoted as $\mathbf{R}^t = \{R_i^t\}$ where R_i^t represents the i-th cell blob at frame t.

One may notice that the segmented blobs exclude some portions of the cells, such as long, thin parts, called processes that extend from them. This exclusion is intentional because these portions, while important for later analysis, tend to confuse the tracking process since they deform significantly over time. They can be included later for further processing once cell identities are established.

2.2.2 Cell association

The tracking system assigns a unique ID to each tracked cell, its parent identifier Parent-ID, and its state information (i.e., centroid, contour shape of the cell region, and a cell event type) at each frame. The parent identifiers are important for maintaining its cell lineage information (Parent-ID=0 for cells that appear in the first frame, or enter the image view).

In each frame, the method determines the behavior of each cell based on both the spatio-

temporal history of cells up to the previous frame and the detected blobs in the current frame. Following seven possible events can be taken for a cell between time $t-1$ and t: (1) Migration: a cell deforms and moves; (2) Exit: a cell leaves the field of view; (3) Enter: a cell appears from the out of the field of view; (4) Mitosis: a cell divides into two new cells; (5) Overlapping: two or more cells touch or partially overlap with each other and make a single cluster together; (6) Joint migration in a cluster: two or more cells that have been in a cluster in the previous frame are again associated with one blob; and (7) Separation from a cluster: two or more cells that have been in a single cluster are now associated with two or more blobs. Here, the proposed method re-segments a touching cell cluster to its member cells for each frame when a cluster is identified (i.e., IDs and regions of the member cells are identified in the previous frame). Thus, I treat event (6) the same as event (5), and event (7) the same as event (1) in the cell association step.

In order to find the right association between blobs that are segmented out in the current frame t, and the cells that have been tracked till the previous frame $t-1$, the method considers all of these possible events for each cell as hypotheses, and determine optimum set of hypotheses, i.e., the optimum association, in terms of visual similarities and locational vicinity between associated cells, trajectory continuity, and consistency constraints (such as no cells to be left unexplained, or the number of cells to be conserved with respect to enter, exit, and division).

Let N_c be the number of cells at frame $t-1$, N_b the number of blobs at frame t, and H the total number of hypotheses that I have to consider for all the cell. I construct a H-dimensional vector $\boldsymbol{\rho}$ (called score vector), whose h-th component $\boldsymbol{\rho}(h)$ of the score vector represents an association score of hypothesis h if it were considered, where an association score indicates how likely the association hypothesis is in an optimal solution. I also construct an $H \times (N_c + N_b)$ binary-valued matrix C (called Constraint Matrix), whose h-th row represents which cells are associated to which blobs under that hypothesis. A set of hypotheses that I have to consider is generated, and the corresponding entries of $\boldsymbol{\rho}$ and C are defined as follows.

1. Migration hypothesis:
   If the space distance between cell l at $t-1$ and blob m at t is less than threshold d_e, the cell migration hypothesis $cell_l \rightarrow blob_m$ needs to be considered, and thus is added to the hypotheses set. Let h be the index of a new hypothesis. I append h-th row to C and a corresponding score to $\boldsymbol{\rho}$:

$$\begin{aligned} C(h,i) &= \begin{cases} 1, & \text{for } i = l \text{ and } i = N_c + m \\ 0, & \text{otherwise.} \end{cases} \\ \boldsymbol{\rho}(h) &= P_{mig}(blob_m | cell_l) \end{aligned}$$

   where $P_{mig}(blob_m|cell_l)$ is computed based on the cell location similarity and shape

similarity between the cell at frame t and the detected blob at frame $t-1$. The detail of the current particular implementations of computing this score are provided in the next section.

2. Exit hypothesis:
   If the l-th cell at $t-1$ is near the boundary of the field of view, the exit hypothesis $cell_l \rightarrow$ none needs to be considered, and thus is added to the hypotheses set. New entries for C and $\boldsymbol{\rho}$ are defined as:

$$\begin{aligned} C(h,i) &= \begin{cases} 1, & \text{if } i = l \\ 0, & \text{otherwise.} \end{cases} \\ \boldsymbol{\rho}(h) &= P_{exit}(\text{exit}|cell_l) \end{aligned}$$

   where $P_{exit}(\text{exit}|cell_l)$ is computed based on a distance from the image boundary. (Refer to the next section)

3. Enter hypothesis:
   If the m-th blob at t appears near the boundary of the field of view, the enter hypothesis none $\rightarrow blob_m$ is added.

$$\begin{aligned} C(h,i) &= \begin{cases} 1, & \text{if } i = N_c + m \\ 0, & \text{otherwise.} \end{cases} \\ \boldsymbol{\rho}(h) &= P_{entr}(\text{enter}|blob_m) \end{aligned}$$

4. Division hypothesis:
   If both of m_1-th and m_2-th blobs appear near the l-th cell, the division hypothesis $cell_l \rightarrow (blob_{m_1}, blob_{m_2})$ is added.

$$\begin{aligned} C(h,i) &= \begin{cases} 1, \text{ for } i = l, i = N_c + m_1, \text{ and } i = N_c + m_2 \\ 0, \text{ otherwise.} \end{cases} \\ \boldsymbol{\rho}(h) &= P_{div}(blob_{m_1}, blob_{m_2}|cell_l) \end{aligned}$$

5. Overlaping hypothesis:
   If both of l_1-th and l_2-th cells appear near the m-th blob, the overlapping hypothesis $(cell_{l_1}, cell_{l_2}) \rightarrow blob_m$ is added.

$$\begin{aligned} C(h,i) &= \begin{cases} 1, \text{ for } i = l_1, i = l_2, \text{ and } i = N_c + m \\ 0, \text{ otherwise.} \end{cases} \\ \boldsymbol{\rho}(h) &= P_{over}(blob_m|cell_{l_1}, cell_{l_2}) \end{aligned}$$

Figure 2.3 illustrates a simple example, where the number of cells at $t-1$ is 5, the number of blobs at t is 5, and the number of hypotheses is 13.

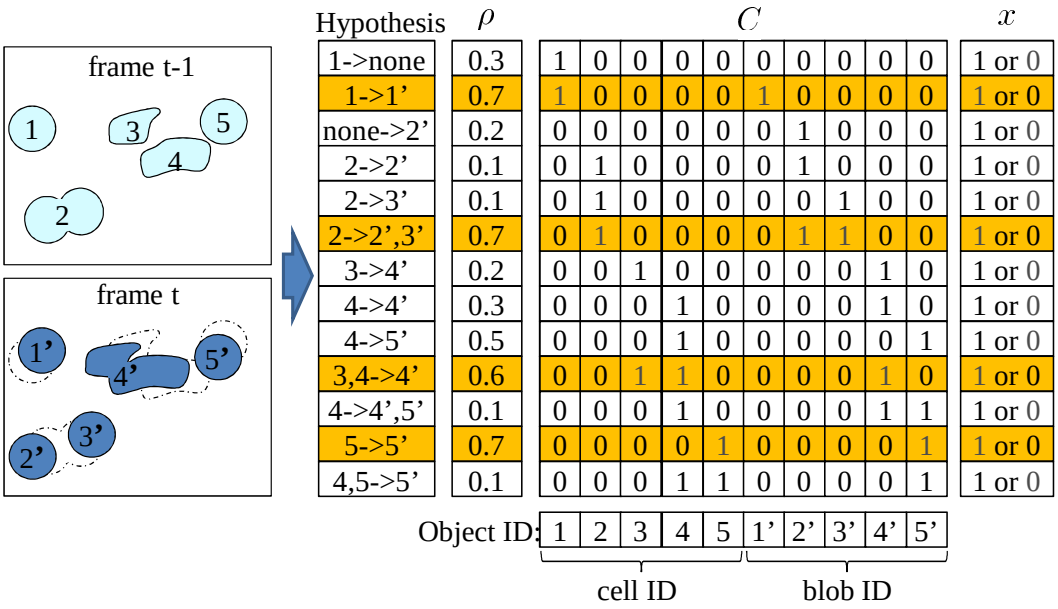


Figure 2.3: Example of binary linear programming where the optimal solution is highlighted by orange.

Once I make H hypotheses over N_c cells at $t-1$ and N_b blobs at t, the association problem selects a subset of rows of C such that the sum of corresponding elements in $\boldsymbol{\rho}$ is maximized, under the constraint where any cell or blob appears on at most one hypothesis. This can be formulated as the following binary linear programming problem:

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}} \boldsymbol{\rho}^T \boldsymbol{x}, \quad s.t.\ C^T \boldsymbol{x} \leq \mathbf{1} \tag{2.1}$$

where $\boldsymbol{x} = (x_1, \cdots, x_H)^t$ is a $H \times 1$ binary vector, and $x_h = 1$ means the h-th hypothesis is selected in the optimal solution. The constraint $C^T \boldsymbol{x} \leq \mathbf{1}$ guarantees that each cell ID or blob ID appears in only one selected hypothesis.

To solve the binary linear programing problem, I use the branch-and-bound algorithm [Ross75], which divides the original problem into all possible sub-problem and then solves a series of LP-relaxation problems for each sub-problem. For the case of Figure 2.3, the optimal solution is that cell 1 migrates to blob $1'$, cell 2 divides to blob $2'$ and $3'$, cell 3, 4 overlap and make a cluster blob $4'$, and cell 5 migrates blob 5'.

When an overlapping event is selected in the optimal solution, the event is furthermore identified as either event (5) newly overlapping or (6) joint migration in a cluster by using the previous state of cells.

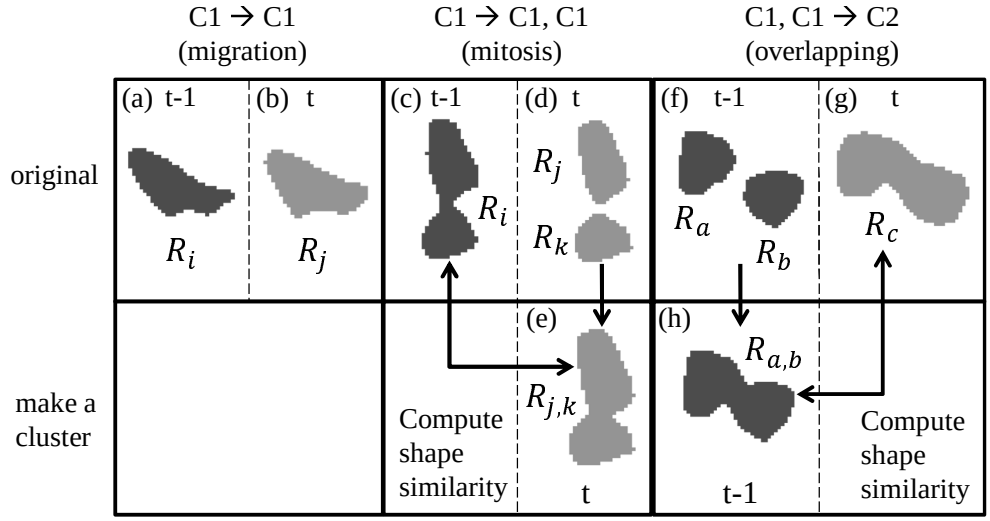


Figure 2.4: Upper : Cell shapes at $t-1$ and the detected blob shapes at t for each event at the original mask image. Bottom : When there are multiple cells or blobs, the method combined the shapes to a cluster. The mitosis event at t, $R_{j,k}$ is a combined cluster by R_j and R_k. The overlapping event at $t-1$, $R_{a,b}$ is a combined cluster by R_a and R_b.

2.2.3 Computing scores of hypotheses

In this section, the implementation of computing the score of the hypothesis for cell association is presented. The score of each hypothesis are computed based on the cell location similarity and shape similarity between a cell at frame t and a detected blob at $t-1$. The score of cell migration hypothesis is defined as:

$$\begin{aligned} P_{mig}(blob_j|cell_i) &= \omega_\zeta \cdot \zeta(R_i, R_j) + \omega_\eta \cdot \eta(R_i, R_j) \\ \omega_\zeta + \omega_\eta &= 1, \;\; 0 \leq \omega_\zeta, \omega_\eta \leq 1 \end{aligned} \tag{2.2}$$

where ω_ζ and ω_η are weights of each term, R_i indicates the detection results of cell or blob i, $\zeta(R_i, R_j)$ indicates position similarity that is the degree of overlapping between R_i and R_j (i.e., $\frac{R_i \cap R_j}{R_i \cup R_j}$), $\eta(R_i, R_j)$ indicates the shape similarity that is calculated by the distance of Fourier descriptors between R_i and R_j [FontouraCosta01]. For one-to-one association, such as the migration hypothesis, the shape similarity can be computed. However the shape similarity of one-to-two association, such as the division hypothesis and overlapping hypothesis, cannot be directly computed.

As shown in the Figures 2.4 (c) and (d), when a cell regions divide into two cells, the shapes of these two children cells are similar with the partial shapes before division. As shown in the Figures 2.4 (f) and (g), when two cells partially overlap and make a cluster, these two cells before overlapping are also similar with the partial shapes of the cluster. In

order to compute the shape similarity of mitosis hypothesis, I translate two regions of the blobs R_j, R_k and make a cluster $R_{j,k}$ until overlapped area ratio is 5% of the total as shown in Figure 2.4 (e). For overlapping hypothesis, I also make a cluster $R_{a,b}$ by translating the two cells R_a, R_b as shown in Figure 2.4 (h). Then, the score between the combined shape $R_{a,b}$ and the blob shape R_c is calculated. The scores are defined for each event as:

$$\begin{aligned} P_{div}(blob_j, blob_k|cell_i) &= \omega_\zeta \cdot \zeta(R_i, R_{j,k}) + \omega_\eta \cdot \eta(R_i, R_{j,k}) \\ P_{over}(blob_c|cell_a, cell_b) &= \omega_\zeta \cdot \zeta(R_{a,b}, R_c) + \omega_\eta \cdot \eta(R_{a,b}, R_c) \end{aligned} \tag{2.3}$$

For enter and exit hypotheses (i.e., none-to-one and one-to-none association), the spatial distance between the boundary of the field of view and the cell centroid since a cell does not newly appear in the field of view except boundary area. The scores are defined as:

$$P_{entr}(\text{enter}|blob_l) = \begin{cases} e^{-\frac{ds(R_l)}{\lambda_1}}, & \text{if } ds(R_l) < \theta_s \\ \xi & \text{otherwise } (\xi \text{ is small}) \end{cases} \tag{2.4}$$

$$P_{exit}(\text{exit}|cell_l) = \begin{cases} e^{-\frac{ds(R_l)}{\lambda_1}}, & \text{if } ds(R_l) < \theta_s \\ \xi & \text{otherwise } (\xi \text{ is small}) \end{cases} \tag{2.5}$$

where $ds(R_l)$ is the distance between the centroid of the detection region R_l and the image boundary. λ_1 is free parameters to adjust the distribution. I set $\lambda_1 = 1$.

2.2.4 Segmentation of overlapping cells

When the method detects event (5) and (6) that may involve a cluster of multiple cells, the method tries to segment the blob of overlapping cells into its member cells. I obtain the optimal contour matching of the blob contour with partial contours of candidate member cells [Bise09]. The method relies on the fact that as multiple cells touch or partially overlap, the blob contour of the resultant cluster must be comprised of partial contours of the member cells, and the contours of the member cells gradually deform with time while touched cells jointly migrate in the cluster (i.e., a contour shape of a member cell at the next frame is similar to that at the previous frame). Here, I first explain the partial matching method for event (6) joint migration since the method for event (5) can be consider as a special case for event (6).

Let me consider a case of tracking two overlapping cells whose respective boundary contours (partial or whole) in the previous frame are known. Figure 2.5 shows an example case. The partial contour shape of its member cells have been established for frame $t-1$ as shown in Figure 2.5 (a) (i.e., a cell contour is drawn by red, the other by white). Now the task is to match these with the blob contour in frame t shown in Figure 2.5 (b) as black contour.

The matching method proceeds in five steps: 1) In each contour, detect flexion points (local extremum points of curvature) that are candidate locations at which the contour is split; 2) Generate all possible combinations of matching flexion points between frame t-1 and frame t; 3) For each combination, compute the optimal partial matches between the contour segments split by the flexion points with the use of dynamic programming (DP) to account for cell shape deformation. Figure 2.5 (c) indicates cost matrices for DP matching, in which the green lines are optimal paths corresponding to the optimal matching; 4) Select the overall optimal partial matches among all combinations; and, 5) Propagate the cell identity information, and update the member cells' contour shape according to the selected match.

In the following description, I first consider the case in which two cells c_a^{t-1} and c_b^{t-1} overlap and form a cluster c_{ab}^t. The symbols c_a^{t-1} and c_b^{t-1} denote the contours of cells a and b in frame $t-1$, respectively. The contour of the cell cluster ab in frame t is denoted by c_{ab}^t. The problem is to find the best match between the partial segments of c_a^{t-1}, c_b^{t-1} and those of c_{ab}^t.

Flexion point detection and matching

Flexion points on a contour are defined as the points at which the curvature of the contour exceeds a certain magnitude. At each point $\vec{p}_i = (x_i, y_i)$ on a contour, the curvature κ_i is computed as

$$\kappa_i = \frac{|(\vec{p}_{i+1} - \vec{p}_i) \times (\vec{p}_{i-1} - \vec{p}_i)|}{|\vec{p}_{i+1} - \vec{p}_{i-1}||\vec{p}_{i+1} - \vec{p}_i||\vec{p}_{i-1} - \vec{p}_i|} \tag{2.6}$$

where $\vec{p}_{i-1}$ and $\vec{p}_{i+1}$ are neighbor points of $\vec{p}_i$; $\times$ denotes a cross product; and $|\cdot|$ is the Euclidean norm of a vector. A flexion point is detected if the curvature takes a local maximum and its absolute value satisfies $|\kappa_i| > th_\kappa$. The parameter is adjusted such that the number of frexion points on a blob roughly corresponds to the possible number of separation points of clustered cells.

Once flexion points are detected, the method generates a list of matching combinations between all flexion points on c_a^{t-1}, c_b^{t-1} and those on c_{ab}^t. Each combination is a quadruple in the form of $\{(\vec{f}_a^{\,l}, \vec{f}_{ab}^{\,k}), (\vec{f}_b^{\,m}, \vec{f}_{ab}^{\,k'})\}$, where $\vec{f}_a^{\,l}$, $\vec{f}_b^{\,m}$ represent flexion points on c_a^{t-1} and c_b^{t-1}, respectively, and $\vec{f}_{ab}^{\,k}, \vec{f}_{ab}^{\,k'}$ are two flexion points on c_{ab}^t. The combinations are generated using the following procedure:

- For each cell $\alpha \in \{a, b\}$:
  - for each pair $(\vec{f}_\alpha^{\,l}, \vec{f}_{ab}^{\,k})$, compute the matching distance

$$\begin{aligned} \mathrm{d}(\vec{f}_\alpha^{\,l}, \vec{f}_{ab}^{\,k}) &= w_{curv}\mathrm{d}_{curv}(\vec{f}_\alpha^{\,l}, \vec{f}_{ab}^{\,k}) \\ &+ w_{pos}\mathrm{d}_{pos}(\vec{f}_\alpha^{\,l}, \vec{f}_{ab}^{\,k}) \end{aligned} \tag{2.7}$$

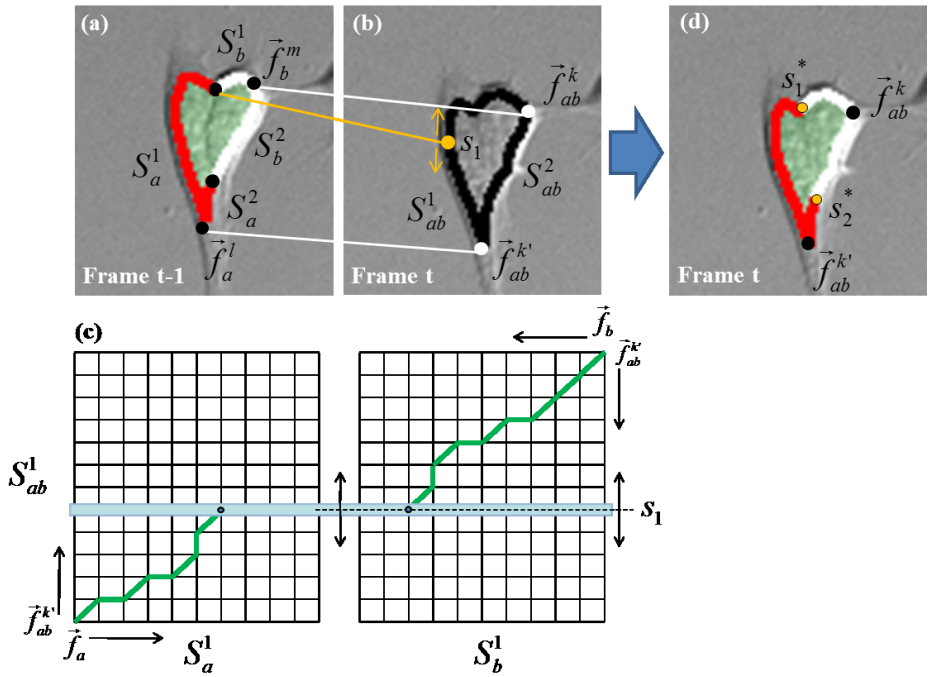


Figure 2.5: Flexion points and contour segment matching. (a),(b) A combination of matching flexion points in consecutive frames. (c) Cost matrices for dynamic programming, in which the green lines are back-traced optimal paths corresponding to the optimal matching. (d) Results of matched partial contours. The results match well with human perception of segmentation due to the faint dark boundary extending vertically (though that is accidental coincidence because that information has not been used yet by the proposed method).

  - Sort the matching distances in ascending order, and select the top $\min(Np, 5)$ pairs, where Np is the total number of flexion point pairs.

- Enumerate all combinations $\{(\vec{f}_a^{\,l}, \vec{f}_{ab}^{\,k}), (\vec{f}_b^{\,m}, \vec{f}_{ab}^{\,k'})\}$ among the selected pairs.

In Eq. 2.7, $\mathrm{d}(\vec{f}_\alpha^{\,l}, \vec{f}_{ab}^{\,k})$ denotes the dissimilarity measure between flexion points $\vec{f}_\alpha^{\,l}$ and $\vec{f}_{ab}^{\,k}$, which is a weighted combination of the curve difference d_{curv} and the position distance d_{pos} with weights w_{curv} and w_{pos}. These parameters are adjusted so that $w_{curv}\mathrm{d}_{curv}$ and $w_{pos}\mathrm{d}_{pos}$ are of the same order of magnitude. The curvature distance is defined by

$$\mathrm{d}_{curv}(\vec{f}_\alpha^{\,l}, \vec{f}_{ab}^{\,k}) = \frac{1}{2P+1} \sum_{\kappa_i \in F_\alpha^l, \kappa_j \in F_{ab}^k} |\kappa_i - \kappa_j| \tag{2.8}$$

where F_α^l and F_{ab}^k are contour segments of length $2P+1$ centered at $\vec{f}_\alpha^{\,l}$ and $\vec{f}_{ab}^{\,k}$, respectively. The position distance $\mathrm{d}_{pos}(\vec{f}_\alpha^{\,l}, \vec{f}_{ab}^{\,k})$ is simply the 2D Euclidean distance.

Matching contour segments by dynamic programming

For each combination of matching flexion points $\{(\vec{f}_a^{\,l}, \vec{f}_{ab}^{\,k}), (\vec{f}_b^{\,m}, \vec{f}_{ab}^{\,k'})\}$, the contours are split into segments:

$$S_a^1 = \langle \text{start}(c_a^{t-1}), \vec{f}_a^{\,l} \rangle, S_a^2 = \langle \vec{f}_a^{\,l}, \text{end}(c_a^{t-1}) \rangle,$$
$$S_b^1 = \langle \text{start}(c_b^{t-1}), \vec{f}_b^{\,l} \rangle, S_b^2 = \langle \vec{f}_b^{\,l}, \text{end}(c_b^{t-1}) \rangle,$$
$$S_{ab}^1 = \langle \vec{f}_{ab}^{\,k}, \vec{f}_{ab}^{\,k'} \rangle, \textit{and } S_{ab}^2 = \langle \vec{f}_{ab}^{\,k'}, \vec{f}_{ab}^{\,k} \rangle$$

Here, $\langle \vec{p}_1, \vec{p}_2 \rangle$ represents the contour segment from $\vec{p}_1$ to $\vec{p}_2$, and $\text{start}(c_a^{t-1})$, $\text{end}(c_a^{t-1})$ represent the endpoints of contour c_a^{t-1}. Note that for a closed contour, the endpoints coincide with the flexion point. Figure 2.5 illustrates the contour segments corresponding to a matching combination of flexion points in two successive frames.

Dynamic programming is used to compute the optimal partial matching between (S_a^n, S_b^n) and S_{ab}^n ($n \in \{1, 2\}$). The method consists of two steps.

First, two cost matrices are constructed: one for computing the partial matching between S_a^n and S_{ab}^n, denoted by $\mathbf{G}_a^n$; the other for the partial matching between S_b^n and $\bar{S}_{ab}^n$, denoted by $\mathbf{G}_b^n$, where $\bar{S}_{ab}^n$ denotes the reverse of S_{ab}^n. Suppose that segment S_{ab}^n consists of P_a^n points $\{\vec{p}_i | i = 1, \ldots, P_{ab}^n\}$, and S_a^n of P_a^n points $\{\vec{p}_j | j = 1, \ldots, P_a^n\}$. Then $\mathbf{G}_a^n$ is an $P_{ab}^n \times P_a^n$ matrix with entries

$$g_a^n(i,j) = \min \begin{cases} g_a^n(i-1, j-2) + 2\text{u}(i, j-1) + \text{u}(i,j), \\ g_a^n(i-1, j-1) + 2\text{u}(i,j), \\ g_a^n(i-2, j-1) + 2\text{u}(i-1, j) + \text{u}(i,j) \end{cases} \tag{2.9}$$

where $\text{u}(i,j) = |\kappa_i - \kappa_j|$ is the unsigned distance between the curvatures at $\vec{p}_i$ and $\vec{p}_j$ for $i, j > 0$, and $\text{u}(i,j) = 0$ otherwise. The values of $g_a^n(i,j)$ for $i <= 0$ and/or $j <= 0$ are defined as:

$$g_a^n(i,j) = \begin{cases} 0, & \text{if } (i \leq 0 \text{ and } j \leq 0), \\ \textit{Large}, & \text{otherwise}, \end{cases} \tag{2.10}$$

where *Large* represents a large positive value that is greater than the total cost of any path in the cost matrix. The cost matrix $\mathbf{G}_b^n$ is constructed analogously. Then, the method computes the optimal separation point $\vec{s}_n$ on segment S_{ab}^n that splits S_{ab}^n into two partial segments, $S_{ab}^{n,a}$ and $S_{ab}^{n,b}$, such that the overall matching cost between these partial segments and S_a^n, S_a^n is minimized. The overall matching cost $Cost(\vec{s}_1, \vec{s}_2)$ for $\vec{s}_1 \in S_{ab}^1$ and $\vec{s}_2 \in S_{ab}^2$

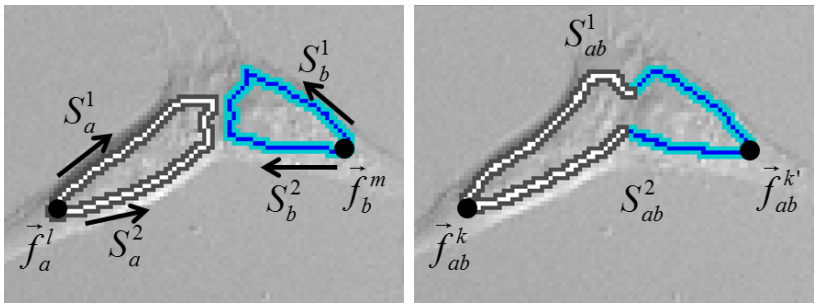


Figure 2.6: Flexion points and contour segment matching for event (5) when two individual cells newly touch and form a cluster. The left image shows the previous frame, the right image shows the next frame.

is defined as:

$$\begin{aligned} Cost(\vec{s}_1, \vec{s}_2) = cost(S_a^1, S_{ab}^{1,a}) + cost(S_b^1, S_{ab}^{1,b}) \\ + cost(S_a^2, S_{ab}^{2,a}) + cost(S_b^2, S_{ab}^{2,b}) \\ + \omega_\kappa(\kappa_1 + \kappa_2), \\ \text{with } \; cost(S_a^n, S_{ab}^{n,a}) = \min_j g_a^n(i_{\vec{s}_n}, j), \\ \text{and } \; cost(S_b^n, S_{ab}^{n,b}) = \min_j g_b^n(i_{\vec{s}_n}, j) \end{aligned} \tag{2.11}$$

Because the contours near the separating points are more likely to be locally concave than convex, the curvatures κ_1,κ_2 at $\vec{s}_1$ and $\vec{s}_2$ are added to the cost function in Eq. 2.11 with weighting coefficient ω_κ in order to penalize convex, thus favoring concave, separation points.

With the optimal separation points identified, the optimally matching partial contours are obtained by back tracing the corresponding cost matrices.

Figure 2.5 shows the result of this process. The combination of flexion point matching shown is the one that has produced the best match and the resultant assignments of the partial contour is shown in Figure 2.5 (d).

Before concluding this sub-section, it is worthwhile to mention that partial shape matching techniques using dynamic programming have been used in shape retrieval applications [Petrakis02][Milios00] for handling distorted shapes. In my application, I must handle multiple shapes that (partially) overlap. However, considering all combinations of all possible endpoints of the overlapping is computationally very expensive. The use of flexion point (Step 1) allows efficiently identifying probable endpoints of partial overlap.

Newly overlapping case

Segmentation method for a case when two cells newly touch and form a cluster (i.e., event (5)) is a special case of segmentation for event (6). The difference is the definition of the segments of contours. For each combination of matching flexion points $\{(\vec{f}_a^l, \vec{f}_{ab}^k), (\vec{f}_b^m, \vec{f}_{ab}^{k'})\}$, the contours are split into segments:

$$S_a^1 = \langle \vec{f}_a^l, \vec{f}_a^l \rangle_{clock}, S_a^2 = \bar{S}_a^1,$$
$$S_b^1 = \bar{S}_b^2, S_b^2 = \langle \vec{f}_b^l, \vec{f}_b^l \rangle_{clock},$$
$$S_{ab}^1 = \langle \vec{f}_{ab}^k, \vec{f}_{ab}^{k'} \rangle, and\ S_{ab}^2 = \langle \vec{f}_{ab}^{k'}, \vec{f}_{ab}^k \rangle$$

where $\langle \vec{f}_a^l, \vec{f}_a^l \rangle_{clock}$ represents a whole contour of cell c_a^{t-1} with end point $\vec{f}_a^l$ in a clockwise direction, S_a^2 is the reverse of S_a^1. To compute the optimal partial matching, the same process for event (6) is used since the dynamic programming can compute a matching cost between a partial contour of S_a^1 and the contour of segment S_{ab}^1. Figure 2.6 illustrates the contour segments corresponding to a matching combination of flexion points in two successive frames for event (5).

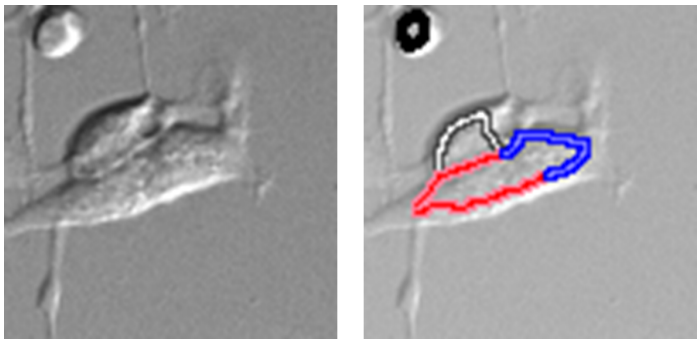


Figure 2.7: Example of segmenting a three-cell cluster into members. Left: Original image. Right: Segmentation results.

Handling N-cell clusters ($N > 2$)

So far I have discussed partial contour matching for the two-cell cluster case. It is straightforward to extend the method to handle three or more cells. Consider the example that the contours of three cells c_a^{t-1}, c_b^{t-1} and c_d^{t-1} in frame $t-1$ merge into a single contour c_{abd}^{t-1} in frame t. Matching can be obtained by first considering the contours c_a^{t-1} and $c_{bd}^{t-1} = c_b^{t-1} \bigcup c_d^{t-1}$, and computing their optimal partial matching to c_{abd}^t using the above method. Then, the partial contour of c_{abd}^t that is matched to c_{bd}^{t-1} is converted into a closed contour c_{bd}^t, by interpolation. The above method is applied again to obtain a matching between c_b^{t-1}, c_d^{t-1} and c_{bd}^t. Further extensions can be made by following the same approach.

Figure 2.7 shows an example in which three cells are involved. In practice, however, tracking individual motions of four or more overlapping cells with unclear intercellular boundaries is extremely difficult, even for expert cell biologists. I propose tracking methods to solve the problems in Chapters 5 and 6.

2.3 Experiments and results

2.3.1 Data

I tested two DIC microscopy image sequences of human CNS stem cell populations, each of which are captured over 66 hours using 12-bit Orca ER (Hamamatsu) CCD camera mounted on a Zeiss Axiovert 135 TV microscope with a 40x, 1.3 NA oil-immersion DIC objective. A 0.6x lens was installed in front of the camera to increase the visual field. The image size is 640×512 pixels. One of the sequence (seq 1) was captured every 5 minutes for 800 frames (66.66 hours), and another sequence (seq 2) was captured every 2.5 minutes for 1600 frames (66.66 hours). The cell population varied in the range of 16 to 50 cells per frame, and cells often overlapped with each other. Manual cell tracking was performed by an expert biologist for a total of whole frames containing 39206 cells which appeared in the initial frame and their progeny. The results were confirmed by two other biologists and served as ground truth.

For implementation, I set the parameters $d_e = 30$, $th_k = 0.5$, $w_{curv} = 0.9$, and $w_{pos} = 0.9$. The code of the proposed method was implemented by MATLAB.

2.3.2 Tracking examples

Two examples of tracking two and three partially overlapping cells are shown in Figures 2.8 (a) and (b). Figures 2.8 (a-1) and (b-1) show the tracking results by a method utilizing level-set and motion filter [Li08] which failed separating overlapping cells and tracking one of the cells. The results by the proposed method, shown in Figures 2.8 (a-2) and (b-2), successfully tracked all the cells throughout the long overlapping period.

Figure 2.9 shows a more complicated example that involves cell mitosis (cell division). Frame 920 contains two separated cells (shown by white and blue contours). In frame 936, the white and blue cells began to touch, and by frame 963, they overlapped fairly severely. In frame 981 and 982, the blue cell divided to two cells (shown by red and purple). Then, these three cells made a cluster, and in the following sequence, the 3-cell cluster returned back to three separated cells. The tracking method had correctly tracked through the event in spite of shape deformation. The right-hand side figure on Figure 2.9 shows the space-time (x-y-t) trajectory plot.

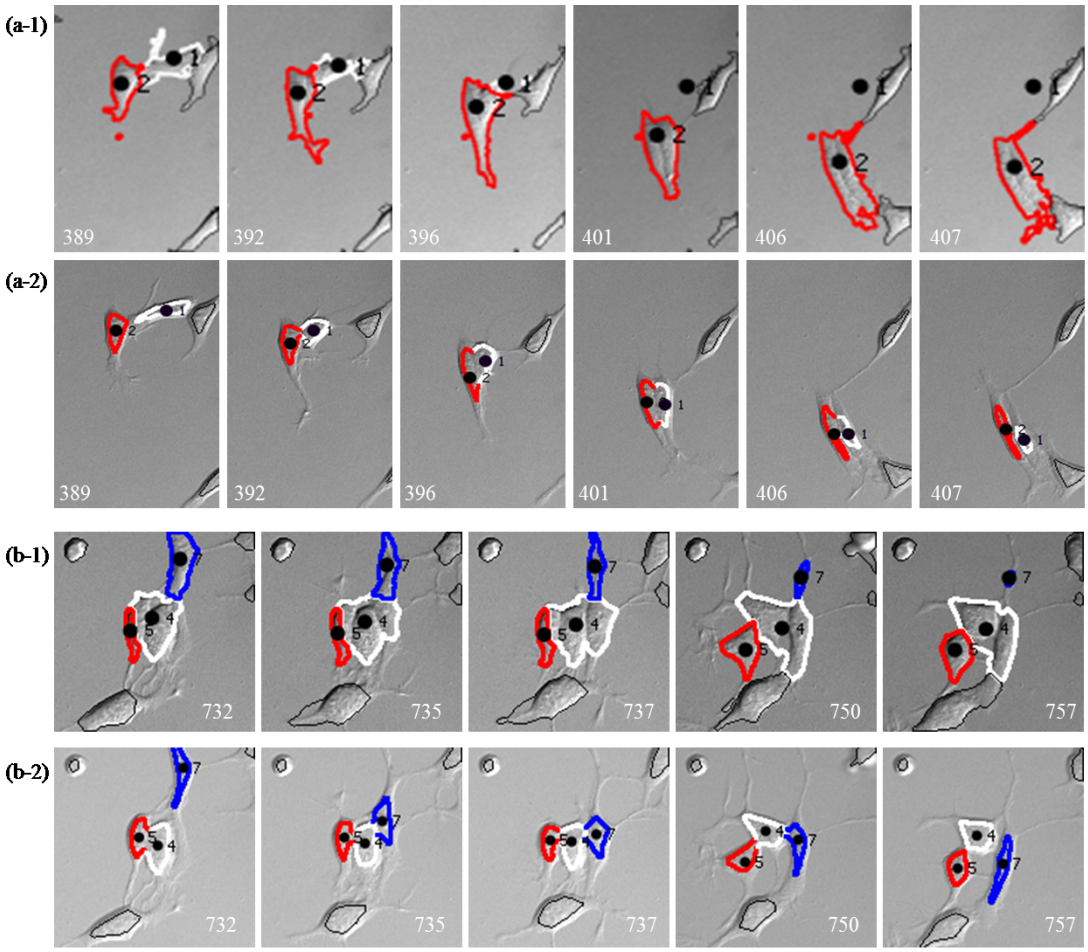


Figure 2.8: Examples of tracking results. (a-1), (b-1): Results by a method using level-set and IMM Filter [Li08]. (a-2) (b-2): Results by the proposed method. Black dots are centroid of cell contour.

Figure 2.10 (a) shows a result for one whole frame, and the total results of tracking and cell identification for the whole sequence can generate a time-space tree, shown in Figure 2.10 (b), which represents the complete motions of all the cells as well as their lineage information.

2.3.3 Quantitative evaluation

Detection accuracy.

Table 2.1 summarizes detection accuracy for two sequences where the detection results were provided after the re-segmentation of overlapped cells. The results include a total of 39206 true positives (TPs), 2341 false positives (FPs), and 253 false negatives (FNs). In seq 2, since dead cells were detected as cell region, FP rate were larger than one of seq 1. In terms

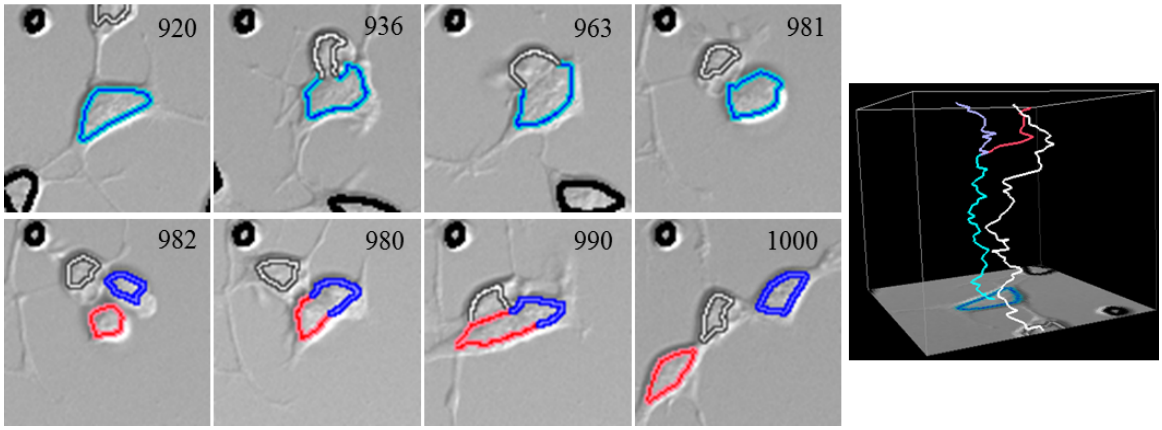


Figure 2.9: More complicated example of tracking cells through a sequence that involves touching, overlapping, separation and mitosis.

	#Frame	#Cell	TP	FP	FN	Precision	Recall
seq 1	800	24683	24534	842	149	0.967	0.994
seq 2	1600	14523	14419	1499	104	0.906	0.993
Total	2400	39206	38953	2341	253	0.943	0.994

Table 2.1: Accuracy of cell identification.

of $precision = \frac{TP}{TP+FP}$ and $recall = \frac{TP}{TP+FN}$, the proposed method achieved a precision of 94.3%, and a high recall of 99.4%. The details of the metrics are described in Appendix A.

Tracking accuracy for touching events.

To show the proposed method capability of tracking touching cells, I evaluated the tracking accuracies for different types of cell cluster motion events (formation, migration, separation) in the two sequences that involve up to three cells as shown in Table 2.2. In the table, $C1$ stands for one cell, and $C2$ and $C3$ correspond to cell clusters containing two and three cells, respectively. "$C1, C1 \rightarrow C2$" stands for the event in which two cells come to overlap to form one cluster; "$C2 \rightarrow C2$" for the event of two cells migrating jointly as a cluster; "$C3 \rightarrow C1, C2$" for the event of a three-cell cluster separating into one cell and a two-cell cluster; and so on. The method recognized these different cases explicitly. Overall, a 96.3% accuracy was achieved.

Target effectiveness

I used target effectiveness [Blackman86] to assess the system performance. The definition of target effectiveness is described in Appendix A. Figure 2.11 shows the target effectiveness

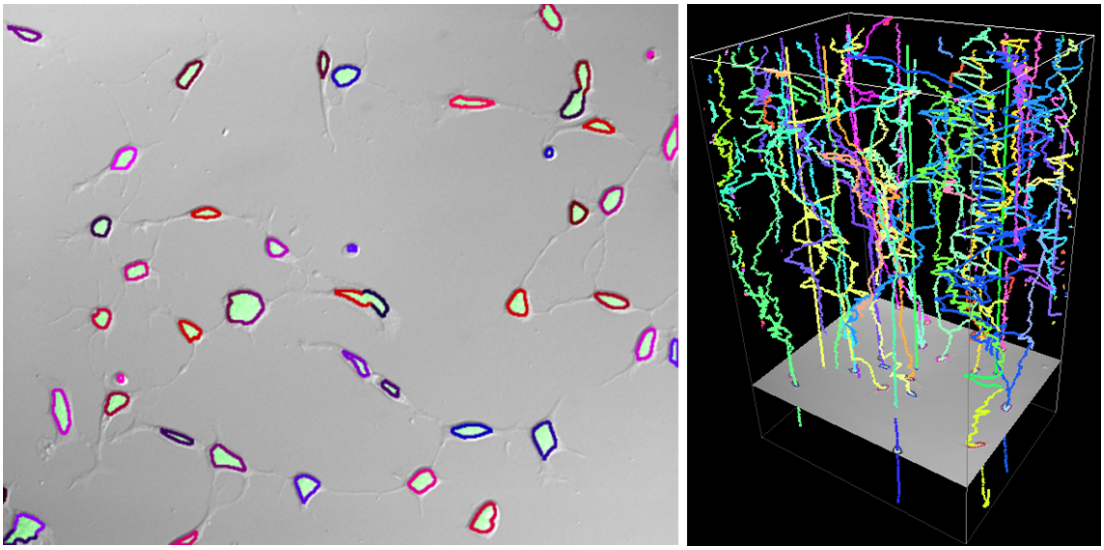


Figure 2.10: Left: Example of tracking results for a frame. Right: Complete space-time track representation.

Event	Count	Errors	Accuracy
$C1, C1 \rightarrow C2$	200	9	0.955
$C2 \rightarrow C2$	921	25	0.973
$C2 \rightarrow C1, C1$	181	10	0.945
$C1, C2 \rightarrow C3$	27	3	0.889
$C3 \rightarrow C3$	57	4	0.930
$C3 \rightarrow C1, C2$	26	2	0.923
Total	1412	53	0.963

Table 2.2: Tracking accuracies of cell cluster motion for various events.

over time. Given a specific time, the metric was computed with only tracks that are formed until the time. It is shown that the overall performance had a tendency to reduce somewhat since the number of cells increases and they more often touch and overlap. The final values of the last frame was 98% in seq 1, and 91% in seq 2.

Mitosis branching correctness

Cell lineage is especially important for stem cell research [Ravin08][Chisholm00], as it provides fine-grain quantification measurements of mitosis, such as division time and synchrony of mitosis. To accurately generate cell lineages, the mitotic branching correctness [Bise11] is also an important measure. The mitotic branching correctness measured the accuracy of

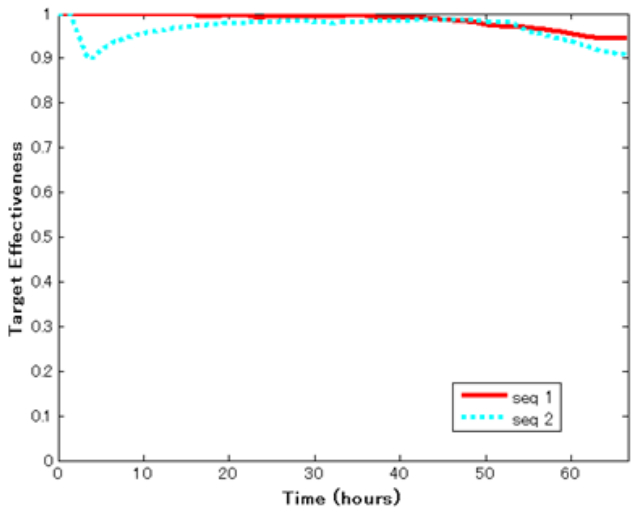


Figure 2.11: Accumulated target effectiveness over time for the two test sequences. Red line is seq 1. Blue dots line is seq 2.

mother-daughter relationships between tree branches. The correctness of mitotic branching is the number of the correctly detected mitosis branching over the total number of the mitotic events. In the evaluation, I set the parameter as $\theta_\epsilon = 10$. The details of the metrics is described in Appendix A.

The proposed method achieved 91% accuracy of mitosis branching correctness in sequence 1 and 94% accuracy in sequence 2. Figure 2.12 shows an example of the linage trees compared to human annotated one. Horizontal red line indicate tracks that follow the ground-truth, vertical red lines indicate that the mitosis branching is correctly detected on the branch nodes of the linage tree. The results show that the lineage trees were well constructed.

2.4 Conclusion

I have presented a cell-tracking method based on partial contour matching using dynamic programming. The method is capable of tracking migrating cells that sometimes partially overlap, while maintaining the identity information of individual cells throughout the process from their initial contact to eventual separation. Therefore, the proposed method have achieved high accuracy where 91% of the paths from the initial cells to the descendant cells in the last frame are fully correct in both its mother-daughter relationships and its locational trajectories despite frequent formation of multiple-cell clusters.

The proposed method can track two or three touching cells, however, it still remains several challenges. When false positives of cell detection appears near a cell, the method

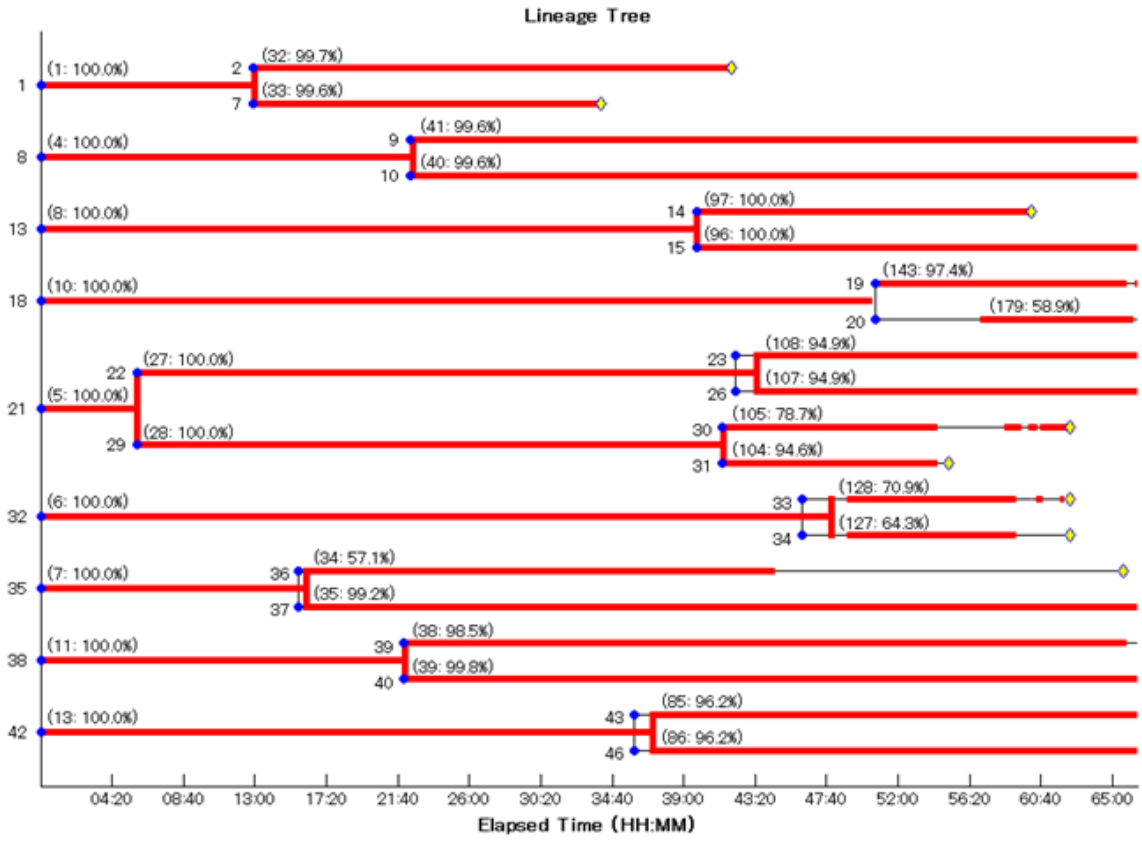


Figure 2.12: Tracking performance evaluation with lineage trees in terms of target effectiveness. The black lines indicate the lineage tree of the ground truth. The red lines indicate correctly tracked parts in the ground truth. The numbers located in the front of each line indicate track/cell IDs. The horizontal axes indicate time (HH:MM).

may confuse the false positive as a divided cell. For such case, I propose a cell-tracking method by global data association in Chapter 3

In practice, tracking individual motions of four or more overlapping cells with unclear intercellular boundaries is extremely difficult, even for expert cell biologists. For such dense culture conditions, I developed the other methods. These methods are explained in Chapters 5 and 6.

Chapter 3

Cell Tracking by Global Data Association

3.1 Introduction

Frame-by-frame association methods, such as the method proposed in Chapter 2, exhibit high tracking accuracy based on trajectory-level evaluation (how well ground-truth cells are followed by computer-generated tracks). However, it is still challenging to achieve high accuracy based on lineage-level (tree structure) evaluation including the correctness of the mother-daughter relationship. For example, when a false positive segmentation appears near a mitotic cell, the local temporal association methods may result in a mother-daughter relationship error. Global temporal information is important to solve this problem. If we observe the cells for several frames after the birth event, we can easily determine that one of the children cells is false positive since false positives usually quickly disappear. This allows us to correct the relationship.

Recently, global spatio-temporal data association methods have been proposed for general object tracking. Multi-Hypothesis Tracking (MHT) [Reid79] and Joint Probabilistic Data Association Filters (JPDAF) [Fortmann83] are two representative examples for associating multiple trajectories over time. Tracklet stitching [Huang09] was proposed for reducing computational cost. Huang *et al.* [Huang09] first generated reliable tracklets that are fragments of tracks formed by conservative grouping of detection responses. The tracklets were then connected using the Hungarian algorithm [Kuhn55]. Bonneau *et al.* [Bonneau05] proposed a tracklet linking method with which a minimal path between tracklets is obtained using dynamic programming in order to track quantum dots in a living cell. Zhang *et al.* [Zhang08] proposed a minimum-cost flow network to resolve the global data association of multiple objects over time. These global association methods are known to exhibit higher accuracy of tracking general objects than frame-by-frame association meth-

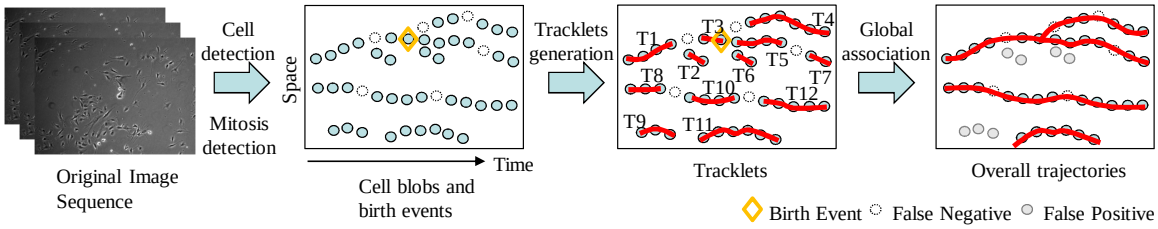


Figure 3.1: Method overview.

ods. However, these methods cannot be applied to cell tracking directly since they do not take into account cell division (a mother cell divides into two daughter cells to form a tree structure in the trajectory).

In this chapter, I propose a global spatio-temporal data association method for tree structures to obtain cell trajectories and lineage trees. Reliable tracklets (i.e., short trajectories) are first generated by linking detection responses based on frame-by-frame association. The global tracklet association for the tree structures is then formulated as a maximum-a-posteriori (MAP) problem. The MAP problem is solved by linear programming to provide the cell trajectories and lineage trees. This method was evaluated on five sequences with thousands of cells captured over several days. The results show an improvement in tracking performance compared to the state-of-the-art method proposed by Li *et. al.* [Li08].

3.2 Method

Figure 3.1 shows the overview of my cell-tracking method. First, the cell-detection step segments cell blobs from input images that may include false positives and false negatives, and the mitosis detection step locates birth events where and when one cell divides into two cells. Next, the detected cell blobs are associated to reliable tracklets by frame-by-frame data association. Finally, the global association step associates the tracklets to obtain cell trajectories and lineage trees.

3.2.1 Cell detection

Due to the interference optics of a phase contrast microscope, cells are surrounded by bright halos, and cellular fluid inside the membrane has similar intensity as the background. To facilitate segmentation, I have adopted the image restoration technique recently developed in [Yin10a]. The technique utilizes the optophysical principle of image formation by phase contrast microscope, and transforms an input image to an artifact-free image by minimizing a regularized quadratic cost function. In the restored image, cells appear as regions of positive values against a uniformly-zero background. A simple thresholding method, such

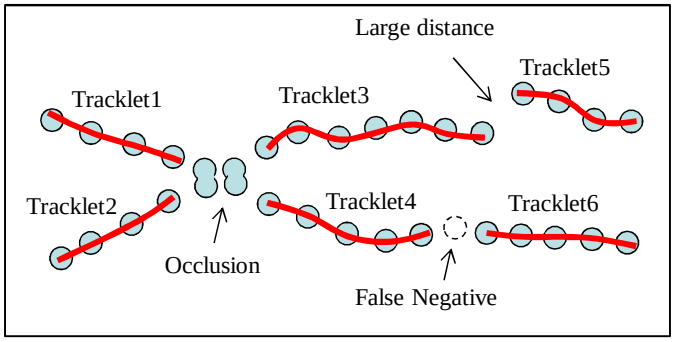


Figure 3.2: Examples of tracklets.

as Otsu thresholding, can segment out the cell regions. The set of detection results is denoted as $\mathbf{R} = \{R_i\}$ where R_i represents the i-th cell blob.

3.2.2 Cell mitosis event detection

To detect the birth events (time and location at which one cell divides into two cells), I have adopted the mitosis detection technique recently developed in [Huh11]. Firstly, as a mitosis event generally exhibits an increase of brightness, bright regions are extracted as patches, and then candidate patch sequences are constructed by associated patches. Next, the gradient histogram features are extracted from the patches. Finally, a probabilistic model named Event Detection Conditional Random Field (EDCRF) is applied to determine whether each candidate patch sequence contains a birth event and which frame the birth event is located in. The set of the detected mitosis events is represented as $\mathbf{M} = \{M_i\}$ where M_i is a detected mitosis event.

3.2.3 Tracklet generation

Since a long trajectory obtained via frame-by-frame association may include more failures, such as drift and occlusion, than a short trajectory, the detected blobs are first associated to make reliable tracklets. A tracklet is considered reliable when cell blobs in consecutive frames are close enough, and there are no extra confusing blobs near the cell. Figure 3.2 shows examples of extracted tracklets in which two cells migrate. In this example, when the occlusions occur, it is not clear if a detection response right after the occlusion is associated with tracklet 1 or 2, so tracklet 1 and 2 are terminated at that time. False negatives and large distance between the blobs also cause uncertain association, so tracklets 3 and 4 are also terminated at that time.

For the implementation, I use a frame-by-frame association method to generate tracklets. The cell association method makes hypotheses of all possible cell translation and computes

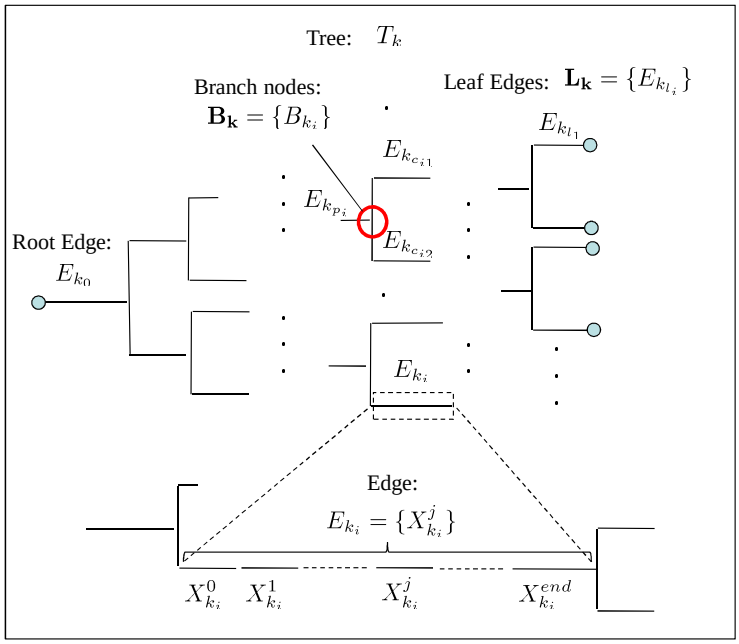


Figure 3.3: Example of a tree structure hypothesis. Bottom illustration shows zooming of an edge which consists of tracklets.

their score as:

$$P_{link}(b_j|c_i) = e^{-\frac{\|f(c_i)-f(b_j)\|}{\sigma}}$$

where c_i represents the i-th cell in the previous frame and b_j represents the j-th blob in the current frame. $f(\cdot)$ computes an object's feature vector where different types of features can be incorporated such as appearance histogram, shape and motion history. Then, the optimal association from the hypothesis set is found by solving a binary optimization problem which is similar to the optimization approach proposed in Chapter 2. The detected blob is linked to a cell if and only if their score is higher than a threshold.

Based on the frame-by-frame association, a set of reliable tracklets $\mathbf{X} = \{X_i\}$ is generated. The term $X_i = \{R_{i_j}\}$ is a tracklet consisting of an order list of detection results where R_{i_j} indicates the j-th detection result on tracklet X_i. Any isolated detection response that is not linked with any other one is considered as a tracklet and also included in $\mathbf{X}$. Unclear associations are solved on the next step by using the global data association.

3.2.4 Global data association

In this section, I propose a global data association method which addresses the tree structure association problem.

Let $\mathbf{T} = \{T_k\}$ be a hypotheses set of cell trajectory trees over the entire video. Each tree T_k, corresponding to a cell family from the ancestor to all of its descendents, is formed by

associated tracklets. I define a tree structure hypothesis on T_k using the following notations (Figure 3.3):

1. $\mathbf{E_k} = \{E_{k_i}\}$: a set of edges of tree T_k. Each edge is defined as an order list of tracklets, i.e., $E_{k_i} = \{X^j_{k_i}\}$ where $X^j_{k_i}$ is j-th tracklet on edge E_{k_i}. Specifically, E_{k_0} denotes the root edge of the tree.
2. $\mathbf{B_k} = \{B_{k_i}\}$: a set of branch nodes of tree T_k. Each branch node B_{k_i} defines a parent-children relationship, $B_{k_i} = \{E_{k_{p_i}}, E_{k_{c_{i1}}}, E_{k_{c_{i2}}}\}$ ($E_{k_{p_i}}$ is a parent, and $E_{k_{c_{i1}}}, E_{k_{c_{i2}}}$ are children.)
3. $\mathbf{L_k} = \{E_{k_{l_i}}\}$: a set of leaf edges of tree T_k.

Given the observation tracklet set $\mathbf{X}$, the posteriori probability is maximized to solve for the best hypothesis $\mathbf{T}^*$.

$$\begin{aligned}\mathbf{T}^* &= \arg\max_{\mathbf{T}} P(\mathbf{T}|\mathbf{X}) \\ &= \arg\max_{\mathbf{T}} P(\mathbf{X}|\mathbf{T})P(\mathbf{T}) \\ &= \arg\max_{\mathbf{T}} \prod_{X_i \in \mathbf{X}} P(X_i|\mathbf{T}) \prod_{T_k \in \mathbf{T}} P_{Tree}(T_k) \end{aligned} \tag{3.1}$$

In Eq. 3.1, I assume that the likelihoods of input tracklets are conditionally independent given $\mathbf{T}$, and $T_k \in \mathbf{T}$ can not overlap with each other, i.e., $T_k \cap T_l = \phi, \forall k \neq l$. The likelihood of observed tracklet X_i is

$$P(X_i|\mathbf{T}) = \begin{cases} P_{TP}(X_i), & \text{if } \exists T_k \in \mathbf{T}, X_i \in T_k \\ P_{FP}(X_i), & \text{otherwise} \end{cases} \tag{3.2}$$

where $P_{TP}(X_i)$ is the probability for X_i being a true positive, and $P_{FP}(X_i)$ is the probability for X_i being a false alarm. $P_{tree}(T_k)$ is modeled as a Markov chain:

$$\begin{aligned} P_{Tree}(T_k) &= P_{ini}(E_{k_0}) \times \prod_{E_{k_i} \in T_k} P_{edge}(E_{k_i}) \\ &\quad \times \prod_{\{E_{k_{p_i}}, E_{k_{c_{i1}}}, E_{k_{c_{i2}}}\} \in \mathbf{B_k},\ \mathbf{B_k} \in T_k} P_{div}(E_{k_{c_{i1}}}, E_{k_{c_{i2}}}|E_{k_{p_i}}) \\ &\quad \times \prod_{E_{k_{l_i}} \in \mathbf{L_k},\ \mathbf{L_k} \in T_k} P_{term}(E_{k_{l_i}}) \end{aligned} \tag{3.3}$$

where P_{ini} is an initialization probability on the root of the tree, and P_{term} is a termination probability on a leaf of the tree. $P_{div}(E_{k_{c_{i1}}}, E_{k_{c_{i2}}}|E_{k_{p_i}})$ is an edge dividing probability in which edge $E_{k_{p_i}}$ divides into two edges $E_{k_{c_{i1}}}$, $E_{k_{c_{i2}}}$. Under the Markov assumption, the edge probability can be formulated as:

$$P_{edge}(E_{k_i}) = \prod_{j=1:N_{k_i}-1} P_{link}(X^j_{k_i}|X^{j-1}_{k_i}) \tag{3.4}$$

where $P_{link}(X_{k_i}^j | X_{k_i}^{j-1})$ is the probability to link tacklets $X_{k_i}^j$ and $X_{k_i}^{j-1}$ together, N_{k_i} is the number of tracklets on the edge E_{k_i}. Let $X_{k_i}^0$ be the first tracklet of E_{k_i}, and $X_{k_i}^{end}$ be the last tracklet of E_{k_i}. Under the Markov assumption, the initialization, termination, and dividing probabilities can be formulated as:

$$P_{ini}(E_{k_0}) = P_{ini}(X_{k_0}^0), \tag{3.5}$$

$$P_{term}(E_{k_{l_i}}) = P_{term}(X_{k_{l_i}}^{end}), \tag{3.6}$$

$$P_{div}(E_{k_{c_{i1}}}, E_{k_{c_{i2}}} | E_{k_{p_i}}) = P_{div}(X_{k_{c_{i1}}}^0, X_{k_{c_{i2}}}^0 | X_{k_{p_i}}^{end}) \tag{3.7}$$

After substituting Eqs. 3.2-3.7 into Eq. 3.1, I take a logarithm on the objective function:

$$\begin{aligned}
\mathbf{T}^* \;=\; & \arg\max_{\mathbf{T}} \{ \sum_{X_i \notin T_k,\ \forall T_k \in \mathbf{T}} \log P_{FP}(X_i) \\
& + \sum_{X_i \in T_k,\ \forall T_k \in \mathbf{T}} \log P_{TP}(X_i) \\
& + \sum_{X_{k_0}^0 \in E_{k_0},\ E_{k_0} \in T_k,\ \forall T_k \in \mathbf{T}} \log P_{ini}(X_{k_0}^0) \\
& + \sum_{X_{k_i}^j,\ X_{k_i}^{j-1} \in E_{k_i},\ \forall E_{k_i} \in T_k,\ \forall T_k \in \mathbf{T}} \log P_{link}(X_{k_i}^j | X_{k_i}^{j-1}) \\
& + \sum_{\{X_{k_{p_i}}^{end},\ X_{k_{c_{i1}}}^0,\ X_{k_{c_{i1}}}^0\} \in \mathbf{B_k},\ \mathbf{B_k} \in T_k,\ \forall T_k \in \mathbf{T}} \log P_{div}(X_{k_{c_{i1}}}^0, X_{k_{c_{i2}}}^0 | X_{k_{p_i}}^{end}) \\
& + \sum_{X_{k_{l_i}}^{end} \in E_{k_{l_i}},\ \forall E_{k_{l_i}} \in \mathbf{L_k},\ \mathbf{L_k} \in T_k,\ \forall T_k \in \mathbf{T}} \log P_{term}(X_{k_{l_i}}^{end}) \}
\end{aligned} \tag{3.8}$$

The above MAP problem is solved by binary linear programming. Let $N_{\mathbf{X}}$ be the number of tracklets in the entire sequence, vector $\boldsymbol{\rho}$ stores the likelihoods of every possible hypothesis and matrix C stores the constraints to avoid conflict hypotheses, where each row of C has $2N_{\mathbf{X}}$ columns and each column indicates tracklet index on the association between two tracklets. The entries of $\boldsymbol{\rho}$ and C are computed based on the following hypotheses.

1. Initialization hypothesis:
   If the first blob of tracklet X_k appears in the beginning of the sequence or appears near the boundary of the field of view, the tracklet is a candidate of a initial tracklet. Let h be the index of a new hypothesis, I append a new row to C and a corresponding likelihood to $\boldsymbol{\rho}$:

$$\begin{aligned}
C(h,i) &= \begin{cases} 1, & \text{if } i = N_{\mathbf{X}} + k \\ 0, & \text{otherwise.} \end{cases} \\
\boldsymbol{\rho}(h) &= \log P_{ini}(X_k) + 0.5 \log P_{TP}(X_k)
\end{aligned}$$

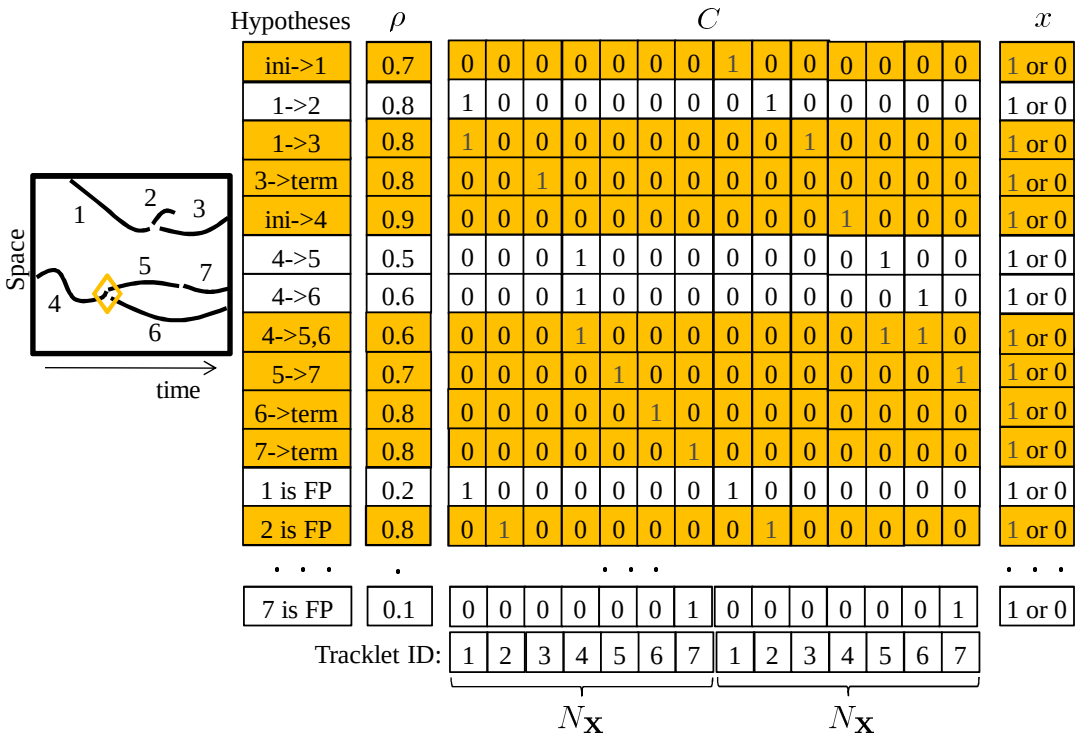


Figure 3.4: Example of binary linear programming where the optimal solution is highlighted by orange.

2. Termination hypothesis:
   If the last blob of tracklet X_k appears in the end of the sequence or appears near the boundary of the field of view, the tracklet is a candidate of a termination tracklet. New entries for C and $\boldsymbol{\rho}$ are defined as:

$$\begin{aligned} C(h,i) &= \begin{cases} 1, & \text{if } i = k \\ 0, & \text{otherwise.} \end{cases} \\ \boldsymbol{\rho}(h) &= \log P_{term}(X_k) + 0.5 \log P_{TP}(X_k) \end{aligned}$$

3. Translation hypothesis:
   If the time and space distances between the last blob of tracklet X_{k_1} and the first blob of X_{k_2} are less than thresholds, $X_{k_1} \rightarrow X_{k_2}$ is a candidate of a tracklet translation. New entries for C and $\boldsymbol{\rho}$ are defined as:

$$\begin{aligned} C(h,i) &= \begin{cases} 1, \text{if } i = k_1 \text{ or } i = N_{\mathbf{X}} + k_2 \\ 0, \text{otherwise.} \end{cases} \\ \boldsymbol{\rho}(h) &= \log P_{link}(X_{k_2}|X_{k_1}) \\ &+ 0.5 \log P_{TP}(X_{k_1}) + 0.5 \log P_{TP}(X_{k_2}) \end{aligned}$$

4. Dividing hypothesis:
   If the last blob of tracklet X_p is near a birth event detected by the mitosis detection step, the tracklet is a candidate of the parent tracklet, and if the first blobs of some other tracklets X_{c1}, X_{c2} are near the candidate parent tracklet, these tracklets are candidates of the children tracklets, New entries for C and $\boldsymbol{\rho}$ are defined as:

$$\begin{aligned} C(h,i) &= \begin{cases} 1, \text{ if } i = p \text{ or } i = N_{\mathbf{X}} + c_1, \\ \qquad \text{or } i = N_{\mathbf{X}} + c_2 \\ 0, \text{ otherwise.} \end{cases} \\ \boldsymbol{\rho}(h) &= \log P_{div}(X_{c_1}, X_{c_2}|X_p) + 0.5 \log P_{TP}(X_p) \\ &+ 0.5 \log P_{TP}(X_{c_1}) + 0.5 \log P_{TP}(X_{c_2}) \end{aligned}$$

5. False positive hypothesis:
   All of the tracklets can be false positive. When X_k is a candidate of a false positive on hypothesis h, New entries for C and $\boldsymbol{\rho}$ are defined as:

$$\begin{aligned} C(h,i) &= \begin{cases} 1, \text{ if } i = k \text{ or } i = N_{\mathbf{X}} + k \\ 0, \text{ otherwise.} \end{cases} \\ \boldsymbol{\rho}(h) &= \log P_{FP}(X_k) \end{aligned}$$

A true positive tracklet appears in two and only two associations in the optimal solution: the first blob of the tracklet appears in an initialization, translation or dividing hypothesis, and the last blob of the tracklet appears in a translation, dividing or termination hypothesis. Thus, $\log P_{TP}(X_k)$ in the second term of Eq. 3.8 is divided into two halves that are integrated into the two neighboring transition hypotheses respectively, as described in hypotheses 1-4.

After making M hypotheses over $N_{\mathbf{X}}$ tracklets, the MAP problem in Eq. 3.8 can be considered as selecting a subset of rows of C such that the sum of corresponding elements in $\boldsymbol{\rho}$ is maximized, under the constraint where any trees can not overlap with each other. This can be formulated as the following binary optimization problem:

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}} \boldsymbol{\rho}^T \boldsymbol{x}, \quad s.t. \ C^T \boldsymbol{x} = \mathbf{1} \tag{3.9}$$

where $\boldsymbol{x}$ is a $M \times 1$ binary vector, and $x_k = 1$ means the k-th hypothesis is selected in the global optimal solution. The term $\mathbf{1}$ is a $M \times 1$ vector in which all elements are 1. The constraint $C^T \boldsymbol{x} = \mathbf{1}$ guarantees that each tracklet ID appears in only one associated tree or false positive tracklet. Figure 3.4 shows a simple example of the linear programming where the number of tracklets is 7 and the number of hypotheses is 18. In the optimal solution, initial tracklet 1 is linked to tracklet 3 (i.e., tracklets 1, 3 are associated as a single edge tree). Initial tracklet 4 divides into 5 and 6, tracklet 5 is linked to tracklet 7, and tracklets 6 and 7 are termination tracklets (i.e., tracklets 4, 5, 6 and 7 are associated as a binary tree).

3.2.5 Implementation details

In this section, I describe the estimation of the probabilities in my method. Let α be the miss detection rate of the cell detector, and $|X_i|$ be the number of total detection responses in a tracklet X_i. The probabilities of false positive and true positive are defined as:

$$P_{FP}(X_i) = \alpha^{|X_i|} \tag{3.10}$$

$$P_{TP}(X_i) = 1 - P_{FP}(X_i) \tag{3.11}$$

The initialization probability is defined based on the time distance between the beginning of the sequence and the first appearance frame of the tracklet, or the spatial distance between the boundary of the field of view and the cell centroid for the cell entering case.

$$P_{ini}(X_i) = \begin{cases} e^{-\frac{dt_0(R_{i_0})}{\lambda_1}}, & \text{if } dt_0(R_{i_0}) < \theta_t \\ e^{-\frac{ds(R_{i_0})}{\lambda_2}}, & \text{if } ds(R_{i_0}) < \theta_s \\ \xi & \text{otherwise } (\xi \text{ is small}) \end{cases} \tag{3.12}$$

where R_{i_0} is the first detection response of tracklet X_i, $dt_0(R_i)$ is a time distance between the first frame of the sequence and the frame when the detection response R_i appears. $ds(R_i)$ is the distance between the centroid of detection response R_i and the image boundary. λ_1 and λ_2 are free parameters to adjust the distribution. If the first detection response of the tracklet appears in both beginning of the sequence and near the boundary, a maximam one is taken for the probability.

The termination probability is defined in a similar way as the initialization probability.

$$P_{term}(X_i) = \begin{cases} e^{-\frac{dt_{end}(R_{i_{end}})}{\lambda_1}}, & \text{if } dt_{end}(R_{i_{end}}) < \theta_t \\ e^{-\frac{ds(R_{i_{end}})}{\lambda_2}}, & \text{if } ds(R_{i_{end}}) < \theta_s \\ \xi, & \text{otherwise } (\xi \text{ is small}) \end{cases} \tag{3.13}$$

where $R_{i_{end}}$ is the last detection response of tracklet X_i, $dt_{end}(R_i)$ is a time distance between the last frame of the sequence and the frame when detection response R_i appears.

The link probability between two tracklets and the dividing probability that one tracklet divides to two tracklets are defined as:

$$P_{link}(X_j|X_i) = e^{-\|g(R_{j_0}) - g(R_{i_{end}})\|/\lambda_3} \tag{3.14}$$

$$P_{div}(X_{c1}, X_{c2}|X_p) = e^{-(\|g(R_{p_{end}}) - g(R_{c1_0})\| + \|g(R_{p_{end}}) - g(R_{c2_0})\|)/2\lambda_3} \tag{3.15}$$

where $g(\cdot)$ computes an object's feature vector in which different types of features can be incorporated such as appearance time and motion history. λ_3 is a free parameter to adjust the distribution. Based on the cell movement history, these parameters are set as: $\lambda_1 = 5$, $\lambda_2 = 30$, $\lambda_3 = 25$, $\theta_t = 15$, $\theta_s = 40$.

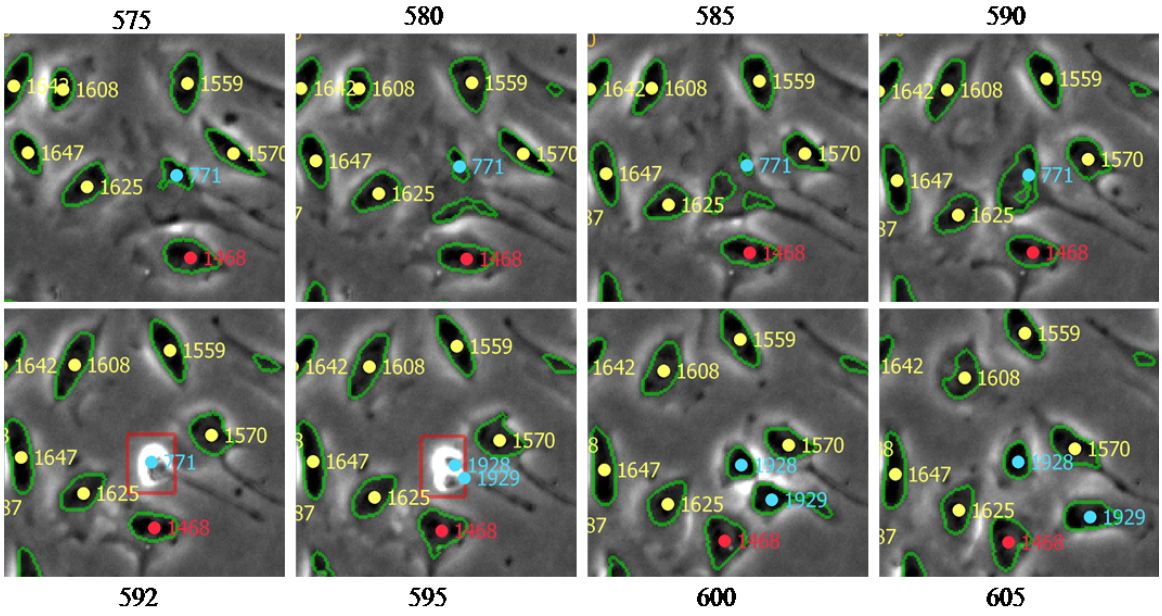


Figure 3.5: Example images of tracking results. Green contours are segmented cell boundaries. Red color boxes are detected mitosis events. The numbers in the images are cell ID. The number on the top and bottom of the images are frame indexes. The colors of cell IDs indicate their family identity. Cells with the same color have the same ancestor.

3.3 Experimental results

3.3.1 Tracking results

Figure 3.5 shows an example sequence of tracking results. The cell 771 on the center of the image spreads out and the boundary is ambiguous, thus, from frame 580 to 585, the cell are segmented to multiple regions some of which are false positives. These false positives disappear in several frames, and only one region can be associated with the tree. Since my global association method uses not only space and appearance information but also temporal information, my method tracks the cells well and recognizes the false positives. Using the detected mitosis event information (red box in Figure 3.5), the proposed method makes a hypothesis of cell division, thus, the two children cells 1928 and 1929 are correctly associated to the parent cell 771.

Figure 3.6(a) shows the tracklets before the global association and Figure 3.6(b) shows the associated tree after the global association. There are 38 tracklets in Figure 3.6(a) including false positives and false negatives. The true positive tracklets are well associated to a tree and false positive tracklets are removed by global data association as shown in Figure 3.6(b).

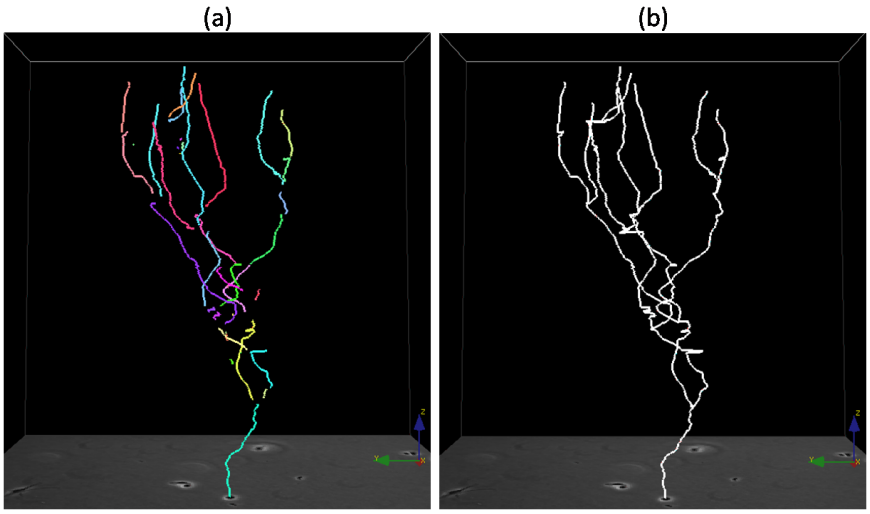


Figure 3.6: Example of space-time trajectories of a cell family. (a) Tracklets. (b) A tree in which tracklets are associated by global data association.

3.3.2 Quantitative evaluation

Data

Five sequences were captured at the resolution of 1040×1392 pixels where C2C12 muscle stem cells growing from 30+ to 600+ are imaged every 5 minutes by ZEISS Axiovert 135TV phase contrast microscope at 5X magnification over 65 hours (780 images). For one image sequence, all cells are annotated. Since it is extremely time-consuming to annotate all cells, for the other four image sequences, three cells are randomly selected in the initial frame and their progeny cells are manually tracked. The total number of annotated cells in the five sequences is $124,621$.

Metrics

I use three quantitative criteria to assess the system performance: track purity, target effectiveness [Blackman86], and mitosis branching correctness.

To compute target effectiveness, each target (human annotated) is first assigned to a track (computer-generated) that contains the most observations from that ground-truth. Then target effectiveness is computed as the number of the assigned track observations over the total number of frames of the target. It indicates how many frames of targets are followed by computer-generated tracks. Similarly, track purity is defined as how well tracks are followed by targets.

The mitotic branching correctness measured the accuracy of mother-daughter relationships between tree branches. The definition of the metric is described in Appendix A. In this evaluation, I set the parameter as $\theta_\epsilon = 10$.

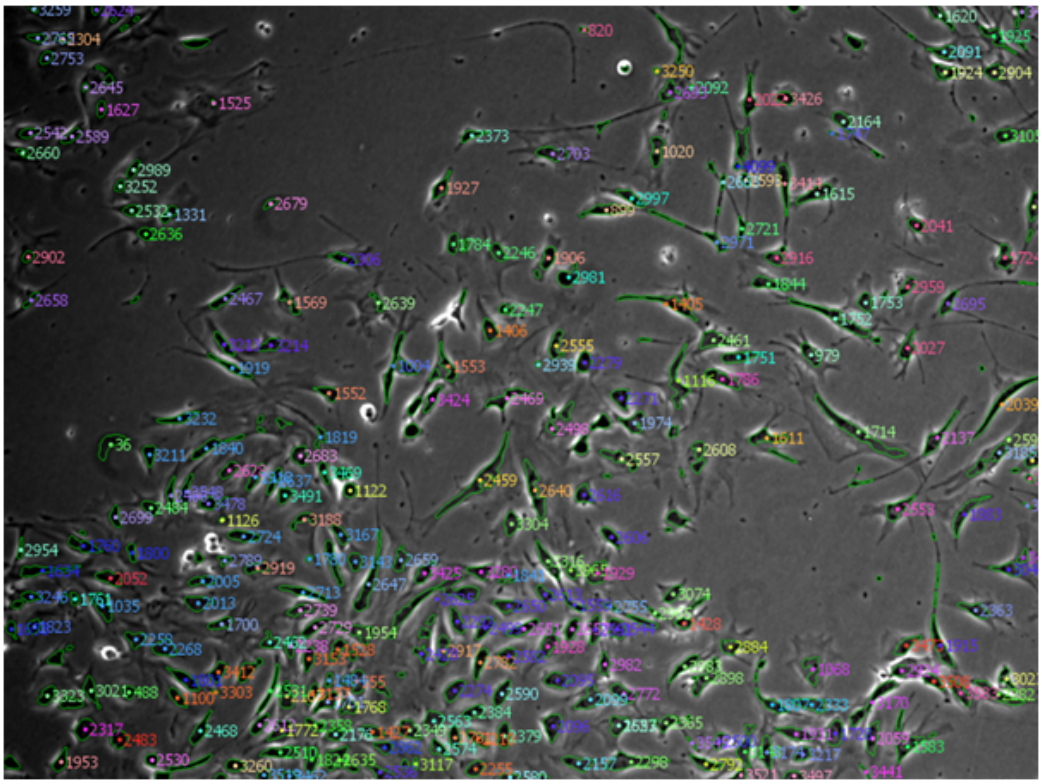


Figure 3.7: Example of tracking result image with track IDs and segmented regions.

Performance evaluation

Figure 3.7 shows an example image of cell tracking with track IDs and segmented regions. Cells are well segmented and tracked in the population. Figure 3.8 shows the space-time trajectory plot of the whole sequence. It represents the complete history of the cell population: motions of all the cells and their lineage information. Figure 3.9 shows examples of the lineage trees compared to human annotated ones. Horizontal red lines indicate tracks that follow the ground-truth, vertical red lines indicate that the mitosis branching is correctly detected on the branch nodes of the lineage tree. The results show that the lineage trees are well constructed.

As shown in Table 3.1, my method achieves higher accuracy on all of the performance metrics than the state-of-the-art method in [Li08] on the full-annotated sequence. Table 3.2 summarizes the target effectiveness and mitotic branching correctness comparison on four image sequences[1]. On average, the proposed method improved 19% on target effectiveness and 27% on mitosis branching correctness compared with [Li08].

[1]I am not able to compute track purity for the four partially-annotated sequences because it needs all cells to be annotated.

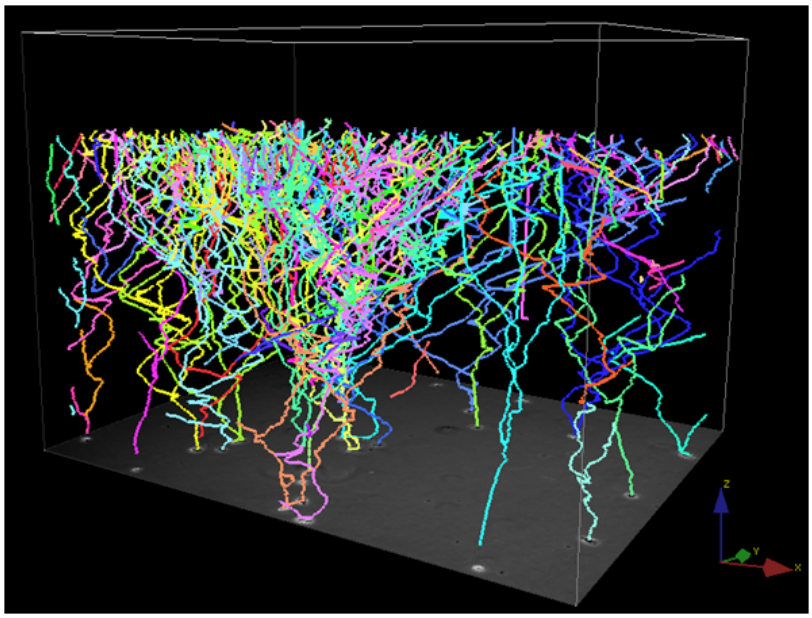


Figure 3.8: Examples of space-time trajectories of the whole sequence. X and Y axes represent 2D space, Z axis represents time.

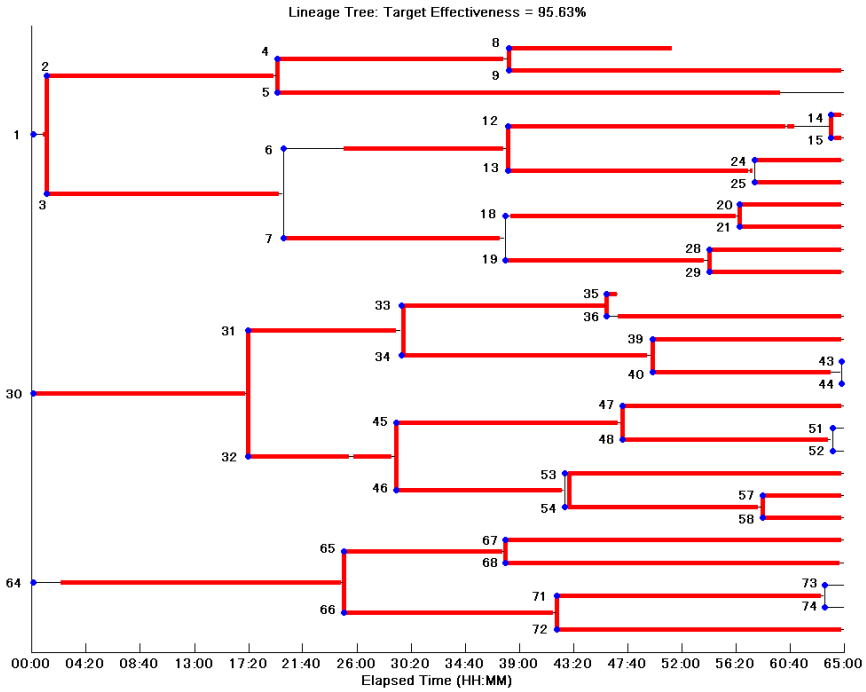


Figure 3.9: Lineage trees and performance evaluation (thin black lines: three human annotated lineage trees; overlaid thick red lines: correctly-tracked cells by the proposed method).

	Track Purity	Target Effectiveness	Mitosis Branching Correctness
Li et al. [Li08]	0.62	0.70	0.46
Proposed	0.81	0.87	0.65

Table 3.1: Comparison of my method with [Li08] on a sequence with all cells annotated.

	Target Effectiveness		Mitosis Branching Correctness	
	Proposed	Li et al.	Proposed	Li et al
exp1	0.96	0.75	0.75	0.25
exp2	0.87	0.7	0.65	0.63
exp3	0.87	0.68	0.59	0.39
exp4	0.78	0.6	0.57	0.2
average	0.87	0.68	0.64	0.37

Table 3.2: Comparison of my method with [Li08] on four sequences.

3.4 Conclusion

I proposed a global data association method for cell tracking problem. The proposed method can associate tracklets to form not only sequential structures but also tree structures. The results of the data association provide the full cell trajectories and lineage trees. Experimental results on a challenging data set show that the proposed method significantly improves the tracking performance including target effectiveness, track purity, mitosis branching correctness by globally associating tracklets.

Chapter 4

Cell Detection from Redundant Candidate Regions under Non-Overlapping Constraints

4.1 Introduction

Automated cell detection in microscopy images is one of the most important tasks in cell behavior analysis in biological research. Robust cell detection and segmentation play a crucial role in developing cell-tracking methods. Even though many cell-detection methods have been proposed, cell detection under high-density conditions still remains a non-trivial task. There are several difficulties in cell detection. First, microscopy images often have inhomogeneous backgrounds and noise, and their contrast is usually very low. In such images, the pixel intensity of a cell is often lower than that of the background. Second, cells often touch other cells and form a cluster with blurry inter-cellular boundaries.

In Chapters 2 and 3, I discussed current cell-detection methods, which apply preconditioning methods [Li09][Yin10a] then segment cell regions by Otsu thresholding [Otsu79]. However, as summarized in the introduction, these cell-detection methods do not simultaneously overcome all of the difficulties in detecting cells under high-density conditions: multiple cells are mistakenly merged, a single cell is segmented into multiple regions, and low-intensity cells are miss detected. Therefore, I propose a cell-detection method which detects cells from redundant candidate regions under non-overlapping constraints to overcome these difficulties. It first detects redundant candidate regions by allowing candidates to overlap; thereby, avoiding miss detections. It avoids over-detection by producing an optimal set of cell regions from the redundant regions under non-overlapping constraints: a selected region looks like a single cell and does not overlap other cells. I formulate this problem of optimal region selection under non-overlapping constraints as a binary linear

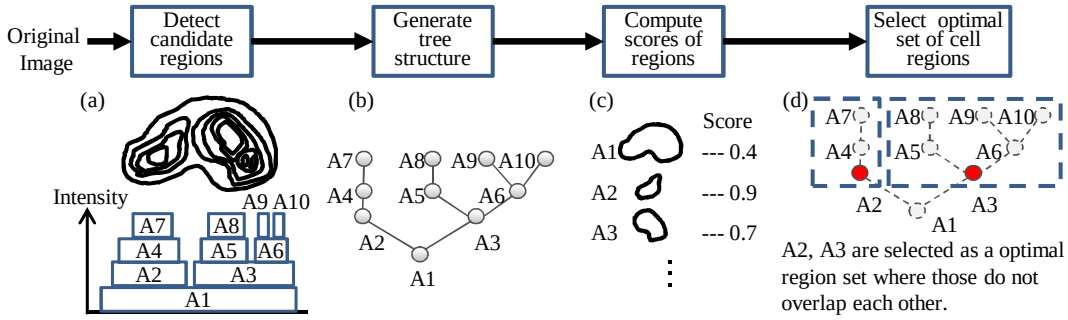


Figure 4.1: Method overview. Bottom images illustrate an example of a case in which two cells touch and form a cluster in each step. (a) Example illustration of detected candidates. (b) Tree consists of candidate nodes, and each node represents a candidate region. (c) Example scores of candidate regions. (d) Optimal solution set in which selected regions do not overlap.

programming problem by using a tree structure that represents the overlapping relationships of candidates.

The proposed cell-detection method was evaluated in comparison with five other cell-detection methods: Otsu thresholding [Otsu79], thresholding with classification [Yin12], thresholding with separating touching cells (CellProfiler) [Carpenter06], level-set [Li10], and finding local maxima with morphological operations (FIJI) [Schindelin12]. Since my aim was to detect individual cells rather than segmenting complex cell shapes, the evaluation was done using recall, precision, and F-measure for cell detection as the metrics. The datasets included 2D microscopy images of several types of cells: stained zebrafish cells (digital scanned light-sheet microscopy images), stained bovine aortic endothelial cells (fluorescence microscopy images), and human CNS stem cells (DIC microscopy images). The proposed method exhibited the best performance, with an F-measure of over 0.9 for all datasets. I also applied the method to 3D data. The results suggest that it also works for the 3D cell-detection problem.

4.2 Cell detection from redundant candidate regions

Figure 4.1 shows an overview of the proposed method. The method consists of four steps,

- *Detect candidate regions:* A set of redundant candidate regions is produced by using a multiple threshold method. The set includes many false positives but is expected in turn to contain very few false negatives (Figure 4.1(a)). This indicates that most true positives are included in the candidate set.

- *Generate tree structure:* A tree structure is generated in which the candidate regions are nodes, and the relationships between nodes are generated on the basis of information about overlapping (Figure 4.1(b)). This tree represents the non-overlapping constraints and is used for weighting the scores in the fourth step.

- *Compute scores of candidate regions:* The scores of the candidate regions are computed on the basis of how much each region looks like a single cell as determined by using cell appearance features (Figure 4.1(c)).

- *Select optimal set of cell regions:* An optimal set of cell regions is produced from the redundant candidate regions under non-overlapping constraints (Figure 4.1(d)). This optimization problem is formulated as a binary linear programming problem by using the generated tree and computed candidate scores. This binary linear programming maximizes the sum of the weighted scores of the selected regions under non-overlapping constraints.

These steps are next described in detail.

4.2.1 Detect set of candidate regions

The goal in this step is to produce a set of candidate regions that may include many false positives but in turn very few false negatives.

My method is based on the fact that cell regions appear bright under fluorescence and preconditioned images and on the fact that the intensities on the inside of a cell are slightly higher than those at its boundaries among touching cells. Candidate regions are identified by segmenting all regions by using multiple-level thresholding. I set K level thresholds $\mathbf{T} = \{T_i | i = 1, ..., K\}$, that are equally spaced, and each threshold is used to segment images at a particular level of intensity. The holes of the foreground region are filled since a cell does not have holes. Finally, the segmentation results are registered as candidate cell regions. A set of candidate cell regions is denoted as $\mathbf{A} = \{A_i, i = 1, 2, ..., N\}$, where A_i represents the i-th candidate cell region, and N is the number of candidate regions. For the example in Figure 4.1, all the segmentation results $\{A1, A2, A3, A4, A5, A6, A7, A8, A9, A10\}$ are registered as candidate regions. These candidate regions include almost all the true positives even though many false positives are included.

This process can also be applied to non-invasive microscopy images by using preconditioning methods developed for phase contrast images [Yin10a] and for DIC images [Li09]. In preconditioned images, a large pixel value indicates the foreground and a small pixel value indicates the background. After preconditioning has been applied, non-invasive microscopy images can be treated the same as fluorescent images.

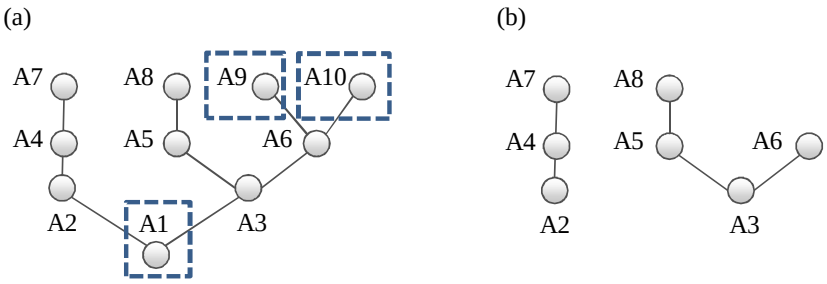


Figure 4.2: Example of pruning process. (a) Original tree. Dotted boxes indicate nodes outside pre-defined size range. (b) Pruned tree.

4.2.2 Generate tree structure

The detected cell candidates constitute a tree structure with the cell candidates as nodes. If candidate region A_i is inside candidate region A_j, A_j is one of the root components of A_i in the tree. The set of all candidate regions under all thresholds is obviously hierarchically ordered by subset inclusion, as shown in Figure 4.1(a). This hierarchical order produces a tree structure, as shown in Figure 4.1(b).

To reduce computational costs, some nodes of the tree are pruned. A set of candidate regions typically includes large and small regions with a size that does not match the expected cell size. In the pruning step, nodes smaller than θ_{min} or larger than θ_{max} are eliminated: θ_{min} and θ_{max} are pre-defined parameters used to remove noise. An example of this pruning step is illustrated in Figure 4.2, in which nodes $A1$, $A9$, and $A10$ are eliminated.

This tree structure is the key to performing the fourth step, effectively finding an optimal set of cells. The tree structure is used to handle non-overlapping constraints and to define the target function of the optimization.

Intensity level tree representation is used for image matching tasks. Mattes *et.al.* [Mattes99] proposed an image matching method that uses a tree representation matching strategy. Matas *et.al.* [Matas02] proposed a maximally stable extremal region (MSER) detection method for finding the correspondences between image elements from two images with different viewpoints. There is a key difference between these methods and my method: my method uses tree representation to formulate the problem of optimal region selection from candidate regions under non-overlapping constraints.

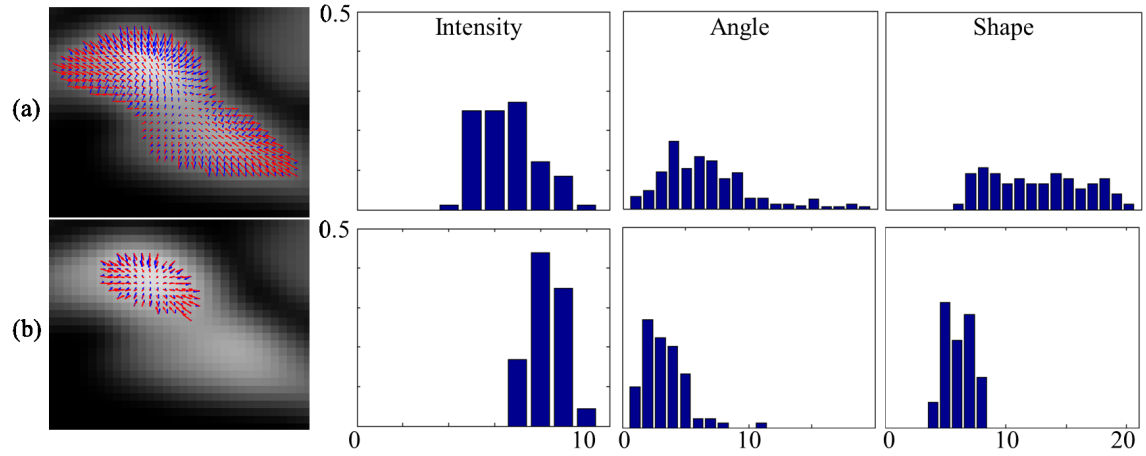


Figure 4.3: Examples of visual features used to compute the score representing resemblance of a candidate region to a single cell. (a) Example of a false candidate in which the region consists of two cells. (b) Example of a true candidate in which the region consists of single cell. Horizontal axes show bin IDs; vertical axes show feature frequencies.

4.2.3 Compute score of each candidate region

A score representing how likely the candidate region contains the main part of a single cell is computed for each cell candidates.

Figure 4.3 shows typical examples of a region consisting of two touching cells (Figure 4.3(a)) and a region consisting of a single cell (Figure 4.3 (b)). As mentioned above, cells usually exhibit higher intensity inside the cell, and the intensity gradually decreases toward the edge since cell boundary is thinner than the center area, and the center area tends to be ellipsoid. This means that appearance-related items are important for judging whether a candidate region consists of a single cell. 1) Intensity distribution: the intensity histogram of a cluster region has lower intensity values, and the variance is larger compared with that of a single cell region, as shown by the intensity charts in Figure 4.3. 2) Distribution of angles between gradient direction and direction toward centroid from point in each pixel: the average and variance of the distribution of angles for a cluster region are usually larger than those for a single cell region since a cluster region has several local peaks, as shown by the angle charts in Figure 4.3. 3) Distribution of centroid distances [Mingqiang08]: the centroid distance is the distance to a boundary point from the cell center, and the centroid distance distribution for a cluster region has a larger variance than for a single cell since a touching cell region often has concavities, as shown by the shape charts in Figure 4.3.

A binary classifier is learned via an SVM algorithm [Cortes95] by using the feature

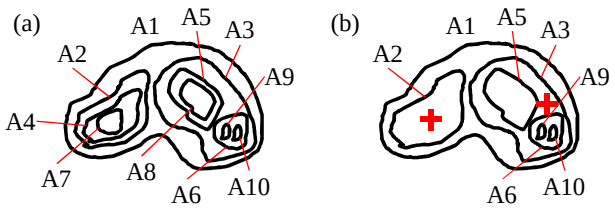
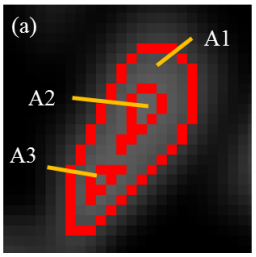


Figure 4.4: (a) Example of a set of candidate regions. (b) Pruned candidate regions and clicked positions. 'red plus' indicates a clicked position for annotation.

vectors and training data. In the inference step, the class membership probabilities are computed as the scores of regions that look like cell regions via the trained SVM [Drish01]. The computed score of a candidate region A_i is represented as $P(A_i)$, where $P(A_i)$ is normalized in the 0 to 1 range.

To obtain training data, an annotator selects only true positives from the candidate regions rather than annotating cell region boundaries since the aim here is to detect individual cell regions rather than to extract complex cell shapes. Moreover, annotating the training data in this way reduces the amount of time and effort. A candidate region set and a corresponding tree structure are generated by using multiple-level thresholding, as described in section II.A and B. To reduce the number of candidates, if a node has only one child, the child node is pruned, as shown in Figure 4.4. The annotator is shown the original image and the corresponding pruned candidate regions. The annotator then selects the true positive regions from the displayed regions. A true positive region is one of cell soma. Any desired cell, for instance A2 and A3 in Figure 4.4, can be selected by clicking any position inside the cell except one included in any of the cell's child regions as shown in Figure 4.4 (b). After all cells in an image are annotated, the selected regions are labeled as positive samples, and the non-selected ones are labeled as negative samples. For example in Figure 4.4, $\{A2, A3\}$ are positive samples, and $\{A1, A5, A6, A9, A10\}$ are negative samples. A pre-pruned tree is used to extend the training data, as shown in Figure 4.4 (a). If a region is a descendant of a positive sample and the ratio of its size to that of the positive region exceeds a threshold, it is also labeled as a positive sample. Descendant regions of a negative sample are also labeled as negative samples. Here, I set the threshold ratio to 0.8. For example in Figure 4.4, if the ratio of the size of $A4$ to that of $A2$ exceeds the threshold and the ratio of the size of $A7$ to that of $A2$ does not exceed the threshold, $A4$ is labeled a positive sample, and $A7$ is labeled a negative sample. $A8$ is also labeled a negative sample since $A8$ is a child of negative sample $A5$.



(b) Scores

ID	Score
A1	0.8
A2	0.5
A3	0.4

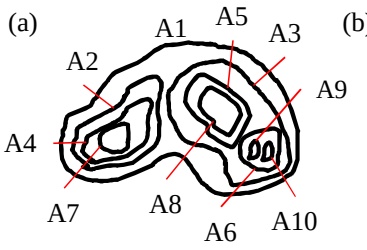
Figure 4.5: (a) Simple example of set of candidate regions. (b) Scores for three candidate.

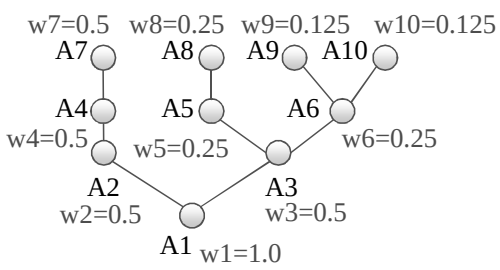
4.2.4 Select optimal set of cell regions

An optimal set of regions is then selected from candidate regions under non-overlapping constraints (i.e., the selected regions do not overlap). This is a problem of maximizing the sum of the selected regions' scores. However, if the scores computed in the previous step were simply used, the leaf regions would tend to be selected since the larger the number of selected regions, the higher the sum. For example, consider a simple case in which root region A_1 with $P(A_1) = 0.8$ has two children A_2 and A_3 with $P(A_2) = 0.5$, and $P(A_3) = 0.4$, where A_2 and A_3 are inside A_1, as shown in Figure 4.5. There are two hypothetical solutions including $\{A_1\}$ and $\{A_2, A_3\}$ that satisfy the non-overlapping constraints. In this case, even though the score of A_1 is higher than those of both A_2 and A_3, A_2 and A_3 are selected since $P(A_2) + P(A_3) > P(A_1)$, which maximizes the sum of the scores of the selected regions. For fair comparison, I compared the score of root region $P(A_1)$ and the average scores of the leaf regions $\frac{1}{2}(P(A_2) + P(A_3))$. To extend this idea to all nodes in the tree, each node is weighted, where the weight of the root region is 1 and the weight of the next layer is equally divided by the number of edges. This weighting process is iteratively applied until the leaf of the tree resembles a flow of water. The weight w_l of A_l is

$$w_l = \prod_{A_m \in \Psi(A_l)} \frac{1}{B(A_m)} \tag{4.1}$$

where $\Psi(A_l)$ is the set of ancestor nodes of A_l in the generated tree, and $B(A_m)$ is the number of brothers of A_m in the tree where the number of brothers includes A_m. For example, consider the more complex case shown in Figure 4.6. In this example, $\Psi(A_8)$ is $\{A_8, A_5, A_3, A_1\}$, and $B(A_5)$ is 2 since its brothers are A_5 and A_6. Here, w_8, which is the weight for region candidate A_8, can be computed as $w_8 = \frac{1}{B(A_1)} \times \frac{1}{B(A_3)} \times \frac{1}{B(A_5)} \times \frac{1}{B(A_8)} = \frac{1}{1} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{1} = \frac{1}{4}$. From this process, the weight of a parent node equals the sum of the weights of the child nodes in each layer. In Figure 4.6, for example, the weight of A_3 ($w_3 = 0.5$) equals the sum of the weights of A_5 ($w_5 = 0.25$) and A_6 ($w_6 = 0.25$). This means that the score of each node can be compared with the average of the scores of its children





(c)

ID	P	w_l	ρ
A1	0.4	1	0.4
A2	0.9	0.5	0.45
A3	0.7	0.5	0.35
A4	0.7	0.5	0.35
A5	0.5	0.25	0.125
A6	0.4	0.25	0.09
A7	0.6	0.5	0.3
A8	0.6	0.25	0.15
A9	0.2	0.125	0.025
A10	0.2	0.125	0.025

Score vector ρ

(d)

ID	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	x
A1	1	1	1	1	1	1	1	1	1	1	0 or 1
A2	0	1	0	1	0	0	1	0	0	0	0 or 1
A3	0	0	1	0	1	1	0	1	1	1	0 or 1
A4	0	0	0	1	0	0	1	0	0	0	0 or 1
A5	0	0	0	0	1	0	0	1	0	0	0 or 1
A6	0	0	0	0	0	1	0	0	1	1	0 or 1
A7	0	0	0	0	0	0	1	0	0	0	0 or 1
A8	0	0	0	0	0	0	0	1	0	0	0 or 1
A9	0	0	0	0	0	0	0	0	1	0	0 or 1
A10	0	0	0	0	0	0	0	0	0	1	0 or 1

Constraint matrix C

Figure 4.6: (a) Example set of candidate regions. (b) Tree consisting of all candidates. (c) Example scores, weight, and weighted score vector for each candidate. (d) Constraint matrix of candidate set.

nodes by multiplying each score by the appropriate weight.

I formalize this optimization task, i.e., maximizing the sum of the weighted scores of the selected regions under non-overlapping constraints, as a binary linear programming problem [Wolsey98]. Let N be the number of candidate regions; an $N \times 1$ vector $\boldsymbol{\rho}$ stores the weighted score for every candidate region, and an $N \times N$ matrix C stores the constraints needed to avoid overlapping regions, where each row and column index represents the index of a candidate region. Let l be the index of a new hypothesis in which I append a new row to C and a corresponding weighted score to $\boldsymbol{\rho}$:

$$C(l,i) = \begin{cases} 1, & \text{if } i = m_r, m_r \in \Omega(A_l) \\ 0, & \text{otherwise.} \end{cases} \tag{4.2}$$

$$\boldsymbol{\rho}(l) = w_l P(A_l) \tag{4.3}$$

where $\Omega(A_l)$ is a set of candidate region indexes that are all descendant node indexes of A_l in the tree. The set includes A_l. For the example in Figure 4.6(b), $\Omega(A_3)$ is $\{3,5,6,8,9,10\}$. After score vector $\boldsymbol{\rho}$ and constraint matrix C are generated for all N candidate regions, the optimization problem can be formulated as a binary linear programming problem:

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}} \boldsymbol{\rho}^T \boldsymbol{x}, \quad s.t.\ C^T \boldsymbol{x} \leq \mathbf{1}, \tag{4.4}$$

where $\boldsymbol{x}$ is an $N \times 1$ binary vector and $x_k = 1$ means the k-th candidate region is selected in the optimal solution. To solve this problem, I use the branch-and-bound algorithm [Ross75], which is based on the simple concept of "divide and conquer". The algorithm works like a tree search: the original problem is divided into all possible sub-problems, and each sub-problem is solved. A binary linear programming problem is solved by solving a series of LP-relaxation problems in each sub-problem. The binary integer requirement on the variables is replaced with a weaker constraint, $\mathbf{0} < \boldsymbol{x} < \mathbf{1}$. The constraint $C^T\boldsymbol{x} \leq \mathbf{1}$ guarantees that overlapping regions are not included in the solution. In the example in Figure 4.6, the set of overlapping candidates with A_2 is $\{1, 2, 4, 7\}$. This constraint means that candidates $\{1, 4, 7\}$ are not included in the optimal solution if A_2 is selected as a cell region.

Linear programming (LP) is used in cell tracking [Kofahi06][Bise11] and segmentation methods [Keuper11][Wood13]. Al-Kofahi *et.al.* [Kofahi06] use LP to associate cells between successive frames for cell tracking. Keuper *et.al.* [Keuper11] use LP to solve Markov random field problems for cell and subcellular segmentation. The proposed method is the first to use LP to find an optimal set of cell regions from candidate regions.

4.3 Experiments with 2D images

I evaluated the 2D image performance of the proposed method and compared it with those of several current methods.

4.3.1 Data-sets

For evaluation, I prepared three data-sets for two types of microscopic images (fluorescent microscopy and DIC microscopy) and several types of cells. Three images were used for learning, and 20 were used for performance testing for each data-set.

- Data-set A: Stained zebrafish cells in fluorescent microscopy images.

  Zebrafish cells in vivo were stained using enhanced green fluorescent protein (EGFP), which caused the nuclei to exhibit bright green fluorescence. A z-slice of the stained cells in vivo was captured at a resolution of 360 × 290 pixels per image with a digital scanned light-sheet microscope (DSLM). The number of cells was 100 to 150 per image, and the total number of cells was 2306. The long radius of the cell is ranged from 7 to 20 pixels. An example image in dataset A is shown in Figure 4.9(a). The contrast is low, and the background is inhomogeneous, so the cell boundaries are unclear (i.e., some adjacent cells have blurry boundaries).

- Data-set B : Stained BAEC in fluorescent microscopy images.

Bovine aortic endothelial cell (BAEC) populations were stained using green fluorescent protein (GFP), which caused the entire cell body, including the membranes and nuclei, to exhibit bright green fluorescence. The stained cells in vitro were captured at a resolution of 512 × 512 pixels per image using a 20× objective with a fluorescent microscope. The number of cells was 40 to 50 per image, and the total number of cells was 968. The long radius of the cell is ranged from 10 to 25 pixels. An example image is shown in Figure 4.9(a). The contrast is very low, and the background is inhomogeneous, so the cell boundaries of cells are unclear, and the intensity of the background is sometimes higher than that in the cell regions.

- Data-set C : Human CNS stem cell in DIC microscopy images.

  Human central nervous system (CNS) stem cell populations were captured at a resolution of 512 × 512 pixels per image using a 40× objective with DIC optics. The number of cells was 120 to 150 per image, and the total number of cells was 2960. The long radius of the cell is ranged from 8 to 20 pixels. These images were provided by Hoeppner *et al.* [Ravin08]. Preconditioned images prepared by Li and Kanade [Li09] were used before detection instead of directly using DIC images. An example original image is shown in Figure 4.10(a), and the image after preconditioning is shown in Figure 4.10(b). There are blurred cell boundaries and multiple touching cells, as indicated by the red dashed-line box in Figure 4.10(b).

4.3.2 Metrics

I evaluated detection performance by using the centroid points of cells. To create a ground truth, I roughly annotated the cell centers by hand. Since the annotation was done in a subjective manner, the annotated positions were not exactly the center positions of the cells. For data-set A, the cell-detection task was ambiguous even for human annotators as evidenced by differences in results between annotators. To mitigate this problem, I used an evaluation method [Milan13] that uses several annotation sets and averages the performance.

I used precision ($\frac{TP}{TP+FP}$), recall ($\frac{TP}{TP+FN}$), and F-measure ($2 \cdot \frac{precision \cdot recall}{precision+recall}$) as the metrics, where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives. Each manually detected cell was assigned to an automatic detection result by solving linear programming problem in which the sum of the distances between the detection results and the ground-truth were minimized so that multiple detection results were not assigned to one ground truth. After the detection results were assigned to ground-truth, if the distance between the ground truth and corresponding detection result was less than the threshold, that result was assigned. Assigned detection

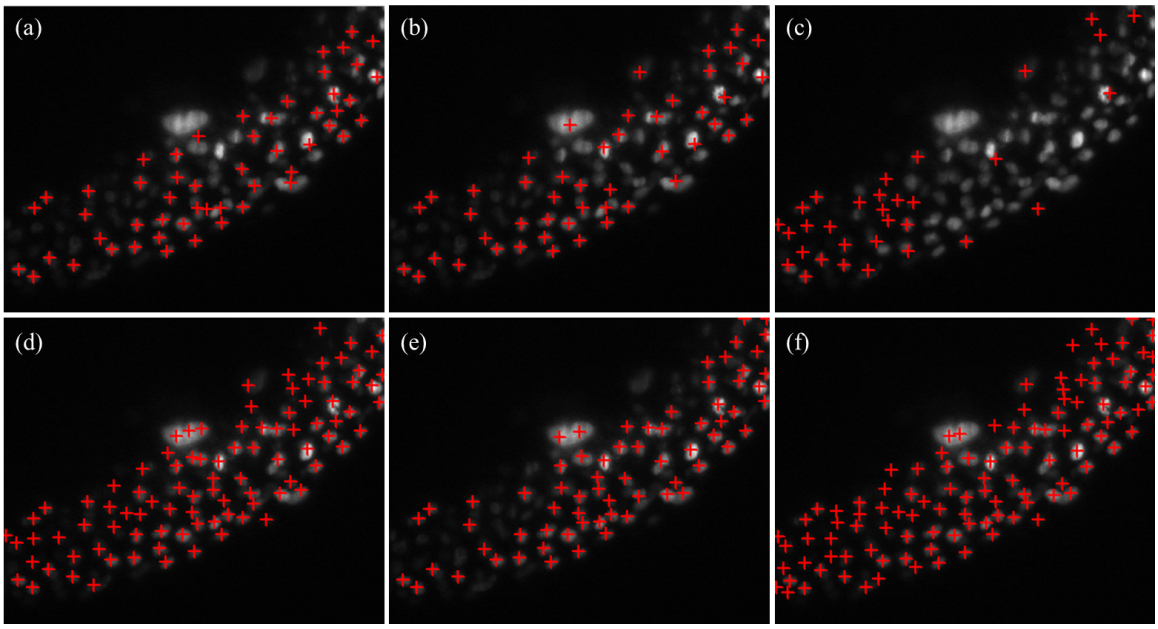


Figure 4.7: Examples of cell-detection results for data-set A. (a) Otsu thresholding [Otsu79], (b) thresholding with classification [Yin12], (c) CellProfiler [Carpenter06], (d) level-set [Li10], (e) FIJI [Schindelin12], and (f) proposed method.

results were counted as TP, non-assigned detection results were counted as FP, and non-assigned ground truths were counted as FN. I set the threshold to 10 pixels given that the cell diameter was 15 to 40 pixels.

4.3.3 Evaluation

I evaluated the performance of the proposed cell-detection method in comparison with those of five other methods, as following. The Otsu thresholding method [Otsu79] with noise reduction removes noise on the basis of the cell size parameter ($\theta_{min} = 30$). This thresholding method usually detects many small false positives. The cell size parameter was used to remove these small regions. The cell candidate detection with classification first detects candidate regions and then classifies the candidate regions as either cells or non-cells using a trained SVM classifier [Yin12]. CellProfiler [Carpenter06] which first segments the foreground and background and then separates the touching cells by using the intensity peaks and shapes. The level-set method proposed by Li *et al.* [Li10] segments cell regions by maximizing the energy function. The FIJI [Schindelin12] detection method first computes the difference of Gaussian (DoG) and then detects intensity maxima that satisfy pre-defined conditions as cells.

For implementation, I set $\theta_{min} = 30$ and $\theta_{max} = 2000$ for all datasets. These pa-

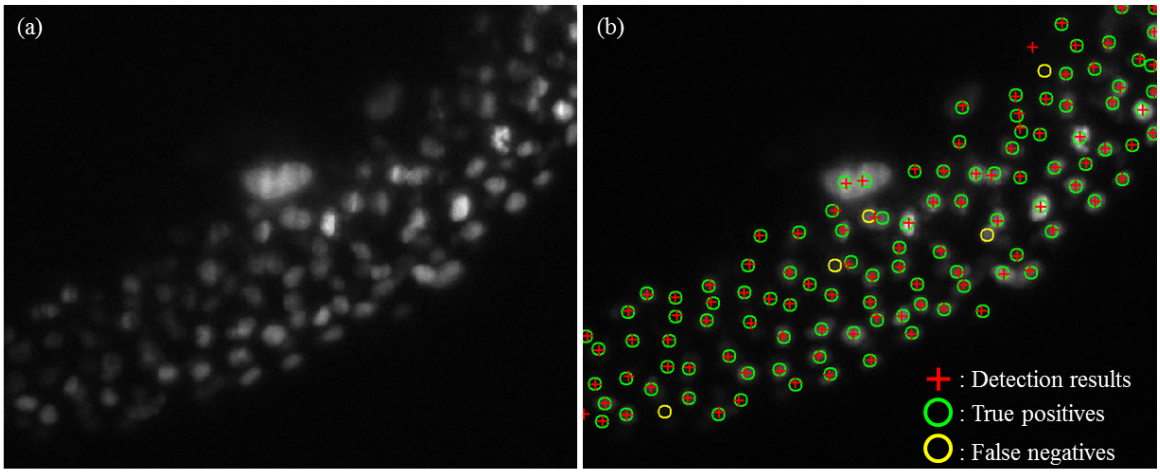


Figure 4.8: Evaluation of cell-detection performance. (a) input image in data-set A using fluorescent microscope. (image contrast manually adjusted for visualization purposes), (b) input image with detected cells compared with ground truth.

Method	Recall	Precision	F-measure
Otsu thresholding [Otsu79]	0.533	0.981	0.69S0
Thresholding with classification [Yin12]	0.481	**0.998**	0.648
CellProfiler [Carpenter06]	0.688	0.990	0.811
Level set [Li10]	0.212	0.931	0.338
FIJI [Schindelin12]	0.8227	0.9673	0.8884
Proposed	**0.938**	0.971	**0.954**

Table 4.1: Performance comparison for dataset A

rameters are not sensitive since they are used for reducing computational costs. For Otsu thresholding [Otsu79], cell candidate detection with classification [Yin12], level-set [Li10], and the proposed method, all the codes were written in MATLAB. The SVM classifier was implemented using LIVSVM [Chan11]. For level-set [Li10], the provided code [Li10] from MATLAB Central was used. For local maxima detection in DoG map [Schindelin12], DoG and the local maxima module in FIJI were used. For foreground segmentation and separation of touching cells [Carpenter06], the IdentifyPrimaryObjects module in CellProfiler was used. The parameters for each method were adjusted by using the training data for each dataset.

Figure 4.7 shows examples of the cell-detection results for each method for dataset A. Table 4.1 summarizes detection performance in terms of recall, precision, and F-measure for dataset A. The best performance is shown in bold type for each metric in the table. With

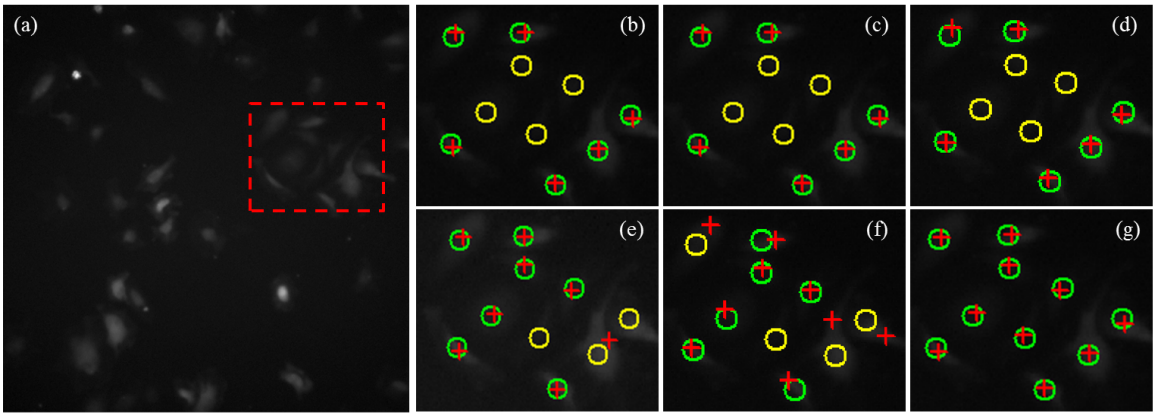


Figure 4.9: Evaluation of cell-detection performance: (a) input image in dataset B using fluorescent microscope (image contrast manually adjusted for visualization purposes) and results from (b) Otsu thresholding [Otsu79], (c) thresholding with classification [Yin12], (d) CellProfiler [Carpenter06], (e) level-set method [Li10], (f) FIJI [Schindelin12], and (g) proposed method.

Method	Recall	Precision	F-measure
Otsu thresholding [Otsu79]	0.699	0.924	0.795
Thresholding with classification [Yin12]	0.613	**0.985**	0.754
CellProfiler [Carpenter06]	0.826	0.948	0.882
Level-set [Li10]	0.743	0.921	0.822
FIJI [Schindelin12]	0.7467	0.7882	0.7656
Proposed	**0.937**	0.908	**0.920**

Table 4.2: Performance comparison for dataset B

Otsu thresholding (Figure 4.7(a)), higher intensity regions were detected and touching cells were detected as a single blob. Therefore, the performance, recall in particular, was not good. Even though the threshold was manually adjusted, a single threshold does not result in good segmentation for such low contrast and closely located cells. With thresholding with classification [Yin12] (Figure 4.8(b)), blobs consisting of touching cells were removed as noise by the classifier. Thus, precision was better, but recall was worse. Therefore, the F-measure was lower than with Otsu thresholding. This method works well if the set of candidate regions mostly includes true positives and false positives, i.e., mis-detected non-cell regions, but performs poorly if an image has low contrast and the cells are located closely together. With CellProfiler (Figure 4.8(c)), higher intensity regions were detected, and touching cells were correctly detected since the method first applies a thresholding

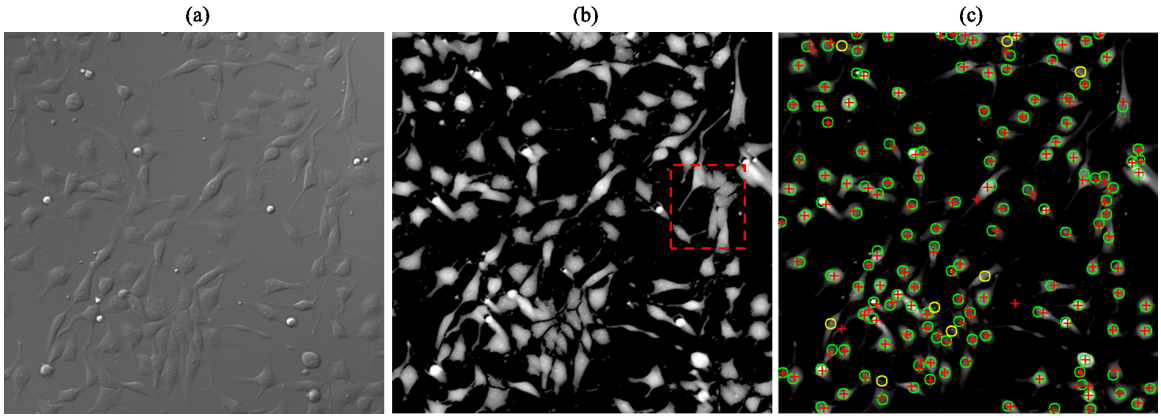


Figure 4.10: Evaluation of cell-detection performance: (a) input image in data-set C using DIC microscope, (b) input image after preconditioning, and (c) input image with detected cells compared with ground truth.

method and then separates touching cells. Therefore, the recall and F-measure were better than with Otsu thresholding. However, low intensity regions were not detected. With the level-set method [Li10] (Figure 4.8(d)), blobs with lower intensity were detected compared with Otsu thresholding and thresholding with classification. However, many closely located cells were detected as a large cluster. A disadvantage of this method is that closely located cells are often detected as a blob since the energy function is locally optimized. Therefore, the performance of this method was the worst. With FIJI (Figure 4.8(e)), many cells were correctly detected. Therefore, the F-measure was better than for the other methods. Local maxima often include false positives since contrast enhancement usually not only enhances the contrast between cells and background but also the noise. Thus, the method uses a pre-defined condition, such as intensity range. If the conditions between the training data and test data differ, performance is degraded. Thus, the F-measure with FIJI was less than that with the proposed method. With the proposed method (Figure 4.8(f)), many cells were correctly detected even though the intensities were low, and closely located cells were separated. Therefore, the method achieved the best F-measure in the comparison. Figure 4.8(b) shows a comparison of the detection results between the proposed method and manual annotation. The red pluses indicate the results with the proposed method, the green circles indicate true positives, and the yellow circles indicate false negatives. The results show that the method worked well when the image had low contrast and the cells were closely located though several deeply touched cells were mis-detected.

Figure 4.9 and Table 4.2 show the detection results for dataset B, for which the cell density was not high but the contrast was very low. With Otsu thresholding, thresholding with classification and CellProfiler (Figures 4.9(b),(c),(d)), low intensity cells were not de-

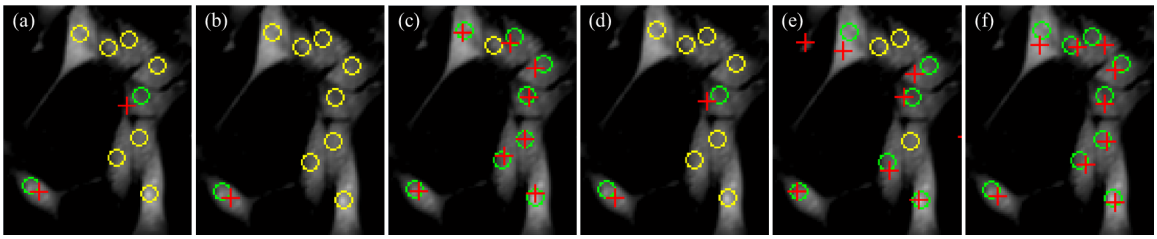


Figure 4.11: Enlarged images of area indicated by dashed-line box in Figure 4.10(b). Results from (a) Otsu thresholding [Otsu79], (b) thresholding with classification [Yin12], (c) CellProfiler [Carpenter06], (d) level-set method [Li10], (e) FIJI [Schindelin12] and (f) proposed method.

Method	Recall	Precision	F-measure
Otsu thresholding [Otsu79]	0.783	0.909	0.840
Thresholding with classification [Yin12]	0.732	**0.966**	0.832
CellProfiler [Carpenter06]	0.830	0.894	0.861
Level-set [Li10]	0.756	0.875	0.810
FIJI [Schindelin12]	0.8723	0.7675	0.8158
Proposed	**0.926**	0.886	**0.905**

Table 4.3: Performance comparison for dataset C

tected. With level-set method (Figure 4.9(e)), some cells were detected as a cluster. This method worked better than thresholding methods since the cell density was not high. With FIJI (Figure 4.9(f)), the detected local maxima were misaligned from the ground-truth cell positions. Therefore, the F-measure was less with dataset A. With the proposed method (Figure 4.9(f)), the cells were correctly detected. The F-measure with the proposed method was the best.

To confirm that the proposed method can be applied to non-invasive microscopy images, I used DIC images in dataset C. Before each method was applied, DIC preconditioning [Li09] was applied to facilitate segmentation. Figure 4.10 shows the detection results for an example image in which multiple cells touched. Table 4.3 summarizes the detection performances for dataset C. With Otsu thresholding (Figure 4.11(a)), touching cells were detected as one cell. With thresholding with classification (Figure 4.11(b)), a cluster was classified as noise since the shape of the cluster did not look like a single cell. With CellProfiler (Figure 4.11(c)), most touching cells were correctly separated after thresholding. With level-set (Figure 4.11(d)), touching cells were also detected as one cell. With FIJI (Figure 4.11(e)), several touching cells were not detected, and a small false positive was detected. With the proposed method (Figure 4.11(f)), the cells were correctly detected. The F-measure with

Method	Recall	Precision	F-measure
Otsu thresholding [Otsu79]	0.672	0.938	0.775
Thresholding with classification [Yin12]	0.609	**0.983**	0.745
CellProfiler [Carpenter06]	0.781	0.944	0.851
Level set [Li10]	0.570	0.909	0.657
FIJI [Schindelin12]	0.8139	0.8410	0.8233
Proposed	**0.9320**	0.9216	**0.9262**

Table 4.4: Average performance for all data set.

Method	Recall	Precision	F-measure
Otsu thresholding [Otsu79]	0.447	**0.993**	0.616
Proposed	**0.906**	0.966	**0.935**

Table 4.5: Performance comparison for 3D volume data

the proposed method was the best.

Table 4.4 summarizes the average performance for all datasets. Overall, the proposed method improved recall without degrading precision. Therefore, the recalls and the F-measures were the best for all datasets compared with those of the other methods.

4.4 Experiments with 3D images

I applied the proposed detection method to z-stack images to determine whether the proposed method can detect cells in 3D as well. Cells in the embryo of zebrafish were captured using DSLM. The image size was 300 $\times$ 300 pixels (0.645 μm/pixel), and the number of z-slices was 74 (2 μm/pixel). An example of a projection image of the 3D volume is shown in Figure 4.12(a), and examples of the z-slices are shown in Figures 4.12 (b), (c), and (d). To apply the proposed method, I treated the z-stack images as 3D volume data (i.e., one pixel in a z-stack image was treated as a voxel in 3D).

The proposed method was easily extended to detect cells in 3D. In 3D volume data, a threshold also makes a region which is a connected blob as a candidate region. When the threshold is slightly increased, the new candidate region must be the same size or smaller than the candidate region generated by the lower threshold, and the new candidate must be inside the previous candidate region. This means that a tree is also generated in 3D data. In addition, the features for computing scores including the intensity distribution, the distribution of angles between gradient direction and the direction toward the centroid from a point in each pixel, and the distribution of centroid distance were also computed for

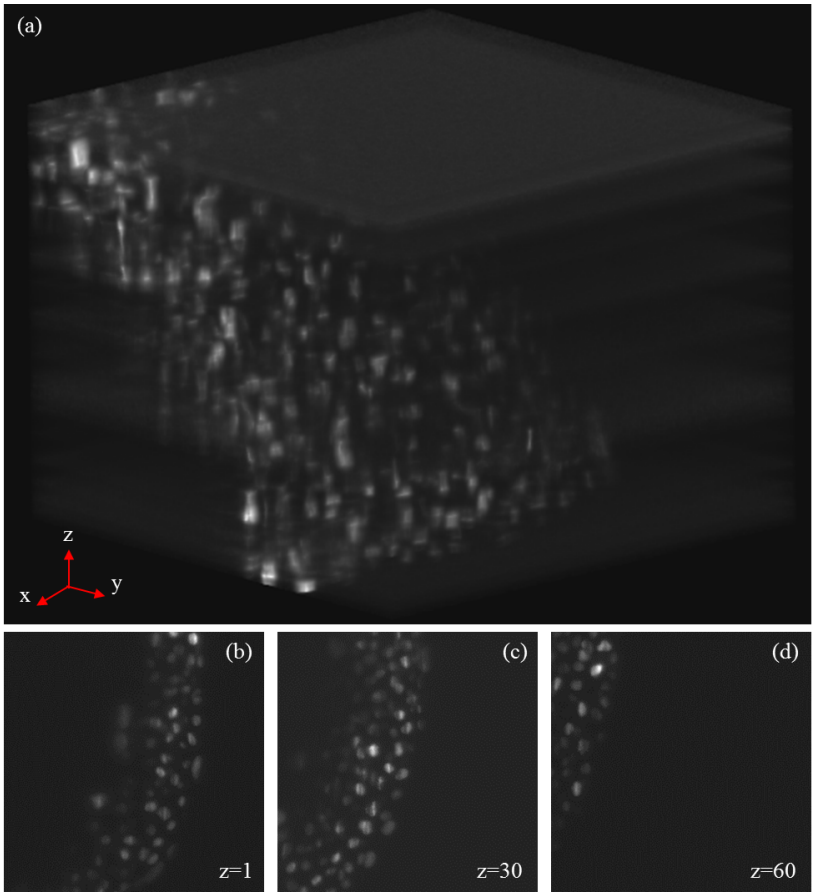


Figure 4.12: Examples of z-stacks for 3D data. (a) Projection image. (b),(c), and (d) Examples of z-slice images. Image contrast was manually adjusted for visualization.

3D regions. The proposed method is thus applicable in 3D data.

The segmentation results for the proposed method are shown in Figures 4.13(a), (b), and (c). For comparison, the results for Otsu thresholding [Otsu79] are shown in Figures 4.13(d), (e) and (f). Figures 4.13(b),(e) and (c),(f) respectively show examples of the enlarged image results from the same viewpoint. As shown in Figures 4.13(b) and (e), the proposed method detected individual cells while the Otsu method detected several closely located cells as a cluster. As shown in Figures 4.13(c) and (f), the proposed method detected low-intensity cells that were not detected with the Otsu method. Table 4.5 summarizes the evaluation results. The proposed method was significantly better in terms of recall and F-measure for 3D data.

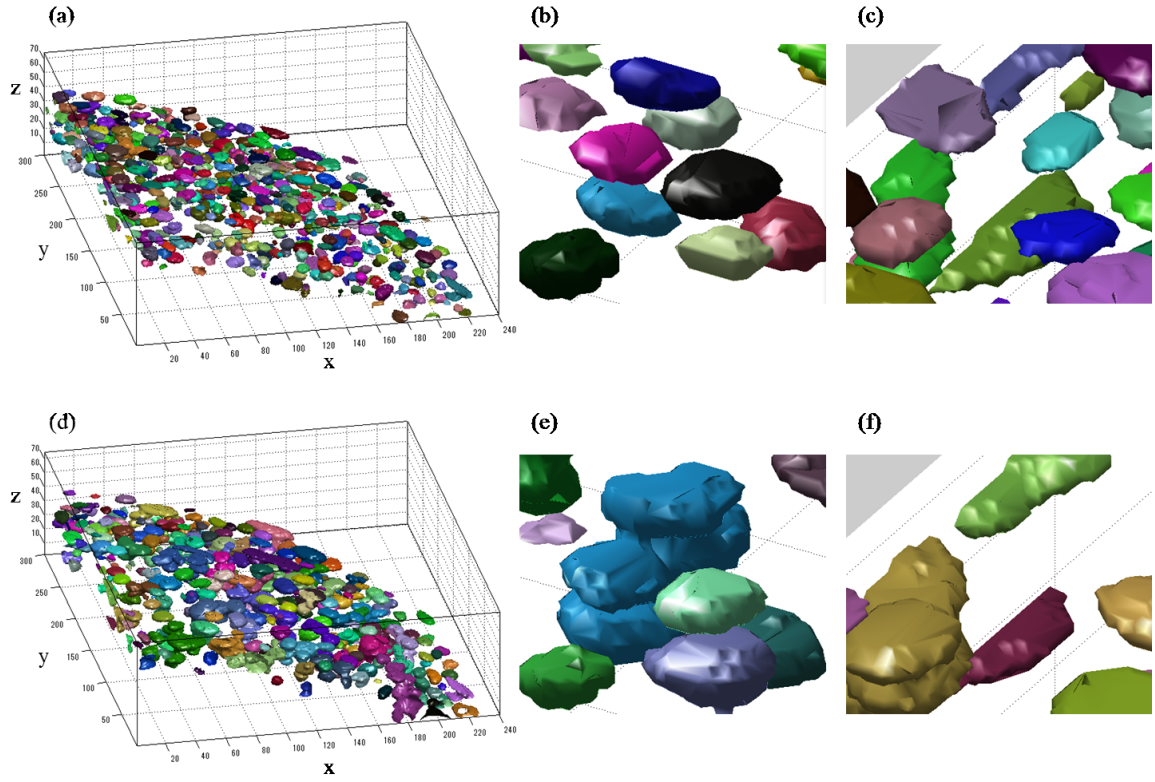


Figure 4.13: Examples of 3D detection results. (a) Overall results for the proposed method. (b) and (c) Enlarged image results for the proposed method. (d) Overall results for the Otsu thresholding method. (e) and (f) Enlarged image results for the Otsu thresholding method. Different colors indicate individual cells.

4.5 Conclusion and discussion

The proposed cell-detection method addresses all of the difficulties in detecting dense cells simultaneously under high-density conditions, including the mistaken merging of multiple cells, the segmentation of single cells into multiple regions, and the misdetection of low-intensity cells. The proposed method first detects candidate regions that include many false positives but in turn very few false negatives: i.e., the candidates include almost all correct solutions due to using multiple thresholds. The candidate set forms a tree structure. Next, the score for each candidate region is computed on the basis of cell appearance (shape, intensity, intensity gradients, *etc.*). Then the problem of selecting an optimal set of cell regions from the candidate regions is formulated as a binary linear programming problem under non-overlapping constraints by using the tree structure.

Evaluation of the proposed method and comparison of its detection performance with those of current cell-detection methods for several types of cells and microscopy images

demonstrated that the proposed method significantly improves the F-measure for various types of data-sets. Experiments demonstrated that the proposed method can be applied to 3D volume data successfully.

The proposed method improved the cell-detection accuracy, however, it is still far from perfect, i.e., the detection results includes some false positives and negatives. In these case, time-lapse information usually help us to correct these detection errors. Thus, I propose cell-tracking methods in following chapters to mitigate the problem.

Chapter 5

Cell Tracking by Solving Both Cell Detection and Association

5.1 Introduction

The global spatio-temporal data association method proposed in Chapter 3 works well if the tracklets are reliably generated. However, under high-density conditions, cells often touch or partially overlap and form cell clusters with blurry intercellular boundaries. Such conditions make it difficult to generate reliable tracklets. In such conditions, the method proposed in Chapter 2 also does not exhibit accurate tracking since cells often touch four or more cells.

In this chapter, I propose a tracking method that tracks cells under high-density conditions by solving a joint problem of both association and optimal region selection from redundant candidate regions. One of the disadvantages of detection-and-association methods, including the methods proposed in Chapter 2, is that the tracking process depends heavily on the detection step because these method first detect cell regions then associate them; in other words, the detection step is independent of the associate step and the errors of the detection step directly propagate to the association step. To mitigate this problem, the cell region information from the previous frame is used to help segment the blurry cells, rather than relying on only the image appearance at the current frame. The proposed method first detects cell region candidates in a process that may include many false positives but also features very few false negatives. Then, the optimal cell regions are selected from among the candidate regions by solving the association problem between the candidate regions at the current frame and the tracking results from the previous frame. Since the candidate regions may overlap, conventional association methods [Kofahi06],[House09] and [Padfield11] cannot be directly applied to this problem. I therefore formulate this problem as a binary linear programming problem containing constraints to avoid conflict associations. In addition, a

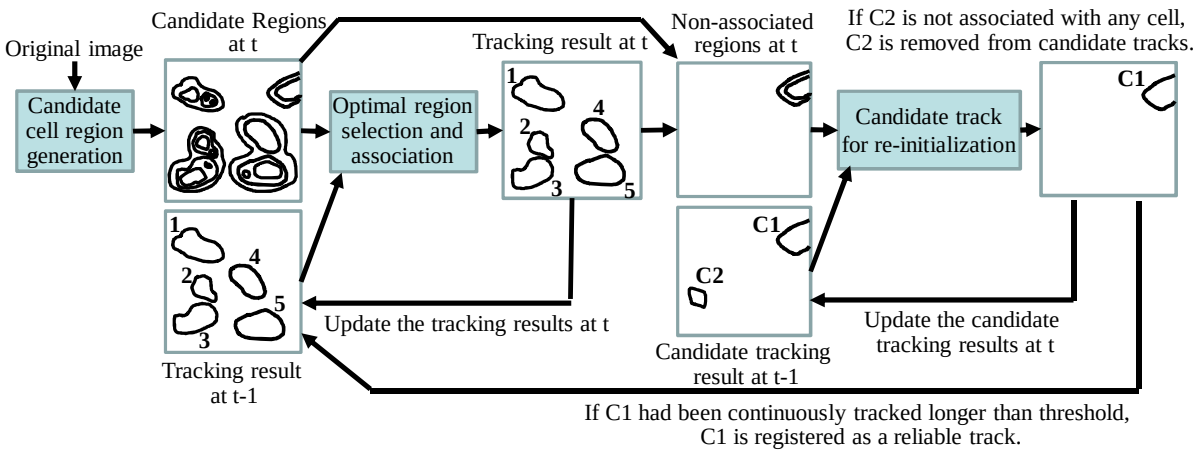


Figure 5.1: Method overview.

re-initialization step starts the tracking process for a candidate region only if the boundary of the cell has been clear for several frames continuously. This proposed method can create more reliable tracks under high-density conditions. I evaluated the proposed method on four sequences with hundreds of cells cultured under high-density conditions. The results show that the proposed method has an improved tracking performance compared with the state-of-the-art methods proposed in Chapters 2 and 3.

5.2 Method for jointly solving detection and association

The overview of the proposed method is shown in Figure 5.1 and the pseudo code of the method is shown in Algorithm 1. First, the method creates candidate cell regions (which may include many false positives but also very few false negatives) by using multiple thresholds for each frame (lines 4 to 9 in the pseudo code). Next, the method solves both problems that is, selecting the optimal cell regions from among the candidates and associating the optimal cell regions with tracked cell regions at the previous frame while avoiding conflicts by using binary linear programming (lines 10 to 19). Then, the non-associated candidate regions in the optimal region selection and association step are associated with candidate tracking results to judge if the candidate regions should be registered as the new track. If the candidate track has been tracked for several frames continuously, the method starts the tracking process (lines 20 to 33). This process is iteratively performed for each frame until the end of the sequence to make tracklets. Finally, post-processing is applied to connect the generated tracklets by using global spatio-temporal information (lines 34 to 35). The method is described in more detail below.

Algorithm 1 Cell tracking

Input: Sequence of images $I^1, ..., I^K$; the number of multiple-threshold N; the threshold th_{length}
Ouput: Sequence of segmentation images and tracking results $\mathbf{T}^*$.
for $t = 1$ **to** K **do**
  1) Candidate cell region generation
  for $n = 1$ **to** N **do**
    % Detect candidate regions by each threshold.
    $\mathbf{A_n^t} \leftarrow CandidateRegionDetection(I^t, th_n)$
    Add $\mathbf{A_n^t}$ **to** $\mathbf{A^t}$
  end for
  2) Optimal region selection and association
  % Get all hypotheses of association
  $\mathbf{H^t} \leftarrow GetHypotheses(\mathbf{X^{t-1}}, \mathbf{A_n^t})$
  for all hypothesis $h = 1$ **to** $length(\mathbf{H^t})$ **do**
    % Append a new row of constraint matrix C for the hypothesis $\mathbf{H_h^t}$
    $C(h, :) \leftarrow AddConstraints(\mathbf{H_h^t})$
    $\boldsymbol{\rho}(h, :) \leftarrow CmpLikeliness(\mathbf{H_h^t}, \mathbf{X^{t-1}}, \mathbf{A_n^t})$
  end for
  % Solve the association problem.
  $\mathbf{X^t} \leftarrow ResolveAssociation(C, \boldsymbol{\rho})$
  3) Candidate tracking for re-initialization
  $\mathbf{A2^t} \leftarrow GetNonAssociatedRegions(\mathbf{A^t}, \mathbf{X^t})$
  $\mathbf{H2^t} \leftarrow GetHypotheses(\mathbf{X2^{t-1}}, \mathbf{A2^t})$
  for all hypothesis $h = 1$ **to** $length(\mathbf{H2^t})$ **do**
    $C2(h, :) \leftarrow AddConstraints(\mathbf{H2_h^t})$
    $\boldsymbol{\rho 2}(h, :) \leftarrow CmpLikeliness(\mathbf{H2_h^t}, \mathbf{X2^{t-1}}, \mathbf{A2^t})$
  end for
  $\mathbf{X2^t} \leftarrow ResolveAssociation(C2, \boldsymbol{\rho 2})$
  for all candidate track $i = 1$ **to** $length(\mathbf{X2^t})$ **do**
    if $length(\mathbf{X2_i^t}) > th_{length}$ **then**
      Add $\mathbf{X2_i^t}$ **to** $\mathbf{X^t}$
    end if
  end for
end for
4) Post processing (global data association)
$\mathbf{T}^* \leftarrow GlobalAssociation(\mathbf{X})$

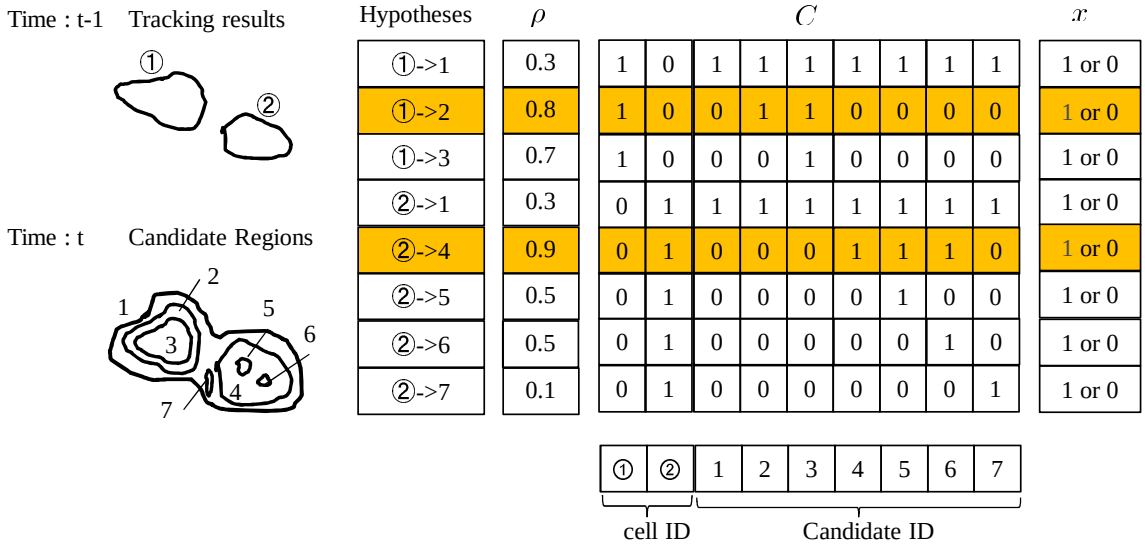


Figure 5.2: Examples of a constraint matrix and score vector for jointly solving detection and association.

5.2.1 Candidate cell region generation

In this step, the method generates candidates of cell regions that may include many false positives but also very few false negatives. For this step, the same process with a step of detecting set of candidate regions with the method proposed in Chapter 4 is used. I denote the set of candidate cell regions as $\mathbf{A^t} = \{A_i^t\}$, where A_i^t represents the ith candidate cell blob at frame t.

5.2.2 Optimal region selection and association

After generating the candidate cell regions, the method determines optimized cell regions by solving the association between candidate regions in the current frame t and cells that have been tracked up to frame $t-1$. This problem is formulated as a binary linear programming problem.

Let N_1 be the number of cells at the frame $t-1$, let N_2 be the number of candidate cell regions at the frame t, let vector $\boldsymbol{\rho}$ store the scores of every possible hypothesis, and let matrix C store the constraints to avoid conflict hypotheses, where each row of C has $N_1 + N_2$ columns and each column on 1 to N_1 indicates cell index and each column on N_1+1 to N_1+N_2 indicates candidate region index on the association between track results and candidate regions. $\Omega(A_i^t)$ is a set of candidate region indexes that are all descendant node indexes of A_i^t in the tree. The set includes A_i^t. For example, in Figure 5.2, $\Omega(A_1^t)$ is $\{1,2,3,4,5,6,7\}$ since the candidate region A_1^t is overlapped with all of the other candidates. $\Omega(A_2^t)$ is $\{2,3\}$. If the region of cell l at $t-1$ overlaps with candidate m at t, the cell

migration hypothesis $cell_l \rightarrow A_m^t$ is added to the hypotheses set. Let h be the index of a new hypothesis, where a new row to C and a corresponding score to $\boldsymbol{\rho}$ are appended:

$$C(h,i) = \begin{cases} 1, \text{ if } i = l \text{ or } i = N_1 + m_r, m_r \in \Omega(A_m^t) \\ 0, \text{ otherwise.} \end{cases} \tag{5.1}$$

$$\boldsymbol{\rho}(h) = P_{mig}(A_m^t | cell_l) P_{TP}(A_m^t), \tag{5.2}$$

where $P_{mig}(A_m^t|cell_l)$ is the score of migration hypothesis $cell_l \rightarrow A_m^t$. $P_{TP}(A_m^t)$ is the score in which the region of A_m^t is a single cell region. For the implementation, since the cells do not move fast under high-density, I use 'relative overlap' between cell region $cell_l$ and candidate region A_m^t, i.e.,

$$P_{mig}(A_m^t|cell_l) = \frac{cell_l \cap A_m^t}{cell_l \cup A_m^t} \tag{5.3}$$

I assume that the area surrounded by a clearer edge is more likely to contain a single cell since the intensity gradient of the cell boundaries is usually higher than the intensity gradient of the boundaries of the cell nucleus in precondition images. For the implementation, I compute the score in which the candidate region A_m^t is a single cell region as

$$P_{TP}(A_m^t) = e^{-\frac{1}{E_{edge}} \frac{1}{\sigma}} \tag{5.4}$$

$$E_{edge}(A_m^t) = \frac{1}{length(\Phi_{A_m^t})} \int_{\Phi_{A_m^t}} e(\Phi_{A_m^t}) dl, \tag{5.5}$$

where the edge energy E_{edge} measures the edgeness along the region boundaries. $\Phi_{A_m^t}$ is the region boundary of candidate region A_m^t. The function $e(\cdot)$ is the edgeness metric, which takes a large value if the intensity gradient on $\Phi_{A_m^t}$ is large. $length(\Phi_{A_m^t})$ indicates the length of the region boundary. When the edgeness of the region boundary takes a larger value, the score $P_{TP}(A_m^t)$ closes to 1. After computing the score vector $\boldsymbol{\rho}$ and constraint matrix C of all H hypotheses over N_1 cells and N_2 candidate regions, the association problem can be formulated as the following binary linear programming:

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}} \boldsymbol{\rho}^T \boldsymbol{x}, \quad s.t. \ C^T \boldsymbol{x} \leq \mathbf{1}, \tag{5.6}$$

where x is a $H \times 1$ binary vector and $x_k = 1$ means the kth hypothesis is selected in an optimal solution.

Figure 5.2 shows a simple example of a binary linear programming problem in which the number of tracking results at the previous frame is 2 and the number of candidate regions is 7. In the hypothesis $cell① \rightarrow A_1^t$, the set of descendant candidates with A_1^t is $\{1,2,3,4,5,6,7\}$. This constraint indicates that the candidates $\{2,3,4,5,6,7\}$ are not associated with any cells if the region A_1^t is selected as an optimal solution. In this example,

the hypotheses $cell① \rightarrow A_2^t$ and $cell② \rightarrow A_4^t$ are selected as the optimal solution. Based on this process, a set of tracklets $\mathbf{X} = \{X_i\}$ is generated. The term $X_i = \{R_{i_j}\}$ is a trajectory consisting of an order list of associated cell regions where the associated candidate region is registered as R_{i_j}, which indicates the jth detection result on tracklet X_i.

5.2.3 Initialization of cell regions for tracking

The initialization of cell regions is a key stage for tracking methods. If the detected region at the previous frame is not reliable, the tracking result at the current frame is also unreliable since the tracking method uses the results from the previous frame. In a sequence, a boundary of a cell is sometimes clear and sometimes blurry while the cell migrates under high-density. Generally, it is difficult to segment cell regions that have blurry boundaries. However, I can make informed guesses about the blurry regions by consulting the cell regions from the previous frames if they have clearer boundaries than the target frame. The initialization step is based on this idea. As I discussed in the previous section, the proposed method terminates the tracking process if the reliability is less than a threshold. This means that some cells are not tracked. Here, the initialization process is intended to determine which cell regions are reliable in the set of candidate cell regions and which are not tracked. The proposed method only initializes cell regions and starts the tracking process if the boundary of the cell has been clear for several frames continuously.

Since the targets of the initialization are the cells that are not tracked, to initialize tracklets at frame t, the method first removes candidates that are overlapped with selected optimal cell regions in the process described in section 5.2.2 at each frame $t-1$ and t. Here, the rest of the candidate blobs is denoted as $\mathbf{A2^t} = \{A2_i^t\}$. Next, the method determines which cell regions have boundaries that are continuously clear in several consecutive frames, as these are considered reliable. The candidates $\mathbf{A2^t}$ and $\mathbf{A2^{t-1}}$ are associated by using the same method proposed in section 5.2.2, where the only difference is the association score $\boldsymbol{\rho}(h)$ of the hypothesis $A2_l^{t-1} \rightarrow A2_m^t$.

$$\boldsymbol{\rho}(h) = P_{mig}(A2_m^t | A2_l^{t-1}) P_{TP}(A2_m^t) P_{TP}(A2_l^{t-1}) \tag{5.7}$$

Using this score, the optimal candidate regions and associations are solved by linear programming. Here, the high value of the score indicates that both boundaries of the regions $A2_m^t$ and $A2_l^{t-1}$ are clear and that these regions appear to be the same object. The purpose of this step is to initialize the tracker for clear cell regions. Thus, the associated candidate region is linked to a trajectory if and only if its score is higher than a threshold. These selected regions are tracked until the reliability of the association is less than threshold by using the method proposed in the previous section. Then, if the length of the generated tracklet is longer than a threshold (default is 5), the tracklet is registered as reliable.

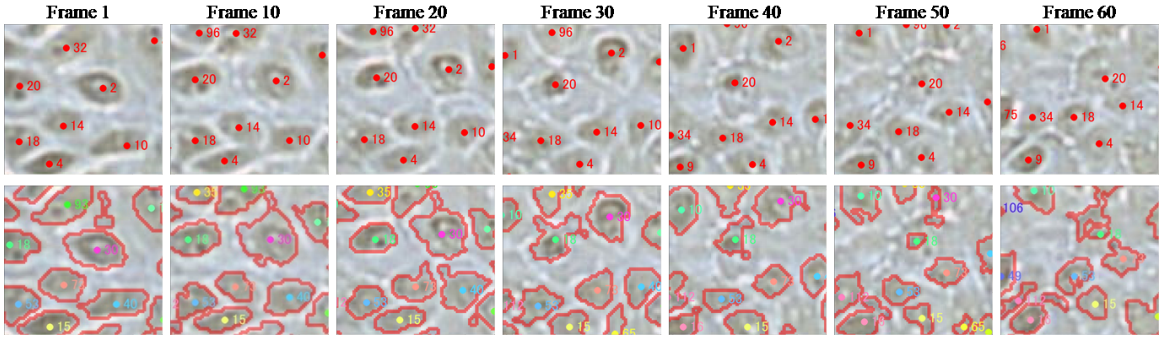


Figure 5.3: Example images of tracking results. Top: Original images with manual annotation. The red points are annotated cell positions. Bottom: Tracking results. The numbers in the images are cell IDs. The red outline indicates the cell contour.

5.2.4 Post processing (global data association)

Generated tracklets are globally associated over time to obtain final cell trajectories and lineage trees. To solve the global data association among generated tracklets, I use the tree structure association method proposed in Chapter 3.

5.3 Experimental results

5.3.1 Data

My collaborators cultured retinal pigment epithelium (RPE) cells for 30 days. Since it is an enormous task to analyze the entire data, I used four sequences captured at a resolution of 320 $\times$ 320 pixels (1.03 μm/pixel). The cells were imaged every 2.5 minutes by bright field microscopy at 7.5X magnification over 4 hours (100 images) and I evaluated them at day 1, day 7, day 14, and day 21. The population of the cells ranged from 100 to 200 and the density ranged from 90% to 100%. For each image sequence, 100 cells were randomly picked in the initial frame and their progeny cells were manually tracked.

5.3.2 Tracking results and evaluation

Figure 5.3 shows an example sequence of the tracking results. In this sequence, the cells migrated and became more blurry as time went by. At frame 60, it is difficult to segment the cells even manually if an annotator does not use the previous frame information since the boundaries of the cells are so blurry. Despite such challenging images, the cells were well segmented at the beginning of the sequence and then effectively tracked over all the frames. Figure 5.4 (a)(c) shows example images of cell tracking with track IDs and segmented

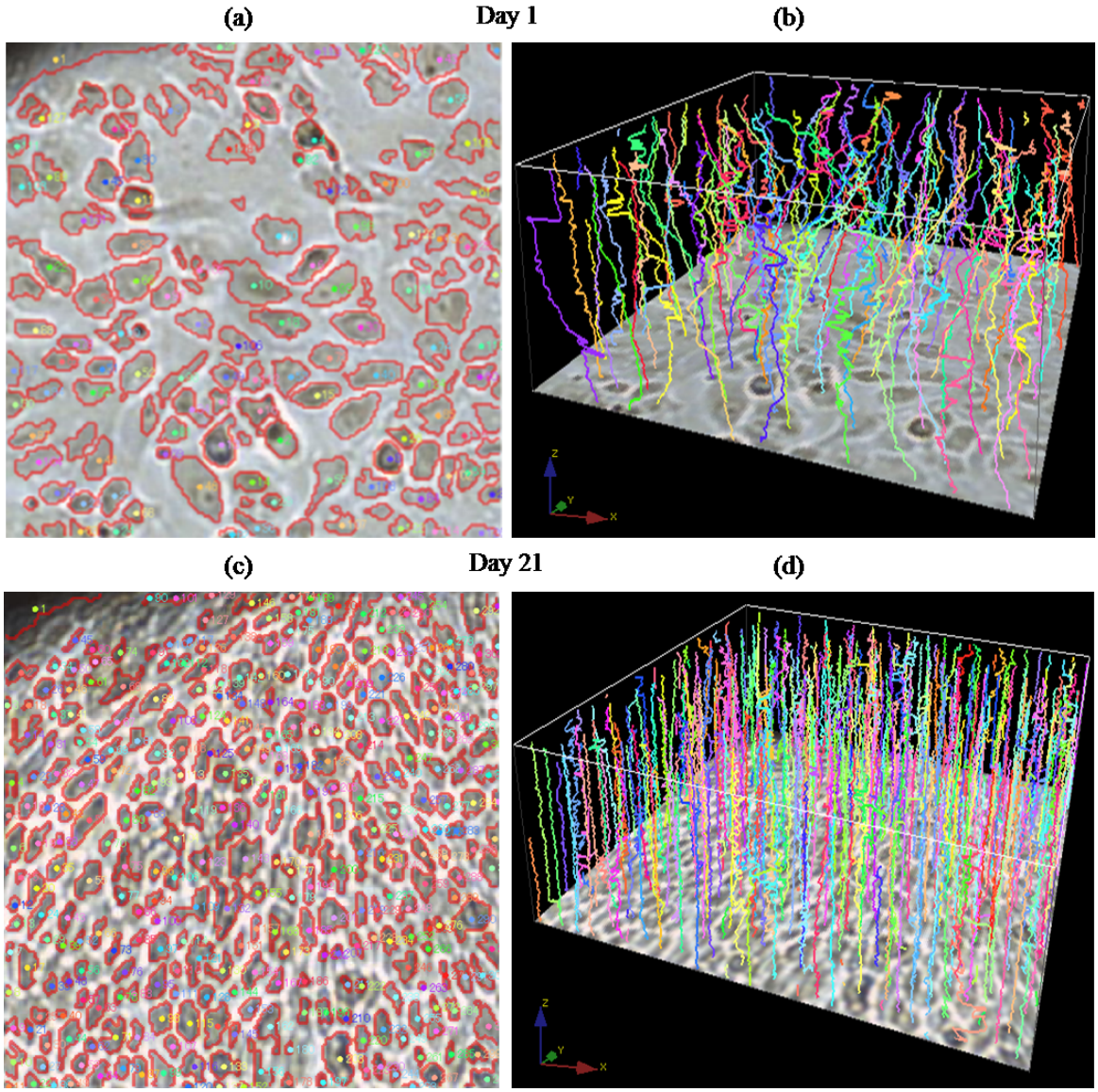


Figure 5.4: Examples of tracking result images with track IDs and segmented regions in the (a) day 1 and (c) day 21 sequences. Space-time trajectory plot of entire sequences in (b) day 1 and (d) day 21. X and Y axes are 2D space and Z axis is time.

regions, and Figures. 5.4 (b)(d) shows space-time trajectory plots of the entire sequence for days 1 and 21. These results show that the cells were accurately tracked.

I used target effectiveness [Blackman86] to assess the tracking performance, where the details of the metrics is described in Appendix A. Here, I did not evaluate branching correctness since almost cells do not divide in this experiments.

For the quantitative evaluation, I compared the proposed method with four methods:

	Day 1	Day 7	Day 14	Day 21	Average
frame-by-frame association	0.6891	0.4068	0.4266	0.5462	0.5172
global association	0.7014	0.4142	0.4445	0.5508	0.5277
frame-by-frame association with improved detection	0.6902	0.7055	0.7428	0.8587	0.7493
global association with improved detection	0.8331	0.7183	0.7678	0.865	0.7961
Proposed without global association	0.8267	0.7940	0.8354	0.9766	0.8582
Proposed with global association	0.8483	0.840	0.8482	0.9766	0.8782

Table 5.1: Comparison of target effectiveness of the proposed method with the methods proposed in Chapters 2 and 3 on four sequences.

frame-by-frame association (Chapter 2); global data association (Chapter 3); frame-by-frame association with improved detection (Chapter 2+4); global association with improved detection (Chapter 3+4). Since both of methods proposed in Chapters 2 and 3 use a simple threshold method after preconditioning in the detection step, they cannot perform well when it comes to images under a dense condition, and segmentation errors harm the tracking accuracy. To ensure a fair comparison, I used the detection method proposed in Chapter 4. The segmentation results were better than the original simple threshold method in frame-by-frame association and global data association. To better highlight the advantages of the method proposed in this chapter, I used it without post processing (global association).

As shown in Table 5.1, the method proposed in this chapter achieved the best performance on all of the sequences compared with frame-by-frame association method, and global association method. The global association method performed slightly better than the frame-by-frame association method. In the results of global association, tracklets were not reliably generated since the high-density conditions caused many segmentation errors. Even though the accuracies of frame-by-frame association and global association with the improved segmentation were much better than those with the original segmentation results. The method proposed in this chapter performed better than both. The proposed method with post processing was only a little bit better (2%) than the method without post processing. These results indicate that the tracking method for jointly solving detection and association has a greater effect on improving the accuracy under high-density conditions than post processing (global data association). On average, the proposed method improved

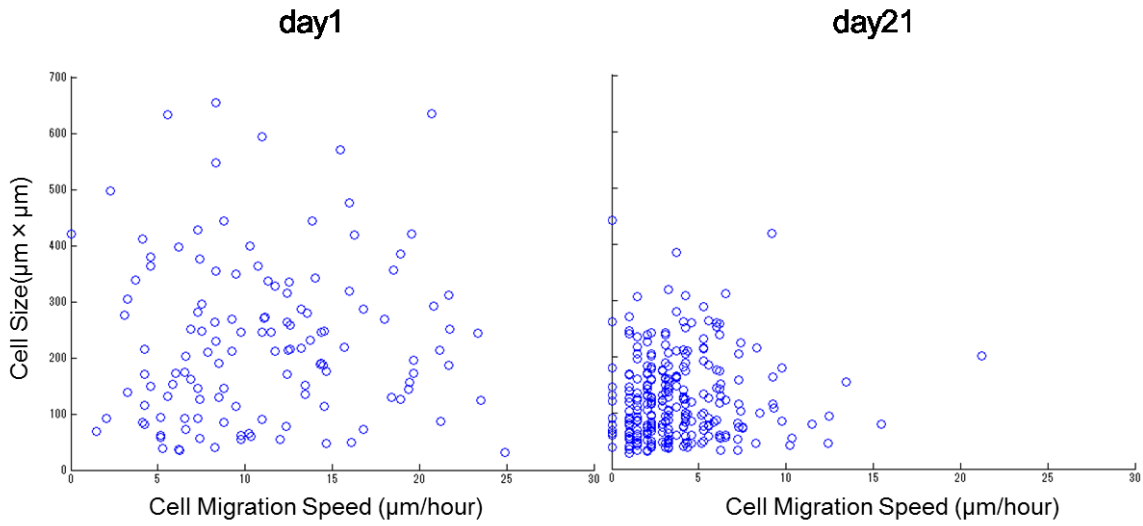


Figure 5.5: Example images of cell behavior metrics. Left: The scatter plot on day 1. Right: The scatter plot on day 21. The horizontal axis is the cell migration speed (μm) and the vertical axis is cell size (μm^2).

	Day 1	Day 7	Day 14	Day 21
Average cell size ($\mu^2 m$)	225.01	157.48	152.84	108.57
Average migration speed (μm/hour)	10.90	3.42	3.249	3.18

Table 5.2: Cell behavior metrics on four sequences.

the target effectiveness over 8% compared with the other methods.

5.4 Biological applications

In stem cell research, it is important to assess the quality of cells for transplant by non-invasive methods. I applied my cell-tracking method to a biological study on the relationship between cell behaviors and the maturity of retinal pigment epithelium (RPE) cells. RPE cells gradually mature as they are cultured for 30 days and are therefore more mature on day 21 than on day 1. To analyze the difference in cell behaviors depending on the amount of culturing days, I computed the cell migration speed and the cell size by using the tracking results. Figure 5.5 shows the scatter graphs of the cell size and migration speed on days 1 and 21. On day 1, both the cell size and the migration speed were bigger and more variable than on day 21. In addition, the average cell size and migration speed on day 1 were much bigger than the other sequences (Table 5.2). These metrics slightly decreased as

the days passed. These results indicate that the metrics may be related to the degree of cell maturity. I intend to continue researching metrics for evaluating cell maturity. The results also demonstrate that my tracking system can track cells non-invasively under high-density conditions, thus making it of enormous use to biological research.

5.5 Conclusion

I proposed a tracking method which jointly solves detection and association to track cells under high-density conditions. To reduce the problems evident in conventional methods based on detenction-and-association methods, which heavily depend on the detection results, the method proposed in this chapter determines the detection results in the tracking process by using both image features in the current frame and the tracking results from the previous frame. Although I used a multiple-threshold method to generate candidate regions, the proposed method can be applied to other types of detecting methods that have many false positives but few false negatives. For example, super-pixel segmentation methods can be applied. In this case, a combination of super-pixels can function as a candidate region.

I evaluated the proposed method using challenging image sequences in which cells were cultured under a high-density condition and had blurry boundaries. Results show that the proposed method significantly improves the tracking performance on target effectiveness.

Chapter 6

Cell Tracking by Jointly Solving Tracklet Selection and Global Association

6.1 Introduction

The method proposed in Chapter 5 determines the detection results in the association step by using both image features in the current frame and the tracking results in the previous frame. This helps to generate more reliable tracklets than traditional frame-by-frame methods. However, the method still depends on initialization of cell detection since the joint problem of optimal region selection and association is solved at each frame independently. To avoid error propagation from detection/initialization to data association, it is ideal to simultaneously solve both global association and detection by optimizing a single objective function. However, to the best of my knowledge, no ideal approach has been proposed.

In this chapter, I propose a tracking method that simultaneously solves both global association and tracklet selection from redundant tracklets. Since jointly solving segmentation and association is still difficult, I simplify the cell segmentation problem by, for example, selecting an optimal tracklets from redundant candidate tracklets. This method first generates redundant candidate tracklets, which include many false positives but in turn very few false negatives, by allowing tracklets to overlap. To generate reliable tracklets, the association problem is solved with constraints that preserve three structures of detection candidate between successive frames. Next, the problems of selecting an optimal set of cell tracklets from the redundant tracklets and associating the tracklets over frames under non-overlapping constraints are solved simultaneously. This idea of optimization is based on the methods proposed in Chapters 3 and 5. The method proposed in this chapter achieved the best performance compared with the tracking methods proposed in Chapters 2, 3, and 5

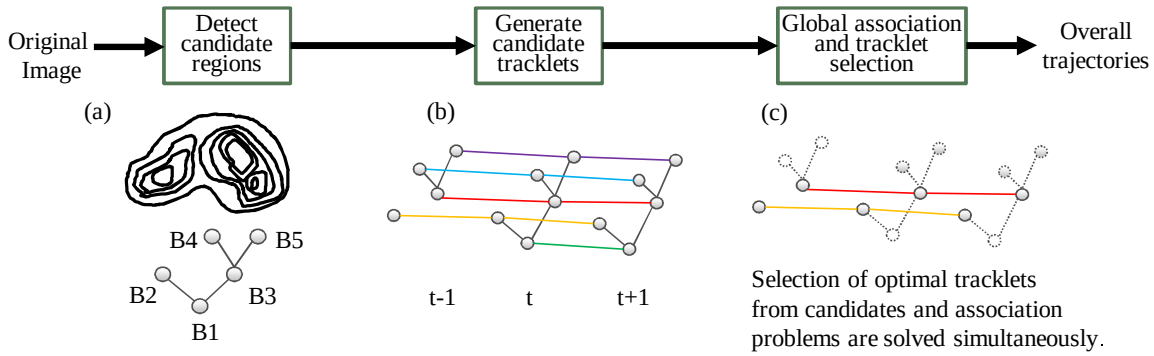


Figure 6.1: Cell-tracking method overview. Bottom images illustrate example in which two cells touch and form cluster in each step. (a) Example illustration of detected candidates. (b) Redundant candidate tracklets generated by linking detection results. (c) Optimal solution set in which selected tracklets are not in conflict.

under dense culture conditions.

6.2 Method for jointly solving tracklet selection and global association

Figure 6.1 shows an overview of the proposed cell-tracking method. First, the method generates candidate cell regions that may include many false positives but very few false negatives, by using multiple thresholds for each frame. The generated detection candidates form tree structures, as shown in Figure 6.1(a). Next, the candidate detection results are associated with redundant candidate tracklets, which may also include many false positives but in tern very few false negatives. To generate reliable tracklets, the association problem is solved with constraints that preserve tree structures of detection candidates between successive frames. The generated tracklets may conflict with each other as shown in Figure 6.1(b). Finally, the problems of selecting an optimal set of cell tracklets from the redundant tracklets and associating the tracklets over frames under non-overlapping constraints are solved simultaneously by binary linear programming. The conflicted tracklets with the selected tracklets are eliminated, as shown in Figure 6.1(c). The method is described in more detail below.

6.2.1 Cell detection

For this step, the same process with a step of detecting set of candidate regions with the methods proposed in Chapter 4 and 5 is used with the following parameters: K level thresholds that are equally spaced, and each threshold is used to segment images at a particular level of intensity, and the minimum and maximum cell sizes θ_{min} and θ_{max}. The

set of candidate cell regions is denoted as $\mathbf{A}^t = \{A_i^t\}$. The scores of the candidate regions are computed as explained in Chapter 4, where the score indicates how much each region looks like a single cell. The score of candidate region A_i^t is denoted as $P_{shape}(A_i^t)$. A tree structure is also generated in which the candidate regions are nodes, and the relationships between nodes are generated on the basis of information about the containment relationship. This tree is used to represent constraints for generating reliable tracklets in the second step and to avoid selecting conflicting tracklets in the third step.

6.2.2 Tracklet generation

As discussed in Chapters 2 and 5, it is difficult to determine cell regions from local temporal information when multiple cells touch and make a cluster for several frames. In this step, a set of redundant candidate tracklets from candidate detections that include an optimal set of reliable tracklets is generated. Figure 6.2 shows examples of generated tracklets, in which two cells touch and form a cluster then eventually separate into individual cells.

In the tracklet generating step of the method proposed in Chapter 3, a tracklet is considered reliable when cell blobs in consecutive frames are close enough and there are no extra confusing blobs near the cell. However, this idea cannot be directly applied under dense cell culture conditions since there are many confusing candidates, such as parent and children nodes of the target candidate regions in the generated tree, near a cell under such conditions. Therefore, I also propose a frame-by-frame tree matching method that preserves tree structures of associated detection candidates between successive frames to avoid generating tracklets that include association errors. This frame-by-frame tree matching method maintains the containment relationships among the associated detection candidates between the consecutive frames.

Figure 6.3 shows examples of possible associations and conflicting associations. There are three types of containment relationships in a tree, parents, brothers, and children. For example in Figure 6.3, nodes 2 and 3 are child nodes of node 1; node 1 is a parent node of node 2; node 3 is a brother node of node 2. Association of the same types of relationships is possible. In this example, if '1 to a' is a true association, '2 to b', '2 to c', '3 to b', and '3 to c' are possible associations. Association between different types of relationships is not allowed. For example, if '2 to a' is a true association, '1 to b' and '1 to c' are not allowed. The other types of conflicting associations are described in Figure 6.3.

This association problem of maximizing an association score function with constraints that preserve such containment relationships in a tree is formulated by binary linear programming. In this association, '1 to 1' association (translation hypothesis) is only considered, i.e., the other types of associations including, enter, exit, and divisions, are solved at the next step. All possible translation hypotheses are first listed if a relative overlap

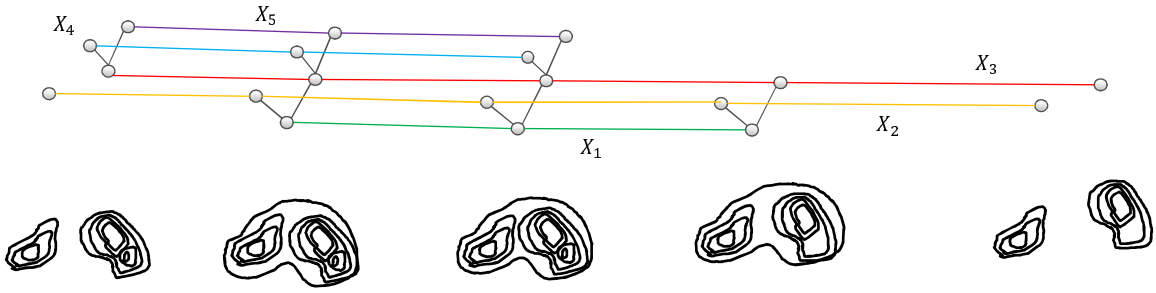


Figure 6.2: Examples of tracklets in which two cells touch and form cluster then eventually separate into individual cells.

between candidates (Eq. 5.3 in Chapter 5) is larger than threshold θ_d since it is considered that the association is not reliable if the relative overlap is small. The optimal association from the hypothesis set is found by solving a binary linear programming problem that is similar to the optimization approach proposed in the previous chapters. The difference is introducing a constraint to preserve tree structures of candidates between successive frames. The formulation is described as follows.

Let N_t be the number of cells at frame t, and H be the total number of hypotheses that I have to consider for all the cell. An $H \times 1$ vector $\boldsymbol{\rho}$ stores the association score for every hypothesis, $H \times (N_t + N_{t+1})$ matrix C stores the constraints needed to avoid selecting conflicted hypotheses, and $H \times H$ matrix C_{tree} stores the constraints needed to preserve tree structures of candidates between successive frames. Let h be the index of a hypothesis in which the hypothesis is 'A_i^t to A_j^{t+1}'. The h-th row of C,C_{tree} and $\boldsymbol{\rho}$ are determined as follows:

$$
\begin{aligned}
C(h,k) &= \begin{cases} 1, & \text{for } k = i \text{ and } k = N_t + j \\ 0, & \text{otherwise.} \end{cases} \\
C_{tree}(h,k) &= \begin{cases} H, & \text{if } k = h \\ 1, & \text{if } k \in \Psi(A_i^t, A_j^{t+1}) \\ 0, & \text{otherwise.} \end{cases} \\
\boldsymbol{\rho}(h) &= P_{mig}(A_j^{t+1} | A_i^t)
\end{aligned} \tag{6.1}
$$

where $\Psi(A_i^t, A_j^{t+1})$ is a set of hypothesis indices that conflict with hypothesis h. The conflicted hypothesis index set $\Psi(A_i^t, A_j^{t+1})$ includes hypotheses of $\Lambda(A_i^t) \rightarrow \Upsilon(A_j^{t+1})$ (parent to brother), $\Lambda(A_i^t) \rightarrow \Gamma(A_j^{t+1})$ (parent to child), $\Upsilon(A_i^t) \rightarrow \Lambda(A_j^{t+1})$ (brother to parent), $\Upsilon(A_i^t) \rightarrow \Gamma(A_j^{t+1})$ (brother to child), $\Gamma(A_i^t) \rightarrow \Lambda(A_j^{t+1})$ (child to parent), and $\Gamma(A_i^t) \rightarrow \Upsilon(A_j^{t+1})$ (child to brother), where $\Lambda(A_i^t)$ indicates a set of ancestor nodes of A_i^t, $\Upsilon(A_i^t)$ indicates a set of brother nodes of A_i^t, and $\Gamma(A_i^t)$ indicates a set of descendant nodes of A_i^t. The probability $P_{mig}(A_j^{t+1} | A_i^t)$ is a relative overlap between A_i^t and A_j^{t+1} as defined at Eq. 5.3.

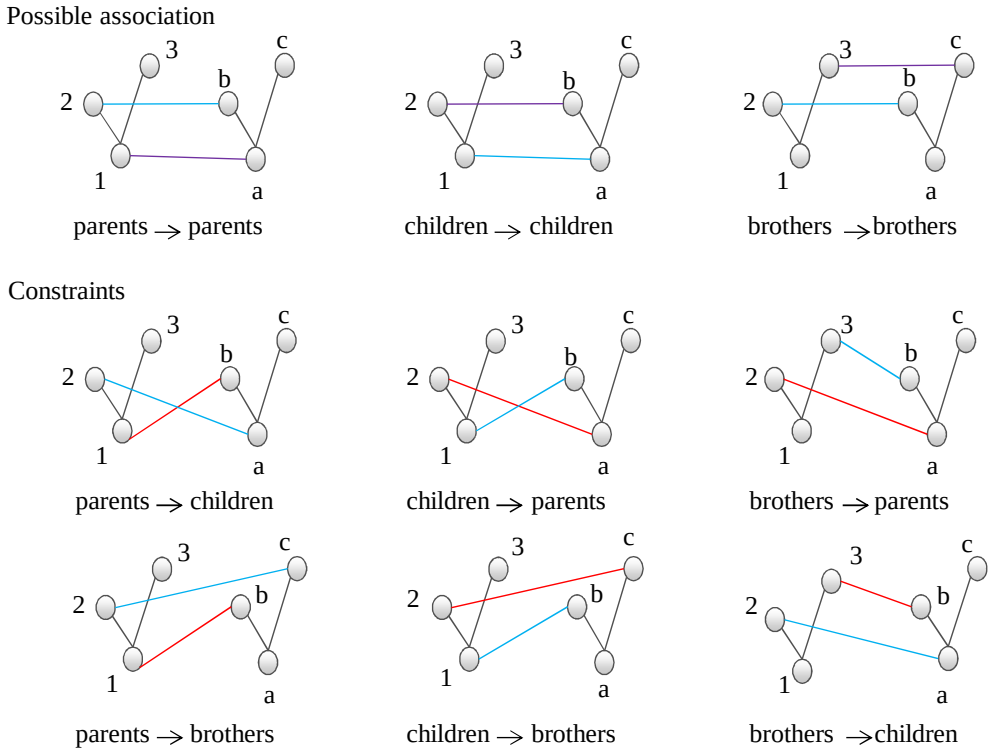


Figure 6.3: Examples of possible associations and conflicting associations. Light blue indicates base association. Purple indicates possible association that does not conflict with light blue. Red indicates conflicting association with light blue.

After score vector $\boldsymbol{\rho}$ and constraint matrices C, C_{tree} are generated, the optimization problem can be formulated as a binary linear programming problem:

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}} \boldsymbol{\rho}^T \boldsymbol{x}, \quad s.t.\ C^T \boldsymbol{x} \leq \mathbf{1}, C_{tree} \boldsymbol{x} \leq \boldsymbol{H} \tag{6.2}$$

where $\boldsymbol{x}$ is an $H \times 1$ binary vector and $x_k = 1$ means the k-th candidate region is selected in the optimal solution. The term $\mathbf{1}$ is an $H \times 1$ vector in which all elements are 1, and $\boldsymbol{H}$ is an $H \times 1$ vector in which all elements are H. The optimization problem is solved using the branch-and-bound algorithm [Ross75]. The constraint $C^T \boldsymbol{x} \leq \mathbf{1}$ guarantees that each candidate region ID appears in only one selected hypothesis, i.e., a candidate is associated with only another candidate between successive frames. The constraint $C_{tree} \boldsymbol{x} \leq \boldsymbol{H}$ guarantees to preserve tree structures of candidates between successive frames. Figure 6.4 shows an example of the tree matching constraints. This constraint means that the i-th column of other optimal solutions should be 0 if the i-th hypothesis is selected as an optimal solution. For example, if the first hypothesis '1 $\rightarrow$ a' is selected in an optimal solution, hypotheses 2, 3, 4, 5, 6, and 8 are not selected in the optimal solution since these first column are 1, i.e., these hypotheses conflict with hypothesis '1 $\rightarrow$ a'. In this example,

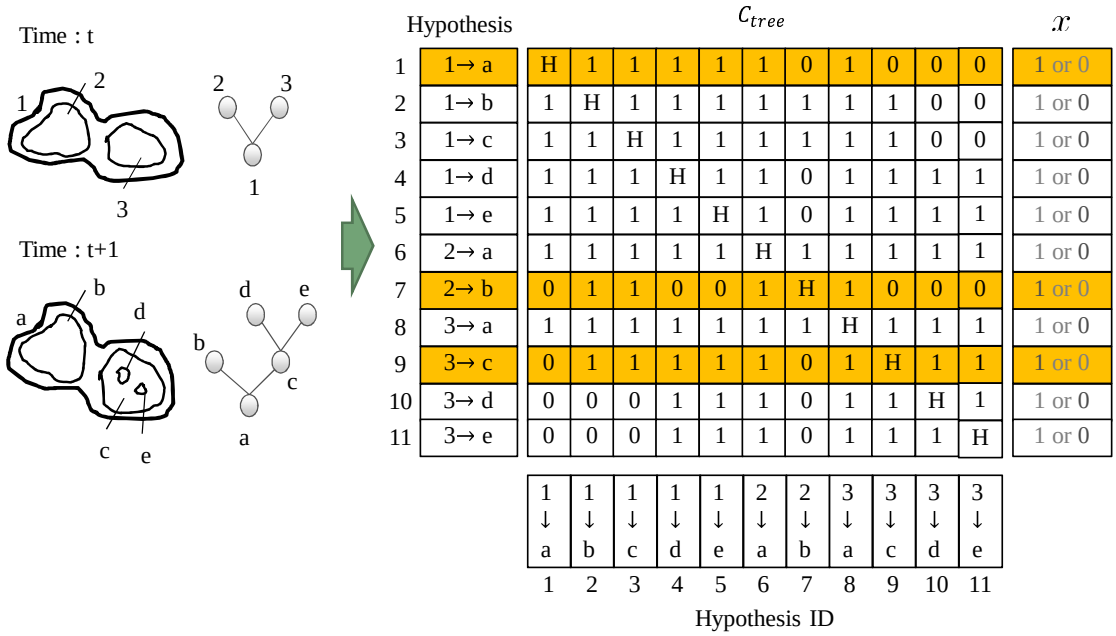


Figure 6.4: Examples of constraint matrix. Optimal solution is highlighted in orange.

'$1 \rightarrow a$', '$2 \rightarrow b$', and '$3 \rightarrow c$' are selected as an optimal solution set. 10

Based on this tree-structure preserving association method, a set of reliable tracklets $\mathbf{X} = \{X_i\}$ is generated. The term $X_i = \{A_{i_j}\}$ is a tracklet consisting of an order list of detection results in which A_{i_j} indicates the jth detection result on tracklet X_i. Any isolated detection response that is not linked with any other is considered a tracklet and also included in $\mathbf{X}$. The generated tracklets also form a tree structure on the basis of the containment relationships of detection candidates. Unclear associations are solved at the next step by using global data association.

6.2.3 Global data association

In this section, I propose a method that simultaneously solves both global association and tracklet selection from redundant tracklets with non-overlapping constraints in which conflicted tracklets are not selected. The global association problem is similar to the method proposed in Chapter 3. The differences are introducing the non-overlapping constraints to avoid selecting conflicted tracklets and computing manner of $P_{TP}(\mathbf{X}_i)$ and $P_{FP}(\mathbf{X}_i)$. For the sake of explicitness, I explain the entire of this proposed method, which partly overlaps with the method proposed in Chapter 3, in detail.

Given the redundant tracklet set $\mathbf{X}$, the posteriori probability is maximized to solve for

the best hypothesis $\mathbf{T}^*$.

$$\begin{aligned}\mathbf{T}^* &= \arg\max_{\mathbf{T}} P(\mathbf{T}|\mathbf{X}) \\ &= \arg\max_{\mathbf{T}} P(\mathbf{X}|\mathbf{T})P(\mathbf{T}) \\ &= \arg\max_{\mathbf{T}} \prod_{X_i \in \mathbf{X}} P(X_i|\mathbf{T}) \prod_{T_k \in \mathbf{T}} P_{Tree}(T_k) \end{aligned} \tag{6.3}$$

As explained in Chapter 3, the MAP problem can be formulated as

$$\begin{aligned}\mathbf{T}^* = \arg\max_{\mathbf{T}} \{ & \sum_{X_i \notin T_k,\ \forall T_k \in \mathbf{T}} \log P_{FP}(X_i) \\ & + \sum_{X_i \in T_k,\ \forall T_k \in \mathbf{T}} \log P_{TP}(X_i) \\ & + \sum_{X^0_{k_0} \in E_{k_0},\ E_{k_0} \in T_k,\ \forall T_k \in \mathbf{T}} \log P_{ini}(X^0_{k_0}) \\ & + \sum_{X^j_{k_i},\ X^{j-1}_{k_i} \in E_{k_i},\ \forall E_{k_i} \in T_k,\ \forall T_k \in \mathbf{T}} \log P_{link}(X^j_{k_i}|X^{j-1}_{k_i}) \\ & + \sum_{\{X^{end}_{k_{p_i}},\ X^0_{k_{c_{i1}}},\ X^0_{k_{c_{i1}}}\} \in \mathbf{B_k},\ \mathbf{B_k} \in T_k,\ \forall T_k \in \mathbf{T}} \log P_{div}(X^0_{k_{c_{i1}}}, X^0_{k_{c_{i2}}}|X^{end}_{k_{p_i}}) \\ & + \sum_{X^{end}_{k_{l_i}} \in E_{k_{l_i}},\ \forall E_{k_{l_i}} \in \mathbf{L_k},\ \mathbf{L_k} \in T_k,\ \forall T_k \in \mathbf{T}} \log P_{term}(X^{end}_{k_{l_i}}) \} \end{aligned} \tag{6.4}$$

I should consider the non-overlapping constraints since candidate tracklets may overlap, i.e., some tracklets are conflicting. Introducing these constraints differs from that with the method proposed in Chapter 3.

Let $N_{\mathbf{X}}$ be the number of tracklets in the entire sequence, M the number of tracklet association hypotheses. $M \times 1$ vector $\boldsymbol{\rho}_g$ stores the score of every possible hypothesis, and $M \times 2N_{\mathbf{X}}$ matrix C_g stores the constraints to avoid conflicting hypotheses, where each column indicates a tracklet index on the association between two tracklets. In addition to these matrix and vectors, I define the non-overlapping $M \times N_{\mathbf{X}}$ constraint matrix C_{over}, in which each column indicates a tracklet index, to avoid selecting conflicting tracklets.

1. Initialization hypothesis:
   If the first blob of a tracklet X_k appears at the beginning of the sequence or appears near the boundary of the field of view, the tracklet is a candidate of a initial tracklet. Let h be the index of the hypothesis. The h-th rows of C_g, $\boldsymbol{\rho}_g$ and C_{over} are determined

as:

$$\begin{aligned} C_g(h,i) &= \begin{cases} 1, & \text{if } i = N_{\mathbf{X}} + k \\ 0, & \text{otherwise.} \end{cases} \\ C_{over}(h,i) &= \begin{cases} 1, & \text{if } i = k \text{ or } i = m_r,\ \ m_r \in \Omega(X_k) \\ 0, & \text{otherwise.} \end{cases} \\ \boldsymbol{\rho}_g(h) &= \log P_{ini}(X_k) + 0.5 \log P_{TP}(X_k) \end{aligned} \tag{6.5}$$

where $\Omega(X_k)$ is a set of hypothesis indices that are all descendant tracklet indeces of $\mathbf{X}_k$.

2. Termination hypothesis:
   If the last blob of tracklet X_k appears at the end of the sequence or appears near the boundary of the field of view, the tracklet is a candidate of a termination tracklet. The h-th rows of C_g, $\boldsymbol{\rho}_g$ and C_{over} are determined as:

$$\begin{aligned} C_g(h,i) &= \begin{cases} 1, & \text{if } i = k \\ 0, & \text{otherwise.} \end{cases} \\ C_{over}(h,i) &= 0 \\ \boldsymbol{\rho}_g(h) &= \log P_{term}(X_k) + 0.5 \log P_{TP}(X_k) \end{aligned} \tag{6.6}$$

3. Translation hypothesis:
   If the time and space distances between the last blob of tracklets X_{k_1} and the first blob of X_{k_2} are less than the thresholds, $X_{k_1} \rightarrow X_{k_2}$ is a candidate of a tracklet translation. The h-th rows of C_g, $\boldsymbol{\rho}_g$ and C_{over} are determined as

$$\begin{aligned} C_g(h,i) &= \begin{cases} 1, \text{if } i = k_1 \text{ or } i = N_{\mathbf{X}} + k_2 \\ 0, \text{otherwise.} \end{cases} \\ C_{over}(h,i) &= \begin{cases} 1, & \text{if } i = k_1 \text{ or } i = m_r,\ \ m_r \in \Omega(X_{k_1}) \\ 0, & \text{otherwise.} \end{cases} \\ \boldsymbol{\rho}_g(h) &= \log P_{link}(X_{k_2}|X_{k_1}) \\ &+ 0.5 \log P_{TP}(X_{k_1}) + 0.5 \log P_{TP}(X_{k_2}) \end{aligned} \tag{6.7}$$

4. Dividing hypothesis:
   If the last blob of a tracklet X_p is near a birth event detected by a mitosis detection module, the tracklet is a candidate of the parent tracklet, and if the first blobs of some other tracklets X_{c1}, X_{c2} are near the candidate parent tracklet, these tracklets are candidates of the children tracklets. The h-th rows of C_g, $\boldsymbol{\rho}_g$ and C_{over} are

determined as:

$$C_g(h,i) = \begin{cases} 1, \text{ if } i = p \text{ or } i = N_{\mathbf{X}} + c_1, \\ \qquad \text{or } i = N_{\mathbf{X}} + c_2 \\ 0, \text{ otherwise.} \end{cases}$$

$$C_{over}(h,i) = \begin{cases} 1, & \text{if } i = p \text{ or } i = m_r, \ \ m_r \in \Omega(X_p) \\ 0, & \text{otherwise.} \end{cases}$$

$$\begin{aligned} \boldsymbol{\rho}_g(h) &= \log P_{div}(X_{c_1}, X_{c_2}|X_p) + 0.5 \log P_{TP}(X_p) \\ &+ 0.5 \log P_{TP}(X_{c_1}) + 0.5 \log P_{TP}(X_{c_2}) \end{aligned} \tag{6.8}$$

5. False positive hypothesis:
   All the tracklets can be false positive. When X_k is a candidate of a false positive on hypothesis h, the h-th rows of C_g, $\boldsymbol{\rho}_g$ and C_{over} are determined as:

$$\begin{aligned} C_g(h,i) &= \begin{cases} 1, \text{ if } i = k \text{ or } i = N_{\mathbf{X}} + k \\ 0, \text{ otherwise.} \end{cases} \\ C_{over}(h,i) &= 0 \\ \boldsymbol{\rho}_g(h) &= \log P_{FP}(X_k) \end{aligned} \tag{6.9}$$

A true positive tracklet appears in two and only two associations in the optimal solution: the first blob of the tracklet appears in an initialization, translation, or dividing hypothesis, and the last blob of the tracklet appears in a translation, dividing, or termination hypothesis. Thus, $\log P_{TP}(X_k)$ in the second term of Eq. 6.4 is divided into two halves that are integrated into the two neighboring transition hypotheses, as described in hypotheses 1-4. For the same reason, the non-overlapping constraint matrix C_{over} has a non-zero value for the one-side only; for example, the non-overlapping constraints of X_{k_1} are only represented in the h-th row of C_{over} for the h-th hypothesis $X_{k_1} \rightarrow X_{k_2}$.

After generating score vector $\boldsymbol{\rho}_g$ and constraints C_g, C_{over} over $N_{\mathbf{X}}$ tracklets, the MAP problem can be considered for selecting a subset of hypotheses, such that the sum of the corresponding elements in $\boldsymbol{\rho}_g$ is maximized, under the constraints in which no tree can overlap and no tracklets are in conflict. This can be formulated as the following binary optimization problem:

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}} \boldsymbol{\rho}_g^T \boldsymbol{x}, \quad \text{s.t. } C_g^T \boldsymbol{x} = \mathbf{1}, C_{over}^T \boldsymbol{x} \leq \mathbf{1} \tag{6.10}$$

where $\boldsymbol{x}$ is a $M \times 1$ binary vector, and $x_k = 1$ means that the kth hypothesis is selected in the global optimal solution. The constraint $C_g^T \boldsymbol{x} = \mathbf{1}$ guarantees that each tracklet ID appears in only one associated tree or false positive tracklet. The non-overlapping constraint $C_{over}^T \boldsymbol{x} \leq \mathbf{1}$ guarantees that selected tracklets do not overlap.

6.2.4 Implementation details

In this section, I describe the estimation of the probabilities in my framework. The difference from the method proposed in Chapter 3 is the computing manner of $P_{TP}(X_i)$ and $P_{FP}(X_i)$ where $X_i = \{A_{i_j} | j = 1, ..., \alpha\}$. These terms are defined as

$$P_{TP}(X_i) = \frac{1}{2} \sum_{j=1,...,\alpha} (P_{shape}(A_{i_j})) + \frac{1}{2} \sum_{j=1,...,\alpha-1} (P_{mig}(A_{i_j}, A_{i_{j+1}})) \quad (6.11)$$

$$P_{FP}(X_i) = \frac{1}{2} \sum_{j=1,...,\alpha} (1 - P_{shape}(A_{i_j})) + \frac{1}{2} \sum_{j=1,...,\alpha-1} (1 - P_{mig}(A_{i_j}, A_{i_{j+1}})) \quad (6.12)$$

where $P_{shape}(A_{i_j})$ is a shape score of candidate region A_{i_j} and $P_{mig}(A_{i_j}, A_{i_{j+1}})$ is a translation score from candidate region A_{i_j} to A_{i_j+1} as defined above. The term$P_{TP}(X_i)$ indicates the score of how much the tracklet looks like a true cell-trajectory, which is calculated by the average of the sum of the shape scores and the sum of the translation scores. The term $P_{FP}(X_i)$ indicates the score of how much the tracklet looks like a false positive. Here, $P_{TP}(X_i)$ is not normalized in the 0 to 1 range since the number of short tracklets tend to be selected in optimal solutions if it is normalized. Thus, these terms are normalized in the range of 0 to the length of each tracklet.

The initialization probability is defined based on the time distance between the beginning of the sequence and the first appearance frame of the tracklet, or the spatial distance between the boundary of the field of view and the cell centroid for the cell entering case.

$$P_{ini}(X_i) = \begin{cases} e^{-\frac{dt_0(R_{i_0})}{\lambda_1}}, & \text{if } dt_0(R_{i_0}) < \theta_t \\ e^{-\frac{ds(R_{i_0})}{\lambda_2}}, & \text{if } ds(R_{i_0}) < \theta_s \\ \xi & \text{otherwise } (\xi \text{ is small}) \end{cases} \quad (6.13)$$

where R_{i_0} is the first detection response of tracklet X_i, $dt_0(R_i)$ is the time distance between the first frame of the sequence and the frame when the detection response R_i appears, $ds(R_i)$ is the distance between the centroid of R_i and the image boundary, and λ_1 and λ_2 are free parameters to adjust the distribution. If the first detection response of the tracklet appears at both the beginning of the sequence and near the boundary, I take the maximum one as the probability.

The termination probability is defined similarly to the initialization probability.

$$P_{term}(X_i) = \begin{cases} e^{-\frac{dt_{end}(R_{i_{end}})}{\lambda_1}}, & \text{if } dt_{end}(R_{i_{end}}) < \theta_t \\ e^{-\frac{ds(R_{i_{end}})}{\lambda_2}}, & \text{if } ds(R_{i_{end}}) < \theta_s \\ \xi, & \text{otherwise } (\xi \text{ is small}) \end{cases} \quad (6.14)$$

where $R_{i_{end}}$ is the last detection response of tracklet X_i, and $dt_{end}(R_i)$ is the time distance between the last frame of the sequence and the frame when R_i appears.

The link probability between two tracklets and the dividing probability that one tracklet divides into two tracklets are defined as

$$P_{link}(X_j|X_i) = e^{-\|g(R_{j_0})-g(R_{i_{end}})\|)/\lambda_3} \tag{6.15}$$

$$P_{div}(X_{c1}, X_{c2}|X_p) = e^{-(\|g(R_{p_{end}})-g(R_{c1_0})\|+\|g(R_{p_{end}})-g(R_{c2_0})\|)/2\lambda_3} \tag{6.16}$$

where $g(\cdot)$ computes an object's feature vector in which different types of features can be incorporated such as appearance time and motion history. The term λ_3 is a free parameter to adjust the distribution. Based on the cell movement history, I set these parameter as: $K = 100$, $\theta_{min} = 30$, $\theta_{max} = 2000$, $\lambda_1 = 5$, $\lambda_2 = 30$, $\lambda_3 = 25$, $\theta_d = 0.5$, $\theta_t = 3$, and $\theta_s = 30$.

To reduce computation cost, I divided the entire sequence into several sequences then applied the method proposed in this chapter for each divided sequence iteratively. After obtaining all the results, I combined the results to obtain whole trajectories.

6.3 Experimental results

6.3.1 Data

I tested a DIC microscopy image sequence of human CNS stem cell populations, which was captured over 8 hours using a 12-bit Orea ER (Hamamatsu) CCD camera mounted on a Zeiss Axiovert 135 TV microscope with a 40x, 1.3 NA oil-immersion DIC objective. A 0.6x lens was installed in front of the camera to increase the visual field. The image size was 512×512 pixels. The sequence was captured every 5 minutes for 100 frames (8.33 hours). The cell population varied in the range of 120 to 150 cells per frame, and four or more cells often overlapped. The long radius of the cell ranged from 10 to 25 pixels. Preconditioning images prepared by Li and Kanade [Li09] were used before the detection step instead of directly using DIC images. Manual cell tracking was done for whole frames containing a total of 11883 cells, which appeared in the initial frame and their progeny.

6.3.2 Metrics

I used two quantitative criteria to assess the tracking performance: association accuracy and target effectiveness [Blackman86] as explained in Appendix A. To compute association accuracy, each target (human annotated) was assigned to a track (computer-generated) for each frame. The association accuracy was computed as the number of true positive associations divided by the number of associations in the ground-truth.

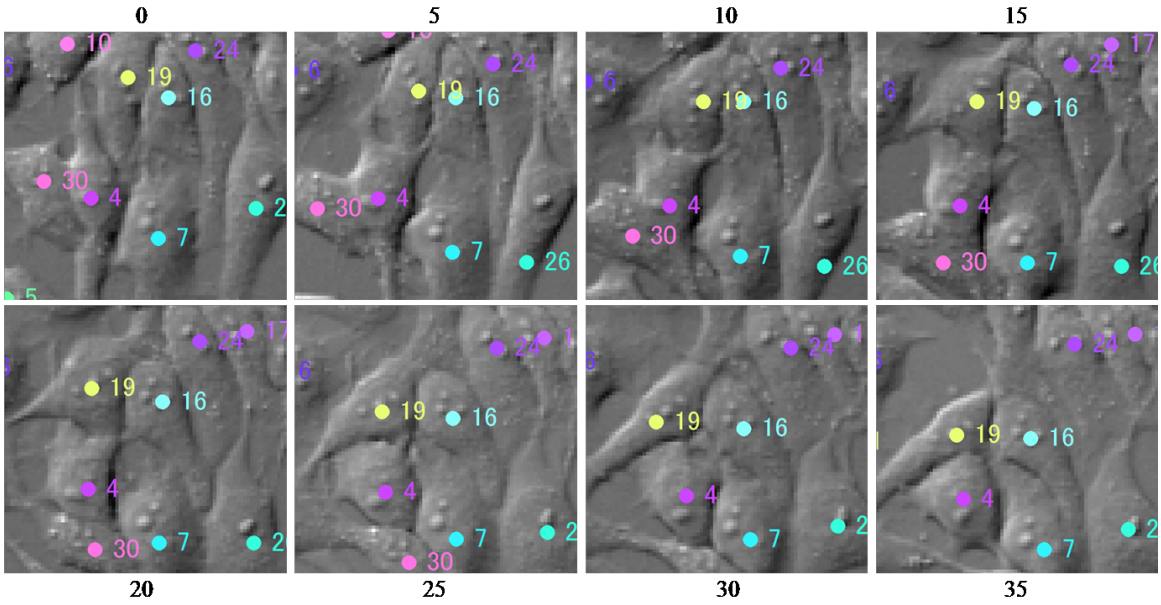


Figure 6.5: Example images of tracking results. Numbers in images are cell IDs. Number on top and at bottom of images are frame indexes. Colors of cell IDs indicate individual cells.

6.3.3 Performance evaluation

I evaluated the performance of the proposed cell-tracking method in comparison with those of six other proposed methods: the frame-by-frame association method proposed in Chapter 2, in which precondition [Li09] and Otsu thresholding [Otsu79] were used for the cell-detection step (Chapter 2); the frame-by-frame association method proposed in Chapter 2, in which the detection method proposed in Chapter 4 was used for the cell-detection step (Chapter 2 + 4); the global association method proposed in Chapter 3, in which precondition [Li09] and Otsu thresholding [Otsu79] were used for the cell-detection step (Chapter 3); the global association method proposed in Chapter 3, in which the detection method proposed in Chapter 4 was used for the cell-detection step (Chapter 3 + 4); the method proposed in Chapter 5 without global association (Chapter 5); the method proposed in Chapter 5 with global association (Chapter 5 + 3);

Figure 6.5 shows an example image of cell tracking with track IDs. In this sequence, four or more cells often touch and form clusters with blurry intercellular boundaries. Despite such conditions, the cells were detected and tracked over all the frames. Figure 6.6 shows example images of each comparison method. From the tracking results of the methods proposed in Chapters 2 and 3 (Figure 6.6 (b),(d)), several cells were not tracked due to miss-detection. For these methods using improved detection (Figure 6.6 (c),(e)), the tracking results also

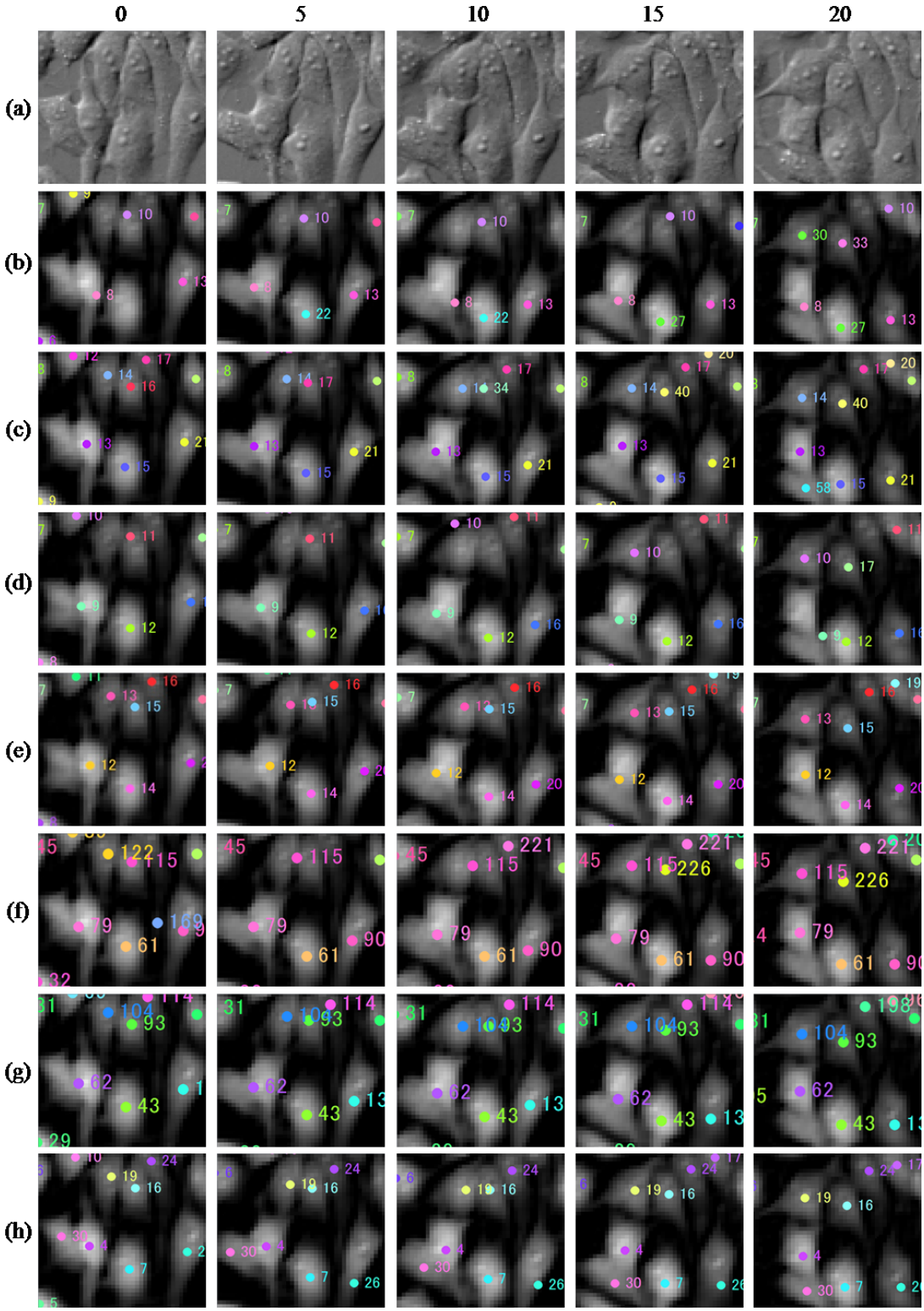


Figure 6.6: Comparison of the proposed method with other methods: (a) original image and tracking results from (b) frame-by-frame association (Chapter 2), (c) frame-by-frame association with improved detection (Chapter 2+4), (d) global data association (Chapter 3), (e) global data association with improved detection (Chapter 3+4), (f) jointly solving detection and association (Chapter 5), (g) jointly solving detection and association with global association (Chapter 5+3), (h) jointly solving tracklet selection and global association.

	Association Accuracy	Target Effectiveness
(Chapter 2)	0.757	0.427
(Chapter 2 + 4)	0.881	0.601
(Chapter 3)	0.799	0.593
(Chapter 3 + 4)	0.863	0.587
(Chapter 5)	0.910	0.650
(Chapter 5 + 3)	0.911	0.636
Proposed	**0.945**	**0.683**

Table 6.1: Comparison of proposed method with six other proposed method.

improved. From the tracking results of the method proposed in Chapter 5 (Figure 6.6 (f),(g)), the tracking results were better than those of the methods proposed in Chapters 2 and 3. However, a cell was still not tracked. From the tracking results of the method proposed in this chapter (Figure 6.6 (h)), all cells were well tracked.

Table 6.1 shows tracking performances of each comparison methods. The tracking method proposed in this chapter exhibited the best performance for two metrics.

6.4 Conclusion

I proposed a tracking method that first generates redundant candidate tracklets, then solves the joint problem of optimal tracklets selection and global association. The method generates reliable tracklets by introducing constraints that preserve three structures of detection candidate between successive frames for association. Then the problem of both selecting an optimal set of cell tracklets from the redundant tracklets and associating the tracklates over frames under non-overlapping constraints is solved simultaneously.

Experimental results on a challenging data set show that the proposed tracking method improves the tracking performance including association accuracy and target effectiveness compared with the method proposed in the previous chapters.

Chapter 7

3D Cell Tracking using Global Motion and Local Spatial Relationships

7.1 Introduction

In the previous chapters, I assumed that cell movement in successive frames is small under dense culture conditions since the motion of a cell is constrained by its surrounding cells. This is true when a small area in 2D is observed and analyzed. However, the increment in time-lapse imaging cannot be shortened to monitor a wide area, which makes the problem even more challenging. In 2D images for example, it takes time to obtain a large image covering a wide area at a high resolution where the system captures multiple images that are then merged to generate a larger image. In 3D images, it also takes time to obtain 3D volume data for a wide range of specimens.

The main difficulty arises from two aspects of this problem. First, the movements of cells between successive frames are often larger than the distances to the nearby cells. This makes it almost impossible to associate cells between frames based on their proximity. Figure 7.1 shows an example of cells moving large distances in successive frames. The distance of corresponding cells between successive frames is often larger than that between non-corresponding cells. For instance, the distance between B^t and B^{t-1} is larger than B^t and A^{t-1}. Second, it is more likely that other cells will be found to have a similar appearance in the local region. This makes it difficult for appearance-based association methods to properly work.

In this chapter, I propose a method for tracking a large number of cells successfully under such a densely populated condition in 3D. The key observation behind this method is that nearby cells under such highly populated conditions exhibit similar motion patterns since

the motion of a cell is constrained by its surrounding cells. This leaves the relative position of cells largely unaltered. The proposed method exploits this observation by introducing the following two stages.

- *Global motion estimation:* First, the global motion of cells is estimated using non-rigid alignment between successive frames. The global motion can be seen as a flow field. Therefore, the global motion of cells does not provide the exact motion of each cell, but rather provides a rough estimate of a cell's position in the next frame that is then used in the following stage of target association. To the best of my knowledge, this is the first attempt at utilizing non-rigid alignment to estimate the global motion of multiple target tracking.

- *Multiple cell association by using the local pairwise spatial relationships:* Second, all the cells are individually tracked by using the detection-and-association approach, where pairwise spatial relationships are taken into account to evaluate the association scores. An advantage of introducing the pairwise spatial relationships is to keep the relative positions of the cells among the successive frames. In addition, the method identifies new tracks and connects them with its mother cells to deal with any cell division.

The proposed method was evaluated and compared with other recent tracking methods [Kofahi06][Xiao10] based on the tracking accuracy using synthetic 3D point-set data. Since a cell can be divided into two cells, both the correctness of the movement (1-to-1) association and the division (1-to-2) association are evaluated. The results obtained from the evaluation indicate that both stages (global motion estimation and target association using pairwise spatial relationships) contribute to a higher level of accuracy in tracking than the existing methods. I also introduce the application of the proposed method to the field of biological research to analyze the somite formation process of a zebrafish embryo.

7.2 Overview of proposed method

The proposed method consists of three stages: 1) Cell detection to detect and locate the positions of cells in each frame. 2) Global motion estimation to estimate the global motion of cells by using non-rigid alignment in 3D between successive frames. The estimated global motion is used to provide the estimated position of each cell in the next stage. 3) Multiple cell association to find the optimal association among all the possible cell movement hypotheses between successive frames by using the local pairwise spatial relationships from the neighbors. The method overview is shown in Figure 7.2.

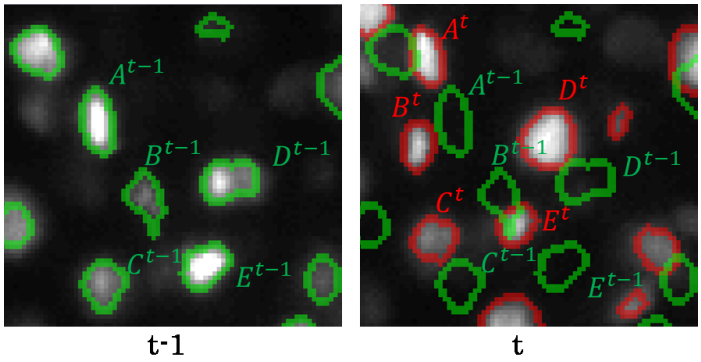


Figure 7.1: Examples of cells moving large distance in successive frames. The left image shows cell regions in a z-slice of 3D data in the previous frame. The right image shows overlaid cell regions of the previous and current frames where red indicates the contour in the current frame and green indicates that in the previous frame. The increment in time-lapse is 3 minutes.

In the cell-detection stage, I have adopted cell-detection method proposed in Chapter 4 that can detect the cell regions in 3D volume data. The method first detects many of the cell candidates with possible overlaps among them, and then solves an optimization problem to find the optimal cell-detection set within the candidate set based on the non-overlapping constraints. Blob regions are obtained on each frame when using this method. The set of blob regions at t is denoted as $\mathbf{R}_t = \{R_m^t, m = 1, 2, ..., M_t\}$.

The details of the other two stages, which is the main contribution of this chapter, are explained in the following sections.

7.3 Global motion estimation

In this stage, I estimate the global motion of cells based on the key observation that nearby cells under highly populated conditions exhibit similar motion patterns. This is done by non-rigid alignment, which aligns the structure of the target objects among multiple data. Non-rigid alignment has been successfully used in medical image analysis, including the registration, matching, and shape tracking of a single object [Mcinerney96]. Unlike in previous works, I use non-rigid alignment for the global motion estimation in multiple object tracking. A crowd of cells under dense cell culture conditions cluster, and the form of the crowd changes over time. In this process, even though the cells individually migrate, the relative positions of the cells are largely unaltered since the motion of the cells is constrained by the surrounding cells. For example, a cell near the boundary of a crowd usually stays near the boundary in the next time frame. As stated in the introduction, one of the novelties of the proposed method is that I directly align the crowd of cells between successive frames

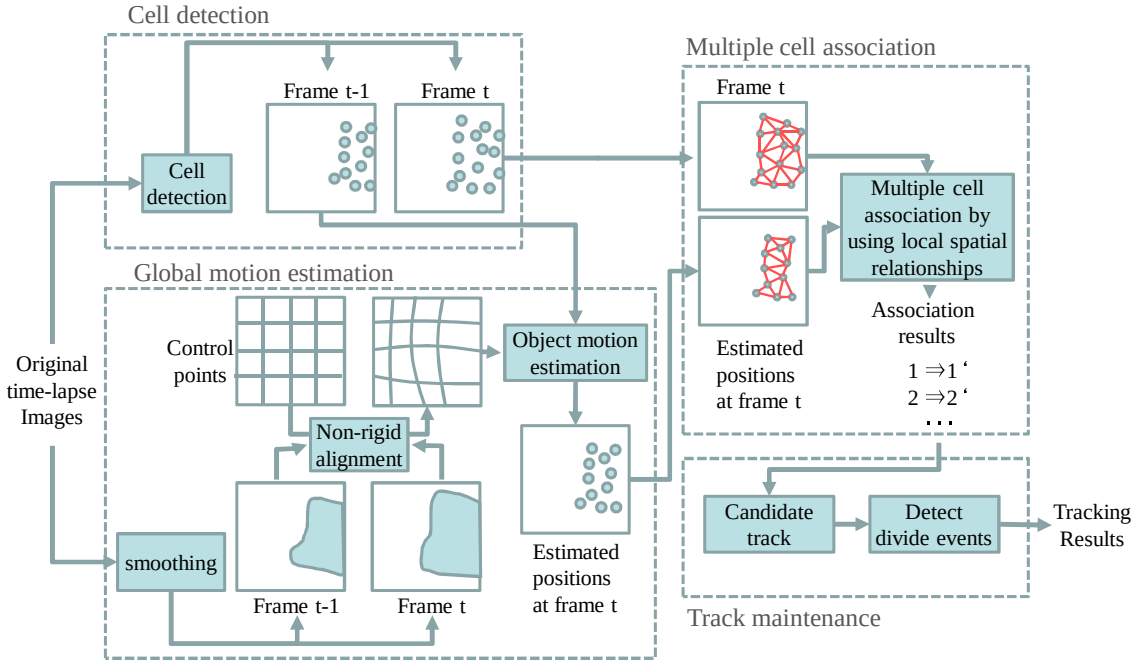


Figure 7.2: Method overview. The proposed method consists of three stages, cell-detection, global motion estimation, and multiple cell association. I must note that this is illustrated using 2D images for clarity purposes, but all the steps are conducted within a 3D volume constructed from multiple 2D layers.

using non-rigid alignment to estimate the motion of each cell before multi-target association.

Here, the data from my research target is a z-stack of 2D images. I treat the z-stack images as 3D volume data (i.e., one pixel in a z-stack image is treated as a voxel in 3D) in order to use 3D alignment. I use B-spline Free-Form Deformation (FFD) [Rueckert99] for 3D alignment, which has been successfully used for the automatic image registration of three-dimensional breast MRI images. Spline-based FFD deforms an image to minimize the similarity function between the deformed image and the target image by manipulating the underlying mesh of control points. The optimal transformation of the control points is found by using the steepest descent optimization of a similarity function between the input and target images. This process is illustrated in the middle of Figure 7.2. For the sake of self-completeness, I briefly explain the deformation method.

Let Φ be a $n_x \times n_y \times n_z$ 3-D mesh of control points $\phi_{i,j,k}$ whose initial positions are uniformly spaced by δ, and $T(x, y, z|\Phi)$ be a non-rigid transformation function where any

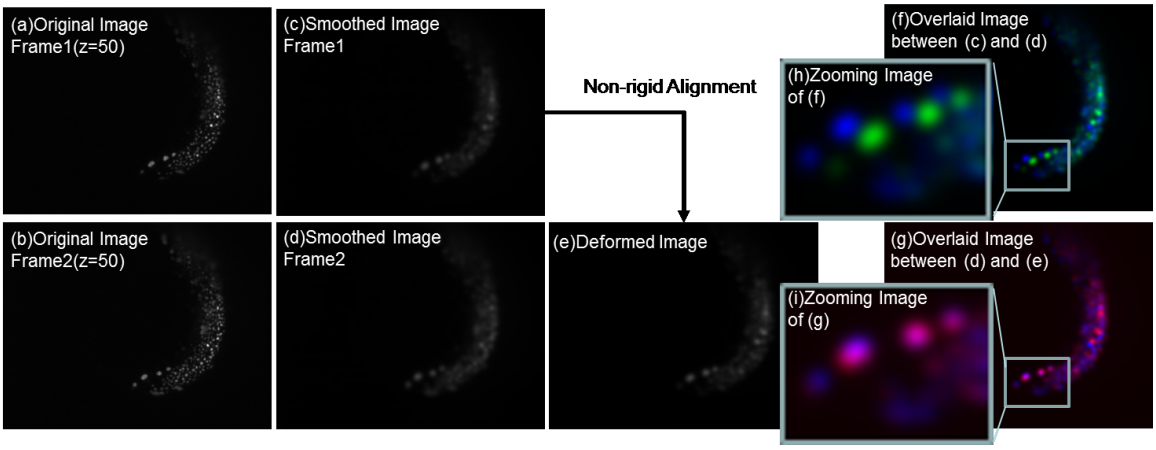


Figure 7.3: Examples of alignment results of 3D volume data. The images are on the 3D z-slice. (a) Original image in the previous frame on z-slice (z=50). (b) Original image in the next frame (z=50). (c) Smoothed image of (a). (d) Smoothed image of (b). (e) Deformed image in which (c) is aligned with (d). (f) Overlaid image of (c) and (d), where blue represent image (d) and green is (c). (g) Overlaid image of (d) and (e), where blue represents (d) and red is (e). (h) Zooming image of (f). (i) Zooming image of (g).

points (x, y, z) are transformed into its corresponding point as

$$\begin{aligned} & T(x, y, z|\Phi) \\ &= \sum_{l=0}^{3}\sum_{m=0}^{3}\sum_{n=0}^{3} B_l(u)B_m(v)B_n(w)\phi_{i+l,j+m,k+n} \end{aligned} \tag{7.1}$$

where $i = \lfloor x/n_x \rfloor, j = \lfloor y/n_y \rfloor - 1, k = \lfloor z/n_z \rfloor - 1, u = x/n_x - \lfloor x/n_x \rfloor, v = y/n_y - \lfloor y/n_y \rfloor, w = z/n_z - \lfloor z/n_z \rfloor$, and B_l represents the lth basis function of the B-spline [Lee97]. The problem of finding the optimal transformation of the control points is formulated using non-rigid transformation as follows:

$$\Phi^* = \arg\min_{\Phi} -C_{sim}(I(X, t), I(T(X|\Phi), t-1)) \tag{7.2}$$

where X is a list of all points in the 3D volume, and $I(T(X|\Phi), t-1)$ is a transformed image from volume data $I(t-1)$ by T. C_{sim} is the similarity measure between two sets of volume data $I(t)$ and $I(T(X|\Phi), t-1)$ such as the sum of squared distance (SSD) or the normalized mutual information (NMI). Eq. (2) is solved by using steepest descent optimization. The advantage of this method is that the displacement of a control point affects only the translation in the local neighborhood of the control point, making the method computationally more efficient than the other methods. Please refer to [Rueckert99] for further explanations.

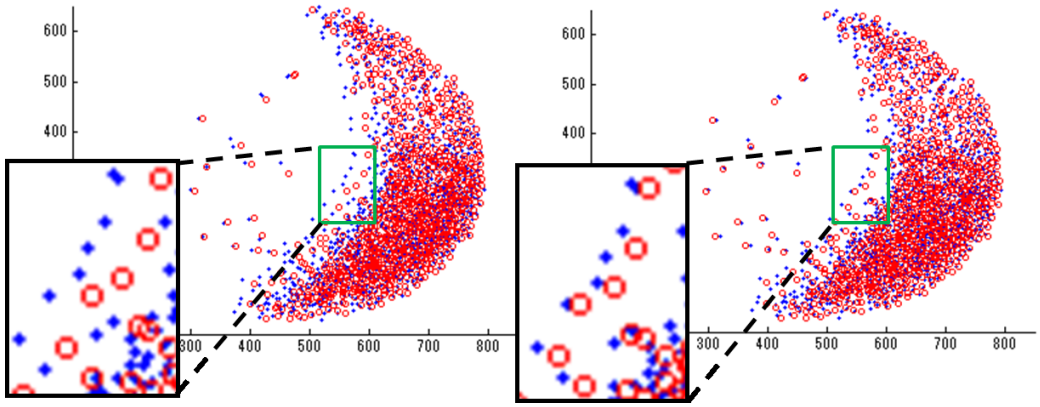


Figure 7.4: Examples of results for estimating positions by using results from alignment. The graph at the left indicates the positions of objects in two consecutive frames. Red represents the positions in the previous frame, and blue is those in the next frame. The graph at the right represents the estimated positions (red) and those in the next frame (blue).

Unlike for alignment of 3D MRI images, special care has to be taken to deal with two issues in my case. First, the intensity of a cell may fluctuate due to the change of fluorescent substances in the cell over time (e.g., some cells suddenly become much brighter). Such cells introduce errors in non-rigid alignment. Second, cells may enter or leave the field of view, and therefore some cells near the boundary of the field of view may not have corresponding cells between two frames. This also results in errors in non-rigid alignment.

These issues are handled in my method in the following way. First, the volume data is smoothed by using a 3D Gaussian filter to relive the problem of local minima in non-rigid alignment. Examples of the z-slice images of the smoothed 3D volume data are shown in Figures 2(c) and (d). Non-rigid alignment is then applied to the original volume data for fine adjustment by using the resulting positions from alignment of the smoothed volume data as the initial positions. In addition, a coarse-to-fine approach is used for better computational efficiency. The largess of the spatial interval δ of the control points is used first, and then the smaller one is iteratively used just as in [Lee97].

Second, the boundary area of the 3D volume data is down-weighted in non-rigid alignment to reduce any negative effects from the entering or existing cells. More specifically, the following exponential decay function is multiplied to the volume data

$$f(k) = \begin{cases} e^{-\frac{K-k}{\lambda}}, \text{ if } k \leq K \\ 1, \quad \text{otherwise.} \end{cases}, k \geq 0 \tag{7.3}$$

where k is the distance from the field of the view, K is a parameter that is the width of the boundary area to be multiplied, and λ is the parameter of the decay constant.

Figure 7.3 shows some examples of alignment of the 3D volume data. Only 2D slices at a certain depth are shown for visualization purposes. After smoothing, the smoothed data is deformed by the non-rigid alignment, and the result is shown in Figure 7.3(e). The overlaid images between the next image (blue) and the previous image (green) are shown in Figure 7.3(f), and that between the next image (blue) and the deformed image (red) are shown in Figure 7.3(g) to show how the alignment works for real data. The green and blue regions are not overlaid in Figure 7.3 (f). This indicates that the cells are in different positions in the two successive frames. On the other hand, there are many purple regions (i.e., blue and red are overlaid) in Figure 7.3 (g), showing that the images are roughly aligned. The sum of squared differences between two frames is reduced by 50% using non-rigid alignment.

The position of each cell in the next frame is approximately estimated by using the optimal deformation results. Figure 7.4 shows the estimated positions using the results from non-rigid alignment shown in Figure 7.3. These results indicate that the proposed method provided a better estimation than the original positions. Based on the estimated positions, the cells among successive frames are associated on the next stage.

7.4 Multiple cell association by using local pairwise spatial relationship

Multiple target association approaches first segment and locate the targets at each frame, and then find the optimal association among all the possible cell movement hypotheses where a cell in the previous frame moves to a position of a detected cell in the next frame. This section explains how multiple target association is conducted using the proposed method.

The set of tracked cells at frame $t-1$ is denoted as $\mathbf{C}_{t-1} = \{\boldsymbol{c}_l^{t-1}, l = 1, ..., N_{t-1}\}$ where $\boldsymbol{c}_l^{t-1}$ stores each cell's state, which contains its centroid and region. For each frame, a detected blob R_m^t in the next frame t is associated with a cell $\boldsymbol{c}_l^{t-1}$. Each cell has its parent's index Parent-ID for maintaining its lineage information and its state in each frame. Parent-ID is 0 for cells with no parent. If a blob is not associated with any cell, the blob is removed as noise. If a cell is not associated with any blob, the track of the cell is terminated as a lost track. Here, I denote an estimated position of $\boldsymbol{c}_l^{t-1}$ on the next frame t from the previous stage as $T_{t-1}(\boldsymbol{c}_l^{t-1})$.

In the proposed method, the following processes are iterated for each pair of two successive frames to obtain tracks.

- *Listing all possible cell movement hypotheses:*
  All possible cell movement hypotheses are listed. Since a cell usually cannot move a great distance, if the distance between an estimated position $T_{t-1}(\boldsymbol{c}_l^{t-1})$ and a blob's position R_m^t is smaller than a threshold τ, a movement hypothesis $\boldsymbol{c}_l^{t-1} \rightarrow R_m^t$ is

created. The hypotheses set is denoted as $\mathbf{H}_t = \{\boldsymbol{h}_k^t, k = 1, 2, ..., H_t\}$ where H_t is the number of hypotheses in frame t.

- *Computing association scores of hypotheses*
  For each hypothesis, the association score of the hypothesis is computed by incorporating the local pairwise spatial relationship similarity.

- *Solving optimal association*
  The cell association problem is formulated as binary linear programming. I generate an association score vector and a constraint matrix to accomplish this. The constraint matrix guarantees that a cell cannot move to or come from different multiple places at the same time. The association score vector and the constraint matrix are then used to find the optimal association by using binary linear programming.

- *Track maintenance*
  A new track that suddenly appears in the sequence is identified by using temporal information. Such a track is regarded as a divided cell and assigned to its mother cell.

The following sections explain each step in details.

7.4.1 Compute association scores of hypotheses

An association score for each hypothesis is computed in this step. Unlike when using the existing cell-tracking methods that use only the target similarity to compute association scores, I use the local pairwise spatial relationships to compute the association score of a movement hypothesis. As a result, as many of the spatial relationships among cells between successive frames as possible are preserved.

The pairwise spatial relationships are incorporated into the association score computation by using a graph similarity measure in hyper graph matching. Such a measure has previously been successfully used for the image matching problem[Cour06][Zass08][Lee11]. Hyper graph matching is used to find a mapping between the two edges, and it is able to preserve as many of the spatial relationships between nodes as possible. However, hyper graph matching often becomes expensive in both memory usage and computational cost in comparison to vertices matching since hyper graph matching takes into consideration all of the hypotheses of edge matching. This problem becomes particularly evident in my case of dense cell tracking where the number of targets is significantly larger than that of the interest points used for typical image matching. I use the local pairwise spatial similarity among neighbor cells rather than the global one since the global spatial structure is mostly preserved by the previous global motion estimation stage to reduce the computational cost.

As discussed above, the local pairwise spatial relationships are taken into account to compute the association scores. Given hypothesis $\boldsymbol{c}_l^{t-1} \rightarrow R_m^t$, which indicates that cell

CHA

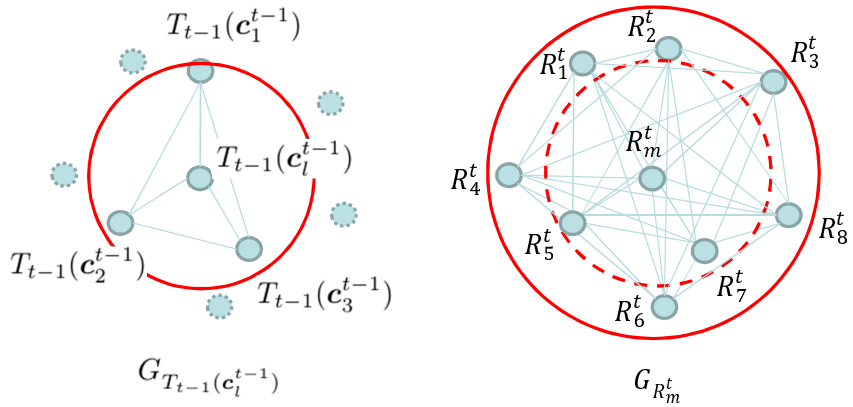


Figure 7.5: Examples of graphs $G_{T_{t-1}(\boldsymbol{c}_l^{t-1})}$ and $G_{R_m^t}$. The red circles represent the local window to generate the graph. The local window of graph $G_{R_m^t}$ is larger than that of graph $G_{T_{t-1}(\boldsymbol{c}_l^{t-1})}$.

$\boldsymbol{c}_l^{t-1}$ moves to blob R_m^t, the association score is defined as a total of a vertex (target) matching score [Kofahi06][Padfield11][Kanade11] and a partial graph matching score:

$$S(R_m^t, \boldsymbol{c}_l^{t-1}) = S_V(R_m^t, T_{t-1}(\boldsymbol{c}_l^{t-1})) + \alpha S_G(G_{R_m^t}, G_{T_{t-1}(\boldsymbol{c}_l^{t-1})}) \tag{7.4}$$

where $S_V(R_m^t, T_{t-1}(\boldsymbol{c}_l^{t-1}))$ is the vertex matching score, which indicates the similarity measure between targets $\boldsymbol{c}_l^{t-1}$ and R_m^t. The vertex matching score is calculated by using $exp(-d_{lm}^2/\sigma)$, where d_{lm} is the distance between $T_{t-1}(\boldsymbol{c}_l^{t-1})$ and R_m^t, and σ is the control parameter. $S_G(G_{R_m^t}, G_{T_{t-1}(\boldsymbol{c}_l^{t-1})})$ is the partial graph matching score, which represents the similarity measure between the $G_{T_{t-1}(\boldsymbol{c}_l^{t-1})}$ and $G_{R_m^t}$ graphs, where $G_{\boldsymbol{c}}$ is the local graph consisting of the neighboring cells of cell $\boldsymbol{c}$. The relative weight α of the two scores is automatically determined, so the average of $S_V(R_m^t, T_{t-1}(\boldsymbol{c}_l^{t-1}))$ is equal to the average of $\alpha S_G(G_{R_m^t}, G_{T_{t-1}(\boldsymbol{c}_l^{t-1})})$.

A local spatial window is used to generate the local graph $G_{T_{t-1}(\boldsymbol{c}_l^{t-1})} = \{V, E\}$. First, all the estimated positions of the cells whose distance from $T_{t-1}(\boldsymbol{c}_l^{t-1})$ is smaller than the threshold κ_1 are registered in V. Then, all the pairwise combinations of V are registered in E. Graph $G_{R_m^t} = \{V', E'\}$ is generated in the same way using a threshold κ_2 that is larger than κ_1. Note that two different thresholds are used for creating the two local graphs. A cell near the boundary of the local spatial window of a target cell may move away from $T_{t-1}(\boldsymbol{c}_l^{t-1})$ in the next frame. As a result, the corresponding cell may not be found in $G_{R_m^t}^t$ if the size of local window is the same between successive frames. An example is shown in Figure 7.5, where node R_2^t, which corresponds to node $T_{t-1}(\boldsymbol{c}_1^{t-1})$, is outside the local window. The size of $G_{R_m^t}$ is set to be larger than that of $G_{T_{t-1}(\boldsymbol{c}_l^{t-1})}$ by using two different

κ_1 and κ_2 thresholds to avoid this problem. κ_2 is set to be $1.5\kappa_1$ in my experiments.

After generating local graphs, the partial graph matching score $S_G(G_{R_m^t}, G_{T_{t-1}(\boldsymbol{c}_l^{t-1})})$ is computed by maximizing the sum of the edge distance similarities of the corresponding edges among the graphs. This is given as

$$S_G(G_{R_m^t}, G_{T_{t-1}(\boldsymbol{c}_l^{t-1})}) = \max \sum_{\{ij,i'j'\} \in (G_{R_m^t}, G_{T_{t-1}(\boldsymbol{c}_l^{t-1})})} S_E(E'_{i'j'}, E_{ij}) \tag{7.5}$$

$$S_E(E'_{i'j'}, E_{ij}) = exp(-|d_{ij} - d'_{i'j'}|^2) \tag{7.6}$$

where $\{ij, i'j'\} \in (G, G')$ is a set of any corresponding edges in the graphs. $S_E(E'_{i'j'}, E_{ij})$ is the edge similarity between $E'_{i'j'}$ and E_{ij}, where d_{ij} and $d'_{i'j'}$ are the lengths of edges E_{ij} and $E'_{i'j'}$ respectively. Eq. (5) is used to attain the optimal partial graph matching score between the smaller graph $G_{T_{t-1}(\boldsymbol{c}_l^{t-1})}$ and the optimal partial graph in the larger graph $G_{R_m^t}$. For example, in Figure 7.5, the node of the optimal partial graph in $G_{R_m^t}$ is $V' = \{R_m^t, R_2^t, R_5^t, R_7^t\}$ corresponding to the graph $G_{T_{t-1}(\boldsymbol{c}_l^{t-1})}$ as $V = \{T_{t-1}(\boldsymbol{c}_l^{t-1}), T_{t-1}(\boldsymbol{c}_1^{t-1}), T_{t-1}(\boldsymbol{c}_2^{t-1}), T_{t-1}(\boldsymbol{c}_3^{t-1})\}$.

This partial graph matching problem can be solved by using any existing graph matching method[Zass08][Lee11]. It can also be solved by conducting a full search as well since the number of edges are limited, and one of the matching ($\boldsymbol{c}_l^{t-1}$ matches with R_m^t) is known in my case. The computed association scores of the movement hypotheses set are used in the next step to find the optimal association set from all the hypotheses.

7.4.2 Finding optimal association by using binary programming

In this section, I explain how the optimal association set is found from all the movement hypotheses $\mathbf{H}_t$ without conflict. The problem of optimal association is similarly formulated to that in [Li08][Kanade11] based on binary programming. I briefly explain the association method to make this paper self-contained.

The binary programming is formulated based on two inputs: a constraint matrix C and a score vector $\boldsymbol{\rho}$. The $H_t \times 1$ score vector $\boldsymbol{\rho}$ stores the association scores where the k-th row is the association score of the k-th hypothesis $\boldsymbol{h}_k^t$ computed by using the previous step. Let C be the $H_t \times (N_{t-1} + M_t)$ constraint matrix used to avoid conflicting hypotheses. Each row corresponds to a hypothesis $\boldsymbol{h}_k^t (k = 1, ..., H_t)$ in $\mathbf{H}_t$. Each of the first N_{t-1} columns indicates the cell index at $t-1$ of the hypothesis $\boldsymbol{h}_k^t$, and each of the remaining M_t columns is an index of the detected blob in the next frame t of $\boldsymbol{h}_k^t$. The constraint matrix C and the score vector $\boldsymbol{\rho}$ are constructed as follows.

For each hypothesis, a new row is appended to both the constraint matrix C and the score vector $\boldsymbol{\rho}$. Let k be the index of the movement hypothesis $\boldsymbol{h}_k^t = \{\boldsymbol{c}_l^{t-1} \rightarrow R_m^t\}$, and I

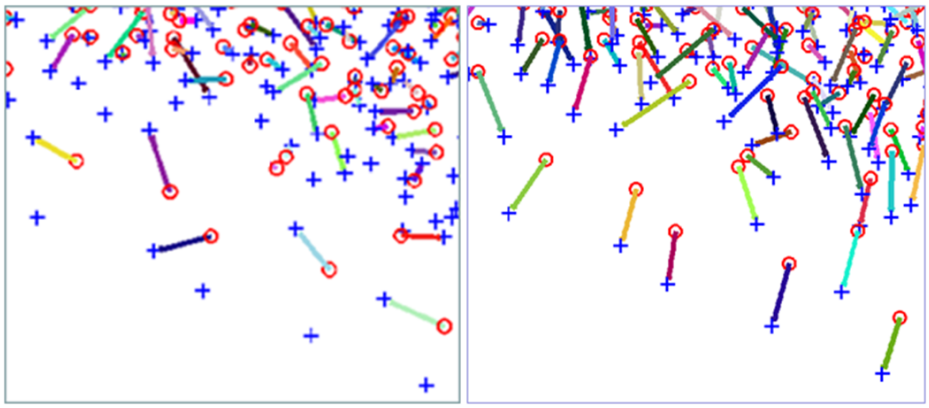


Figure 7.6: Examples of association results. The image on the left shows the results of using the multiple hypotheses association method without the pair-wise similarity. The one on the right shows the results of using the proposed method. '∘' points indicate the positions in the previous frame and '+' points indicate the positions in the next frame.

append the k-th row to C and $\boldsymbol{\rho}$:

$$C(k,i) = \begin{cases} 1, \text{ if } i = l \text{ or } i = N_{t-1} + m \\ 0, \text{ otherwise.} \end{cases} \tag{7.7}$$

$$\boldsymbol{\rho}(k) = S(R_m^t, \boldsymbol{c}_l^{t-1}) \tag{7.8}$$

After constructing C and $\boldsymbol{\rho}$ for all the possible hypotheses in $\mathbf{H}_t$, the association problem can be considered as selecting a subset of rows on $\boldsymbol{\rho}$ such that the sum of the corresponding elements in $\boldsymbol{\rho}$ is maximized under the constraint in which any cell or detected blob appears in at most one hypothesis. This can be formulated as the following binary linear programming problem:

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}} \boldsymbol{\rho}^T \boldsymbol{x}, \tag{7.9}$$
$$s.t.\ \ C^T \boldsymbol{x} \leq \mathbf{1}, \boldsymbol{x}_k \in \{0,1\}\ (k = 1, ..., H_t)$$

where $\mathbf{1}$ is a $H_t \times 1$ vector of ones. $\boldsymbol{x}$ is a $H_t \times 1$ binary vector, and $\boldsymbol{x}_k = 1$ means the k-th hypothesis is selected as the optimal solution. Constraint $C^T \boldsymbol{x} \leq \mathbf{1}$ guarantees that each cell index and blob index appears in only one selected hypothesis (i.e., the same cell index/blob index does not appear in multiple hypotheses). The problem in Eq. (7.9) can be solved since binary programming always has a binary-valued solution if the constraint matrix C is totally uni-modular (i.e., the determinant of any square submatrix takes one of the values in $-1, 0, 1$), and the right-hand sides of the constraints are all integers [Papadimitriou98].

Figure 7.6 shows the results of the association between successive frames. The cells in the results for the existing method (left image) tended to be associated with the closest

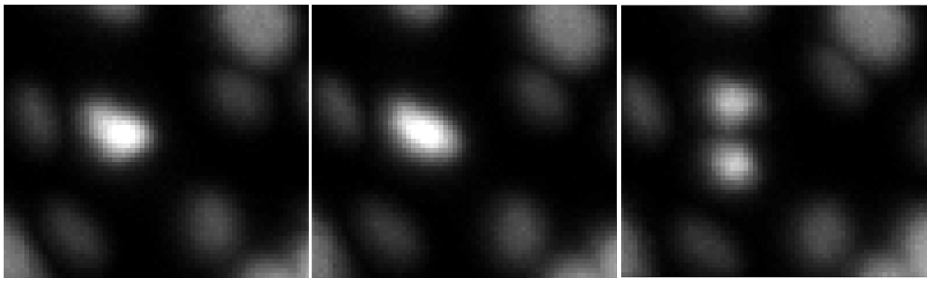


Figure 7.7: Example of a cell division event where a cell divides two cells on curve slice in 3D data. Two children cells usually appear near the mother cell.

one. In the results of the proposed method (right image), the cells tended to move in the same direction as their neighboring cells, and the structures of the relative positions are preserved after movement.

7.4.3 Track maintenance

The track maintenance step is used after the association step to deal with cell division events. An example of a cell division event is shown in Figure 7.7. I use a similar approach to that discussed in Chapter 5 to handle such division events.

The track maintenance step identifies a new track and connects the track with its mother cell. Here, I make three assumptions. 1) If a new cell suddenly appears in the field of view outside the boundary of the volume, the new cell is considered a divided cell since a cell does not appear unless it enters from outside or divides from another cell. 2) A detected noise such as a cell mistakenly detected as two blobs usually disappears within several frames. 3) The positions of the divided children cells are usually close to their mother cell. I identify a new track based on the assumptions.

First, I register a blob that is not associated with any cells in the previous frame as a candidate cell. Then, the candidate cell is continuously tracked until the length of the candidate track is longer than a threshold η or the candidate track is lost. If the length of the candidate track exceeds the threshold, the candidate cell is registered as a reliable cell, and if the candidate cell is lost before the registration, the candidate cell is removed as noise. Next, its mother cell is identified by finding a cell that minimizes the distance from the center position of the cell and the newly registered cell to the previous position of the cell. Then, the identified cell is connected with the child cell as the mother cell. Figure 7.8 illustrates an example of this step.

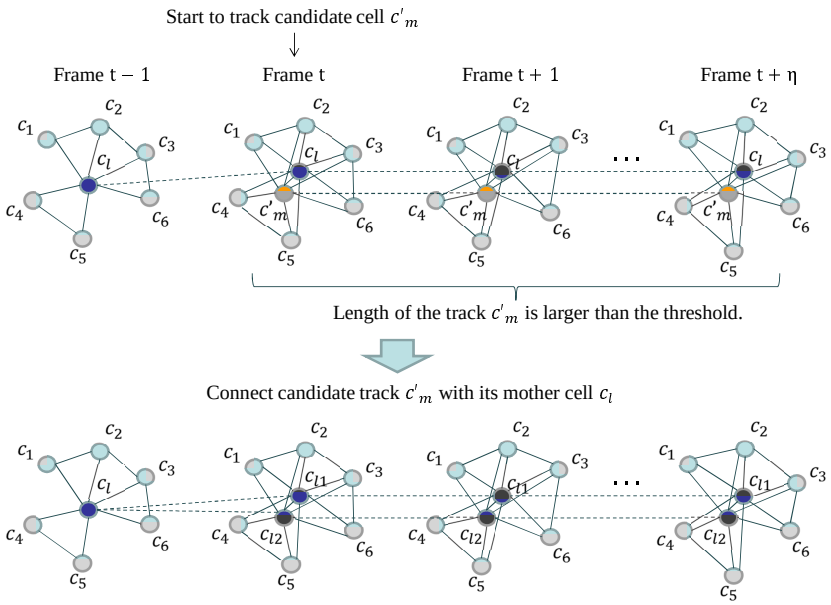


Figure 7.8: Example when a candidate cell is registered as a child cell and connected with its mother cell.

7.5 Experiments

I evaluated and compared the performance of the proposed method with the existing tracking methods by using synthetic point-set data. Both the correctness of the movement association and the division association were evaluated to evaluate the tracking accuracy. I used a detection-and-association method like in [Kanade11] and [Kofahi06] as a baseline, where the association score is computed based on the proximity without using the pairwise relationships. In addition, a state-of-the-art object tracking method that is based on graph matching to track multiple vehicles in aerial videos [Xiao10] was also used. This method incorporates edge (i.e., pairwise) matching to vertices (i.e., targets) matching based on hyper graph matching [Zass08] for better tracking accuracy. Five methods including baseline1 ([Kanade11][Kofahi06]), baseline2 ([Xiao10]), baseline1 with global motion estimation, baseline2 with global motion estimation, and the proposed method (global motion estimation + pair-wise relationship) were evaluated to show the effectiveness of both the global motion estimation and target association using the pair-wise spatial relationships, respectively.

These methods were implemented in Matlab. The code 'B-spline Grid, Image and Point based Registration' [Kron08] in Matlab Central was used for non-rigid alignment. The parameters were set as follows. The sizes of the spatial interval δ of the control points were $\{\frac{I}{2}, \frac{I}{4}, \frac{I}{8}, \frac{I}{16}\}$ where $I = \{n_x, n_y, n_z\}$ is the size of the input data, and the size of the boundary of the field of view K was 5 pixels. The window size to create movement hypotheses τ was 20

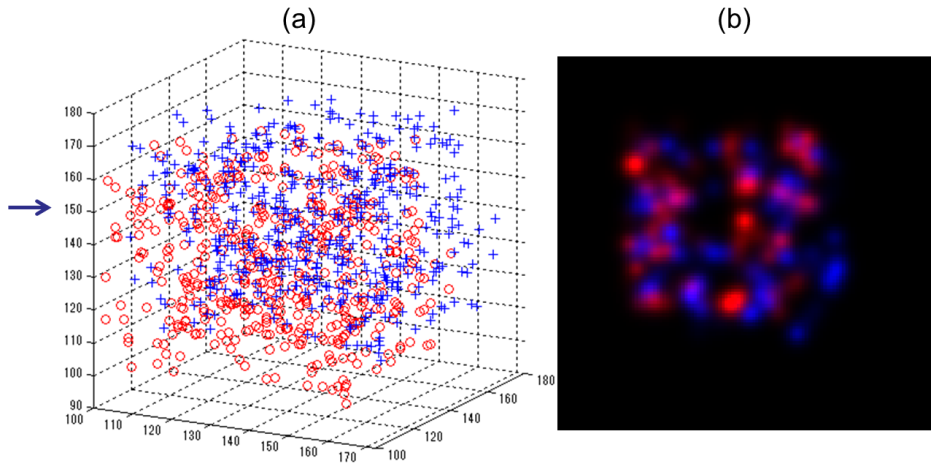


Figure 7.9: Example of evaluation data. D is 0.25. The red circles indicate the cell positions of the previous frame and the blue ones are the cell positions of the next frame.

pixels. κ_1 and κ_2 were automatically decided based on the means of the distances between neighboring cells; κ_1 was set to the mean multiplied by 1.5, $\kappa_2 = 1.5\kappa_1$. The threshold η was 4 frames. Each quantitative result in my synthetic experiments was obtained as the average of 30 random trials.

7.5.1 Data

The accuracy of the cell tracking was evaluated by using synthetic sequential point set data, because it is difficult to provide the ground truth data of dense cell tracking in 3D microscopic images. I first randomly generated $N = 500$ points as the cells using uniformly distributed pseudorandom numbers in 3D space to determine the initial positions of cells in the data. A Gaussian filter ($sigma = 2$) was added to make cells with given sizes. Then, two types of cell motions, which are the global motion and the individual motion, were added to simulate the real cell movement for each frame.

Every point was deformed by the B-spline free-form deformation using randomly decided parameters to simulate the global motion of cells that indicate the deformation of cell crowds. The free-form deformation was used four times at different sizes $\delta = \{\frac{I}{2}, \frac{I}{4}, \frac{I}{8}, \frac{I}{16}\}$. Each time all the control grid points were randomly displaced using uniformly distributed pseudorandom numbers where the max of the displacement noise was $D \times \delta$. I call the deformation noise level D here. Gaussian noise was independently added to each point to simulate the individual motion of the cells. The mean of the Gaussian noise was 0.2 times the mean of the distance between neighboring cells, and the variance was 1.

In addition, since the results of the cell-detection step usually include errors, the de-

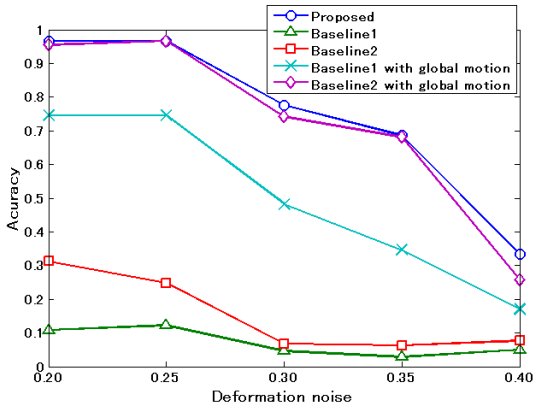


Figure 7.10: Performance according to change in deformation level. Here, $N = 500$ and $E = 0$.

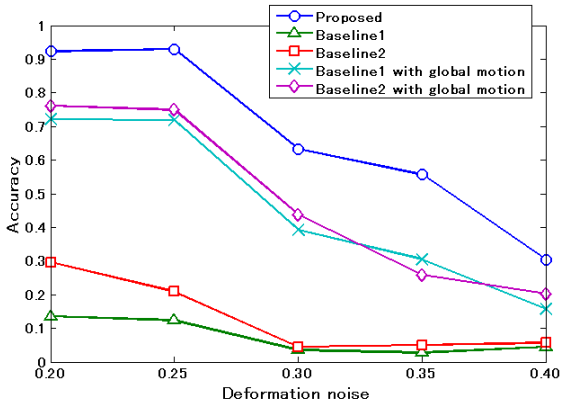


Figure 7.11: Performance according to change in deformation level. Here, $N = 500$ and $E = 0.05$.

tection errors were added. $E \times N_t$ points were removed (i.e., false negatives), and $E \times N_t$ random points were added (i.e., false positives) in each to simulate the detection errors, where the detection error E was the rate of the number of detection errors and the number of total cells. Figure 7.9 shows an example of the synthetic data. Figure 7.9(a) shows the centroids of cell positions in frames $t-1$ (red) and t (blue). Figure 7.9(b) shows a z-slice image on the 3D volume. The cell migration distance was much larger than the distance between neighboring cells.

In addition, cell division events were added to evaluate the correctness of the cell division association. Several cells were randomly selected as the mother cells of the division events for each frame to simulate the cell division events. In the next frame, the positions of two children cells were decided based on the selected mother cells for each division event. Gaussian noise was added to the position of a mother cell after adding the global motion to determine the positions of the two children. I set the number of cells in the initial frame at $N_1 = 500$, the rate of the number of cell divisions at 0.01, where the number of division cells on each frame is $0.01 \times N_t$ cells (i.e., after 30 frames, the number of cells was $N_{30} = 674$).

7.5.2 Evaluation for the cell movement.

The performances of each method were measured as the accuracy, which was the ratio between the number of correctly selected associations on the tracking results and the number of associations on the ground truth. I evaluated the methods by using different values for each simulation parameter (D and E), respectively, to investigate how the method is robust for the deformation noise and the detection error.

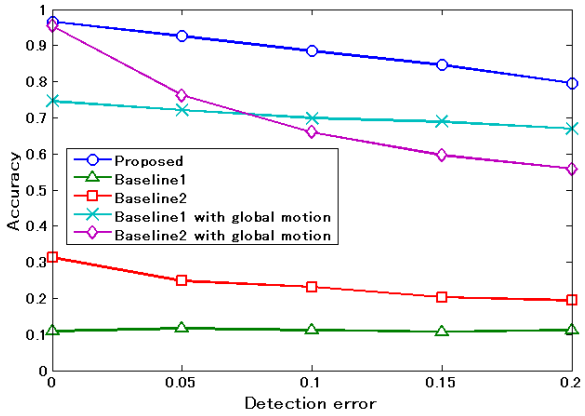


Figure 7.12: Performance according to detection error change. The horizontal axis is the detection error. Here, $N = 500$ and $D = 0.2$.

Figures 7.10 and 7.11 show the experimental results where the horizontal axis is the deformation noise level D and the vertical axis indicates the tracking accuracy. The detection error was fixed in these experiments; $E = 0$ in Figure 7.10 and $E = 0.05$ in Figure 7.11. In Figure 7.10, baseline1, which uses the position similarity score for the cell association, did not work because of large deformation. The accuracy of baseline2, which uses the graph matching adding to the proximity score, was slightly better than baseline1. The accuracy of baseline1 using global motion estimation was clearly improved compared to baseline1. This indicates that the global motion estimation by using non-rigid alignment contributed to improving the performance. Moreover, the proposed method outperformed baseline1 using global motion estimation. This indicates that the target association that uses the pair-wise relationship contributed to improving the performance. The accuracy of baseline2 using global motion estimation was competitive with the proposed method in Figure 7.10 because it uses graph matching for target association, which also contributes to preserving the relative positions of the cells among successive frames. In Figure 7.11, where the detection error was 0.05, the accuracy of baseline2 using global motion estimation severely decreased compared with the other methods. The proposed method outperformed the other methods on all deformation levels.

Figure 7.12 shows the experimental results, where the horizontal axis is the detection error and the vertical axis is the accuracy. This graph illustrates how the methods are robust for the detection error. In these experiments, the accuracies of baseline1 and baseline2 using global motion estimation were also clearly improved compared to the baseline methods. The accuracy of baseline2 using global motion estimation declined more quickly with the detection error than the proposed method. In the real data, the detection results must have some errors (several percentages), and thus, the proposed method can be considered as the best method in comparison.

Method	Rate of false positives	Identification of mitosis occurrence			Mitotic Branching Correctness		
		Recall	Precision	F-measure	Recall	Precision	F-measure
Proposed	0.00	1.000	1.000	1.000	0.8077	0.8077	0.8077
	0.05	1.000	0.936	0.967	0.8308	0.766	0.797
	0.10	1.000	0.675	0.809	0.7923	0.5309	0.6358
	0.15	0.954	0.490	0.648	0.7308	0.3755	0.4961
Baseline1 using global motion	0.00	0.662	0.623	0.642	0.4690	0.4416	0.4548
	0.05	0.697	0.284	0.403	0.4621	0.1882	0.2675
	0.10	0.807	0.194	0.312	0.4276	0.1026	0.1656
	0.15	0.717	0.118	0.202	0.4000	0.0656	0.1127

Table 7.1: Comparison of the proposed method with baseline1 using global motion estimation for terms of identification of mitosis occurrence and mitotic branching correctness.

7.5.3 Evaluation of cell division

I evaluated the accuracy of identifying cell division events on synthetic data. It is believed that the cell division accuracy is affected by the false positives of cell detection. I evaluated the methods by using different false positive rates (0, 0.05, 0.1, and 0.15) to investigate how robust the method is for the rate of false positives. I set the deformation noise D at 0.2.

I evaluated the accuracy of identifying cell division events based on two metrics including the identification of the mitosis occurrence [Huh11] and the mitotic branching correctness [Bise11]. For the identification of mitosis occurrence, a true positive is defined as when a division event is correctly identified (i.e., a newly identified cell is one of the children cells in the ground truth). The mitotic branching correctness measured the accuracy of the mother-daughter relationships of the cell division event as described in Appendix A. In this case, a correct mitotic branching is defined as when both its mother cell and children cells are correctly identified. The precision, recall, and F-measure were evaluated for both metrics. Here, the precision is defined as $\frac{TP}{TP+FP}$, the recall is $\frac{TP}{TP+FN}$, and the F-measure is $\frac{2precision \times recall}{precision+recall}$, where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives.

I used baseline1 based on [Kofahi06], which detects cell division events by resolving a multiple object matching problem for comparison. Baseline2 cannot be used to detect the cell division because the targets of the method are vehicles, which do not divide. The baseline method did not work on the association of the cell movement. Thus, I used baseline1 using global motion for the comparison.

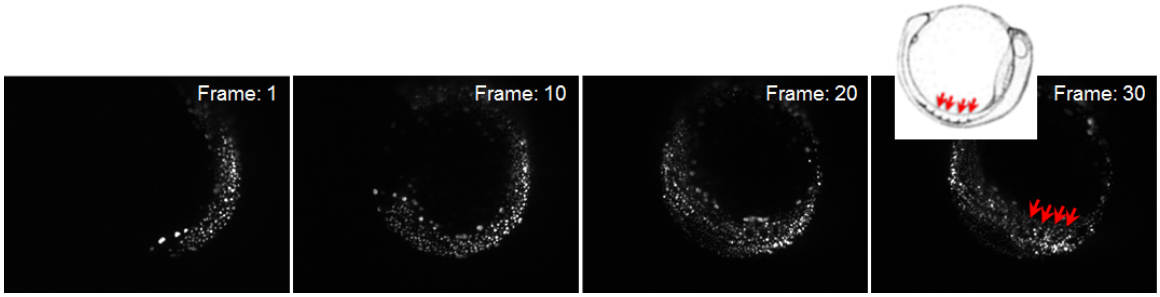


Figure 7.13: Examples of time-lapse images on same z-level ($z = 50$). The cells move and form the somite of a zebrafish. The illustration of the zebrafish was taken from [Kimmel95].

Table 7.1 shows the evaluation results for the identification of mitosis occurrence and mitotic branching correctness. The proposed method outperformed baseline1 in all of the identification of mitosis occurrence for all the error rates. The proposed method achieved 100% accuracies of all the measures when the detection error rate was 0. Then, these accuracies decreased with the error rate. Recall maintained a high level of accuracy at a high error rate 0.15. The precision decreased more quickly than for the recall. This means that the number of false positives of cell divisions was affected by the false positives of the detection results since the false positives were detected as cell division events. The performance of the mitotic branching correctness was lower than the identification of mitosis occurrence for both methods since the mitotic branching correctness is stricter metrics. For this metric, the proposed method also outperformed baseline1 for all of the metrics.

7.6 Application

In this section, I introduce the application of the proposed cell-tracking method in biological research to study the somite formation process of a zebrafish embryo. More specifically, the cell migration trajectories were analyzed to study the cell derivation where the cells came from.

A digital scanned light-sheet microscope (DSLM) was used to capture the time-lapse images every 10 minutes for 5 hours (30 frames). The image size was 1344 $\times$ 1024 pixels (0.645μ m/pixel) and the number of z-slices was 100 (2μm/pixel). Examples of the time-lapse images on the same z-slice are shown in Figure 7.13. The cells moved and formed the somite of the zebrafish. In frame 30, the area indicated by red arrows corresponds to that of the illustration. Figure 7.14 shows an example of the detection results. The image was sliced at the surface curve of the embryo in 3D. Figures 7.15 and 7.16 show the tracking results. In the initial frame, the cells were placed on the right side, then the cells spread

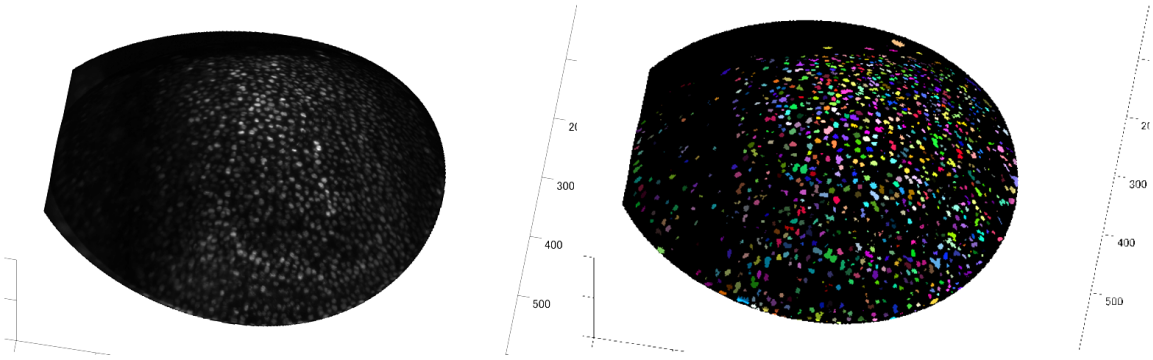


Figure 7.14: Examples of detection results. The results show that the image is sliced at the surface. The image on the left is the original image, and the one on the right is the detection results.

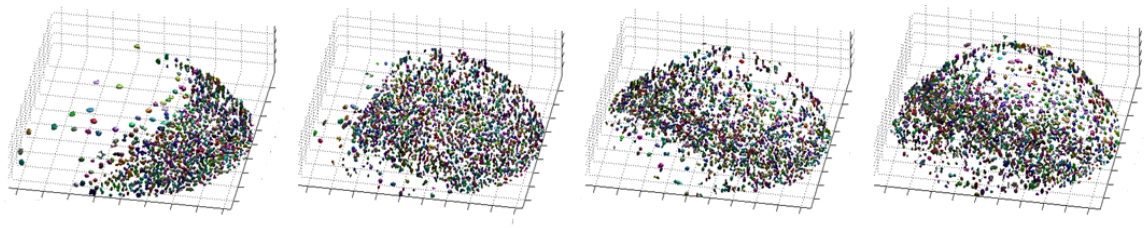


Figure 7.15: Examples of tracking results corresponding with Figure 7.13.

toward the left side of the image along the surface of the semi-sphere. Finally, the cells gathered and formed the somite of the zebrafish. These tracking results allows biologists to more easily study the cell trajectories rather than manually investigating the z-slice stacks.

7.7 Conclusion

I proposed a tracking method that exploits the observation in which nearby cells under high dense conditions exhibit similar motion patterns. This is done by introducing two stages of tracking. First, the global motion of cells is coarsely estimated by treating the problem as non-rigid alignment in 3D. Then each cell is tracked using multiple target association between successive frames, which uses both pairwise spatial relationships and the estimated global motion. The proposed method was evaluated on synthetic point-set and compared against the existing methods. The evaluation results showed that both stages (global motion estimation and target association using pairwise spatial relationships) contribute to

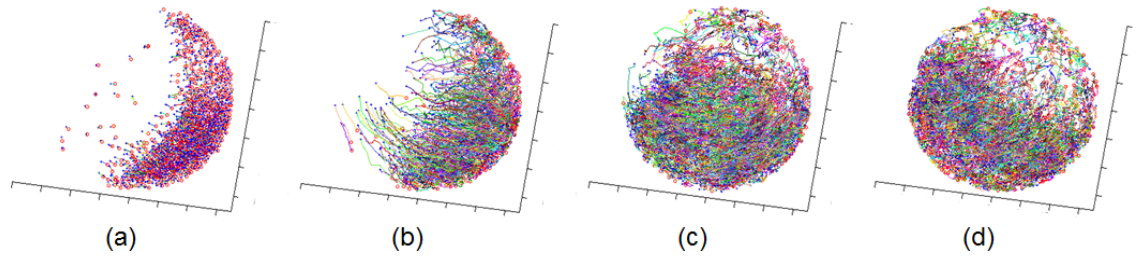


Figure 7.16: Examples of track trajectories corresponding with Figure 7.15. (a) Trajectories from frames 1 to 2, (b) trajectories from 1 to 10, (c) trajectories from 10 to 20, and (d) trajectories from 20 to 30.

achieving a better tracking accuracy. I also introduced an application of the proposed cell-tracking method in biological research that analyzes the somite formation process of the zebrafish embryo.

Chapter 8

Biological Applications

8.1 Introduction

In this chapter, I aim to show how easily and effectively our automated cell-tracking system can provide detailed spatio-temporal cell behavior measurements for biological analysis. Spatio-temporal measurements of cell behaviors are important for critical analysis, because the cell culture conditions vary with time and space on the dish. For example, the effect of a medicine may change with time and space since cell density can differ in a different space. I present an application of automatic cell-tracking for wound healing assay *in vitro* under three different culture conditions to demonstrate how easily and effectively automated cell-tracking systems can provide detailed spatio-temporal cell behavior measurements for biological research.

The wound healing assay is an easy and low-cost method to allow for observing cell migration *in vitro* [ChunChi07]. In this assay, cells are firstly grown to form a confluent monolayer *in vitro*. An artificial wound is generated by scratching and displacing a group of cells at the center as shown in Figure 8.1, and then the healing process is observed while neighboring cells fill in the wound area as shown in Figure 8.1 (a-c). This healing process takes 3 to 24 hours, depending on cell types and culture conditions. The healing process is monitored by a sequence of microscopic images. Liang *et al.* [ChunChi07] compared several migration assays *in vitro* and described advantages of using the wound healing assay that mimics cell migration *in vivo*. For example, endothelial cells (ECs) *in vitro* mimic the process in which ECs in the blood vessels migrate into the denuded area to close the wound. Yarrow *et al.* [Yarrow04] measured the healing speed by observing the size of the wound area in order to analyze the effectiveness of different culture conditions. For further analysis of the effectiveness of the cell culture condtions, more detailed measurements of the cell behaviors are often required. For example, Abbi *et al.* [Abbi02] analyzed the cell migration path to assess the effects of expression of exogenous genes on migration of

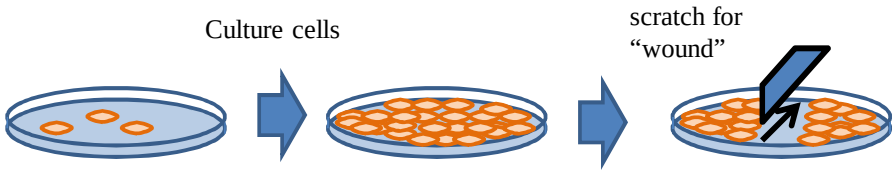


Figure 8.1: Process of making wound.

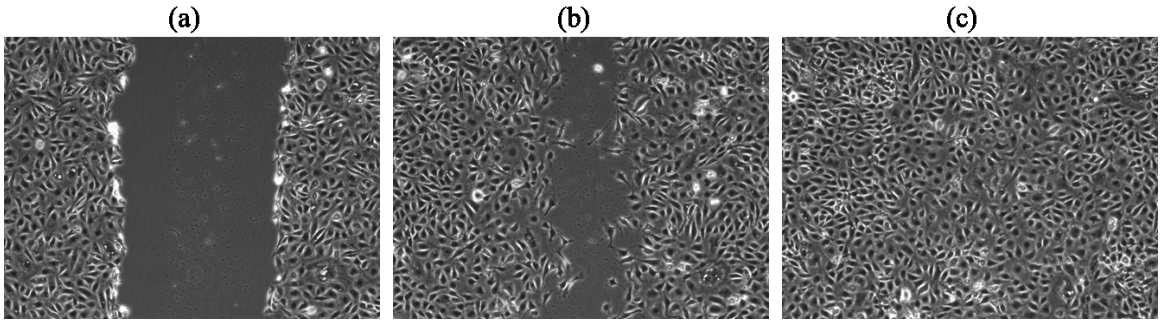


Figure 8.2: Example images of the wound healing process. (a) The initial image on the healing process. (b) An image at which cells move to wound area. (c) An image at which cells fill the wound area.

individual cells. Nikolic *et al.* [Nikolic06] manually tracked cell migration in wound healing assay in order to understand how multiple cells execute highly dynamic and coordinated movements during the healing process.

Cell tracking allowed them to analyze how individual aspects of the wound contribute to the coordinated dynamics of cells. Zahm *et al.* [Zahm97] used a computer-assisted technique to quantitatively study the cell proliferation and migration during the wound healing process. Citing the difficulty of tracking cells in phase-contrast microscopy images, they used chemical compounds to create fluorescent images to track cells and count proliferative cells. Such chemical compounds generally interfere with the efficacy of drug candidates.

I applied our automatic cell-tracking system to sequences of phase-contrast microscopy images of a wound healing assay *in vitro* under three different culture conditions (i.e., three different amounts of medicine “Latrunculin B ”that interferes with cellular activity), where the cell-tracking system is developed based on the method proposed in Chapter 2. Our system can locate cell regions and track more than hundreds of cells individually under non-invasive imaging. It allows us to compute spatio-temporal measurements including the cell density, migration speed and direction, statistics of mitosis events, and their mutual dependency in order to analyze how the cell culture conditions (i.e., amount of the medicine) effect the cell behaviors over time and space. These measurements can provide critical information for investigating the healing process.

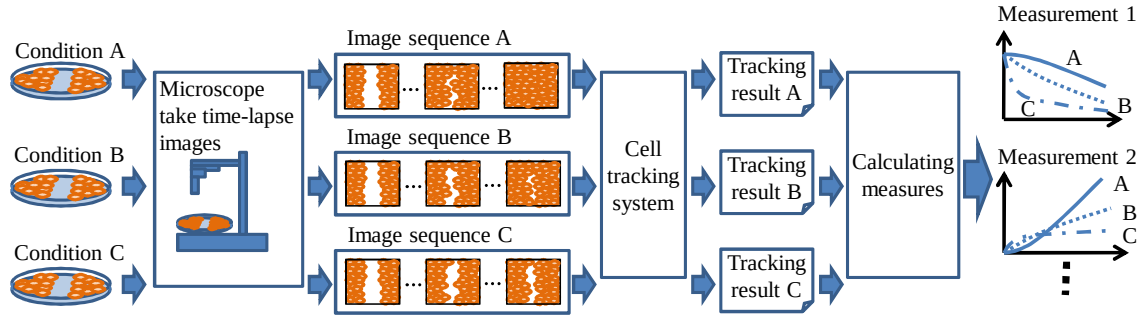


Figure 8.3: Flow of the wound healing assay experiments under three culture conditions.

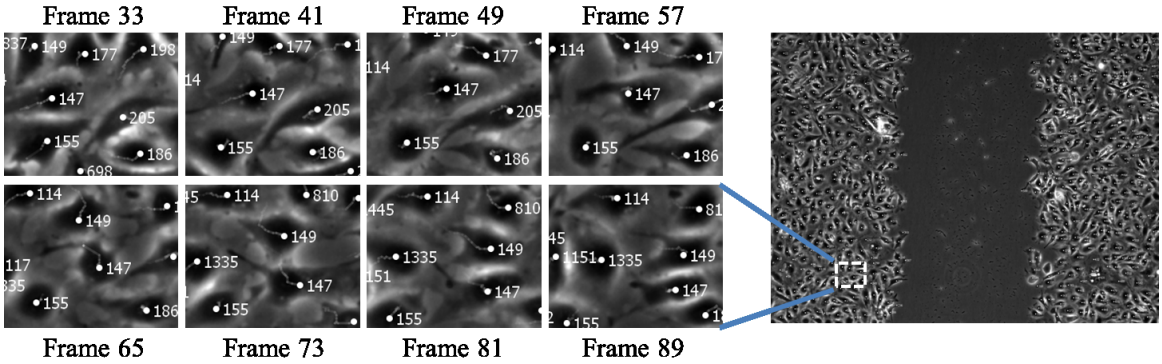


Figure 8.4: Right: Example image of the tracking result. Left: Example image sequence of the zooming images that correspond with the white dot rectangle in the right image.

8.2 Space-time analysis of cell behaviors of wound healing

Automated cell-tracking allows us to analyze the space-time transition of cell behaviors quantitatively and in detail in the wound healing assay. I applied our tracking system recently developed in [Kanade11] for experiments which aims to screen out cell migration inhibitors. Cell migration inhibitor is important for inhibiting the migration ability of cancer cells [Yarrow04][Pril09]. In general, the cell behaviors where the cells move toward to an open wound in a cellular mono-layer is thought to predict their migratory ability. Figure 8.3 shows the overall flow of wound healing analysis experiment. Firstly, the culture dishes with wound area are prepared under difference types of culture conditions. These dishes are observed by microscope, generating a time-lapse image sequence. The image sequences are inputted to the automatic cell-tracking system. From the tracking results, various measurements that characterize the cell behavior are calculated. In the experiments, I applied our automatic cell-tracking system to sequences of phase-contrast microscopy images of a wound healing assay *in vitro* under three different culture conditions (i.e.,

three different amounts of medicine "Latrunculin B "that interferes with cellular activity). Figure 8.4 shows an example of the tracking result where the cells are well tracked in high confluence. The tracking system assigns a positive integer ID to each cell that is being tracked as its unique identifier. As its descriptor, each cell has its parent identifier Parent-ID for maintaining its lineage information (Parent-ID=0 for cells with no parent, i.e., those cells that appear in the very first frames) and its state information (i.e., its centroid and contour shape of the cell region) at each frame. Using this information, various cell behavior measurements can be computed. It allows us to compute spatio-temporal measurements including the cell density, migration speed and direction, statistics of mitosis events, and their mutual dependency. These measurements can provide critical data for investigating the healing process.

8.2.1 Cell culture conditions and imaging modality

The followings are how I prepared the cell culture dishes and how the image sequences were obtained for the experiments.

Cell culture conditions

On three dishes, BAEC(bovine aortic endothelial cells) were cultured under three different culture conditions. For each dish, a group of cells at the center of the dish was scratched and displaced on a confluent monolayer. Different amount of medicine was added to each dish.

Condition A : control (no medinine)
Condition B : 10nM (nano molar) of Latrunculin B
Condition C : 100nM of (nano molar) Latrunculin B

Time-lapse imaging

The area around the wound area in each dish is observed with a Leica DMI 6000B inverted microscope using a 10X objective with phase optics until neighboring cells fill completely. Images were acquired every 5 minutes for 17 hours using a 12-bit CCD camera with each image of 1040×1392 pixels.

8.2.2 Cell behavior characteristic measurements

Using tracking results, various measurements of cell behavior characteristics were calculated. In cell behavior analysis, the change of the cell density over time on the whole area is a useful index [Brett09][Tamura98]. To investigate how the cell culture conditions affect the cell migration, speed and direction of the cell migration are often measured [Brett09][Gu99].

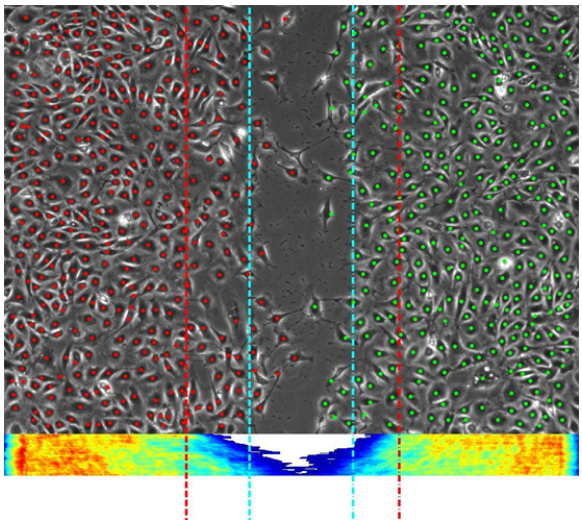


Figure 8.5: Example of the jet map of cell density changes over time with tracking results, each row of which represents the density. In the jet map, red color shows higher density, blue color shows lower density.

Cell culture condition usually affects both migration and proliferation. To separate these affects, the statistics of the mitotic events are important. Our system allows us to compute all of these spatio-temporal behavior characteristics in detail, including cell density, the speed and the direction of cell migration, and the statistics of mitosis events.

Cell density

Figures 8.5 and 8.6 show how the cell density changes over time and location. Since cells generally migrate horizontally in the experiments as shown in Figure 8.1, the cell density is computed over narrow vertical window (the width of the window is 40 pixels, i.e., 36.5 μm) as shown in Figure 8.6 (a). The vertical lines show that the 95% cell migration front of left and right sides, which is defined as the 95th percentile line of the total cell count of each side. Red color lines indicate the 95% line at the inital frame, blue dotted lines indicate the 95% line at the current frame. Figure 8.6 (b-d) show the comparison of the space-time transision of the cell density under three different conditions. At the second row (condition A), the cell density in the wound area is low at frame 1. Then, cells in the left and right regions migrate into the central area and the density in the wound area increases until it becomes flat in frame 200. I observe the similar behaviors for the other conditions, but cells on condition C (100nM) migrate more slowly than those in condition A and B. The density in the wound area is still low at the end of the sequence.

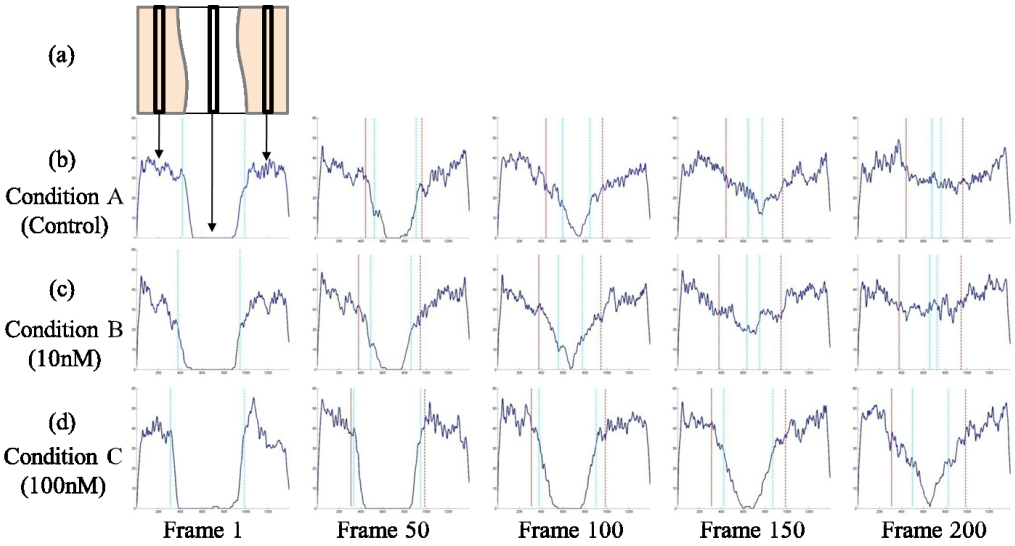


Figure 8.6: Example of the space-time transition of the cell density.

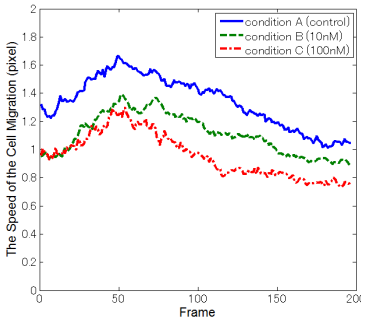


Figure 8.7: Average speed of tl cell migration over the time.

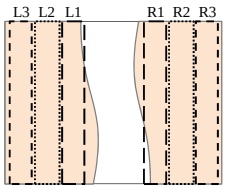


Figure 8.8: Local areas.

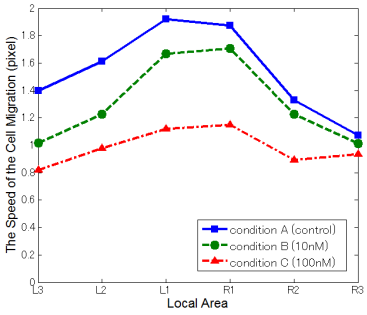


Figure 8.9: Average speed of the cell migration at each local area.

Speed of cell migration

To analyze how the speed of cell migration changes over time, the average speed of the cell migration was computed over the whole area in each frame. The results presented in Figure 8.7 show that the speed in the condition A (control) is consistently higher than those in other conditions, and the speed in the condition C (100nM) is the slowest. The migration speed firstly increases until frame 50, and then it continuously decrease with the time.

It is conceivable that the speed of the cell migration depends on the distance from the wound area. To know whether this is the case, the filled area (i.e., the left and right side of the wound area) is divided into six local areas as shown in Figure 8.8. These local areas were defined based on the distance from the wound area, from L1 to L3 at the left side and from R1 to R3 at the right side away from the wound area. The cells on the edges of the wound area migrate toward the center, therefore, these local areas also move toward

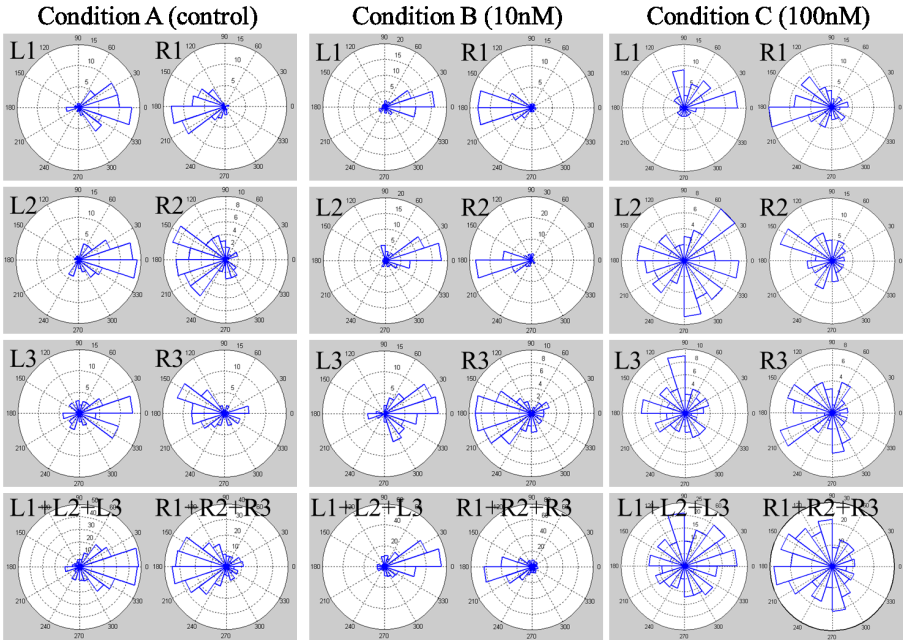


Figure 8.10: Rose diagrams of cell migration directions on each local area.

the center with the time. For each local area, the average speed of the cell migration is computed as shown in Figure 8.9. Understandably, the cells in condition A (control) move faster than the others in every local area, and the cells in condition C (100nM) are the slowest. The graph indicates another interesting phenomenon in that the order of speeds are L1, L2 and L3 for the left side, and R1, R2 and R3 for the right side; that is, the speed of cell migration decreases with the distance from the wound edge.

Direction of cell migration

To quantitatively analyze the direction of the cell migration, the distribution of cell migration directions on each condition was plotted by an angular histogram (rose diagram) as shown in Figure 8.10. The first three rows show the distributions of the direction on each local area, respectively, for conditions A, B and C, and the bottom row shows the distributions on the whole areas left and right. I can observe that the cells tend to migrate toward the wound area in every local area on condition A and B. The cell migrations in L1, R1 (nearest area to the wound) are most highly directional to the direction to the wound. Also the graphs indicate an interesting phenomenon that cell migration in condition C (100nM) is less directed to the wound area (0° for the left side, 180° for the right side). This means that the speed of the cell migration toward the wound area is the slowest in condition C (Figure 8.6 (d)) not because the migration speed is slow, but because the migration direction is less directed.

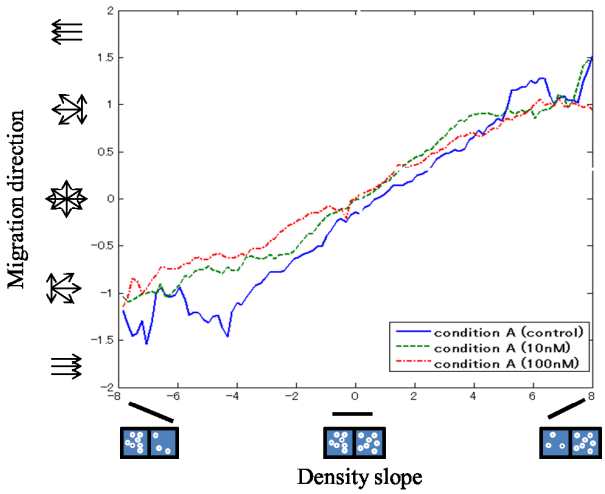


Figure 8.11: Relationship between density slope and migration direction.

Density slope and migration direction

It is conceivable that the direction and speed of the cell migration depend on the density slope. For example, when the density slope is high, the cells on the area may move from high-density area to low density area. To confirm the hypothesis, the graph, in which shows the relationship between density slope and migration direction, was computed as shown in Figure 8.11. Th horizontal axis indicates the cell density slope which is computed as how the cell density change over x axis on Figure 8.6. As shown in the illustration described below the graph, minus density slope indicates the density decreases from left to right direction on the image, plus density slope indicates the density increases. As shown in the illustration described at left side of the graph, vertical axis indicates the directional migration speed in which the cells move toward to the same direction as the slope. When the value of the vertical axis is 0, the cell migration direction on the area is less directed. When the absolute value of the vertical axis is high, the cell migration is highly directional to the direction from high-density to the low density. I can observe that cell migration direction is highly directional on the area where the density slope is high, and migration direction is less directed on the area where the density is flat.

Number of cell mitosis events

To analyze how the number of cell mitosis events changes over time, the number of accumulated mitosis events was computed over the whole area. The results presented in Figure 8.12 show that the number of mitosis events in the condition A (control) is higher than those in the other conditions. The curves of the accumulated number are almost linear. It indicates that the mitosis events occur consistently.

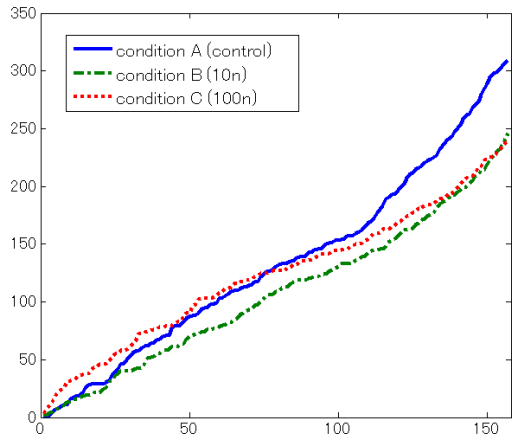


Figure 8.12: The number of accumulated cell mitosis events at each frame.

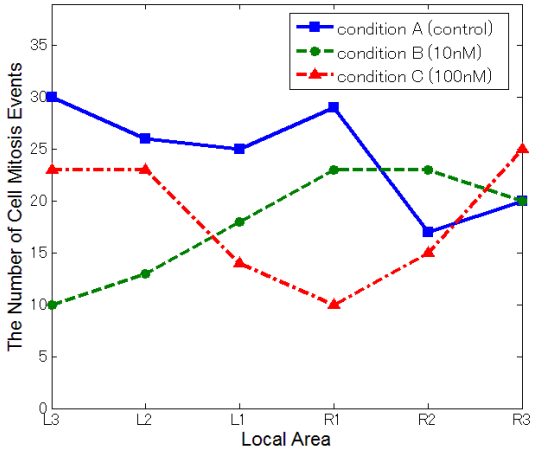


Figure 8.13: The number of cell mitosis events at each local area.

To show how the number of the cell mitosis events changes by location, the number of the cell mitosis events is computed over each local area as shown in Figure 8.13. In this graph, I observe no distinctive characteristics. This means that the culture condition has more effects to the number of mitosis events than the distance from the wound area does.

8.3 Discussion and conclusion

I presented the application of automatic cell-tracking in phase-contrast microscopy images for wound healing assays *in vitro* in order to produce detailed quantitative analysis of the cell behaviors under three different culture conditions.

Understanding the cell behaviors is important to studying the influence of the environments including the types of medicines, the amount of the medicine, and materials in which these cells can grow. In the past, the simple measurement, such as the size of the wound area is often used, but such simple measures cannot provide the detailed analysis of cell behaviors. The spatio-temporal measurements of cell behaviors are important for critical analysis, because the cell culture conditions vary with time and space on the dish. Our system can provide such spatio-temporal cell behaviors measurements: the cell density, cell migration speed and direction, and statistics of cell mitosis events. The results of the experiments demonstrated the effectiveness of automatic cell-tracking for quantitatively analyzing cell behaviors.

Chapter 9

Conclusion

9.1 Summary

In this thesis, I have proposed several cell-tracking methods to address the following difficulties in tracking cells under high-density conditions: multiple cells touching and forming a cluster with blurry intercellular boundaries, the number of cells changing due to cell division and cells entering or leaving the field of view, and cells moving large distances.

I first proposed tracking methods based on detection-and-association tracking methods in Chapters 2 and 3 for low-to-middle density conditions in which several cells partially overlap with blurry inter-cellular boundaries. In Chapter 2, I proposed a frame-by-frame association tracking method that identifies touching-cell clusters in the association step then re-segments the clusters to its member cells by partial contour matching. The method enables robustly tracking of two or three partially overlapping cells.

In Chapter 3, I proposed a global spatio-temporal data association tracking method that first generates reliable tracklets by frame-by-frame association, then the tracklets are globally linked using integer linear programming to reduce the negative influence of false positive detections. These proposed methods outperform a state-of-the-art tracking method proposed by Li *et. al.* [Li08].

I then proposed detection and tracking methods based on an idea of selecting an optimal set of detection and association from redundant candidates under non-overlapping constraints to address tracking under high-density conditions. In Chapter 4, I proposed a cell-detection method for obtaining high-accuracy detection results under dense conditions. It first detects redundant candidate regions, which may include many false positives but in turn very few false negatives, by allowing candidates to overlap. This problem of optimal region selection under non-overlapping constraints is solved by a binary linear programming problem under non-overlapping constraints. This method exhibited the best performance, with an F-measure of over 0.9 for all datasets in experiments.

In Chapter 5, the idea of selecting an optimal set from redundant candidates was extended to tracking. One of the drawbacks of current detection-and-association methods is that they heavily depend on the detection results since the errors of the detection step directly propagate to the association step. To mitigate this problem, the proposed tracking method determines the detection results in the tracking process by solving concurrently both optimal set selection of detections from redundant candidates and association between successive frames. This method makes it possible to generate reliable tracklets for the global spatio-temporal association method proposed in Chapter 3. In Chapter 6, I furthermore extended the method to solve the joint problem of selecting an optimal set of tracklets from redundant candidate tracklets and global associating tracklets in order to become more robust under dense conditions.

In Chapter 7, I proposed a cell-tracking method that enables us to track cells that move largely when the increment in time-lapse imaging cannot be shortened to monitor a wide area in 3D. The method exploits the observation in which nearby cells under high-density conditions exhibit similar motion patterns by introducing global motion estimation and local pairwise spatial relationships. This method exhibited better tracking accuracy on synthetic point-sets compared to current methods [Kofahi06][Xiao10].

Finally, in Chapter 8, I presented an application of automatic cell tracking for wound healing assay in vitro to demonstrate how effectively automated cell-tracking systems can provide detailed spatio-temporal cell behavior metrics for biological analysis.

9.2 Comparison of the proposed methods

In this section, I discuss advantages and disadvantages of the proposed methods.

The tracking-methods proposed for high-density conditions in Chapters 5 and 6 may not outperform tracking-methods proposed in Chapters 2 and 3 when an experiment is done under low-to-middle density conditions. Therefore, for low-to-middle density conditions, the methods proposed in Chapters 2 and 3 are better from the aspect of computational cost. Thus, I discuss advantages of methods for low-to-middle and high density conditions respectively.

I first discuss the methods proposed for low-to-middle density conditions. When cells partially overlap through long-term, the partial contour matching method proposed in Chapter 2 is better than the global association method proposed in Chapter 3 since the global data association method cannot link tracklets where their temporal-distance is larger than a spatio-temporal window. For example, neural stem cell tend to partially overlap through long-term. In such case, the partial contour matching method has advantages.

On the other hand, in the partial contour matching method, a switching error occurs when a false positive segmentation appears near a mitotic cell. Therefore, the global associ-

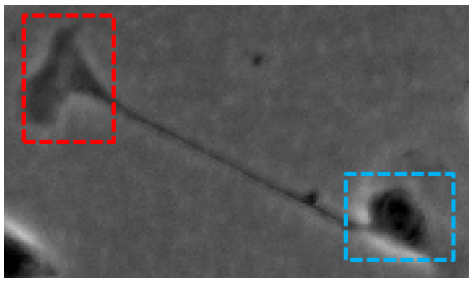


Figure 9.1: Example image of false positive. Red box indicates a chip of a cell. Blue box indicates a cell main body. In this example, current cell detection methods tend to detect both of them.

ation method is better when false positives of detection tend to occur rather than long-term partial overlapping. For example, a cell type tend to extend their bodies during migration as shown in Figure 9.1. In that case, a chip of a cell is often detected as a false positive by automatic detection. In such case, the global association method has advantages.

Next, I discuss the methods proposed for high-density conditions. The method for solving both cell detection and association proposed in Chapter 5 has an advantage from the aspect of computational cost. The method well works when the segmentation results from initialization are good. For example in Figure 5.3, the boundaries of the cells in the initial frame are clearer and the cell regions are well segmented. In such case, the method proposed in Chapter 5 is better. The method proposed in Chapters 5 and 6 use multiple-thresholding in preconditioning images for candidate region detection based on an assumption that the intensities on the inside of a cell are slightly higher than those at its boundaries among touching cells. Thus, these methods have a disadvantage when cells often deeply overlap, in which the intensities at boundaries are higher than those on the inside of a cell. This work addressing the problem remains as a future work.

The method proposed in Chapters 7 has advantages for a case when cells move largely since the increment in time-lapse imaging cannot be shortened to monitor a wide area in 3D. However, the methods does not address the cases that detection results include many errors under high-density. Combining the method using global motion and local pairwise relationships to the method proposed in Chapter 6 may improve the performance for such problem.

9.3 Future directions

Although my tracking methods exhibited better tracking performances compared with current tracking methods under dense conditions, they have limitations. For more detailed

cell image analysis including detailed cell shape analysis, differentiation analysis, and cell lineage analysis, it is necessary to achieve higher accuracy of tracking and individual cell segmentation. In this chapter, I discuss several ideas to improve and extend the proposed methods.

Improving tracking accuracy: associating score of tracklets

With my methods, the associating score was computed on the basis of a spatio-temporal distance in global tracklet association. After associating the tracklets, the cell trajectories are interpolated. However, there were some unnatural results, in which a cell trajectory passed in the background area, when I compared the tracking results with original data. I argue that using interpolated trajectories with original data to compute association scores of tracklets mitigates this type of error. For example, if the interpolated trajectory between associated tracklets passes in the background, the association score is penalized. This idea may contribute to selecting the correct tracklet association.

Improving tracking accuracy: tracking using landmark

With my methods, all cell tracklets are equally treated in the objective function of optimizing association, and the optimization problems are solved at once while detection and association of some cells are clear and those of others cells are not. On the other hand, a human annotator first detects and annotates the clearer cells, which are brighter or located on the boundary of a cluster, then identifies the other cells based on the first identified cells as a landmark when a movement distance is large. For example, to find landmark cells, a saliency map and a pair-wise structure of cells may be used. The position in relation to the landmarks is used for association of the other cells after identifying the landmarks. This idea may contribute to improving the tracking accuracy.

Detailed cell shape segmentation

In this thesis, my aim was cell-position tracking rather than segmentation of an entire cell region with a complex shape. The proposed methods contribute to analyzing cell behavior metrics including the number of cells, migration speed, and number of mitoses over time. However, in biological research, detailed cell shape analysis is an important task. For example, cell appearance differs with cell quality, for instance, an old fibroblast cell tends to spread and become larger than a younger cell. It is necessary to accurate segment detailed cell shapes. This remains for future work. In individual detailed cell shape analysis, the density of cell shape is usually not too high since cell shape may be limited due to surrounding cells under confluence conditions and it is too difficult to recognize the detailed cell shape. Thus, the target condition is low-to-middle. Several methods have been

proposed to segment cell regions and background under such conditions [Yin10b]. These methods can segment cell regions though they cannot segment individual cell regions. My tracking results reported in this thesis should be of use in segmenting cells with complex shapes by incorporating the foreground-and-background segmentation methods.

Interactive system for cell image analysis

Automated cell-tracking systems have not been widely used for biological research yet though many tracking methods have been proposed. One of the main reasons is that there are many variations in analysis targets of cell images including cell types, culture conditions, magnification of microscope, and microscope types. Tracking accuracy tends to dramatically decline if the target images are different. In particular, cell detection and segmentation are adversely affected by the image conditions. It is difficult to develop the perfect tracking method for all cell image conditions. If multiple methods are integrated in an automated system, a biological user usually cannot select an optimal method from the menu. An interactive system is important to address this issue. Given few image samples that are manually annotated and answers for prepared questions, which include cell and microscope types *etc.*, the system automatically suggests optimal tracking and shows the tracking results of the sample data by comparing the performance of methods. The sample annotation data are also used for training data of learning algorithms.

In addition, the required tracking accuracy is different from the analysis, in which high accuracy is not required to analyze cell motion and almost perfect accuracy is required for cell lineage analysis. To obtain perfect tracking results, manual correction is required. To facilitate manual correction, the system shows the area when the automated results do not seem reliable. The collection results are also used as training data to boost a more accurate model. I argue that the interactive system contributes to cell behavior analysis in biological research.

Appendix A

Performance metrics

In this appendix, details of the performance metrics used for this thesis are introduced.

One of the main aims of the automated cell tracking is obtaining fine-grained information of cell behaviors. For the measurements of cell behaviors, the number of cells, the density of cells, the speed and direction of cell migration, the number of mitosis events and the cell division time are often computed and analyzed in many biological works. To obtain these measurements, cell tracking provides the track information and cell mitosis information. In general, the track information consists of an identity of the track, and a sequence of cell locations (i.e., positions or regions) at each frame. The cell mitosis information indicates the cell division event that includes the mother-daughter relationship and timing. These output information can be divided to three types of information i.e., cell detection, track identity and mitosis information. I introduce the performance metrics for evaluating these elements respectively.

A.1 Preparation for evaluation

To create a ground-truth, we roughly annotated the cell centers at each frame by hand using an annotating system which was developed in CMU Cell Image Analysis Consortium. The other annotation systems are provided for plug-in of Image-J. Figure A.1 shows tracking annotation results under high-density conditions. In the annotation, a cell ID is assigned to a cell. An annotator clicks a center-like position of a target cell at each frame until the end of the sequence in a subjective manner, where the annotated positions were not exactly the center positions of the cell. The annotated position and the cell ID are registered to the system for each cell at each frame, representing '(x, y), frame-index, cell-ID'. (x, y) indicates the annotated position in the image.

When a tracking target cell divides to two cells. The track of the dividing cell is terminated and two new IDs are assigned for the children cells. The relationships of the children

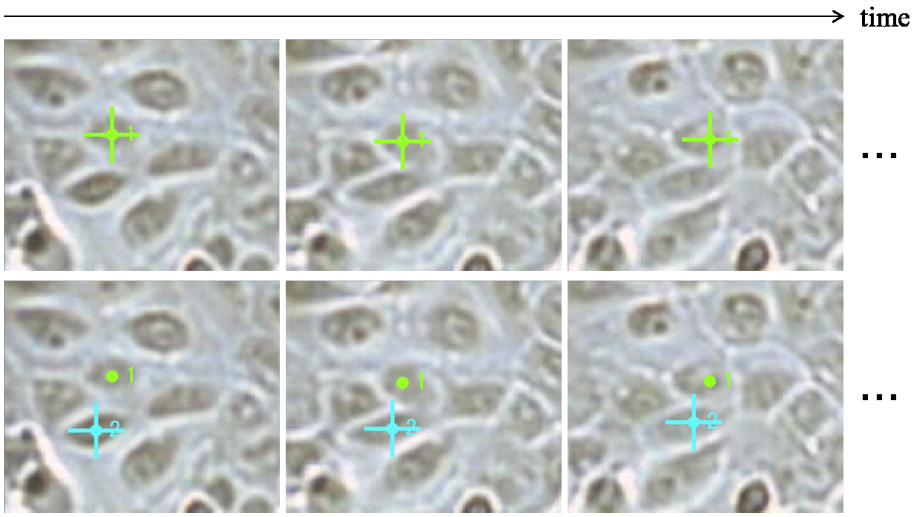
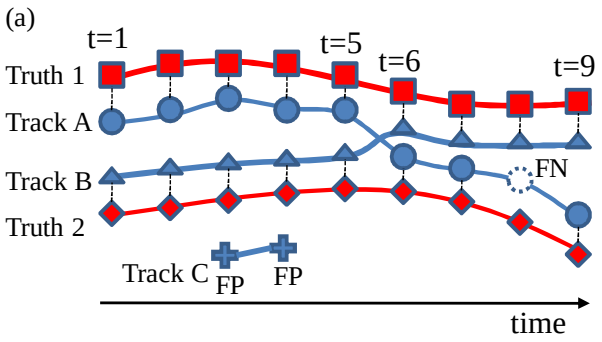


Figure A.1: Example images of manual annotation. Top: annotation results of cell '1'. Bottom: annotation results of cell '2'.

IDs and the parent ID are registered as representing 'parent-ID, child-ID'. This annotation process was done for all target cells. Here, automated tracking results were also registered using the same format.

After annotating cells, correspondences between tracking results and ground-truths are identified at each frame. I here use a global nearest neighbor (GNN) method with Euclidean distance between annotation positions and center positions of automated tracking at each frame. The distance defines a cost to assign a tracking result to a ground-truth at each frame. A tracking result and a ground-truth are paired only if the distance between them is smaller than a threshold. If the distance is larger than the threshold, the tracking results and ground-truth have no pair i.e., the tracking results is counted as a false positive and the ground-truth is counted as a false negative. Basically, the threshold is set based on an average of major radii of cells.

As the results of this step, a truth-to-track paring map is obtained. I show example trajectories and the results of truth-to-track paring map in Figure A.2 (a) and (b), respectively. Track A is paired with truth 1 from $t = 1$ to 5, then Track A is paired with Truth 2 after $t = 6$. Track C has no pair on $t = 3$ and 4, these are false positives. Truth 2 has no pair on $t = 8$, this is a false negative. Once truth-to-track pairing map is made, performance metrics can be calculated.



(b)

truth \ time	1	2	3	4	5	6	7	8	9
Truth 1	A	A	A	A	A	B	B	B	B
Truth 2	B	B	B	B	B	A	A	-	A

Figure A.2: (a) Example trajectories of track and gorund-truth. (b) Truth-track pairing map. Here, truth indicates a trajectory of an actual cell, Track indicates a trajectory on tracking results.

A.2 Performance metrics for detection

To evaluate the accuracy of cell detection, recall and precision are often used. Recall is the ratio of the number of detected cells to the total number of cells in ground-truth. Precision is the ratio of the number of detected cells to the number of the detected objects. The trade-off between recall and precision is well-known. Greater precision decreases recall and greater recall leads to decreased precision. The F-measure is the harmonic-mean of Precision and Recall and takes account of both measures. The metrics can be computed as:

$$recall = \frac{TP}{(TP + FN)} \tag{A.1}$$

$$precision = \frac{TP}{TP + FP} \tag{A.2}$$

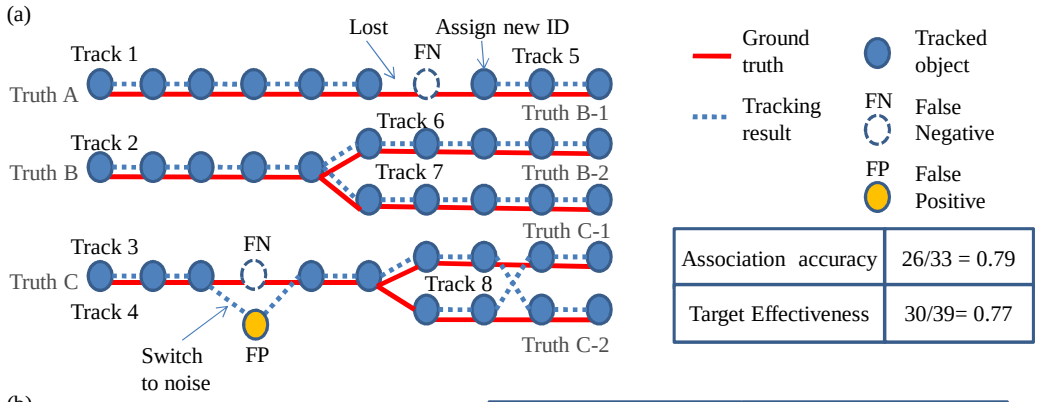
$$F - measure = \frac{2 \times recall \times precision}{(recall + precision)} \tag{A.3}$$

where TP is true positives; FP is false positives; TN is true negatives; and FN is false negatives. In the example in Figure A.2, TP is 17, FP is 2, and FN is 1. Thus, recall is 0.944, precision is 0.895, and F-measure is 0.919.

A.3 Performance metrics for tracking

To evaluate the accuracy of cell tracking, I used the association accuracy and the target effectiveness.The association accuracy is the number of correct associations divided by the number of total associations on the ground-truth. I show an example of the tracking results in Figure A.3 (a). The red color branch indicates one association between successive frames

(a)



Association accuracy	26/33 = 0.79
Target Effectiveness	30/39= 0.77

(b)

time / truth	1	2	3	4	5	6	7	8	9	10	Association accuracy: #correct association	Association accuracy: #association of truth	Target Effectiveness: #observation of track	Target Effectiveness: #observation of truth
Truth A	1	1	1	1	1	1	N	5	5	5	7	9	6	10
Truth B	2	2	2	2	2	-	-	-	-	-	4	5	5	5
Truth B-1	-	-	-	-	-	6	6	6	6	6	4	4	5	5
Truth B-1	-	-	-	-	-	7	7	7	7	7	4	4	5	5
Truth C	3	3	3	N	3	3	-	-	-	-	3	5	5	6
Truth C-1	-	-	-	-	-	-	3	3	8	8	2	3	2	4
Truth C-2	-	-	-	-	-	-	8	8	3	3	2	3	2	4
								Total			26	33	30	39

Figure A.3: (a) Examples of correspondence of tracking results and ground-truth. (b) Truth-track pairing map of (a) and target effectiveness for each truth trajectory. N indicates no assigning.

on the ground-truth, the blue circles are track objects that are paired with the cells on the ground-truth, and the blue color branch indicates one association between successive frames on the tracking results. If assigned track IDs are same between two successive frames in the truth-to-track paring map, the association is counted on a correct association. For example in Figure A.3 (b), Truth A has 9 associations from frame 1 to 10. An association between frame 1 and 2 (1-to-2) for Truth A is counted on a correct association. Associations for frame 6-to-7 and 7-to-8 are counted on incorrect associations. In this example, the total number of correct associations is 26 and the total number of associations on the ground-truth is 33. Therefore, the association accuracy is 0.79 (26/33).

To compute target effectiveness, each truth (human annotated) is first assigned to a track (computer generated) that contains the most observations from that ground-truth. Then target effectiveness is computed as the number of the assigned track observations over the total number of frames of the truth. It indicates how many frames of targets are followed by computer-generated tracks.

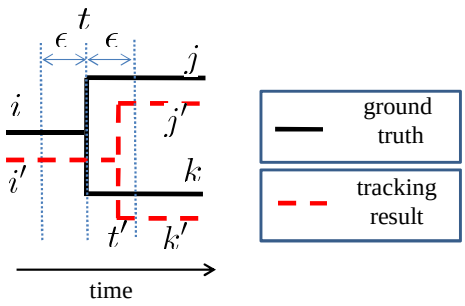


Figure A.4: Evaluation of mitosis branching.

I show an example of truth-track paring map in Figure A.3(b). Truth A is paired with both of track 1 and 5. The number of observation of track 1 paired with Truth A (i.e., the number is 6) is larger than the number of track 5 paired with truth A (i.e., the number is 3). Therefore, truth A is assigned with track 1. In the same way, longest observation track is assigned with each truth. Then, target effectiveness is calculated. In this example, the total number of the assigned track observations is 30 and the total number of frames of truth A is 39. Therefore, the target effectiveness is 0.77 (30/39).

A.4 Performance metrics for identifying cell division

Cell lineage is also important for stem cell research [Ravin08][Chisholm00], as it provides fine-grain quantification measurements of mitosis, such as division time and synchrony of mitosis. To accurately generate cell lineages, the mitotic branching correctness [Bise11] is also an important measure. The mitotic branching correctness measured the accuracy of mother-daughter relationships between tree branches. Figure A.4 shows an example of a mitosis branch, black lines indicate ground-truth trajectory, and red dotted lines indicate tracking results. In the ground-truth, there is a birth event at time t in which cell i divides into cell j and k. If the automatic tracking results include a birth event of the cell i' that corresponds to cell i, and children j',k' of the cell i' are also corresponded to cell j and k, and the time distance between the two birth events, $\epsilon = \|t - t'\|$, is close enough (i.e., $\epsilon < \theta_\epsilon$), it is considered as a correctly detected mitosis branching (TP). If the condition is not satisfied, then it is a false negative error (FN). For a mitosis event that is detected by automatic tracking, if there is no mitosis event in the ground-truth that can correspond to it in terms of cell identities and timing, it is considered as a false positive error (FP). With these definition, recall and precision of mitotic branching correctness are computed. In example in Figure A.3, recall is 0.5, precision is 1.0.

Bibliography

[Abbi02] S. Abbi, H. Ueda, C. Aheng, L.A. Cooper, J. Zhao, R. Christopher, J.L. Guan, Regulation of focal adhesion kinase by a novel protein inhibitor FIP200. *Molecular Biology of the Cell*, 13, 3178-3191, 2002.

[Bise09] R. Bise, K. Li, S. Eom, and T. Kanade, Reliably tracking partially overlapping neural stem cells in DIC microscopy image sequence. *Proceedings of MICCAI Workshop on Optical Tissue Image analysis in Microscopy, Histopathology and Endoscopy*, pp.67-77, 2009.

[Bise11] R. Bise, Z. Yin, and T. Kanade, Reliable cell tracking by global data association. *Proceedings of IEEE International Symposium on Biomedical Imaging(ISBI)*, 1004—1010, 2011.

[Bise13] R. Bise, Y. Maeda, M. Kim, and M. Kino-oka, Cell tracking under high confluency conditions by candidate cell region detection-based-association approach. *Proceedings of IASTED Biomedical Engineering*, 791-057, 2013.

[Bise15] R. Bise, and Y. Sato, Cell detection from redundant candidate regions under non-overlapping constraints. *IEEE Trans. on Medical Imaging, 2015. (under review).*

[Blackman86] S. Blackman, Multiple-target tracking with radar applications. *Artech House Publishersl*, 1986.

[Bonneau05] S. Bonneau, M. Dahan, L.D. Cohen, Single quantum dot tracking based on perceptual grouping using minimal paths in a spatiotemporal volume. *IEEE Trans. on Image Processing*, 14, 9, 1384 - 1395, 2005.

[Brett09] C.I. Brett, P.A. DiMillaemail, M. Walker, S. Kim, Joyce Y. Wong, Vascular Smooth Muscle Cell Durotaxis Depends on Substrate Stiffness Gradient Strength. *Biopysical Jounal*, 97, 1313-1322, 2009.

[Isenberg09] C.B. Isenberg, P.A. DiMilla, M. Walker, S. Kim, J.Y. Wong, Vascular Smooth Muscle Cell Durotaxis Depends on Substrate Stiffness Gradient Strength. *Biopysical Jounal*, 97, 1313-1322, 2009.

[Campbell05] P.G. Campbell, E.D. Miller, G.W. Fisher, L.M. Walker, L.E. Weiss, Engineered spatial pattern of FGF-2 immobilized on fibrin direct cell organization. *Biomaterials*, 26(33), pp.6762-6770, 2005.

[Carpenter06] A.E. Carpenter, T.R. Jones M.R. Lamprecht, C. Clarke, I.H. Kang, O. Friman, D.A. Guertin, J.H. Chang, R.A. Lindquist, J. Moffat, P. Golland, and D.M. Sabatini, CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biology*, 7, R100, 2006.

[Chan11] C.C. Chang, and C.J. Lin, LIVSVM : a library for support vector machines. *ACM Transactjions on Intelligent Systems and Technology*, 2(3):27:1-27:27, 2011. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm

[Chan01] T.F. Chan, and L.A. Vese, Active contours without edges. *IEEE Trans. on Image Processing*, 10, pp.266-277, 2001.

[Chenouard10] N. Chenouard, I. Bloch, and J.-C. Olivo-Marin, Multiple-hypothesis tracking in microscopy images. *Proceedings of IEEE International Symposium on Biomedical Imaging(ISBI)*, pp. 1346-1349, 2010.

[Chisholm00] A.D. Chisholm, Cell lineage. *Encyclopedia of Genetics. Academic Press*, 2000.

[ChunChi07] L. Chun-Chi, A.Y. park, J.L. Guan, In vitro scratch assay: a convenient and inexpensive method for analysis of cell migration in vitro. *Nature Protocols*, 2(2), 329-332, 2007.

[Chunming05] T. Chunming, B. Ewert, Segmentation and tracking of neural stem cell. *Advances in Intelligent Computing*, pp.851-859, 2005.

[Costa01] L. Fontoura-Costa, R.M. Cesar, Shape analysis and classification. *CRC Press*, 2001.

[Cortes95] C. Cortes, and V. Vapnik, Support vector network. *Machine Learning*, 20, 273-297, 1995.

[Cour06] T. Cour, P. Srinivasan, and J. Shi, Balanced graph matching. *Proceedings of Neural Information Processing Systems (NIPS)*, 313-320, 2006.

[Debeir05] O. Debeir, P.V. Ham, R. Kiss, C. Decaestecker, Tracking of migrating cells under phase-contrast video microscopy with combined mean-shift processes. *IEEE Trans. on Medical Imaging*, 24(6), pp.697-711, 2005.

[Degerman06] J. Degerman, J. Faijerson, K. Althoff, T. Thorlin, J.H. Rodriguez, T. Gustavsson, A Comparative study between level set and watershed image segmentation for tracking stem cells in time-lapse microscopy. *Proceeding of Workshop of Microscopic Image Analysis with Applications in Biology*, pp.1-5, 2006.

[Drish01] J. Drish, Obtaining calibrated probability estimates from support vector machines. *Technical report, University of California*, San Diego, June 2001.

[Dufour05] A. Dufour, V. Shinin, S. Tajbaksh, N. Guillen, J.-C. Olivo-Marin, and C. Zimmer, Segmenting and tracking fluorescent cells in dynamic 3-D microscopy with coupled active surfaces. *IEEE trans. on Image Processing*, 14, 9, pp. 1396-1410, 2005.

[Dufour11] A. Dufour, R. Thibeaux, E. Labruyere, N. Guillen, and J.C. Olivo-Marin, 3-D Active meshes: fast discrete deformable models for cell tracking in 3-D time-lapse microscopy. *IEEE trans. on Image Processing*, 20, 7, pp. 1925-1937, 2011.

[Dzyubachyk10] O. Dzyubachyk, W.A. Cappellen, J. Essers, W.J. Nissen, and E. Meijering, Advanced level-set-based cell tracking in time-lapse fluorescence microscopy. *IEEE trans. on Medical Imaging*, 29, 3, pp. 852-867, 2010.

[Eom10] S. Eom, R. Bise, and T. Kanade, Detection of hematopoietic stem cells in microscopy images using a bank of ring filters. *Proceedings of IEEE International Symposium on Biomedical Imaging(ISBI)*, pp.137-140, 2010.

[FontouraCosta01] L. Fontoura-Costa, R.M. Cesar, Shape analysis and classification. *CRC Press*, 2001.

[Fortmann83] T. Fortmann, Y.B. Shalom, and M. Scheffe, Sonar tracking of multiple targets using joint probabilistic data association. *IEEE J. Oceanic Engineering*, 8, 173-184, 1983.

[Gu99] J. Gu, M. Tamaru, R. Pankov, E.H. Danen, T. Takino, K. Matsumoto, K.M. Yamada, Shc and FAK Differentially Regulate Cell Motility and Directionality Modulated by PTEN. *Cell Biology*, 146, 390-403, 1999.

[Hand09] A.J. Hand, T. Sun, D.C. Barber, D.R. Hose, S. MacNeil, Automated Tracking of Migrating Cells in Phase-contrast Video Microscopy Sequences using Image Registration. *Journal of Microscopy*, 234, pp. 62-79, 2009.

[House09] D. House, M.L. Walker, W. Zheng, Y. Joyce, W. Betke, M. Betke, Tracking of cell populations to understand their spatio-temporal behavior in response to physical stimuli. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshop*, 2009.

[Huang09] C. Huang, B. Wu, and R. Nevatia, Robust Object Tracking by Hierarchical Association of Detection Responses. *Proceedings of European Conf. on Computer Vision*, p788-801, 2008.

[Huh11] S. Huh, D.F.E. Ker, R. Bise, M. Chen, and T. Kanade, Automated mitosis detection of stem cell populations in phase-contrast microscopy images. *IEEE Trans. on Medical Imaging*, 30(3), pp. 586—596, 2011.

[Huh11b] S. Huh, S. Eom, R. Bise, Z. Yin, and T. Kanade, Mitosis detection for stem cell tracking in phase-contrast microscopy images. *Proceedings of IEEE International Symposium on Biomedical Imaging(ISBI)*, pp. 2121-2127, 2011.

[Indhumathi11] C. Indhumathi, Y.Y. Cai, Y.Q. Guan, and M. Opas, An automatic segmentation algorithm for 3D cell cluster splitting using volumetric confocal images. *Journal of Microscopy*, 243, pp. 60—76, 2011.

[Kachouie08] N.N. Kachouie, P. Fieguth, and E. Jervis, Watershed deconvolution for cell segmentation. *Proceedings of Engineering in Medicine and Biology Society (EMBS)*, 2008.

[Kanade11] T. Kanade, Z. Yin, R. Bise, S. Huh, and S. Eom, Cell image analysis: Algorithms, system and applications. *Proceedings of IEEE Workshop on Applications of Computer Vision (WACV)*, 2011.

[Kausler12] B. Kausler, M. Schiegg, B. Andres, M. Lindner, and U. K. *et. al.*, A discrete chain graph model for 3d+t cell tracking with high misdetection robustness. *Proceedings of European Conf. on Computer Vision*, pp. 144–157, 2012.

[Keuper11] M. Keuper, T. Schmidt, M.R. Franco, W. Schamel, T. Brox, H. Burkhardt, and O. Ronneberger, Hierarchical Markov random fields for mast cell segmentation in electron microscopic recordings. *Proceedings of IEEE International Symposium on Biomedical Imaging(ISBI)*, pp. 973—978, 2011.

[Kimmel95] C.B. Kimmel, W.M. Ballard, S.R. Kimmel, B. Ullmann, and T.F. Schilling, Stages of embryonic development of the zebrafish. *Dev. Dyn*, 203, 253-310, 1995.

[Kofahi06] O. Al-Kofahi, R. Radke, S. 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), 327-335, 2006.

[Kron08] Matlab Central, B-spline grid, image and point based registration. http://www.mathworks.jp/matlabcentral/fileexchange/20057-b-spline-grid-image-and-point-based-registration, 26 May 2008.

[Kuhn55] H.W. Kuhn, The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2, 83-87, 1955.

[Lee11] J. Lee, M. Cho, and K.M. Lee, Hyper-graph matching via reweighted random walks. *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR),* 2011.

[Lee97] S. Lee, G. Wolberg, and Y. Shin, Scattered data interpolation with multilevel B-splines. *IEEE Trans. on Visualization Comput. Graph.*, vol. 3, 228-244, 1997.

[Lesko10] M. Lesko, Z. Kato, A. Nagy, I. Gombos, Z. Torok, L. Vigh, and L. Vigh, Live Cell Segmentation in Fluorescence Microscopy via Graph Cut. *Proceedings of International Conference on Pattern Recognition (ICPR),* 2010.

[Li10] C. Li, C. Xu, C. Gui, and M.D. Fox, Distance regularized level set evolution and its application to image segmentation. *IEEE Trans. on Image Processing*, 19(12), pp.3243-3254, 2010.

[Li08] K. Li, E.D. Miller, M. Chen, T. Kanade, L.E.Weiss, and P.G. Campbell, Cell population tracking and lineage construction with spatiotemporal context. *Med Image Anal*, 12(5), 546-566, 2008.

[Li08b] K. Li, E.D. Miller, M. Chen, T. Kanade, L.E.Weiss, and P.G. Campbell, Computer vision tracking of stemness. *Proceedings of IEEE International Symposium on Biomedical Imaging(ISBI)*, pp. 847—850, 2008.

[Li09] K. Li, T. Kanade, Nonnegative mixed-norm preconditioning for microscopy image segmentation. *Proceedings of Int. Conf. Information Processing in Medical Imaging (IPMI)*, pp.362-373, 2009.

[Lin05] G. Lin, M.K. Chawla, K. Olson, J.F. Guzowski, C.A. Barnes, and B. Roysam, Hierarchical, model-based merging of multiple fragments for improved three-dimensional segmentation of nuclei. *Cytometry A*, 63, pp. 20—33, 2005.

[Liu10] A. Liu, K. Li, and T. Kanade, Mitosis sequence detection using hidden conditional random fields. *Proceedings of IEEE International Symposium on Biomedical Imaging(ISBI)*, pp. 580—583, 2010.

[Long07] X. Lou, U. Koethe, J. Wittbrodt, F.A. Hamprecht, Automatic segmentation of nuclei in 3D microscopy images of C. Elegans. *Proceedings of IEEE International Symposium on Biomedical Imaging(ISBI)*, pp. 536—539, 2007.

[Lou12] F. Long, H, Peng, and E. Myers, Learning to segment dense cell nuclei with shape prior. *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp.1012—1018, 2012.

[Matas02] J. Matas, O. Chum, M. Urban, and T. Pajdla, Robust wide baseline stereo from maximally stable extremal regions. *Proceedings of British Machine Vision Conference*, pp. 384-396, 2002.

[Mattes99] J. Mattes, M. Richard, and J. Demongeot, Tree representation for image matching and object recognition. *Lecture Notes in Computer Science*, 1568, pp. 298-309, 1999.

[Maska04] M. Maska, V. Ulman, D. Svoboda, Pavel. Matula, Petr Matula, C. Ederra, *et. al.*, A benchmark for comparison of cell tracking algorithms. *Bio Infomatics*, 30, 11, pp. 1609-1617, 2014.

[Meijering09] E. Meijering, O. Dzyubachyk, I. Smal, W.A. Cappellen, Tracking in Cell and Developmental Biology. *Seminars in Cell and Developmental Biology*, 20(8), pp. 894-902, 2009.

[Mcinerney96] T. Mcinerney, and D. Terzopoulos, Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1(2), pp.91-108, 1996.

[Milan13] A. Milan, K. Schindler, and S. Roth, Challenges of ground truth evaluation of multi-target tracking. *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) Workshop*, pp.735-742, 2013.

[Milios00] E. Milios, E.G.M. Petrakis, Shape retrieval based on dynamic programming. *IEEE Trans. on Image Processing*, 9(1), pp.141-147, 2000.

[Mingqiang08] Y. Mingqiang, K. Kidiyo, and R. Joseph, A survey of shape feature extraction techniques. *Proceedings of Pattern Recognition Techniques, Technology and Applications, Vienna: i-Tech*, 2008.

[Nikolic06] L.D. Nikolic, A.N. Boettiger, D. Bar-Saqi, J.D. Carbeck, S.Y. Shvartsman, Role of boundary conditions in an experimental model of epithelial wound healing. *Physical Cell Physiol*, 291, pp.68-75, 2006.

[Okuma04] K. Okuma, A. Taleghani, N.D. Freitas, L.J. James and D.G. Lowe, A boosted particle filter: multitarget detection and tracking. *Proceedings of European Conf. on Computer Vision*, pp.11-14, 2004.

[Otsu79] N. Otsu, A threshold selection method from gray-level histograms. *IEEE Trans. on Systems, Man, and Cybernetics*, 9(1), pp.62-66, 1979.

[Padfiled09] D. Padfield, J. Rittscher, N. Thomas, and B. Roysam, Spatio-temporal cell cycle phase analysis using level sets and fast marching methods. *Medical Image Analysis*, 13(1), pp.143-155, 2009.

[Padfield11] D. Padfield, J. Rittscher, and B. Roysam, Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis. *Medical Image Analysis*, 15(4), 650-668, 2011.

[Padfield11b] D. Padfield *et al.*, Quantitative biological studies enabled by robust cell tracking. *Proceedings of IEEE International Symposium on Biomedical Imaging(ISBI)*, 2011.

[Pan10] D. Padfield *et al.*, Heterogeneous conditional random field: realizing joint detection and segmentation of cell regions in microscopic images. *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp.2940—2947, 2010.

[Papadimitriou98] C.H. Papadimitriou, and K. Steiglitz, Combinatorial optimization: algorithms and complexity. *Dover Publications*, 1998.

[Petrakis02] E.G. Petrakis, A. Diplaros, E. Milios, Matching and retrieval of distorted and occluded shapes using dynamic programming. *IEEE Trans. on Pattern Anal. Machine Intell*, 24(11), 1501-1516, 2002.

[Peng09a] H. Peng, X. Zhou, F. Li, X. Xia, and S.T.C. Wong, Integrating multi-scale blob/curvilinear detection techniques and multi-level sets for automated segmentation of stem cell images. *Proceedings of IEEE IEEE International Symposium on Biomedical Imaging(ISBI)*, pp. 1362-1365, 2009.

[Peng09b] J.Y. Peng, and C.N. Hsu, Adaptive local thresholding for fluorescence cell micrographs. *echnical Rep*, No. TR-IIS-09-008, Nov. 11, 2009.

[Praveen06] S. Praveen, and A.K. Vijayan, Adaptive thresholding Based Cell Segmentation for Cell-Destruction Activity Verification. *Proceeding of the 35th Applied Imagery and Pattern Recognition Workshop*, 2006.

[Pril09] R. Pril, T. Perera, A. Lekkerkerker, A high-content screen for inhibitors of cell migration in cancer metastasis using adenoviral knock-down. *BTi* April, 12, 385-395, 2009.

[Reid79] D. Reid, An algorithm for tracking multiple targets. *IEEE Trans. on Automatic Control*, 24, 843-854, 1979.

[Ravin08] R. Ravin, J.D. Hoeppner, D.M. Munno, L. Carmel, J. Sullivan, D.L. Levitt, J.L. Miller, C. Athaide, D.M. Panchision, R.D. McKay, Potency and fate specification in CNS stem cell populations in vitro. *Cell Stem Cell*, 3, pp.670-680, 2008.

[Rittscher10] J. Rittscher, Characterization of Biological Processes through Automated Image Analysis. *Annual Review of Biomedical Engineering*, 12, pp. 315-344, 2010.

[Ross75] G.T. Ross, A branch and bound algorithm for the generalized assignment problem. *Mathematical Programming*, 8, pp.91-103, 1975.

[Rueckert99] D. Rueckert, L.I. Sonoda, C. Hayes, D.L.G. Hill, M.O. Leach, and D.J. Hawkes, Nonrigid registration using free-form deformations: application to breast MR images. *IEEE Trans. on Medical Imaging*, vol.8, 712-721, 1999.

[Schiegg13] M. Schiegg, P. Hanslovsky, B. X. Kausler, L. Hufnagel, and F. Hamprecht, Conservation tracking. *IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 2928–2935, 2013.

[Schmid13] B. Schmid, G. Shah N. Scherf, M. Weber, K. Thierbach, C.P. Campos, I. Roeder, P. Aanstad, and J. Huisken, High-speed panoramic light-sheet microscopy reveals global endodermal cell dynamics. *Nature Communications*, 3207, 2013.

[Schindelin12] J. Schindelin, I.A. Carreras, E. Frise, V. Kaynig, and M. Longair *etc.*, FIJI: an open-source platform for biological-image analysis. *Nature Methods*, 9, 676-682, 2012.

[Smal06] I. Smal, W. Niessen, and E. Meijering, Bayesian tracking for fluorescence microscopic imaging. *Proceedings of IEEE IEEE International Symposium on Biomedical Imaging(ISBI)*, pp. 550—553, 2006.

[Tamura98] M. Tamura, K. Matsumoto, S. Aota, R. Parsons, K.M. Yamada, Inhibition of Cell Migration, Spreading, and Focal Adhesions by Tumor Suppressor PTEN. *Science*, 280, 1614-1617, 1998.

[Tcheslavski13] G.V. Tcheslavski, Morphological image processing: gray-scale morphology. *Retrieved*, 4, 2013.

[Tse09] S. Tse, L. Bradbury, J.W.L. Wan, H. Djambazian, R. Sladek, and T. Hudson, A combined watershed and level set method for segmentation of brightfield cell images *Proceedings of SPIE Symposium on Medical Imaging: Image Processing*, vol 7259, 2009.

[Vincent91] L. Vincent, and P. Soille, Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Trans. on Pattern Analysis*, 13(6), pp.583-598, 1991.

[Wang07] X. Wang, W. He, D. Metaxas, R. Mathew, and E. White, Texture-adaptive snakes used in cell segmentation and tracking. *Proceedings of IEEE International Symposium on Biomedical Imaging(ISBI)*, pp. 101-104, 2007.

[Wolsey98] L.A. Wolsey, Integer programming. New York, J Wiley, 1998.

[Wood13] S.T. Wood, B.C. Dean, and D. Dean, A linear programming approach to reconstructing subcellular structures from confocal images for automated generation of representative 3D cellular models. *Medical Image Analysis*, 17, pp. 337-347, 2013.

[Wu04] T.F. Wu, C.J. Lin, and R.C. Weng, Probability estimates for multi-class classification by pairwise coupling. *Machine Learning*, 5, pp.975-1005, 2004.

[Xiao10] J. Xiao, H. Cheng, H. Sawhney, and F. Han, Vehicle detection and tracking in wide field-of-view aerial video. *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 679-684, 2010.

[Xiaoxu07] W. Xiaoxu, H. Weijun, and M. Dimitris, Cell segmentation and tracking using texture adaptive snakes. *Proceedings of IEEE International Symposium on Biomedical Imaging(ISBI)*, 101-104, 2007.

[Yang05] F. Yang, M. Mackey, F. Ianzini, G. Gallardo, and M. Sonka, Cell segmentation, tracking, and mitosis detection using temporal context. *Proceedings of Medical Image Computing and Computer Assisted Intervention(MICCAI)*, 302-309, 2005.

[Yarrow04] C.J. Yarrow, Z.E. Perlman, N.J. Westwood, T.J. Mitchison, A high-throuput cell migration assay using scratch wound healing, a comparison of image-based readout methods. *BMC Biotechnology*, 4(21), 2004.

[Yu08] W. Yu, H.K. Lee, S. Hariharan, W. Bu, and S. Ahmed, Level set segmentation of cellular images based on topological dependence. *Advances in Visual Computing Lecture Notes in Computer Science*, Volume 5358, pp.540-551, 2008.

[Yin10a] Z. Yin, K. Li, T. Kanade, M. Chen, Understanding the Optics to Aid Microscopy Image Segmentation. *Proceedings of Medical Image Computing and Computer Assisted Intervention(MICCAI)*, 13(1), 209-217 2010.

[Yin10b] Z. Yin, R. Bise, M. Chen, and T. Kanade, Cell Segmentation in Microscopy Imagery Using a Bag of Local Bayesian Classifiers. *Proceedings of IEEE International Symposium on Biomedical Imaging(ISBI)*, pp.125-128, 2010.

[Yin12] Z. Yin, K. Li, T. Kanade, M. Chen, Understanding the phase contrast optics to restore artifact-free microscopy images for segmentation. *Medical Image Analysis*, 16(5), pp.1047-1062, 2012.

[Zahm97] J.M. Zahm, A.L. Herard, F. Doriot, P. Somelette, E. Puchelle, Cell Migration and Proliferation During the In Vitro Wound Repair of the Respiratory Epithelium. *Cell Motility and the Cytoskeleton*, 37, 33-43, 1997.

[Zass08] R. Zass, and A. Shashua, Probabilistic graph and hyper graph matching. *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[Zhang04] B. Zhang, C. Zimmer, and J.-C. Olivo-Marin, Tracking fluorescent cells with coupled geometric active contours. *Proceedings of IEEE International Symposium on Biomedical Imaging(ISBI)*, pp. 476—479, 2004.

[Zhang08] C. Zimmer, E. Labruyere, V. Meas-Yedid, N. Guillen, and J.-C. Olivo-Marin, Global data association for multi-object tracking using network flows. *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[Zimmer02] R. Zass, and A. Shashua, Segmentation and tracking of migrating cells in video microscopy with parametric active contours: A tool for cell-based drug testing. *IEEE Trans. on Medical Imaging*, 21(10), pp. 1212-1221, 2002.

[Zuiderveld94] K. Zuiderveld, Contrast limited adaptive histograph equalization. *Graphics gems IV*, Academic Press Professional, Inc., pp.474-485, 1994.

Publications

Journals

1. R. Bise and Y. Sato, "Cell Detection from Redundant Candidate Regions under Non-Overlapping Constraints", IEEE Trans. Medical Imaging, (accepted).

2. R. Bise, Y. Sato, A. Kondo, T. Kobayashi, K. Ohnuma, and S. Nonaka, "3D Cell Tracking under Dense Cell Culture Conditions using Global Motion and Local Spatial Relationships", IEEE Trans. Medical Imaging, (under review).

Refereed conferences

1. R. Bise, Y. Maeda, M.H. Kim, and M. Kino-oka, "Cell Tracking Under High Confluency Conditions by Candidate Cell Region Detection Based Association Approach," Proceedings of IASTED BioMed 2013.

2. R. Bise, T. Kanade, Z. Yin, and S. Huh, "Automatic Cell Tracking Applied to Analysis of Cell Migration in Wound Healing Assay", Proceedings of IEEE EMBC, pp. 6174-6179, 2011

3. R. Bise, Z. Yin, and T. Kanade, "Reliable Cell Tracking by Global Data Association", Proceedings of IEEE ISBI, pp. 1004-1010, 2011

4. R. Bise, K. Li, S. Eom, and T. Kanade, "Reliably Tracking Partially Overlapping Neural Stem Cells in DIC Microscopy Image Sequences", Proceedings of MICCAI Workshop on OPTMHisE, 2010.

Others

1. Ryoma Bise, Yoichi Sato, Akiko Kondow, Tetsuya Kobayashi, Kiyoshi Okuma, "Cell tracking under dense cell culture conditions by preserving the structure of neighbor cells", International Symposium on Biomedical Imaging (ISBI 2014), poster, April 2014.

2. 備瀬 竜馬他，”密な細胞画像を対象とした細胞検出及びトラッキング手法の提案”，バイオイメージインフォマティクス，2014

3. 備瀬 竜馬他，”冗長な細胞候補領域の中からの最適な細胞群選択による細胞検出手法の提案”，MIRU，2014

4. 備瀬 竜馬他，”大局的細胞移動推定及び局所的相対位置関係類似度を用いた密な状態における 3 次元細胞トラッキング”，MIRU，2013

5. 備瀬 竜馬他，”密な状況での細胞群の 3 次元追跡”，第 13 回生命科学シンポジウム，2013 年，優秀ポスター賞受賞

6. 備瀬 竜馬他，”密な状況での細胞群の 3 次元追跡”，第 5 回定量生物学の会

7. 備瀬 竜馬，”細胞培養画像解析技術による細胞挙動評価”，第 63 回日本生物工学会，ワークショップ 2011．招待公演

8. 備瀬竜馬他，”第 6 章　画像解析による培養品質管理”，書籍「再生医療事業の課題解決のための手引書」，技術情報協会，2013 年 9 月

9. 備瀬竜馬他，”第 42 章　画像解析による培養品質管理”，書籍「幹細胞医療の実用化技術と産業展望」，シーエムシー出版，2012 年 12 月

10. 備瀬竜馬，”品質管理が開く再生医療 - 画像解析による培養品質管理”，医療機器学会誌，81(6)，459 － 464，2011 年