

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Feature Extraction in Volumetric Bioimages

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

by

Min Liu

June 2012

Dissertation Committee:

Professor Amit K. Roy-Chowdhury, Chairperson
Professor Venugopala Gonehal Reddy
Professor Ertem Tuncel

Copyright by
Min Liu
2012

The Dissertation of Min Liu is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

First and foremost I would like to express my heartfelt gratitude to my advisor, Dr. Amit Roy-Chowdhury, who guided me through the research in the fields of computer vision and image processing. He has taught me, both consciously and unconsciously, how to become a sincere PH.D. student and passionate researcher. I appreciate his great contributions of time, ideas and funding to make my Ph.D. experience stimulating and productive. I am also thankful for the excellent example he has provided as an enthusiasm and successful professor.

I am grateful to my committee members, Dr. Venugopala Gonhal Reddy and Dr. Ertem Tuncel. Dr. Reddy has provided me extremely valuable datasets to work with and in the meantime given me insightful advises on my PH.D. project from a biological perspective. I really appreciate every effort he has made in advising me in my Ph.D. pursuit. Thanks to Dr. Tuncel for giving me valuable and helpful suggestions to my oral exam, my thesis and my defense. And it is really enjoyable to attend his classes. Also, I would like to thank Dr. Stefano Lonardi for giving me thoughtful feedback and constructive comments on my oral exam, thesis and defense from the perspective of bioinformatics. Thanks to Dr. Paul De Ley for providing me the nematode data-set to work with and giving me valuable suggestions on this project.

The members of the video-computing group have contributed immensely to my professional and personal time at University of California, Riverside. I thank all my fellow lab-mates. It is because you are there, the group has been a source of friendships as well as good advice, discussion and collaboration, and also it is an important source of fun for me.

Lastly, I would like to thank my parents for giving birth to me at the first place and supporting me spiritually and unconditionally throughout my life. I owe them everything and wish I

could show them just how much I love and appreciate them. My wife, Yingxin, whose love, faithful and patient support allowed me to finish this journey. Without her help and encouragement, this study would not have been completed.

ABSTRACT OF THE DISSERTATION

Feature Extraction in Volumetric Bioimages

by

Min Liu

Doctor of Philosophy, Graduate Program in Electrical Engineering
University of California, Riverside, June 2012
Professor Amit K. Roy-Chowdhury, Chairperson

Automatic analysis of biological image datasets is one of the important achievements of applied image processing research. Feature extraction in volumetric bioimages obtained from numerous biomedical imaging techniques is becoming extremely critical for biologists and medical professionals to find the answers to many problems. In this work, we show how to extract effective features in two kinds of volumetric bioimages - plant shoot apical meristem (SAM) cell images taken by Confocal Laser Scanning Microscopy (CLSM) and the nematode images taken by Differential Interference Contrast Microscopy.

For actively developing tissues, a computational platform capable of automatically segmenting and tracking cells in the volumetric image stacks is very critical to obtaining high-throughput and quantitative spatiotemporal measurements of a range of cell behaviors, which will lead to a better understanding of the underlying dynamics of morphogenesis. The cells in the SAM are tightly clustered in space and have very similar shapes and intensity distributions, thus choosing reliable features to compute cell correspondences in space and time can be very challenging. In our research we propose a local graph matching based method to track the cells both spatially and temporally, and

identify cell divisions. The geometric structure and topology of the cells' relative positions are efficiently exploited as the basic feature to match the cells. Furthermore, we build a joint segmentation and tracking system, where the tracking output acts as an indicator of the quality of segmentation and, in turn, the optimized segmentation can be improved to obtain better tracking results. In the end, the cell correspondences across multiple slices and time windows are fused together to obtain the final cell lineages. Experiments on multiple plant datasets show the proposed image analysis pipeline can effectively segment and track the SAMs cells.

Another contribution of this work on volumetric bioimages analysis lies in multilinear feature extraction and classification for nematode Digital Multi-focal Images (DMI). In such images, morphological information for a transparent specimen is captured in the form of a stack of high quality images, representing individual focal planes through the specimen's body. We present a method that can effectively exploit the entire information in the stack using the 3D X-Ray projections at different viewing angles. These DMI stacks represent the effect of different factors - shape, texture, viewpoint, different instances within the same class and different classes of specimens. For this purpose, we embed the 3D X-Ray Transform within a multilinear framework and propose a Multilinear X-Ray Transform feature representation. The experimental results on the nematode DMI data show that the proposed feature extraction and analysis method can give reliable recognition rate on a real-life database.

Contents

List of Figures	x
List of Tables	xv
1 Introduction	1
1.1 Volumetric Bioimages	2
1.1.1 Plant Cell Image Stacks by Confocal Laser Scanning Microscopy	2
1.1.2 Nematode Image Stacks by Differential Interference Contrast Microscopy .	5
1.2 Image Analysis Challenges in Volumetric Bioimages	6
1.3 Related Work	8
1.3.1 Plant Cell Tracking	8
1.3.2 Image Stacks Classification	10
1.4 Contribution of This Work	11
1.4.1 Cell Tracking in Volumetric Plant Cell Image Stacks	11
1.4.2 Nematode Image Stacks Classification	13
1.5 Outline of Thesis	14
2 Feature Extraction in Live Cell Imaging of Plant Tissues	15
2.1 Introduction	15
2.2 Live Cell Imaging of Plant Tissues	17
2.3 Proposed Pipeline	20
2.4 Cell Segmentation	21
2.4.1 Level Set Segmentation	21
2.4.2 Watershed Segmentation	23
2.5 Cell Tracking Based on Local Feature Graph	24
2.5.1 From Cells to Graphs	25
2.5.2 Temporal Tracker	26
2.5.3 Spatial Tracker	31
2.6 Experimental Results	32
2.6.1 Choice of The Parameters for The Trackers	32
2.6.2 Preliminary Tracking Results	33

2.6.3	Robustness	34
2.7	Conclusion	35
3	Integrated Segmentation and Tracking for Cell Lineage Computation	39
3.1	Introduction	39
3.2	Adaptive Segmentation and Tracking	40
3.2.1	Design of Integrated Optimization Function	40
3.2.2	Optimization Scheme	45
3.3	Fusion of Spatial and Temporal Tracks	47
3.4	Computing Cell Lineages	49
3.5	Experimental Results	51
3.5.1	Integrated Segmentation and Tracking	51
3.5.2	Estimated Cell Lineages	55
3.5.3	Quantitative Results	57
3.5.4	Statistical Analysis	61
3.6	Conclusion	62
4	A Multilinear Framework for Species Classification Using 3D X-Ray Transform	64
4.1	Introduction	64
4.2	Broad Overview of Solution Strategy	66
4.3	Detailed Methodology	68
4.3.1	3D X-Ray Transform	68
4.3.2	Multilinear Subspace Analysis	71
4.3.3	X-Ray Based Tensor Texture Analysis	72
4.3.4	X-Ray Based Tensor Shape Analysis	74
4.4	Experimental Results	77
4.4.1	Tensor Texture Analysis	77
4.4.2	Tensor Shape Analysis	81
4.4.3	Combination of Tensor Texture and Shape Feature	81
4.4.4	Tensor Analysis on Key Frames	83
4.5	Conclusion	85
5	Conclusion and Future Work	86
5.1	Conclusions	86
5.2	Future Work	87
Bibliography		89

List of Figures

1.1	The development of cell type-specific gene expression map representing three cell types, stem-cells, niche cells, and differentiating cells. The central zone (CZ) of the SAM harbours a set of stem-cells (B and C in the figure). The progeny of stem cells enter the flanking peripheral zone (PZ) (F, G) and the Rib-meristem (RZ) where they differentiate (D,E). The cells of the RZ function as niche cells by providing signals required for maintenance of stem cells in the CZ. Images C, E, and G represent the reconstructed side views of the images B, D, and F respectively.	3
1.2	(A), (C) and (E) are the first time points and the (B), (D) and (F) represent the same SAMs after the elapsed time indicated on top right hand corner of each panel. (A) and (B) show the Z-stack revealing all the cells in the L1 layer. Differentiating organs at different stages of development are marked as P1, P2 and P3. Arrows indicate the same cells prior to and after division. (C) and (D) represent cells in the L2 layer (8 μm deep) and the arrow points to cell division events. (E) and (F) are the cells in the L3 layer/corpus (12 μm deep) and arrow points to cell division events. Scale at the bottom of each image is 20 μm . XYZ represents 3D image as stacks. XYZT represents 3D image across time.	4
1.3	Examples of DMI stacks taken from 3 nematode species (horizontal direction). Each stack contains multiple focal planes (frames) taken from the top to the bottom of the specimen, with only 3 key frames of each shown here (vertical direction).	6
2.1	Multiple layer structure of the SAM.	18

2.2	Time-lapse imagery of SAMs labeled with plasma membrane localized YFP. Examples of three image slices (vertical columns) at three time instants (horizontal columns; arrows A, B and C show same set of cells at time points noted on each panel). Cross-sectional images of SAMs depicted in the vertical columns are separated by $1.5 \mu\text{m}$, while the size of a single cell is about $5 \mu\text{m}$ in diameter. Therefore each cell is represented in two or three consecutive slices (arrows in A, D and G correspond to the optical sections of same cells located in different depths). (A, B and C) belong to slice 5, (D, E and F) belong to slice 6 and (G, H and I) belong to slice 7. Note the representation of several cells in three consecutive sections (arrows).	19
2.3	Overall Imaging and Image Analysis Framework. (A) SAM located at the tip Arabidopsis shoot. (B) Time-lapse imagery of SAMs labeled with plasma membrane localized YFP. Examples of three image slices (vertical columns) at three time instants (horizontal columns). The red circles denote a cell division. (C) The diagram of the cell segmentation and tracking pipeline.	21
2.4	Level-set based segmentation results of SAM cells. Segmentation results (B and E) of the original confocal cross sections acquired at different depths (A and D), and cell centroid detection (C and F) from the segmentation. Note that the cells in the central part of the image (yellow circle) are not segmented properly because of noise.	22
2.5	(A)Original image slice. (B)Watershed segmentation result. (C)Level-set segmentation result.	23
2.6	(A)Original image slice. (B) Over-segmentation case denoted by red circle. (C) Under-segmentation case by red circle.	25
2.7	Local graph matching.	26
2.8	Illustration of the correspondence growing process starting from the seed pair to the neighboring cells. Red colored cells are the seed pair. Yellow colored cells are the matched cells in the first growing iteration. Purple colored cells are the matched cells in the second growing iteration.	29
2.9	A diagram to describe the relative position of the mother cell and the daughter cells.	30
2.10	Tracking results with cell division, where the mother cells are denoted by purple dots in the first frame, while the sibling cells are in white circles in the second frame.	34
2.11	Tracking results across slices in noisy cell images, where some cells in the central part are totally missing.	35
2.12	(A), (C) and (E) are the original image slices obtained at the 21st hour, 24th hour, and 27th hour. (B), (D) and (F) are the level-set segmentation result and the tracking result of the image slices of (A), (C) and (E). The same color denotes the same cell across different time instances.	36

2.13 Robustness of tracker by considering multiple correspondence growing paths. (A, C) are the original image frames before segmentation, while (B, D) represent tracking output including 8 cell division events (denoted by red dots), where the corresponding, non-dividing cells are denoted by the same color. For a dividing cell, the parent cells and corresponding sibling cells are denoted by red color. We show three such paths between cell ‘a’ and cell ‘b’ for two time points. The three paths in red color, yellow color and green color demonstrate the feature of this algorithm that it can automatically grow cell correspondences from one pair to another pair by multiple paths.	37
3.1 Schematic showing on how to integrate the spatial and temporal trackers for 4D volumetric image stacks.	42
3.2 (A) The illustration of the number of faulty cell merging events (green plot) or spurious cell divisions (red plot) with respect to the changing of H-minima threshold h , in one noisy image slice. (B) The total number of over-segmentation and under-segmentation errors with respect to the changing of H-minima threshold h	44
3.3 Adaptive segmentation and tracking scheme for a certain image slice S_k^t (the k^{th} slice at the t time point).	45
3.4 The diagram showing integration of the tracking results in 3 consecutive slices by fusing the spatial and temporal tracks to compute the final cell correspondence.	48
3.5 The tracking results after the spatial and temporal track fusion process, where nine newly tracked cells are denoted by white circles.	49
3.6 Cell lineage (from time t_1 to t) computation process by fusing along multiple tracking paths (Path 1, Path 2 and Path 3).	50
3.7 (A, B) Cell lineages from time 6 hr to 18 hr, computed by Path 1. (C, D) Cell lineages computed by Path 2. (D, F) Cell lineages computed by the combination of Path 1 and Path 2. Note that the system may lose correspondence of some cells along a particular path, but maintained their correspondences in an alternate tracking path. Compare (A), (C) and (D), cells in green circles were lost in (A) and (B) while cells in red circles were lost in (C) and (D), but all of them were gained back in (E) and (F) due to the fusion process.	52
3.8 Computed cell lineages of SAM cells. The cell lineages across 24 time instants (a total of 72 hours of time lapse data) were computed (only 6 of them are show here), where the same color denotes the cell lineage. After 72 hours of tracking, most of cells have divided at least once with several of them divided twice or thrice.	53

3.9 Illustration of faulty cell merging events caused by under-segmentation or spurious cell divisions caused by over-segmentation. (A) The fourth slice of confocal Z-stack taken at 30th hr time point. (B) The correct segmentation result with the optimal H-minima threshold value 0.05, which was found by the proposed adaptive method; (C) and (D) are the temporal tracking results (from 30th hr time point to the 27th hr time point) with faulty cell merging events (denoted by purple dots), which are caused by under-segmentation with H-minima threshold value 0.065. (E) and (F) are the spatial tracking results (from 3rd confocal slice to the 4th confocal slice along the z-scale) with spurious cell divisions (denoted by purple dots) and this is caused by over-segmentation. The same number denotes two corresponding slices along the Z-direction.	54
3.10 An example of the estimated cell lineages. The first row is the original image slice series at the 9th hr, 15th hr, 21st hr, 27th hr and 33rd hr. The second row shows the segmented cells and the estimated cell lineages using the adaptive segmentation and tracking algorithm. The third row is the result using the level set segmentation, and cell segmentation and tracking are not integrated. The same color in consecutive frames denotes the same cell. Cell division examples are denoted by red circles in the second row.	56
3.11 The segmentation and tracking results in 3D stacks at selected time instances. The segmented cells (by watershed method) shown in same color across consecutive slices (2nd slice, 3rd slice and 4th slice) represent same cells.	57
3.12 Total number of automatically-tracked cells across different time instants measuring up to 72 hours. Note the stability of tracking algorithm as most of the cells identified at first time instant could be detected at later time instants. The sudden decrease in the number of correctly tracked cells at 36 hours is mostly due to the incorporation of cells located at the edge of the PZ and subsequent displacement of cells into actively growing organ primordia.	59
3.13 Statistical analysis results based on the segmentation and tracking results using our proposed adaptive method. (A) The histogram showing cell cycle length. (B) The number of cell divisions across each 3 hour intervals.	62
4.1 The diagram for training and recognition of the proposed 3D X-Ray Transform based multilinear analysis method.	67
4.2 The illustration of the three dimensional X-Ray Transform.	69
4.3 3D X-Ray Transform output images (B-F) at 5 poses (angle pairs) ($1(\pi/2, \pi/2)$, $2(0, \pi/2)$, $3(\pi, \pi)$, $4(\pi/5, \pi/3)$ and $5(\pi/3, \pi/5)$) respectively, of the 1st DMI stack shown in Figure 1.2. A is the illustration of different poses around a given DMI stack for the 3D X-Ray Transform.	70

List of Tables

2.1	Comparison of the number of cell division events between manual tracking (both in the unsegmented data and the segmented data by Level-set method, the latter being in parenthesis), and the automated method used in this study. The data for different slices is shown.	38
3.1	Cells being tracked manually (in the unsegmented data and in the segmented data by Level-Set method, the latter is in the brackets) and automatically (before the fusion process and after the fusion process). We have chosen some representative examples from the dataset.	58
3.2	Total numbers of cells and cell divisions tracked in the stack of the top 12 slices in different time intervals. The denominator is the ground truth in the segmented data by Level-Set method (the number of cells tracked manually or the number of cells divisions identified manually); the numerator is the tracking result obtained from the algorithm developed in this study.	59
3.3	The number of cells correctly tracked using level-set segmentation, the watershed segmentation but no adaptation, and the integrated segmentation and tracking method shown in the left column.	60
3.4	The average time period over which the lineages could be tracked is shown in the right column.	61
4.1	Recognition rate comparison between different classification methods. The 4th-7th rows are the recognition rates on the training data composed of 4 samples from each class, the 8th row is the recognition rate on the training data of another 4 samples from each class, while the 9th row is the recognition rate on the training data of 5 samples from each class. The training and testing data for the first 3 rows are described in the text.	84

Chapter 1

Introduction

Automatic processing and analysis of biological image datasets is one of the most important achievements of applied image processing research [1, 2, 3, 4]. There are numerous biomedical imaging techniques developed in the last several decades, ranging from the whole organism level down to the single molecule level [5, 6, 7, 8], and the number of biological images taken by such imaging methods is growing rapidly [2]. Usually, these image data contains 3D spatial information or 4D spatio-temporal information for developing tissues or moving cells [2, 9, 10, 11]. Such kind of volumetric images consisting of stacks of two-dimensional images are usually called bioimages [2]. The field of image processing on volumetric bioimages is called bioimage informatics, which is becoming extremely critical for biologists to find the answers to many biological problems [2].

In this thesis, we are dealing with two popular kinds of volumetric microscopy bioimages. The first one is shoot apical meristem plant cell images taken by Confocal Laser Scanning Microscopy, which is usually used to image opaque specimens and mostly at the cellular level [12, 13]. The other one is nematode images taken by Differential Interference Contrast Microscopy,

which is usually used to take transparent specimens and mostly at the organism level [14]. We will show how to extract effective features from those two kinds of bioimage stacks with specific applications in each field.

1.1 Volumetric Bioimages

1.1.1 Plant Cell Image Stacks by Confocal Laser Scanning Microscopy

The shoot apical meristems, also referred to as the stem-cell niche, is the most important part of the plant body because it supplies cells for all the above ground plant parts such as leaves, branches and stem. SAMs of model plant *Arabidopsis thaliana* consist of approximately 500 cells (at inflorescence stages) and they are organized into distinct spatial domains as well as multiple cell layers that are clonally distinct from each other. Usually, the SAM cells are imaged by the confocal laser scanning microscopy, which is a technique for obtaining high-resolution optical images with depth selectivity [6]. Usually, the CLSM setup combines the laser scanning technique with the 3D detection of biological objects labeled with fluorescent markers. It can acquire in-focus images from certain depths, by a process called optical sectioning.

The multi-layer structure of SAMs is shown in Figure 1.1. The central zone (CZ) of the SAM harbors a set of stem-cells that divide at a relatively slower rate. The progeny of stem-cells enter the flanking peripheral zone (PZ) and also the Rib-meristem (RM) which is located beneath the CZ where they differentiate [12]. Apart from this radial organization, the SAM of *Arabidopsis* is a multilayered-structure, organized into three clonally distinct layers of cells. The cells in the outermost L1 layer and the sub-epidermal L2 layer divide in anticlinal orientation (perpendicular to

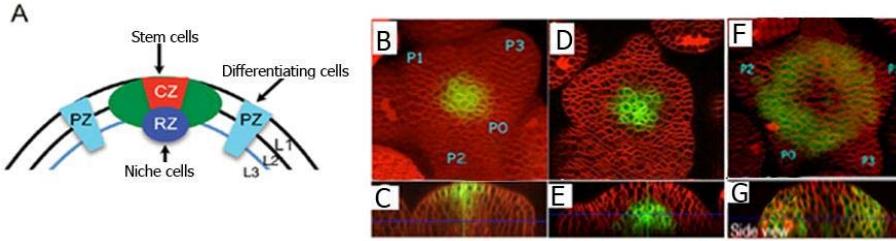


Figure 1.1: The development of cell type-specific gene expression map representing three cell types, stem-cells, niche cells, and differentiating cells. The central zone (CZ) of the SAM harbours a set of stem-cells (B and C in the figure). The progeny of stem cells enter the flanking peripheral zone (PZ) (F, G) and the Rib-meristem (RZ) where they differentiate (D,E). The cells of the RZ function as niche cells by providing signals required for maintenance of stem cells in the CZ. Images C, E, and G represent the reconstructed side views of the images B, D, and F respectively.

the SAM surface), while the underlying corpus forms a multi-layered structure where cells divide in random planes. Thus, the SAM stem-cell niche represents a dynamic and interacting network of functionally distinct cell types, exhibiting a coordination between cell division dynamics and displacement of the progeny cells, both within and across clonally distinct layers [12]. Despite a continuous displacement and diversion of cells into a differentiation program, the size of stem-cell niche remains relatively constant. Therefore, a tight spatio-temporal co-ordination between stem-cell division dynamics and rates of differentiation of stem-cell progeny is critical to maintain a stable SAM.

Plasma membrane-localized Yellow fluorescent protein (YFP), 35S:: YFP29-1 expressed over ubiquitous promoter is used to visualize cell outlines or boundaries of all cells to follow cell division and expansion patterns (Figure 1.2) [15][16]. By the CLSM based live imaging method, it is possible to get a sequence of time-lapse SAM images (as shown in Figure 2.2). By changing the

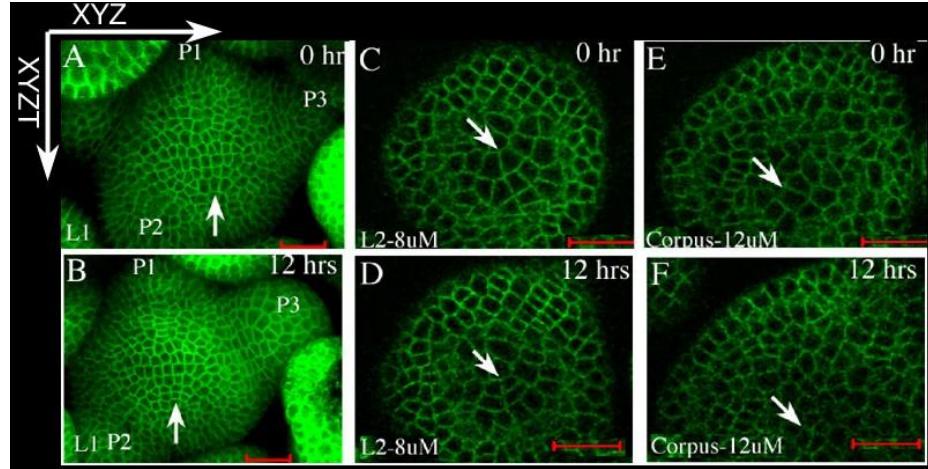


Figure 1.2: (A), (C) and (E) are the first time points and the (B), (D) and (F) represent the same SAMs after the elapsed time indicated on top right hand corner of each panel. (A) and (B) show the Z-stack revealing all the cells in the L1 layer. Differentiating organs at different stages of development are marked as P1, P2 and P3. Arrows indicate the same cells prior to and after division. (C) and (D) represent cells in the L2 layer ($8 \mu\text{m}$ deep) and the arrow points to cell division events. (E) and (F) are the cells in the L3 layer/corpus ($12\mu\text{m}$ deep) and arrow points to cell division events. Scale at the bottom of each image is $20 \mu\text{m}$. XYZ represents 3D image as stacks. XYZT represents 3D image across time.

depth of the focal plane, CLSM can provide in-focus images from various depths of the specimen [17]. The set of images, thus obtained at each time point, constitute a 3D stack. Each stack is imaged at a certain time interval (e.g. 3 hours between successive observations) and it is comprised of a series of optical cross sections of SAMs. This data is then analyzed by the biologist to gain insight into cell and SAM growth dynamics [16]. Figure 1.2 shows a typical set of the spatio-temporal stack of fluorescent images from confocal microscope. Annotations are added to these images for the purpose of illustration.

1.1.2 Nematode Image Stacks by Differential Interference Contrast Microscopy

Efficient representation and effortless distribution of taxonomic information is a very challenging problem, especially for organisms which are transparent with 3D internal and external diagnostic features [18, 19, 20]. Highly diverse groups of invertebrates as nematodes, tardigrades and gastrotrichs are such typical examples. For instance, nematodes are the most diverse phylum of pseudocoelomates, and one of the most diverse of all animals [20]. It is very difficult to distinguish the nematode species, over 28,000 of which have been described [18]. In fact, the total number of nematode species has been estimated to be about 1,000,000. Identification and classification of such kind of organisms requires using high-resolution light microscopy to visualize structures from multiple focal planes as initial observations, which is usually accomplished by differential interference contrast microscopy [20].

Differential interference contrast microscopy is an optical microscopy illumination technique used to enhance the contrast when imaging the unstained, transparent samples. It is also called Nomarski Interference Contrast or Nomarski microscopy [14]. Based on the principle of interferometry, the DIC microscopy is able to gain information about the optical path length of the sample and see otherwise invisible features [14]. Image produced by this microscopy is similar to that produced by phase contrast microscopy, but without the bright diffraction halo [14].

Image stacks taken by Differential Interference Contrast, named as Digital Multi-focal Images (DMI), is a new way for documentation and communication of specimen data [18]. In this approach, morphological information for a transparent specimen can be captured in the form of a stack of high-quality images, representing individual focal planes through the specimen's body, as shown in Figure 1.3. With the data from DMI, the viewer can reconstruct a three-dimensional image

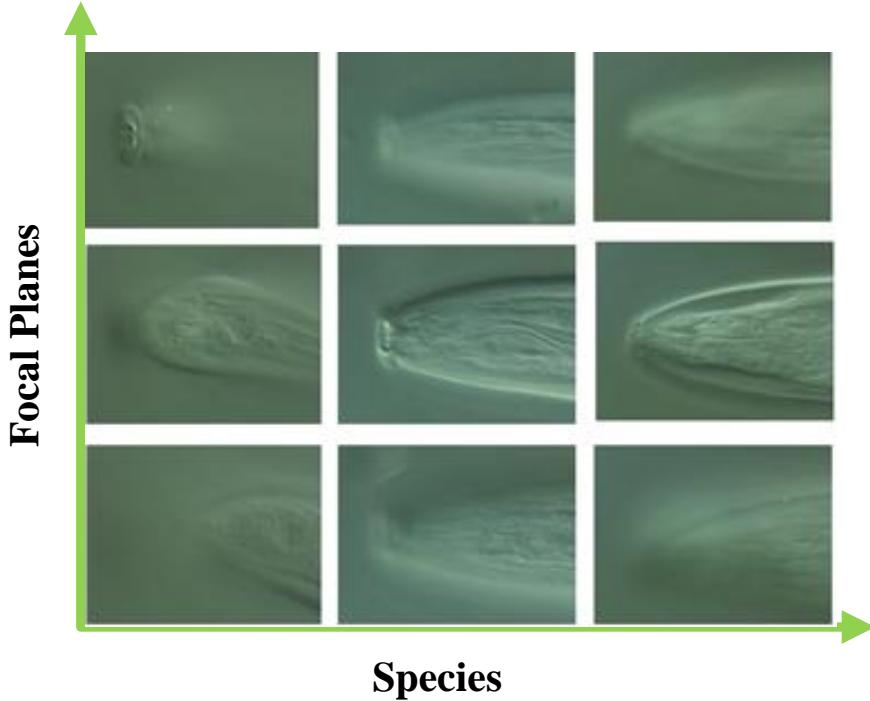


Figure 1.3: Examples of DMI stacks taken from 3 nematode species (horizontal direction). Each stack contains multiple focal planes (frames) taken from the top to the bottom of the specimen, with only 3 key frames of each shown here (vertical direction).

of the specimen. Thus, the use of DMI overcomes the limitations imposed by traditional methods, which omit three-dimensional topology of structures, especially all topological information along the focal axis [18, 19].

1.2 Image Analysis Challenges in Volumetric Bioimages

The deluge of complicated volumetric bioimages poses significant challenges for the image computing and computer vision community [2]. Due to the great complexity and information content in the volumetric bioimages, such as the very high density plant SAM cells in a 4D movie

or very high resolution 3D nematode image stacks, it is very difficult to directly apply existing medical image analysis methods to these bioimage informatics problems [2]. In addition, even a single biological image stack usually has a large size (from several hundreds of megabytes to several gigabytes) and multiple color channels. The objects of interest in such an image, could have dramatic variations of morphology and intensity from image to image [9, 11]. Feature extraction from thousands of such bioimages would cost months or years of manual work.

In the plant cell field, most of the recent work [21] [22][23][24] are based on manual analysis of the CLSM volumetric images, which is extremely laborious and practically impossible, as there are 500-600 cells that need to be identified and tracked in image stacks over multiple time points. The typical tracking methods are not applicable in this case, as most of the features such as shape and color are not discriminative among the cells [25]. More importantly, manual analysis is not likely to provide quantitative information on cell behaviors expecting a few simple statistics like cell division information. Therefore, we need to develop an automated image processing and analysis pipeline, wherein the input would be time-lapse confocal image stacks and the output would be the cell lineages, cell division statistics and other quantitative information. These can then be provided into statistical models to further understand the spatio-temporal dynamics of cell growth and division.

For a classification problem, effective feature extraction would be the most important step. However, the classification of 3D volumetric bioimage stacks is not a typical problem at all, because most of the work on image feature extraction and classification are done through the 2D image processing methods [26, 27, 28]. As mentioned before, the objects of interest in such bioimage stacks, could have dramatic variations of morphology and intensity from image to image, such as

the Nematode DMI image stacks. Usually these kind of images are very high-quality and large-size, and there are hundreds of such images in one single stack. So although the use of DMI includes three-dimensional topology of structures, especially all topological information along the focal axis, how to extract a good feature and then effectively classify those high-throughput volumetric image stacks still is not clear.

1.3 Related Work

1.3.1 Plant Cell Tracking

Pattern formation in developmental fields involves precise spatial arrangement of different cell types in a dynamic landscape wherein cells exhibit a variety of behaviors such as cell division, cell expansion and cell migration [12]. A local spatio-temporal coordination of cell growth and cell division is important for proper development of organs in both plant and animal systems [25]. In the plant field, regulated patterns of cell division is critical to ensure SAM function because other mechanisms of tissue homeostasis such as cell migration and cell death are not detected during SAM development. Earlier genetic studies have revealed that the cellular identities, their functions and growth patterns are regulated by diverse positional and environmental signals. The challenge for the future is to understand how diverse signals impinge on varieties of cellular behaviors such as cell division rates and cell expansion patterns to ensure tissue homeostasis in SAMs.

Our current knowledge of cell division patterns is derived from earlier studies carried out on several different plant species by employing varieties of analytical methods [12][29] [30]. Cytological studies and analysis of distribution of mitotic cells have revealed that the CZ cells

divide more infrequently than cells in the PZ. The serial replica method has been employed to generate developmental time series of SAM surface and these studies have produced quantitative description of cell expansion behavior and associated changes in SAM shape [21] [22][23][24]. A careful and comprehensive morphometries analysis involving counting of mitotic figures from a large number of SAMs in *Arabidopsis thaliana* has yielded a composite view of mitotic activity [31].

Though these studies as introduced before have yielded a dynamic description of mitotic activity, the causal link between cell growth and cell division and how they, in turn, affect organ formation is not well understood [16]. This is mainly due to a lack of quantitative understanding of cell growth patterns. Therefore, the development of computational platforms which are capable of identification of cellular coordinates and automated tracking of cells and cell division events from fluorescent 4D volumetric bioimage stacks acquired by using laser scanning confocal microscopy is important. Such computational platforms would facilitate the quantification of cellular parameters such as rates and patterns of cell expansion, orientation and rates of cell division, and extraction of such information may lead to the development of growth models that can explain the causal relationships between cell deformation dynamics, cell growth and cell division patterns. This is a computational challenge that has universal application to all developmental fields, both animals and plants. The development of live-imaging methods and suitable fluorescent markers have facilitated continuous visualization of given SAMs for several days [16][32].

There has been some work in automated processing and analysis of cellular images [25, 33, 34, 35, 36, 37, 38, 39]. Methods such as [40, 41] show that individual cells can be efficiently segmented in a multi-cellular field and methods such as [42] provide automated tools to track indi-

vidual cells over time. But most of these methods require high SNR data to generate good results. There is a recent work by Gonzelez et. al. that proposed an imaging and image analysis pipeline [43], based on a multi-angle image acquisition of the SAM. However, their imaging process is different from our case. Because high intensity laser radiation can be toxic to plant cells and causes photodynamic damage to live SAM cells, in order to keep the cells alive and growing, we can not get both high spatial resolution in image acquisition and time resolution. For analyzing the growth dynamics of cell clusters where the time gap between successive cell divisions is in the range of 30 to 36 hours, we need a much higher time resolution (usually 3 hours to 6 hours) in imaging in order to capture the exact growth dynamics. This forces a sacrifice in the spatial or depth resolution in order to have more frequent observations in time and hence the number of image slices in which a cell can be present is often really small. This necessitates the development of a new image analysis method which is capable of providing accurate cell growth and cell division statistics at frequent time intervals, without damaging the cells in the *Arabidopsis* SAM.

1.3.2 Image Stacks Classification

As mentioned in Section 1.1.2 and shown in Figure 1.3, we can see that in the DMI bioimage stacks, the morphological information for a transparent specimen can be captured in the form of a stack of high-quality images, representing individual focal planes through the specimens body. The use of DMI overcomes the limitations imposed by traditional methods, because it contains the three-dimensional structural information of the specimen, especially all topological information along the focal axis. However, there is very little work in the automatic image analysis of DMI images, although they are becoming very popular in biological imaging. Especially, the classification

of 3D volumetric bioimage stacks is a very challenging problem. It is mostly because the objects of interest could have dramatic variations of morphology and intensity information from image to image in a stack, and the image size usually is pretty large, and the number of images in each stack could be up to hundreds as well.

Most of the work on image feature extraction and classification are done through the 2D image processing methods [26, 27, 28]. We could use those methods to extract the texture, shape or other features from the key images (most typical images) in a stack to classify them, but then the 3D structural information of a stack is not explored at all, which should be integrated in the feature to make the classifier more accurate. Or we could extract the texture or shape feature from individual images of a stack, but how to integrate all those extracted features together as a unique feature to distinguish the stacks from each other is still an open question.

1.4 Contribution of This Work

1.4.1 Cell Tracking in Volumetric Plant Cell Image Stacks

We build an image analysis pipeline that has multiple components including image registration, cell segmentation, a temporal tracking module to obtain cells correspondences between images along time and a spatial tracking algorithm to obtain correspondences between spatial image slices.

In our study, the cells are segmented by level-set method [40] or watershed algorithm [44] and then tracked by local-graph matching method [10], which exploits the cells' relative position to match the cells both spatially and temporally. This heuristic graph matching method is described

in Chapter 2, wherein the matching problem is solved in a progressive manner (i.e., cell by cell) by obtaining correspondences from local graphs generated at different time instants. The local geometrical and topological features of cells are exploited to generate graphs of the local neighborhood of each cell. This process is followed by matching of the relative positional information of cells, such as the length and orientation of the edges with respect to their nearest neighbors to find the most similar local graph pair between two time instants. In the proposed local graph matching framework, the tracker first finds a seed (initial) cell pair and the seed pair is used as a starting point to calculate similarities between local regions in the graph by progressively moving outwards (from this seed pair) to obtain correspondences of neighboring cells. This process is continued recursively to find correspondences of all cells. The cell tracker not only computes the correspondences of cells across spatio-temporal scale, but it also detects cell division events, and identifies daughter cells upon divisions, thus allowing automated estimation of cell lineages from images captured over a period of 72 hours

Furthermore, we integrate the segmentation and tracking modules to reach an optimized segmentation and tracking result [45]. We segment and track cells through the use of a closed loop, adaptive segmentation and tracking approach. The tracking output acts as an indicator of the quality of segmentation and, in turn, the segmentation can be improved to obtain better tracking results. We construct an optimization function that minimizes the segmentation error, which is, in turn, estimated from the tracking results. This adaptive approach significantly improves both tracking and segmentation when compared to an open loop framework where segmentation and tracking modules operate separately.

Finally, since the cells not only have temporal correspondence but also have spatial cor-

respondence, the spatial and temporal correspondences are fused together to obtain a single unified track. This allowed us to track those cells that may have been poorly imaged in one slice, but are of a higher quality in a neighboring slice. In order to maintain the consistency of the cell lineages for long time periods, we have developed a method to integrate the tracks from multiple slices at a given time instant with that of tracks obtained across time periods.

1.4.2 Nematode Image Stacks Classification

Given stacks of multifocal images as shown in Figure 1.3, how do we efficiently integrate the information from all layers and build an effective classifier to distinguish them? To reach this objective, we propose the 3D X-Ray Transform to explore the information through different image layers, so that we can make better feature extraction and classification for DMI images. For example, the first 2 DMI stacks in Figure 1.3 belong to the same nematode class, but it is possible to make the mistake that they are from different classes by shape or texture based classification, relying only on the original image frames. However, if we project those 2 DMI stacks along different angles with the 3D X-Ray Transform, those projections in the same direction look similar, as shown in the top 2 rows of Figure 4.4. Moreover, although the original key frames in the 2nd and 3rd stacks look similar, they belong to different species, and in Figure 4.4 we can see that their projections along the 2nd and 3rd directions are different. Based on this we can make the right classification for those 3 DMI stacks. The 3D X-Ray Transform allows us to analyze the entire multi-focal stack of images from different viewing angles - something direct analysis of the images would not allow.

Using the 3D X-Ray projections at different viewing angles of the 3D nematode image stacks, we can effectively exploit the entire information in the stack. Because these DMI stacks

represent the effect of different factors - shape, texture, viewpoint, different instances within the same class and different classes of specimens, we embed the 3D X-Ray Transform within a multilinear framework and propose a Multilinear X-Ray Transform feature representation. By combining the tensor texture and shape information we reach better recognition rates than just relying on the original or key frames of DMI stacks.

1.5 Outline of Thesis

In Chapter 2, we show how to extract features in live cell imaging of plant tissues for the tracking of SAMs cells. We use the watershed or level-set method for cell segmentation, and propose the local graph matching based method to track the cells both spatially and temporally, and identify cell divisions at the same time. In Chapter 3, we build a joint segmentation and tracking system, and evaluate our proposed image analysis pipeline on multiple datasets. Chapter 4 presents a 3D X-Ray Transform based multilinear feature extraction and classification method for nematode DMI stacks. Chapter 5 concludes and outlines the future work.

Chapter 2

Feature Extraction in Live Cell Imaging of Plant Tissues

2.1 Introduction

As introduced in Section 1.3.1, we know that an image analysis pipeline capable of providing accurate cell growth and cell division statistics is critical to understand the cell growth dynamics. In such pipeline, the most important part is how to track the cells. One of the well-known approaches in tracking cells is based on level-sets. A level-set is a collection of points over which a function takes on a constant value. The multiple-level-set approach is an active-contour based algorithm, which simultaneously segments cells and also tracks them [25, 34, 35, 46, 47, 48]. In this case, every cell is represented by a separate level-set function, wherein every level set function behaves like an active contour which gradually evolves toward the boundary of cells. Therefore it has been adopted for obtaining cell segmentation [40]. However, the level set method is not suitable

for tracking of SAM cells because of several reasons. First, the cells are in close contact with each other, and can not move freely. Second, the SAM cells share similar features with respect to their shapes and sizes, so features such as shape, intensity and so on are not reliable. Third, large parts of images may be noisy at a particular time instant due to a low signal to noise ratio. So we do not use level-set base tracking algorithm to track the cells in the 4D image dataset, thought the level-set based segmentation algorithm [40] can be used to segment individual cells for further analysis such as tracking and measurements.

In fact, given sets of segmented cells from the time lapse imagery, tracking of cells over time is essentially a point matching problem. One of the most popular solutions to point matching problem has been the Softassign Procrustes algorithm, which has been applied to compute cell lineages, and it has been improved further to detect cell divisions [49, 50]. The Softassign method uses the information on point location to simultaneously solve both the problem of global correspondence as well as the problem of affine transformation between two time instants iteratively. Though, this method can be applied in aligning global features, it can produce errors in finding the local correspondences of individual cells. Therefore, calculation of local neighborhood structures of cells is crucial in tracking cells of similar shapes that lie in close proximity as in the case of SAMs. Based upon this idea, we propose our local-graph feature based tracking method.

In this chapter, we show how to extract effective features from the volumetric cell images, and based on which we present a local graph matching method for automated-tracking of cells and cell divisions in SAMs of *Arabidopsis thaliana*. In the local graph matching framework, the matching problem is solved in a progressive manner (i.e., cell by cell), by obtaining correspondences from local graphs generated at different time instants [51, 50]. We have exploited the local geometrical

and topological features of cells to generate graphs of the local neighborhood of each cell. This is followed by matching of the relative positional information of cells, such as the length and orientation of the edges with respect to their nearest neighbors to find the most similar local graph pair between two time instants. This process provides a seed (initial) cell pair and the seed pair is used as a starting point to calculate similarities between local regions in the graph by progressively moving outwards (from this seed pair) to obtain correspondences of neighboring cells [52]. This process is continued recursively to find correspondences of all cells.

2.2 Live Cell Imaging of Plant Tissues

As introduced in Section 1.1.1, the SAM stem-cell niche of model plant *Arabidopsis thaliana* consists of about 500 cells, located at the tip of the shoot, and it gives rise to all of the above ground organs (see Figure 2.1). These cells are organized into three distinct cell layers (L1, L2, and L3 in Figure 2.1) that are clonally distinct from one another. The cells in the outermost L1 layer and the sub-epidermal L2 layer divide in planes perpendicular to the SAM surface (anticlinal), while the cells in underlying L3 layer divide in perpendicular planes as well as parallel (periclinal) planes. Thus, the SAM stem-cell niche represents a dynamic 3D network in which cell growth dynamics, gene expression and organ growth are dynamically regulated so that the correct cellular identity, shape and size of the SAM is preserved. This necessitates the use of an imaging technique which can capture cell images from multiple layers of the SAM (C1, C2, C3 in Figure 2.1).

The development of stable transgenic plants ubiquitously expressing a plasma membrane marker (35S::29-1 YFP) has been described earlier [16]. Plants are grown on soil or on plates, and maintained in continuous light at a temperature of 20-22°C. For the live imaging of the plant tissues,

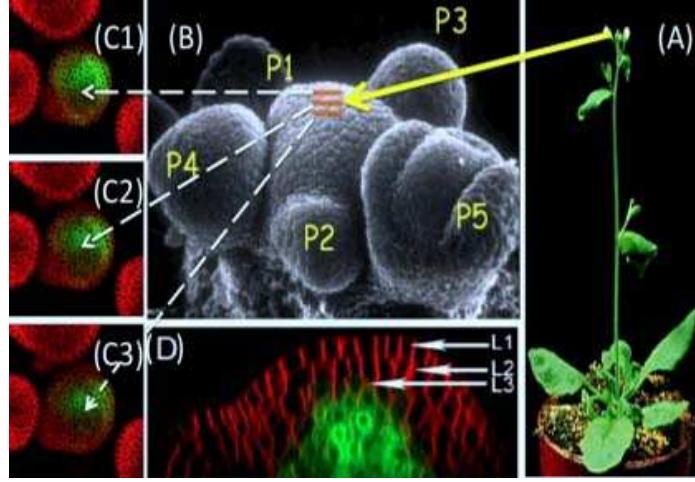


Figure 2.1: Multiple layer structure of the SAM.

before they were transferred into the clear plastic boxes containing MS-agar, they were grown on MS-agar plates for 10 days [53]. Upon bolting, the older floral buds were removed to expose the SAM surface in the time-lapse imaging and the rosette was stabilized by applying 1.5% molten agarose onto the stem the plants. The CLSM based live imaging set-up is used to acquire a sequence of timelapse volumetric images. Plants were imaged on Zeiss510 up-right confocal microscope by using 63X water dipping achroplan lens with an argon laser at 515nm [16]. By changing the depth of the focal plane, CLSM can provide in-focus images from various depths of the specimen, as shown in Figure 2.2.

During the Live Imaging process, the set of images obtained at each time point, constitute a 3D stack, also known as the Z-stack. Each Z-stack is imaged at a certain time interval (e.g. 3 hours between successive observations) and it is comprised of a series of optical cross sections of SAMs that are separated by approximate $1.5 \mu\text{m}$. A standard shoot apical meristematic cell has a diameter

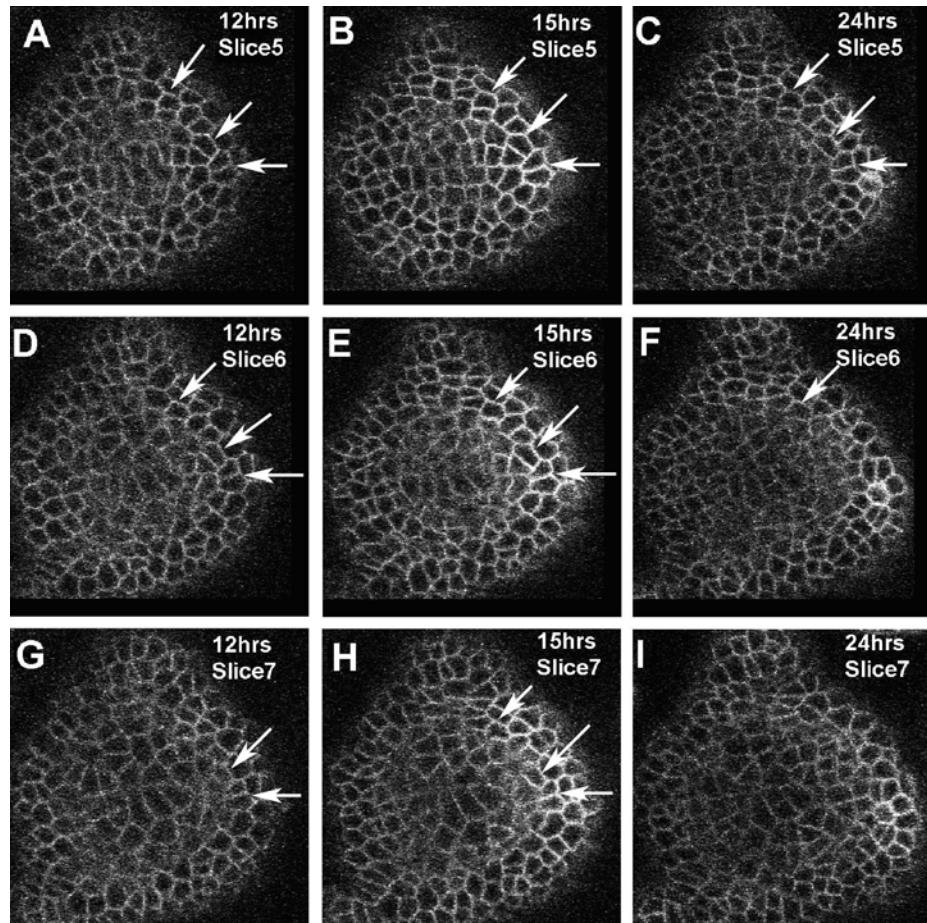


Figure 2.2: Time-lapse imagery of SAMs labeled with plasma membrane localized YFP. Examples of three image slices (vertical columns) at three time instants (horizontal columns; arrows A, B and C show same set of cells at time points noted on each panel). Cross-sectional images of SAMs depicted in the vertical columns are separated by $1.5 \mu\text{m}$, while the size of a single cell is about $5 \mu\text{m}$ in diameter. Therefore each cell is represented in two or three consecutive slices (arrows in A, D and G correspond to the optical sections of same cells located in different depths). (A, B and C) belong to slice 5, (D, E and F) belong to slice 6 and (G, H and I) belong to slice 7. Note the representation of several cells in three consecutive sections (arrows).

of about $5\mu\text{m}$ and hence in most cases, a single cell is not visible in more than 3-4 slices. To account for any minor shift in the alignment of the images in the 3D stack, each stack is registered by a method of maximization of mutual information [54] or using an existing image registration software, MIRIT [55].

2.3 Proposed Pipeline

Given bioimage stacks taken by the live-imaging method in Section 2.2, we build an image analysis pipeline that has multiple components including cell segmentation, a temporal tracker, a spatial tracker and a fusion process to integrate cells' temporal correspondences and spatial correspondences, as shown in Figure 2.3. The cells are segmented by level-set method [40] or watershed algorithm [44] and then tracked by local-graph matching method [10], which exploits the cells' relative position to match the cells both spatially and temporally [10]. Furthermore, the segmentation and tracking modules are integrated to reach an optimized segmentation and tracking result [45]. Finally, since the cells not only have temporal correspondence but also have spatial correspondence, the spatial and temporal correspondences are fused together to obtain a single unified track. In order to maintain the consistency of the cell lineages for long time periods, we have developed a method to integrate the tracks from multiple slices at a given time instant with that of tracks obtained across time periods. The segmentation and local graph matching based tracking are described in this chapter, while the adaptive segmentation and tracking scheme and the lineage fusion process are introduced in Chapter 3.

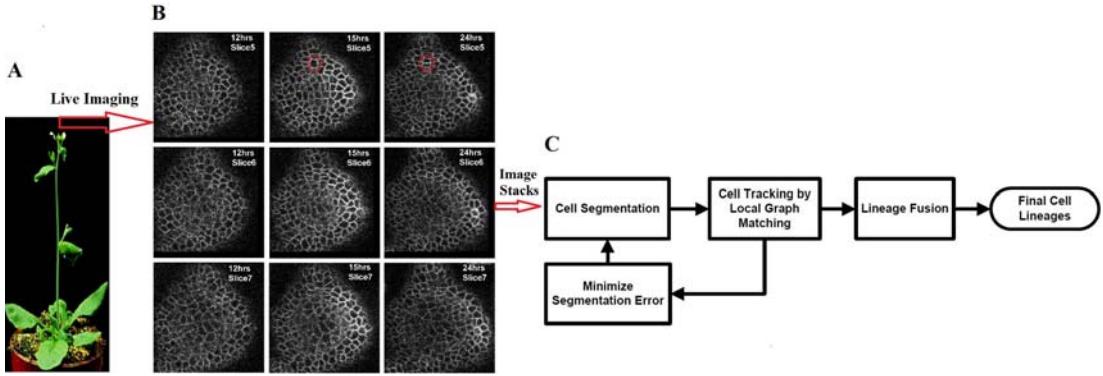


Figure 2.3: Overall Imaging and Image Analysis Framework. (A) SAM located at the tip Arabidopsis shoot. (B) Time-lapse imagery of SAMs labeled with plasma membrane localized YFP. Examples of three image slices (vertical columns) at three time instants (horizontal columns). The red circles denote a cell division. (C) The diagram of the cell segmentation and tracking pipeline.

2.4 Cell Segmentation

The first step for cell tracking and cell division identification is cell segmentation. In our research, we tested two most popular segmentation methods, the level-set based algorithm and watershed algorithm [40, 56, 10, 45]. Though, it turns out both segmentation methods can provide effective segmentation results for our proposed local graph matching based tracker, the watershed segmentation algorithm works better in two aspects: first, it reaches more accurate cell boundaries; second, it works more fast, so it is more suitable for the further building of an adaptive segmentation and tracking system.

2.4.1 Level Set Segmentation

We adapt the active contour models implemented via level set methods [40], for segmentation, because this method can detect objects without sharp edges, and it allows for automatic

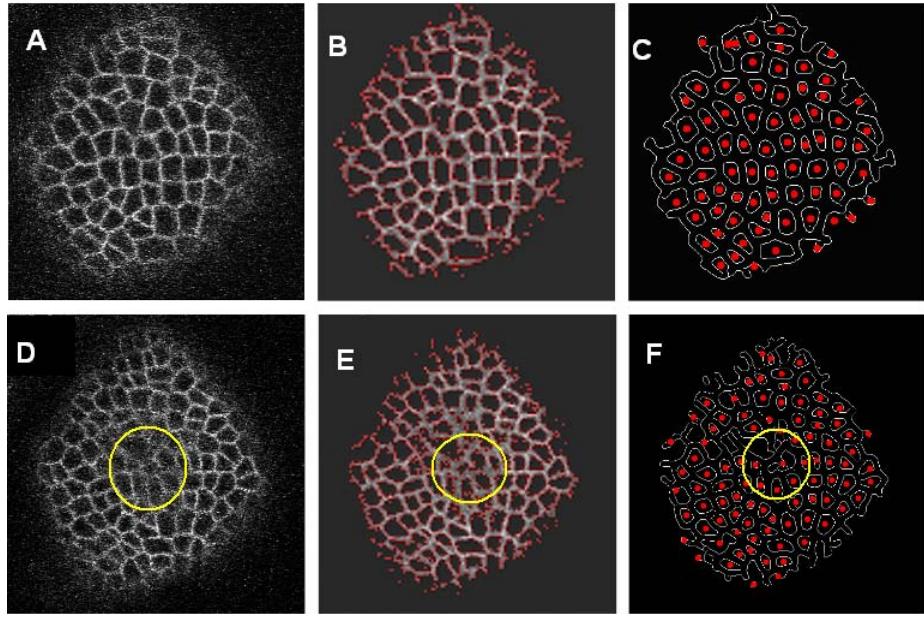


Figure 2.4: Level-set based segmentation results of SAM cells. Segmentation results (B and E) of the original confocal cross sections acquired at different depths (A and D), and cell centroid detection (C and F) from the segmentation. Note that the cells in the central part of the image (yellow circle) are not segmented properly because of noise.

change of topology. The idea behind active contours, or deformable models, for image segmentation is quite simple [57, 58, 35, 59]. The user specifies an initial guess for the target object contour, which is then moved by certain image driven forces to the boundaries of the desired objects. For more details about active contour model and level set based segmentation, please refer to [60, 40].

Some segmentation results using the Chan-Vese level-set model are shown in Figure 2.4. If the signal to noise ratio is low, the cell boundaries are not easy to segment, and the cells detection may fail. An example is the central part denoted by yellow circles in Figure 2.4 D-F, where some cells can not be properly segmented and detected. Also, we can notice that the segmented cells' boundaries are not quite close to the cells' real boundaries, there are quite certain void between the

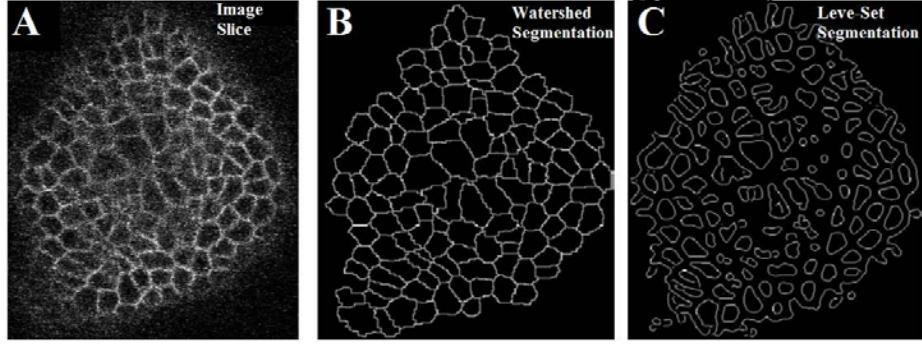


Figure 2.5: (A)Original image slice. (B)Watershed segmentation result. (C)Level-set segmentation result.

segmented cells.

2.4.2 Watershed Segmentation

Watershed [61, 56] treats the input image as a continuous field of basins (low intensity pixel regions) and barriers (high intensity pixel regions), and outputs the barriers which represent cell boundaries. It has been used to segment cells of *Arabidopsis thaliana* root meristem [62]. Compared to Level-Set based segmentation, watershed method reaches more accurate cell boundaries, which can be clearly noticed by comparing Figure 2.5B and 2.5C. However, the main drawback of watershed algorithm is that it results in both over-segmentation and under-segmentation of cells, especially those from deeper layers of SAMs which are noisy. So prior to applying the watershed algorithm, the raw confocal microscopy images undergo H-minima transformation in which all the pixels below a certain threshold percentage h are discarded [63]. The H-minima operator was used to suppress shallow minima, i.e., those whose depth is lower than or equal to the given h -value. The watershed segmentation after the H-minima operator with a proper threshold can produce much

better segmentation results than level-set segmentation.

Since the H-minima threshold value h plays a very crucial role in the watershed algorithm, especially when the input images are noisy, it is extremely important to choose an appropriate threshold value such that only the correct cell boundaries are detected. Generally, a higher value of the threshold parameter h performs under-segmentation of the images (Figure 2.6 C), and conversely a lower value over-segments the images (Figure 2.6 B). This is also evident in noisy image slices, that the number of over-segmented cells increases as we choose lower values for the H-minima threshold h and on the other hand, a larger value of h produces more under-segmented cells. Since the cell size is fairly uniform for most cells of the SAM, the watershed should ideally produce a segmented image that contains similar sized cells. Thus a good starting threshold would be the value of h such that variance of cell areas in the segmented image is minimum. In the next chapter, we will explain how to obtain an optimal value of h for watershed segmentation, using the feedback from the tracking results.

2.5 Cell Tracking Based on Local Feature Graph

In a tracking problem, selecting an effective feature is the most critical step. In our case, the cells are in close contact with each other and the SAM cells share similar features with respect to their shapes and sizes. So the common features such as shape, color are not reliable to track the plant cells. In the research, we find that an unique feature can be used to track the cells in our special case is the relative position between the neighboring cells. In fact, this relative position information is also the knowledge used by biologists when they do the manual tracking: identify a seed cell pair, then start from it to match the neighboring cells and so on. So we match the cells by the local graph

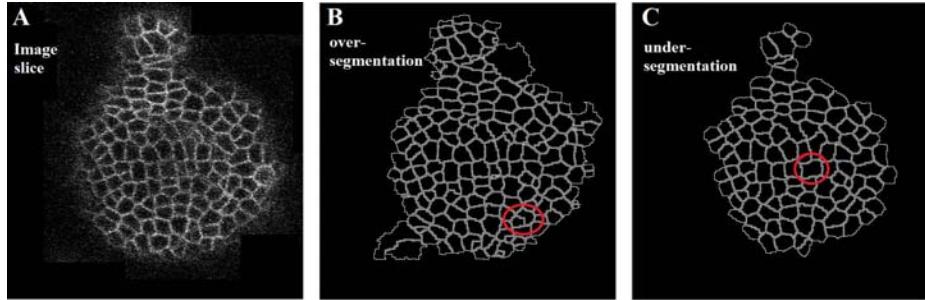


Figure 2.6: (A)Original image slice. (B) Over-segmentation case denoted by red circle. (C) Under-segmentation case by red circle.

matching method, which exploits the corresponding local structures of the cells to match them [10].

2.5.1 From Cells to Graphs

Upon obtaining a collection of segmented cells from time-lapse imagery, we created a graphical abstraction of segmented cells. This process involved representing every cell by a vertex and connecting neighboring vertices by an edge (Figure 2.7 A and 2.7 B). Those close cells within certain distance from a certain central cell were considered as its neighboring cells and they were denoted by the neighborhood cell set (Figure 2.7 A and 2.7 B). The structure of these graphs automatically includes the relative positional information of cells, such as the distance between two neighboring cells (the edge length) and the edge orientation. The topology and the geometry of the local graphs are not expected to change dramatically between two consecutive time instants separated by approximately 3 hours unless the cells divide or the images are noisy. With these conditions satisfied, the corresponding cells across time were identified by matching the respective local graphs with a similarity (distance) measure, as described in the following section. We illustrate the local-graph matching idea in details in the temporal tracker, the method for the spatial tracker is similar

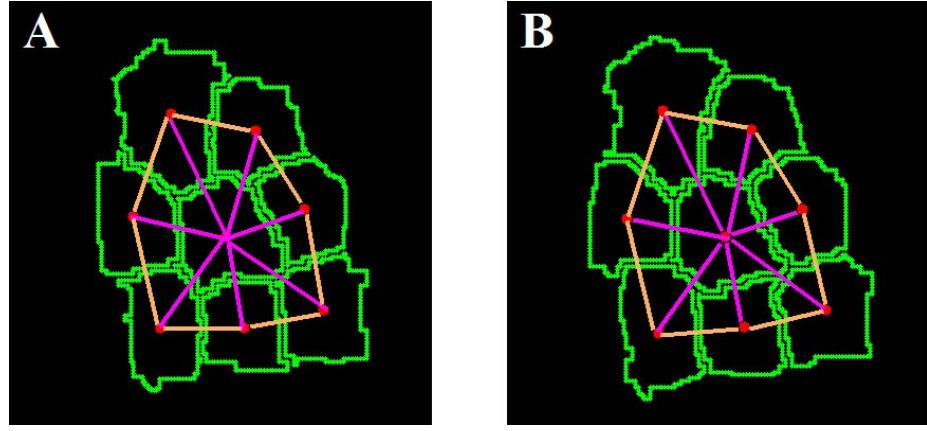


Figure 2.7: Local graph matching.

and briefly described.

2.5.2 Temporal Tracker

Local Graph matching

In the local graph-matching framework, a vertex in the graph represents a cell and neighboring vertices are connected by an edge. The graph structure automatically includes the relative position information of the cells, such as the relative distance between two neighboring cells (the edge length) and the edge orientation. Given two local graphs G_1 and G_2 containing the same number of neighboring cells, corresponding to cell c_i at time t and c_j at time $t + 1$ respectively, the distance measure, D_T , for these two graphs consists of three parts: the normalized difference of the edge lengths at consecutive time instants t and $t + 1$, the difference of the orientation angles

between the edges, and normalized area difference information of cells, as follows

$$D_T(c_i, c_j) = \frac{\lambda_1}{M} \cdot \sum_{c_m \in N(c_i), c_n \in N(c_j)} \frac{|l_{c_m, c_i}(t) - l_{c_n, c_j}(t+1)|}{l_{c_m, c_i}(t)} + \lambda_2 \cdot \frac{|A_{c_i}(t) - A_{c_j}(t+1)|}{A_{c_i}(t)}, \quad (2.1)$$

$$+ \frac{\lambda_3}{M} \cdot \sum_{c_m \in N(c_i), c_n \in N(c_j)} \frac{|\theta_{c_m, c_i}(t) - \theta_{c_n, c_j}(t+1)|}{\theta_{c_m, c_i}(t)} (m = n = 1, \dots, M)$$

where M is the number of neighboring cells in the two local graphs, c_m is the m^{th} neighboring cell of c_i , and c_n is the n^{th} neighboring cell of c_j , $l_{c_m, c_i}(t)$ and $l_{c_n, c_j}(t+1)$ are the edge lengths, $\theta_{c_m, c_i}(t)$ and $\theta_{c_n, c_j}(t+1)$ are the orientation angles in radians of the edges measured relative to a horizontal axis, $A_{c_i}(t)$ and $A_{c_j}(t+1)$ are the size of the central cells' (c_i and c_j) area. The neighboring cells in the two graphs are ordered by orientation and distances are computed between the central cell and a neighboring cell having the same ordering number, i.e., $m = n = \{1, \dots, M\}$. If two local graphs match (i.e., the distance measure is small), the central cells (i.e., c_i and c_j) in those two local graphs are corresponding cells (a cell pair).

Finding the Seed/Initial Cell(s) and Recursive Matching of Neighboring Cells

The above distance computation in Equation 2.1 finds the similarity between two local graphs through the distance function based on the assumption that the topology of local graphs does not change between two consecutive time instants and the changes in geometry of the local graphs are minimal. However, this may not always be true with SAM cells which show continuous growth and also every cell in the image field may not be imaged properly. Therefore, we devised a two part tracking strategy which involves finding the most similar cell pair as the seed pair, and then employing a slightly different distance measure to grow the matching process radially among the neighboring cells of the seed pair. The main difference is that while the seed pair is computed

from the neighborhood structure using local graph matching, tracks of other cells are found by computing the similarities along different paths of the graph [52]. Mathematically, this means that in the seed computation (Equation 2.1) the distances are summed over the neighborhood, while in the Equation 2.2, we only consider the distance between two neighbors without summation over the neighborhood set.

Let us assume that the first image has N cells, and the second image has M cells. According to the distance function in Equation 2.1, we can get the distance value $D_T(c_i, c_j)$ of every cell pair (c_i, c_j) , and pick the most similar cell pair (c_a, c_b) that satisfies

$$D_T(c_a, c_b) = \min_{c_i, c_j} \{D_T(c_i, c_j), i = 1, \dots, N, j = 1, \dots, M\}. \quad (2.2)$$

Alternatively, we could find the top few most similar cell pairs instead. Then the tracking process will be faster, because we can start from more corresponding cell pairs (more seeds). This automatic seed picking procedure is used in this paper. However, the users have the freedom and choice to find the seed pair manually or other particular methods, especially in some complicated situations, where the SNR (Signal to Noise Ratio) is too low and the seed pair is not easy to choose automatically.

Given a matching cell pair, we now describe the process to compute matches for the neighboring cells. This distance function is composed of two parts: the normalized difference of the edge lengths and the difference of the orientation angles between the edges. Let c and c_j be two correctly matched cells respectively at time instants t and $t + 1$, and c_m and c_n be two neighboring

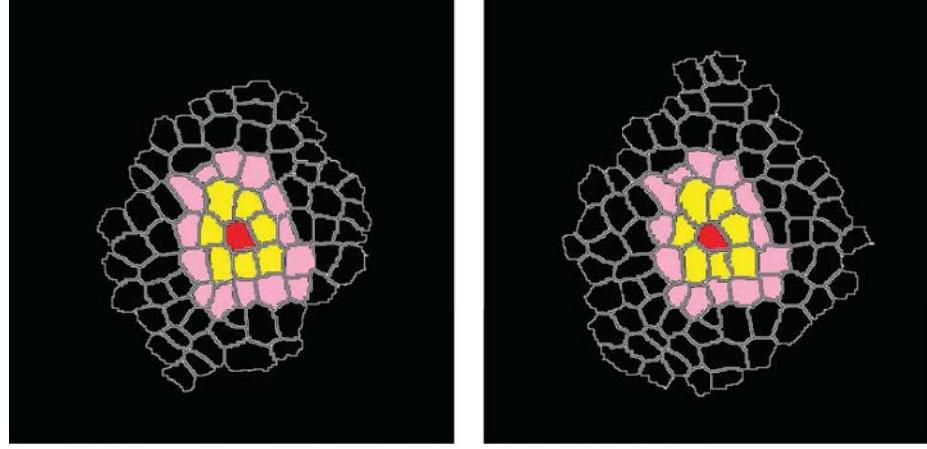


Figure 2.8: Illustration of the correspondence growing process starting from the seed pair to the neighboring cells. Red colored cells are the seed pair. Yellow colored cells are the matched cells in the first growing iteration. Purple colored cells are the matched cells in the second growing iteration.

cells around them respectively. Then the distance measure function is

$$D_N(c_m, c_n; c_i, c_j) = \lambda_4 \cdot \frac{|l_{c_m, c_i}(t) - l_{c_n, c_j}(t+1)|}{l_{c_m, c_i}(t)} + \lambda_5 \cdot \frac{|A_{c_m}(t) - A_{c_n}(t+1)|}{A_{c_m}(t)} + \lambda_6 \cdot \frac{|\theta_{c_m, c_i}(t) - \theta_{c_n, c_j}(t+1)|}{\theta_{c_m, c_i}(t)} \quad (2.3)$$

where c_m is the m_{th} neighboring cell of c_i , and c_n is the n_{th} neighboring cell of c_j , $l_{c_m, c_i}(t)$ and $l_{c_n, c_j}(t+1)$ are the edge lengths, $\theta_{c_m, c_i}(t)$ and $\theta_{c_n, c_j}(t+1)$ are the orientation angles in radians of the edges measured relative to a horizontal axis, $A_{c_m}(t)$ and $A_{c_n}(t+1)$ are the cell area sizes.

If the distance is small, c_m and c_n are identified as matching cells, and the process continues. This leads to a recursive procedure for matching cells from a seed to the neighbors. The process is illustrated in Figure 2.8, where the seeds denoted by red color are used to grow the tracking recursively in the neighboring cells as denoted by the yellow color and the purple color.

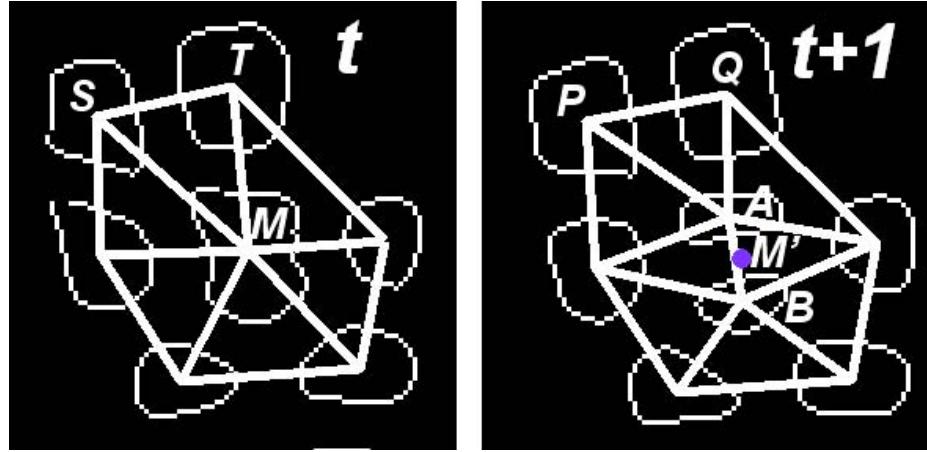


Figure 2.9: A diagram to describe the relative position of the mother cell and the daughter cells.

Detecting Cell Divisions

Based on the location, the cells of the SAMs divide at different rates [32, 16]. It is expected that the topology of a given local graph will change upon cell division. Therefore, the changes in topology of local graphs act as good indicators of cell division events.

Take the local graphs illustrated in Figure 2.9 for example. Here a cell M at time t is divided into two sibling cells A and B at time $t + 1$. We can first find the cell pair (M, A) through the correspondence growing procedure starting from cell pair (T, Q) and/or (S, P) . Then we check the area difference of cells M and A . If the difference is over about half of the area of an adult cell, then we could say that there maybe a cell division event, and cell M may be the mother cell and cell A may be one of the daughter cells. If there is really a cell division, we should be able to find the other sibling cell in the neighboring cells of A . Specifically, we can search among the neighborhood around the original location M' of the mother cell at time $t + 1$, and the cell B with the shortest distance from M' should be the candidate for the other sibling cell. Furthermore, this

distance $|BM'|$ should be within about half of the average distance between two neighboring adult cells.

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

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

2.5.3 Spatial Tracker

In the temporal tracking case, the same cell's local graph structure will not change much across different time instances unless there are some cell division events happening, while in the spatial tracking case, the same cell's local graph structure can change drastically across different slices, because every cell only occupies about 3 slices in the spatial direction. That is to say, most of the cells will decrease in the 4th slice, while new cells are coming in this slice. So we need to use a different distance function for the spatial tracker. In fact, the most reliable feature to identify the same cell across different spatial slices is the location information and the area size information.

This is because even the local graph structure changes, the cells spatial location will not change since there is no shift or movement in the image plane across the Z-stack.

Similar to the temporal tracker, given two local graphs G_1 and G_2 containing the same number of neighboring cells, corresponding to cell c_i at slice S_1 and c_j at slice S_2 respectively, the distance measure D_S for these two graphs consists of three parts: the normalized location difference information between the cells and the normalized area size difference, as follows

$$D_S(c_i, c_j) = \lambda_7 \cdot \frac{\|\underline{P}_{c_i}(S_1) - \underline{P}_{c_j}(S_2)\|}{\Delta} + \lambda_8 \cdot \frac{|A_{c_i}(S_1) - A_{c_j}(S_2)|}{A_{c_i}(S_1)}, \quad (2.4)$$

where $\underline{P}_{c_i}(t)$ and $\underline{P}_{c_j}(t+1)$ are the cell position vectors (here they are the central cells' coordinates in the registered image plane), and Δ is the average distance between two neighboring cells. $|\cdot|$ represents the magnitude of the difference between two scalars, while $\|\cdot\|$ represents the Euclidean distance between two vectors. $A_{c_i}(S_1)$ and $A_{c_j}(S_2)$ are the cell area sizes.

2.6 Experimental Results

2.6.1 Choice of The Parameters for The Trackers

The parameter settings for the distance functions in the previous sections depend upon the characteristics of the actual data. In our 3 hours interval data, the average distance between two neighboring cells is about 50 pixels, so if the relative distance of two cells is within 50 pixels, they are regarded as neighbors in the implementation. On the average, if two cells belong to the same cell in different images, the normalized length difference of the same edge is less than 0.1, the

orientation difference of the same edge is within 0.17 radians (10°), the cells' normalized location difference is within 0.2, and the normalized cell area difference is within 0.2. Because every term in the distance functions is normalized, so we can set all λ_s to 1. The average cell area size is 400 pixel², so the area of the divided daughter cell is about 200 pixel². If two matched cell pair's area difference is beyond 200 pixel², we consider that there may be a cell division. If there really is a cell division, the distance between the two daughter cells should be about half of the average distance between two neighboring adult cells, i.e., 25 pixels.

2.6.2 Preliminary Tracking Results

Figure 2.10 shows the tracking results across time instants with cell division. The same cells are denoted by the same number while for the cell division, the mother cell and daughter cells are denoted by circles. Figure 2.11 is the case of tracking across image slices, with noise in the central part of the cell images. More tracking results are shown in Figure 2.12. In Figure 2.13, the local graph matching method demonstrates its robustness to find the corresponding cell pairs. From the tracking results, we can notice that even when the images of the cells in the central part are almost totally corrupted, the correctly segmented cells are effectively tracked. The robustness of this local graph matching method comes from the ability to grow the correspondences to the neighborhood from a seed pair.

In order to evaluate the effectiveness of the cell tracking algorithm in identifying cell division events, we compared the cell division events counted manually with the events identified through our proposed method (Table 2.1). This analysis reveals that the local graph matching method can detect 100% of the cell divisions that are properly segmented and 96% in the unseg-

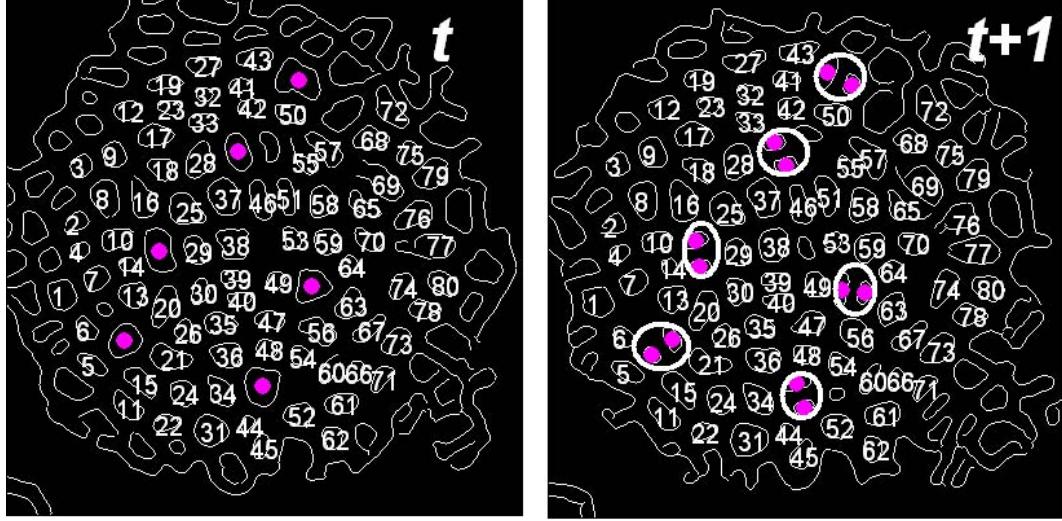


Figure 2.10: Tracking results with cell division, where the mother cells are denoted by purple dots in the first frame, while the sibling cells are in white circles in the second frame.

mented image data suggesting that the small percentage of error is due to the improper segmentation. A small fraction of over representation of cell division events has also been observed and these false positives that appear in the automatic method are due to the improper segmentation of cells (Table 2.1). Though our method is robust to some amount of noise in imaging and errors in segmentation, a consistently poor segmentation across all the slices and consecutive time instants may lead to poor performance.

2.6.3 Robustness

Figures 2.13 B and D show the tracking of cells across two consecutive time instants including seven cell division events (the corresponding cells are denoted by the same color). This example also demonstrates the robustness of the tracking method in finding the corresponding cell pairs despite noisy images, poor segmentation and also cell division events. This robustness is

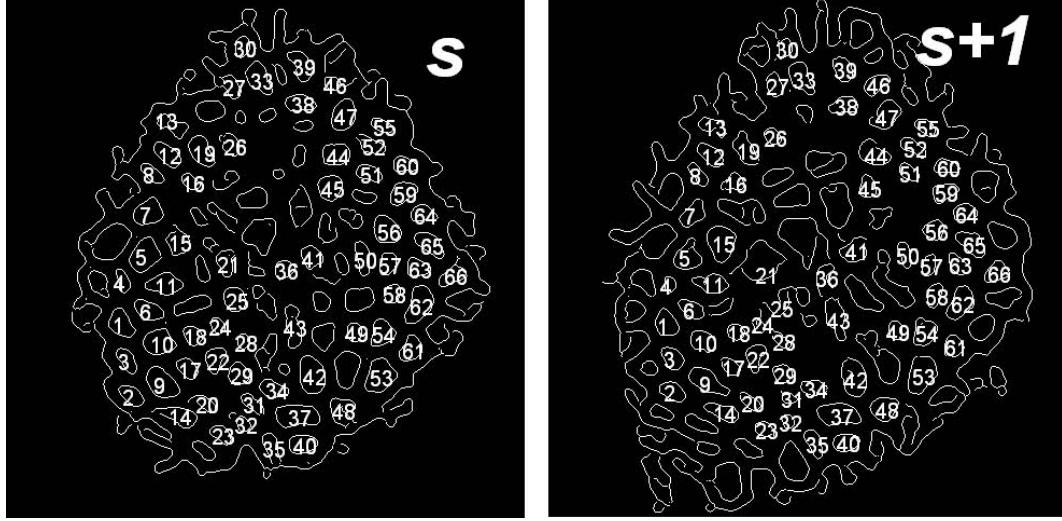


Figure 2.11: Tracking results across slices in noisy cell images, where some cells in the central part are totally missing.

attributable to the local graph matching method which has the capability to grow the correspondences to the neighborhood starting from a seed pair. In Figure 2.13, three such paths for growing the correspondences between two representative cells (“a” and “b”) across two time instants have been shown which reveal that as long as the right seed pairs are chosen, the tracking algorithm will automatically identify a right path from the seed pair to any other cell pairs in the neighborhood. Therefore, even if some parts of the image are noisy, the correspondences among given cell pairs can be found with the availability of at least one feasible path.

2.7 Conclusion

In this chapter, we showed how to extract effective features from the volumetric cell images, and based on which we presented a local graph matching method for automated-tracking of cells and cell divisions in SAMs of *Arabidopsis thaliana*. The local graph matching principle effi-

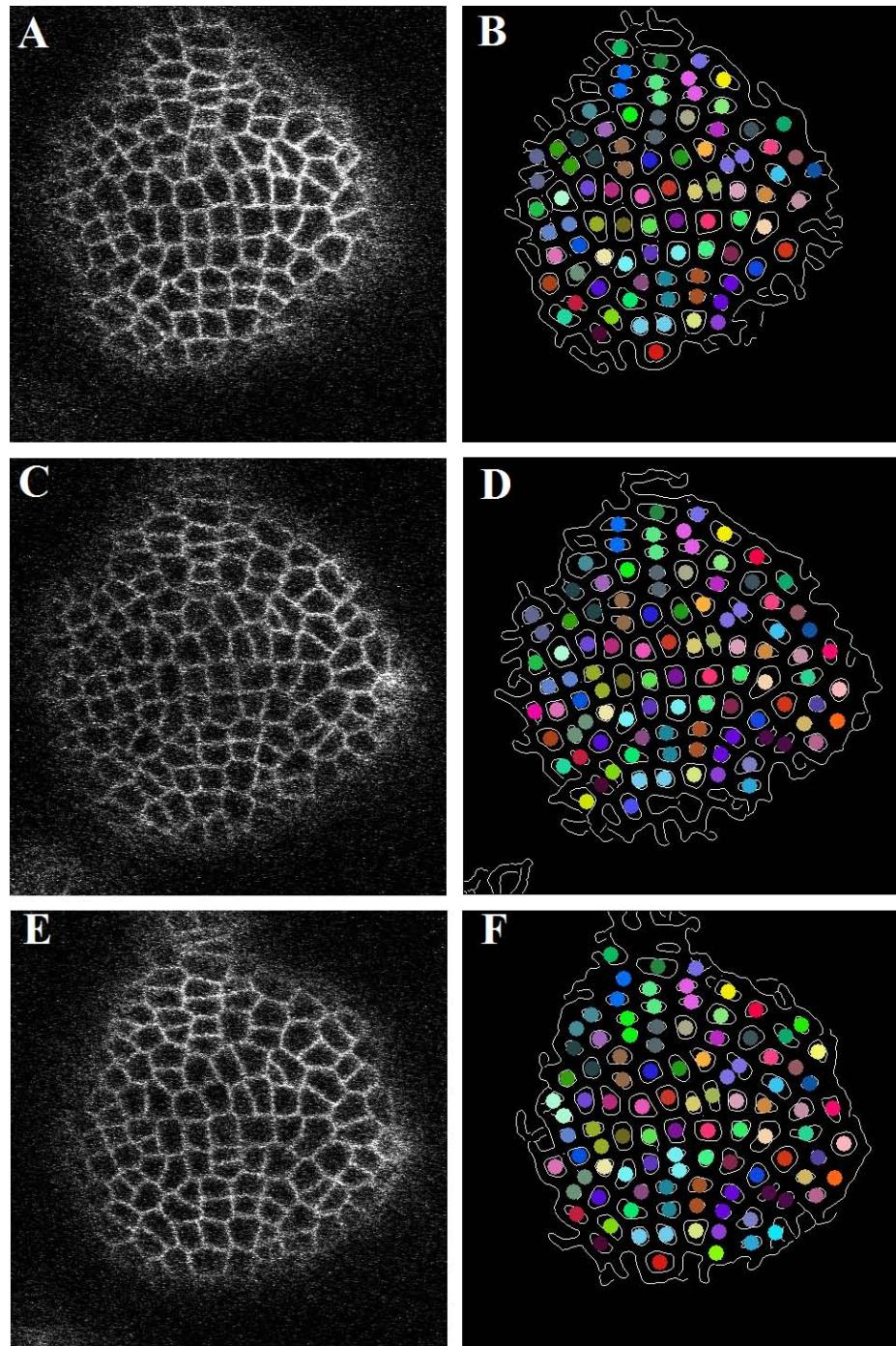


Figure 2.12: (A), (C) and (E) are the original image slices obtained at the 21st hour, 24th hour, and 27th hour. (B), (D) and (F) are the level-set segmentation result and the tracking result of the image slices of (A), (C) and (E). The same color denotes the same cell across different time instances.

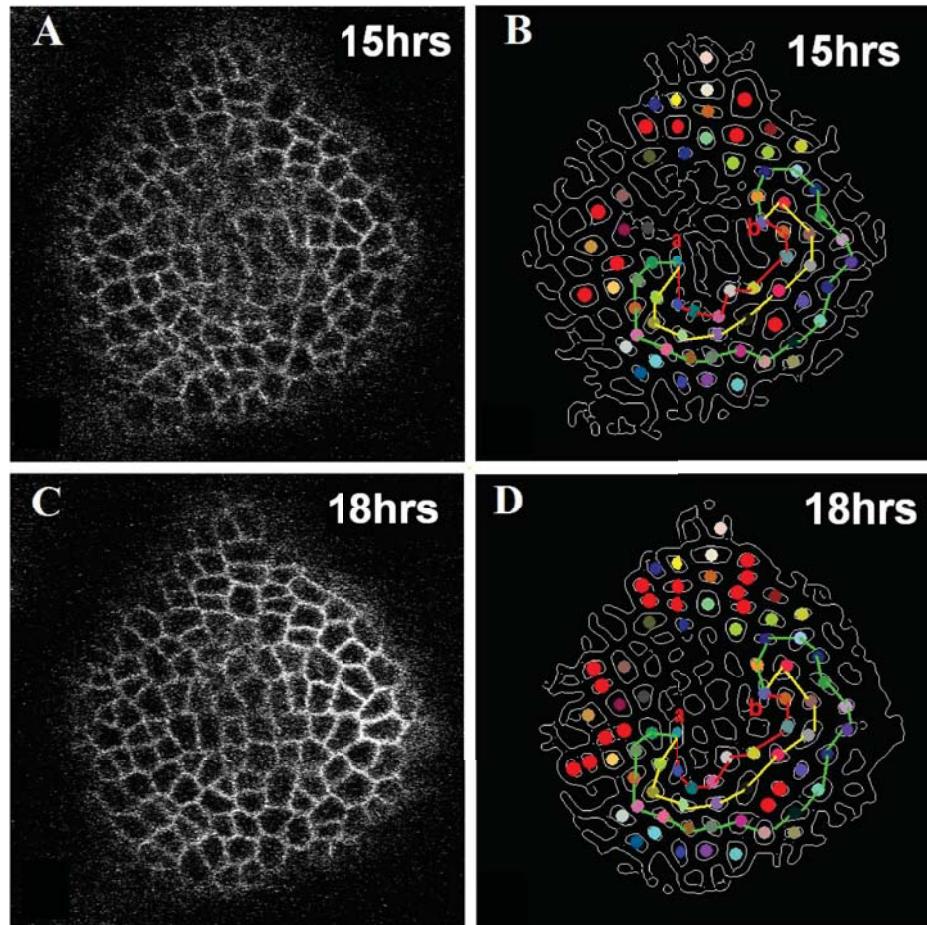


Figure 2.13: Robustness of tracker by considering multiple correspondence growing paths. (A, C) are the original image frames before segmentation, while (B, D) represent tracking output including 8 cell division events (denoted by red dots), where the corresponding, non-dividing cells are denoted by the same color. For a dividing cell, the parent cells and corresponding sibling cells are denoted by red color. We show three such paths between cell ‘a’ and cell ‘b’ for two time points. The three paths in red color, yellow color and green color demonstrate the feature of this algorithm that it can automatically grow cell correspondences from one pair to another pair by multiple paths.

Slice number	Time interval (hours)	Manual detection	Automated method
2	0-30	14 (14)	14
3	0-39	21 (20)	20
4	0-42	34 (32)	32
5	0-57	48 (49)	49
6	0-63	55 (56)	56
7	0-72	55 (53)	53
8	30-72	38 (38)	38
9	30-72	54 (54)	54

Table 2.1: Comparison of the number of cell division events between manual tracking (both in the unsegmented data and the segmented data by Level-set method, the latter being in parenthesis), and the automated method used in this study. The data for different slices is shown.

ciently exploits the geometric structure and topology of the relative positions of cells in obtaining spatio-temporal correspondences. By relying on the local geometry and topology, the method is able to track cells in areas of high curvature such as regions of primordial outgrowth. The cell tracker not only computes the correspondences of cells across spatio-temporal scale, but it also detects cell division events, and identifies daughter cells upon divisions, thus allowing automated estimation of cell lineages from images captured over a period of 72 hours. However, in this chapter, we did not make any attempt to integrate segmentation and tracking so as to minimize the segmentation and tracking errors, which are major concerns in noisy live imagery. We are going to present the adaptive segmentation and tracking framework in the next chapter.

Chapter 3

Integrated Segmentation and Tracking for Cell Lineage Computation

3.1 Introduction

In Chapter 2, we used level-set or watershed segmentation and local graph matching method to find correspondence of cells in the volumetric SAMs images. In the local graph-matching framework, a vertex in the graph represents a cell and neighboring vertices are connected by an edge. The graph structure automatically includes the relative position information of the cells, such as the relative distance between two neighboring cells (the edge length) and the edge orientation. We can find the most similar cell pair (known as the ‘Seed Pair’) by matching the relative positions of cells with respect to their nearest neighbors through the local graph for any two consecutive time points. Starting from this seed pair, we grow the number of matched cells by computing the similarities between local regions in the graph. Cell divisions are detected by detecting changes in the topology

of the graph.

The tracker performance depends heavily on the quality of the segmentation output. However, due to a low Signal-to-Noise Ratio (SNR) in the live cell imaging dataset, the cells are often over-segmented or under-segmented. Previously, we did not make an attempt to integrate segmentation and tracking so as to minimize the segmentation and tracking errors, which are major concerns in noisy live imagery. In this chapter, we have combined the local-graph matching based tracking methodology from [10] with the watershed segmentation in an adaptive framework, where tracking output is integrated with the segmentation. The tracking output for a particular slice acts as an indicator of the quality of segmentation and the segmentation can be improved so as to obtain the best tracking result.

3.2 Adaptive Segmentation and Tracking

3.2.1 Design of Integrated Optimization Function

Due to the rapid deterioration of image quality in deeper layers of the Z-stack, the existing segmentation algorithms tend to under-segment or over-segment image regions (especially in the central part of the image slices). Even a manual segmentation of cells is not always guaranteed to be accurate if each slice in the deeper layers is considered separately due to very low SNR. In fact, in such cases we consider the neighboring slices, which can provide additional contextual information to perform segmentation of the noisy slice in a way that provides the best correspondence for all the segmented cells within the neighborhood. The automated method of integrated segmentation and tracking proposed here involves correcting faulty segmentation of cells by integrating their spatial

and temporal correspondences with the immediate neighbors as a feedback from the tracking to the segmentation module. In the next few paragraphs, we formalize this framework as a spatial and temporal optimization problem and elaborate the proposed iterative solution strategy which yields the best segmentation and tracking results for all the cell slices in the 4D image stack.

The advantage of using watershed segmentation is that it can accurately find the cell boundaries, while its main drawback is over-segmentation and under-segmentation (as shown in Section 2.2.2), which can be reduced by choosing the proper H-minima threshold, which is introduced in Section 2.2.2 [63]. Due to the over-segmentation errors in regions of very low SNR, the watershed algorithm often tends to generate spurious edges through the cells. In cases where a cell is imaged at multiple slices along the Z-stack and is over-segmented in one of the slices, the tracker can identify this over-segmentation error as a spurious ‘spatial cell division’ event. Clearly, this is not legitimate and is a result of faulty segmentation. Additionally, cell merging in the temporal direction (again an impossible event) can arise from under-segmentation, where the watershed algorithm fails to detect a legitimate edge between two neighboring cells. The intuition behind the proposed method in this paper is to reduce the over-segmentation errors by minimizing the spurious spatial ‘cell divisions’ and reduce the under-segmentation errors by minimizing the number of merged cells. Specifically, for frame S_k^t , the k_{th} image slice at time point t , we are going to minimize the number of spurious ‘cell divisions’ between it and its spatial neighbor S_{k-1}^t , and the number of spurious cell merging events in S_k^t from its temporal predecessor S_k^{t-1} , as shown in Figure 3.1.

The spatial and temporal correspondences between cell slices across neighboring confocal images can be represented through correspondence matrices. As an example, $C_{(k-1,k)}^t$ is the

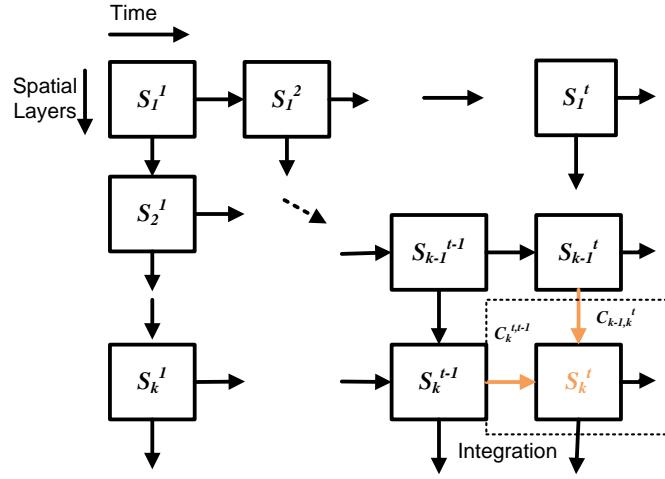


Figure 3.1: Schematic showing on how to integrate the spatial and temporal trackers for 4D volumetric image stacks.

correspondence matrix from the slice S_{k-1}^t to S_k^t , and $C_k^{(t,t-1)}$ is the correspondence matrix from the frame S_k^t to S_k^{t-1} . Each element $C[i, j]$ in correspondence matrix C denotes whether two cell i, j in two images (may be at different time instances or different slices at the same time) are the same cell or not, i.e.,

$$\begin{cases} C[i, j] = 0, & \text{if } i \text{ and } j \text{ are not matched,} \\ C[i, j] = 1, & \text{if } i \text{ and } j \text{ are matched.} \end{cases} \quad (3.1)$$

For a cell $i \in S_k^{t-1}$, if there exists a cell pair $\{j_1, j_2\} \in S_k^t$ such that $C[i, j_1] = C[i, j_2] = 1$, then we can identify that there is a cell division event where the cell has divided into two daughter cells j_1 and j_2 . On the other hand, if for a cell $j \in S_k^t$, there exists a cell pair $\{i_1, i_2\} \in S_k^{t-1}$ such that $C[i_1, j] = C[i_2, j] = 1$, then a cell merging event is readily detected, where cells i_1 and i_2 from the previous slice S_k^{t-1} have merged into a single cell j in slice S_k^t . This is clearly the result

of under-segmentation on the slice S_k^t that needs to be corrected. In a very similar way, the errors caused by over-segmentation can also be detected from the tracking output. For any two spatially consecutive slices S_{k-1}^t and S_k^t in the same Z-stack, if multiple cells in S_k^t correspond to one single cell in S_{k-1}^t (i.e. there exists a one-to-many correspondence between cells in spatially consecutive slices), an over-segmentation error in the slice S_k^t is detected.

The under-segmentation and the over-segmentation error can be quantitatively represented as the number of faulty ‘cell merging events’ and the ‘spurious cell divisions’ as obtained from the spatio-temporal correspondences generated by the tracker. For example, the number of the illegal cell-merging events can be counted as,

$$\begin{cases} Q_k^{(t,t-1)} = \{\text{cell } i \text{ in } S_k^t | \exists \text{ a pair } (j_1, j_2) \text{ in } S_k^{t-1} \text{ for which } C_k^{(t,t-1)}[i, j_1] = C_k^{(t,t-1)}[i, j_2] = 1\}, \\ N_k^{(t,t-1)} = \text{sizeof} (Q_k^{(t,t-1)}) , \end{cases} \quad (3.2)$$

where $Q_k^{(t,t-1)}$ is the set of under-segmented cells in S_k^t , and $N_k^{(t,t-1)}$ is the total number of such cells. Similarly, we can compute $N_{(k-1,k)}^t$ as the number of spurious cell divisions (as described previously) in the spatial tracking from S_{k-1}^t to S_k^t .

The optimization goal here is to minimize the number of spurious cell divisions ($N_{(k-1,k)}^t$) for the frame S_k^t from its upper slice S_{k-1}^t and the number of cell merging events ($N_k^{(t,t-1)}$) in S_k^t from its previous slice S_k^{t-1} . As can be seen in Figure 4C, the error caused by over-segmentation, $N_{(k-1,k)}^t$, monotonically decreases (red curve in Figure 3.2A) with the increment in threshold h , whereas, the error due to under-segmentation, $N_k^{(t,t-1)}$, monotonically increases (green curve in Figure 3.2A) with h . Hence, the optimal segmentation result can be obtained through finding the value of h for which the summation $h_k^t = \min_h (N_k^{(t,t-1)} + N_{(k-1,k)}^t)$ attains a minimum (as the

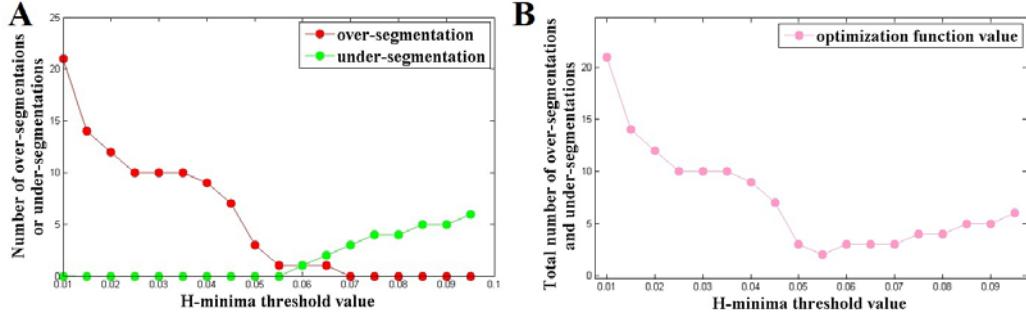


Figure 3.2: (A) The illustration of the number of faulty cell merging events (green plot) or spurious cell divisions (red plot) with respect to the changing of H-minima threshold h , in one noisy image slice. (B) The total number of over-segmentation and under-segmentation errors with respect to the changing of H-minima threshold h .

lowest point in the pink curve in Figure 3.2B). The cost function $h_k^t = \min_h \left(N_k^{(t,t-1)} + N_{(k-1,k)}^t \right)$ is essentially an indicator of the overall error in segmentation (combining both over and under-segmentation) and can be optimized by varying the H-minima threshold h for S_k^t . Formally, the optimal value h_k^t is found as a solution to the following optimization problem:

$$h_k^t = \min_h \left(N_k^{(t,t-1)}(h) + N_{(k-1,k)}^t(h) \right). \quad (3.3)$$

With the variation of h_k^t (by either increasing or decreasing), the cost function decreases to a minimum (ideally 0). The threshold h_k^t for which the cost function attains this minimum is the optimum value of the threshold for H-minima Transformation. Note that, given a value of h , we can compute N , although an analytical representation relating the two is unknown.

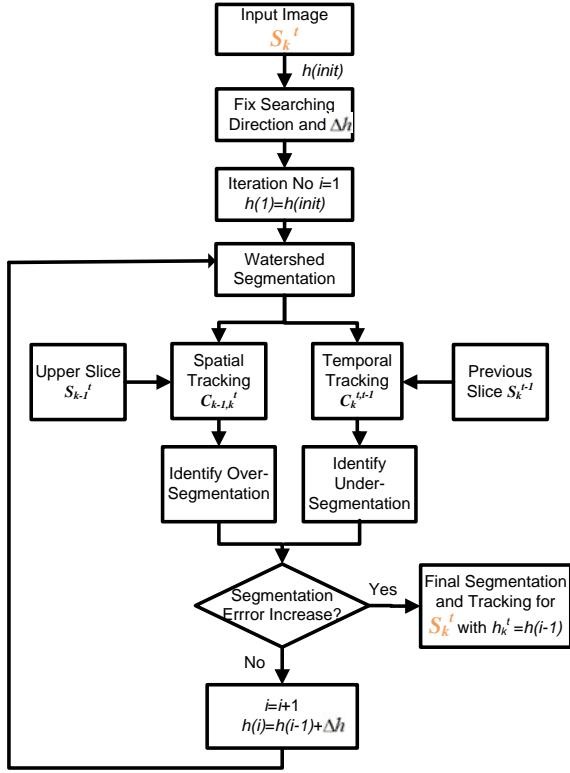


Figure 3.3: Adaptive segmentation and tracking scheme for a certain image slice S_k^t (the k th slice at the t time point).

3.2.2 Optimization Scheme

Given 4D image stack series $\{S_k^t, k = 1 : K, t = 1 : T\}$ as shown in Figure 3.3 (where k is the index for depth and t is for time), the algorithm proceeds as follows:

1. We first estimate $h_{k(\text{init})}^t$ for every image slice that minimizes the variances in cell areas in each of those slices and perform watershed segmentation with these estimated thresholds.

These act as the initial estimates of the segmentation thresholds in our algorithm.

2. For image slices $\{S_k^t, k = 2 : K, t = 1\}$, we compute the optimum watershed thresholds

$h_k^t \in [h_{\min}, h_{\max}]$ by minimizing the number of cell divisions $N_{(k-1,k)}^1$ in each of these slices. We repeat this process for slices $\{S_k^t, k = 1, t = 2 : T\}$ and obtain optimum thresholds by minimizing $N_1^{(t,t-1)}$.

3. For other image slices $\{S_k^t, k = 2 : K, t = 2 : T\}$, we vary the thresholds h_k^t in the range

$[h_{\min}, h_{\max}]$ starting from the initial estimate $h_{k(\text{init})}^t$. The search direction, i.e. either of $h_k^t > h_{k(\text{init})}^t$ or $h_k^t < h_{k(\text{init})}^t$ is chosen such that there is a decrease in the cost function $F_k^t = (N_k^{(t,t-1)} + N_{(k-1,k)}^t)$ with a change in h_k^t along the search direction. This can be done using a simple exhaustive search in a local neighborhood around $h_{k(\text{init})}^t$. Once the search direction is fixed, we keep on varying h_k^t along the search direction until we encounter an increment in the cost-function. At this value of h_k^t , beyond which the cost function increases, a minimum of the cost function is reached. We stop the search at this point and set h_k^t as the optimum watershed segmentation threshold for S_k^t . We segment S_k^t with this threshold and obtain optimum tracking result while tracked from its immediate neighbors.

The search method described above is guaranteed to converge to a minimum of the cost function. As we have observed throughout our experiments, this minimum is generally the global minimum, though it cannot be shown analytically that a global minimum is always ensured. If run time of the algorithm is not an issue, an exhaustive search method over the entire range $[h_{\min}, h_{\max}]$ can also be employed to guarantee that a global minimum is always attained.

3.3 Fusion of Spatial and Temporal Tracks

As introduced in Section 1.1.1, we know that the relative positions of SAM cells not only have temporal consistency, but they also have spatial consistency. It is because every cell, in most cases, is represented in three consecutive optical slices (as shown in Figure 2.2). This feature was exploited by the local graph matching method to find the spatial correspondence of cells across consecutive slices taken at different depths. Subsequently, the spatial and temporal correspondences can be fused together to obtain a single unified track. This allows us to track those cells that may have been poorly imaged in one slice, but are of a higher quality in a neighboring slice.

We now show how to fuse the spatial and temporal correspondences in order to get the final cell tracks. As explained in Figure 2.2, let us assume a cell is separated 3 slices (that is true for most of the cells), so we need to integrate the tracking results in every 3 consecutive slices. The fusion process is described in Figure 3.4 and below.

As illustrated in Section 3.2.1, in the tracking process, we build a correspondence matrix C to denote whether two cells c_i, c_j in two images (may be at different times or different slices at the same time) are the same cell or not. If the two cells are the same, we set the correspondence value $C[i, j]$ to 1, otherwise we set it to 0. The correspondence across time instants t and $t + 1$ at slice S_k is denoted as $C_k^{(t,t+1)}$, and the correspondence across slices S_k and S_{k+1} at time t as $C_{(k,k+1)}^t$. The fusion process is composed of the upward fusion matrix $C_{(k-1,k)}^{(t,t+1)}$ and downward fusion matrix $C_{(k,k+1)}^{(t,t+1)}$, which can be easily proved to be products of the individual correspondence matrices as follows:

$$\begin{aligned} C_{(k-1,k)}^{(t,t+1)} &= C_{(k,k-1)}^t \cdot C_{k-1}^{(t,t+1)} \cdot C_{(k-1,k)}^{t+1}; \\ C_{(k,k+1)}^{(t,t+1)} &= C_{(k,k+1)}^t \cdot C_{k+1}^{(t,t+1)} \cdot C_{(k+1,k)}^{t+1}. \end{aligned} \quad (3.4)$$

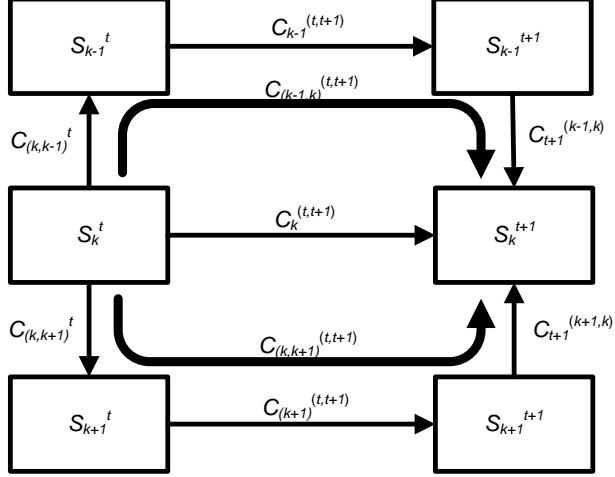


Figure 3.4: The diagram showing integration of the tracking results in 3 consecutive slices by fusing the spatial and temporal tracks to compute the final cell correspondence.

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

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

With this fusion process, we can find new cell pairs, which are not identified by single slice tracking, but can be found through the adjacent slices, and we can track most of the properly segmented cells. The effect of the fusion in identifying additional cells is shown in Figure 3.5.

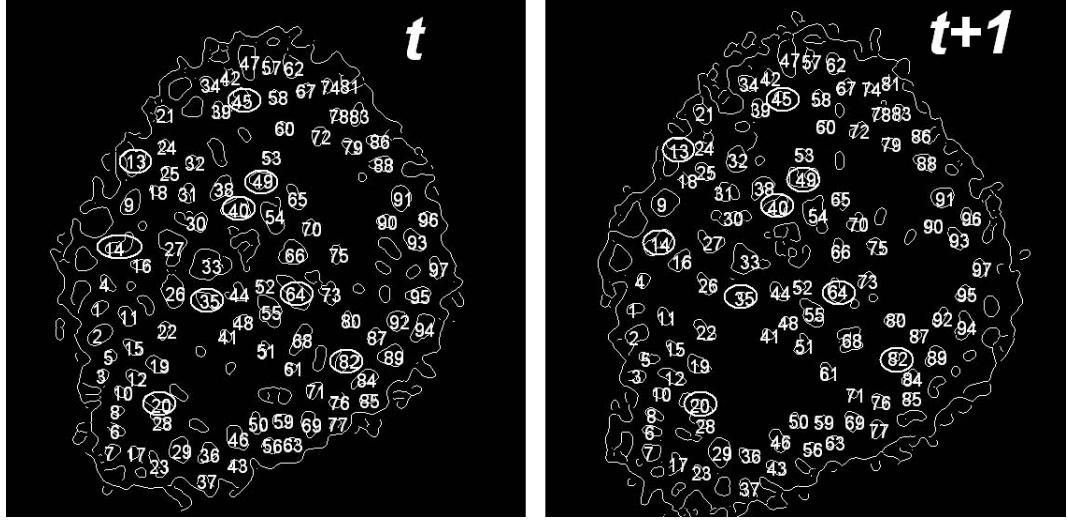


Figure 3.5: The tracking results after the spatial and temporal track fusion process, where nine newly tracked cells are denoted by white circles.

3.4 Computing Cell Lineages

The cell cycle lengths of cells located in distinct regions of SAMs vary from 18 hours to 90 hours and plastochron length which reflects the time of initiation of successive organ primordia varies from 14-20 hours [64, 16]. Therefore understanding the cellular basis of morphogenesis requires tracking of cell lineages over longer periods of time. Computing cell lineages just by repeating the two-frame tracking process over the entire time series will result in a gradual loss of cells in the later time instants due to imaging noise, improper segmentation and tracking errors. In order to maintain the consistency of the cell lineages for long time periods, we have developed a method to integrate the tracks from multiple slices at a given time instant with that of tracks obtained across time periods.

The idea here is that the lineage may be computed through multiple temporal paths (not

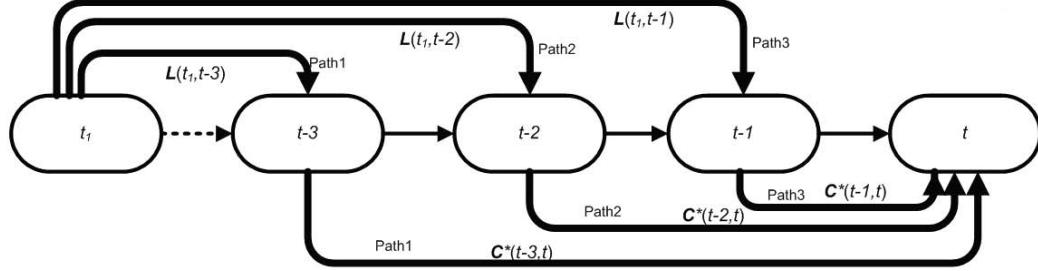


Figure 3.6: Cell lineage (from time t_1 to t) computation process by fusing along multiple tracking paths (Path 1, Path 2 and Path 3).

to be confused with the spatial paths of Figure 2.13) as shown in Figure 3.6 and then fused together.

Let us assume that t_1 denotes the first time instant. Then the lineage from the first time instant t_1 to any time instant t ($t > t_1$), i.e. $L(t_1, t)$, can be mathematically represented as follows

$$L(t_1, t) = \bigcup_{i=1,2,3, t-i \geq t_1} L(t_1, t-i) C^*(t-i, t) \quad (3.5)$$

where C^* denotes the output of the fusion process and this ‘union operator’ \cup will combine the lineage results from the 3 paths shown in Figure 3.6 based on a majority voting process. (Since the lineages are computed for the cells after the fusion process of the previous section, we drop the index k in the correspondence matrix.) In this process, a matched cell pair is accepted if the matching result agrees in a majority of the paths. To be specific, a pair that is matched in all those three paths should agree in at least two, a pair matched in two paths should agree, while a pair matched in only one path will be accepted. The reason why a cell pair may not be matched along all paths is because not all cells may be imaged or detected at each time instant. This process is explained diagrammatically in Figure 3.7. An example of the computed cell lineages along 24 time

instants (72hours) is presented in Figure 3.8.

By this method, a cell that may not have been tracked along a particular path can still be part of final lineage computation if it was tracked in any of the other paths. Though this method introduces a delay in the system, this is not a major concern in this application domain as it will lead to higher accuracy in tracking and real-time performance is not necessary for this application domain.

3.5 Experimental Results

The experimental results are shown on 4D volumetric cell image stacks obtained from plant cells and observed along 24 consecutive time instants, with the time interval of 3 hours between two consecutive instants. The cells were segmented by existing level set segmentation methods [40] or watershed method [45]. The time series of confocal Z-stacks of SAMs requires to be aligned and the 3D alignment can be done by using available software packages that utilize information theory to maximize mutual information across image stacks to register images at subpixel resolution [54]. The 3D registration/alignment can also be done by using appropriate image registration module in commercial software packages, such as AMIRA (Visage Imaging).

3.5.1 Integrated Segmentation and Tracking

The adaptive segmentation and tracking method is run on every 2 consecutive images in both the spatial and temporal directions. The thresholds of H-minima for images in the given 4D image stack are determined sequentially along the direction of the arrows shown in Figure 3.1. In the segmentation, we normalized the image intensities in the range of [0 1] and set the searching

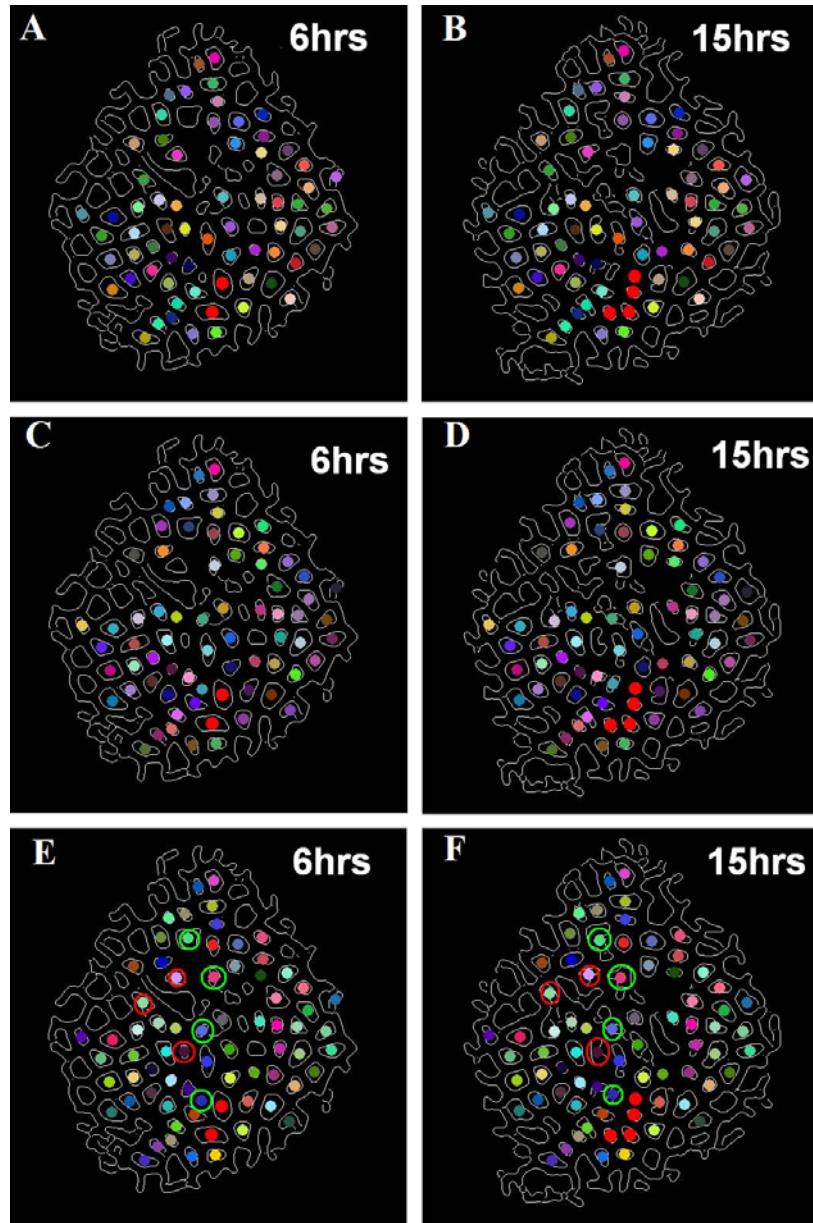


Figure 3.7: (A, B) Cell lineages from time 6 hr to 18 hr, computed by Path 1. (C, D) Cell lineages computed by Path 2. (E, F) Cell lineages computed by the combination of Path 1 and Path 2. Note that the system may lose correspondence of some cells along a particular path, but maintained their correspondences in an alternate tracking path. Compare (A), (C) and (D), cells in green circles were lost in (A) and (B) while cells in red circles were lost in (C) and (D), but all of them were gained back in (E) and (F) due to the fusion process.

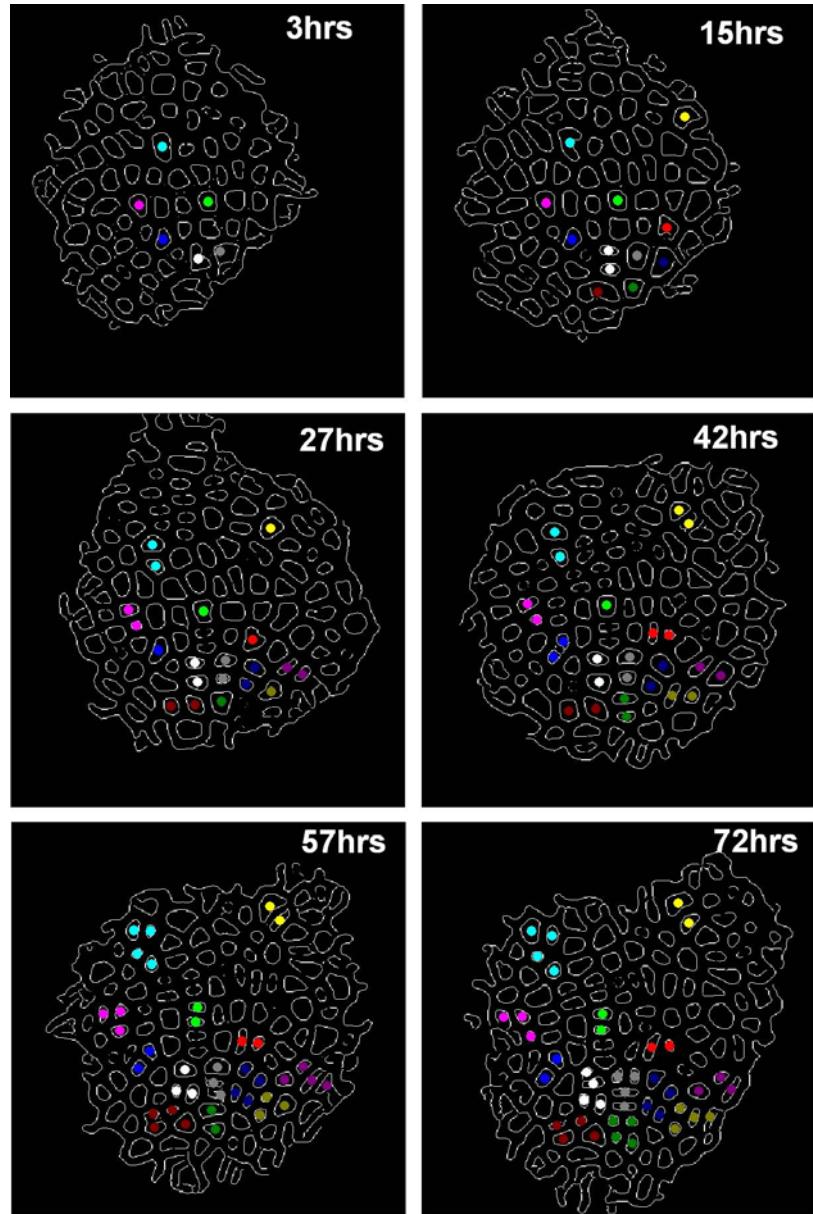


Figure 3.8: Computed cell lineages of SAM cells. The cell lineages across 24 time instants (a total of 72 hours of time lapse data) were computed (only 6 of them are show here), where the same color denotes the cell lineage. After 72 hours of tracking, most of cells have divided at least once with several of them divided twice or thrice.

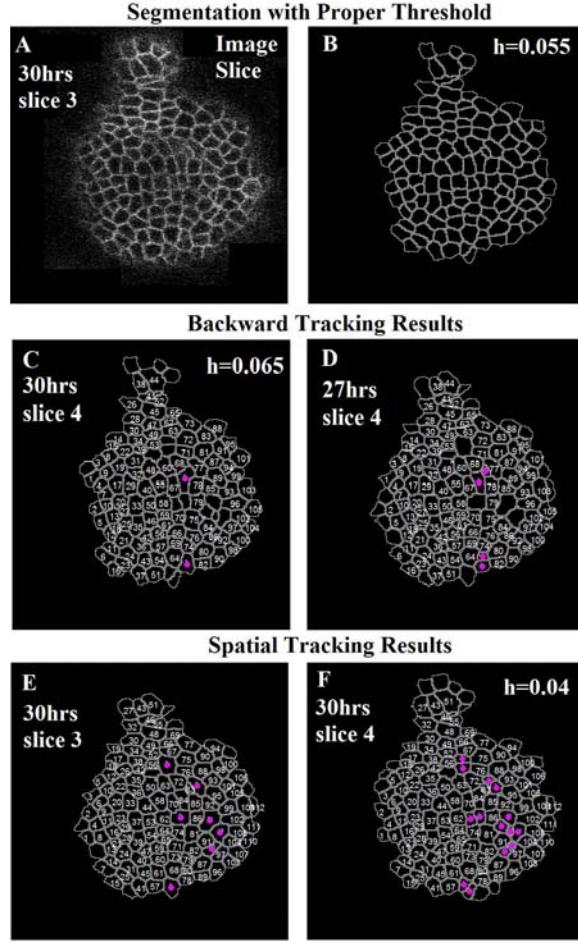


Figure 3.9: Illustration of faulty cell merging events caused by under-segmentation or spurious cell divisions caused by over-segmentation. (A) The fourth slice of confocal Z-stack taken at 30th hr time point. (B) The correct segmentation result with the optimal H-minima threshold value 0.05, which was found by the proposed adaptive method; (C) and (D) are the temporal tracking results (from 30th hr time point to the 27th hr time point) with faulty cell merging events (denoted by purple dots), which are caused by under-segmentation with H-minima threshold value 0.065. (E) and (F) are the spatial tracking results (from 3rd confocal slice to the 4th confocal slice along the z-scale) with spurious cell divisions (denoted by purple dots) and this is caused by over-segmentation. The same number denotes two corresponding slices along the Z-direction.

range for the optimal H-minima threshold h in [0.005 0.09]. The step size used in the search was 0.005, and the sign depends on the search direction ($h_k^t > h_{k(init)}^t$ or $h_k^t < h_{k(init)}^t$). We manually verified the accuracy of the cell lineages obtained by the proposed algorithm.

Using the adaptive segmentation and tracking, we can find the optimal threshold for watershed segmentation with minimal over-segmentation and under-segmentation. The whole idea can be illustrated through an example in Figure 3.9. A is the original image at 30hrs and in the 4th slice, and B is the segmentation result by our proposed adaptive method with the optimal H-minima threshold $h=0.055$, which is found by minimizing the spatial faulty cell divisions (caused by over-segmentation) and temporal faulty cell merging events (caused by under-segmentation). Using watershed segmentation with other H-minima thresholds, there will be either spatial over-segmentation or temporal under-segmentation. For example, there are two faulty cell merging events in the temporal tracking (as shown in C and D) which indicate under-segmentation caused by too high a H-minima threshold ($h=0.065$), while the spurious cell divisions in the spatial tracking (as shown in E and F) indicate over-segmentation caused by too low a H-minima threshold ($h=0.04$).

3.5.2 Estimated Cell Lineages

In Figure 3.8, we show the tracking results on computing the cell lineages by repeating the two-frame tracking process over the entire data collection of 72 hours. Figure 3.10 is an example of the estimated cell lineages across 24 time points (72 hours) with 5 specific time points shown. The first row is the original image slice series, the second row shows the segmented cells and the estimated cell lineages using the adaptive segmentation and tracking algorithm, while the third row is the result using the level set segmentation, and cell segmentation and tracking are not integrated.

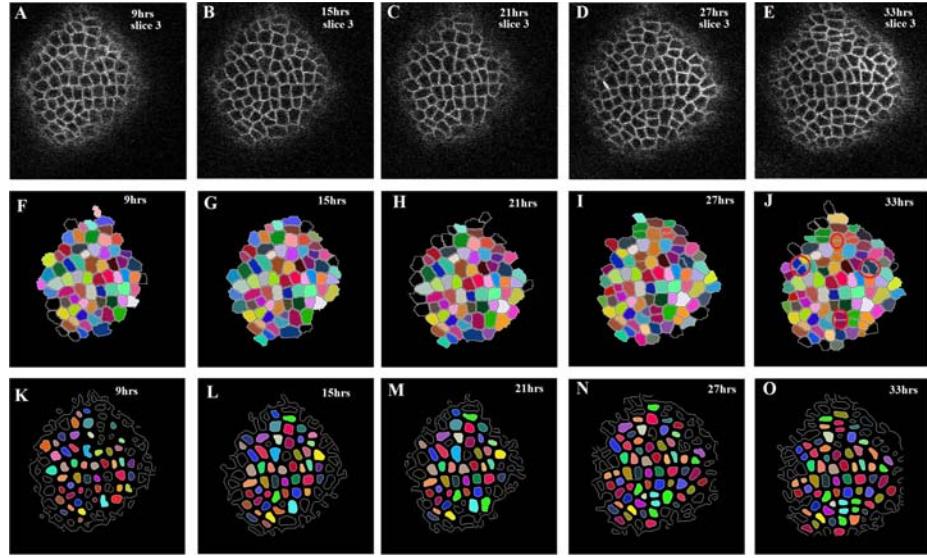


Figure 3.10: An example of the estimated cell lineages. The first row is the original image slice series at the 9th hr, 15th hr, 21st hr, 27th hr and 33rd hr. The second row shows the segmented cells and the estimated cell lineages using the adaptive segmentation and tracking algorithm. The third row is the result using the level set segmentation, and cell segmentation and tracking are not integrated. The same color in consecutive frames denotes the same cell. Cell division examples are denoted by red circles in the second row.

The same color in consecutive frames denotes the same cell at different time points. Some cell division examples are highlighted by red circles. It demonstrates the difference in segmentation and cell lineages achieved using level set segmentation and watershed segmentation. In Figure 3.11, the segmentation and tracking results in selected 3D volumetric image stacks along 3 time instances (6hrs, 9hrs and 15hrs) are shown. The tracker is able to compute both spatial and temporal correspondences with high accuracy as a result of improved segmentation.

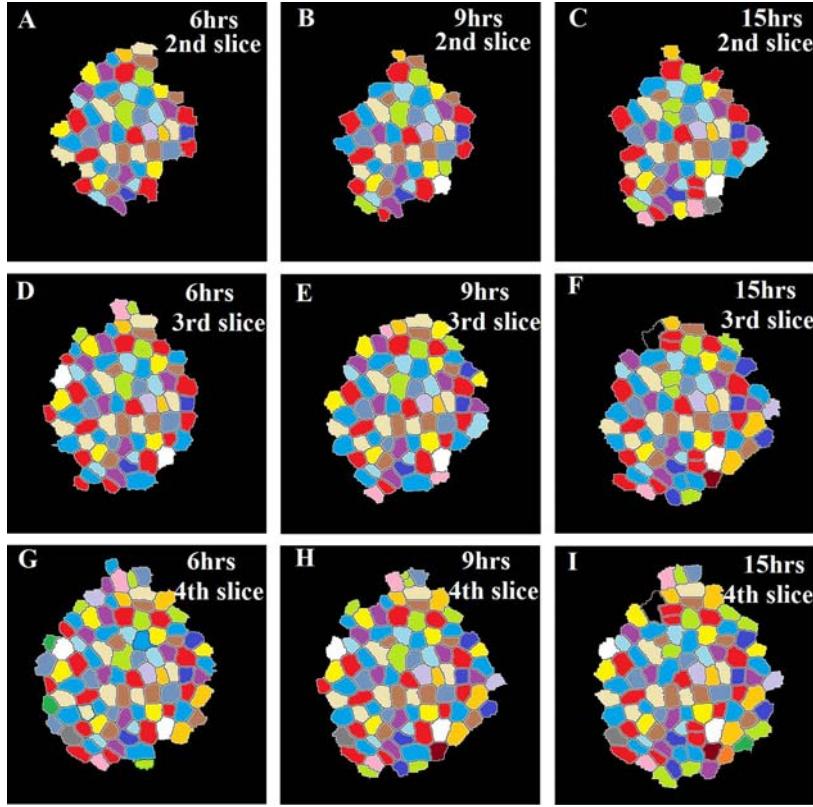


Figure 3.11: The segmentation and tracking results in 3D stacks at selected time instances. The segmented cells (by watershed method) shown in same color across consecutive slices (2nd slice, 3rd slice and 4th slice) represent same cells.

3.5.3 Quantitative Results

In order to estimate the effectiveness of fusion of spatial and temporal tracks of cells in increasing the robustness of tracking, we compared the number of cells that are correctly tracked across two consecutive time instants with or without the fusion process (Table 3.1). The number of the cells obtained from manual counting both in the unsegmented image data and the automatically segmented data were used as ground truth to calculate the efficiency of the fusion process. The number of cells identified from three consecutive slices across two time instants from both the manual

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

Table 3.1: Cells being tracked manually (in the unsegmented data and in the segmented data by Level-Set method, the latter is in the brackets) and automatically (before the fusion process and after the fusion process). We have chosen some representative examples from the dataset.

counting and automated tracking are also listed for comparison. This analysis reveals that fusion of the tracking output obtained from three independent consecutive slices of a given set of cells improves the tracking efficiency by increasing the robustness in obtaining temporal correspondence of cells (Table 3.1).

The shape of the SAM and the imaging method used for obtaining thin cross sections will result in sequential distribution of cells along different optical slices. Therefore obtaining correspondences among consecutive slices (Z-sections) will be crucial not only to introduce robustness to the tracking process but also to account for all cells of SAMs. However, maintaining the stability of tracks of cells located at different depths along the Z-axis of image stack can be challenging. Therefore, we tested the stability of tracks obtained for the topmost 12 slices of a Z-stack which account for all cells located within the CZ and the PZ of the L1 layer (Table 3.2). The comparison of tracking output for any two time instants with manual counting revealed that up to 95% of the correctly segmented cells could be tracked and up to 93% of cell division events could be detected (Table 3.2).

We have quantified the performance of our long-term cell-tracking algorithm in maintain-

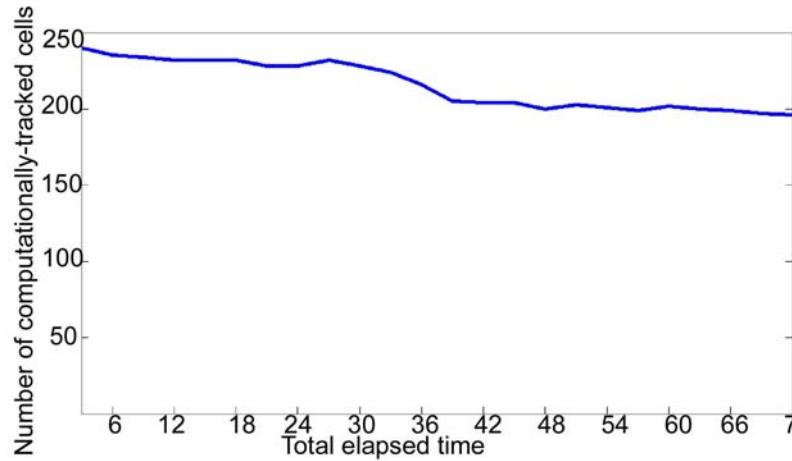


Figure 3.12: Total number of automatically-tracked cells across different time instants measuring up to 72 hours. Note the stability of tracking algorithm as most of the cells identified at first time instant could be detected at later time instants. The sudden decrease in the number of correctly tracked cells at 36 hours is mostly due to the incorporation of cells located at the edge of the PZ and subsequent displacement of cells into actively growing organ primordia.

Time Interval (hours)	Automatically tracked cells	Automatically tracked cell divisions
3-6	219/218	1/1
6-9	251/261	6/7
9-12	263/274	8/8
12-15	231/240	20/21
15-18	185/196	28/30

Table 3.2: Total numbers of cells and cell divisions tracked in the stack of the top 12 slices in different time intervals. The denominator is the ground truth in the segmented data by Level-Set method (the number of cells tracked manually or the number of cells divisions identified manually); the numerator is the tracking result obtained from the algorithm developed in this study.

Dataset	Level-Set without integration	Watershed without integration	Watershed with integration
Dataset1	71%	90%	98 %
Dataset2	36%	83%	95 %

Table 3.3: The number of cells correctly tracked using level-set segmentation, the watershed segmentation but no adaptation, and the integrated segmentation and tracking method shown in the left column.

ing consistency of the cell lineages in time lapse series taken at 3 hour intervals and extending up to a total of 72 hours (Figure 3.12). We considered cells located in the top eight slices of each image stack for this analysis. This analysis reveals that about 90% of the cells that were identified in the first time instant could be found 36 hours later. The sudden and a small decrease in the number of correctly tracked cells at 36 hours is mostly due to the displacement of PZ cells into actively developing organ primordia. We have also noticed that certain cells that were undetected from tracked output at 21 hours could be recovered at 27 hours indicating the usefulness of the method in long-term tracking. The consistency of tracked output was maintained over 72 hours despite repeated divisions of corresponding cells at different time intervals. The frequency of cell division events at different time intervals reported by the tracker revealed temporal fluctuations in mitotic activity of SAM cells which is consistent with results obtained from manual tracking methods described earlier [16]. Because cells are represented in more than one consecutive slice, there is a possibility that they may be tracked several times. In order avoid this redundancy in cell lineage estimation process, care was taken not to count a given cell more than once by finding correspondences of given cells in the previous and the subsequent slices of the image stack. Similarly, when computing the final track for each cell, we provide only one track that was obtained from fusing the spatial and temporal correspondences.

Dataset	Level-Set without integration	Watershed without integration	Watershed with integration
Dataset1	25hrs	50hrs	56hrs
Dataset2	6hrs	22hrs	33hrs

Table 3.4: The average time period over which the lineages could be tracked is shown in the right column.

In order to verify the overall improvement of the integrated segmentation and tracking algorithm, we compared the cell tracking statistics from the integrated method [10]. The number of correctly tracked cells obtained by tracking images in consecutive time points with a 72 hour period is compared in Table 3.3 and Table 3.4. The comparison is done on different datasets and we can see significant increase in the percentage of tracked cells obtained from the integrated method. Another important data to obtain is the cell's lineage lengths. Table 1 also shows the comparison of the average length of cell lineages between the integrated method and the algorithm with no integration [10], and confirms that the integrated method greatly improves the tracking results, especially in the deeper central regions of the SAM. There are 27 more cells (21 of them are in the deeper central region) that maintain their lineages over a complete 72 hours by the integrated method, compared to the method without integration [10].

3.5.4 Statistical Analysis

The closed loop segmentation and tracking approach minimizes over-segmentation and under-segmentation errors by adapting the segmentation parameters based on the tracking output. This improvement in cell segmentation and tracking has enabled us to obtain a set of sufficiently accurate statistics that could be very useful in plant cell quantitative growth analysis and modeling. With the greatly improved tracking results, we can quantify cellular dynamics such as cell cycle

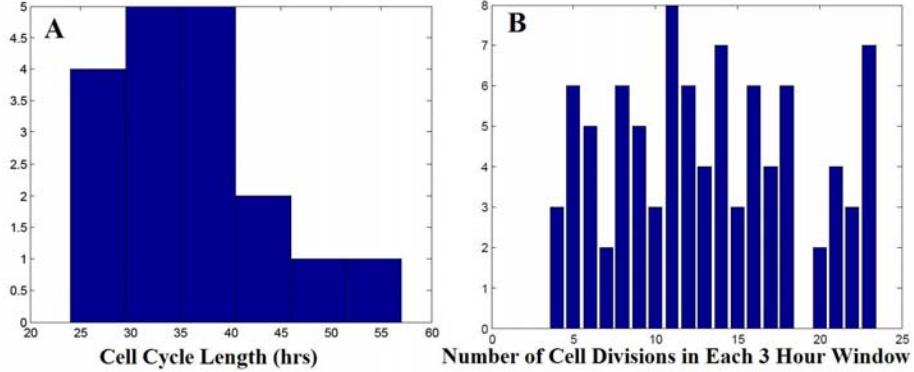


Figure 3.13: Statistical analysis results based on the segmentation and tracking results using our proposed adaptive method. (A) The histogram showing cell cycle length. (B) The number of cell divisions across each 3 hour intervals.

length and number of cell division events within a given interval (Figure 3.13A and B). The statistics obtained through the proposed automated framework show very close similarity to the nature of cell growth and division statistics obtained in the earlier work [16]. Such statistical information is extremely useful in modeling the cell growth dynamics [38, 65, 66].

3.6 Conclusion

In this section, we proposed an adaptive segmentation and tracking framework. The novel contribution of this adaptive framework lies in its ability to segment and track cells, through the use of a closed loop, adaptive segmentation and tracking approach. The tracking output acts as an indicator of the quality of segmentation and in turn the segmentation can be improved to obtain better tracking results. We construct an optimization function that minimizes the segmentation error, which is, in turn, estimated from the tracking results. By testing the integrated segmentation and tracking pipeline on multiple datasets, we confirm that this adaptive approach significantly improves

both tracking and segmentation when compared to an open loop framework where segmentation and tracking modules operate separately.

Chapter 4

A Multilinear Framework for Species Classification Using 3D X-Ray Transform

4.1 Introduction

As a very highly diverse group of invertebrates, nematodes are very difficult to distinguish.

As introduced in Section 1.2, nematode species are usually imaged by DIC microscope and stored in Digital Multi-focal Images. The DMI bioimage stack, as a new way for documentation and communication of specimen data in the biological field, has the advantage that the morphological information for a transparent specimen can be captured in the form of high-quality bioimage stack, representing individual focal planes through the specimens body. However, the objects of interest (such as nematode species) in such bioimage stacks could have dramatic variations of morphology

and intensity from image to image, and the images are very high-quality and large-size (the regular size of each stack is several hundred megabytes). So although the use of DMI includes three-dimensional topology and structure information of the nematode species, how to extract a good feature and then effectively classify these high-throughput nematode volumetric image stacks is still not clear.

Most of the work on image feature extraction and classification is done through 2D image processing methods [26, 27, 28]. Since we want to integrate all the 2D feature information across different image layers, we propose the use of 3D X-Ray Transform, which will explore the 3D structural information out of a multifocal image stack (as shown in Figure 1.3) from different viewing directions. As shown in Figure 4.3 B-F, the 3D X-Ray Transform of the 1st DMI stack in Figure 1.3 gives us different information about this stack at 5 different angles. Recent methods for 3D feature extraction like spatio-temporal interest points are not suited for this purpose because the images in multiple focal planes have different characteristics compared to a space-time volume [67].

For the classification of DMI stacks, there are multiple factors that need to be analyzed - shape, texture, viewpoint of the X-Ray Transform, different instances within the same class and different classes of specimens. For this purpose, we embed the 3D X-Ray Transform within a multilinear framework. In fact, in the face recognition field, the multilinear image analysis method plays an effective role for the recognition of natural face images formed by different factors, such as expression, illumination, pose and so on [68, 69]. In multilinear analysis, the facial image ensemble is represented as a high dimensional tensor. By a high-order generalization of Principal Component Analysis (PCA) and high-order singular value decomposition (HSVD), the collection of facial

images is represented by separating the different modes underlying the formation of facial images. This idea can be applied to the classification of DMI stacks, since they also represent the combination of various factors - poses (viewpoints) of the X-Ray Transform, different instances within the same class and different classes of specimens. Therefore it is intuitive to use the multilinear method for feature extraction and classification of the DMI data and we do so by combining it with the 3D X-Ray Transform.

This multilinear X-Ray Transform (MXRT) based classification method allows us to model the inter-relationships between the different variables, while at the same time allowing independent analysis along each dimension. We will show the application of this feature extraction and representation technique in classification of nematodes - a species that is very difficult to classify since there are over 80,000 of them. This is possibly the first approach that looks at the automated analysis of DMI images, which is becoming increasingly popular in biology.

4.2 Broad Overview of Solution Strategy

Here we will show how to derive features for DMI data sources that take into account the shape and texture information across the focal planes, as well as the variability when viewed from different directions. We will show how to design classifiers based on this Multilinear X-Ray Transform feature representation and provide experimental results on its usefulness in large real-life databases. The diagram of this proposed method is shown in Figure 4.1, and the training and testing steps are listed below.

1. Training

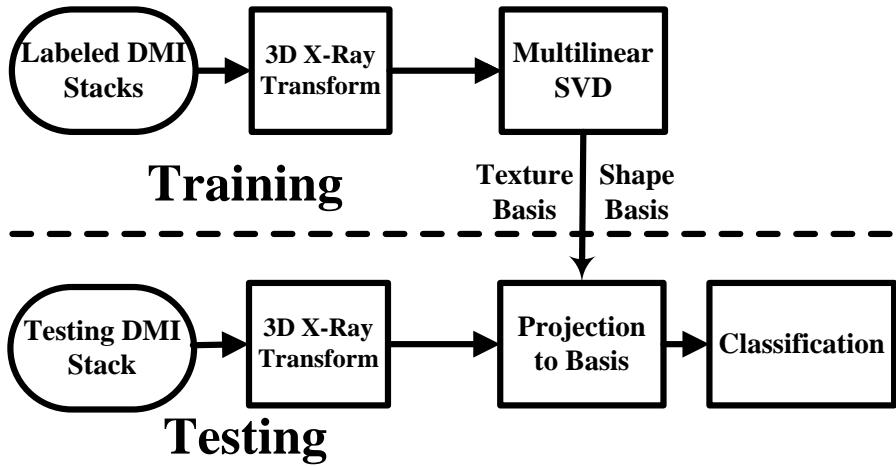


Figure 4.1: The diagram for training and recognition of the proposed 3D X-Ray Transform based multilinear analysis method.

- (a) The inputs are labeled DMI stacks.
- (b) For each training DMI image stack, compute the 3D X-Ray Transform at defined angles (poses).
- (c) For 3D X-Ray Transform output images at each pose (angle), obtain the tensor texture basis and tensor shape basis through multilinear SVD.

2. Testing

- (a) Given an unknown DMI stack, get the 3D X-Ray Transform output images at the same defined poses (angles) as in the training procedure.
- (b) Project the 3D X-Ray Transform output images at each pose to the corresponding tensor texture basis and tensor shape basis, and obtain the distance matrices for texture and shape.

- (c) Get the distance matrix by combining the tensor texture and shape distance matrices, and then find the class label through nearest neighbor criteria for the 3D X-Ray Transform output image at each pose of the given DMI stack.
- (d) Obtain the final recognition result by a majority voting process between the labels for each X-Ray Transform pose from the previous step.

4.3 Detailed Methodology

4.3.1 3D X-Ray Transform

The 2D X-Ray Transform of an image is the projection of the image intensity along a certain angle [70], similar as the Radon Transform [71]. In the 3D case, the X-Ray Transform is the object's projection on a plane, whose orientation is identified by a pair of angles (θ, ϕ) , as shown in Figure 4.2 [72]. It can be mathematically defined as:

$$R(x', y', \theta, \phi) = \int_{z'} S(x, y, z) dz', \quad (4.1)$$

where $S(x, y, z)$ denotes the DMI images along the z direction, x, y and x', y' are the coordinates in the original 2D image plane and the transformed image plane respectively, and the transformation of those coordinates is expressed as [72]

$$\begin{cases} x' = x \cos \phi \cos \theta + y \sin \phi \cos \theta + z \sin \theta \\ y' = -x \sin \phi + y \cos \phi \\ z' = -x \cos \phi \sin \theta - y \sin \phi \sin \theta + z \cos \theta \end{cases}. \quad (4.2)$$

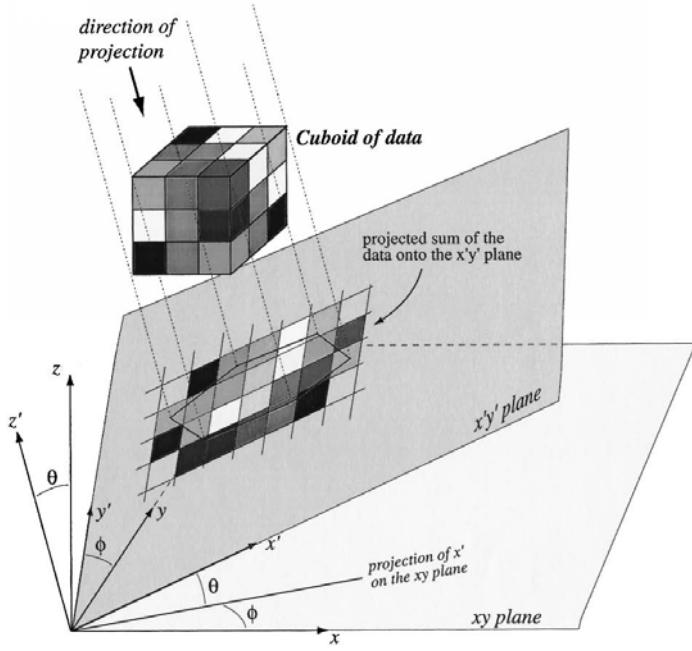


Figure 4.2: The illustration of the three dimensional X-Ray Transform.

As explained in Section 1.1.2, the biggest advantage of DMI is that it can keep the 3D topology and structure information of the specimen. The topological and structural information can be effectively captured by the 3D X-Ray Transform at different poses (i.e., angles) as in Figure 4.3 A. Figures 4.3 B-F are the projections of 3D X-Ray Transform of the 1st DMI stack in Figure 1.3, at 5 different poses, $(\pi/2, \pi/2)$, $(0, \pi/2)$, (π, π) , $(\pi/5, \pi/3)$ and $(\pi/3, \pi/5)$ respectively. From Figure 4.3 we can see that different poses of projections capture very different characteristics of the DMI stack. The 3D X-Ray Transform output images at the above 5 poses will be used for multilinear feature extraction and classification of different DMI stacks in the experiments.

Based on the outputs from 3D X-Ray Transform of DMI stacks, we can make very accurate feature extraction and classification from those multifocal image stacks. Figure 4.4 shows the 3D X-Ray Transform of the 1st, 2nd and 3rd DMI stacks shown in Figure 1.2 at 5 poses. We can

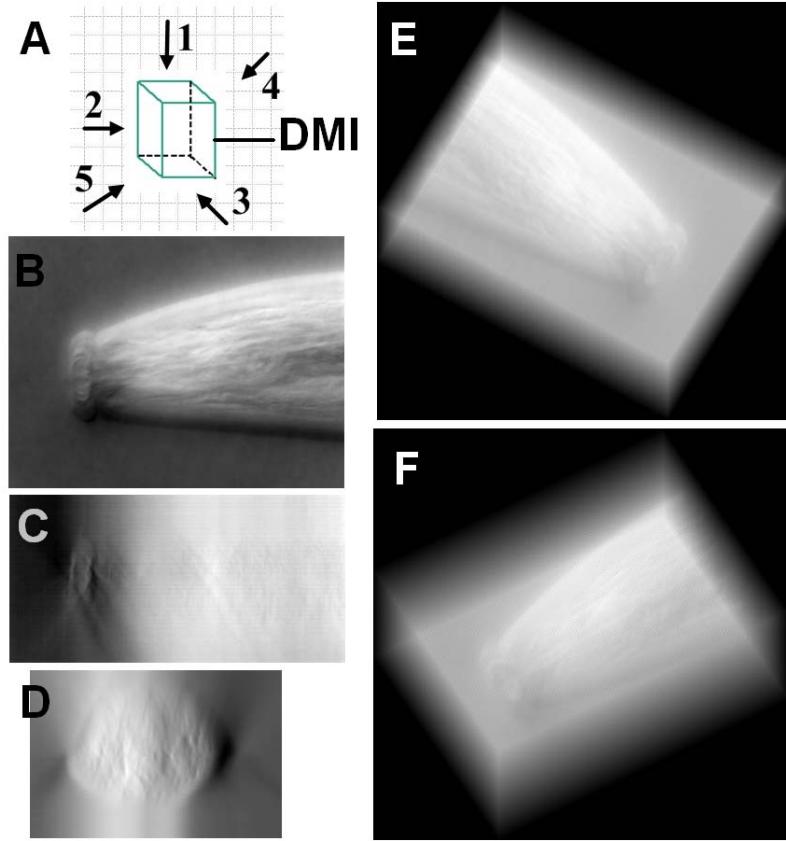


Figure 4.3: 3D X-Ray Transform output images (B-F) at 5 poses (angle pairs) ($1(\pi/2, \pi/2)$, $2(0, \pi/2)$, $3(\pi, \pi)$, $4(\pi/5, \pi/3)$ and $5(\pi/3, \pi/5)$) respectively, of the 1st DMI stack shown in Figure 1.2. A is the illustration of different poses around a given DMI stack for the 3D X-Ray Transform.

see that although the original 2D image frames in the 1st stack and 2nd stack are totally different, their X-Ray Transform projections are similar. If we do the classification just relying on the 2D image frames, we will wrongly classify the first 2 DMI stacks. But by looking at those 3D X-Ray Transform output images, we can make the correct classification.

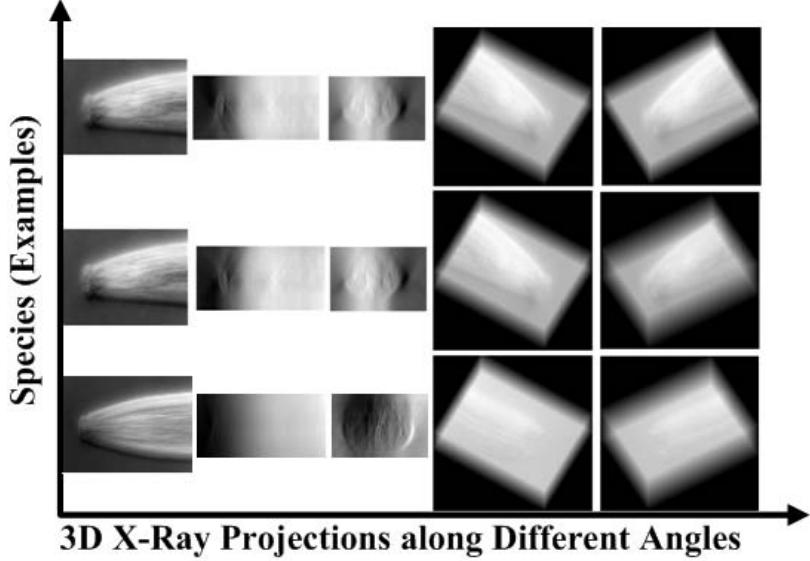


Figure 4.4: 3D X-Ray projections along 5 directions $((\pi/2, \pi/2), (0, \pi/2), (\pi, \pi), (\pi/5, \pi/3)$ and $(\pi/3, \pi/5))$, for the DMI stacks (represented row-wise) shown in Figure 1.2.

4.3.2 Multilinear Subspace Analysis

A tensor is a multidimensional matrix or we can regard it as high order matrix or n-mode matrix; it is the generalization of a vector (first order tensor) and a matrix (second order tensor). For an Nth order tensor $A \in \mathbb{R}^{I_1 \times \dots \times I_n \times \dots \times I_N}$, whose elements can be denoted as $A_{i_1 \dots i_n \dots i_N}$, where $1 \leq i_n \leq I_n$, its mode-n vectors are I_n -dimensional vectors obtained from A by varying index i_n while keeping the other indices fixed. The mode-n vectors are the column vectors of matrix $A_{(n)}$ that results by mode-n flattening the tensor A [69].

As the Singular Value Decomposition (SVD) in matrix analysis orthogonalizes the row space and column space, an N-mode tensor also can be decomposed by N-mode SVD. It is an extension of SVD that orthogonalizes these N spaces and expresses the tensor as the mode-n product [69],

denoted \times_n , of N-orthogonal spaces as

$$D = Z \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_n \mathbf{U}_n \cdots \times_N \mathbf{U}_N, \quad (4.3)$$

where Z is the core tensor, similar as the diagonal singular value matrix in conventional matrix SVD, and \mathbf{U}_n is the mode-n matrix, containing the orthonormal vectors spanning the column space of matrix $\mathbf{D}_{(n)}$ resulting from the mode-n flattening of D . The core tensor Z governs the interaction between the mode matrices $\mathbf{U}_1, \dots, \mathbf{U}_N$.

We will embed the 3D X-Ray Transform of the DMI stacks into the multilinear analysis to learn the shape and texture basis for each of the classes. We will also compare this against using key-frames from each stack replacing the 3D X-Ray Transform output images.

4.3.3 X-Ray Based Tensor Texture Analysis

In the tensor texture analysis, we train the 3D X-Ray projections (as shown in Figures 4.4 and 4.5) in P poses (angle pairs) separately. We introduce the training process and recognition process step by step, as follows.

1. In the training process, the training data are DMI stacks from N classes, and for every class we pick M samples. Let $\mathbf{R}_{p,i}$ denote the 3D X-Ray Transform output image at the p_{th} pose (angle) for the i_{th} DMI stack. Every image is converted to a row vector. The training data of the 3D X-Ray Transform output images at the p_{th} pose is an $N \times M \times L_p$ tensor $D_{\text{tex},p}$, where L_p is the length of $\mathbf{R}_{p,i}$.

2. The multilinear analysis of $D_{\text{tex},p}$ can be expressed as

$$D_{\text{tex},p} = Z_{\text{tex},p} \times_1 \mathbf{U}_{\text{tex,class},p} \times_2 \mathbf{U}_{\text{tex,sample},p} \times_3 \mathbf{U}_{\text{tex,RT},p}, p = 1, \dots, P, \quad (4.4)$$

where the $N \times M \times L_p$ core tensor $Z_{\text{tex},p}$ governs the interaction between the factors represented in the 3 mode matrices: the $N \times N$ mode matrix $\mathbf{U}_{\text{tex,class},p}$ spans the space of class parameters (which is composed of the row vectors $\mathbf{c}_{\text{tex},p,n}^T, n = 1, \dots, N$), the $M \times M$ mode matrix $\mathbf{U}_{\text{tex,sample},p}$ spans the space of sample parameters, while the $L_p \times (N \cdot M)$ mode matrix $\mathbf{U}_{\text{tex,RT},p}$ orthogonally spans the space of all training X-Ray projections at the p_{th} pose. Here the subscript ‘RT’ denotes the 3D X-Ray Transform.

3. The multilinear method enables us to represent each class by the texture basis of the $N \times M \times L_p$ tensor

$$B_{\text{tex},p} = Z_{\text{tex},p} \times_2 \mathbf{U}_{\text{tex,sample},p} \times_3 \mathbf{U}_{\text{tex,RT},p}. \quad (4.5)$$

Then each class $n (n = 1, \dots, N)$ can be represented by a coefficient vector $\mathbf{c}_{\text{tex},p,n}$ (size N), which is the row vector from the class matrix $\mathbf{U}_{\text{tex,class},p}$. For a certain sample m , we can get the basis tensor $B_{\text{tex},p,m} (m = 1, \dots, M)$ of size $N \times 1 \times L_p$. We flatten $B_{\text{tex},p,m}$ along the class mode to get the tensor texture basis matrix $\mathbf{B}_{\text{tex},p,m}$ for the 3D X-Ray projections at the p_{th} pose.

4. In the recognition process, for a given DMI stack, we get its 3D X-Ray projections at those P defined poses (θ_p, ϕ_p) as \mathbf{R}'_p .
5. For each \mathbf{R}'_p , we project it to the corresponding tensor texture basis $\mathbf{B}_{\text{tex},p,m}$ to get the

coefficient vector

$$\mathbf{c}'_{\text{tex},p,m} = \mathbf{B}_{\text{tex},p,m}^{-T} \mathbf{R}'_p, p = 1, \dots, P. \quad (4.6)$$

6. For each \mathbf{R}'_p we obtain the texture distance matrix

$$\begin{aligned} \mathbf{D}_{\text{tex},p}(m, n) &= \|\mathbf{c}'_{\text{tex},p,m} - \mathbf{c}_{\text{tex},p,n}\| \\ (p &= 1, \dots, P; n = 1, \dots, N; m = 1, \dots, M). \end{aligned} \quad (4.7)$$

By the nearest neighboring method, we find the index n_p that yields the smallest value of the texture distance for the p_{th} pose, and we identify \mathbf{R}'_p as class n_p .

7. After we get n_p for all 3D X-Ray Transform output images $\mathbf{R}'_p (p = 1, \dots, P)$, we do the majority voting among n_p to get the final recognition result for this testing DMI stack.

4.3.4 X-Ray Based Tensor Shape Analysis

The analysis of the shape needs to be incorporated into the texture, since the discriminating information is contained in both texture and shape features. We show how to compute the shape feature in the multilinear framework.

In the multilinear shape analysis, the training data are the shape vectors, which can be extracted by the method introduced in [73]. There the authors defined the preshape as the geometric information that remains when location and scale effects are filtered out. A centered preshape is obtained by subtracting the mean from the shape vector and then scaling to norm one. Mathematically,

if the original shape vector is denoted by X , the centered preshape is given by

$$S = \frac{CX}{\|CX\|}, \text{ where } C = I_k - \frac{1}{k}1_k1_k^T, \quad (4.8)$$

where I_k is a $k \times k$ identity matrix and 1_k is a k -dimensional vector of ones. To get shape vector from preshape, we need to remove the rotational component. Since our images are all aligned, we do not need to compensate for the in-plane rotation and hence can work with the preshape.

After we get the preshape vectors $\mathbf{S}_{p,i}$ from the X-Ray Transform output images $\mathbf{R}_{p,i}$, we train the extracted shape tensor in P poses similarly as in the tensor texture case.

1. The training shape tensor, $D_{\text{shp},p}$, at the p_{th} pose is a $N \times M \times J_p$ tensor (J_p is the length of the preshape vectors $\mathbf{S}_{p,i}$ extracted from the 3D X-Ray projections at the p_{th} angle for the i_{th} DMI stack.)
2. The shape tensor analysis yields

$$\begin{aligned} D_{\text{shp},p} = & Z_{\text{shp},p} \times_1 \mathbf{U}_{\text{shp,class},p} \times_2 \mathbf{U}_{\text{shp,sample},p} \\ & \times_3 \mathbf{U}_{\text{shp,RT},p}, p = 1, \dots, P, \end{aligned} \quad (4.9)$$

where the $N \times M \times J_p$ core tensor $Z_{\text{shp},p}$ governs the interaction between the factors represented in the 3 mode matrices: $\mathbf{U}_{\text{shp,class},p}$ spans the space of class parameters (which is composed of the row vector $\mathbf{c}_{\text{shp},p,n}^T$), $\mathbf{U}_{\text{shp,sample},p}$ spans the space of sample parameters, while the $J_p \times (N \cdot M)$ mode matrix $\mathbf{U}_{\text{shp,RT},p}$ orthogonally spans the space of all training shape vectors $\mathbf{S}_{p,i}$ extracted from the 3D X-Ray Transform output images $\mathbf{R}_{p,i}$ at the p_{th} pose.

3. The shape basis of the $N \times M \times J_p$ shape tensor is

$$B_{\text{shp},p} = Z_{\text{shp},p} \times_2 \mathbf{U}_{\text{shp,sample},p} \times_3 \mathbf{U}_{\text{shp,RT},p}. \quad (4.10)$$

For a certain sample m , we get the shape basis tensor $B_{\text{shp},p,m}$ ($m = 1, \dots, M$) of size $N \times 1 \times J_p$. We flatten it along the class mode to get shape basis matrix $\mathbf{B}_{\text{shp},p,m}$ for the 3D X-Ray projections at the p_{th} pose.

4. In the recognition process, for a given DMI stack, we get its 3D X-Ray Transform output images \mathbf{R}'_p , $p = 1, \dots, P$, and extract the shape vector \mathbf{S}'_p for each \mathbf{R}'_p .

5. We project each \mathbf{S}'_p to the corresponding tensor shape basis $\mathbf{B}'_{\text{shp},p,m}$ to get the coefficient vector

$$\mathbf{c}'_{\text{shp},p,m} = \mathbf{B}_{\text{shp},p,m}^{-T} \mathbf{S}'_p, p = 1, \dots, P. \quad (4.11)$$

6. We obtain the shape distance matrix

$$\mathbf{D}_{\text{shp},p}(m, n) = \|\mathbf{c}'_{\text{shp},p,m} - \mathbf{c}'_{\text{shp},p,n}\| \quad (4.12)$$

for \mathbf{S}'_p . By nearest neighboring method we find the index n_p with the smallest value of the shape distance for the p_{th} pose, and we identify \mathbf{S}'_p as class n_p .

7. After we get the recognition results n_p for all the 3D X-Ray Transform output images \mathbf{R}'_p ($p = 1, \dots, P$), we do majority voting among n_p to get the final recognition result for this testing DMI.

4.4 Experimental Results

The data for our experiment is the nematode DMI stacks, containing 5 classes and each class containing 10 samples (as the DMI stacks shown in Figure 4.5). Each image stack consists of more than 100 multi-focal frames obtained from a Differential Interference Contrast microscope [14], as introduced in Section 1.1.2. It is a type of light microscopy where contours are enhanced compared to ‘plain’ bright-field microscopy with little or no loss of resolution (unlike phase contrast) and it does not involve fluorescent excitation from laser sources (unlike confocal). All nematode images are registered to have the same mouth location. In the experiments, we just use one channel data of the original color images for feature extraction and classification, and the image intensity is normalized to remove the effect of illumination. For every class, 4 samples (DMI stacks) are used for training and the rest for testing. We select $(\pi/2, \pi/2)$, $(0, \pi/2)$, (π, π) , $(\pi/5, \pi/3)$ and $(\pi/3, \pi/5)$ as the 5 poses (angles) of the 3D X-Ray Transform, and the 3D X-Ray Transform output images at the first 3 poses of those training DMIs are shown in Figure 4.6. We can see that different angles capture very different characteristics of the image stacks. Since every DMI stack contains different numbers of images, their 3D X-Ray Transform output images are not the same size (even the ratios of image length to width are different). So we have to normalize them to be the same size for the X-Ray projections at the same pose.

4.4.1 Tensor Texture Analysis

In the tensor texture analysis, we compute the basis for each direction separately, because the images at different angles have different ratios of image length to width. The images at 3 poses, $(\pi/2, \pi/2)$, $(0, \pi/2)$ and (π, π) , for the training nematode DMI stacks are shown in Figure 4.6. The

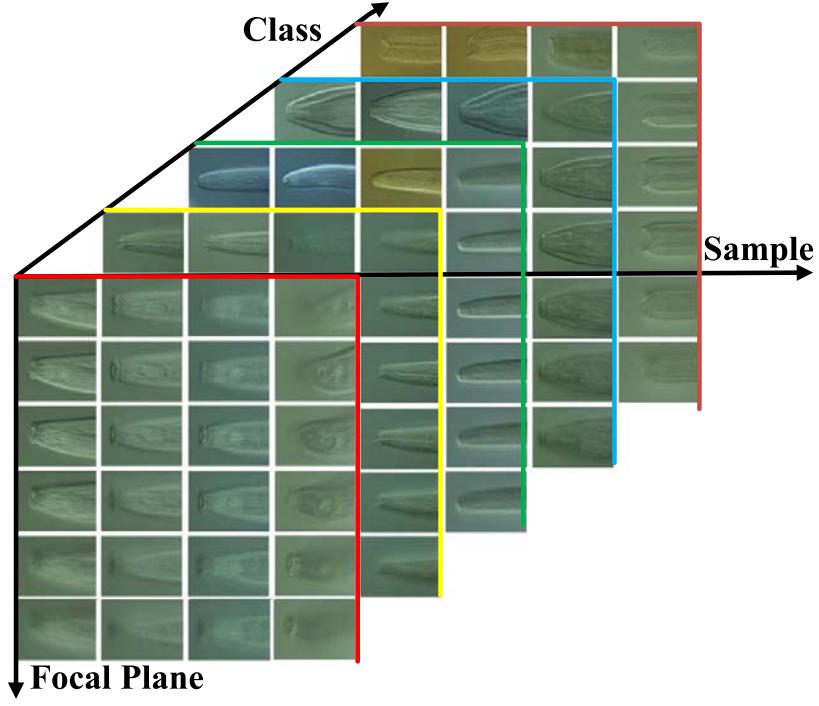


Figure 4.5: The nematode DMI volumetric stack database, including 5 classes of nematodes, and each class includes 10 samples (species). Each sample is represented by a DMI stack containing more than 100 focal planes (frames) taken from the top of the specimen to the bottom. Only 6 key frames from different focal planes of every stack are shown. They are 20 samples from each of the 5 classes (4 samples are picked for each class for illustration purposes).

training tensor data at those five poses are of sizes $5 \times 4 \times 97200$, $5 \times 4 \times 48600$, $5 \times 4 \times 64800$, $5 \times 4 \times 196245$ and $5 \times 4 \times 196245$, respectively.

In the tensor texture training step, for the X-Ray Transform output images $\mathbf{R}_{p,i}$ at the $p_{th} (p = 1, \dots, 5)$ pose, we get, from Equation (4.4), the core tensor $Z_{\text{tex},p}$, the 5×5 mode matrix $\mathbf{U}_{\text{tex},\text{class},p}$ spanning the class space (which is composed of the row vector $\mathbf{c}_{\text{tex},p,n}^T, n = 1, \dots, 5$), the 4×4 mode matrix $\mathbf{U}_{\text{tex},\text{sample},p}$ spanning the sample space, and the mode matrix $\mathbf{U}_{\text{tex},\text{RT},p}$ spanning the space of all training X-Ray Transform output images at the p_h pose. From Equation

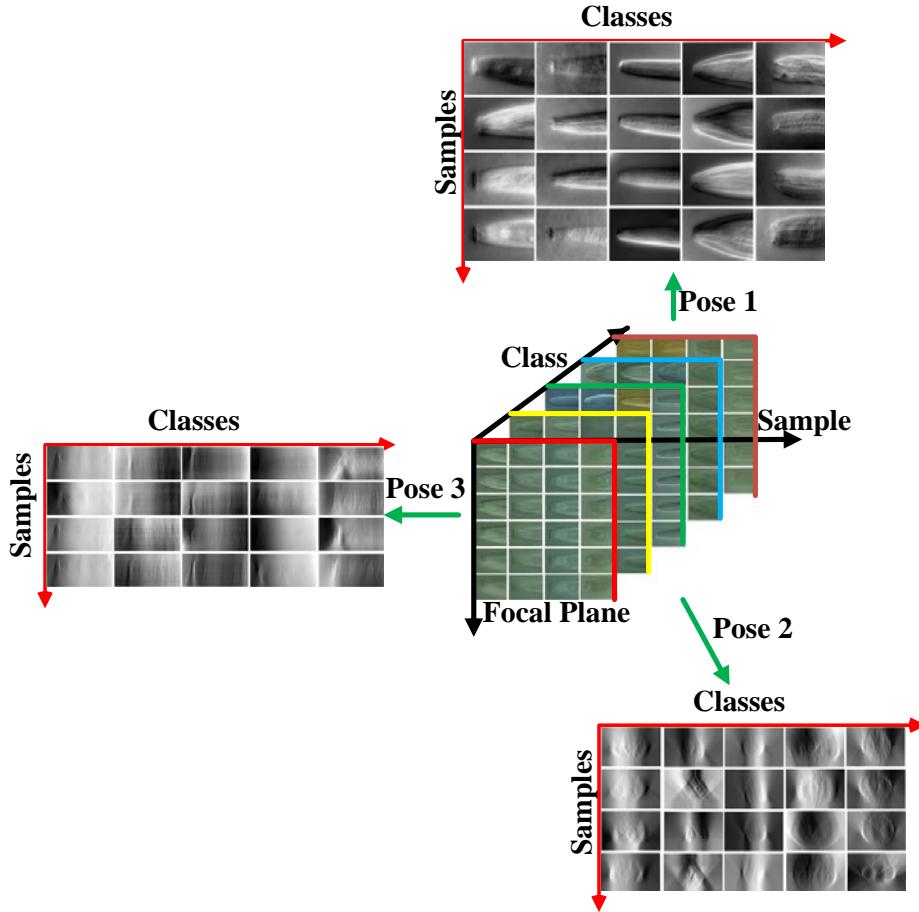


Figure 4.6: 3D X-Ray Transform projections at 3 poses, $(\pi/2, \pi/2)$, $(0, \pi/2)$ and (π, π) , for the training DMI stacks.

(5), we get the texture basis tensor $B_{\text{tex},p,m}$ for the m_{th} sample ($m = 1, \dots, 4$), which we flatten along the class mode to get the texture basis matrix $\mathbf{B}_{\text{tex},p,m}$. The visualization of those bases is shown in Figure 4.7.

Given a testing DMI stack, first we get its 3D X-Ray Transform output images \mathbf{R}'_p at different poses. Then we project them to the corresponding texture basis to get the coefficient vector and obtain the texture distance matrix $\mathbf{D}_{\text{tex},p}(m, n)$ ($m = 1, \dots, 4; n = 1, \dots, 5$). Through the

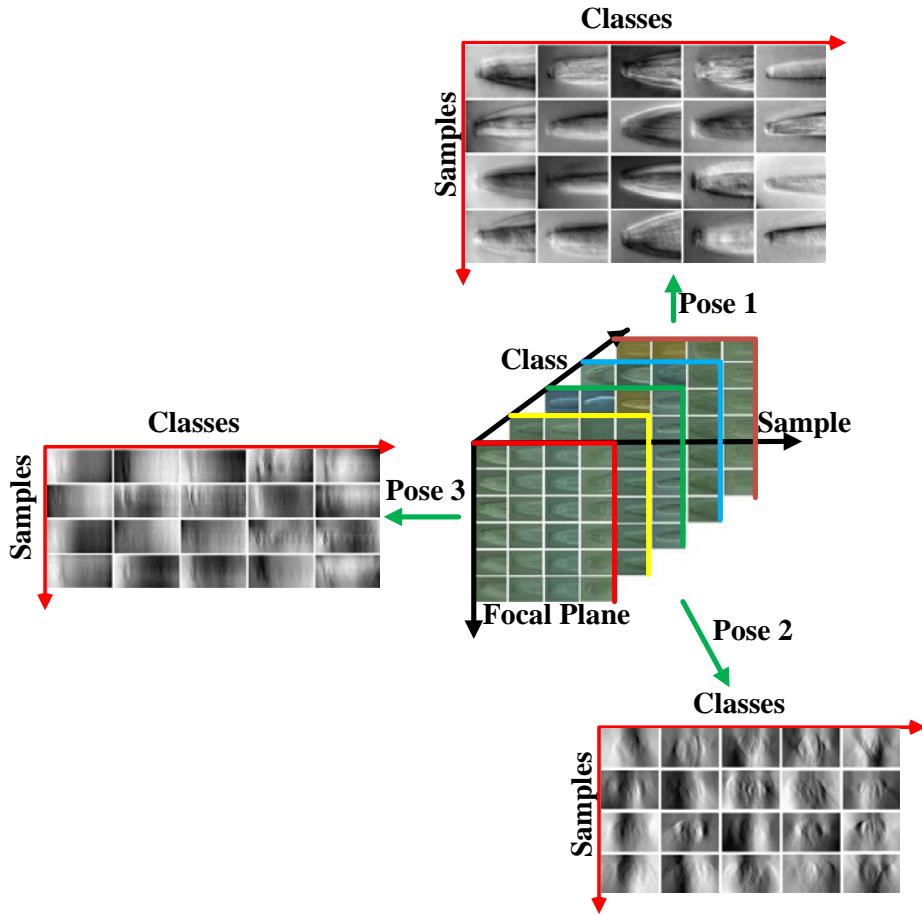


Figure 4.7: Tensor texture bases of the 3D X-Ray Transform projections at 3 poses, $(\pi/2, \pi/2)$, $(0, \pi/2)$ and (π, π) , for the training DMI stacks.

nearest neighboring criteria we find n_p that yields the smallest value of the texture distance for the p_{th} pose. Finally by majority voting among $n_p (p = 1, \dots, 5)$ we get the recognition result for this testing DMI stack. This 3D X-Ray Transform based multilinear texture analysis method obtains a recognition rate of 87%. The recognition performance of this multilinear texture analysis is shown in Figure 4.8.

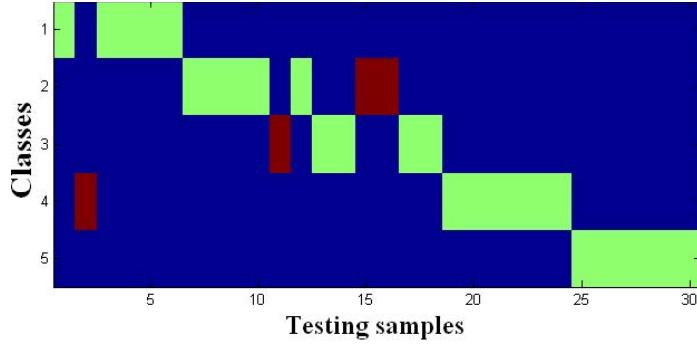


Figure 4.8: Multilinear texture recognition performance on the 3D X-Ray Transform output images of 30 nematode testing DMI sample stacks. In the vertical direction, numbers 1-5 denote the 5 nematode classes. In the horizontal direction, numbers 1-6 denote the 6 testing samples from class 1, 7-12 from class 2, 13-18 from class 3, 19-24 from class 4 and 25-30 from class 5. The green color denotes correct classification results, and the red color denotes wrong classification results (as viewed on a color monitor).

4.4.2 Tensor Shape Analysis

For the tensor shape analysis, not all 3D X-Ray Transform images have prominent shape information. From Figures 4.3-4.4 and Figure 4.6, we can see that 3D X-Ray Transform outputs at poses, $(\pi/2, \pi/2)$, (π, π) , $(\pi/5, \pi/3)$ and $(\pi/3, \pi/5)$, can be used to extract the shape information, while the images from the 2nd pose $(0, \pi/2)$, do not have much shape information. Some extracted shape contours at the pose $(\pi/2, \pi/2)$ are shown in Figure 4.9. Similar to the above description, we do the multilinear shape analysis and get the recognition performance shown in Figure 4.10.

4.4.3 Combination of Tensor Texture and Shape Feature

We combine the multilinear texture and shape analysis to build a more accurate classifier. Specifically, for a given DMI stack, through the 3D X-Ray Transform at the p_h pose, we get

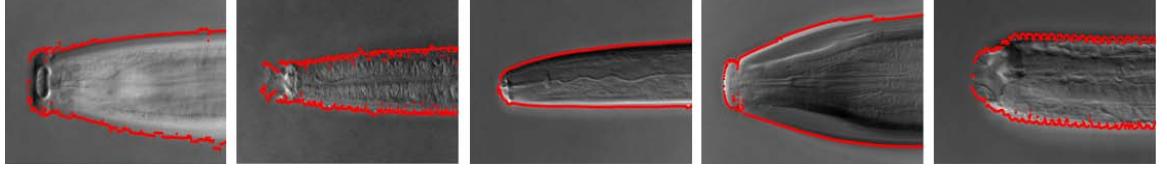


Figure 4.9: Extracted shape contours from the 3D X-Ray Transform output images at the pose $(\pi/2, \pi/2)$ for 5 DMI stacks (one DMI stack picked from each class).

$\mathbf{R}'_p (p = 1, \dots, 5)$ and $\mathbf{S}'_p (p = 1, 3, 4, 5)$. After projecting them to the corresponding tensor texture basis and shape basis, we obtain the texture distance matrix $\mathbf{D}_{\text{tex},p}(m, n) (m = 1, \dots, 4; n = 1, \dots, 5)$ for $\mathbf{R}'_p (p = 1, \dots, 5)$, and the shape distance matrix $\mathbf{D}_{\text{shp},p}(m, n)$ for $\mathbf{S}_p (p = 1, 3, 4, 5)$. When $p = 2$, $\mathbf{D}_{\text{shp},p}(m, n) = 1$, because we are not using the shape information at this pose.

We combine the shape and texture distance matrices as

$$\mathbf{D}_p(m, n) = \mathbf{D}_{\text{tex},p}(m, n) \times \mathbf{D}_{\text{shp},p}(m, n), \quad (4.13)$$

where \mathbf{D}_p is the combined distance matrix for the 3D X-Ray Transform output image \mathbf{R}'_p at the p_{th} pose. Then we identify \mathbf{R}'_p as class n_p through the nearest neighbor method for all poses, and after the majority voting among $n_p (p = 1, \dots, 5)$ we finally get the recognition result for this given DMI stack.

The combination of the tensor texture and shape information improves the recognition rate, which is shown in the recognition performance comparison table (Table 4.1) between different classification methods. The training data for the classifiers using PCA and Independent Components Analysis (ICA) [74] is 20 key frames, coming from those 20 DMI stacks used for the MXRT training (one DMI stack, one key frame), and the testing data is 30 key frames from the 30 testing

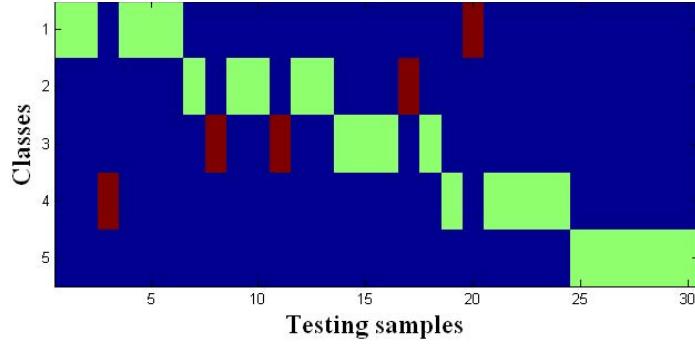


Figure 4.10: Multilinear shape recognition performance on the 3D X-Ray Transform output images of nematode DMI stacks of 30 testing samples. In the vertical direction, numbers 1-5 denote the 5 class labels. In the horizontal direction, numbers 1-6 denotes the 6 testing samples from class 1, 7-12 from class 2, 13-18 from class 3, 19-24 from class 4 and 25-30 from class 5. The green color denotes correct results, and the red color denotes wrong results.

DMI stacks for the MXRT testing. The classification based on Gabor texture is also done for those 50 key frames (from 50 DMI stacks) [26]. The classification performance of those 3 methods is also listed in Table 4.1.

We have done the classification by changing the training data and testing data, and got similar recognition rate. Also, we increased the amount of the training data, and got slightly better recognition results. These results are summarized in Table 4.1.

4.4.4 Tensor Analysis on Key Frames

Given stacks of multifocal image frames, an alternative and intuitive way for classification is to do the tensor analysis on the original image frames. So here we give a short description of the multilinear analysis on the key image frames (i.e., some selected focal planes) from every DMI stack, and the recognition performance is also listed in Table 4.1.

Classifier	Recognition Rate
PCA	60%
ICA	57%
Gabor Texture	60%
Tensor Texture	87%
Tensor Shape	83%
Key Frames Tensor	73%
Tensor Texture and Shape 1	94%
Tensor Texture and Shape 2	97%
Tensor Texture and Shape 3	97%

Table 4.1: Recognition rate comparison between different classification methods. The 4th-7th rows are the recognition rates on the training data composed of 4 samples from each class, the 8th row is the recognition rate on the training data of another 4 samples from each class, while the 9th row is the recognition rate on the training data of 5 samples from each class. The training and testing data for the first 3 rows are described in the text.

In the training step, we pick the same number of the most typical frames as the key frames from the same focal plane (for example, 6 frames for each stack), as in Figure 4.5. Then the training data is a $5 \times 4 \times 6 \times 97200$ tensor. After the multilinear SVD analysis as introduced in the previous section [69], we get

$$D = Z \times_1 \mathbf{U}_{\text{class}} \times_2 \mathbf{U}_{\text{sample}} \times_3 \mathbf{U}_{\text{frame}} \times_4 \mathbf{U}_{\text{pixel}}, \quad (4.14)$$

with the 5×5 mode matrix $\mathbf{U}_{\text{class}}$ spanning the class space (which is composed of the row vector $\mathbf{c}_{\text{tex},n}^T, n = 1, \dots, 5$), the 4×4 mode matrix $\mathbf{U}_{\text{sample}}$ spanning the sample space, the 6×6 mode matrix $\mathbf{U}_{\text{frame}}$ spanning the focal plane space, and the mode matrix $\mathbf{U}_{\text{pixel}}$ orthogonally spanning the space of all training key frames.

The basis for different classes of this training tensor is

$$B = Z \times_2 \mathbf{U}_{\text{sample}} \times_3 \mathbf{U}_{\text{frame}} \times_4 \mathbf{U}_{\text{pixel}}. \quad (4.15)$$

Then each class $n (n = 1, \dots, 5)$ can be represented by a coefficient vector of size 5, \mathbf{c}_n , which is the row vector from the class matrix $\mathbf{U}_{\text{class}}$. For a certain sample m and a certain frame f , we can get the basis tensor $B_{m,f} (m = 1, \dots, 4; f = 1, \dots, 6)$ of size $5 \times 1 \times 1 \times 97200$. We flatten $B_{m,f}$ along the class mode to get the basis matrix $\mathbf{B}_{m,f}$.

In the recognition process, given a testing image \mathbf{T} , we project it onto those bases and get the coefficients for every (m, f) pair, i.e., $\mathbf{c}_{m,f} = \mathbf{B}_{m,f}^{-T} \mathbf{T}$. Then we obtain the 3rd order distance matrix $\mathbf{D}(m, f, n) = \|\mathbf{c}_{m,f} - \mathbf{c}_n\|$, and by the nearest neighboring method, we find the index n that yields the smallest value of the distance matrix, and we identify this testing image \mathbf{T} as class n .

4.5 Conclusion

In this Chapter, we presented a 3D X-Ray Transform based multilinear feature extraction and classification method for Digital Multi-focal Images. The 3D structure information of a multifocal image stack are captured through the 3D X-Ray Transform at different angle views. We embedded the 3D X-Ray Transform within a multilinear framework to analyze multiple factors - shape, texture, viewpoint, different instances within the same class and different classes of specimens. The experimental results on the nematode DMI data show that the proposed 3D X-Ray Transform based multilinear analysis method can effectively improve the recognition rate from 60% (PCA) to 97%.

Chapter 5

Conclusion and Future Work

5.1 Conclusions

In this work, we have proposed methods on how to extract features in two different kinds of volumetric bioimage stacks, plant shoot apical meristems cell images taken by Confocal Laser Scanning Microscopy and nematode images taken by Differential Interference Contrast Microscopy.

We developed a useful image processing and analysis pipeline for cell segmentation and cell lineage construction in time-lapse volumetric image stacks obtained using CLSM. It is able to segment and find the correspondence of cells in densely packed tissue. After the segmentation by the watershed method or level-set method, we use our proposed local graph matching to track the cells both spatially and temporally, and identify cell divisions at the same time. The geometric structure and topology of the relative positions of the cells are efficiently exploited to match them. Furthermore, we built a joint segmentation and tracking system, where the tracking output acts as an indicator of the quality of segmentation and, in turn, the optimized segmentation can be improved to

obtain better tracking results. Finally, the spatial and temporal correspondences are fused together to obtain a single unified track. We integrate the tracks from multiple slices at a given time instant with that of tracks obtained across time periods, in order to maintain the consistency of the cell lineages for long time periods. We tested this image analysis pipeline on multiple datasets, and it significantly improved both segmentation and tracking when compared to an open-loop framework in which segmentation and tracking modules operate separately.

We also presented a multilinear feature extraction and classification method for nematode DMI image stacks. Our proposed method can effectively exploit the entire information in the stack using the 3D X-Ray projections at different viewing angles. These DMI stacks represent the effect of different factors - shape, texture, viewpoint, different instances within the same class and different classes of specimens. For this purpose, we embed the 3D X-Ray Transform within a multilinear framework and propose a Multilinear X-Ray Transform (MXRT) feature representation. By combining the tensor texture and shape information we can get better recognition rates than just relying on the original or key frames of DMI stacks. The experimental results on the nematode DMI data show that the 3D X-Ray Transform based multilinear feature extraction and analysis method can effectively give very reliable recognition rate on a real-life database.

5.2 Future Work

One of the goals of the future work will be to make the plant cell image analysis pipeline more robust to imaging noise. The quality of the images can be improved by studying on the physical characteristic of the microscope and the imaging process [75]. Second, the algorithm's capability to track long-term cell lineage could be improved by developing an error correction module.

Third, there is a need to generate more cell lineages on multiple SAMs datasets, and based on those estimated lineages develop robust 3D reconstruction methods for the SAMs multilayered structure [38, 65]. Furthermore, how to build a dynamical model about the local spatio-temporal coordination of cell growth and cell division for the plant tissues would be another goal of the future work [66]. Finally, the classification method need to be evaluated on larger and more complicated datasets.

Bibliography

- [1] B. S. Manjunath, B. Sumengen, Z. Bi, J. Byun, M. El-Saban, D. Fedorov, and N. Vu, “Towards automated bioimage analysis: from features to semantics,” in *IEEE International Symposium on Biomedical Imaging: Nano to Macro*, pp. 255 –258, 2006.
- [2] H. Peng, “Bioimage informatics: a new area of engineering biology.,” *Bioinformatics*, vol. 24, no. 17, pp. 1827–1836, 2008.
- [3] K. Kvilekval, D. Fedorov, B. Obara, A. Singh, and B. S. Manjunath, “Bisque: A platform for bioimage analysis and management,” *Bioinformatics*, vol. 26, no. 4, pp. 544–552, 2010.
- [4] L. Pedro Coelho, E. Glory-afshar, J. Kangas, S. Quinn, A. Shariff, and R. Murphy, “Principles of bioimage informatics: Focus on machine learning of cell patterns,” in *Lecture Notes in Computer Science*, 2010.
- [5] D. B. Murphy, *Fundamentals of light microscopy and electronic imaging*. Wiley-Liss, 2001.
- [6] J. B. Pawley, ed., *Handbook of Biological Confocal Microscopy*. Springer, 3rd ed., 2006.
- [7] J. J. Bozzola and L. D. Russell, *Electron microscopy: principles and techniques for biologists*. Jones Bartlett Learning, 1999.
- [8] R. Y. Tsien, “Imagining imaging’s future,” *Nature Reviews Molecular Cell Biology*, 2003.
- [9] M. Liu, H. Peng, A. K. Roy-Chowdhury, and E. Myers, “3d neuron tip detection in volumetric microscopy images,” in *IEEE International Conference on Bioinformatics and Biomedicine*, pp. 366 –371, 2011.
- [10] M. Liu, R. K. Yadav, A. Roy-Chowdhury, and G. V. Reddy, “Automated tracking of stem cell lineages of arabidopsis shoot apex using local graph matching,” *Plant Journal*, vol. 62, pp. 135–147, 2010.
- [11] M. Liu and A. K. Roy-Chowdhury, “Multilinear feature extraction and classification of multi-focal images, with applications in nematode taxonomy,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2823 –2830, 2010.

- [12] E. M. Meyerowitz, “Genetic control of cell division patterns in developing plants.,” *Cell*, vol. 88, no. 3, pp. 299–308, 1997.
- [13] M. K. Barton, “Twenty years on: The inner workings of the shoot apical meristem, a developmental dynamo,” *Developmental Biology*, vol. 341, no. 1, pp. 95 – 113, 2010.
- [14] R. D. Allen, G. B. David, and G. Nomarski, “The Zeiss-Nomarski differential interference equipment for transmitted light microscopy,” *Z. Wiss. Mikrosk. Mikrosk. Tech.*, vol. 69, pp. 193–221, 1969.
- [15] S. R. Cutler, D. W. Ehrhardt, J. S. Griffitts, and C. R. Somerville, “Random gfp::cdna fusions enable visualization of subcellular structures in cells of arabidopsis at a high frequency,” *Proceedings of the National Academy of Sciences*, vol. 97, pp. 3718–3723, 2000.
- [16] G. V. Reddy, M. G. Heisler, D. W. Ehrhardt, and E. M. Meyerowitz, “Real-time lineage analysis reveals oriented cell divisions associated with morphogenesis at the shoot apex of arabidopsis thaliana,” *Development*, 2004.
- [17] G. V. Reddy, “Live-imaging stem-cell homeostasis in the arabidopsis shoot apex,” *Current Opinion in Plant Biology*, vol. 11, no. 1, pp. 88 – 93, 2008.
- [18] P. De Ley and W. Bert, “Video capture and editing as a tool for the storage, distribution, and illustration of morphological characters of nematodes,” *Journal of Nematology*, vol. 34, pp. 296–302, 2002.
- [19] Eyualem-Abebe, J. G. Baldwin, B. Adams, D. Hope, S. Gardner, R. Huettel, T. Mullin, P. and Powers, J. Sharma, W. Ye, and W. K. Thomas, “A position paper on the electronic publication of nematode taxonomic manuscripts,” *Journal of Nematology*, vol. 38, no. 3, pp. 305–311, 2006.
- [20] M. Yoder, “Vouchering 3d morphology of challenging organisms using high-definition multi-focal images and their practical applications,” *American Entomologist*, vol. 54, no. 4, pp. 225–228, 2008.
- [21] L. F. Hernandez, A. Havelange, G. Bernier, and P. B. Green, “Growth behavior of single epidermal cells during flower formation: sequential scanning electron micrographs provide kinematic patterns for anagallis,” *Planta*, vol. 185, pp. 139–147, 1991.
- [22] J. Dumais and D. Kwiatkowska, “Analysis of surface growth in shoot apices,” *The Plant Journal*, vol. 31, no. 2, pp. 229–241, 2001.
- [23] D. Kwiatkowska and J. Dumais, “Growth and morphogenesis at the vegetative shoot apex of anagallis arvensis l.,” *Journal of Experimental Botany*, vol. 54, no. 387, pp. 1585–1595, 2003.
- [24] D. Kwiatkowska, “Surface growth at the reproductive shoot apex of arabidopsis thaliana pin-formed 1 and wild type,” *Journal of Experimental Botany*, vol. 55, no. 399, pp. 1021–1032, 2004.

- [25] K. Li, E. D. Miller, L. E. Weiss, P. G. Campbell, and T. Kanade, “Online tracking of migrating and proliferating cells imaged with phase-contrast microscopy,” in *IEEE Computer Society Workshop on Mathematical Methods in Biomedical Image Analysis*, p. 65, 2006.
- [26] B. S. Manjunath and W. Y. Ma, “Texture features for browsing and retrieval of image data,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837–842, 1996.
- [27] A. Veeraraghavan, A. K. Roy-Chowdhury, and R. Chellappa, “Matching shape sequences in video with applications in human movement analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1896–1909, 2005.
- [28] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, “Face recognition: A literature survey,” *ACM Computing Surveys*, vol. 35, no. 4, pp. 399–458, 2003.
- [29] T. A. Steeves and I. M. Sussex, *Patterns in plant development*. Cambridge University Press, 1989.
- [30] R. F. Lyndon, *The shoot apical meristem: its growth and development*. Cambridge University Press, 1998.
- [31] P. Laufs, “Cellular parameters of the shoot apical meristem in arabidopsis,” *Plant Cell*, vol. 10, pp. 1375–1389, 1998.
- [32] O. Grandjean, T. VERNOUX, P. Laufs, K. Belcram, Y. Mizukami, and J. Traas, “In vivo analysis of cell division, cell growth, and differentiation at the shoot apical meristem in arabidopsis,” *Plant Cell*, vol. 16, no. 1, pp. 74–87, 2003.
- [33] O. Debeir, P. Van Ham, R. Kiss, and C. Decaestecker, “Tracking of migrating cells under phase-contrast video microscopy with combined mean-shift processes,” *IEEE Transactions on Medical Imaging*, vol. 24, no. 6, pp. 697 –711, 2005.
- [34] O. Dzyubachyk, W. Niessen, and E. Meijering, “A variational model for level-set based cell tracking in time-lapse fluorescence microscopy images,” in *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 97 –100, 2007.
- [35] C. Zimmer, E. Labruyere, V. Meas-Yedid, N. Guillen, and J. C. Olivo-Marin, “Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: a tool for cell-based drug testing,” *IEEE Transactions on Medical Imaging*, vol. 21, no. 10, pp. 1212 –1221, 2002.
- [36] O. M. Tataw, M. Liu, A. Roy-Chowdhury, R. K. Yadav, and G. V. Reddy, “Pattern analysis of stem cell growth dynamics in the shoot apex of arabidopsis,” in *IEEE International Conference on Image Processing*, pp. 3617 –3620, 2010.
- [37] A. Chakraborty, R. Yadav, M. Liu, M. Tataw, K. Mkrtchyan, A. Roy-Chowdhury, and G. V. Reddy, “Computational tools for quantitative analysis of cell growth patterns and morphogenesis in actively developing plant stem cell niches,” in *Plant Signalling Networks*, vol. 876, pp. 305–316, Humana Press, 2012.

- [38] A. Chakraborty, M. Liu, K. Mkrtchyan, G. V. Reddy, and A. Roy-Chowdhury, “Cell volume estimation from a sparse collection of noisy confocal image slices,” in *Indian Conference on Computer Vision, Graphics and Image Processing*, pp. 183–189, 2010.
- [39] A. L. Cunha, A. H. K. Roeder, and E. M. Meyerowitz, “Segmenting the sepal and shoot apical meristem of arabidopsis thaliana,” in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 5338 –5342, 2010.
- [40] T. F. Chan and L. A. Vese, “Active contours without edges,” *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.
- [41] Z. Yin, R. Bise, M. Chen, and T. Kanade, “Cell segmentation in microscopy imagery using a bag of local bayesian classifiers,” in *The IEEE International Symposium on Biomedical Imaging*, 2010.
- [42] V. Gor, M. Elowitz, T. Bacarian, and E. Mjolsness, “Tracking cell signals in fluorescent images,” in *IEEE Workshop on Computer Vision Methods for Bioinformatics*, pp. 142–149, 2005.
- [43] H. Peng, Z. Ruan, F. Long, J. H. Simpson, and E. W. Myers, “V3d enables real-time 3d visualization and quantitative analysis of large-scale biological image data sets,” *Nature Biotechnology*, vol. 28, pp. 348–353, 2010.
- [44] K. Mkrtchyan, D. Singh, M. Liu, G. Reddy, A. Roy-Chowdhury, and M. Gopi, “Efficient cell segmentation and tracking of developing plant meristem,” *IEEE International Conference on Image Processing*, 2011.
- [45] M. Liu, A. Chakraborty, D. Singh, M. Gopi, R. Yadav, G. V. Reddy, and A. Roy-Chowdhury, “Adaptive cell segmentation and tracking for volumetric confocal microscopy images of a developing plant meristem,” *Molecular Plant*, vol. 4, no. 5, pp. 922–931, 2011.
- [46] N. Ray and S. Acton, “Active contours for cell tracking,” in *IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 7–9, 2002.
- [47] D. P. Mukherjee, N. Ray, and S. T. Acton, “Level set analysis for leukocyte detection and tracking,” *IEEE Transactions on Image Processing*, vol. 13, pp. 562–572, 2004.
- [48] A. Dufour, V. Shinin, S. Tajbakhsh, N. Guillen-Aghion, J. C. Olivo-Marin, and C. Zimmer, “Segmenting and tracking fluorescent cells in dynamic 3-d microscopy with coupled active surfaces,” *IEEE Transactions on Image Processing*, vol. 14, no. 9, pp. 1396 –1410, 2005.
- [49] A. Rangarajan, H. Chui, and F. L. Bookstein, “The softassign procrustes matching algorithm,” in *Information Processing in Medical Imaging*, pp. 29–42, Springer, 1997.
- [50] S. Gold and A. Rangarajan, “A graduated assignment algorithm for graph matching,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 4, pp. 377–388, 1996.
- [51] E. Fazl-Ersi, J. S. Zelek, and J. K. Tsotsos, “Robust face recognition through local graph matching,” *Journal Of Multimedia*, vol. 2, no. 5, pp. 31–37, 2007.

- [52] P. B. Reuille, I. Bohn-Courseau, C. Godin, and J. Traas, “A protocol to analyse cellular dynamics during plant development,” *The Plant Journal*, vol. 44, pp. 1045–1053, 2005.
- [53] G. V. Reddy and A. Roy-Chowdhury, “Live-imaging and image processing of shoot apical meristems of arabidopsis thaliana,” in *Plant Systems Biology*, vol. 553, pp. 305–316, Humana Press, 2009.
- [54] P. Viola and I. Wells, W. M., “Alignment by maximization of mutual information,” *International Conference on Computer Vision*, pp. 16–23, 1995.
- [55] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens, “Multimodality image registration by maximization of mutual information,” *IEEE transactions on Medical Imaging*, vol. 16, pp. 187–198, 1997.
- [56] L. Najman and M. Schmitt, “Watershed of a continuous function,” *Signal Processing*, vol. 38, no. 1, pp. 99 – 112, 1994.
- [57] S. K. Nath, K. Palaniappan, and F. Bunyak, “Cell segmentation using coupled level sets and graph-vertex coloring,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention - Volume Part I*, pp. 101–108, 2006.
- [58] A. Gelas, K. Mosaliganti, A. Gouaillard, L. Souhait, R. Noche, N. Obholzer, and S. G. Megason, “Variational level-set with gaussian shape model for cell segmentation,” in *IEEE International Conference on Image processing*, pp. 1085–1088, 2009.
- [59] 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*, vol. 13, no. 1, pp. 143 – 155, 2009.
- [60] S. Osher and J. A. Sethian, “Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations,” *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, 1988.
- [61] S. Beucher and C. Lantuejoul, “Use of Watersheds in Contour Detection,” in *International Workshop on Image Processing: Real-time Edge and Motion Detection/Estimation, Rennes, France.*, 1979.
- [62] M. Marcuzzo, P. Quelhas, A. Campilho, and A. M. Mendonca, “Automatic cell segmentation from confocal microscopy images of the arabidopsis root,” in *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 712 –715, 2008.
- [63] P. Soille, *Morphological Image Analysis: Principles and Applications*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2 ed., 2003.
- [64] J. D. Callos, M. DiRado, B. Xu, F. J. Behringer, B. M. Link, and J. I. Medford, “The forever young gene encodes an oxidoreductase required for proper development of the arabidopsis vegetative shoot apex,” *The Plant Journal*, vol. 6, no. 6, pp. 835–847, 1994.

- [65] A. Chakraborty, R. K. Yadav, G. V. Reddy, and A. K. Roy-Chowdhury, “Cell resolution 3d reconstruction of developing multilayer tissues from sparsely sampled volumetric microscopy images,” in *IEEE International Conference on Bioinformatics and Biomedicine*, pp. 378–383, 2011.
- [66] R. K. Yadav, T. Girke, S. Pasala, M. Xie, and V. Reddy, “Gene expression map of the Arabidopsis shoot apical meristem stem cell niche,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 12, 2009.
- [67] I. Laptev and T. Lindeberg, “Space-time interest points,” in *IEEE International Conference on Computer Vision*, vol. 1, pp. 432–439, 2003.
- [68] L. De Lathauwer, B. De Moor, and J. Vandewalle, “A multilinear singular value decomposition,” *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.
- [69] M. Alex, O. Vasilescu, and D. Terzopoulos, “Multilinear subspace analysis of image ensembles,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 93–99, 2003.
- [70] F. Natterer, *The Mathematics of Computerized Tomography*. Wiley, 1986.
- [71] S. Deans, *The Radon Transform and Some of Its Applications*. Krieger Publishing Company, 1983.
- [72] P. G. Challenor, P. Cipollini, and D. Cromwell, “Use of the 3D Radon Transform to Examine the Properties of Oceanic Rossby Waves,” *Journal of Atmospheric and Oceanic Technology*, vol. 18, no. 9, pp. 1558–1566, 2001.
- [73] I. L. Dryden and K. V. Mardia, *Directional Statistics and Shape Analysis*. John Wiley and Sons, 1998.
- [74] P. Comon, “Independent component analysis, a new concept?,” *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994.
- [75] Z. Yin, D. F. E. Ker, and T. Kanade, “Restoring dic microscopy images from multiple shear directions,” in *IEEE International Symposium on Biomedical Imaging*, 2011.
- [76] J. Xie, T. Zhao, T. Lee, E. W. Myers, and H. Peng, “Automatic neuron tracing in volumetric microscopy images with anisotropic path searching,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 472–479, 2010.
- [77] H. Chui, “A new algorithm for non-rigid point matching,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 44–51, 2000.
- [78] L. Wiskott, J. M. Fellous, N. Krger, and M. von der Malsburg, “Face recognition by elastic bunch graph matching,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 775–779, 1997.

- [79] K. Li and T. Kanade, “Cell population tracking and lineage construction using multiple-model dynamics filters and spatiotemporal optimization,” in *International Workshop on Microscopic Image Analysis with Applications in Biology*, 2007.
- [80] Y. Zheng and D. Doermann, “Robust point matching for nonrigid shapes by preserving local neighborhood structures,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, p. 643, 2006.
- [81] F. Li and P. Perona, “A bayesian hierarchical model for learning natural scene categories,” in *IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 524–531, 2005.
- [82] M. Alex, O. Vasilescu, and D. Terzopoulos, “Multilinear analysis of image ensembles: Tensorfaces,” in *European Conference on Computer Vision-Part I*, pp. 447–460, 2002.
- [83] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, “Multilinear principal component analysis of tensor objects for recognition,” in *International Conference on Pattern Recognition*, vol. 2, pp. 776–779, 2006.
- [84] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H. J. Zhang, “Multilinear discriminant analysis for face recognition,” *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 212–220, 2007.
- [85] O. Vasilescu, M. Alex, and D. Terzopoulos, “Tensortextures: multilinear image-based rendering,” *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 336–342, 2004.
- [86] M. Alex, O. Vasilescu, and D. Terzopoulos, “Multilinear image analysis for facial recognition,” in *International Conference on Pattern Recognition*, vol. 2, pp. 511–514, 2002.