

Mapping Developmental Mechanics in *Drosophila melanogaster*

with Segmentation and Force Inference

By

David N. Mashburn

Dissertation

Submitted to the Faculty of the
Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Physics

May, 2015

Nashville, Tennessee

Approved:

M. Shane Hutson, Ph.D.

Thomas W. Kephart, Ph.D.

Charles F. Maguire, Ph.D.

Norman H. Tolks, Ph.D.

John P. Wikswo, Ph.D.

Copyright © 2015 by David Mashburn
All Rights Reserved

Dedicated to Whitney, my best friend and the love of my life,
and to the good Lord above without whom this work
(and the flies) would never have been possible.

ACKNOWLEDGEMENTS

I want to thank everyone that has made this work possible: my adviser Shane Hutson, my committee, all the great administrative staff in the physics department and the graduate school that keep everything humming smoothly, the NSF and NIH for providing my research funding, my fellow graduate students (particularly Sarah Crews and Ty McCleery), and our collaborators Jim Veldhuis and Wayne Brodland from the University of Waterloo.

I especially want to thank my family for their constant encouragement, help, and support: my wife Whitney and her parents Kent and Ellen Blocker.

Prologue

O, thou great pestilence of the fruit bowl. That winged insect that torments both scientist and layperson alike. Might the solution to all sort of perplexing medical mysteries lie buried inside that translucent membrane of vitteline? Are not your deepest secrets the stuff of Nobel Laureates? Nay, though the lance and harpoon be too lofty to smite thee and thy great enigmas, might the precise spot of a well focused laser, reveal them?

TABLE OF CONTENTS

	Page
DEDICATION.....	iii
ACKNOWLEDGEMENTS.....	iv
PROLOGUE.....	v
LIST OF TABLES.....	x
LIST OF FIGURES.....	xi
Chapter 1.....	1
1. Introduction.....	1
Chapter 2.....	5
2. General Background.....	5
2.1. Introduction.....	5
2.2. Drosophila structure and development.....	7
2.3. Microscopic mechanical models.....	12
2.4. Image processing and segmentation.....	15
2.4.1. Image processing.....	15
2.4.2. Watershed segmentation.....	16
2.5. Macroscopic tissue models.....	18
2.5.1. Foam Model.....	18
2.5.2. Finite Element Simulations.....	18
2.6. Force Inference with Video Force Microscopy.....	20
Chapter 3.....	21
3. A Review of State-of-the-Art in vivo Force Measurement Techniques.....	21
3.1. Introduction.....	21
3.2. Measuring Forces.....	22

3.3. Direct Contact Techniques.....	24
3.4. Magnetic Tweezers.....	25
3.5. Optical Tweezers.....	27
3.6. Laser ablation.....	28
3.7. Traction Force Microscopy and Bed-of-Nails.....	31
3.8. Embedded deformable objects.....	31
3.9. Force Inference.....	34
3.10. Conclusions.....	35
Chapter 4.....	37
4. Enabling user-guided segmentation and tracking of surface-labeled cells in time-lapse image sets of living tissues.....	37
4.1. Introduction.....	37
4.2. Materials and Methods.....	39
4.2.1. Sample Preparation and Imaging.....	39
4.2.2. Segmentation Algorithms.....	39
4.2.3. Technical Implementation.....	41
4.3. Results.....	42
4.3.1. Tracking.....	47
4.3.2. Robustness to Noise.....	49
4.3.3. Speed.....	49
4.3.4. Segmentation Quality.....	52
4.3.5. Measuring Cell Shape Oscillations.....	54
4.4. Discussion.....	56
Chapter 5.....	59
5. Interactive 4D (xyzt) segmentation of epithelial cells in time-lapse confocal image stacks.....	59

5.1. Introduction.....	59
5.2. Implementation Details.....	61
5.3. Algorithms and Mathematics.....	69
5.3.1. N-Dimensional Line and Triangle Rasterization.....	69
5.3.1.1. ND Bresenham Algorithm.....	69
5.3.1.2. ND Bresenham Triangle Algorithm.....	70
5.3.1.2.1. Relative Dimension of Steepest Slope.....	71
5.3.1.2.2. Computing RDSS values.....	72
5.3.1.2.3. Final Algorithm.....	73
5.3.2. Undo Implementation.....	74
5.3.3. ND Sparse Array Implementation.....	74
5.4. Software Package Information.....	76
5.5. Comparisons with other techniques.....	77
5.6. Advice on 2D vs. 3D segmentation.....	79
5.7. Conclusion.....	79
Chapter 6.....	81
6. Description of wound dynamics through segmentation in 2D and 3D.....	81
6.1. Introduction.....	81
6.2. Materials and Methods.....	82
6.2.1. Laser Microsurgery.....	83
6.2.2. Image Analysis.....	83
6.3. Results.....	83
6.4. Conclusions.....	90
Chapter 7.....	92
7. Errors in Force Inference Techniques.....	92

7.1. Introduction.....	92
7.2. Results and Discussion.....	98
Chapter 8.....	108
8. Conclusions and Future Directions.....	108
Supplement 1.....	113
S1. Three-dimensional shape changes associated with apical contraction cycles.....	113
Supplement 2.....	119
S2. Spring Network Models of Embryonic Tissue.....	119
S2.1. Epithelial elastic sheet model.....	119
S2.2. Derivation of Relative Displacements After Hole Drilling.....	119
S2.3. Results of the Warping Algorithm.....	123
S2.4. One Dimensional Model.....	124
S2.5. Radial spring networks and nonlinear behavior.....	129
S2.5.1. Effective Spring Constants.....	131
S2.5.2. Model Construction.....	132
Supplement 3.....	139
S3. Segmenting a wound in 3D+time.....	139
REFERENCES.....	143

LIST OF TABLES

Table	Page
4.1: Speed and accuracy of various segmentation methods.....	54

LIST OF FIGURES

Figure	Page
2.1: Stages of fly embryogenesis.....	8
2.2: Germ band retraction, lateral view.....	10
2.3: Maxwell Element and an example creep response graph (displacement vs. time) for a given time varying (step-wise) external force (bottom left).....	13
2.4: Kelvin-Voight Element and an example creep response graph (displacement vs. time) for a given time varying (step-wise) external force (bottom left).....	14
2.5: Basic idea behind watershed segmentation.....	17
2.6: Summary of components in the the Cellular Finite Element Model from Brodland et al.....	19
4.1: Common segmentation difficulties in confocal images of living, cadherin-GFP stained fruit fly embryos.....	43
4.2: Adding and deleting seeds manually.....	45
4.3: Adding multiple extra seeds to correct mis-segmentation of cellular subregions.....	46
4.4: Cell tracking when the frame-to-frame movements of cells are large.....	48
4.5: Comparison of segmentation speed and accuracy for a typical data set: 190 frames with an average of 64 cells per frame.....	51
4.6: Time-and-space pair correlation function of triple-junction velocities for a data set with 66 segmented cells.....	56
5.1: Core user interface of SWS4D.....	62
5.2: Sample Lines and Triangles using the ND Bresenham algorithms.....	64
5.3: Bresenham triangles can be patterned into any curved strip.....	65
5.4: Full SWS4D user interface.....	67
5.5: Result of applying a mask to a segmentation.....	69
6.1: Normalized areas and mean radial aspect ratios vs. time for the wounded region and successive rings of nearest-neighboring cells.....	85
6.2: Normalized lengths of wound perimeter, radial contours, and first nearest-neighbor	

outer perimeter.....	87
6.3: Mean-normalized volume, area, and average thickness of cells from the Resille-117 embryo.....	88
6.4: Wounded region and neighboring cells' occupation at different depths and times.....	90
7.1: Overview of Video Force Microscopy.....	94
7.2: Effect of Noise on VFM.....	100
7.3: Standard Errors reported by VFM for the Noise Study.....	103
7.4: Effect of Rasterization on VFM.....	105
7.5: Poorly Constrained Modes of the Solution.....	107
8.1: Registering the motion of a cell-cell boundary from one frame to another.....	111
S1.1: Three-dimensional dynamic changes in amnioserosa cell shape.....	114
S1.2: Correlations of volumetric measurements for seven individual amnioserosa cells.	116
S2.1: Warps for Cell Edge Cut and Cell Center Cut.....	123
S2.2: Spring Diagram.....	125
S2.3: Force v. Displacement Curve for linear and nonlinear springs.....	126
S2.4: Equilibrium Position v Nonlinear Constant.....	126
S2.5: Connection to 2D Experiments.....	127
S2.6: Simple Model Cannot Distinguish Between Competing Factors.....	129
S2.7: Polar Decomposition of 2D Sheet.....	130
S2.8: Linearized Spring Network.....	131
S2.9: Radial Spring Network Solutions.....	134
S2.10: Computed stress on each node before and after the cut.....	135
S2.11: Comparison of Polar Spring Network and Analytical Model.....	136
S2.12: Family of Cuts with different nonlinear factors.....	138
S3.1: Initial view after opening SWS4D.....	139

S3.2: Same dataset after segmentation is finished.....	140
S3.3: Example of changing the contrast using the Vmin/Vmax sliders.....	140
S3.4: Example of changing the current cell to ID 5, which also changes the outlined cell.	141
S3.5: Sample use of Tissue Segmentation.....	141
S3.6: Updated segmentation using the Tissue Segmentation from Figure S3.5 as a mask.	142
S3.7: Example of changing the Z-index to go deeper into the tissue.....	142

Chapter 1

1. Introduction

In this thesis, I present the work I have done towards understanding the developmental processes of *Drosophila melanogaster*, or the fruit fly, focusing specifically on the amnioserosa tissue during the process of dorsal closure. Throughout this work, my main objective has been to accurately characterize shapes and motions of developing cells as well as the spatial patterns of forces underlying these motions. I have tried as much as possible to do my work with an eye towards reuse and have publicly shared the most important analysis systems and libraries of routines I have developed online so that they can be examined and reused by future researchers.

The rest of this introduction is a very brief description of each chapter.

Chapter 2 is a high-level overview of some fundamental concepts and existing research that are tantamount to the understanding of subsequent chapters. It begins with a basic phenomenological introduction to developmental processes and also includes the anatomy and life cycle of *Drosophila* embryos. This chapter also discusses viscoelastic materials, image processing as a tool for quantitative biological measurement, perspectives on the most appropriate mechanical models for epithelia, and basic background on deriving forces from image data, specifically using the technique Video Force Microscopy (VFM) developed by our collaborators [1].

Chapter 3 is a critical review of state-of-the-art force assessment methods for *in vivo* biological matter, specifically cells in tissues. This chapter covers atomic force microscopy, traction force microscopy, magnetic and optical tweezers, laser ablation, and force inference.

Next, Chapter 4 describes the image analysis tool I developed called SeedWater

Segmenter (SWS). SWS enables segmentation of images of epithelial cells, accurately extracting the boundaries of closely packed cells in tissue and tracking cellular regions through time. This tool has become a critical building block for the bulk of the work in this thesis and has been a foundational tool for the work of many of my fellow students both within the Hutson lab and at large. This chapter also describes a brief but important result concerning the propagating nature of cellular pulsations in the amnioserosa. This work was published as D. N. Mashburn, H. E. Lynch, X. Ma, and M. S. Hutson, "Enabling user-guided segmentation and tracking of surface-labeled cells in time-lapse image sets of living tissues," *Cytometry Part A* 81A(5): 409–418, 2012 [2].

Chapter 5 describes the extension of the ideas from Chapter 4 into the third spatial dimension, including an example 3D+time segmentation of a sheet of epithelial cells (*Drosophila* amnioserosa cells/tissue). In addition, this chapter describes some novel algorithms I have developed, both for rasterizing simple geometric shapes in N-dimension voxel grids and also for N-dimensional sparse matrix handling; each of these were foundational components in this SWS4D, a simple but complete system for 3D+time segmentation.

Chapter 6 describes the application of SWS and SWS4D to data from laser hole drilling experiments. This study has provided (and confirmed) some basic insights into the mechanics of wound expansion and healing. This also includes the first 3D analysis of a laser ablation wound and a discussion of my attempts to use force inference on images of amnioserosa during dorsal closure and wound healing and why the system is not a good fit for network-based force inference.

Chapter 7 is a paper in preparation that describes the details of when and how image-based force inference techniques succeed (or fail) in finding an accurate solution.

This chapter uses synthetic data with a known solution to aid in this analysis and is a fairly extensive look at these methods' strengths and weaknesses. After careful consideration, however, it has been assessed that this work needs further evaluation and application to other data sets.

Chapter 8 summarizes the major conclusions from this core work and describes some futures directions of inquiry. My hope for this research is that it can be an informative component in the larger push to understand the physical underpinnings behind biological processes.

Supplement S1 is a section from a larger paper published by our lab: A. K. Jayasinghe, S. M. Crews, D. N. Mashburn, and M. S. Hutson, "Apical Oscillations in Amnioserosa Cells: Basolateral Coupling and Mechanical Autonomy," *Biophysical Journal*, vol. 105(1): 255–265, Jul. 2013 [3]. This paper examined the role of contractile phase on the subsequent expansion of a cell after point ablation. It also showed that CO₂ anesthetization not only causes embryonic tissue to stop contracting but also prevents it from recoiling from laser ablations and can even pause the healing process; this effect is only temporary and removal of the CO₂ restores the embryo's activity. This paper also included a section examining the dynamics of pulsations in three dimensions, showing that apical contractions lead to basolateral expansion and general preservation of the overall volume of the cell. In addition to supporting the work on the pulsation/wound expansion study, I also performed the analysis and wrote the majority of this last section on volumetric analysis, which is what I have included here.

Supplement S2 describes my model for tissue recoil after laser point ablation. It includes a section that follows Timoshenko in describing analytical solutions for the stress around a hole in a continuous 2D surface [4]. We used this solution to measure the

relative stress parameters from live images before and after a point ablation; this was achieved by applying various displacement fields to warp one of the image to match the other (by trial and error). It also includes a section that describes a numerical method for computing the forces from a wound given a nonlinear spring constant.

Supplement S3 is a series of still frames illustrating usage of SWS4D, described more fully in Chapter 5.

Chapter 2

2. General Background

2.1. Introduction

Developmental disorders, be they genetic or environmental, are the root cause of birth defects in infants. The key developmental processes are also applicable to other phenomena and disorders such as cancer (which in one sense is development gone awry). Traditionally, developmental processes have been studied using phenomenology and genetics, but there are limits to the insights that these can tools alone can provide. At some level, gene expression and its coupled biochemical pathways must have some physical manifestation; mechanics is often the missing link between a genetic cause and its resulting phenotype [5]. By tackling the hard problems of defining the shapes, motions, and forces at the cell and tissue levels during developmental processes, we get one step closer to understanding these disorders in the detail necessary to identify real underlying causes and solutions.

Because of its ubiquity in genetic studies, the large body of previous literature, and the ease of experimental accessibility, we are studying these processes in *Drosophila melanogaster*. Embryogenesis is a complex process, and varies greatly between organisms, but the general principles of biomechanics apply equally well to all embryos, regardless of species.

After gastrulation, the most dramatic motions on the surface of the developing embryo occur during the processes of germ band extension, germ band retraction, and dorsal closure. Identification of mechanistic underpinnings for these processes has proved to be extremely formidable. Germ band extension has been widely studied and the mechanism of action (intercalation and convergent extension) is fairly well understood

[6], [7]. Dorsal closure has also been highly studied, and significant progress has been made towards understanding the processes involved [8]–[18], but a number of unanswered questions remain surrounding it. Lastly, germ band retraction is only beginning to yield details about the underlying mechanisms [19]–[22].

The system in question consists of live and developing tissue, which presents some unique challenges in terms of mechanical inquiry. First of all, we go into the problem with a distinct lack of information; for instance, we lack knowledge about the basic tissue properties like elasticity and viscosity. In fact, we even lack knowledge about basic structures and connectivity; for instance, we do not know if the basement membrane of the epithelial layer is attached to the yolk in some way to produce drag or even active force. We also do not know details at the microscopic level, namely which proteins are involved (laminin, integrin, etc.) and where they are localized. To add even more layers of complication, this heterogeneous material is active and can change its properties based on developmental timing or due to chemical and/or mechanical stimuli. Furthermore, some tissue properties previously considered passive, like elasticity, have now been recognized as resulting from active processes [3].

Two of the most powerful tools we use for studying these complex processes are live imaging and laser microsurgery. Live imaging provides detailed information about the cells, tissues, and embryo in space and time, including depth (3D) information. Laser microsurgery is a widely used method for probing mechanics by generating an instantaneous, relatively repeatable perturbation to the system. It is especially important for *Drosophila* studies, because the vitelline membrane renders the embryo inaccessible to direct mechanical probing. Laser microsurgery will be compared and contrasted to other force assessment methods in Chapter 3.

A critical component for using both imaging and laser microsurgery for scientific inquiry is the extraction of data from time-lapse images. Generating quantitative data is the only way to effectively compare experiments with models. Beyond basic analysis tools like counting cells or measuring distances and sizes by hand, we need powerful tools to tackle difficult image analysis problems. This need has motivated us to build such tools, specifically software systems to study time-varying cell shapes, cell connectivity, and even mechanical stress.

2.2. *Drosophila* structure and development

As stated above, *Drosophila melanogaster*, the fruit fly, is used as a model system for a number of reasons. First and foremost, *Drosophila* has a short lifespan and reproduces readily, which makes it not only amenable to studies at all stages of development but also to genetic modifications that require generational tracking, crossing different strains, etc. *Drosophila* also has a relatively simple chromosomal structure that can be leveraged to link subtle genetic changes to readily observable characteristics such as eye color and body shape. Because of this simple structure and its ubiquity in genetic research, the *Drosophila* genome was one of the first to be sequenced among multicellular organisms (2000) [23], second only to the worm *C. elegans* (1998) [24].

Embryonic *Drosophila* is also a model biomechanical system; over spatial scales of ~ 100 μm , its epithelium closely approximates a uniform two dimensional monolayer of cells.

The life cycle and embryonic stages of *Drosophila* have been characterized and refined with considerable care and over a considerable amount of time, especially by Bownes [25]–[27], so here I only hope to highlight some key features of interest. Figure 2.1 shows an overview of these stages.

Time table of embryogenesis



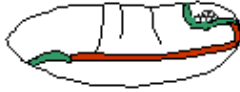

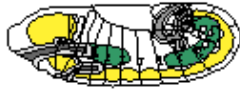


	Stage	Time	Developmental events
	1 - 4	0:00 - 2:10 h	Cleavage
	5	2:10 - 2:50 h	Blastoderm
	6 - 7	2:50 - 3:10 h	Gastrulation
	8 - 11	3:10 - 7:20 h	Germ band elongation
	12 - 13	7:20 - 10:20 h	Germ band retraction
	14 - 15	10:20 - 13:00 h	Head involution and dorsal closure
	16 - 17	13:00 - 22:00 h	Differentiation

Figure 2.1: Stages of fly embryogenesis.
 Reproduced with permission from FlyMove [160]
<http://flymove.uni-muenster.de/Stages/StgTabelle.html>.

The onset of embryogenesis occurs when eggs are fertilized inside adult females. Females may then hold on to the eggs for up to an hour or so before laying them, which can add some uncertainty as to the exact biological age of any given embryo. The

fertilized egg has only three main layers: a tough shell on the outside (the chorion), a thin and waxy membrane just inside that (the vitelline), and a yolk that comprises the interior. Initially, this egg is actually a single cell, with a single free-floating nucleus in the yolk. The initial zygotic (fertilized) nucleus will divide 12 times in succession, resulting in a state with 4096 free nuclei. During division, these free nuclei migrate to the surface of the yolk and afterwards, cell membranes are pulled in between them to create a single layer of cells in a process called cellularization (about three hours after fertilization) [28]. The cells at the surface then comprise a monolayer surrounding the entire yolk.

After one more cell division cycle, this epidermal tissue undergoes gastrulation, an invagination of some of the cells resulting in a small tube within the interior of the embryo. The internalized cells will become the gut and other internal organs (endoderm and mesoderm); the remaining external cells will become ectoderm. These cells remain a monolayer and are largely unaffected by the interior layers.

The next stage of interest is germ band elongation during which the ventral portion of the ectoderm, also known as the germ band, becomes extended, while the cells on the dorsal side, the presumptive amnioserosa, become bunched up. This process is clearly understood to be the result of cell intercalation in the germ band, where cells crowd together with motion towards the ventral midline (along the medial-lateral direction) and then settle back to isotropic shapes, resulting in extension along the proximal-distal axis (on the ventral side) and contraction along the medial-lateral axis [6], [7]. Because this is on the surface of an ellipsoid, the germ band tissue on the ventral side wraps around the caudal end of the embryo and compresses the amnioserosa cells on the dorsal side along the proximal-distal axis (and thus extending them along the medial-lateral direction). Towards the end of this process these bunched, dorsal amnioserosa cells

become obviously differentiated into a distinct tissue. They assist in later developmental processes before eventually invaginating and undergoing programmed cell death, or apoptosis; thus, amnioserosa cells have no descendants in the adult and serve a purely developmental purpose.

After a brief pause, the next stage following germ band elongation is germ band retraction (GBR), starting a little over seven hours after fertilization. During GBR, cell-level forces uncoil the germ band back around the tail end of the embryo, forcing its cells to the ventral side of the embryo and leaving the amnioserosa cells on the dorsal side of the embryo. Figure 2.2 shows a confocal microscope image from a time series of GBR (lateral or side view).

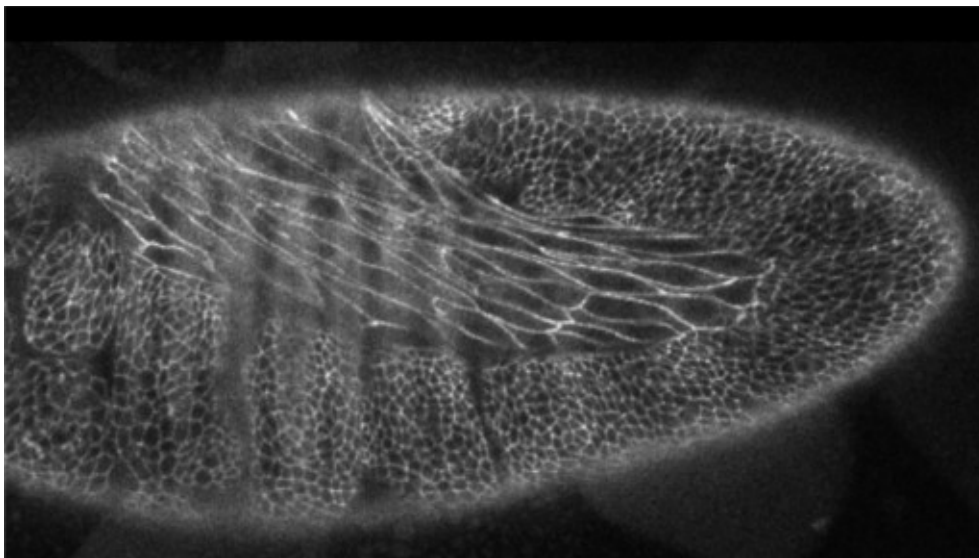


Figure 2.2: Germ band retraction, lateral view.

This is a confocal microscope image of a GFP-cadherin expressing embryo in early germ band retraction. The main tissue with smaller cells is the germ band tissue and the tissue in the upper central region with larger cells is the amnioserosa.

Once the embryo has fully retracted, dorsal closure begins (about ten hours after fertilization). During this process, the germ band cells bordering the amnioserosa move

towards the dorsal midline and eventually cover the entire amnioserosa tissue. Dorsal closure starts with a series of pulsatile contractions in the amnioserosa cells; these cells slowly become more cuboidal as they bunch up and eventually invaginate and apoptose. In addition to contraction of amnioserosa cells, two secondary processes occur that drive dorsal closure. The first process is the formation of an actin-myosin purse-string around the perimeter between the amnioserosa and germ band cells. The second process is filopodia-driven zipping that occurs at the proximal and distal (head and tail) ends of the amnioserosa tissue. Once this zipping process is complete, the two flanks of the germ band fuse together, completely forcing the remaining amnioserosa cells into the embryo's interior where they undergo apoptosis. After this, the embryo forms breathing tubes and develops into the larval stage where it is able to crawl around. Although my research is primarily concerned with the aforementioned processes, there are clearly a myriad of less relevant ones occurring simultaneously.

Because of its unique role and singular features, the amnioserosa tissue is of particular interest. During the latter two processes (germ band retraction and dorsal closure), the amnioserosa undergoes radical shape changes at both the cellular and tissue level. At the beginning of germ band retraction, this tissue forms a U-shape over the dorsal end and sides of the embryo, with individual cells being stretched out into thin filaments with extremely large aspect ratios (roughly 10 or 20 to 1). After retraction the cells become very squamous with an average width of about 10-20 μm and an average thickness of about 2-3 μm . Despite these extreme shapes, these pancake-shaped cells still form a tight two dimensional network. During dorsal closure, these cells become much less squamous as they contract in the plane of the epithelium and extend basally (towards the interior of the embryo).

2.3. Microscopic mechanical models

All of these bizarre and interesting processes are driven primarily by one thing: cellular contractile forces generated by molecular motors. Since the embryos are small (only about 200 μm wide and 500 μm long), the Reynolds number for cells and tissues in the system is extremely low, and inertial effects are negligible. What *do* have great significance are the viscous and elastic properties of tissues. All liquids and/or suspensions are subject to viscosity, and biological systems are no exception. In fact, unlike a simple liquid or suspension, tissue is made up of organelles and fibrils packed tightly together within cell membranes. This creates an effective viscosity that can exceed the viscosity of the medium alone (cytoplasm). If cells were only viscous, however, they would be highly uninteresting, only moving in the direction of an applied external force with some coefficient of damping. Instead, because cells also have a cytoskeleton made up of microtubules and actin filaments, they are capable of both compressive and tensile resistance, or elasticity. As discovered recently, these elastic properties in *Drosophila* tissue are actually due to the combined effect of numerous active contractions that require ATP (adenosine triphosphate), the primary chemical store of cellular metabolic energy [3]. Overall, these effects create a system that is both viscoelastic (something between a solid and liquid) and capable of active contraction (or extension) in any particular direction.

Even without active contractions, materials with passive viscoelasticity can have extremely novel and unusual properties. These so-called soft matter systems have been the subject of intense scrutiny and research over the past few decades [29], [30], [30]–[32]. Since viscoelastic materials are neither solid nor liquid, their behavior can be difficult to understand. To this end, simple mechanical models have been developed to

describe the behavior at the continuum limit of these materials. The core components in these models are the spring obeying Hooke's Law and the dashpot (or damper) which obeys the velocity-dependent (or viscous) force equation for a Newtonian fluid.

Describing the properties of an infinitesimal sub-unit can readily elucidate the properties of the tissue as a whole. The most basic forms of simple mechanical models are one-dimensional but they can be extended into two and three dimensions as well.

The Maxwell element consists of a spring and a dashpot connected in series (Figure 2.3), and in this arrangement the system behaves as a viscoelastic liquid. Over very short timescales, this system will respond elastically, returning to its original state given a brief enough exposure to force. For medium interval forces, the system will deform elastically as well as plastically — once the force is released, it will instantaneously return to a state of partial deformation. Given a constant force over a longer period of time, the system will deform indefinitely. This steady state flowing behavior is what makes this system "liquid-like".

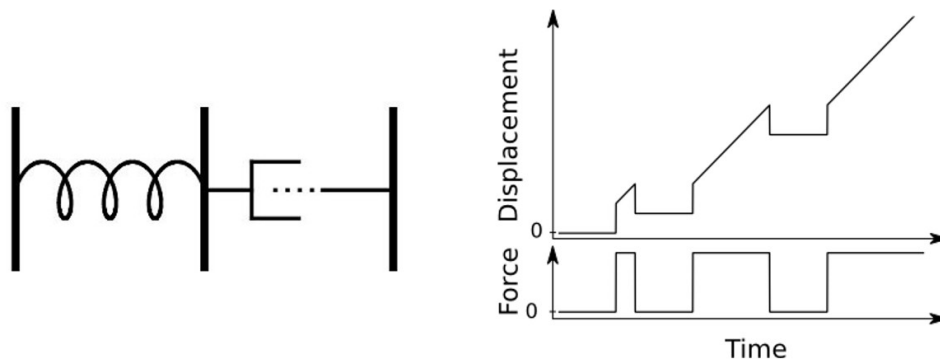


Figure 2.3: Maxwell Element and an example creep response graph (displacement vs. time) for a given time varying (step-wise) external force (bottom left).

The other of the basic mechanical model is the Kelvin-Voight model, which

consists of a spring and a dashpot connected in parallel (Figure 2.4). In this arrangement, the instantaneous response of the system is the opposite of the Maxwell model: a sufficiently brief force will not deform the system at all until the dashpot has time to react. Application of constant force will slowly deform the system, producing an exponentially decaying deformation; over very long times, the system reaches equilibrium. Once that force is released, the system returns to its original state exponentially. The time constant for the motion is proportional to the damping coefficient divided by the spring constant. The steady state equilibrium behavior is what makes this system "solid-like".

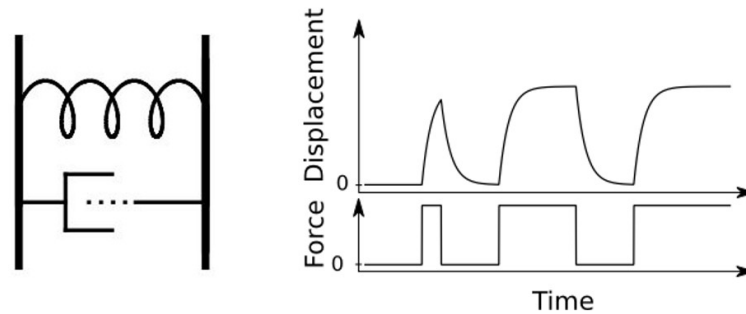


Figure 2.4: Kelvin-Voigt Element and an example creep response graph (displacement vs. time) for a given time varying (step-wise) external force (bottom left).

These mechanical circuits are actually completely analogous to electronic circuits, interchanging voltage, charge, and current for force, position, and velocity. Springs correspond to capacitors (storing energy) and dashpots correspond to resistors (dissipating energy). Moreover the same equations apply to both types of systems, with a series circuit behaving like a resistor and capacitor in parallel, and a parallel circuit behaving like a resistor and capacitor in series.

Much in the same way that complex analog circuits can be constructed out of resistors and capacitors, mechanical models with more complex behaviors can be constructed out of simpler components. Examples of slightly more general models are the simple linear system (SLS) and the Burgers model. The SLS is a Kelvin-Voight element with a spring added in series, which still behaves as a solid, but with an instantaneous elastic response as well as a damped one. The Burgers model is a SLS with a dashpot added in series (leading to liquid behavior after long times) [32]. In some of our previous studies, we have also used two Kelvin-Voight elements joined in series, maintaining a viscoelastic solid character but with two characteristic time constants [31]. These systems have properties of viscoelastic solids on some time scales and viscoelastic liquids on others, a close approximation to the type of behavior that we see when perturbing live cells.

Some studies have also shown that biological materials do not clearly exhibit a single time constant under creep-stress experiments [33], [34]. Rather, they show that biological matter is best described as a power-law material, meaning that the position goes as some power of time. This is due to the fact that there are a variety of time scales corresponding to different scales in the subcellular cytoskeleton. Rather than describe the system with a variety of time constants, it is simpler to just use a power-law description. Power-law exponents ranging between zero (completely solid) and one (completely liquid) then provide a straightforward numerical value for how much the tissue acts like a solid versus a liquid.

2.4. Image processing and segmentation

2.4.1. Image processing

Images, so far as they concern the present work, are essentially two-dimensional

rectangular arrays of numerical values. Because images have such a straightforward mathematical representation, it is easy to perform mathematical operations on them. For instance, it is easy to change the maximum brightness in an image by multiplying all the values in the array by the same number. In a similar way, some features can be quickly identified by cutting off values that are above or below a certain point; this thresholding operation results in enhancement of certain regions of an image at the expense of other features or noise. Images of the same size can be combined to generate new images by applying operations to pixels in the image; for instance, images can be added or subtracted from each other to produce sum or difference images. Information from surrounding pixels can be combined as well. One simple example of this is background subtraction, where the mean value of a region without features is subtracted from the entire image to remove an offset. A more complex example would be a Gaussian blur, where values in the new image are calculated by sampling pixels in the original image with weights that fall off radially with a Gaussian profile. This results in each pixel affecting its neighbors in the final image and produces the same effect as optics that are out of focus.

Still, though these basic tools can be useful and interesting, they are also limited; for instance, thresholding cannot deal with images that have uneven brightness, though features clearly remain distinct to the eye. For this reason, more advanced techniques have been developed, like contrast-limited histogram enhancement (CLAHE) which optimizes each pixel's value based on a histogram of pixels in the surrounding region, giving rise to a contrast enhancement that can simultaneously enhance features in bright and dark regions within the same image [35].

2.4.2. Watershed segmentation

Because our images are derived from the fluorescent emissions of tagged protein molecules, pixels with larger values (brighter) correspond to regions of higher densities of fluorescent molecules. The marker that I am primarily concerned with is GFP-cadherin, which labels membrane-bound cadherin proteins. When using this label, it is easy to discern cell boundaries and cell shapes in images.

Because we have clear edge markers, these images are particularly amenable to the watershed algorithm for digital identification of boundaries. The watershed algorithm, true to its name, simulates the action of water accumulation, and identifies regions similar to the way they are defined for geographical terrain. See Figure 2.5 for more information.

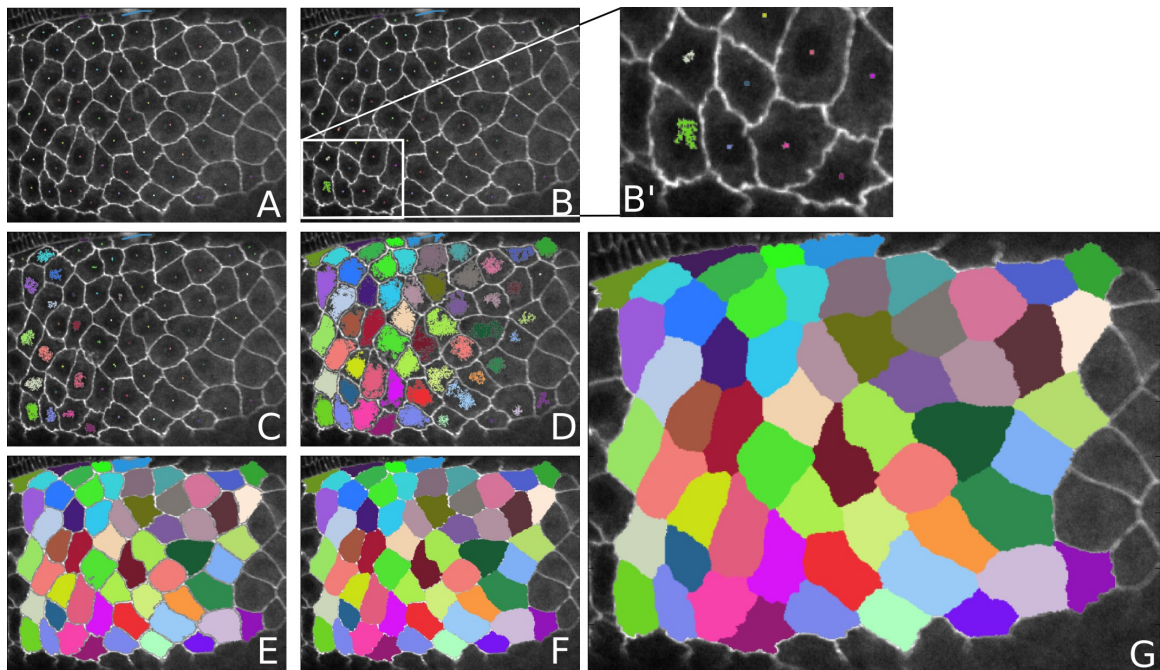


Figure 2.5: Basic idea behind watershed segmentation.

(A-G) show the progression of the watershed algorithm at different time points. (B') is a close-up of the corner of (B). Since the algorithm fills outwards from the seed points, globally moving from the darkest pixels to the lightest ones, this process is analogous to colored pools of water being filled from individual taps within each cell. The rate is maintained to keep the water level uniform in all cells at all times. Once two different pools meet, they prevent each other from continuing, so the process continues until it reaches the brightest values. After this, the regions are identified and the edges are clearly defined by the boundaries between these watershed basins.

2.5. Macroscopic tissue models

2.5.1. Foam Model

Biological material is far from homogeneous; it is often pictured as a net of cell membranes, with each pocket containing a nucleus and cytoskeleton (and also organelles and cytoplasm). Because of this structure, intercellular forces have previously been assumed to act mainly along the borders between cells, as is the case with a foam. Our collaborators at the University of Waterloo have developed finite element codes that they have used to appropriately test these foam models [36]. Their conclusions are that taking this foam model without any additions to account for tensions within the cells, simply does not match the results we see in our data. The pure 2D foam model predicts no recoil at all when cutting a cell center, but large recoils when cutting a cell edge. We clearly see large recoils from both cell edge and cell center wounds in our experiments, and so this realization has led us to move toward the opposite extreme: a homogeneous elastic sheet model. This model represents an actin network on the surface of the embryo connecting all cells. In fact, estimates from previous work by our group show that only about 25% of the forces in the tissue are due to tensions along the cell-cell boundaries. The rest is most likely coming from this actin sheet.

2.5.2. Finite Element Simulations

As mentioned in the previous section, Brodland *et al.* have developed finite element (FE) codes that treat cells as viscous body elements with force-carrying cell-cell interfaces (trusses) [36], [37]. Figure 2.6 summarizes the components of this model and the biological mechanism by which each of these arises.

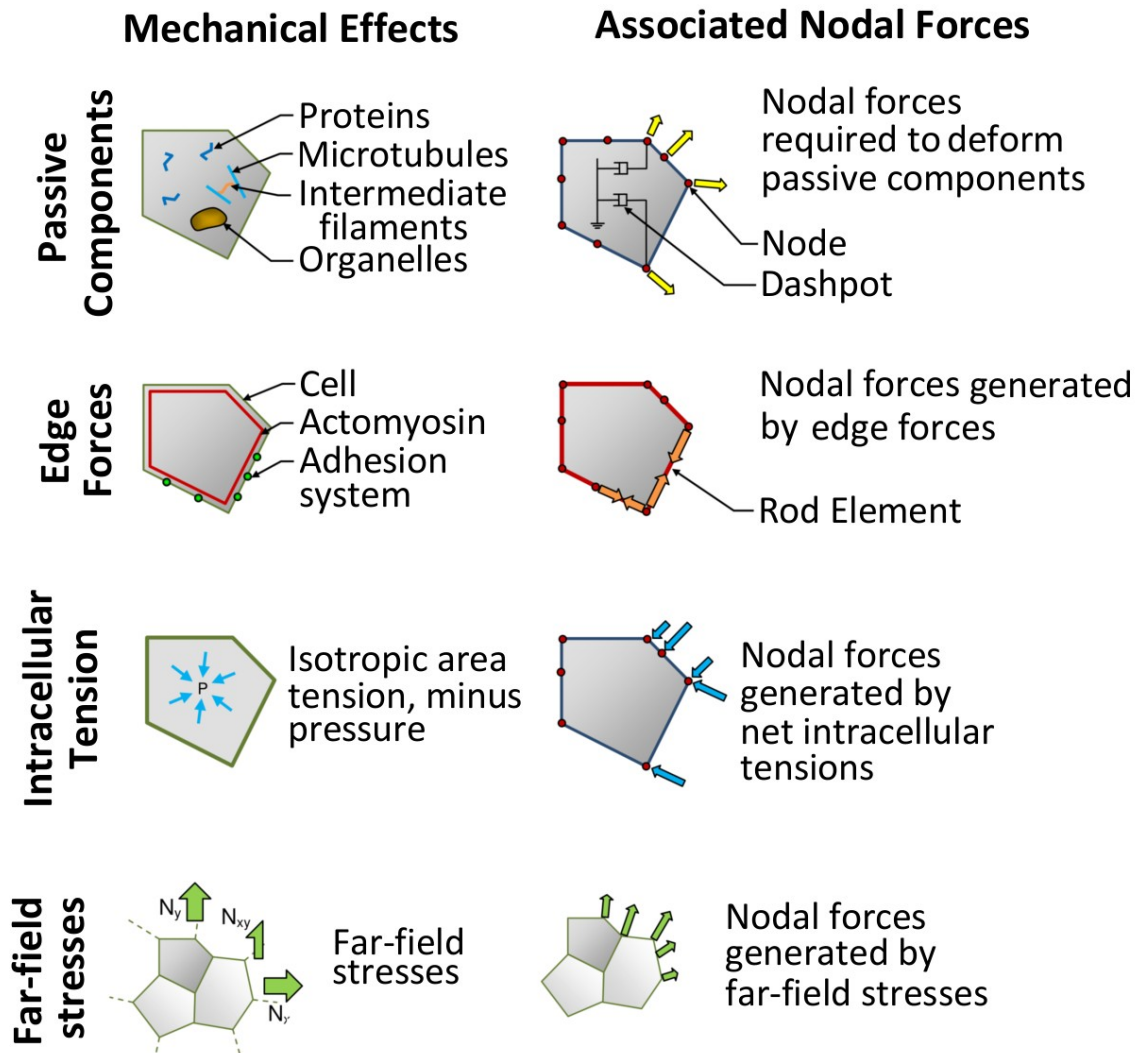


Figure 2.6: Summary of components in the the Cellular Finite Element Model from Brodland *et al.*

[37]. The cytoplasm of the cell body generates viscous damping and is simulated by two orthogonal dashpot networks oriented parallel and perpendicular to the long axis of the cell. The edges of the cells are treated as either active or passive truss or tensile elements (described above); additionally, they may be programmed with a force-response that simulates a spring or viscoelastic element. Thus, the edges can simulate all the behaviors of the actin-myosin filaments known to run along the boundaries of the cells. In addition to the cytoplasm, the cell also has a "pressure" which is a combination of restorative forces generated by true fluid pressure that can build up due to the impermeability of the cell membrane as well as in-plane surface tensions generated by the actin-myosin network near the apical surface of each cell. Stress is applied to the patch at the boundary (with boundary conditions).

Each of these components reacts differently when acted on by a force, and the net

motions are calculated by building a large set of all the coupled force equations and time-stepping the solution with a numerical solver. These models have been proven effective in a number of different biological systems [15], [36], [37].

2.6. Force Inference with Video Force Microscopy

Although FE simulations have been proven quite effective at modeling systems, the process of generating a simulation to match specific live data is not only tedious but also computationally impractical. Recently, Brodland *et al.* have shown that one can set up the finite element equations and alternatively solve the inverse problem to infer the forces driving an observed motion [1]. Solving this inverse problem actually takes significantly less computational time than a single finite element simulation. Not only that, but as we will show in Chapter 7, it is even possible to get error bounds on the solutions! This technique, called Video Force Microscopy (VFM) originally used information across multiple frames of an image sequence to generate the over-determined system necessary to solve for the forces using linear least squares. We have since determined that two frames are sufficient for a dynamic solution and that solving the problem with only a single image is also possible if the system is near static equilibrium. Chapter 7 goes into much more detail about VFM and competing force inference methods [1], [38]–[41].

Chapter 3

3. A Review of State-of-the-Art *in vivo* Force Measurement Techniques

3.1. Introduction

Traditionally, developmental biology has been an observational science. Over time, scientists have developed a sophisticated understanding of genetic processes through many years of discovering and scrutinizing biochemical pathways; these simple biochemical reactions are the origin of a vast multitude of physical attributes, behaviors, and even developmental processes. Development is unique in that a single copy of an organism's genetic code within one cell orchestrates a series of transcriptional, chemical, and physical actions with an unrivaled level of sophistication and coordination, resulting in a fully developed multi-cellular organism.

Over the past decades, more and more effective observational techniques have become available for biological systems, with live imaging being easily one of the most important. We are now not only able to accurately observe spatial patterns with incredible fidelity, but also record them over a wide range of temporal scales; coupled with the advent (and incredible precision) of molecular fluorescent biosensors, this has led to an explosion of information about biological systems. With this modern toolkit, the intricate details of developmental patterns, both molecular and structural, are now accessible. Moreover, complex developmental processes such as tissue extension, retraction, folding, and invagination have been mapped with great accuracy. For example, careful observation has been able to prove that the process of Germ Band Extension (GBE) in *Drosophila melanogaster* is the result of cell intercalation, a carefully coordinated rearrangement of cells. Researchers have even been able to isolate the principal structural molecules that drive this process [42], [43].

Despite resounding successes like GBE, however, mysteries still abound about developmental biomechanics. The purely genetic/observational viewpoint is insufficient for understanding systems that involve delicately balanced forces such as Germ Band Retraction (GBR). For these systems, it is necessary to consider not only the motions of cells, but also the forces driving these motions. With the combined understanding of both the genetic/molecular activity and the physical forces, researchers can also tackle more subtle issues, such as environmental perturbations which are the root cause of many birth defects. Temperature variation, chemical contamination, physical pressure, and physical damage can each have profound effects on development. Combined physical/physiological understanding is key to fully understanding and potentially managing or correcting these types of environmental perturbations [5], [44]–[46].

The internal forces that actually drive morphogenetic movements are surprisingly difficult to measure, and a number of creative methods have been developed to do so. In this chapter, we provide a brief survey of force measurement techniques relevant to biological systems in general and their relative strengths and weaknesses for developmental mechanics.

3.2. Measuring Forces

Measuring physical forces in developing embryos is a very challenging problem, primarily because the key forces are internal to the system and changing continually; also, most (if not all) of the system is often inaccessible to standard mechanical probing. This is especially true for developing *Drosophila*, where the vitelline membrane not only provides a physical constraint for morphogenetic processes like Germ Band Extension and Germ Band Retraction, but also, but also maintains the internal osmolarty of the system and provides a thin extracellular space for signaling molecules.

Modeling is one way to overcome this problem: models that replicate observed behavior can suggest certain underlying patterns of forces. However, in the immortal words of Box and Draper, "all models are wrong, but some are useful" [47] — just because a model reproduces a particular behavior does not make it correct. Multiple models (actually infinite) can be constructed for the exact same patterns of motion. One example is in the models of invagination for developing sea urchins [48], [49]. *In vivo* force measurements are the key to truly resolving discrepancies between models.

Force measurement techniques can be broken into two main categories: perturbative and non-perturbative. Perturbative measurements rely on the response of the tissue to active stimuli, while non-perturbative measurements merely observe the tissue using some form of passive probe. The most important perturbative techniques are:

- Direct contact: atomic force microscopy (AFM), micropipette, and parallel plate/rod experiments
- Field-based: Magnetic and Optical Tweezers (MT, OT)
- Tissue removal: Laser ablation

and the most important non-perturbative ones are:

- Adhesion to a deformable substrate
- Embedding deformable objects (oil droplets or FRET molecules)
- Force Inference

Perturbative techniques induce highly controlled, reproducible and modifiable changes, which are invaluable for hypothesis testing. Non-perturbative methods in general have great potential to simultaneously record measurements across multiple locations at once. Each technique has strengths and weaknesses, and each has been employed successfully

to make insightful measurements, and a number of key studies have relied on combinations of techniques.

Direct contact techniques have excellent accuracy since they can be calibrated to real units unambiguously and as such have experienced excellent results for single cells and cell pairs. However, they suffer because they are only measuring response to external forces normal to the embryo's surface and the forces of biological systems are primarily internal and parallel to the surface; also, many cell-cell contacts are often inaccessible, buried inside surface casings or other tissue layers. Magnetic and optical tweezers can partly overcome these limitations, but have their own challenges and problems. Laser ablation is virtually ubiquitous now as a force assessment tool; although it does provide measurements correlated with the internal stresses in tissue, it can at best measure force ratios but not true magnitudes. It also destroys the integrity of the tissue in question, meaning that subsequent measurements are no longer equivalent to those in unwounded tissue. Adhesion to deformable substrates has enjoyed great success on cells and cultured monolayers, but has yet to be applied to *in vivo* embryonic tissue. Embedded deformable objects and Force Inference have incredible promise, but are still being actively refined and present technical and computational challenges.

3.3. Direct Contact Techniques

The most natural way to measure forces is using a push or a pull with a physical object. The gold standard in direct force measurement is using a cantilever-based atomic force microscope (AFM). AFM has been used to study everything from tensile strength of individual cytoskeletal components like actin fibers and microtubules to the creep response of single living cells [34], [50]–[53]. Similarly, a micropipette can be used to measure contact forces, and Maître *et al.* have successfully used it to measure the

adhesion strength between two cells in contact [54]. Additionally, Harris *et al.* describe an experiment to measure the tensile strength of a cultured sheet by stretching it between two rods after removing the substrate on which it was grown [55].

Although AFM and related techniques have been quite successful at studying molecules, cells, and even cultured epithelia, they struggle for *in vivo* embryonic tissue both due to the inaccessible nature of many embryos and because the forces primarily act perpendicular to the plane of the primary forces in the system. When it is applicable, such as in *Xenopus* embryos which permit direct contact with cells, the results tend to be either used to map the surface features of the embryo or report on bulk properties like Young's modulus [56]. Daniels *et al.* describe it particularly well:

Current cell mechanics methods (AFM), micropipette suction, parallel plates, magnetic bead twisting, and cell poking... require a direct contact between the cell surface and the physical probe to extract the rheological parameters that describe the mechanical properties of the cell. However, cells in their physiological environment are largely inaccessible to mechanical probes because they are often organized within an extracellular matrix and buried in soft and solid tissues. Furthermore, the developing embryos of many organisms are housed within rigid shells, making them inaccessible to physical probes. Therefore, to test the fundamental biophysical properties of cells in their physiological milieu, there is a need to develop a biophysical method that does not require direct contact with the cell surface to obtain information about the mechanical state of cells *in vivo* [57].

3.4. Magnetic Tweezers

Magnetic tweezers (MT) is the application of large magnetic field gradients to paramagnetic beads attached to the sample of interest. The beads move towards regions with higher intensity fields and by controlling the working distance, magnetic field gradient, and bead size, the applied force can be measured very precisely. By controlling the direction of the field, potentially using opposed and tunable gradients, it is also possible to generate arbitrary-directional forces that can be switched at various times within a single experiment. Kollmannsberger and Fabry give an excellent technical

overview of a MT setup [58].

MT have been used in single-cell studies for a variety of purposes. A number of groups have used MT to study creep response, namely how cells deform over time, and have determined that there are multiple time-scales involved in the process [59], [60].

A number of groups have used a variation of MT called magnetic twisting cytometry (MTC), pioneered by Wang *et al.* [61] where two perpendicular fields are used to precisely induce torques instead of straight-line motions for the study of rheological properties of cells [33], [62].

Since MT relies on embedded magnetic beads, in embryonic studies this can present technical difficulties; for instance, placing a bead in one particular location is not always practical. Some embryos like *Xenopus* permit surface placement of beads, but inserting a large bead into a small cell is difficult to do without disrupting or destroying the cell of interest. One novel solution for *Drosophila* embryos (which do not permit direct access at all) is to quite literally "shotgun" it: inject multiple beads in solution during the pre-cellular stages and let cells close around them at random. Using this approach, some beads should statistically end up in locations of interest.

Beads can also be concentrated in certain regions using this injection technique by attracting them with a field gradient before cellularization. This approach has the added benefit that it permits bulk force application to tissues. This is the exact approach taken by Desprat *et al.*, but using ferrofluid instead of magnetic beads [63]. In their experiment, they rescued production of Arm in an embryo which was damaged by laser ablation during Germ Band Extension, proving that mechanical strain triggers Arm production.

Kumar *et al.* used bulk injection of 100nm beads to show that nuclear size and shape are affected either reversibly or irreversibly by varying levels of added force

applied towards the posterior of the embryo during Germ Band Extension [64]. They go on to show how this affects patterned gene expression in GBE and GBR as well as myosin II localization, confirming and providing a possible mechanism for mechanically regulated gene expression. By applying the force from the left or right side instead, they were also able to show that there is a handedness in gene expression.

3.5. Optical Tweezers

Optical tweezers (OT), like magnetic tweezers, rely on field-induced forces rather than contact forces to manipulate microscopic objects. In OT, a focused laser beam creates a central region of high photon density that falls off both along the beam path and laterally. A transmitting object, like a microsphere, feels photon pressure from internal reflections that are balanced at the focal point, but grow the further the object is from the center, pulling the object back towards the center. The forces produced by OT and MT are both very well-defined, however, the maximum force OT can produce is much smaller than what MT can. Whereas MT can produce forces of up to 100 nN [58], OT typically has peak forces around 100 pN [65].

OT often rely on beads attached to the system of interest, especially for molecular studies, but they are also capable of direct manipulation of larger objects without beads; this has some nice advantages over MT. In addition, OT can be holographic using an SLM to create multiple, independently-controllable spots [66]. OT have had tremendous usage in the study of both molecules and cells; Zhang *et al* presents an excellent overview of the uses of OT for single-cell studies [65].

Although OT techniques seem quite promising, they have had very limited use in the developmental mechanics community. Recently, however, Bambardekar *et al.* has used OT for detailed force measurements in early-stage *Drosophila* embryos [67]. They

used OT both with beads and by directly manipulating cell-cell interfaces; they showed that direct manipulation could achieve roughly 30% of the force on a bead given the same laser power. By pulling on boundaries with both oscillatory forces and simple pull-and-release experiments, they precisely determined the interfacial tensions along the boundaries (found to be near 100 pN) and show that there are two time scales involved in the viscoelastic relaxation response. They also include a simple model that can reproduce the relative deformations of nearby and distant cell-cell interfaces in response to oscillatory behavior, proving that Maxwell and Kelvin-Voigt elements are insufficient to reproduce the behavior but that the combined SLS model can; they go on to estimate the spring constants and damping coefficient that best fit the experimental data.

3.6. Laser ablation

Laser hole drilling of epithelial tissue is very much akin to the hole-drilling method for measuring residual stress in solid mechanics [68]. Drilling a hole removes a small (or possibly large) bit of cellular material which results in a deformation response of the remaining material away from the hole. For solids, mapping the displacements around a hole provides information about the internal stress and elasticity of the system. Viscoelastic solids are slightly more complex, exhibiting a time-dependent response before reaching a steady state. In biological tissue, there is also normally a healing response whereby the tissue reverses the expansion and fills back in the wounded region until it restores the continuous epithelium. This all makes hole drilling more complex to interpret but also richer in the potential feedback it can provide [31], [69], [70].

Laser ablation for *in vivo* tissue studies typically relies on short bursts of tightly focused UV light to avoid heating surrounding tissue which would lead to collateral damage of the potentially interesting nearby tissues [71]. UV is normally transmitted

through tissues of interest without interacting, but near a high intensity focal region (small, ellipsoidal), multi-photon absorption leads to rapid heating and tissue ablation. This also means that it is possible to ablate not only the surface of a biological sample, but also within deeper layers [72]. Ablation beams are typically passed through the same optics as fluorescent illumination sources, leading to minimal additional setup near the sample compared to magnetic tweezers and direct contact techniques [73], [74].

One important aspect of laser ablation is that it necessarily perturbs the system in an irreversible way, meaning that repeated measurements of the same animal will not necessarily produce results consistent with unwounded animals (even if the wounds heal and development continues normally). Also, laser ablation relies on laser-tissue interactions which have been shown to be extremely complex [72], [75], [76]. Lastly, displacement maps after laser ablation do not provide a direct way to calculate force or stress; at best they can determine the ratio of stress to Young's modulus. However, given a value for Young's modulus (determined using another technique), it is possible to calibrate laser ablation more precisely for true force measurement [31].

Despite its shortcomings, laser ablation has been used very widely for its simple, comprehensible effect on the tissue: removal of a small bit of tissue and the stress that it was carrying. There is no ambiguity about the location of the perturbation, and laser ablation can be applied to any tissue regardless of its size, shape, or genetic makeup. It has become so ubiquitous that it is often used in conjunction with many of the other force measurement techniques described in this chapter.

Using laser ablation as a force estimation technique requires detailed quantification of temporally varying displacements around the site of the wound. Two important time processes are involved: fast recoil and a slower healing response. When

studying the fast recoil, line scans are an effective method for tracking linear motion of nearby cell edges along a line passing through or close to the site of the wound; scanning a confocal microscope in only one dimension greatly reduces the time per frame and allows for detailed study of the early response. These motions can be built up into a 1D+time kymograph image, showing the displacements of nearby or distant cell edges through time. Fits to these kymographs have revealed a tremendous amount about the viscoelastic properties of embryonic cells and tissues [15], [20], [31], [77], [78]. More complex patterns of motion can be gleaned from the details in the displacement patterns around the site of the wound, and this requires slower live imaging. Many groups have successfully gleaned information from the dynamic patterns around wounds [3], [15], [18], [20], [31], [70], [79]. Light sheet microscopy has the potential to recording both the fast dynamics and the full spatial patterns simultaneously, so future studies doing so are likely to reveal interesting features of the wound response that have not been accessible before.

Many variations of laser ablation have been used to study tissue. Other than point ablations, the most common procedure is the line cut, severing a linear section of tissue [73]. Mayer *et al.* used line cuts to study stress anisotropy in *C. Elegans* embryos, showing that cortical flow of actin leads to polarization [80]. Lynch *et al.* performed stress analysis in *Drosophila* embryo during Germ Band Retraction, mapping the stress magnitude and direction in different germ band segments and analyzing cell shapes to determine the contribution of body stresses and edge tensions on the tissue [21], [81]. Extended cuts can be created with multiple lines, allowing entire tissues to be damaged or cut free [74], [81], [82]. Bonnet *et al.* used laser light focused into an annulus to instantaneously cut free entire sections of epithelial tissue in *Drosophila* pupae; with this, they were able to

estimate stress anisotropy in bulk patches [83]. Jayasinghe *et al.* have developed a system for multipoint ablation using a holographic spatial light modulator (SLM) which can be used to cut loose a single cell from an epithelium, allowing a number of interesting measurements [84], [84]. Hutson *et al.* have used sequential ablations and modeling to show that wounded cells do not immediately lose their integrity, despite the lack of any visible structure at the site of the secondary wound during live imaging [20]. Kiehart *et al.* used multiple ablations to continuously disrupt canthi formation during dorsal closure [73]. Rauzi *et al.* used multiple ablations to disrupt Myosin II activity and show that it is the main mechanical contributor to junction remodeling [85], [86].

3.7. Traction Force Microscopy and Bed-of-Nails

Force measurement through adhesion to deformable substrates has enjoyed great success both for individual cells and cultured layers of cells. Bed-of-nails and traction force microscopy (TFM) using polymer gels can each accurately measure forces of adhesion at multiple locations in a cell simultaneously [87]–[91]. Maruthamuthu *et al.* has extended TFM to use on cells in contact to measure cell-cell adhesion forces [92] and Trepats *et al.* have used it to measure the traction forces for cells in a migrating sheet [93]. Although these techniques have undeniable value for *in vivo* experiments, researchers have yet to apply them to *in vivo* studies of intact embryos because they currently rely on basement contact with a synthetic substrate.

3.8. Embedded deformable objects

Embedded objects provide a simple way to track a variety of motions in cells. Fluorescently tagged proteins are by far the most common type of trackable object in cells, but are rarely used for that purpose because resolving individual molecules requires super-resolution imaging [94]; more often, the fluorescence signal is measured in

aggregate instead. Injected fluorescent polymer beads, on the other hand, can be large enough that normal microscopy can easily distinguish individual beads. The spherical shape of beads also gives them very well-defined drag coefficients, making them ideal for rheological measurements, such as measuring tissue elasticity or viscosity [57], [95], [96].

Deformable embedded objects offer a unique variation on this technique: by placing objects with known stress-strain relationships in series with biological structural elements, *internal* forces in cells and tissues can be measured. Campas *et al.* describe the use of cellular-scale deformable objects (oil droplets) embedded between cells [97] and numerous studies have been performed using FRET (Förster/Fluorescent Resonant Energy Transfer) molecules embedded in key structural proteins [98]–[100]. Additionally, since extensive work has gone into understanding the effect of cytoskeletal forces on nuclear shape, the nucleus itself can be used as a force reporter [101]–[104].

Campas *et al.* used cell-sized droplets of fluorocarbon oil functionalized with ligands (integrin or cadherin) to measure the forces between cells in 3D in mouse mesenchymal cell aggregates and in living mouse embryos (near an incisor tooth bud). Forces are computed by carefully observing an oil droplet's shape and using the local curvature of the surface and solving Laplace's equation over the system. One important conclusion from their work is in showing that maximal cell tension produced in cell aggregates and *in vivo* are roughly equivalent for mammalian mesenchymal cells in living dental mesenchyme (around $1.5 \text{ nN}/\mu\text{m}^2$) [97].

FRET is dipole-dipole mediated nonradiative transfer of energy from a molecular donor chromophore to an acceptor; fluorescently exciting the donor leads to an (attenuated) emission from the acceptor [98]. Since the strength of dipole-dipole

interactions are inversely proportional to the 6th power of distance, this energy transfer is highly dependent on the distance between the donor and acceptor, making the FRET signal (from the acceptor) a sort of molecular ruler [105]. Additionally, inserting a "molecular spring" (a protein that elastically unfolds under tension with some stress-strain relationship) between donor and acceptor produces a complex capable of registering force as well. When placed between the load-bearing structural molecules (*e.g.*, actin, microtubules, integrin, or cadherin), FRET molecules can visually display the magnitude of strain (and therefore force) at each location in a system, allowing forces to be registered as a dynamic three dimensional visual field. For this reason, FRET has incredible potential for revolutionizing biomechanical measurements at the molecular and cellular scales. Cost *et al.* provide an excellent overview of FRET techniques in general [98] and Cai *et al.* has used FRET successfully for *in vivo* studies of the role cadherin plays in developing *Drosophila* oocytes (specifically how it affects cell sorting in nurse cells and border cells) using a cadherin-bound FRET reporter from Grashoff *et al.* [100], [106]. Using this system, they show that the fronts of migrating clusters have higher tensions than the backs.

Despite FRET's advantages, it has a number of shortcomings as a force probe. Because the signal goes as the inverse 6th power of distance, it has a very limited range of action. This means that a suitable linker molecule must be chosen as a force probe for the given interaction and then encoded genetically. Also, chromophores are often highly sensitive to environmental factors like pH, so an additional control is usually required for calibration. Also, it remains to be seen how accurately FRET (a relatively noisy ensemble molecular reporter) can be tuned to make detailed force measurements — by and large at present it is used to assess the location of high force regions rather than make

precise measurements.

3.9. Force Inference

One of the chief disadvantages of mechanical modeling is the ad-hoc nature of setting the model parameters. Force inference (FI) flips this process around: rather than require manually tuning initial conditions and model parameters to get results similar to experiments, FI uses an experimental data set *as* the initial state and *calculates* the model parameters (forces) for it given a set of assumptions. The field is burgeoning and extremely active at the moment.

We are aware of four major approaches to Force Inference at this time: Video Force Microscopy (VFM) from Brodland *et al.* [1], Mechanical Inference (MI) from Chiou *et al.* [38], Bayesian Force Inference (BFI) from Ishihara *et al.* [39], [40], and Cellular Force Inference Toolkit (CellFIT) from Brodland *et al.* [41]. VFM, MI, and BFI all work on the same basic premise: a 2D polygonal representation of cell networks using 3-way cell junctions as nodes (and possibly additional points along the boundaries between cells). CellFIT, on the other hand, treats each cell-cell boundary as a single tensile element under a pressure differential; this can be approximated by a circular arc. Figure 2 from Brodland *et al.* 2014, shows the force balance that this implies [41].

All FI techniques have inherent numerical instabilities that make them uncertain for some types of measurements; for instance, the accuracy of the relative ratios of tensions or pressures between cells separated by many cells has yet to be established. Also, FI necessarily relies on simplifying assumptions about the tissue of interest, so it is up to the researcher to assess the accuracy of these at the moment. Future versions of FI are likely to exploit this property explicitly to help researchers discover information about the tissue anyway, such as which assumptions are violated and potential reasons why.

3.10. Conclusions

Force analysis on living embryos is a difficult problem, and each of the possible techniques has different strengths and weaknesses: magnetic tweezers can produce large, arbitrary directional forces capable of stalling large-scale developmental processes, but have a difficult time discerning localized forces because it is difficult to embed nanoparticles in specific locations. Laser ablation has incredible positional accuracy for force assessment, but it does not directly measure forces and is a one-off procedure not suitable for continuous or distributed measurement of forces. Optical tweezers provide a nice way to make small, localized perturbations to cells, but these are always small in magnitude and not suitable for certain types of measurements. Also, a number of the techniques mentioned here do not measure forces, but rather mechanical properties like elasticity or viscosity, and care must be taken in evaluating exactly what is being measured in each case.

Of all the techniques, FI (and CellFIT in particular) probably holds the most promise as a potential technique for widespread use, because it requires only imaging and no additional equipment. However, it is still relatively young and needs validation across more systems as well as further refinements and enhancements.

The most powerful technique in the future is likely to be FRET-based force measurement. It combines the in-plane nature of laser ablation studies with the distributed measurement ability of FI techniques. Still, it too is in its infancy and will require more testing; also, it relies on genetically modified animals, so it will not be as straightforward to use in new systems.

All the listed techniques can and have been combined with live imaging and with each other. For instance, Desprat *et al.* used magnetic tweezers to rescue an embryo

damaged by laser ablation [63]. FI can easily be combined with laser ablation; we fully expect laser ablation, optical tweezers, and FRET-based imaging to help inform and calibrate future FI techniques.

Chapter 4

4. Enabling user-guided segmentation and tracking of surface-labeled cells in time-lapse image sets of living tissues

Portions of this chapter were published in D.N. Mashburn, H.E. Lynch, X.Ma, and M.S. Hutson, *Cytometry Part A*, vol. 81A, no. 5, pp. 409–418, 2012 [2].

4.1. Introduction

In studying embryonic development and morphogenesis, one often needs to track the morphological changes of individual cells in living tissues. This requires the collection of time-lapse movies of labeled cells, segmentation of each image frame into individual cells, and tracking cell identity across frames. Collecting the image sets using confocal or multiphoton fluorescence is now routine [107], but image segmentation and cell tracking represent substantial analysis bottlenecks. A number of algorithms and tools have been proposed for automated and/or manual segmentation and tracking of surface-labeled cells ([108]–[118], reviewed in depth in [119], [120]), but few can segment and track tens to hundreds of close-packed cells over hundreds of image frames with an accuracy that correctly segments all cells distinguishable by the human eye. Automated methods, such as CellCognition and CellProfiler [108], [110], [111], [113], are fast but do not attain the needed accuracy. Manual interactive tools like TrakEM2 and ITK-SNAP can attain the needed accuracy but are prohibitively slow [115]–[118]. The only currently available tool that is both interactive and capable of segmenting and tracking packed cells in tissue is Packing Analyzer [109]; unfortunately it is still quite slow.

Our approach to solving the segmentation and tracking problem is based on the premise that users know their data best; they may be able to interpret and use image features that are not accounted for in any *a priori* algorithm design. Thus, we have

designed a system that combines a parameter-less and fast watershed algorithm with a suite of manual intervention tools that allows users with little to no specialized knowledge about image processing to efficiently segment images with near-perfect accuracy based on simple user interactions.

In general, the segmentation and tracking process can be broken into three steps: object identification, boundary generation, and object tracking (i.e., maintaining a consistent ID on the cell through time). Each step can be performed either manually or automatically. For example, object identification has been performed by manually clicking on an image to generate a "seed" for each cell [114]–[116] or by automatically finding such seeds using the minima of a Gaussian-filtered image or taking a threshold [110], [117], [118]. Boundary generation has been performed manually by drawing a perimeter [110], [115] or automatically via seed-based space-filling algorithms like a watershed or level set method [113], [117], [118]. Likewise, object tracking can be applied as a manual post-processing step or as an automated post-process technique, *e.g.*, using maximal overlap of segmented regions in different frames to map ID's from one frame to the next [111], [112], [120]. Object tracking has also been automated in-process by using the region centroids from one frame to generate seeds for space-filling algorithms applied to the next frame [113].

Our approach provides an initial automated guess at the seed positions (based on minima of a Gaussian-filtered image or the region centroids from segmentation of a previous frame) and uses an automated watershed algorithm to generate the region boundaries. Manual intervention comes through the use of in-process tools to add, delete, group and move seeds. As each change is made, the watershed algorithm automatically redraws the region boundaries. This process is repeated as necessary – automating the

tedious process of finding the exact boundary locations, but allowing user control of object identification, segmentation, and tracking to any level of user-desired accuracy. Here we show applications of this method to time-lapse image sets of morphogenesis and wound healing in *Drosophila* embryos.

4.2. Materials and Methods

4.2.1. Sample Preparation and Imaging

The primary strain of *Drosophila melanogaster* used in this study is ubi-DE-Cad-GFP (Kyoto *Drosophila* Genetic Resource Center), which ubiquitously expresses a cadherin-GFP fusion that labels epithelial cell junctions [121]. Fly embryos were dechorionated and prepared for live imaging as described previously [31]. Time-lapse image sets were captured on a Zeiss LSM410 laser-scanning confocal microscope (inverted) with a 40x 1.3 NA oil-immersion objective. Cellular ablations were performed with the 3rd harmonic (355 nm) of a Q-switched Nd:YAG laser (Continuum Minilite II, Santa Clara, CA) [31].

4.2.2. Segmentation Algorithms

Our segmentation and tracking system is based on a watershed approach. Initiation of watershed segmentation requires an initial set of starting pixels or seeds. Each seed has a unique identifier value that denotes the segmented region to which it will contribute. This allows multiple seeds for each region. The algorithm then fills the space by expanding the regions around each seed, starting with the darkest pixels first and slowly raising the "water" level. This process continues until the regions meet at boundaries and all pixels are assigned a value [122]–[124]. We chose a watershed approach for three reasons: 1) it does a fairly good and consistent job of determining cell boundaries based on the bright lines of our GFP-cadherin fluorescence images; 2) it has

the flexibility to use different numbers of seeds for each cell – one for most, but two or more for cells that are difficult to segment; and 3) it has no user-selectable parameters. This last point means that the user does not need previous image processing expertise to guide parameter selection.

To initialize the watershed segmentation and tracking procedure for an xyt (or xyz) image stack, we select seeds for the first xy -image based on the local minima after application of a Gaussian filter. This does require a user-selectable parameter – the width of the Gaussian kernel – but it is easy to manually vary this parameter until one obtains a reasonable compromise between under- and over-segmentation as in Figure 4.1 . A Gaussian filter is not used to identify seeds for subsequent images. Instead, once the user is satisfied with the segmentation of image frame j , the centroid of each cell in frame j is used as a seed for frame $j+1$ (using the same identifier value). This approach advantageously and automatically provides in-process cell tracking [113].

Although the automated parts of this process yield generally reasonable results, there are obvious instances of incorrect segmentation. In fact, our time-lapse image sets contain frequent situations (such as sudden motion, movement of cells into or out of the imaging plane, or unusually bright regions internal to a cell) in which it is difficult or perhaps impossible for any automatic algorithm to properly segment the cells (Figure 4.1). We have thus chosen to develop a general-purpose framework and a suite of tools that enable a user to make these difficult decisions efficiently and directly. Although improved algorithms can tune segmentation for specific cases, our approach should be more flexible and more generally applicable.

Our novel approach to a hybrid of manual and automatic segmentation is not to adjust the final cell boundaries, but to directly adjust the seeds themselves. By doing so,

we have a method that is robust, flexible, and easy to use. With a specified set of seeds, boundary generation by the watershed algorithm is fast and requires no user-selectable parameters, so incremental adjustment of the seeds and thus changes to the segmentation can be evaluated in real time. These features have also allowed us to create a simple save and load functionality that allows segmentation to be readjusted or completed at a later date. With this approach, if a user makes detailed changes to one section of an image, these changes will not have to be thrown out if coarse changes are later made in another section (as would be the case with some types of post-processing correction schemes based on manual image correction).

The seed manipulation tools we have developed are based on the ability to quickly add, delete, group and move seeds. There are one-click functions to "Add Seed", "Delete Seed" and expand regions with "Extra Seeds". This last feature allows the user to add secondary seeds for a cell, which the watershed algorithm then uses to expand that cell's boundaries. These extra seeds do not normally propagate in the tracking system, but normal tracking can be bypassed to copy all seeds (including extras) directly from the previous frame. This can be very useful with strangely shaped objects like thin curves or rings. There are also simple mouse/keyboard combinations to "Change Value" or "Swap Values" that change the identifier value associated with each cell. Finally, there is a "Lasso" tool that provides the ability to select individual seeds or groups of seeds (regardless of identifier value) which can then be moved or deleted *en masse*.

4.2.3. Technical Implementation

For the core watershed algorithm, we used the function `cwatershed` from the python package `Mahotas` [124], which is part of the larger `PythonVision` project for computer vision. This watershed function is an implementation of the standard ordered-

queue, heap-based watershed algorithm [122]. To facilitate GUI interactions, we developed a Python program using the packages WxPython and Matplotlib. We also made heavy use of numpy, scipy, and the Python Imaging Library. Ultra high performance functions were implemented in Cython. The complete program, known as SeedWater Segmenter (SWS), is available for download under a BSD license at Google Code (<http://code.google.com/p/seedwater/>).

Completely manual vector segmentation was performed using Inkscape (www.inkscape.org) to generate an SVG file and a custom Python program to extract the SVG/XML data and convert it to polygons for accurate geometric comparisons using the python package "shapely."

4.3. Results

Embryonic epithelial tissues in *Drosophila* are characterized by connected sheets of cells wrapped over a curved surface with yolk beneath. Two main cell types make up the tissues: epidermis cells and amnioserosa cells, each with very different average sizes (Figure 4.1 A). These tissues undergo a number of morphological changes during embryonic development including extreme cell shape changes, cell rearrangements, bulk tissue motion and cell death. In studying these changes and the forces underlying them, researchers often use laser microsurgery to ablate one or more cells, creating artificial (and sometimes very large) perturbations to the remaining cells [31], [73], [74]. Each of the above creates problems when segmenting and tracking cells in time-lapse images of living embryos. We have designed our segmentation and tracking software so that it provides the user with tools that can handle these difficulties. Below, we start with an image that has an initial set of automatically generated seeds and show that our manual correction tools are sufficient to correct common segmentation errors easily and

effectively.

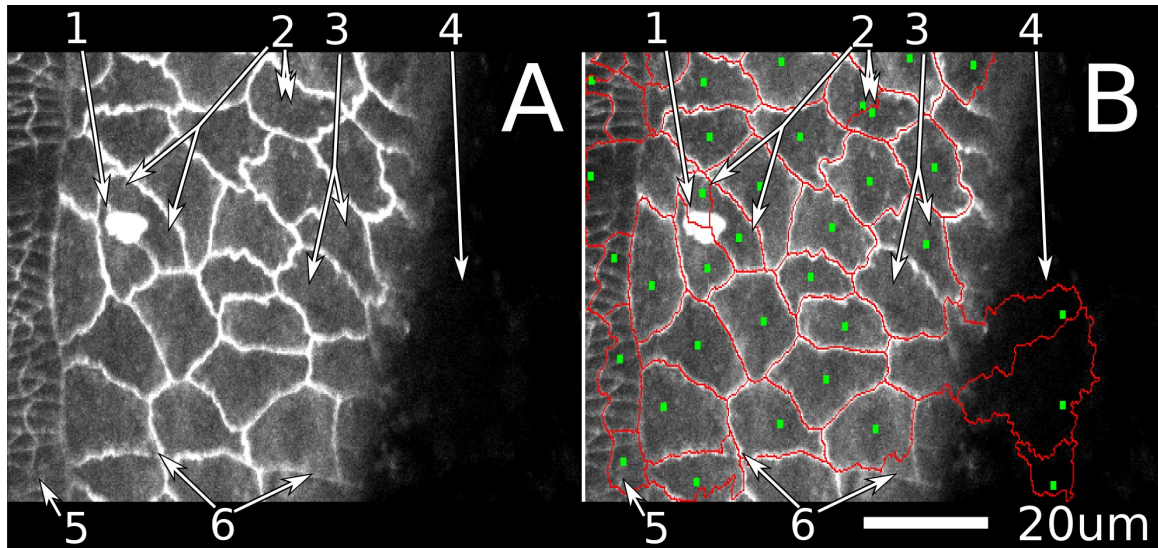


Figure 4.1: Common segmentation difficulties in confocal images of living, cadherin-GFP stained fruit fly embryos.

The large cells on the right of the image are amnioserosa cells; the smaller ones at the left are epidermal. (A) is an unsegmented image and (B) is the same image with an overlay of seeds (small green squares) generated automatically by application of a Gaussian filter ($\sigma = 2.5 \mu\text{m}$) and segment outlines (red lines) generated by a watershed algorithm. The numbered arrows point to several common errors in automatic segmentation. (1) An object obscures the view of the cell edge. (2) A single cell is divided between two seeds, i.e., oversegmentation. (3) Two cells share a single seed, i.e., undersegmentation. (4) A region that should be part of the image background instead receives seeds and is assigned as cells. (5) An area of epidermis cells that is very badly mis-segmented because the Gaussian filter is too large for these smaller cells. The user must decide if segmentation of this region should be completely reworked manually or skipped altogether. A smaller Gaussian filter ($\sigma = 0.625$ instead of 2.5) will effectively generate seeds for these smaller cells, but at the expense of severely oversegmenting the amnioserosa cells (into ~ 10 segments each, not shown). (6) Sub-cellular regions are misassigned. One can often determine which cells these regions belong to based on other images in the time-lapse set.

In the first round of manual intervention, we add and remove seeds to correctly identify the location of all the cells in the image, assisting the Gaussian technique manually. By right-clicking to delete seeds and left-clicking to add seeds, this process proceeds very quickly (Figure 4.2). Also, because of our centroid-based tracking system, adding and deleting entire cells becomes largely unnecessary after the first frame. Note

that the user must typically make some decisions regarding which partially-visible cells are worth segmenting. The tools for adding and deleting cells allow the user to make such decisions on the first frame and yet revisit these decisions later as cells enter or leave the viewing area.

Although add and delete are important tools in identifying which regions are cells, the user also needs the ability to decide which sub-regions belong to which cells (see Figure 4.1 , arrow #6). Advantageously, the watershed algorithm allows for more than one seed of each value to be present in a single cell. This "extra seed" feature gives us the ability to use manual seed intervention not only to identify each cell, but also to define which sub-regions belong in which cell. By placing an extra seed in a subregion, it is very easy to identify that sub-region as part of a particular cell (Figure 4.3 A-E).

This model works very well for most problem cases, but after segmenting a number of data sets, we realize that some boundaries simply "misbehave" unless extra seeds are placed essentially along the whole boundary. This can happen for boundaries that are particularly discontinuous, have a low signal-to-noise ratio or have unusual image topology (such as a gradient that makes one watershed region likely to invade another). To quickly handle such problem cases, we implemented a tool that inserts freehand lines of seeds. With this tool, the user can essentially draw the boundary in directly (Figure 3F-G).

This seed model is simple, intuitive and easy to teach to new users. It benefits from the consistency and speed of the watershed algorithm, yet still allows users to correct the segmentation as much as is needed.

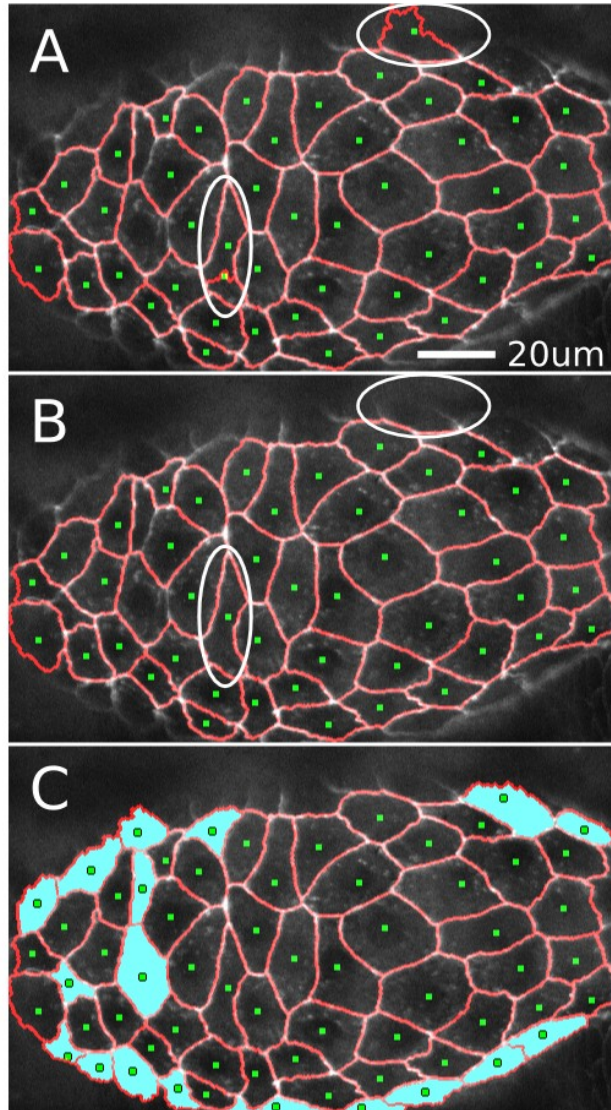


Figure 4.2: Adding and deleting seeds manually.
 (A) Initial automatic segmentation of an image ($\sigma = 2.5 \mu\text{m}$). Over-segmented regions with unwanted seeds are circled. The upper circled region highlights a cell at the edge of the imaging plane with a poorly defined boundary. The lower-left circled region has two seeds dividing a single cell. (B) Segmentation after manual removal of unwanted seeds. (C) Segmentation after manual addition of seeds to correct under-segmented regions (cyan fill). Seeds were added for sixteen cells around the margins of the tissue. These cells had been considered part of the background by the automatic algorithm. Seeds were also added for three internal cells that had not automatically received their own seeds.

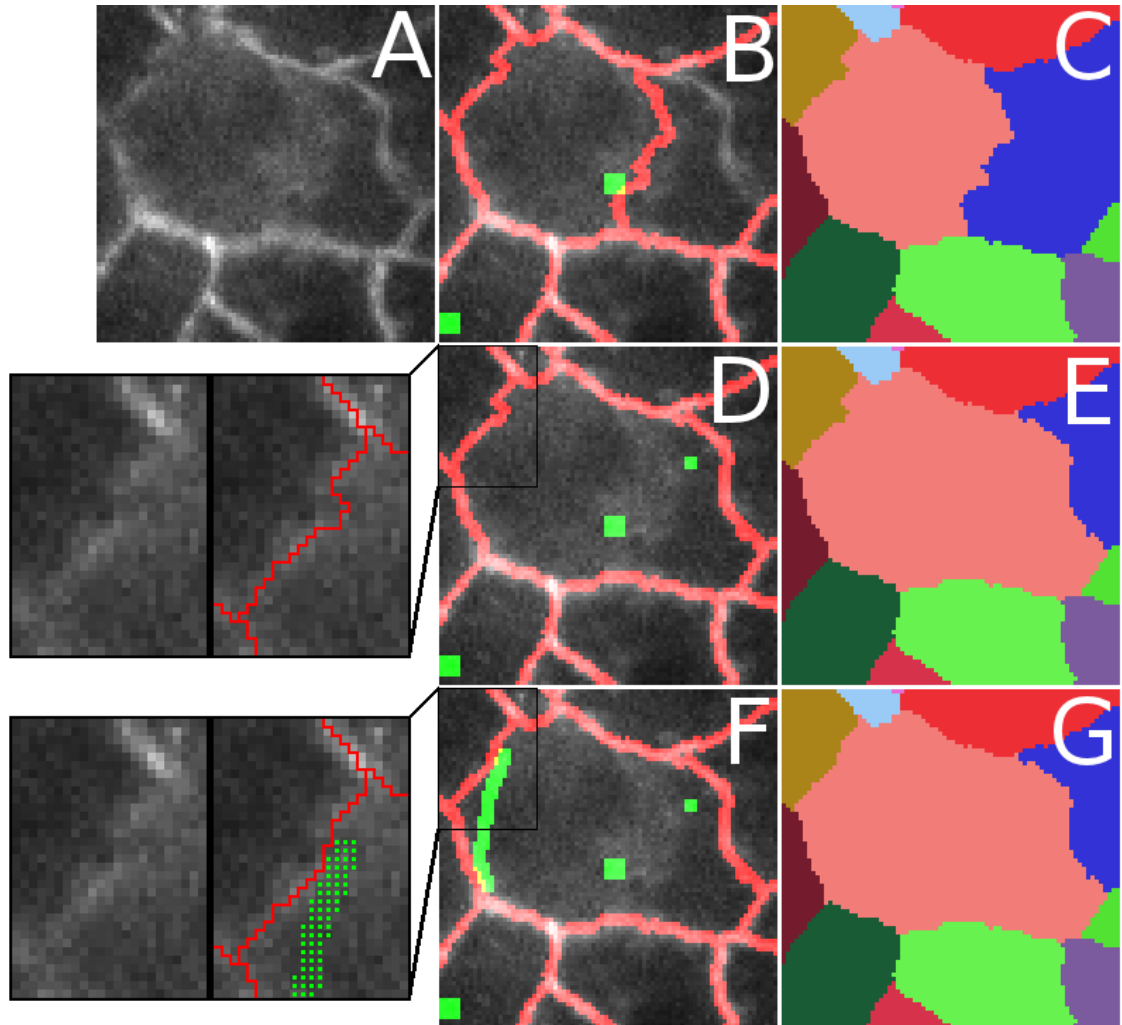


Figure 4.3: Adding multiple extra seeds to correct mis-segmentation of cellular subregions.

The middle column shows seeds (green) and outline overlays (red) on an original image; the right column shows a false-colored cell ID map. (A) Unsegmented image. (B, C) Initial segmentation. The subregion just to the right of the central seed is misassigned to an adjacent cell (blue). The upper left boundary of the central cell (pink) is also not satisfactory. (D, E) By adding a single extra seed, the originally misassigned subregion is reassigned to the appropriate cell (pink instead of blue). For the user, this is a two-click process: a left-click on the region that needs to be expanded followed by a right click to place the extra seed. A zoom in of the upper left corner is shown to the left of (D) with and without overlays. (F, G) The upper left boundary of the central cell (pink) is improved by adding a polyline of extra seeds (green). For the user, creating a line of seeds works as above except multiple right clicks are used. A line segment of seeds is added between each successive right-click location. The result is a slight, but clear improvement in the overlap of the watershed boundary and the imaged cell-cell boundary. A zoom in of the upper left corner is shown to the left of (F) with and without overlays. This more clearly shows the subtle change in the boundary. The precise location was determined by visually tracking its motion in previous and subsequent frames of the image stack (not shown).

We also tried several preprocessing filters including denoise, Gaussian blur, unsharp mask, and CLAHE [35] to see if we could reduce the subsequent need for manual intervention. Filtered and unfiltered image sets generally yield a similar number of errors in cell identification, but filtered images tend to yield artifacts during boundary generation. The one exception was CLAHE filtering, which enhances poorly lit regions for easier viewing and watershed segmentation without compromising the shapes of boundaries that already segmented well. We thus determined that it was preferable to segment unfiltered or CLAHE-filtered images – giving the truest possible representation of cell morphology.

4.3.1. Tracking

Once all cells have been identified and the segmentation has been adjusted to associate all subregions with the correct cell, the next step is to generate seeds for the next frame. Rather than finding these seeds using the same Gaussian minima approach as the first frame, we generate subsequent seeds from the centroids of each cell in the previous frame (Figure 4.4). This has the added advantage of automatic tracking because the seed value will be the same in the second frame as in the first.

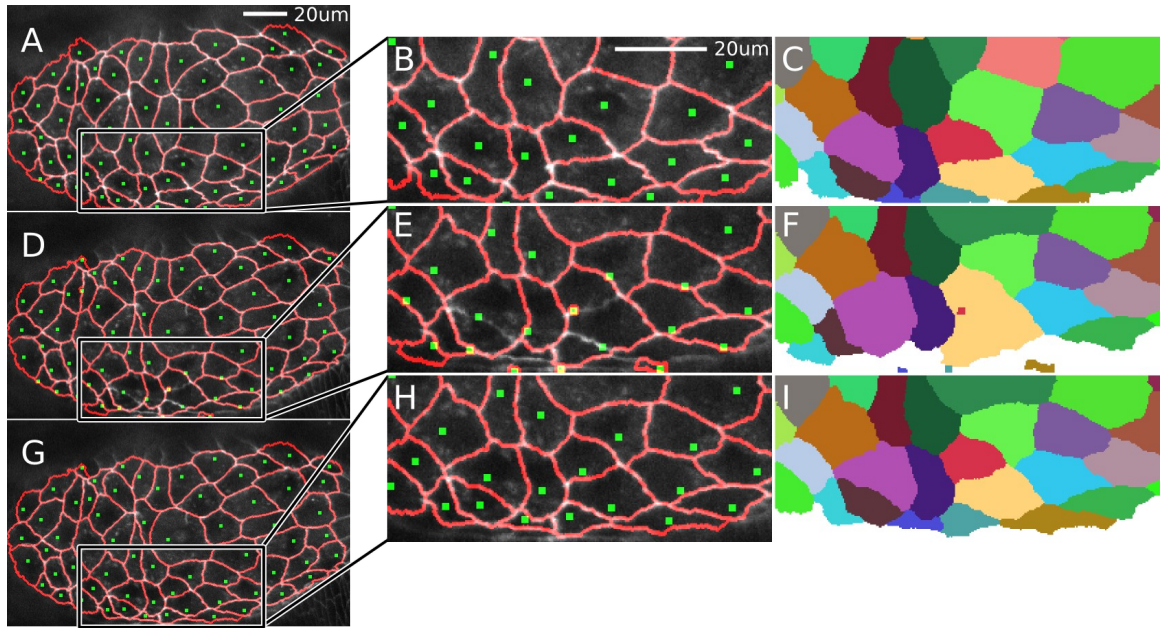


Figure 4.4: Cell tracking when the frame-to-frame movements of cells are large. (A-C) Complete manually assisted segmentation of a cluster of amnioserosa cells. The segmentation overlay shows seeds (green squares) and segment outlines (red lines). (B) is a close-up of the boxed region and (C) is the corresponding false-colored cell ID map. (D-F) Automatic tracking and segmentation of the next frame after laser ablation of a central cell and with a large time interval between frames (70 seconds). The large time interval exaggerates cell motion between frames and causes the centroid-based algorithm to track cells improperly in some regions, especially near the bottom middle of the image (zoomed region in E, F). The errors are clearly discernable in (F) compared to (C). Note that even in this relatively extreme case, the automatic tracking performs very well for most cells in the image, particularly outside the boxed region. Tracking generally works well unless the cell moves over half its diameter. (G-I) Corrected tracking and segmentation after using the "Lasso" and "Move Seeds" tools. The "Lasso" tool works by clicking to form a polygon that encircles multiple seeds. These seeds are then moved using the arrow keys to position them properly. This seed adjustment process is quite fast (a few seconds) by starting with bulk motion and then adjusting individual seeds as needed.

In rare cases, a cell may move so much from one frame to the next that the previous centroid no longer falls within the cell's next boundary. This is easy to correct by simply selecting the seed point and moving it to a more appropriate location. In fact, if there is bulk motion, seeds can be "lassoed" and moved in bulk (Figure 4.4). In addition, if two seeds switched cells, their identifier values can be swapped with "Swap Values";

and if a seed was assigned an incorrect ID (which can happen if it is deleted and re-inserted) then "Change Value" can directly change the assigned ID. In practice, these tracking problems occur infrequently and are relatively simple to fix.

4.3.2. Robustness to Noise

To analyze the resilience of our watershed algorithm to additive and multiplicative noise, we segmented an artificial image of two "cells" separated by a single, central bright line (100 pixels long, 1 pixel wide, and with a brightness S above the background). Automatic watershed segmentation is largely unaffected by noise up to S , begins to fail more often than it succeeds when the noise is $2S$ (signal-to-noise ratio or SNR = 0.5), and almost always fails for SNR = 0.25. At this latter level, a Gaussian filter can rescue about 40% of the segmentations. These results hold for both additive and multiplicative noise. Despite the automatic segmentation failures, users can distinguish the appropriate boundary down to SNR ~ 0.25 . Thus, the SNR-range of 0.25-1 is the real "sweet spot" for a semi-automatic routine like SWS, especially if the noise is spatially varied within this range. We find similar results when the central dividing line has a Gaussian profile (with $\Delta = 1$ pixel).

4.3.3. Speed

We chose a watershed algorithm because it is robust and extremely fast, lending itself to highly responsive interactivity. Upon a change in the seeds, our segmentation program must process a GUI event, update the seed data, re-run the watershed algorithm, and redraw the frame. Nevertheless, all of this is completed in approximately half a second on a typical desktop PC. This short lag means that the user can interactively perform hundreds of operations in a matter of a few of minutes.

To measure the speed and impact of manual intervention on a real image set, we

timed how long it took to segment a typical 190-frame *xyt* stack with approximately 64 cells per frame (62 cells appeared in all frames; 3 cells appeared in just a subset) (Figure 5A-B). First, to illustrate how quickly SWS converges on a solution, we performed minimal tracking and generally let the algorithm do the work automatically. During this "minimal tracking" phase, we inspected the intermediate segmentations, but only to ensure that each seed stayed within the proper region to maintain ID consistency through time; we ignored any problems with borders and sub-regions. This process took 50 minutes (about 16 s per frame). The initial segmentation was then improved by three more rounds of manual intervention, completing segmentation to user-desired accuracy in approximately 6 hours (under 2 minutes per frame). Figure 4.5 A-B shows how the overall user-defined segmentation accuracy improved with time, albeit with diminishing returns. After the first pass segmentation, manual intervention changed the cellular assignment of only 1.4% of the pixels in the entire image set, but in so doing, it changed the borders of 12% of the cells. As shown in Figure 4.5 C, the errors were distributed very non-uniformly over the segmented cell population, with just a few cells having very large errors. We found that the most time consuming part of manual intervention was simply inspecting the current segmentation to see if it was satisfactory. When errors were identified, the corrections were implemented very quickly.

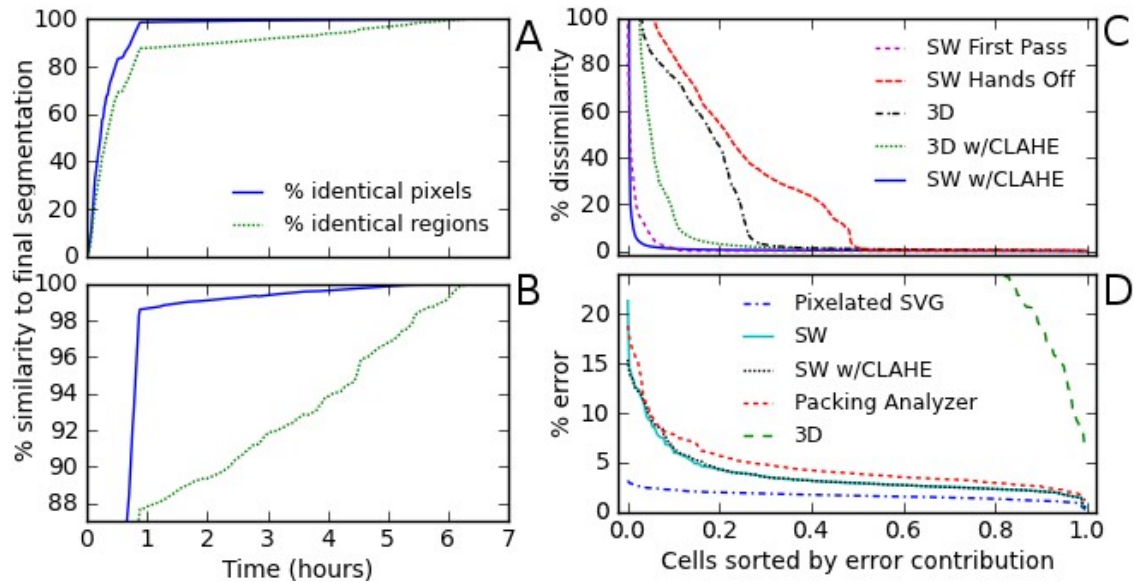


Figure 4.5: Comparison of segmentation speed and accuracy for a typical data set: 190 frames with an average of 64 cells per frame.

(A, B) Improved accuracy versus time spent on manual intervention using SWS. Both graphs represent the same data; (B) simply has a tighter zoom in the y-axis to more clearly show the data after 50 minutes. Intermediate segmentations were saved after each change in watershed borders or at least every 60 s. The accuracy at each intermediate time point was assessed based on either the percentage of pixels whose assignment matched the final segmentation or the percentage of cells whose boundaries matched the final segmentation. The "first pass" segmentation was performed with minimal tracking, generally letting the algorithms do the work automatically (50 min), achieving a pixel-based accuracy of 98.6% and a cell-based accuracy of 88%. We then performed three more rounds of manual intervention and adjustment that improved the segmentation and tracking to user-desired accuracy in approximately six hours (~ 2 min per frame). The efficacy of manual intervention will vary with user experience and imaging quality, but this set is representative. The diminishing returns of continued manual intervention are most evident in the pixel-based comparison, but even this is somewhat linear over large time periods because we made successive passes through the entire stack. (C) Distribution of errors over all segmented cells for selected intermediate segmentations and other techniques. Errors are defined as deviations from the final SWS segmentation. The x-axis is a normalized list of cell indices sorted from largest to smallest relative error. The y-axis is the number of erroneous pixels for each cell divided by the average area of all cells. We compared a "Hands Off" SWS segmentation with no user intervention and a "First Pass" SWS segmentation with minimal manual tracking assistance. For comparison, we include a 3D watershed segmentation, and SWS on an CLAHE-filtered image set. (D) Distribution of errors over all segmented cells in comparison to a "gold standard" manual segmentation using a vector editor. These comparisons are limited to five evenly spaced frames (every 47th of the full data set). Absolute errors are thus compared for SWS segmentation, Packing Analyzer, and 3D watershed segmentation. As a baseline, we include errors induced by pixelating the vector segmentation.

4.3.4. Segmentation Quality

To place the performance of SWS in context, we compare its final segmentation to several automatic segmentation and tracking methods (Figure 5C). First, we compare to a completely automatic watershed segmentation and tracking (SWS in a "hands off" mode). This process took less than four minutes and correctly segmented 45% of the cells, but the rest were fairly bad. In fact, over 20% were half wrong or worse and about 6% were essentially all wrong, i.e., completely outside their proper boundary. This highlights the improvement achieved simply by supervising the tracking in our "first pass" segmentation above. Second, we compare to a 3D watershed algorithm. This process used only the initial seeds in the first frame and segmented the entire image set in only 90 s. It was much more effective than "hands off" SWS, but not nearly as accurate as semi-automatic tracking. With a 3D watershed, almost one fourth of cells had an error >15%. This poorly segmenting subset could be reduced to one tenth of the cells using CLAHE pre-filtering. This is just shy of the performance of semi-automatic tracking. As a final comparison, we pre-filtered with CLAHE and used the exact same seeds as in the final SWS set. Although we expected this pre-filter to slightly improve edge delineation, we instead found that the small changes introduced by CLAHE pre-filtering did not always work well with our manually corrected seeds. This caused a small number of regions to fill improperly, leading to large errors in a few cells (Figure 4.5 C).

To get an absolute measure of SWS performance, we created a "gold standard" based on an expert's manual segmentation using a vector editor. This manual segmentation took over an hour per frame. We thus restrict the gold standard comparison to just five image frames (every 47th from the 190-frame set).

As a baseline, simply pixelating the vector segmentation leads to a fairly uniform error distribution with a per-cell average of 1.7%; any pixel-based segmentation of this image set will have at least this minimum error. In comparison, the final SWS segmentation had an average error of 3.7%. A handful of cells had 10-20% errors, but the overwhelming majority had errors just below the average (Figure 4.5 D). Close inspection revealed that these errors were largely just ambiguity in the exact boundary location. We also tried pre-filtering with CLAHE and readjusting the seeds to correct some of the CLAHE-induced errors, but this eventually yielded results virtually identical to the unfiltered SWS version. We also compared the performance of another semi-automatic routine, Packing Analyzer (3), but did not attempt to use its tracking system. This process took about five min per frame and yielded results similar to the SWS segmentation (4.7% error). Finally, we performed 3D watershed segmentation on this five-frame subset. Although 3D watershed performed surprisingly well on the full data set, it performed very poorly in this subset: 7 of every 10 cells had errors > 30%. This breakdown is expected because the cell boundaries now move substantially between frames (much more than the typical boundary width). A summary of our results on the accuracy and speed of various segmentation methods is presented in Table 4.1 .

Comparison to SWS (190 frames)	Approx. Time Per Frame	Pixel-based %Dissimilarity	%Cells with Dissimilarity >5%	%Identical Cells
Final SWS (reference)	110s	-	-	-
First Pass SWS	16s	1.40%	5.60%	88.00%
“HandsOff” SWS	1.25s	28.10%	48.50%	45.00%
3D Watershed	0.5s	19.90%	27.10%	1.40%
3D w/CLAHE	0.5s	9.70%	14.70%	1.00%
SWS w/CLAHE uncorrected	110s	1.30%	2.10%	6.80%

Comparison to manual Vector segmentation (5 frames)	Approx. Time Per Frame	Pixel-based %Dissimilarity	%Cells with Dissimilarity >5%
Vector Segmentation	2hours	-	-
Pixelation of VectorSegment.	-	1.70%	0.00%
Final SWS	110s	3.70%	14.30%
SWS w/CLAHE corrected	110s	3.80%	16.30%
Packing Analyzer	300s	4.70%	26.60%
3D Watershed	0.5s	65.00%	100.00%

Table 4.1: Speed and accuracy of various segmentation methods.

The first six rows compare to the final SWS segmentation and include all 190 frames of the image stack, as in Fig. 5A-C. The final six rows compare to a manual vector segmentation (gold standard) and include only five representative frames (#1, 48, 95, 142, 189), as in Fig. 5D.

4.3.5. Measuring Cell Shape Oscillations

As a demonstration of SWS performance, we use it here to investigate cell shape oscillations in the amnioserosa [13]–[15], [125]. Prior work has shown that these cells undergo cyclical changes in apical area with a period of 230 ± 76 s [13]. Adjacent cells have a tendency to oscillate out of phase, but they can also quickly switch between in-phase and out-of-phase oscillations. When the cells are visualized with a cell-boundary marker like GFP-armadillo (a β -catenin like protein) and viewed as oscillations in apical cell area, there is no evidence for long-range, wave-like propagation [13]. On the other hand, when these same cells are visualized with a different fluorescent marker – GFP-moesin, which highlights concentrations of filamentous actin [73] – one can clearly

discern wave-like propagation of f-actin accumulations that are correlated with sub-cellular contractions [31]. To address this discrepancy, we performed SWS segmentation of a time-lapse image set of GFP-cadherin-labeled amnioserosa cells (66 cells over 63 image frames spanning 1300 s). We then calculated the autocorrelation functions of both cell area and triple-junction velocity (triple junctions were defined as points that touch three or more cells). Both functions showed clear oscillatory behavior; first minima at 144 s for cell area and 123 s for triple-junction velocity; subsequent maxima at 267 s in both cases. We then used the triple-junction velocities to calculate a time-and-space pair correlation function (Figure 4.6). The $\Delta=0$ line of this function is the velocity autocorrelation and clearly shows the peaks described above. In addition, a density plot of the pair correlation function shows that the extrema move to longer time delays τ at larger spatial separations Δ , i.e., the correlations and anti-correlations propagate. Their propagation speed of 0.14 $\mu\text{m/s}$ is close to that observed previously for apical accumulations of f-actin, 0.2 $\mu\text{m/s}$ [31]. The propagation is however limited and decays within 30 μm (about one and a half cells). Thus, one can observe contraction waves in the amnioserosa using just cell boundary labeling, if one uses a sub-cellular analysis based on triple-junction velocities and one has a complete and accurate segmentation that allows averaging over a large number of triple-junction pairs. The apparent discrepancy in previous observations results from whether one chooses to monitor the oscillations with cellular or sub-cellular resolution.

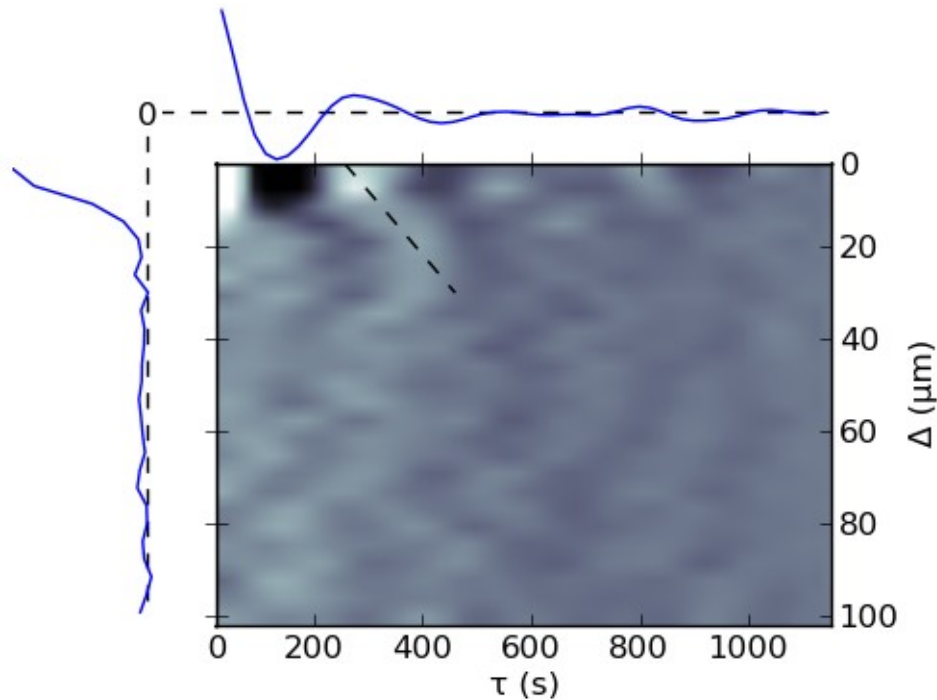


Figure 4.6: Time-and-space pair correlation function of triple-junction velocities for a data set with 66 segmented cells.

The x-axis is the time separation between points (τ) and the y-axis is the distance between points (Δ). Correlations are normalized so that the peak at (0,0) has a value of 1. The $\Delta=0$ axis (autocorrelation function) is plotted above the density plot, and the $\tau=0$ axis (distance correlation) is plotted to the left. Dashed lines in these two plots represent zero correlation. The minima and maxima of the autocorrelation appear as dark and bright spots on the $\Delta=0$ axis of the density plot with the first of each occurring at 123 and 267 s, respectively. In the full pair correlation function (density plot), the extrema move to longer time delays as the distance between the pair increases. This wave-like propagation is demarcated by the angled dashed line, which has a slope and thus velocity of $0.14 \mu\text{m/s}$.

4.4. Discussion

Image segmentation and the quantitative measurement of cell morphology are invaluable in the attempt to link physical and biological processes [5], [126].

Segmentation is a very wide field and a large number of approaches have been previously implemented. These range from direct drawing of segmentation boundaries to automatic registration of an entire image stack [119]. Before designing our own segmentation tool,

we examined several similar tools including ITK-SNAP [118], FIJI's TrakEM2 [115], CellProfiler [110], and Packing Analyzer [109]; none of these was able to segment our typical (and thus imperfect) time-lapse image sequences with the speed and accuracy we desired. ITK-SNAP, FIJI's TrakEM2, and the more recent V3D [127] have tools for direct 3D segmentation of structures. These work very well for the smooth and continuous boundaries of 3D xyz structures in MRI images but have difficulty with some of our cells as 3D xyt structures. These 3D tools either lack the ability to reliably generate non-overlapping regions (because they are focused on single anatomical structures) or provide insufficient manual adjustments. The other tools perform 2D segmentation that can then be repeatedly applied to a sequence of images. Some focus on high segmentation accuracy for a small number of cells (*e.g.*, fast marching tool and pen tool of FIJI's TrackEM2 [115] or the interactive tools of Intelligent Scissors [128]–[130]); others focus on speed and sacrifice some accuracy for high-throughput analysis of large numbers of cells (CellProfiler [110]). The tool designed to handle image segmentation tasks most similar to ours is Packing Analyzer. It is also a hybrid manual/automatic system [109], but is parameter-based and implements post-process correction and tracking. Packing Analyzer works well with many of our best data sets, but is not as efficient nor as flexible as needed for our more difficult data sets.

SWS fills a niche as a user tool for extremely accurate tracking of cells in living tissues when image quality suffers sporadic, but typical problems. By using a simple seeded watershed approach, it gives users a readily understandable way to manipulate segmentation results, save their progress, and even redo and undo changes. Furthermore, because it is a parameter-less system, users with no prior experience in image segmentation can get up to speed in minutes with very little instruction. SWS's approach

is based on a simple and useful paradigm – manual segmentation correction using direct seed manipulation. It is easy to use and understand, but also powerful and efficient.

SWS is presently a stand-alone tool, but it also has potential as a post-processing engine. For instance, one could use an alternative segmentation to generate seed points, load these seeds to initialize SWS, and then use its manual correction tools to improve the quality of the final segmentation. In this way, SWS can also be a powerful tool for "rescuing" imperfect segmentations of existing image sets. The key concept could also be integrated into existing tools like ImageJ, ITK, or V3D, combining efficient manual correction with powerful visualization and analysis tools.

We have considered creating a fully 3D version of SWS using a 3D watershed algorithm, but interactivity would suffer. Compared to 2D, a 3D watershed segmentation is slow, purely based on a linear increase in computation time with the number of frames to be segmented. This could potentially be alleviated on a massively parallel architecture. There are also artifacts created in 3D watersheds of time-lapse images: some cells fall into two different xyt watersheds, one giving way to the other in time; other xyt watersheds split into parallel time branches, creating separated regions with a single ID.

Other possible future improvements include tools to aid in visualization and identification of segmentation errors. Decision-making is often the most time-consuming part of segmentation, so it could be beneficial to visually flag cells that have large shape changes, little overlap with the previous frame, or discontinuities in the paths of centroids or triple-junctions. One can imagine providing a library of user-selectable (and user-expandable) routines for flagging suspicious segmentations. We see this as the direction in which segmentation efficiency can be most rapidly improved.

Chapter 5

5. Interactive 4D (*xyzt*) segmentation of epithelial cells in time-lapse confocal image stacks

5.1. Introduction

Segmentation is in general a hard problem, which is also very domain-specific. Automatic segmentation methods can be extremely useful and quite effective for statistical purposes, but always have corner cases that fail to properly segment (faint edges, noisy images, etc.). For some systems, these corner cases can even come to dominate the final segmentation. When studying the details of cell shape and motion, this can be unacceptable. In our previous paper, we described a system called SeedWater Segmenter (SWS) designed to facilitate manual corrections to automatic segmentation in a way that minimizes user time and effort [2]. Its major innovation was allowing a user to place multiple watershed seeds for each region – enabling a user to improve a coarse initial segmentation to a highly accurate one simply by adding and deleting seed pixels.

The basic concept of seeded watershed extrapolates quite easily from 2D images to 3D image stacks and has been used on numerous occasions before [116], [131]–[133]; however, to our knowledge, there is not an existing solution for highly connected tissues in 3D space that allows quickly defining multiple distributed regions, each with multiple seeds. In reevaluating the user interactions needed for 3D+time, we decided a number of things:

1. Viewing had to be simple, dynamic, and fast. Since volumetric rendering was too slow and difficult to quickly comprehend, we settled on displaying three perpendicular cut planes that could be traversed quickly and easily.
2. Showing outlines for every cell on all cut planes was distracting and too

computationally intensive, and mono-color seeds (green in SWS) failed to convey the necessary separation of regions. We thus chose to color seeds with the same color map as the final segmentation, lending a rough idea of regional separation. This is reasonable to interpret, especially in combination with the outlines for a single cell on the three cut planes.

3. Although region centroids act as a good predictor for future cell locations in fast 2D image sets, 3D sets often require so much elapsed time that cells can shift significantly between time points. Furthermore, automated tracking (such as is found in SWS) adds very little time savings overall given the amount of data each cell represents in 3D; editing time vastly outweighs simple tracking. For these reasons, the frame-by-frame nature of SWS was too restricting in 3D where one rarely wants to segment every cell but does need to view and segment a few cells through time.
4. Watershed in 3D is too slow to run after every user interaction. Instead, the user should be able to make a few modifications before choosing to update the segmentation manually.

This all led us to a remarkably simple paradigm: drawing colored lines and planes anywhere in 4D space. In this system, tracking is achieved by simply drawing a 4D line through the center of a given cell. Given linear motion, a line beginning at $t=0$ and ending at $t=t_{\text{final}}$ requires only 2 clicks to achieve instant tracking, and more complex motions can be easily broken down into piecewise linear steps.

We have used this software to accurately segment and track dozens of *Drosophila* cells in 3D through time, and have used it to measure temporal changes in cell shape and

volume [3].

5.2. Implementation Details

The system we have implemented, called SWS4D, gives a user the ability to view cut planes, define seeds, and see watershed results. Figure 5.1 shows the core of this interface, using sample data from a time series z-stack of cells in tissue (4D/ $xyzt$). This data is displayed using a "tri-panel" view (left side) showing three perpendicular cut planes (xy , xz , and yz) passing through a single point in $xyzt$ space; the coordinates of this reference point are shown by four slider bars (below) as well as a red cursor at that location in xyz space. The largest panel of the three shows the traditional 2D xy image. The other panels show the perpendicular slices as viewed from the side but "folded" down alongside the xy pane; in other words, the x or y axes thus stay aligned with the main panel. This left view just discussed above is referred to as the "data view"; seeds have been drawn in by hand on this view (pink and purple). The resulting segmentation is shown on the right in the "watershed view" using the same cut planes.

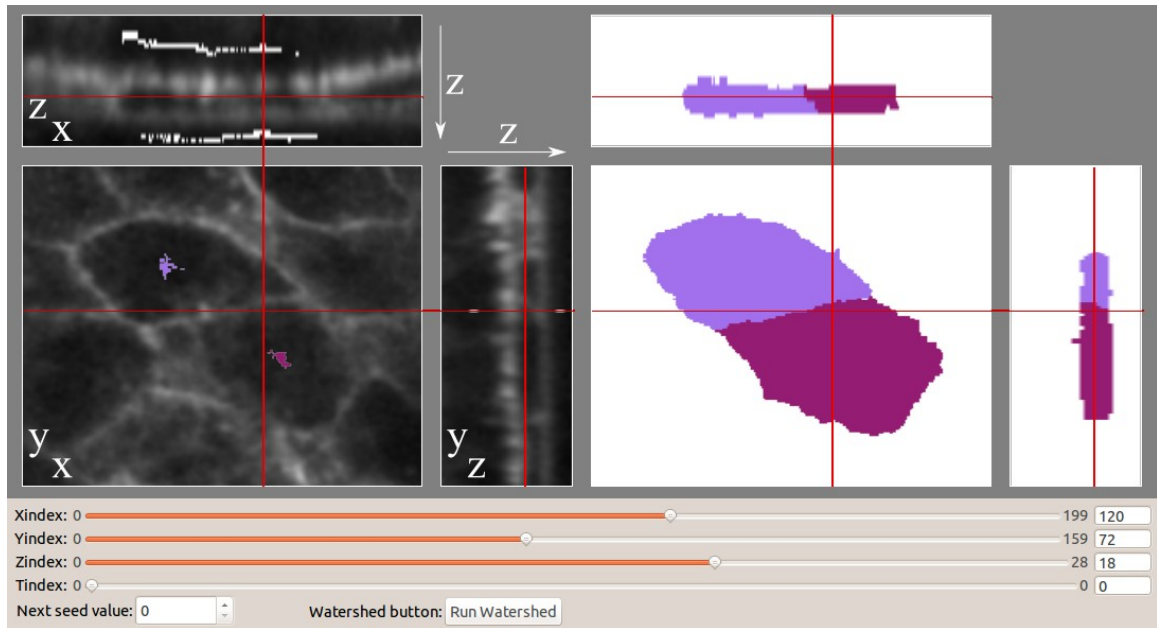


Figure 5.1: Core user interface of SWS4D.

Sample $xyzt$ data, a time series z -stack of cells in tissue, is displayed using the SWS4D software. Two cells have been colored in on the xy panel of the data view (left) and the resulting segmentation (generated by running the watershed algorithm) are shown on the watershed view (right). The core controls are shown at the bottom: slider bars to select the cut planes, a text box to control the ID (and color) of the seeds to draw, and a button to run the watershed algorithm and display the segmentation results. The axes are also labeled here for clarity (in white).

Seeds are added by selecting an ID number from the lower box and clicking on any of the three panels (xy , xz , or yz). Multiple seeds can be created by clicking in multiple places or by dragging, making this analogous to a pencil tool in traditional drawing software; this simplest form of interaction is referred to as "doodling". This mode also supports pixel-wise erasing by setting the current seed ID to zero (pixels without a seed ID have a value of 0 and are not displayed). Similarly, the ID of 1 is reserved for background regions and is always displayed in white; however, the background is actually equivalent to any other regions sans this convention. Both of these conventions are the same as those used in the original SWS (as are the colors associated

with a given cell ID).

Despite this already being a complete system for drawing seeds (and its effectiveness in this particular case), click-and-drag drawing is limited to a single perpendicular plane of the stack and is also very tedious. To overcome this, we implemented a tool for drawing poly-lines in 4D space, connecting the currently clicked point to the last clicked point with a series of voxels. This not only makes it easier to reason about the resulting connectivity, but also yields automatic temporal tracking via lines drawn through time as well as space. These lines are implemented using an N-dimensional (ND) variation on existing 2D and 3D Bresenham line rasterization algorithms [134]–[138].

Although this line tool allows for effective demarcation of cells and sub-cellular regions, some features are still insidiously hard to segment because the 3D watershed algorithm can easily "flow" around such lines. To properly segment difficult cell-cell interfaces in 4D, one needs a way to make "dams" of planar elements. We thus implemented tools for drawing triangular plane sections based on further extrapolation of our ND Bresenham line algorithm. Just as lines can be drawn either straight or oblique in any dimension, triangles can accurately fill between any arbitrary three points; this novel "ND Bresenham Triangle" algorithm is described in detail in the "Algorithms and Mathematics" section below. Figure 5.2 summarizes the basic line and plane seed drawing tools available in SWS4D.

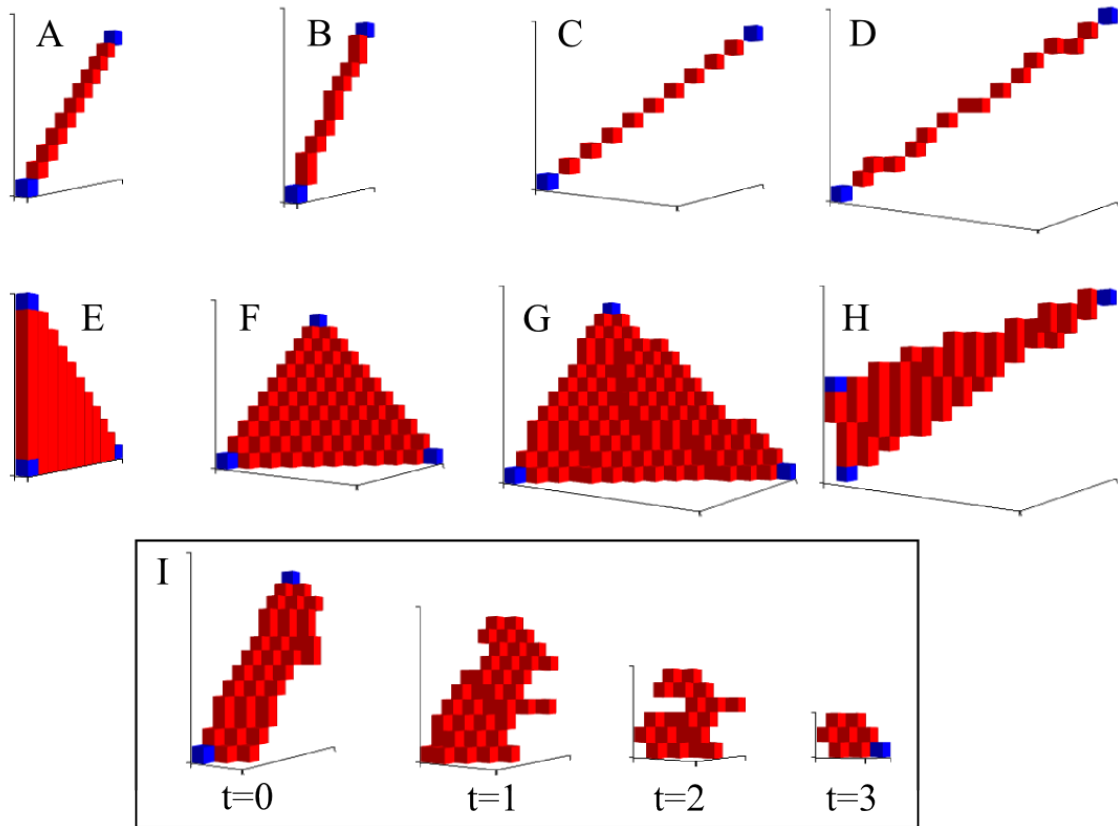


Figure 5.2: Sample Lines and Triangles using the ND Bresenham algorithms. All endpoints are plotted in blue and voxels generated by the algorithm are shown in red. (A) Simple 45 degree diagonal line in two dimensions. (B) Example of a line drawn with the classic 2D Bresenham line algorithm. (C) Simple voxel line with 45 degree azimuth and elevation. (D) Example of a line drawn with the ND Bresenham line algorithm in 3D. Notice how this line forms a minimal connection between the endpoints, just as in the 2D case. (E) Sample 2D triangle rasterized with the classic 2D triangle graphics algorithm which fills in all the pixels between three Bresenham lines. (F) Simple plane with $(1,1,1)$ normal. (G,H) Example triangles rasterized with the ND Bresenham triangle algorithm in 3D. (I) Example of an ND triangle in 4D. The four panels represent $t=0$ to $t=3$ where t is the 4th dimension in this system. Notice how these 3D components fit back together into a solid triangle if this is viewed as a single 3D projection (the same as G).

The triangle algorithm can easily rasterize triangles in higher dimensions as well, but 4D is the maximum needed for our use case. Although triangles cutting obliquely through all four dimensions of a time-lapse z-stack are rarely useful, simpler patterns, like cutting through z and t values at a fixed x - y location can quickly mark the same x - y

point across a large number of times and z planes with a single triangle!

In addition, just as poly-lines can be chained together using the last end point as the next starting point, triangular sections can be chained together using the previous two points and an additional point to make a new triangle. After some practice, this simple feature can be used to draw arbitrary filled 2D polygons and even to define complex 3D surfaces (*e.g.*, ellipsoidal sections). The key to defining a strip is to alternate between either side while advancing forward, building up the strip in a truss-like pattern. Figure 5.3 shows an example usage of this rasterizing technique for a simple 2D strip and a polygonal shape. 3D manifolds can also be generated fairly quickly using this technique by creating small, adjoining strips. Although this process is reasonably effective given how simple this interaction is, the process could definitely be further improved with a dedicated tool for this purpose. This is not usually necessary for small cells, but is important for segmenting more extended objects and will be discussed again below.

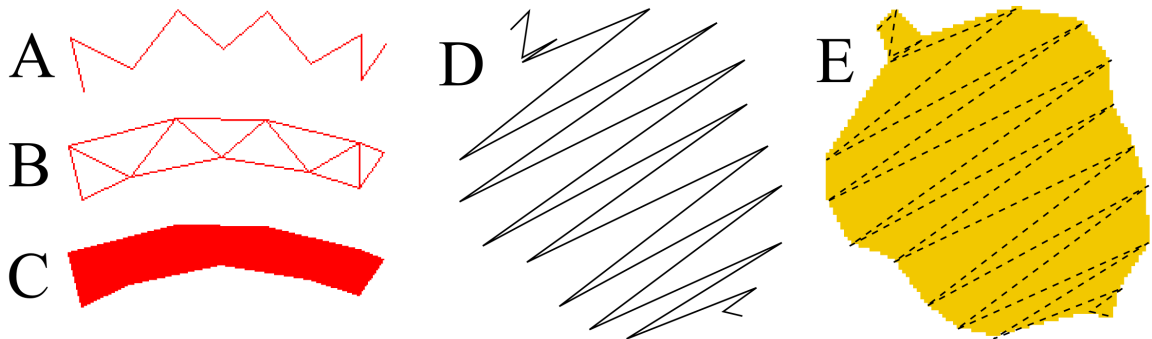


Figure 5.3: Bresenham triangles can be patterned into any curved strip. (A-C) show the original click pattern, all the reconstructed lines, and the resulting filled strip. (D and E) show the click pattern and resulting filled shape for a complex shape (an acorn in this case). In this way, moving back and forth can rasterize any arbitrary shape. Also, by moving around in 3D (or 4D) space, even more complex manifolds can be created in this same fashion.

With the addition of "line" and "triangle" mouse interactions, SWS4D is much

more effective for complex cell segmentation tasks; "triangle" is easily the most useful of the three seed drawing tools and can do anything that "doodle" or "line" can do just as easily. There are also non-drawing interactions called "print" and "move" for inspection and navigation respectively. Clicking in print mode displays that location's 3D coordinates and image brightness (or region ID for the watershed view on the right). Clicking anywhere in move mode changes the cursor position to that location, thereby affecting the views of the two opposing cut planes; as with the slider bars, moving the cursor on the data view also moves the cursor on the watershed view and vice-versa. Figure 5.4 shows the full SWS4D interface including a more completely segmented data set.

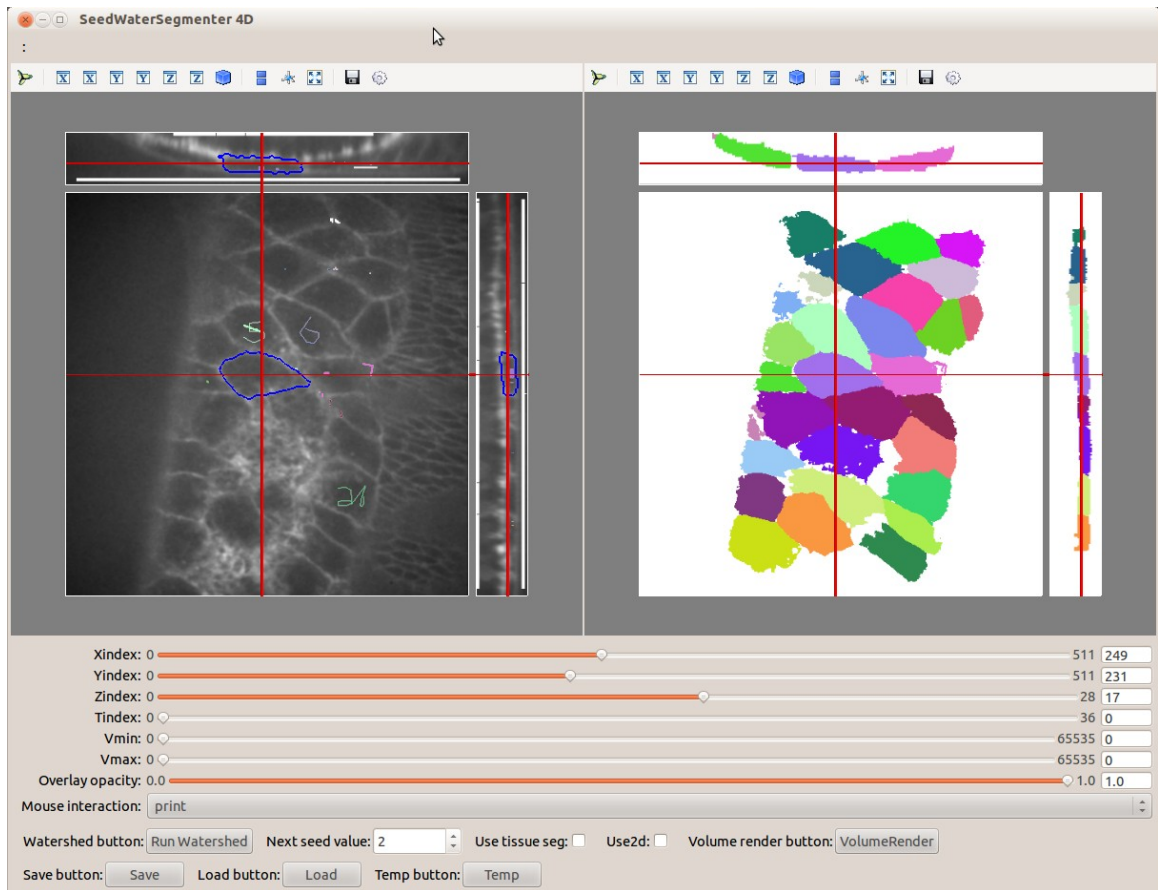


Figure 5.4: Full SWS4D user interface.

In addition to previously described features from Figure 5.1, this also showcases the blue outlines on the three cut planes for the current watershed region, the "Mouse Interaction" selection box to define click actions (doodle, line, plane, print, or move), slider bars to control the relative brightness of the raw data and overlays, options for alternate segmentation modes (described below), and buttons to save and load seeds and segmentation results. This also shows what a more completed segmentation looks like in SWS4D.

Note how on the data view (left) the current watershed region is outlined in blue on all three cut planes. This aids in identifying existing regions to add further seeds (and conversely helps identify regions that have not been initialized yet); this is especially important for identifying contentious subregions that have been misidentified by a coarse segmentation.

Our biological system of interest, the *Drosophila* strain, Resille 117, gives

relatively poor definition of cells' basal surfaces. Segmenting even a simple epithelium requires a good bit of time delineating both the basal surface *and* the boundaries between adjacent cells. This is extremely tedious and makes it easy to lose track of cells and boundaries in 4 dimensions. The optimal solution is thus to separate these concerns: first define the apical and basal surfaces by segmenting the whole tissue as a group and *then* identify boundaries between individual cells. The former "whole tissue" region creates an impervious "masking" barrier to keep background outside and cells inside. We use the following 3-step process for segmenting cells in 4D:

1. Identify the cells of interest by drawing a line through the 3D center of each cell through time and putting in a few background seeds wherever needed. This provides a rough visualization of the location of the cells of interest.
2. Use SWS4D in "tissue segmentation" (or mask) mode to segment the bulk tissue from the background along both the apical and basal surfaces.
3. Go back to "cell segmentation" (or normal) mode and clean up the segmentation of each cell-cell interface.

The segmentation using the mask is compared to the one without in Figure 5.5 .

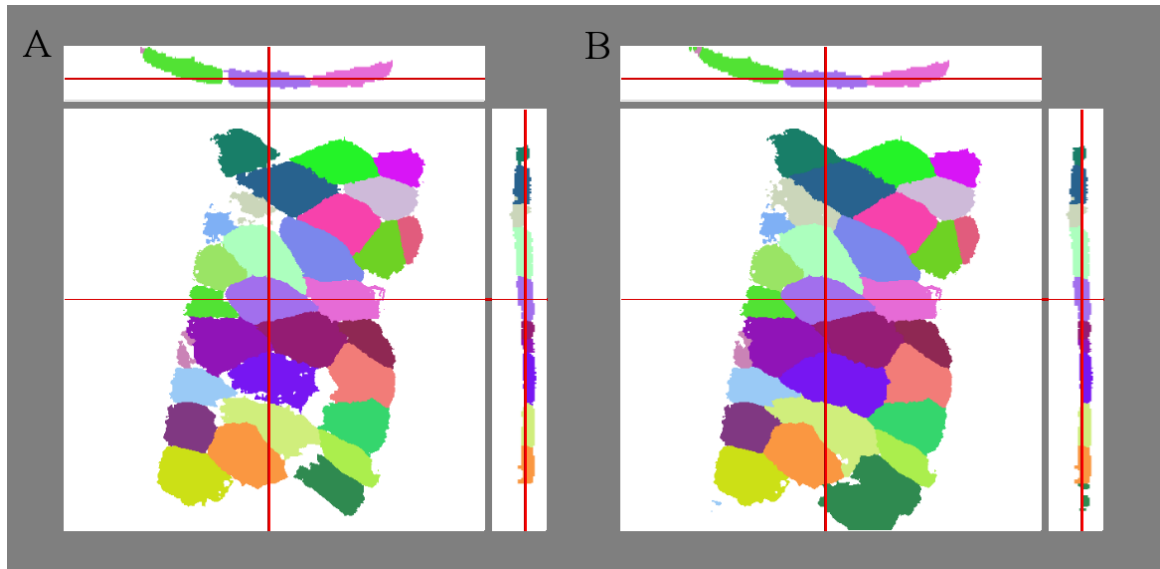


Figure 5.5: Result of applying a mask to a segmentation. (A) shows the segmentation from Figure 5.4 and (B) shows the segmentation using the same seeds, but applying masked segmentation. Notice how the white interior pixels are eliminated when background is excluded using the mask. This is especially noticeable in the xz cross section (top) between the left-most two cells (light green and pale purple).

This process is still time-consuming but is much simpler than without the mask. It also provides a depth of user control far beyond previously described tools.

The data set used in Figure 5.5 was completely segmented over all 37 time frames. For more details about the interactive steps in the the segmentation process, see Supplement S3 below.

5.3. Algorithms and Mathematics

5.3.1. N-Dimensional Line and Triangle Rasterization

5.3.1.1. ND Bresenham Algorithm

The N-dimensional (ND) integer-based line drawing algorithm we used to draw generic 4D lines was extrapolated from the standard 2D Bresenham algorithm and its existing variations in 3D [134]–[139]. As with the 2D Bresenham algorithm, the ND

Bresenham Algorithm (NDBA) procedure involves identifying the "long" dimension (component of the line's vector with greatest absolute value) and stepping one unit at a time in that dimension (pixel by pixel). For each integer value in the long dimension, a value is chosen in the other dimensions based on the usual delta arithmetic prescribed by Bresenham's algorithm to maintain at least corner connections between the hyper-voxels. The implementation we created actually returns the list of coordinates for each pixel in the line; the line is then drawn by blitting these coordinates to the seed array.

5.3.1.2. ND Bresenham Triangle Algorithm

Similar procedures are used for the more complex ND integer-based triangle drawing algorithm (ND Bresenham Triangle Algorithm or NDBTA). Such plane elements are built up using a series of ND lines (using the NDBA); as with the line drawing algorithm, the most important consideration is ordering the dimensions correctly. This process is more complex for a 2D plane than a 1D line, because it requires identifying two dimensions: what we refer to as the *scan* and *line* dimensions. To simplify the drawing process, we limit each line to a fixed and distinct value in the scan dimension, ensuring that lines effectively "scan" the entire triangular plane; another way to think about this is that each line is drawn in a separate (N-1) dimensional subspace defined by a fixed value in the scan dimension. The secondary dimension, called the line dimension, is the same thing as the long dimension that is used in the NDBA; this is the dimension that will be incremented sequentially, leaving the values in the other (N-2) dimensions to be chosen via the delta arithmetic as each line is drawn. The optimal rasterization of the triangular plane is created by properly choosing these dimensions to avoid gaps.

Other algorithms exist to rasterize planar polygons [140], but we could not find one that used a minimal number of voxels, could operate over arbitrary dimensions, and

used integer arithmetic for the main rasterization process. A similar method was also described briefly in Dimitrov *et al.* 2004, but we could not determine the exact procedure described [138].

5.3.1.2.1. Relative Dimension of Steepest Slope

To explain how to choose the scan and line dimensions described above, we must introduce the concept of the *relative dimension of steepest slope* (or RDSS) for an arbitrary, unbounded 2D plane in an ND space. This plane can be defined by three points (p_0, p_1, p_2) where $p_i=(x_i, y_i, z_i, \dots)$. The parametric equations for this plane are:

$$\begin{aligned} x &= x_0 + (x_1 - x_0)s + (x_2 - x_0)t \\ y &= y_0 + (y_1 - y_0)s + (y_2 - y_0)t \\ z &= z_0 + (z_1 - z_0)s + (z_2 - z_0)t \\ &\dots \end{aligned} \tag{5.1}$$

where s and t are parametric variables with $0 \leq (s-t) \leq 1$. So, (p_0, p_1, p_2) can be described in the (s, t) space as simply $((0, 0), (1, 0), (0, 1))$.

For a given dimension A in the ND hyperspace (one of x, y, z, \dots), then for any fixed value of A , the equations of the 2D plane simplify to a line with an $(N-1)$ dimensional vector (unless $A_0=A_1=A_2$, in which case the plane is flat in that dimension, a case we explicitly exclude for this analysis). The component of this $(N-1)$ dimensional vector that has the greatest absolute value is what we define as A 's relative dimension of steepest slope (RDSS).

To derive this quantity, we start by fixing A to a constant value, so that $\Delta A=0$.

From Equation 5.1, we can derive the constraint:

$$0 = \Delta A = (A_1 - A_0)\Delta s + (A_2 - A_0)\Delta t \tag{5.2}$$

which relates Δs to Δt , defining the line. This can be written equivalently as:

$$(A_1 - A_0)\Delta s = -(A_2 - A_0)\Delta t \tag{5.3}$$

For convenience, we define this quantity as Δw :

$$\Delta w = (A_1 - A_0)\Delta s = -(A_2 - A_0)\Delta t \quad (5.4)$$

Now, for each other dimension B , we want some well-defined quantity that can tell us how quickly B grows as the parameters are adjusted: *i.e.* some kind of slope. Although we could use $\Delta B/\Delta s$ or $\Delta B/\Delta t$, a convenient approach is to use $\Delta B/\Delta w$. Again, our eventual goal is to identify which choice for B produces the largest value for this slope.

We can define ΔB just as we did ΔA :

$$\Delta B = (B_1 - B_0)\Delta s + (B_2 - B_0)\Delta t \quad (5.5)$$

Now, since Δs and Δt are related to Δw by Equation 5.4, multiplying Equation 5.5 by $(A_1 - A_0)(A_2 - A_0)$ allows us to perform simple substitutions to eliminate Δs and Δt :

$$(A_1 - A_0)(A_2 - A_0)\Delta B = ((A_2 - A_0)(B_1 - B_0) - (A_1 - A_0)(B_2 - B_0))\Delta w \quad (5.6)$$

Now we can compute $\Delta B/\Delta w$:

$$\frac{\Delta B}{\Delta w} = \frac{(A_2 - A_0)(B_1 - B_0) - (A_1 - A_0)(B_2 - B_0)}{(A_1 - A_0)(A_2 - A_0)} \quad (5.7)$$

From this, we see that the absolute value of $\Delta B/\Delta w$ will be an greatest when the expression:

$$|(A_2 - A_0)(B_1 - B_0) - (A_1 - A_0)(B_2 - B_0)| \quad (5.8)$$

is maximized. This expression can be rearranged even more elegantly as:

$$|A_0(B_2 - B_1) + A_1(B_0 - B_2) + A_2(B_1 - B_0)| \quad (5.9)$$

The RDSS for dimension A is, therefore, the dimension B that maximizes this expression.

5.3.1.2.2. Computing RDSS values

Our first task in rasterizing the triangle is choosing the scan dimension, but first, we need the RDSS for each dimension. The exact computational procedure is this:

1. Given are three points (p_0, p_1, p_2), each with N integer coordinates (x_i, y_i, z_i, \dots).
2. Form a matrix from the three points where each row is a coordinate dimension and each column is a separate point:

$$points^T = \begin{pmatrix} x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \\ z_0 & z_1 & z_2 \\ \dots & & \end{pmatrix} \quad (5.10)$$

3. Calculate the Nx3 "diff" matrix:

$$diff = \begin{pmatrix} |x_2 - x_1| & |x_0 - x_2| & |x_1 - x_0| \\ |y_2 - y_1| & |y_0 - y_2| & |y_1 - y_0| \\ |z_2 - z_1| & |z_0 - z_2| & |z_1 - z_0| \\ \dots & & \end{pmatrix} \quad (5.11)$$

4. Calculate the NxN "pseudo-slope" matrix with entries like Equation 5.9 :

$$points^T \cdot diff^T \quad (5.12)$$

5. For each row in this matrix, find the column(s) (other dimension(s)) with the maximum absolute value; return these column/dimension numbers as a list.

5.3.1.2.3. Final Algorithm

Once we have the RDSS values, the procedure for rasterizing the triangle is fairly straightforward:

1. Find the most common value among all the RDSS values, and use this as the "scan" dimension (x').
2. Given this scan dimension, choose the "line" dimension (y') as the scan dimension's RDSS.
3. Form a list of all the triangle's border points using the NDBA between p_0, p_1 , and p_2 .
4. Sort all the border points so that they go first along the scan dimension (x') and

secondarily along the line dimension (y').

5. Scan over values of x' (move through planes in the scan dimension); within this plane, choose points with the most extreme y' coordinates, and use the NDBA to get the points between them. Collect all the points together and return this as the result

This will rasterize the triangle in N-dimensions, and has been tested in a number of cases; for some examples results, see Figure 5.2 .

5.3.2. Undo Implementation

The original SeedWater Segmenter implements UNDO in a very crude way; two copies of the seed information are stored, one for the current state and one for the previous state. Calling UNDO simply swaps these states. This approach is impractical for SWS4D because the time required for copying seed information adds significant overhead after every modification. This approach also uses double the necessary memory.

SWS4D instead uses an undo scheme based on true inverse operations. Since all seed drawing operations (point, line, and plane) involve changing the seed values for a specified set of points, the inverse operation is just returning those same points to their previous values. Storing this data is significantly less costly in both time and memory usage, enabling the implementation of an unlimited-depth UNDO/REDO system.

5.3.3. ND Sparse Array Implementation

To handle much larger 3D datasets, we manage array handling more consistently by storing seeds as sparse matrices both in memory and on disk. None of the existing sparse matrix file formats met our needs in both space/time efficiency and simplicity/transparency, mostly because we needed to manipulate ND sparse matrices and most existing systems only deal with 1D or 2D sparse matrices. To enable this

functionality, we have two related formats. The simpler of the two is a nested list of 2D sparse matrices. This works well in memory, but is not efficient to store on disk. The second format consists of three separate data structures:

- rcd matrix (row, column, data): a giant 3 column matrix specifying every nonzero voxel's location in XY space (row/column) and its value (data)
- nnzs (number of non-zeros): an (N-2)-dimensional array that holds the number of non-zero elements in each 2D sparse matrix
- shape: the length of the full ND array along each dimension

It is clear that this second 3-part data structure readily admits storage in a compact binary form on disk. The theoretical underpinnings and language used in this format come from the python package "scipy.sparse" which also provides the implementation of the 2D sparse matrices used, particularly the "coo" or coordinate form of sparse matrices. This additionally led to the name for our main nested list format: CooHD, or "higher-dimensional coordinate-form sparse matrix".

Constructing a CooHD from the rcd/nnz/shape format just requires chunking the large rcd matrix data into a pieces and placing them into a list-of-list structure using information from the shape and the nnzs. The reverse process is similarly simple.

This process saves an immense amount of space and i/o time for the seeds, but cannot work for watershed segmentation results where the resulting ND arrays have few if any zeros values. Our solution to this problem is to continue to store the watershed array as a dense array in memory but to store it as a difference format (CooDiff) on disk (and we have actually tested this in-memory as well). To convert a dense 2D matrix with lots of common-value regions to a compressed sparse matrix with the same information,

we just store the first line and then the difference between each subsequent line and the previous one. This generates a large number of zero entries and means the sparse matrices will use less space than the dense matrices. Other compression schemes could work well in this scenario as well, but this method is simple and very fast.

5.4. Software Package Information

Like its 2D+time counterpart, SWS4D is built using python and the standard numerical packages, numpy and scipy, as well as the computer vision library mahotas, which includes the watershed implementation [124].

The rasterization procedures NDBA and NDBTA described above are implemented as the functions BresenhamLine and BresenhamTriangle in the np_utils package, which is a collection of supplemental utility functions, mostly for numerical data.

The sparse array handling is implemented as a separate package called coo_utils. The actual file format saved by this package is a 3-file system, where all three files share the same base name: the shape is stored in a simple plain text format; the rcd and nnzs matrices are stored as *_rcd.npy and *_nnzs.npy respectively, with both using the numpy binary format.

The UI for SWS4D is written using Enthought's Traits and Mayavi packages and is quite dense; the main class for SWS4D is less than 500 lines of code (LOC); the underlying base classes only total to about 300 LOC. In the original SWS, the equivalent classes were about 3000 LOC; however, to be fair, SWS also has a lot more features, including keyboard support and data processing.

Our original unoptimized, dense-array version of SWS4D (SWS4D_Uncompressed.py) is also contained in the source code, but for informational

purposes only. The code is shorter (around 200 LOC) and more legible without sparse-to-dense conversions, but offers no practical advantages for the user, and is no longer maintained.

To deal with exceptionally large datasets, we also implemented a version of SWS4D that can load data as needed for a 3D stack at a given time (SWS_LoadAsNeeded.py). There is a space/time tradeoff here, because this version incurs significant delays when changing the time dimension of the image view. As a workaround to this, we decided to update the xy view dynamically (pulling individual frames directly off the disk), but let the xz and yz views freeze until a refresh operation is initiated (such as moving in x or y after moving in t). Although not ideal, this allows for use on data sets that otherwise could not be loaded at all.

SWS4D is more than just a program; it is really a python module and an interactive dialog for specialized array editing. As such, it is perfectly usable on the python command line or in Python scripts. These modes of operation provide the most general and flexible way to use SWS4D, as data passed into such a script can be in any format as long as the proper transforms are done before passing it to SWS4D. On the other hand, for those who would rather have a stand-alone program we have written a script that looks for a directory of hierarchical tif/gif images or image stacks:

SWS4D_Generic.py. If the image files are the right format and extension and located properly in the directory, this script is a complete and consistent way to run SWS4D.

5.5. Comparisons with other techniques

Numerous other tools exist which use variations on these themes, each with different strengths and target applications, ranging from fully automatic to semi-automatic (manually guided) [114], [116], [118], [127], [141]–[144].

All the existing manually guided 3D segmentation techniques we could find are object-by-object. Not only does this require segmenting each cell-cell interface twice, but boundaries have the potential to cross each other, necessitating some kind of post-processing to maintain consistency. McCullough *et al.* describe a prototypical example of this in which each individual cell is segmented using two or more control points (one body point and one or more surface points) [114]. Many very similar tools exist, both stand-alone and within larger software systems [115], [116], [118], [127], [142]–[148].

Automated 3D segmentation of cells is an extensive field of its own, but surprisingly few tools are focused on segmenting and tracking multiple, tightly connected regions in 3D+time. Pipeline-oriented systems such as CellSegm and TrakEM2 could be suitable for this kind of processing [115], [148]. In addition, recent segmentation algorithms have been created for studying developing plants [145], [146]; fluorescently labeled cell walls produce images that are very similar to membrane-labeled embryonic animals. MARS-ALT in particular has the potential to be useful on animal images, albeit imperfectly since animal cells can have a much wider variety of shapes due to the increased flexibility of animal cell membranes [146].

The only automatic tool capable of truly rivaling SWS4D in segmenting epithelial cells in 3D+time with poorly defined membranes is EDGE4D [147]. This segmentation scheme uses innovative image processing tricks to solve some of the same problems described above such as poor basement membrane definition. EDGE4D and SWS4D complement each other: SWS4D allows for fine-grained control for studying a few very cells in great detail and EDGE4D is capable of segmenting large numbers of cells effectively over large datasets without any manual action. Depending on the requirements, one or the other might do a better job, and a more subtle solution

combining these approaches could also be possible.

5.6. Advice on 2D vs. 3D segmentation

SWS4D segmentation has a number of advantages over the 2D+time SWS, but that does not mean that it is a replacement for SWS. Both have different target applications and different strengths and weaknesses. First, SWS4D is not capable of on-the-fly updates to the watershed segmentation after every user interaction. It is thus not ideal for data handled readily by SWS. In the future, a proper parallel 3D watershed algorithm and a multicore system or sufficiently powerful GPU could overcome this limitation [149]. Second, SWS4D only displays one cell outline at a time, making the full cell network more difficult to visualize quickly. This is not so much a software limitation as a user one; 3D systems have so much complexity that instant recognition of the network is nearly impossible anyway. SWS4D does provide volumetric watershed and plane projections views to display a fairly complete overview of the network with the single-cell outline as an essential tool for characterizing finer details. Third, SWS has a number of seed-based user interactions besides drawing that are not included in SWS4D — selection, movement, swapping, and deletion — but these are less important in 3D and could be implemented in future versions. Finally, we have also lost automatic tracking at the expense of generic 4D drawing. For the use case examined here, tracking is of little help; the time taken to identify each cell through time (by drawing a 4D line through each one) is minuscule compared to the time needed to perform the full segmentation.

5.7. Conclusion

We implemented a watershed-based drawing program for segmenting cells in 4D. We overcame two major hurdles in this process: the need for multi-dimensional voxel plane drawing and the need for separation between bulk tissue segmentation and cell-cell

interface segmentation. We have maintained a number of the advantages of our previous software, SeedWater Segmenter, notably in-process segmentation and tracking (as opposed to post-process correction) and using a space-filling algorithm to capture the properties of densely packed cells in tissue. In addition, we have surpassed SWS's abilities by being able to work effectively with 3D+time data, seamlessly work on orthogonal (xz/yz) views of data, work with seeds in separate colors, "erase" by drawing 0-value seeds, and optionally load image data dynamically to cut down on memory usage.

Despite these advances, SWS4D is far from a silver bullet. Although the watershed algorithm is quite effective in certain circumstances, other methods (like intelligent scissors, GrowCut, or numerous others) can be more effective and/or faster for many tasks, particularly in medical imaging [116], [128], [133], [150]. It is also possible that one or more of these techniques could also be implemented in SWS4D as an enhancement.

We have designed SWS4D to be a simple and effective way to perform 3D watershed segmentation over existing data sets. We see its value as three-fold. First, it provides a convenient way to interactively test and explore the 3D watershed method, helping users evaluate whether it will be effective in their situation. Second, it provides tools necessary to perform detailed segmentations for situations that are not amenable to automation, either due to unusual/inconsistent image features or user expertise; it is particularly effective at simultaneous segmentation of adjacent regions such as cells in epithelial tissue. Third, it introduces a number of new algorithms (most notably the NDBTA for rasterizing higher-dimensional triangles), many of which could be useful in other applications.

Chapter 6

6. Description of wound dynamics through segmentation in 2D and 3D

6.1. Introduction

As discussed in Chapter 3, laser wounding provides an excellent way to measure many properties of epithelial tissues and their constituent cells. Laser ablation releases forces in tissue, leading to a fast recoil (<30 s) of the remaining tissue away from the site of the ablation, followed by a short, quasi-equilibrium state and a slower healing response (~ 30 min). These processes are easily captured with live imaging, although the fastest aspects of the recoil require imaging at time intervals around 10-20 ms [31], which is not achievable with single point, full frame laser scanning confocal microscopy since that is limited to >1 s acquisition times for 512×512 pixel images; the traditional work-around has been scanning along single lines to track movement of cell-cell interfaces accurately in 1D instead [15], [20], [31]. Still, basic information about the recoil processes can be obtained even for very slow imaging (~ 10 s per frame), and in the future, newer techniques like spinning disk confocal now make it possible for researchers to image in two dimensions at close to these rates (~ 30 ms per frame). Irrespective of acquisition time, generating quantitative data from live images of either recoil or healing processes requires accurately identifying cell boundaries, which is effectively accomplished through image segmentation.

The patterns of displacement illustrate a lot about the underlying mechanics, and numerous groups have used them effectively to estimate relative forces and inform theoretical mechanical models for epithelial tissue structure [3], [15], [18], [20], [31], [70], [79]. In addition, ablation activates a host of biochemical pathways that lead to repair, making this a fruitful avenue of study [9], [10], [18], [69], [79]. However, there is

much we still don't know about both the recoil and healing processes and the interplay between the two. In fact, we are still missing quite a bit of basic information, such as the role that the third dimension plays for wound healing in epithelia. This work seeks to help address these questions.

Using our previously developed tools, SeedWater Segmenter (SWS) and SWS4D [2] (described in Chapter 4 and Chapter 5), we have segmented live images of laser wounding events, recording both the recoil and healing response of the tissue. In addition to traditional single plane imaging studies using cadherin-GFP to identify lateral cell-cell boundaries, we have also analyzed a laser wounding event in three-dimensions using embryos expressing the Resille-117 membrane marker; this additionally identifies the apical and basal membranes and has allowed us to access depth information about cells during the healing process. This work both confirms and expands upon previous findings in this area.

6.2. Materials and Methods

The primary strain of *Drosophila melanogaster* used in this study is ubi-DE-Cad-GFP [121] (Kyoto *Drosophila* Genetic Resource Center). This strain ubiquitously expresses a cadherin-GFP chimera that labels epithelial cell junctions. Additional experiments were performed using Resille(117-2)-GFP [151] (gift from J. Zallen, Sloan-Kettering Institute, New York, NY). For imaging and ablation, fly embryos were dechorionated in a 50% bleach solution, immersed in halocarbon oil 27 (Sigma-Aldrich, St Louis, MO), and sandwiched between a cover glass and an oxygen-permeable membrane (YSI, Yellow Spring, OH) [71]. Images were captured on a Zeiss LSM410 laser-scanning confocal microscope (inverted) with a 40×, 1.3 NA oil-immersion objective and 488 nm excitation. The scanning times were 8 s per frame with 2 s delays

between frames (10 s total intervals) for cadherin-GFP single-frame imaging and 4 s per frame with 2 s delays between frames (6 s total intervals) for Resille-117 imaging.

Images were collected at a resolution of 0.31 $\mu\text{m}/\text{pixel}$, and image stacks were collected with 21 frames at 0.5 μm offsets in z (10.5 μm total depth and ~ 133 s total interval between stacks).

6.2.1. Laser Microsurgery

Ablations were performed with the third harmonic (355 nm) of a Q-switched Nd:YAG laser (5 ns pulsewidth, Continuum Minilite II, Santa Clara, CA). This laser was coupled into the Zeiss LSM410 with independent beam steering for simultaneous ablation and imaging [25]. The pulse energy was just high enough (2–3 \times threshold) to ensure consistent single-pulse ablation.

6.2.2. Image Analysis

We used ImageJ (National Institutes of Health, Bethesda, MD) software for basic image-processing tasks. To measure the areas and volumes of cells, we used SeedWater Segmenter [2] and SWS4D, custom watershed-based segmentation software.

6.3. Results

We recorded time-lapse images for 5 cadherin-GFP embryos from a few seconds before the ablation until the wound was essentially closed or imaging became too noisy to accurately assess cell-cell boundaries (between 15 minutes and two hours). We also recorded imaging data for a Resille-117 embryo, collecting two 10.5 μm z -stacks, followed by 19 time-lapse images surrounding the ablation event, and then followed by five more z -stacks. We used this protocol to capture depth information before and after the wound as well as information about the recoil event. By combining these two studies, we have been able to assess how ablated tissue responds both laterally and axially, giving

a more detailed picture of this process.

After images were collected, they were segmented by hand using SWS and SWS4D to identify which image regions corresponded to each cell to high accuracy, paying special attention to the cells targeted by ablation and surrounding the ablation site. Once the segmentation was completed, we analyzed the following metrics:

- Area: computed from the number of pixels
- Perimeter: computed using the region boundary
- Aspect Ratio: computed using moments of inertia
- Radial Aspect Ratio: computed using moments and the centroid of ablated cell(s)

Area is the simplest metric to compute from a segmented image, since it is just the number of pixels in the cell times the squared length of one pixel. Perimeter is slightly more complicated, since the step-wise boundary of a rasterized region overestimates diagonal lengths. To effectively compute perimeter, we therefore use a combination of a border-crawling algorithm and a length-correcting estimation algorithm [152], [153].

Aspect ratio is a measure of the relative elongation of a cell, computed as the square root of the ratio of the principle moments of inertia of the cell's pixel region. Radial aspect ratio is a measure of how surviving cells stretch in response to the wound. Using the centroid of the ablated cell(s) as the reference center, radial aspect ratio is calculated as the square root of the ratio of the radial to the tangential moment of inertia. This measures the relative elongation in the radial direction away from the site of the wound.

In addition to segmentation, we also converted the segmented tissues into a "cell network": a set of 3-way cell junctions (triple junctions) connected by 2-cell boundaries (edges). We used this network to group cells into nth-nearest-neighbor "rings" around the

wounded region; ring 1 cells (wound nearest-neighbors) are adjacent to the wounded region, ring 2 cells (second nearest-neighbors) border ring 1 cells but not the wounded region, etc. By binning cells into these rings, measures like area and radial aspect become much more meaningful. Figure 6.1 shows these values for one of our data sets. The reported areas are normalized such that the mean value before the ablation is equal to 1 to accentuate the trends of these cells.

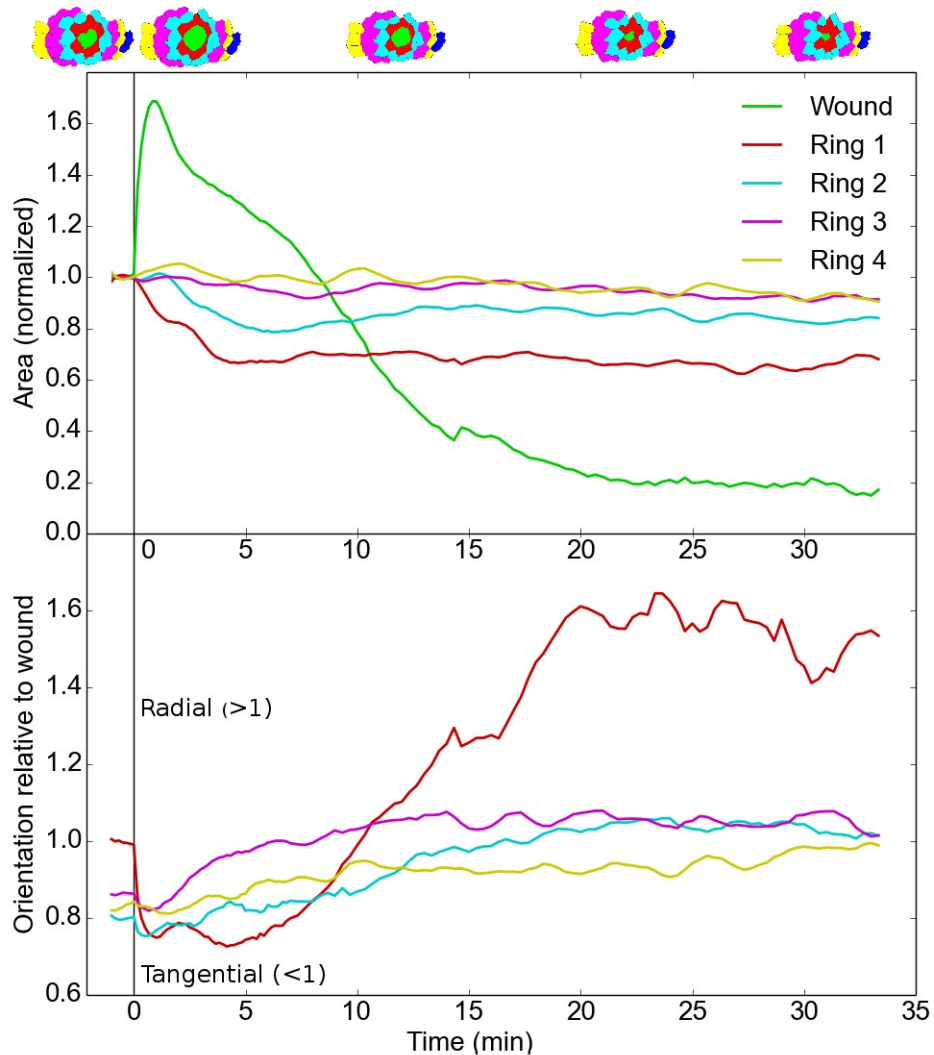


Figure 6.1: Normalized areas and mean radial aspect ratios vs. time for the wounded region and successive rings of nearest-neighboring cells. Areas are normalized by the mean value before ablation. Colored thumbnails of the segmented data set at various points in time are shown along the top of the plot as a guide.

A number of trends are evident here. First, it is clear that the area of the wounded cell increases during the recoil process, pauses, and then shrinks, approaching zero. The surrounding ring (ring 1) simultaneously shrinks in area while becoming elongated tangentially, suggesting that it is being stretched out as the wound opens. Ring 2 shows similar effects, but they are less pronounced. The outer rings do not seem to react to the wound in any significant way, although there is a moderate reaction in the radial aspect due to the recoil from the ablation. The main point of interest for the outer rings is that they show an overall trend towards cells shrinking over time, which is a natural consequence of dorsal closure. Interestingly, the ring 1 cells do not rebound in area but do show a strong reversal in radial aspect ratio as the cells close in around the wound; these cells form rosettes, each extending towards the wound to reform the epithelium. This corroborates the findings of Meghana *et al.* [18].

We also analyzed the lengths of individual cell-cell interfaces in the cell networks using a modified version of the perimeter-estimation algorithm. By binning these values, we computed the lengths of all cell edges radiating out from the wounded region (spokes) and the the perimeter around the outside of ring 1. The wound perimeter and these measures are shown in Figure 6.2

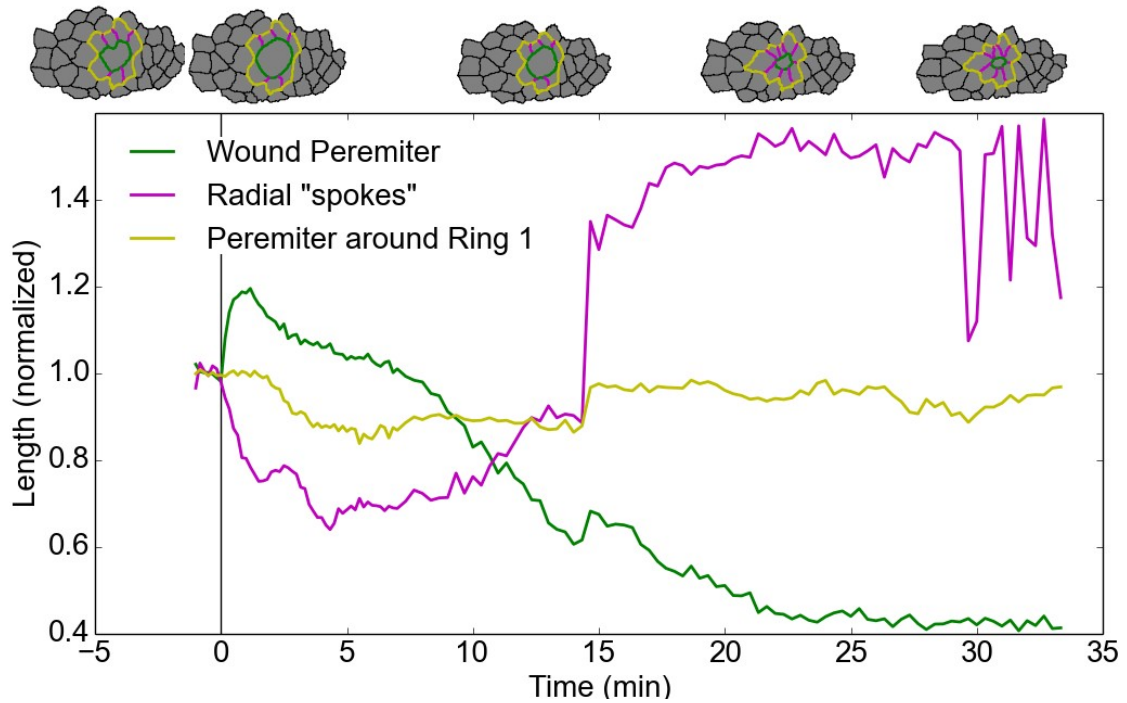


Figure 6.2: Normalized lengths of wound perimeter, radial contours, and first nearest-neighbor outer perimeter. Lengths are normalized by the mean value before ablation. Thumbnails of the segmented data set are shown at along the top, colored according to ring.

The contour lengths show very similar trends to the areas and aspect ratios. The perimeter of the wound matches closely with the area, which is expected since it stays nearly circular the entire time. The spoke lengths follow a similar trend to the radial aspect ratio of the first ring of cells, which is again unsurprising since these features are highly correlated due to the geometry. One interesting point is that there was a neighbor exchange after about 15 minutes which caused this measure to spike due to the addition of a new contour. Even without this, the trend would still remain. Last, the perimeter of ring 1 does not show a strong trend but does behave similarly to ring 1 area, which makes sense; the ring 1 perimeter should behave like the sum of areas of the wounded region and ring 1, and these oppose each other, leading to a muted effect.

For image stacks, we computed volumes and projected apical areas, which correspond to the areas measured for time-lapse sets. We also computed mean thickness as the ratio of volume to area. Figure 6.3 A shows the trends for these measures binned by groups given in the key (Figure 6.3 C). Figure 6.3 B shows the binned areas for the time-lapse taken around the time of ablation (set as $t=0$), and in between z-stacks. The time during which the imaging for Figure 6.3 B took place is marked on Figure 6.3 A with gray hashes.

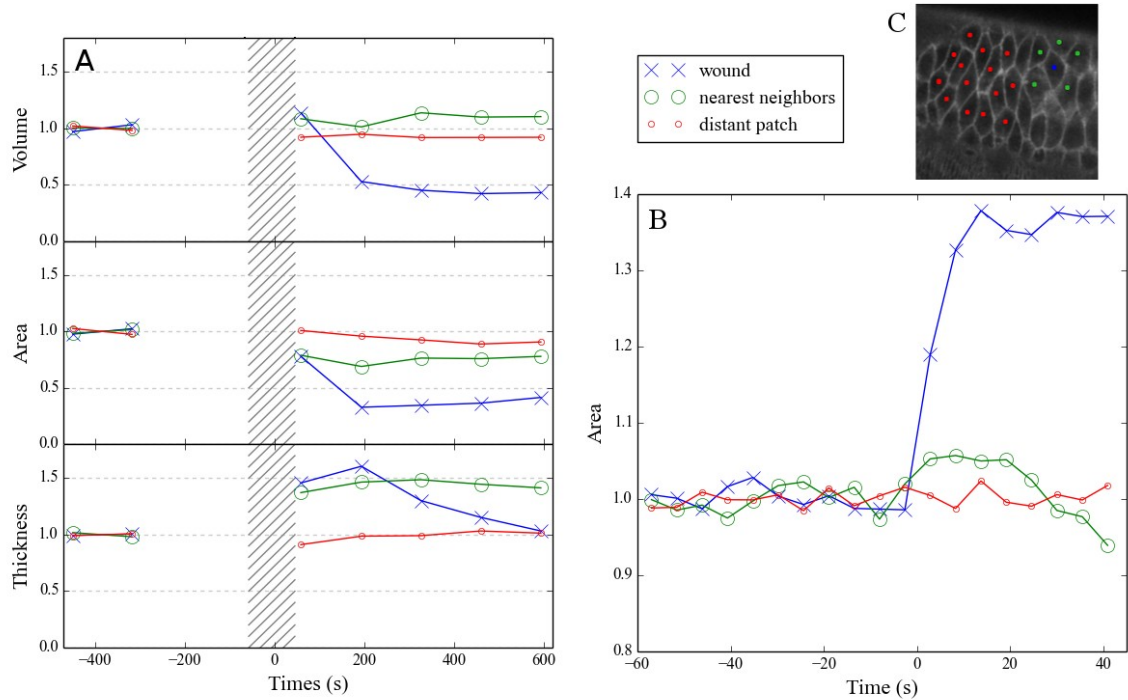


Figure 6.3: Mean-normalized volume, area, and average thickness of cells from the Resille-117 embryo.

(A) Volume, projected area, and average thickness for cells in z-stacks. Hashes denote the time during which time-lapse images were being collected near the ablation time ($t=0$).

(B) Area for cells in time lapse before and after ablation. (C) Key for which cells are in each category; blue for the wounded cell, green for nearest-neighbor ring, and red for cells distant from the ablation. The areas in (B) roughly correspond to those in the middle panel of (A), but are not exactly the same since the projected area takes into account planes other than the apical one.

The trends in volume are very clear, with the undamaged cells maintaining their volume and the wounded cell losing volume. The projected apical area is very similar to that of the cadherin-GFP embryos, showing a large boost in wound area and showing an eventual decrease in ring 1 area. Interesting, the ring 1 area actually grows for ~30 s before it shrinks. The average thickness (z direction) can be computed as the volume divided by the projected area; these trends are equally interesting, showing that the volume of the wounded regions thickens a great deal during recoil. The nearest neighbors thicken similarly, around 50%, and stay that way throughout this experiment.

Since we know that the wound grows in depth and shrinks in area, we also looked at the cross-sectional area with depth. Figure 6.4 shows these profiles which clearly elucidate the difference between the more squamous cell before ablation and the more columnar cell afterwards.

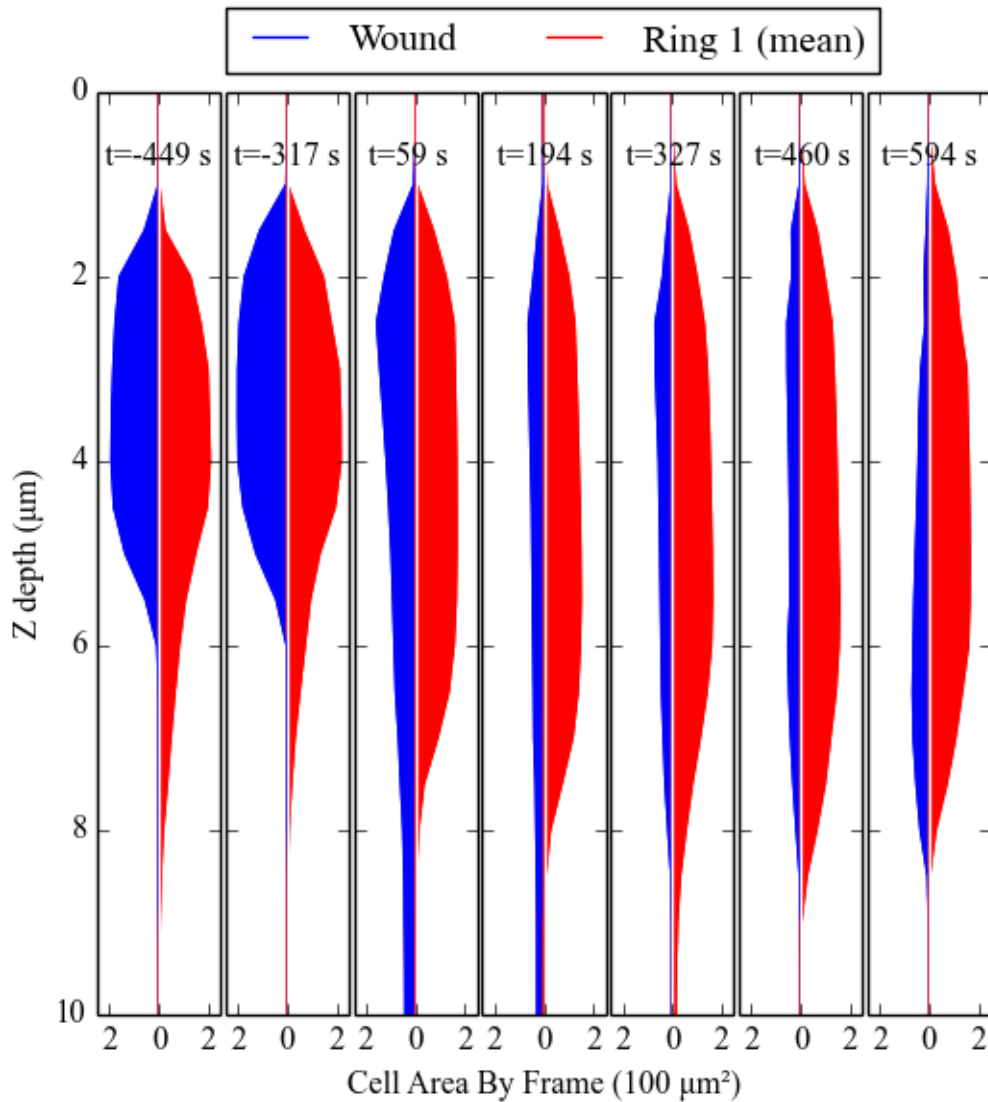


Figure 6.4: Wounded region and neighboring cells' occupation at different depths and times.

At each time point (left to right), the area (in $100 \mu\text{m}^2$ increments) vs. z depth is plotted for both the wounded cell (blue) and the nearest neighbors (red, normalized by the number of cells). This clearly illustrates that the wound goes from wide and short to very tall and narrow, extended deeper into many more z -frames. The surrounding ring of cells also becomes taller and thinner as it recoils away from the ablation site. Notice how the wounded cell loses overall volume over time while the neighbors do not.

6.4. Conclusions

In this paper, we have corroborated the results from Meghana *et al.* [18] about the

nature of the first ring of cells around a laser ablation wound, namely that surviving cells surrounding a wound stretch tangentially into rings during recoil and rearrange during the healing process to become radially oriented, forming a rosette pattern as the epithelium reforms around the wound. Additionally, we have shown that wounded cells get pushed basally into the interior of the embryo and that surrounding cells lose apical area, stretch into a ring, and thicken in the z direction.

This analysis provides a foundation for future research into laser ablation studies; combining and refining these types of image-based metrics will be key in identifying further details about the mechanical nature of wound response and development.

Chapter 7

7. Errors in Force Inference Techniques

7.1. Introduction

Force inference (FI) methods extract cell geometries from experimental images and then solve an inverse problem to find a map of forces that would yield the observed cell shapes [1], [38]–[40]. These methods are the inverse of traditional forward modeling techniques, like the cell-level finite element (FE) method [37], which instead take forces as inputs and incrementally displace nodes in a polygon network until the system equilibrates. Although such forward modeling is certainly effective at uncovering patterns occurring in real systems, matching a simulation to a specific scenario requires a highly user-involved iterative process. FI techniques automate this process completely and typically require less computation time than a single forward simulation.

Although FI techniques have great potential and have proven quite effective in limited circumstances, a number of open issues remain. First, curvatures of cell-cell interfaces have thus far been assumed to be small [1], [38]–[40], but we routinely study systems where these edges are highly curved (Figure 7.1 A). Also, the effects of random positional noise and image resolution on FI quality have not been examined systematically over a wide range of perturbations; these are especially important because all real-world data will necessarily be limited by one or both of them. The present study addresses FI's effects both from large edge curvatures and random noise or image resolution.

As stated above, FI is the inverse problem to the finite element (FE) method. FE models map input forces to individual nodes in a given network and then calculate the displacements necessary to move the system to a state of either static or dynamic

(viscous) equilibrium. FI techniques, instead, start with the final positional patterns (and optionally displacements), symbolically set up the equations of force balance and infer the originating tensions and pressures acting on these nodes to some close approximation by assuming the system is in equilibrium. Figure 7.1 B and C show a mesh and a map of tensions (γ) and pressures (σ) resulting from application of FI on this mesh, in this case reproducing the original FE values with great fidelity (not pictured). As a rule, FI solutions are not guaranteed to find perfect equilibrium, and some residual force vector may remain at each node. If there are exactly the same number of equations and unknowns, the problem becomes a simple matrix inversion, but this is rarely the case and only occurs arbitrarily. More often, the system will instead be either under- or over-determined depending on the number of equations and the number of unknown pressures and tensions. The number of equations is directly dependent on (2x) the number of nodes in the polygonal mesh that FI is acting on; the number of unknown pressures and tensions depends on the mesh topology, but can also be affected by any additional assumptions or constraints. Because there are a number of reasonable assumptions as well as different mathematical inversion methods, FI must be seen as a class of solutions rather than a single clear-cut algorithm.

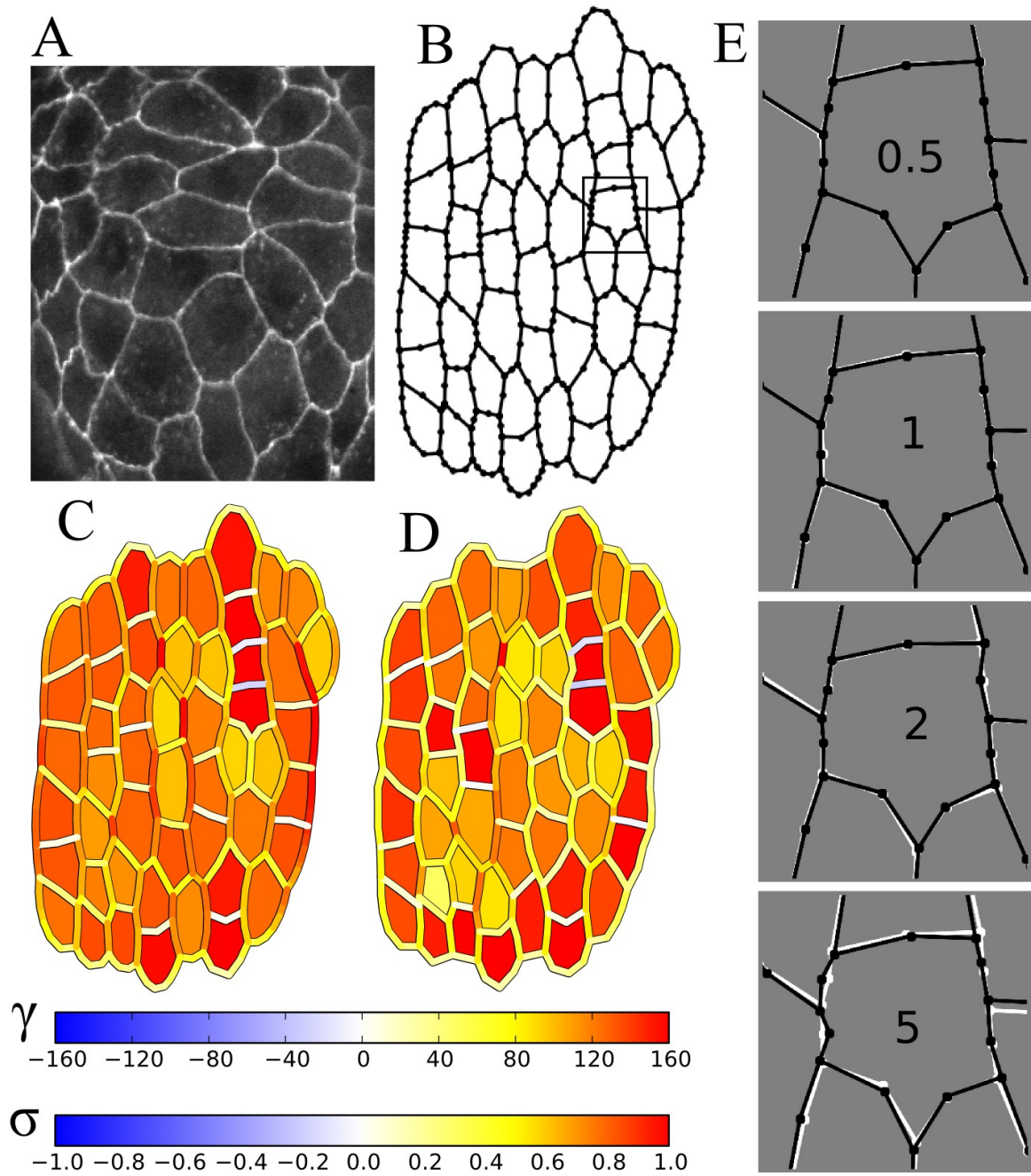


Figure 7.1: Overview of Video Force Microscopy.

(A) Representative image with highly curved edges. (B) Test mesh, produced by a Finite Element (FE) simulation (C) Result of VFM on the mesh in (B), shown as a combined map of edge tensions (γ) and body pressures (σ). Blue is outward stress, white is no stress, and red is inward stress. (D) Example VFM result for a mesh perturbed both by node re-interpolation and by Gaussian positional noise. (E) Example meshes with varying amounts of noise applied to the mesh in (A); the region shown is marked with a small box in (A). Original and perturbed meshes are shown in white and black respectively. Standard deviation of the noise is shown in each panel as a percentage of the mean cell radius (0.5% to 5% from top to bottom).

So far in this discussion, we have taken the mesh as a given, but when applying FI to a real image, the process of creating a mesh leads to additional ambiguities. Figure 7.1 A is an image of a *Drosophila* embryo during dorsal closure, a system of great interest mechanically and one with highly curved cell-cell interfaces (edges). It is obvious that any mesh created from this image should include the 3-way cell-cell interface points (triple junctions), but the edges in between are more complex to represent in a model. The simplest solution is to draw a straight line between each triple junction, but for curved edges, this can easily misrepresent the angles, which are immensely important for FI. Another solution is to represent each edge with a piecewise linear fit, a solution which mimics the types of meshes used in FE models. Figure 7.1 B is a representative example of such a mesh produced using a FE simulation and chosen to mimic large curvatures using intermediate nodes. It is the starting point for the rest of the analysis presented here. Applying VFM to this mesh produces values for edge tensions (γ) and body pressures (σ) as shown in Figure 7.1 C; these match closely to those from the original FE simulation (not pictured). Tensions and pressures are each represented with a color scale ranging from blue (expanding/outward stress) to white (no stress) to red (contractile/inward stress), but the scales are independent of each other as depicted in the colorbars below. VFM is able to determine the relative levels of stress in this case, and there is a clear distinction between the stress in the vertical edges compared to the horizontal ones.

Obviously, noise and image resolution will affect the precise placement of any nodes (Figure 7.1 E), but more importantly, there is no prescribed number of nodes to include. Although the triple junctions are well defined geometrically as points where three cells meet, there is usually no information to aid the placement of any additional nodes along cell-cell interface boundaries (edges), meaning that a valid representative

mesh could contain no intermediate nodes or an unlimited number of them. Some FI techniques ignore intermediate nodes, but this limits the accuracy to which they can model highly curved edges. On the other hand, Chiou *et al.* have shown that adding more equations leads to greater sensitivity to error [38]; this can also be observed in Figure 7.1 E by looking at how the angles around short and long edges respond to noise. In general, this is an open problem for FI.

To assess how FI responds to the number and location of intermediate nodes, we replaced all the intermediate nodes in Figure 7.1 B/C with new ones; Figure 7.1 D shows the VFM results for one such mesh. This mesh has been perturbed both by node re-interpolation and by additive Gaussian positional noise. First, all the intermediate nodes in the mesh were removed, leaving only the triple junctions. Next, new nodes were placed at some even interval along each cell edge; in this case 95 total nodes were added. Finally, all the nodes were shifted by isotropic gaussian noise with a standard deviation of 2% of the mean cell radius. Note that although the mesh is still quite similar to Figure 7.1 B, the slight differences, particularly in the angles, create a clear change in pressures and tensions calculated by VFM.

Different levels of positional noise have a clear effect on the quality of VFM results. Figure 7.1 E illustrates how much various amounts of noise perturb the sample mesh, showing a zoomed view of a small region in Figure 7.1 A (delineated by a small box). In order to accentuate the differences, the original mesh is drawn in white with the perturbed mesh drawn over it in black. The text in center of each example is the standard deviation of the noise as a percentage of the mean cell radius, from 0.5% at the top to 5% at the bottom. Notice how significantly the angles can change with only 5% noise; also note how the smaller the edges, the more sensitive they are to angular noise. Increasing

the number of nodes will make the system more sensitive to coordinate noise, while decreasing the number of nodes will make the system more robust, but less accurate. The most accurate representation will thus be a compromise between these two competing effects; this mesh used in Figure 7.1 D was actually the best compromise for 2% coordinate noise case.

The use of FI techniques for biological systems has received a lot of attention recently and significant progress has been made in understanding and improving them starting with the introduction of Video Force Microscopy (VFM) in 2010 [1]. Currently, three of the most advanced techniques are VFM method, the Mechanical Inverse (MI) method from Chiou *et. al.*, and Bayesian force inference (BFI) from Ishihara *et. al.* [1], [38]–[40].

VFM requires an over-determined system and solves the system with simple linear least squares; it also requires a small number of assumptions to constrain the solver from finding trivial solutions (all forces zero). In order to produce over-determined systems it is often necessary to include some intermediate nodes along the edges; as stated above, this also has the added benefit of improving the accuracy of the geometric depiction of systems with large curvatures. Of the three methods, VFM is most nearly the true inverse of the FE model in Chen *et al.*; in fact, we have been able to show that running VFM directly on the output of a forward model can reproduce the inputs exactly in most cases, leading to zero residuals in the force balance equations.

MI approaches FI in essentially the same way as VFM, but (1) explicitly ignores viscous damping, (2) introduces the idea of solving “open” systems where the outer boundary is excluded from the system, (3) uses straight-line edges, explicitly excluding all intermediate nodes and (4) often adds the assumption that all pressures are constant

[38]. Because open systems frequently become under-determined, the Moore-Penrose pseudo-inverse solver must be used in place of traditional least squares; this chooses the solution vector (from the infinite number of possibilities) which has minimum Euclidean (L^2) norm. The pseudo-inverse is actually a more general way to solve these systems because it performs linear least squares in over-determined situations. Their assumptions are advantageous in situations when they are valid (static equilibrium, small amounts or curvature, and relative constant pressures) because of a decrease in noise sensitivity, as shown in Chiou *et al.* [38].

BFI is another variation on the ideas behind MI and VFM, but uses a Bayesian solver for the inversion and includes the prior assumption that all tensions should be Gaussian-distributed around a target value [39]. Like MI, it ignores viscous damping and applies to open systems without internal nodes, but unlike MI, it is able to very accurately handle cases with significant pressure variation.

All of these techniques have had successes but clearly also miss certain expected features in some cases. We have undertaken a study to explicitly look at why these failures occur, and specifically how FI is affected by varying amounts of noise, varying image resolution, and varying numbers of interior nodes in a system with large curvatures.

7.2. Results and Discussion

Obviously, to do accurate assessments, we need to study a system with known “gold standard” target forces, so it is reasonable to use the output mesh from a FE simulation (like Figure 7.1 B) as the input to our various FI trials. We must be careful when doing this, however, because doing so without qualification may simply “prove” that our method gives back perfect results; Somersalo and Kapiro refer to this application

of an inverse technique to the output of its own forward model as an "inverse crime" [154]. Adding noise is one way to circumvent this, but to further distance the inverse from the forward model, we also intentionally chose a FE model that was undergoing viscous damping and was therefore out of static equilibrium. In addition, for each trial, we also remove the original mesh's interior nodes and replace them with a completely different set distributed at some even interval. Doing this provides a more accurate sense of the real effectiveness of FI, particularly for VFM, which is the direct inverse of the FE simulation.

Our first study investigated the effects of Gaussian positional noise on VFM while varying numbers of intermediate nodes. We started with a single test mesh from a FE simulation, depicted in Figure 7.1 B, modifying the mesh before each trial. The results of this study are presented in Figure 7.2 which shows correlations with the gold standard solution. Examples of positional noise are pictured in Figure 7.1 E, characterized by the distribution's standard deviation measured as a percentage of the mean cell radius (square root of the mean cell area divided by π). For each noise level, 20 trials were averaged together to get each data point (standard deviations of the trials are shown as an envelope around each curve). The left panels show the dependence of the correlation on the number of intermediate nodes in the re-interpolated mesh, with the best case for each noise level marked with a dot. The middle panel shows the same thing as the left panel, but for Bayesian Inference instead of VFM. The right panels show the dependence of the correlation on the level of Gaussian noise in the best case for each curve in the left; the plot markers correspond between all the panels.

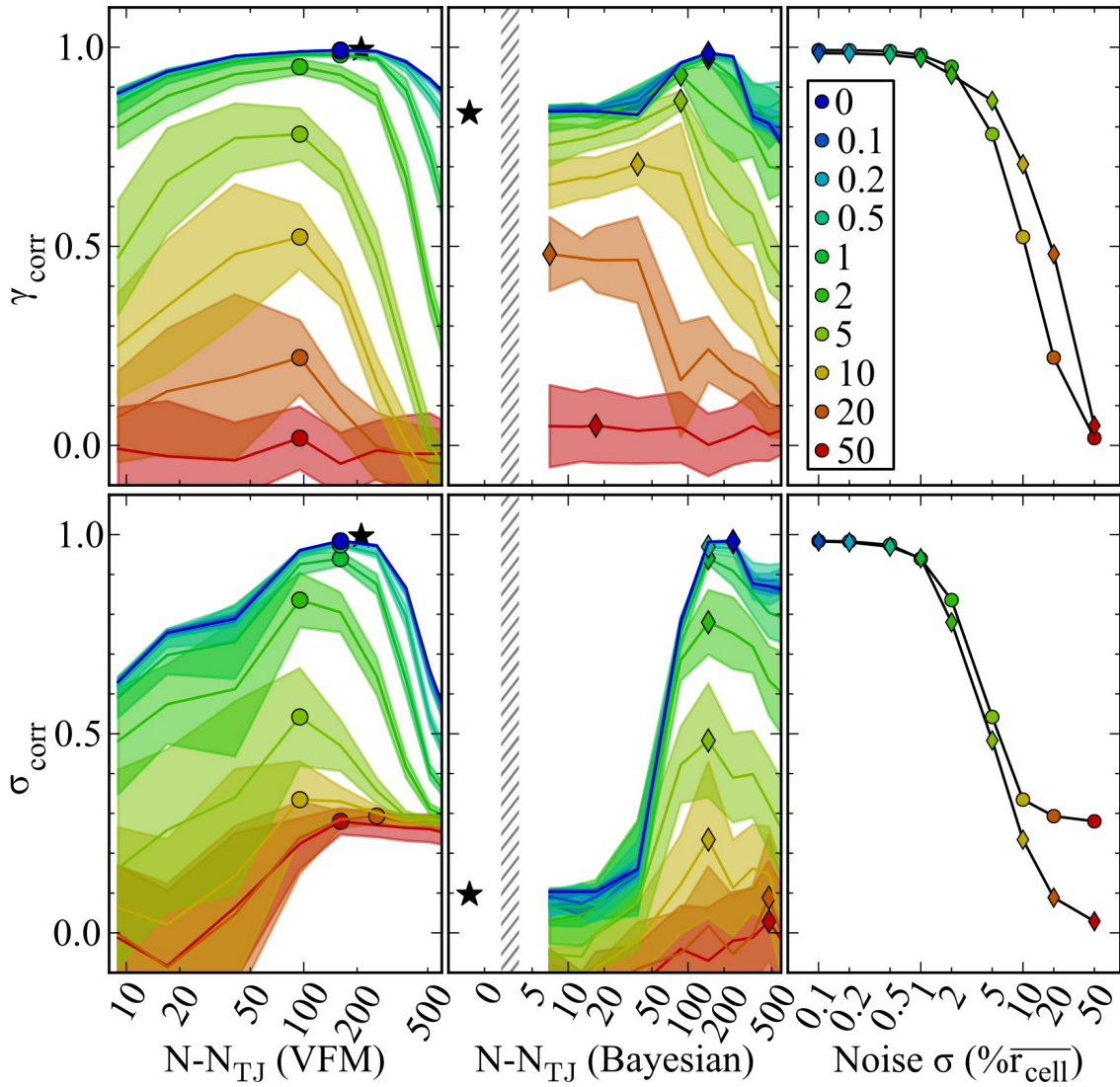


Figure 7.2: Effect of Noise on VFM.

Log-log plot of gold standard correlations for all trials in the noise study. Each data point plotted represents the mean value of 20 random trials and standard deviations are shown as confidence intervals (shaded regions). Correlations were calculated against known values from a finite element model. Left panels show tension (γ) and pressure (σ) correlation vs. number of total added internal nodes (number of nodes minus number of triple junction nodes, $N - N_{\text{TJ}}$); each curve has a particular amount of added Gaussian positional noise, with the standard deviation denoted by a ratio of the mean cell radius in the legend. The unmodified mesh solution is plotted as a black star and can be taken as the theoretical minimum error for this mesh. Importantly, for a given noise level, there is an optimal number of internal nodes (on average, about the same as the number edges, 147). These minima are denoted with a dot and are plotted in the right panel vs. the noise level to make the trend clearer. The central panel shows the same results but for Bayesian Force Inference instead of VFM. Because the standard way to use BFI does not include internal nodes, we present this solution as well (left of the hashes).

When measuring the correlations between the edge tensions (γ) or the cell pressures (σ) with respect to their gold standard values, this looked only at errors in the patterns and ignored problems with scale and offset, parameters which are not well-constrained by FI, especially MI and BFI. In our study, the solutions are essentially uncorrelated with the correct values with noise in the range of 20-50% r_{cell} ; these meshes also look extremely jagged, so this is not surprising (not pictured). The solution of VFM on the unmodified mesh (original nodes, no noise) is pictured on the graph as a star; this can be considered something of a theoretical upper limit for the correlation in this case because as stated above, this example will not reach near-zero errors because the system is not actually in static equilibrium.

Overall, the trends we observed in the results of the study were not unexpected; larger amounts of noise tended to increase the error in the output with 50% correlation occurring at about 10% noise in the best case. We expect noise levels in our image (such as Figure 1A) to be on the level of about 1-10%, so this means that for our images, we can expect some information out of VFM, but nothing near perfection, and the image quality will make a great deal of difference. The other interesting point is that the number of nodes always has an optimal value; this value is relatively close to the number of points in the original mesh.

Since in normal cases, FI will not have the advantage of known solutions, we also considered a number of methods to assess solution quality internally. The simplest measure by far is the condition number of the matrix to be inverted ($G^T \cdot G$), which is the ratio of the largest and smallest singular values in a singular value decomposition. This

provides a quick, rough measure of the sensitivity of the system to noise. Other important metrics in evaluating a FI solutions can be derived from the force residuals. The residuals directly measure the unbalanced forces on the nodes of the mesh, and can be easily calculated by multiplying the original matrix "G" by the vector of tension and pressure solutions and then subtracting any non-zero input forces (if the system was assumed to be non-equilibrium). Globally, these can be collected into a sum of squared residuals, which is the very measure that a least-squares solver minimizes; also, the mean of squared residuals can be a more general way to evaluate between meshes with different numbers of nodes. Residuals can also be used to calculate the "standard errors" for each individual tension and pressure being solved for. Standard errors are the square roots of the diagonals of the variance-covariance matrix, which, in turn, is just G^{-1} times the sum of squared residuals divided by ($\#equations - \#unknowns$). These provide a direct measure of the confidence interval for each tension and pressure, and can show at which locations in the mesh that the solution can be assumed to be accurate and where it cannot. For our test case, the standard errors follow the same basic trends as the gold standard errors, given relatively low noise ($< \sim 10\%$) and reasonable numbers of internal nodes ($< \sim 300$). Figure 7.3 shows these internally computed standard errors for VFM.

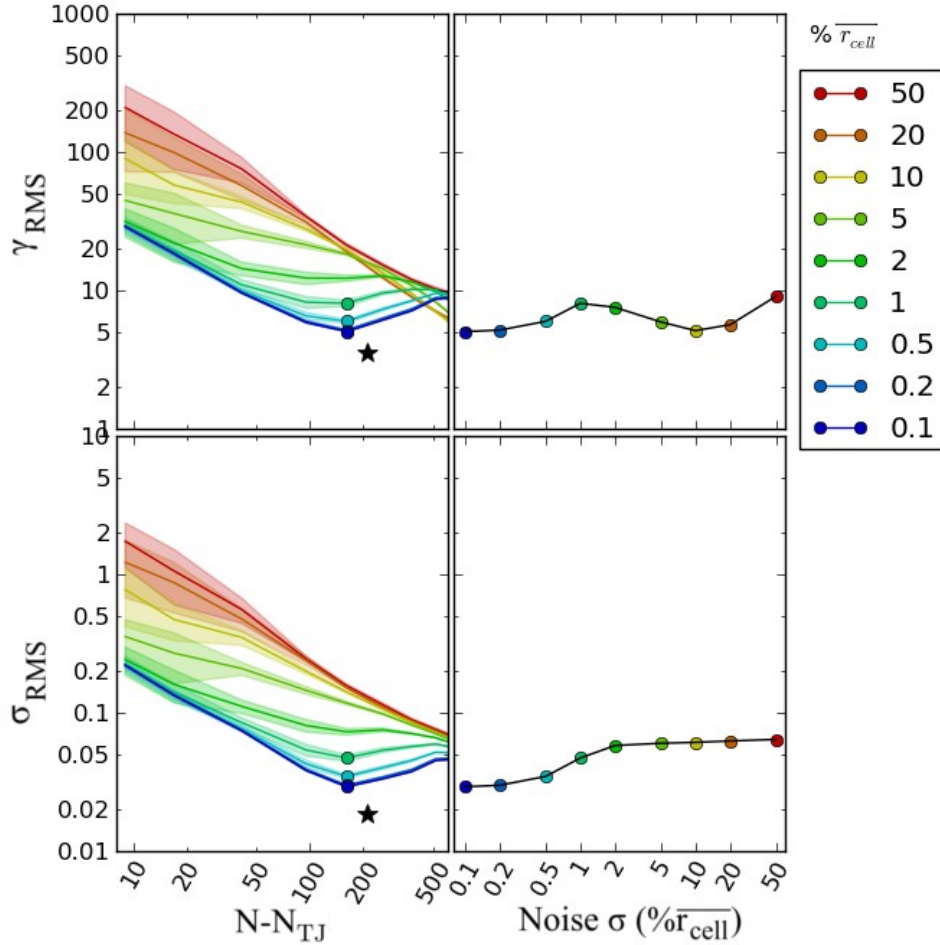


Figure 7.3: Standard Errors reported by VFM for the Noise Study. This is a log-log plot of the internally computed RMS errors for VFM from all trials in the noise study.

In addition to the noise study, we also looked at the effects of image resolution directly, because many of our images (such as Figure 7.1 A) are more affected by rasterization than by pure Gaussian noise. To do this, we still needed a gold standard reference for comparison, and since the FE output is a floating point mesh, we first had to actually draw the lines of this mesh onto an image. By scaling mesh coordinates before drawing, we created smaller or larger images (and conversely, larger and smaller pixel

sizes relative to the mean cell radius). We tried this both with and without spline interpolation, and the latter performed better, avoiding sharp corners except at triple junctions. Next, we segmented these images as if from live data, producing a watershed image and then a crude outline mesh with right-angle boundaries [2]. After that, applying the intermediate node replacement procedure (as described above) created a smooth mesh, and then the coordinates were re-scaled to match back up with the original mesh as closely as possible. As in the noise study, we varied the number of intermediate nodes to adjust between fewer equations and better fit to the geometry.

The results of this image re-meshing are in Figure 7.4 , which has the same format as Figure 7.2 , but we only conducted one trial for each data point since there was no randomness. The trends in this case are very similar to the noise study, with the same maximum correlation occurring with a number of intermediate nodes similar to the original mesh and larger relative pixel sizes leading to smaller correlations (larger errors). Interestingly, the correlation never drops below 0.5 in this case, suggesting that rasterization may be less important than pure noise. The sample dataset in Figure 7.1 A, for reference, has a mean cell radius of about 30 pixels, making the relative pixel size around 3%. With this in mind, we could expect reasonable insight from FI in this case, just as we concluded from the noise study.

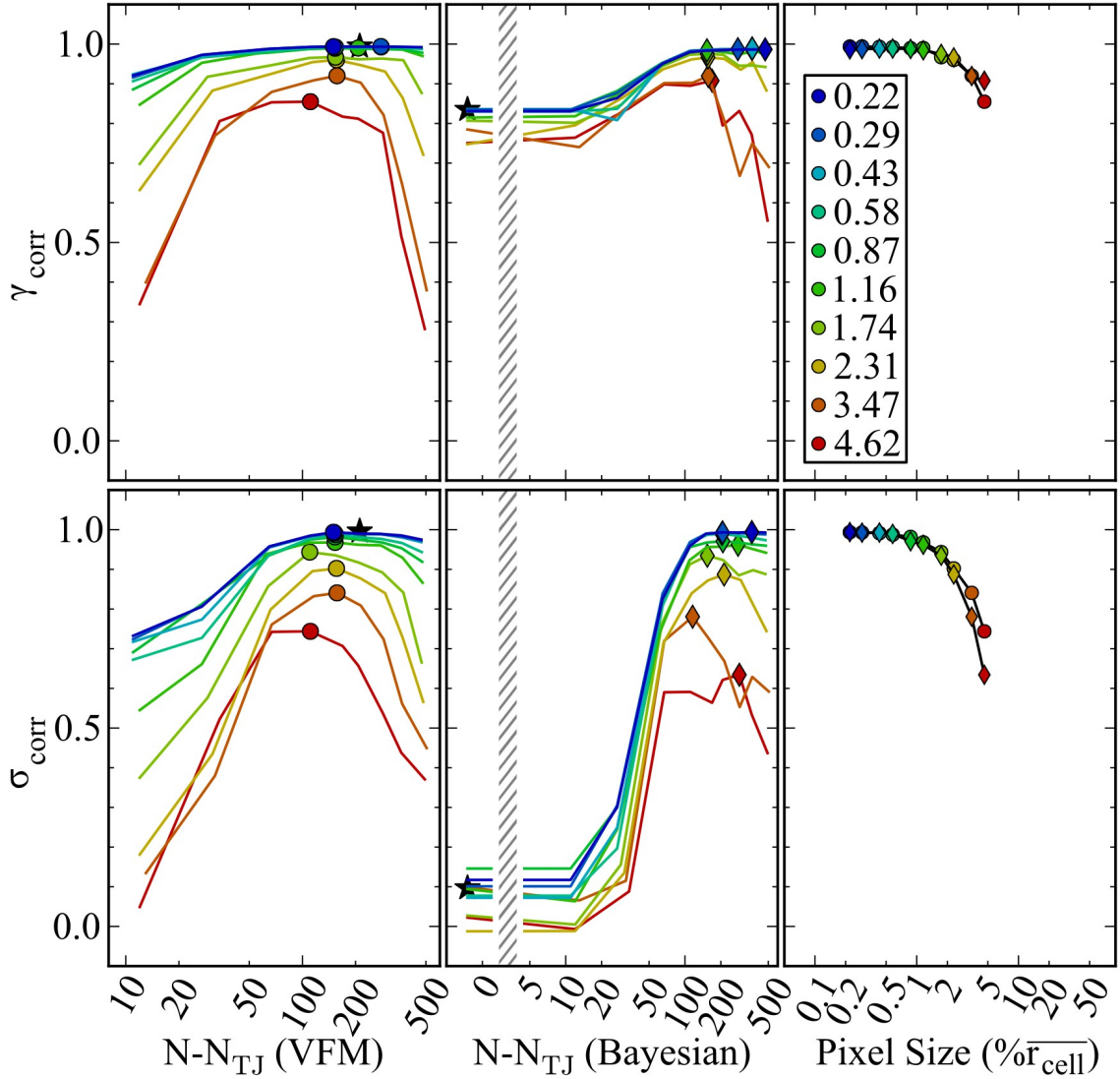


Figure 7.4: Effect of Rasterization on VFM.

Log-log plot of gold standard correlations for all trials in the rasterization study in the same style as Figure 7.2 . There is a strong trend toward lower correlation with larger relative pixel sizes (lower resolution), just as there was with larger amounts of noise. The same trend with added internal nodes is present here as well, making it clear that this is an effect that will be observed in any curved system and is not an artifact of the mesh interpolation.

We also looked at the different FI techniques and found that VFM on the unmodified mesh got closest to the gold standard solution with correlations of 0.994 and 0.996 for γ and σ respectively (plotted as black stars in Figure 7.2 and Figure 7.4). We

also tried VFM with viscous damping forces included, but this did not significantly change the correlations at all.

MI and BFI performed moderately in this case when used as described in Chiou 2012 and Ishihara 2012; MI did not calculate pressures and had a correlation of 0.924 in the tensions, and BFI had correlations of 0.835 and 0.097 for γ and σ . However, if internal nodes were used, the results for BFI were quite similar to VFM's (0.980 and 0.990). Interestingly, MI did almost identically whether internal nodes were used or not, but if the outer boundary was not included, the results became uncorrelated (-0.289). We also tried another variation of VFM where all line segments along each cell-cell interface were constrained to have a common value, (VFM with common edge tensions). This method had correlations of 0.834 and 0.679 (γ and σ) which seems surprisingly poor considering that the only difference was a constraint on internal tension variation.

In summary, FI techniques can be effective under the right circumstances, but current approaches are very sensitive to noise and other types of perturbations. The dependence on the number of internal nodes is critical to understanding how FI performs when trying to match large curvatures. Our initial assumption was that more nodes would create higher accuracy, but this is clearly balanced out by greater sensitivity to noise. Interestingly, as more nodes (or greater amounts of noise) are added, a common radial pattern emerges for all of the VFM solutions; an example of this is depicted inside the box in Figure 7.5 (corresponding to 2% coordinate noise added to the original mesh). To better understand this pattern, we decided to look at the most poorly constrained modes, which mathematically would be the eigenvectors of the geometry matrix with the smallest corresponding eigenvalues. The smallest 9 modes are presented in Figure 7.5 and it is clear that the radial pattern could be composed from these modes.

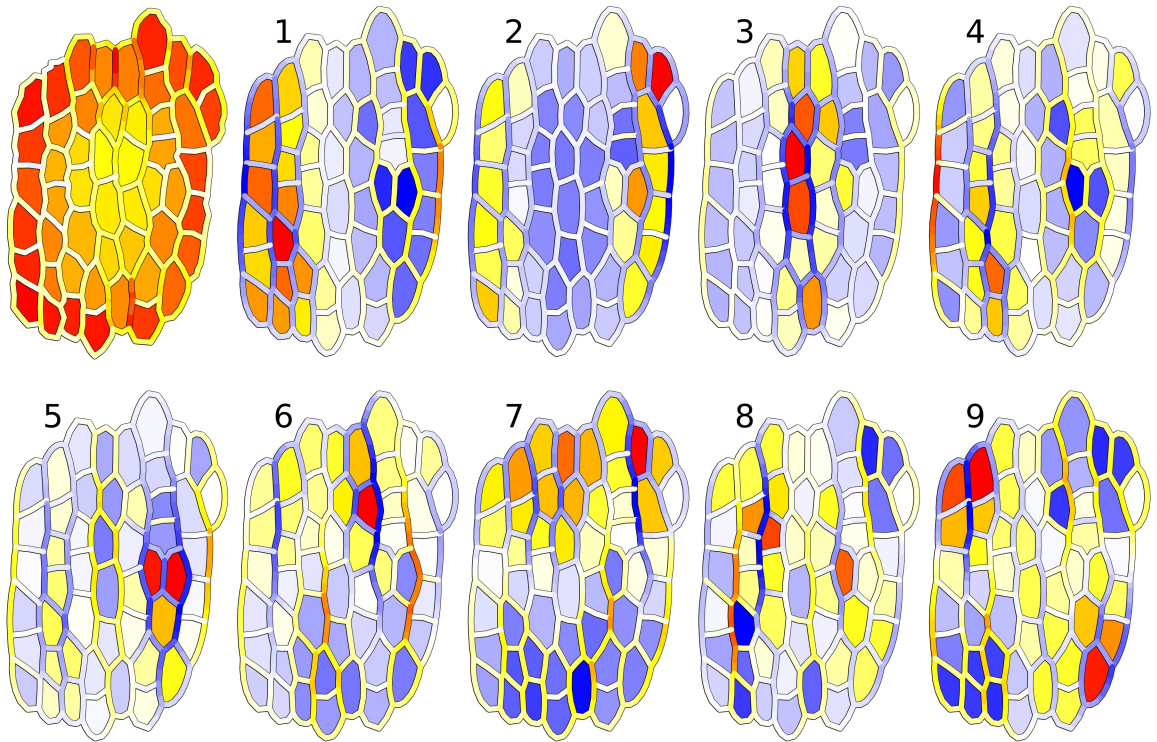


Figure 7.5: Poorly Constrained Modes of the Solution.

A VFM result from a noisy ($2\% r_{\text{cell}}$) simulation (upper left, boxed) and the eigenvectors of the geometry matrix that correspond to the smallest eigenvalues; these are the modes of the system that are the least constrained and most sensitive to noise. Interestingly, the modes 1 and 7 are the x and y gradient modes, 2-4 are bubble modes along x , and 9 is something of a "quadrupole" mode. We assert that the patterns on these in unconstrained modes could lead easily to radial gradient pattern that consistently emerges as more internal nodes are added to the set or more noise is added to the system (as in the boxed panel).

Chapter 8

8. Conclusions and Future Directions

Through the course of this research, we have reached some valuable conclusions:

- Amnioserosa cells, which were known to be pulsatile, have been shown to support wave propagation, though it is highly damped.
- The volume of amnioserosa cells is essentially constant during both cellular oscillations and during wound expansion — so long as the cell remains undamaged.
- Apical contractions seen in two dimensional images have a corresponding basolateral expansion This expansion is nearly prismatic but not exactly so; cell-cell boundaries are often askew from the apical-basal axis of the tissue.
- Force inference is a viable technique given the proper domain, and although it is very sensitive, it can be used to glean valuable insights.

In addition, we have also created a system that essentially solves the segmentation problem for cells in tissue so long as there are discernable boundaries. Because this system avoids specialized algorithms, it is flexible enough for any small-scale segmentation tasks and has been foundational not only for my work, but also for larger studies by our research group:

- Analyzing the dependence of wound expansion rate on the cellular pulsation cycle [3].
- Identifying cell orientation and aspect ratio in germ band retraction [81].
- Analyzing the highly extended shapes of amnioserosa cells during germ band retraction.

The 3D+time segmentation tool, while having fewer use cases, is equally simple and

flexible to use for small tasks in that domain.

This research has focused on two major phenomena in *Drosophila amnioserosa*: cellular pulsations and recoil patterns from laser point ablation. Both of these are easily captured with live imaging, and both are intimately related to the underlying internal forces in the epithelia: pulsations represent the natural cyclical variations in force in the tissue and recoil patterns provide a more fundamental measure of the strain at one point in space and time. Both processes are approachable with force inference (FI), which could potentially provide a rich map of mechanics in both space and time.

Studying natural pulsations with FI seems like a simple problem, but it turns out that techniques like VFM and even the newer CellFIT cannot adequately account for subcellular activity, especially in situations where cell-cell boundaries are long and crenulated (not simply a straight line or arc). FI also struggles with wounds, because the missing tissue is not simply an absent body pressure, but a hole in the apical network of the epithelium.

Clearly, FI will have to evolve to deal with these types of systems, which might be better modeled by a continuous, actively contracting sheet than a foam-style network of cell edges and cell bodies. The passive mechanics of such a system are well-established, with analytical solutions available even for hole drilling experiments. These analytical solutions are re-derived in Supplement S2, and the rest of this supplement describes preliminary work to analyze the ablation recoil patterns in *Drosophila* assuming this continuum model, potentially with nonlinear or viscoelastic properties.

Likewise, for pulsation studies, the obvious way to describe the changes in a sheet from one state to a later state is using a deformation field. We have tried large-scale automatic image registration using spline warps (UnwarpJ), but this did not yield quality

results for cadherin-GFP labeled amnioserosa cells because the cell interiors lack brightness variations, preventing effective image registration. Nonetheless, the cell edge positions are information rich and should be fertile ground for force inference techniques.

Since we can now easily generate a segmentation from an image using SeedWater Segmenter, it is possible to get basic registration between images based on the triple junctions. Figure 8.1 shows the synthesis of segmentation and warping ideas to extract as much information as possible from time-lapse image pairs.

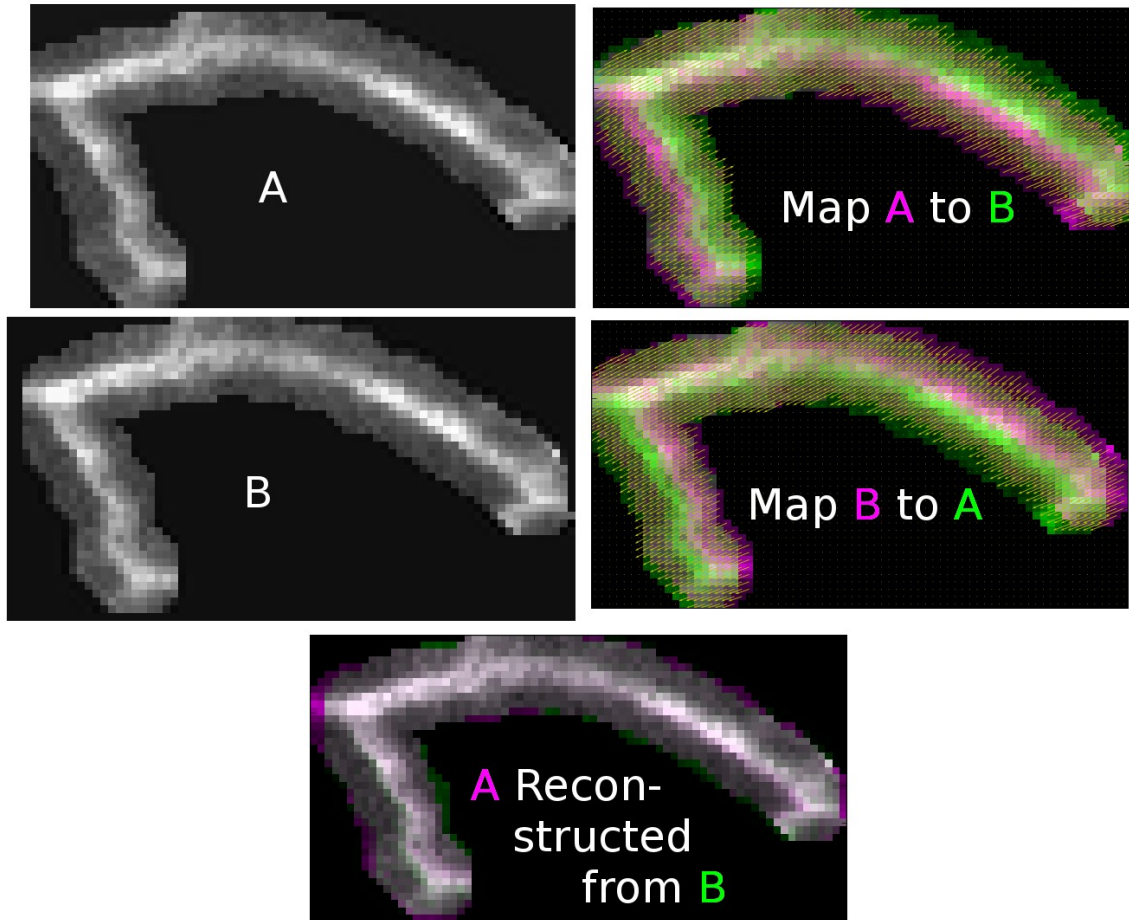


Figure 8.1: Registering the motion of a cell-cell boundary from one frame to another. A and B (left) show the original data for an individual cell-cell boundary extracted from two frames of a larger time series. This edge undergoes an extremely subtle motion from one frame to the next, and the shape of the edge and brightness variation along the edge can both be used to build a highly accurate picture of this motion. The two right panels show the deformation field (fine yellow arrows) that maps A to B or B to A, using magenta and green color channels to show the two images simultaneously. The bottom panel shows the reconstruction for the B to A map, showing the original image for A in the magenta channel and the reconstruction (A') generated from B in the green channel. The relatively uniform (and nearly white) color along the cell edges show that this process was extremely accurate.

In theory, if one had a displacement map for all the edges in a larger image, the interior could be approximately inferred by interpolation. Doing so could possibly not only answer the question of whether cellular pulsations occur in subcellular regions

(which we currently presume but cannot yet prove), but would even allow this phenomenon to be studied over wide spatial scales.

This work has been part of a long line of research trying to bridge the gap between physics and biology. Specifically, this contribution is a continuation of research designed to probe embryonic mechanics in novel ways, create stable and comprehensible computational tools and software, and build creative methods for data extraction and inference.

Supplement 1

S1. Three-dimensional shape changes associated with apical contraction cycles

Portions of this chapter were published in A.K. Jayasinghe, S.M. Crews, D.N. Mashburn, M.S. Hutson. "Apical Oscillations in Amnioserosa Cells: Basolateral Coupling and Mechanical Autonomy." *Biophysical Journal*, vol. 105, pp. 255-265, 2013 [3].

Although both the experiments and their matching simulations in Jayasinghe *et al.* imply pulsations of amnioserosa cells that are strongly mechanically autonomous, this autonomy requires what seems like a very strange mechanical situation: epithelial cells with a net internal force that is directed outwards, i.e., an in-plane compressive stress. Two possible sources of this outward force are pressurization of the cell's cytoplasm or coupling between the apical and basal surfaces of the cell. We thus imaged the three-dimensional structure of pulsing amnioserosa cells using the Resille(117-2)-GFP strain [151], [155], which fluorescently labels all cell borders. These cells undergo apical oscillations, albeit with smaller amplitudes and more variable periods. Despite these differences, Resille-GFP enables segmentation of the entire cell volume and analysis of volumetric measures such as cytoplasmic flow or the relationship between apical and basal contraction.

Three-dimensional reconstructions show that these cells are not rigidly prismatic in shape. Instead, there are considerable dynamic changes in the basolateral portion of the cell (Figure S1.1), including wedging of the cell walls, formation of bulges, and rippling of the basal surface.

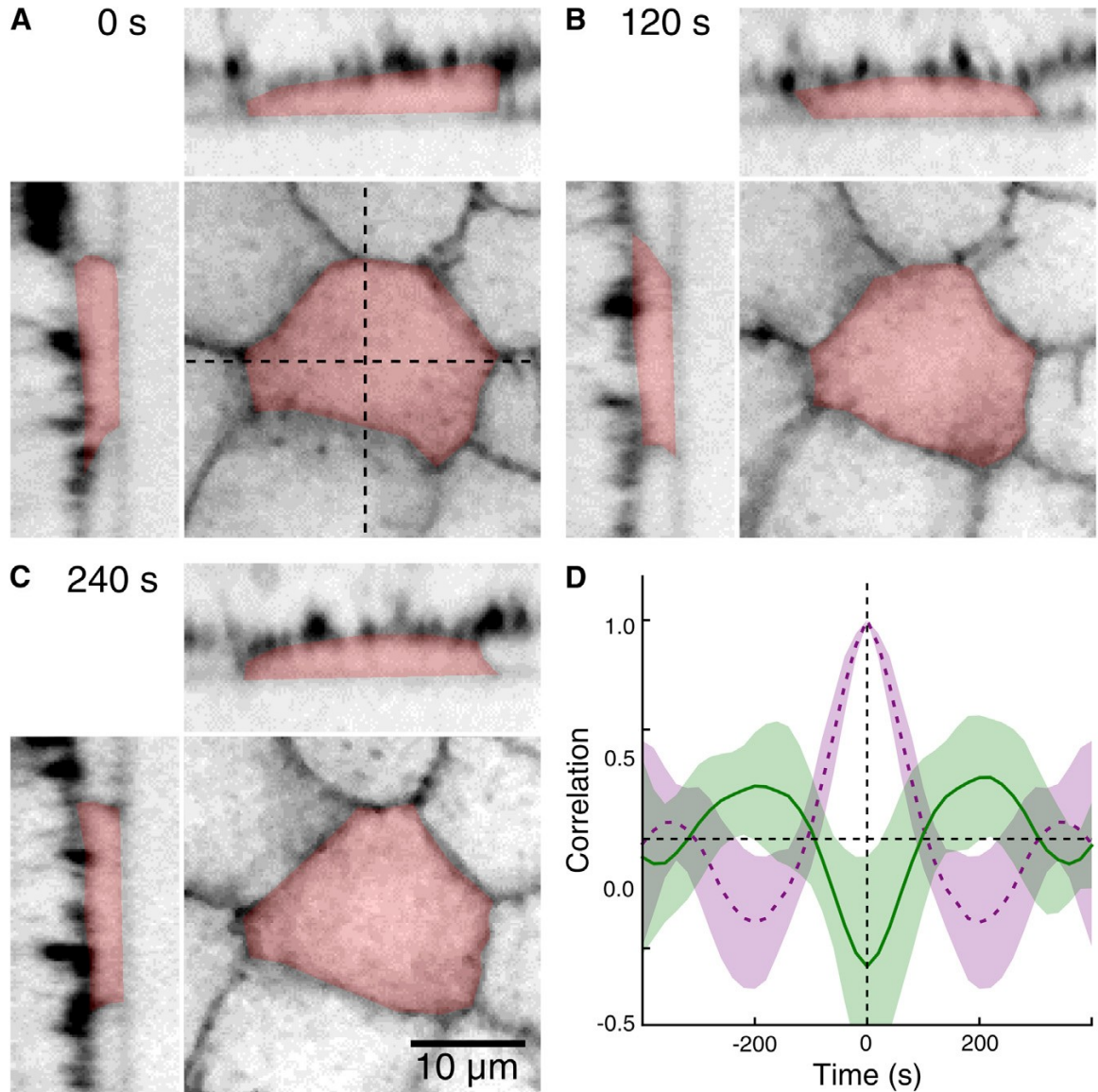


Figure S1.1: Three-dimensional dynamic changes in amnioserosa cell shape. (A–C) Three views are shown for each time point: an xy view of the apical area (bottom right); an xz cross-section (top); and a yz cross-section (left). (Darker/red shading) Extent of one cell. The rougher, outermost surface in each cross-section corresponds to the basal surface. A common scale bar is shown in panel C. (D) Changes in apical area and average cell thickness are anticorrelated: mean area versus thickness cross-correlation (green, solid, $N=6$); mean auto-correlation of apical area (purple, dotted). The full set of apical area and thickness versus time graphs is compiled in Figure S1.2 B. One of the cells in Figure S1.2 B was excluded from the mean correlation functions because its autocorrelation function showed no evidence of oscillation.

To determine whether these complex basolateral dynamics are related to the apical

area oscillations, we calculated cross-correlations between cells' apical area and average thickness. Apical area was defined as the in-plane area of a cell at the z position of its adherens junctions. This z position was calculated by hand, selecting a starting value for each cell and then correcting for drift by subtracting off the average z motion of all cells. Average thickness was defined as the ratio of a cell's total volume to its maximum projected area in the xy plane. We analyzed each of seven cells over a time span of 1100 s. Autocorrelations of the apical areas showed that six of these seven cells were oscillating (periods of 320, 300, 380, 400, 460, and 260 s), but the small amplitudes and variable periods lead to an average autocorrelation (of these six) with only a weak and broad secondary peak (Figure S1.1 D, dashed). Despite these weak oscillations, the cross-correlations showed a clear anticorrelation between apical area and average cell thickness both in the average correlation function (Figure S1.1 D, solid) and in the individual zero-delay correlation coefficients: -0.97, -0.93, -0.81, -0.80, -0.79, -0.70, and 0.53. The one exceptional cell was oscillating weakly, so we have no explanation for its discrepant behavior. On the whole, these results suggest that volume pushed away from the apical surface during a constricted phase of the cycle is collected in the basolateral domain and returned to the apical domain in the next half cycle. Such pressurized flow of cytoplasm is a plausible source for the cell-internal compressive stress. To insure that this observation was not dependent on our exact choice of apical and basal metrics, we performed similar cross-correlation analysis for volume above and below the adherens plane, as well as apical plane volume versus remaining cell volume. All three analyses yielded similar results and are compiled in Figure S1.2 along with the individual traces of these cell metrics.

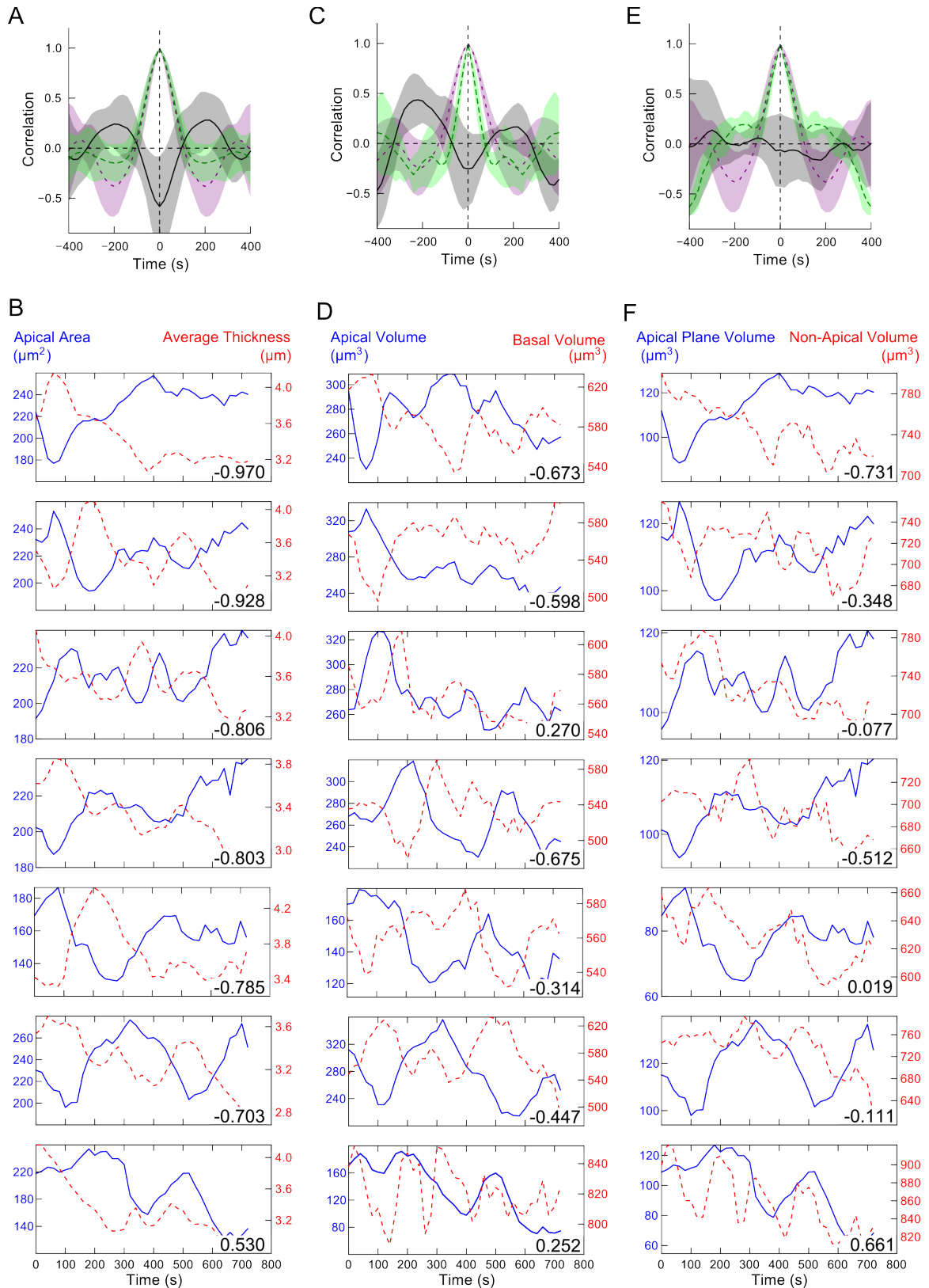


Figure S1.2: Correlations of volumetric measurements for seven individual amnioserosa cells.

(Caption continued on next page)

(A-B) Apical area and average cell thickness. (C-D) Volume above and below the apical plane. (E-F) Apical plane volume (apical area times z spacing) and remaining volume (total volume minus apical plane volume). A, C, and E show the mean auto-correlation of the apical measure (purple, dotted) and the cross-correlation of the two respective measures (green, solid). Shaded regions are \pm one standard deviation (based on $N = 6$). B, D, and F show the full set of graphs for each cell with the zero-delay value of the cross-correlation shown in the lower right corner of each subpanel. The rows (individual cells) are sorted so the cross-correlations of area and cell thickness (left column) are ordered from most negative to most positive. In each graph, the apical measure is solid blue and the other measure is dashed red. The cell shown in the top row of B, D, and F showed no evidence of oscillations in its auto-correlation and was excluded from the means in A, C, and E. The approximate periods of apical area oscillations in the remaining cells were 320, 300, 380, 400, 460 and 260 s, respectively from top to bottom. A and B show that there is a strong anti-correlation between apical area and cell thickness, implying that contractions in the apical xy -plane result in an extrusion in z and vice-versa. C and D show that as volume in the apical region of the cell shrinks, this volume generally moves to the basal portion of the cell and vice-versa. E and F show a weak anti-correlation between volume in just the adherens-level plane and that in the rest of the cell; however, the standard deviation in this last cross-correlation is large enough that the weak anti-correlation is not significantly different from zero.

Our data cannot distinguish whether the return of cytoplasm to the apical surface is a passive response to basolateral pressure built up during the previous apical contraction or is actively driven by a basal actomyosin contraction. Nonetheless, it is instructive to estimate the order of magnitude for the pressure differences needed to drive such flows. One can do so using either considerations of Poiseuille flow [156] or the classic Stefan solution for axisymmetric squeezing flow [157], [158]. Using the latter, the maximum pressure difference in a cylinder undergoing creeping flow deformation is

$$\Delta P = \frac{3\dot{H}R^2}{4H^3} \quad (\text{S1.1})$$

where η is viscosity, H is cylinder half-height, and R is radius. Using appropriate dimensions for amnioserosa cells ($H \sim 3 \mu\text{m}$, $R \sim 10 \mu\text{m}$) and a viscosity estimate from sea urchin embryos ($\eta \sim 10 \text{ Pa}\cdot\text{s}$ [159]), the very slow rates observed for oscillatory cell

thickness change ($\dot{H} < 0.02 \mu\text{m/s}$) imply a maximum pressure difference of $\sim 0.1 \text{ Pa}$, equivalent to a pressure gradient $\sim 0.1 \text{ kPa/cm}$. This is on the low end of pressure gradients observed in other cytoplasmic flows, *e.g.*, hyphal growth in fungi is driven by pressure gradients of $0.05\text{--}10 \text{ kPa/cm}$ [156]. Although we cannot rule out a basal actomyosin contraction as contributing to the return of fluid to the apical surface, the small values of these pressure gradients suggest such contraction is not needed.

Supplement 2

S2. Spring Network Models of Embryonic Tissue

S2.1. Epithelial elastic sheet model

The simple elastic sheet model for epithelial tissue permits the use of known analytical solutions for elastic hole drilling. This model uses standard assumptions from mechanical engineering: linear elasticity, small deformations, small displacements, and an infinitely large sheet. The general case is for a circular hole drilled at the origin of an elastic sheet under uniform, anisotropic stress (different stresses in the x and y directions), resulting in a release of tension and a new equilibrium configuration. We have used this analytical model to compute the relative displacements between the cut solution and the original stressed solution. This was then fed into an image warping algorithm where the parameters were manually tuned to match up with *in vivo* laser hole drilling experiments we performed in *Drosophila melanogaster* amnioserosa tissue.

S2.2. Derivation of Relative Displacements After Hole Drilling

In order to solve the analytical equations, we first solve for a completely uniaxial stress (one axis has zero stress) and then calculate the biaxial solution by superposition. Below we follow this derivation from Timoshenko [4], [29]. Using the summation convention with comma derivatives and starting with body force vector ρb_i , acceleration vector a_i , and stress tensor σ_{ij} , Newton's Second Law for continuum mechanics is:

$$\int_R \rho b_i dV + \int_{\partial R} \sigma_{ij} \cdot n_i dA = \int_R \rho a_i dV \quad (\text{S2.1})$$

for any contiguous region of space, R . V , A , and n_i are volume, surface area, and outward surface normal respectively. To transform middle term, we use Gauss' Law:

$$\int_{\partial R} \sigma_{ij} \cdot n_i dA = \int_{\partial R} \sigma \cdot d\vec{A} = \int_R \sigma_{ij,j} dV \quad (\text{S2.2})$$

With this, Equation S2.1 simplifies to:

$$\int_R (\rho b_i + \sigma_{ij,j} - \rho a_i) dV = 0 \quad (\text{S2.3})$$

and then just:

$$\rho b_i + \sigma_{ij,j} - \rho a_i = 0 \quad (\text{S2.4})$$

In the absence of body forces and at equilibrium, b_i and a_i are zero, so:

$$\sigma_{ij,j} = 0 \quad (\text{S2.5})$$

For 2D problems,

$$\sigma_{ij} = \begin{pmatrix} \sigma_x & \tau_{xy} & 0 \\ \tau_{yx} & \sigma_y & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (\text{S2.6})$$

and by symmetry,

$$\tau_{xy} = \tau_{yx} \quad (\text{S2.7})$$

This means that the equilibrium stresses equations come out as:

$$\begin{aligned} \frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} &= 0 \\ \frac{\partial \sigma_y}{\partial y} + \frac{\partial \tau_{xy}}{\partial x} &= 0 \end{aligned} \quad (\text{S2.8})$$

Next, we use the standard definitions for normal and shear strains:

$$\epsilon_x = \frac{\partial u_x}{\partial x}, \epsilon_y = \frac{\partial u_y}{\partial y}, \gamma_{xy} = \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \quad (\text{S2.9})$$

And relate stresses and strains using the standard equations of linear elasticity:

$$\begin{aligned} \epsilon_x &= \frac{1}{E}(\sigma_x - \nu \sigma_y) \\ \epsilon_y &= \frac{1}{E}(\sigma_y - \nu \sigma_x) \\ \gamma_{xy} &= \frac{2(1+\nu)}{E}(\tau_{xy}) \end{aligned} \quad (\text{S2.10})$$

where ν is Poisson's ratio and E is Young's Modulus

Now we can derive the compatibility condition, a simple derivative relationship

between normal strain and shear strain based on their definitions in terms of displacement:

$$\frac{\partial^2 \epsilon_x}{\partial y^2} + \frac{\partial^2 \epsilon_y}{\partial x^2} = \frac{\partial^2 \gamma_{xy}}{\partial x \partial y} \quad (\text{S2.11})$$

Using the compatibility condition and the equilibrium stress relations, we can derive the equation:

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) (\sigma_x + \sigma_y) = 0 \quad (\text{S2.12})$$

This equation and the equilibrium stress equations now form a complete system of differential equations that we can solve. In solving these equations, it is simplest to define a stress function, a function whose derivatives are the normal and shear stresses:

$$\sigma_x = \frac{\partial^2 \phi}{\partial y^2}, \sigma_y = \frac{\partial^2 \phi}{\partial x^2}, \tau_{xy} = \frac{\partial^2 \phi}{\partial x \partial y} \quad (\text{S2.13})$$

This leads to a fourth-order differential equation:

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \left(\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} \right) = 0 \quad (\text{S2.14})$$

Our problem, however, is in polar coordinates, so without going into the details of the complex derivation of the elastic sheet with hole, I first want to simply take the solution without a hole obtained by nothing more than a simple change in basis. We start with σ_x but no stress in the y direction and no shear. Next, we use a coordinate transformation and obtain:

$$\begin{aligned} \sigma_r &= \sigma_x \cos^2(\theta) = \frac{\sigma_x}{2}(1 + \cos(2\theta)) \\ \sigma_\theta &= \sigma_x \sin^2(\theta) = \frac{\sigma_x}{2}(1 - \cos(2\theta)) \\ \tau_{r\theta} &= -\sigma_x \sin(\theta)\cos(\theta) = -\frac{\sigma_x}{2}\sin(2\theta) \end{aligned} \quad (\text{S2.15})$$

Without going into details, we can then solve the polar form of the stress function with a hole of radius R_0 to obtain:

$$\phi(r, \theta) = \left(\frac{\sigma_x r^2}{4} - \frac{\sigma_x R_0^2}{2} \ln(r)\right) + \left(\frac{\sigma_x R_0^2}{2} - \frac{\sigma_x}{4} r^2 - \frac{\sigma_x R_0^4}{4} \frac{1}{r^2}\right) \cos(2\theta) \quad (\text{S2.16})$$

We can then obtain the stresses for the hole situation by taking derivatives. Then, subtracting the solution without the hole (obtained either by coordinate transformation or using the hole solution with $R_0=0$) we obtain the relaxed stresses:

$$\begin{aligned} \sigma_r &= \frac{\sigma_x}{2} \left(-\frac{R_0^2}{r^2}\right) + \frac{\sigma_x}{2} \left(\frac{3R_0^4}{r^4} - \frac{4R_0^2}{r^2}\right) \cos(2\theta) \\ \sigma_\theta &= \frac{\sigma_x}{2} \left(\frac{R_0^2}{r^2}\right) - \frac{\sigma_x}{2} \left(\frac{3R_0^4}{r^4}\right) \cos(2\theta) \\ \tau_{r\theta} &= -\frac{\sigma_x}{2} \left(-\frac{3R_0^4}{r^4} + \frac{2R_0^2}{r^2}\right) \sin(2\theta) \end{aligned} \quad (\text{S2.17})$$

By solving for the displacements using the equations above (and some known solutions for integrals), we obtain:

$$\begin{aligned} u_{r, \text{nohole}} &= \frac{r\sigma_x}{2E} (1 - \nu) - \frac{r\sigma_x}{2E} (1 + \nu) \cos(2\theta) \\ u_{\theta, \text{nohole}} &= -\frac{r\sigma_x}{2E} (1 + \nu) \sin(2\theta) \end{aligned} \quad (\text{S2.18})$$

and:

$$\begin{aligned} u_{r, \text{relative}} &= u_r - u_{r, \text{nohole}} = A\sigma_x + B\sigma_x \cos(2\theta) \\ u_{\theta, \text{relative}} &= u_\theta - u_{\theta, \text{nohole}} = C\sigma_x \sin(2\theta) \end{aligned} \quad (\text{S2.19})$$

with:

$$\begin{aligned} A &= \frac{R_0}{2E} (1 + \nu) \frac{R_0}{r} \\ B &= \frac{R_0}{2E} \left(\frac{4R_0}{r} - \frac{(1+\nu)R_0^3}{r^3}\right) \\ C &= -\frac{R_0}{2E} \left(\frac{2(1-\nu)R_0}{r} + \frac{(1+\nu)R_0^3}{r^3}\right) \end{aligned} \quad (\text{S2.20})$$

We can add back σ_y which yields a solution that is rotated 90 degrees from the x solution.

The combined result is:

$$\begin{aligned} u_{r, \text{relative}} &= A(\sigma_x + \sigma_y) + B(\sigma_x - \sigma_y) \cos(2\theta) \\ u_{\theta, \text{relative}} &= C(\sigma_x - \sigma_y) \sin(2\theta) \end{aligned} \quad (\text{S2.21})$$

We can also easily add a rigid body solution, too, which is important for its use in the warping algorithm:

$$\begin{aligned}
 u_{r,rigidbody} &= u_0 \cos(\theta - \theta_0) \\
 u_{\theta,rigidbody} &= u_0 \sin(\theta - \theta_0) + r \Delta\theta
 \end{aligned}
 \tag{S2.22}$$

S2.3. Results of the Warping Algorithm

As stated above, the warping algorithm we developed allows us to transform an image of the embryo after the cut into an approximately similar arrangement as before the cut. Figure S2.1 shows the transformation of a cell edge cut and a cell center cut. The agreement is very good except near the hole. Once we have aligned the displacement field, we can compute values for the strains.

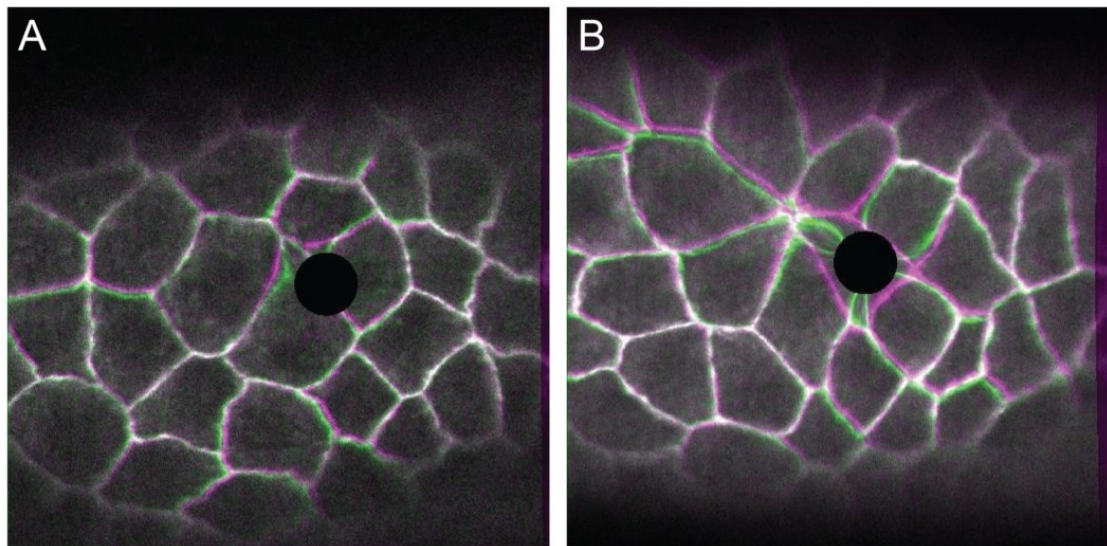


Figure S2.1: Warps for Cell Edge Cut and Cell Center Cut.

The original (uncut) image is shown in magenta and the warped cut image is shown in green. Where the two overlap, the colors combine to make white. The estimated hole is blacked out by a circle. The agreement of the image produced by the warping algorithm with the original is remarkable. However, it is obvious that the algorithm tends to over-warp the area near the hole. This is especially clear in part B with its strange artifacts (green).

When we assume the hole is 30 pixels in radius and the tissue has a Poisson ratio of 0.33, we can get the ratio of stress to E, and the values for the two experiments

pictured in Figure S2.1 are 1.26 and 2.43 respectively. The strains we get directly from the displacement field, and they are 0.8 and 1.6 respectively (zero being completely relaxed). The anisotropy is almost negligible, at values of 0.01 and 0.02.

S2.4. One Dimensional Model

As clean and simple as the analytical model is, it has some major flaws and leaves some questions unanswered. The main discrepancy between the analytical model and experiments is that the model predicts too much deformation near the hole as compared to the rest of the tissue. Also, since it is an analytical solution, there is no way to add complexity without making the problem intractable. In attempting to reconcile the model and experiments, three important points came to light:

1. The solution makes a small deformation approximation that could possibly lead to an error.
2. There could be some nonlinearity in the elasticity of the tissue (x^3 term in the restoring force).
3. The dynamic viscoelastic/viscoplastic behavior of the tissue might lead to a significant difference from simple linear elasticity.

Of these, the first two are most easily tested, and the simplest model to form insight into this system is a simple 1D picture. In this scenario, we have a rubber band tied to another rubber band such that there are four rubber lines (springs) coming out of the knot. When we stretch the two sets of springs in opposite directions until they become stiff, this creates strain hardening, with a negative x^3 coefficient (same direction as the linear Hookean term). If we cut one of the springs, it causes there to be a release of tension in the opposite two springs and an increase in tension of the adjacent spring (Figure S2.2 ,

Figure S2.3 , and Figure S2.4).

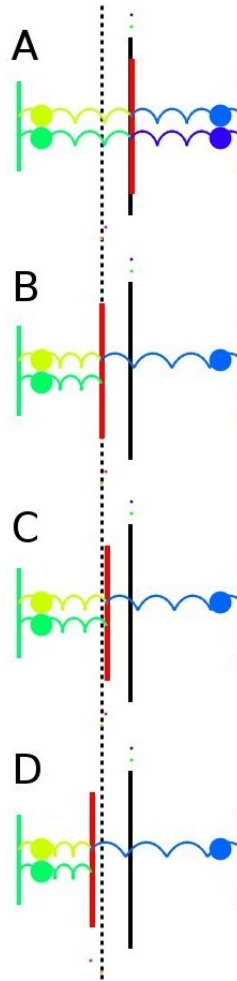


Figure S2.2: Spring Diagram.

This figure shows a graphical solution to the coupled spring equations. The black bar represents the origin, the red bar represents the massless attachment point for the four springs. The colored dots represent the equilibrium positions of the springs, and the respective colored bars and wavy lines are added purely for visual appeal. Springs are offset vertically for visualization only. All springs have the same spring constant and the nonlinear term varies between the models. (A) Purple spring is uncut, so the system is in perfect equilibrium no matter what the nonlinear constant is. (B) Linear solution with spring cut. The dotted line shows this position for reference. (C) Nonlinear solution with “hard” springs. Note that equilibrium now favors the individual spring (right) rather than the pair. (D) Nonlinear solution with “soft” springs. Notice now that the solution favors the pair of springs (left)

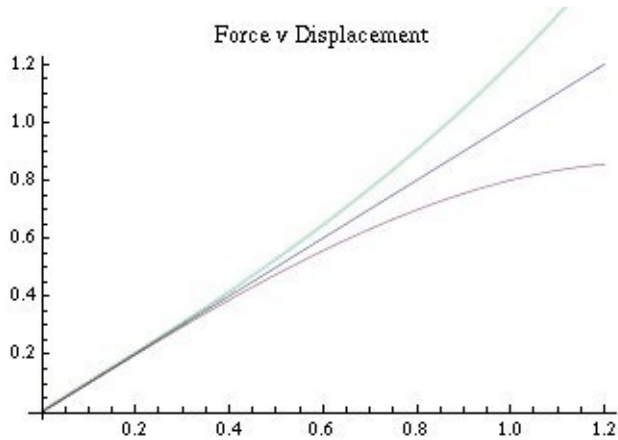


Figure S2.3: Force v. Displacement Curve for linear and nonlinear springs. The central curve is for a pure Hookean spring. The upper curve has a nonlinear constant of 0.2 (strain hardening) and the lower curve has a nonlinear constant of -0.2 (strain softening). These curves correspond to Figure S2.2 C, Figure S2.2 B, and Figure S2.2 D respectively (top to bottom).

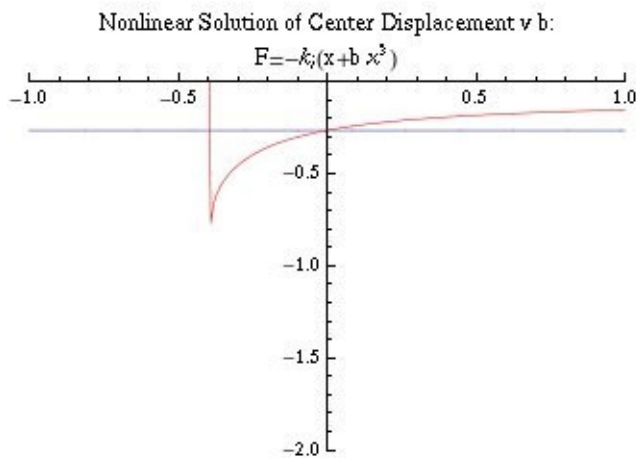


Figure S2.4: Equilibrium Position v Nonlinear Constant. This shows the dependence of the equilibrium position on the nonlinear constant. The blue curve is the reference linear solution. For positive nonlinear constant, the system is strain hardened. For negative nonlinear constants, the system is strain softened. The asymptote represents the point where the nonlinear term overtakes the linear one, destabilizing the system and making it come apart.

This is analogous to what we see when we drill a hole. What we are doing is

destroying the radial component of the elasticity within the hole radius, thereby increasing the tangential tension around the hole, but releasing radial tension. This is just a loose analogy (see Figure S2.5).

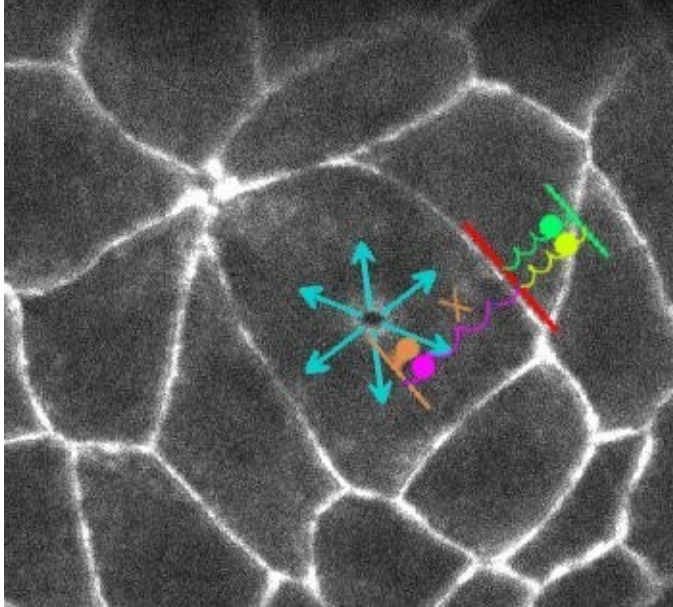


Figure S2.5: Connection to 2D Experiments.

This figure shows a schematic picture of the connection between this four-spring/ rubber band model and the experiments we have done. The cut spring represents the radial spring that gets destroyed and the adjacent spring represents the surrounding theta springs that gain extra tension. The two opposite springs (effectively one spring with twice the spring constant) represent the surrounding radial springs that lose tension. This figure also highlights the looseness of this analogy.

The beauty of this model is that it allows the exploration of the effects of strain hardening and strain softening while still being simple to model and analyze. In fact, even implementing this simple model was rather complex because it required the use of a nonlinear polynomial solver. In summary, with equal starting conditions for the four springs (same relaxed lengths, stretched lengths, and spring constants) a third order nonlinear term added a noticeable change that can be summarized as follows:

- When the nonlinear term was positive (strain hardening), the knot was pulled more

toward the single spring after the cut.

- When the nonlinear term was negative (strain softening), the knot was pulled more toward the pair of springs after the cut. See Figure S2.2 , Figure S2.3 , and Figure S2.4 > for details.

Although this result was an interesting proof of concept, this model is not complete enough to be useful in discerning the discrepancy in our data. For one thing, the analogy is very loose and so no quantitative analysis is possible. Second, certain competing factor cannot be distinguished from one another. For instance it is possible by adjusting the equilibrium lengths and the nonlinear factor to create scenarios that would be indistinguishable from these simple cutting experiments (see Figure S2.6). The main problem is that tracking only a single point of data lacks enough information to determine these quantities.

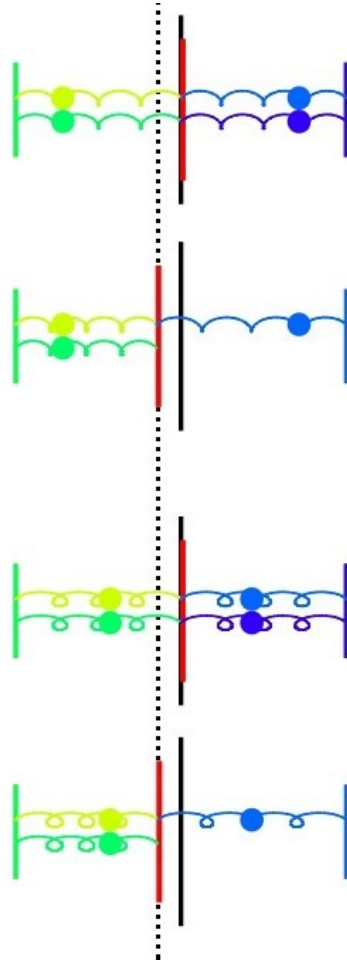


Figure S2.6: Simple Model Cannot Distinguish Between Competing Factors. Simply making a cut and watching the motion of a single point cannot distinguish between greater nonlinearity (A and B) and different equilibrium lengths (C and D). From the outside, there is no way to distinguish these situations.

S2.5. Radial spring networks and nonlinear behavior

What we really need is a quantitative way to characterize the entire 2D displacement field and relate it to either a linear or nonlinear solution.

In order to better investigate this matter, we developed a discretized, radially symmetric model based on a polar decomposition of the elasticity into radial and tangential elements. One major advantage to this model is, unlike the analytical model, it does not include a small deformation approximation. Another major advantage is that this

model in numerical, so it is possible (though much more computationally demanding) to add the same nonlinearity as in the rubber band model.

In order to properly model this system, we first have to describe the effect of radial forces on radial position. To do this, we define effective radial and tangential spring constant based on a rectangular approximation. Figure S2.7 shows a schematic version of this discretization into rings (though it fails to properly show the radial symmetry and continuous nature of each separate ring). Figure S2.8 then shows a linearized version of this with effective spring constants labeled as symbols.

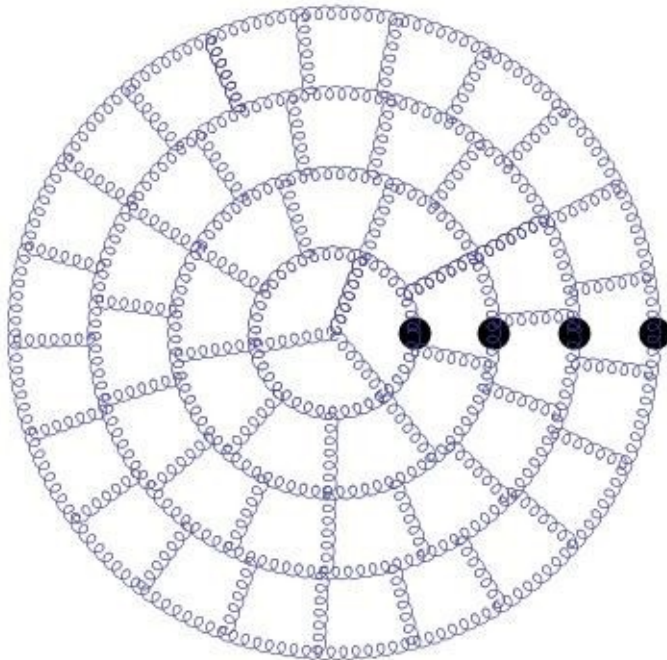


Figure S2.7: Polar Decomposition of 2D Sheet.

Schematic representation of the radially symmetric (1D) model. Black dots are the nodes in the model. "Hoop" springs represent the tangential component of the elasticity. "Spokes" springs (which really should be pictured as a continuum around each ring) represent the radial components of the elasticity. Another way to picture this is as two sheets glued together, one cut into rings and one cut into very thin pie slices.

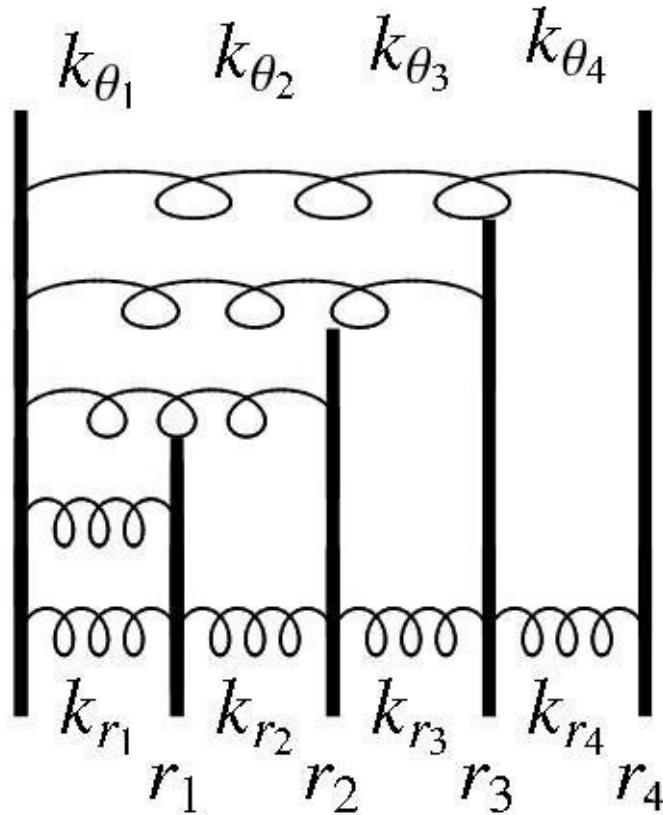


Figure S2.8: Linearized Spring Network.

This is the same set of springs as in the ring diagram. The nodes r_1 - r_4 are the same as the black dots. The longer springs (top) are the same as the hoop springs and the shorter springs (bottom) are the same as the spoke springs. The spring constants for the radial and tangential rings are derived from a rectangular approximation. Especially for the tangential springs, a conversion must be made between internal tension and force delivered at the nodes.

S2.5.1. Effective Spring Constants

In order to calculate the effective spring constants for the rings in either the radial or tangential direction, I first assumed that the springs were independent in the r and θ directions, meaning that I am assuming a Poisson ratio of zero. Next, I used the fact that the radial and tangential springs could be approximated as rectangular, each with the same elastic modulus. The formula relating spring constants and elastic modulus is linear with the area of the material (thickness, t_0 * unit depth) and inversely proportional to

length:

$$k = \frac{Et_0}{L_0} \quad (\text{S2.24})$$

For a radial spring, the length is

$$r_n - r_{n-1} \quad (\text{S2.25})$$

and the effective thickness is the circumference of the average radius circle

$$2\pi \frac{r_{n-1} + r_n}{2} \quad (\text{S2.26})$$

This leads to an effective spring constant of:

$$k_{r_{eff}} = \frac{E\pi(r_{n-1_0} + r_{n_0})}{r_{n_0} - r_{n-1_0}} \quad (\text{S2.27})$$

Here I have used the initial values for the nodes because the spring constant depends only on the initial values of the nodes.

The θ spring is slightly more complex. It requires first knowing its effective length (circumference of the circle) and width (thickness):

$$\frac{2\pi r_n}{\frac{r_{n+1} - r_{n-1}}{2}} \quad (\text{S2.28})$$

It also requires converting from internal spring tension to radial force. This conversion, which can be easily obtained via a polygon method with the number of sides tending to infinity, yields:

$$k_{\theta_{eff}} = \frac{E\pi(r_{n+1_0} - r_{n-1_0})}{r_0} \quad (\text{S2.29})$$

It is intriguing that forms of these two types of elements are so similar.

S2.5.2. Model Construction

In order to construct the model, we first need to describe the forces for each spring. These are given by:

$$\begin{aligned} F_{r_n} &= k_{r_n} (\Delta r_n - \Delta r_{n-1}) \\ F_{\theta_n} &= k_{\theta_n} \Delta r_n \end{aligned} \quad (\text{S2.30})$$

We can then simply add (or subtract) the forces at each node due to each attached spring and use Newton's Second Law to set this sum equal to zero (except at the outside). If we then rearrange the sums and factor out the separate Δr_n 's so that we can write the solution as a matrix. An example matrix is given below. This matrix describes the spring network pictured in Figure S2.8 . Spring constants are then calculated via the formulas derived in part (i) above. Solving the linear system is no more complicated than inverting this matrix and multiplying by the right hand side.

$$\begin{pmatrix} -k_{r_1} - k_{r_2} - k_{\theta_1} & k_{r_2} & 0 & 0 \\ k_{r_2} & -k_{r_2} - k_{r_3} - k_{\theta_2} & k_{r_3} & 0 \\ 0 & k_{r_3} & -k_{r_3} - k_{r_4} - k_{\theta_3} & k_{r_4} \\ 0 & 0 & k_{r_4} & -k_{r_4} - k_{\theta_4} \end{pmatrix} \cdot \begin{pmatrix} \Delta r_1 \\ \Delta r_2 \\ \Delta r_3 \\ \Delta r_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -F_{ext} \end{pmatrix} \quad (\text{S2.31})$$

In Figure S2.9 , the result from a simulation run with 20 nodes is pictured. Only 10 nodes are shown as the outer half of the model is there to better approximate the infinite continuum of the analytical model and minimize end effects. Figure S2.10 shows the computed stresses from the same simulation before and after the cut, making it a good comparison to the rubber band model. In Figure S2.11 , taking the same initial parameters, good agreement is shown between the analytical solution and this simple harmonic network model.

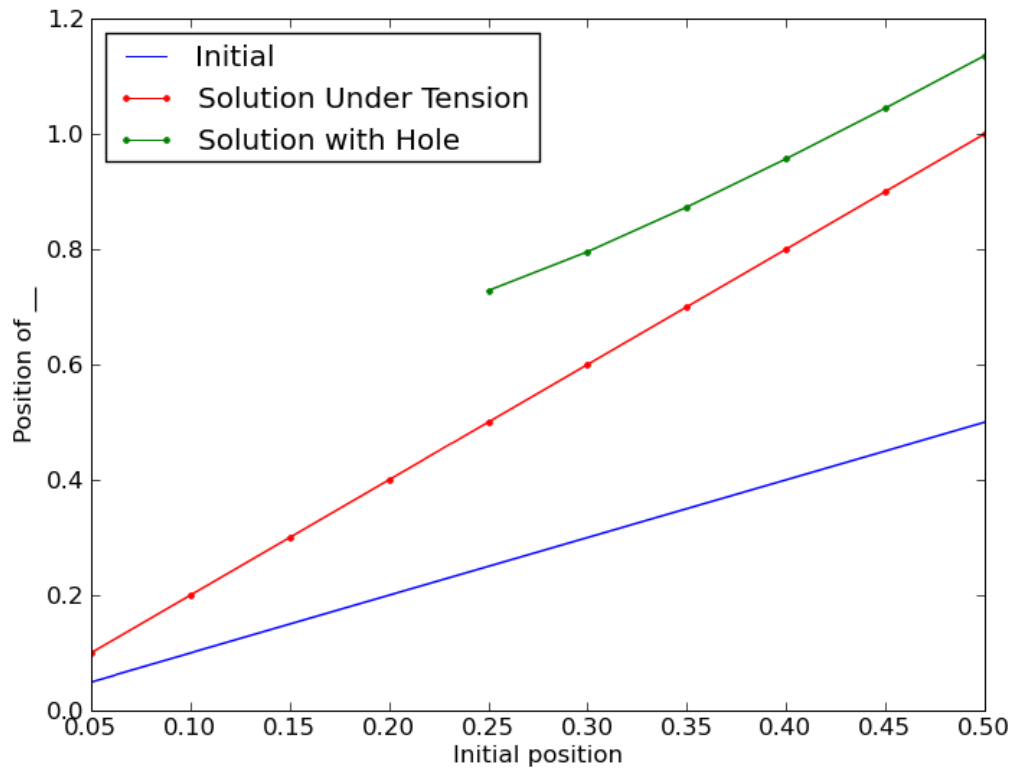


Figure S2.9: Radial Spring Network Solutions.

This figure shows a solution of the radially symmetric simple harmonic model with simple elasticity. The blue line is a simple $y=x$ line to show the unstressed positions of every point. The red line is the stressed positions of every point. The green line is the positions after a cut. In order to minimize end effects, the model is run with twice the pictured length, which more closely approximates the infinite continuum used in the analytical model.

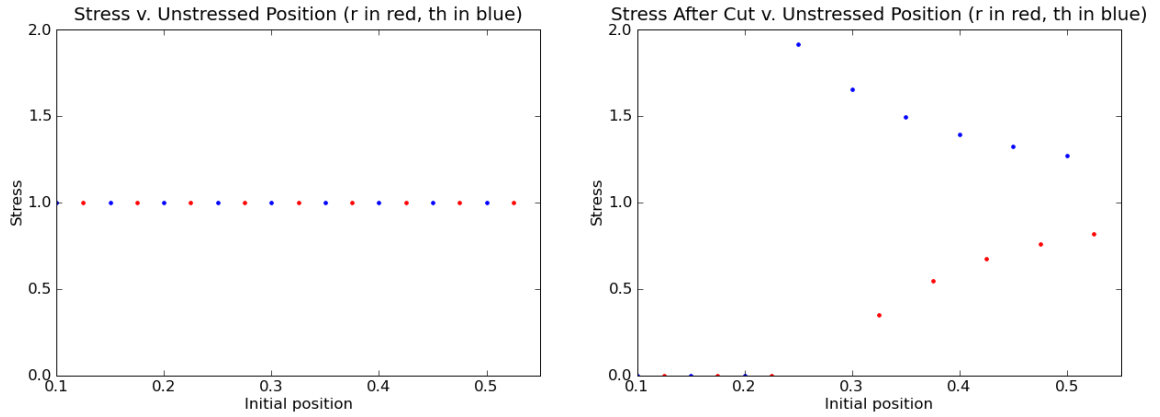


Figure S2.10: Computed stress on each node before and after the cut.

After computing the forces due to each radial and tangential ring element, it is possible to work out the stresses on the springs. As would be expected, before the cut, all the springs are in equilibrium, and since the force was chosen appropriately, the value is one. Radial springs are pictured in red and tangential springs are pictured in blue. The springs are interlaced with radial springs centered between nodes and tangential springs centered on nodes (see Figure S2.7). After the cut we see the tension created in the theta springs and the tension released in the radial springs, just as with the rubber band model.

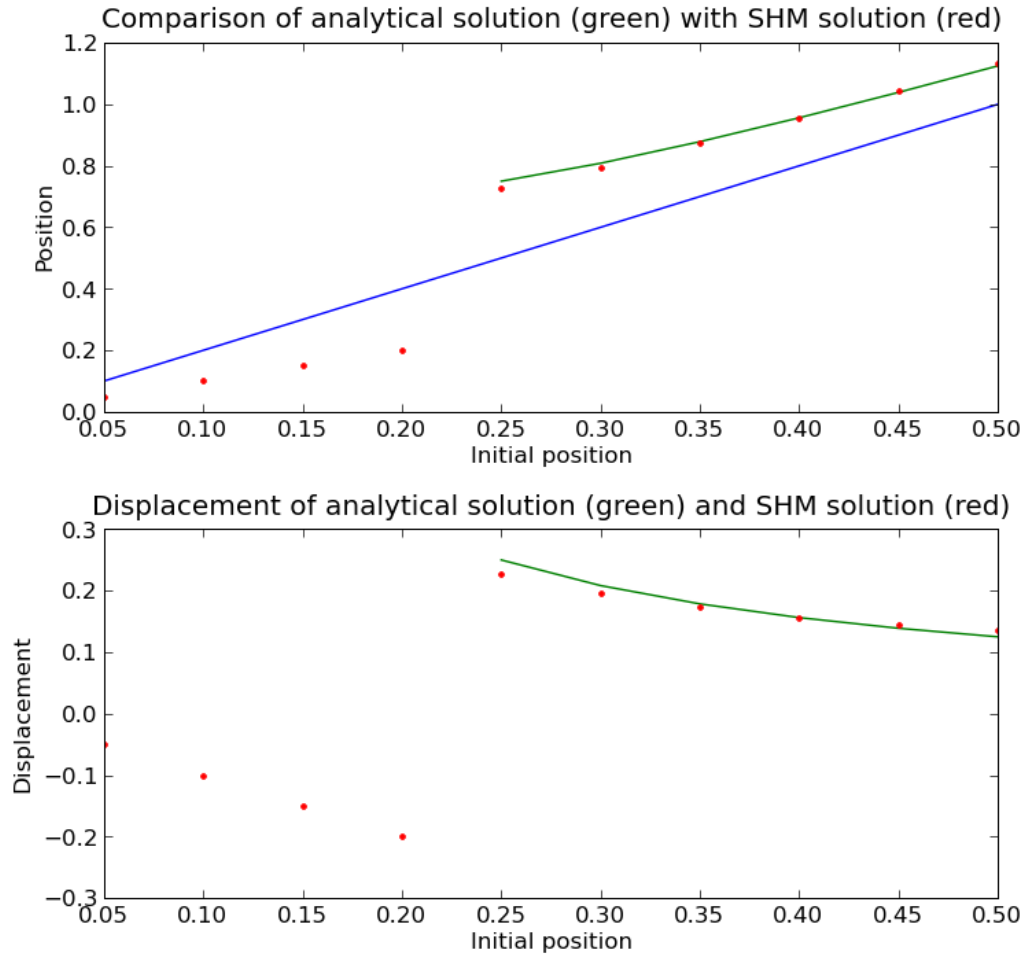


Figure S2.11: Comparison of Polar Spring Network and Analytical Model. The agreement between the Radial Spring Network model (red dots) and the analytical model (green line) run with radial symmetry, a poisson ratio of zero, and the equivalent elastic modulus and initial stress is remarkably good. Other than an end effect (probably attributable to the finite nature of the model) the models are very similar. The top graph shows the agreement in position and the bottom graph shows the agreement in displacement. As before, the blue line at the top is just $y=x$ for reference.

Adding in nonlinearity is a bit more complicated, as no matrix construction is possible. The forces have the forms:

$$\begin{aligned} F_{r_n} &= k_{r_n} (\Delta r_n - \Delta r_{n-1}) + \text{nonlin} \times k_{r_n} (\Delta r_n - \Delta r_{n-1})^3 \\ F_{\theta_n} &= k_{\theta_n} \Delta r_n + \text{nonlin} \times (2\pi)^2 k_{\theta_n} (\Delta r_n)^3 \end{aligned} \quad (\text{S2.32})$$

Where nonlin is the nonlinear factor and the $(2\pi)^2$ factor in the tangential spring is a constant that comes out of the conversion between tension and radial force.

Since it is impossible to construct a linear matrix for the nonlinear solution, I simply constructed a set of polynomials and fed them into a preexisting nonlinear solver. I originally attempted to use Mathematica's `NSolve`, but quickly discovered that as it generates not only the one real solution, but also all complex solutions, making it painstakingly slow to use. For the models presented here, I rewrote the program in Python and used the nonlinear solver in the python module `Sympy` (a simpler, real-only iterative solver) to construct the nonlinear solutions. I used the linear solution as an initial guess to speed up the solver further. Figure S2.12 shows the same data as Figure S2.9 plus many equivalent nonlinear solutions. Our hope is that by comparing these nonlinear curves with our experimental data, we will be able to characterize this nonlinearity factor in *Drosophila* tissue.

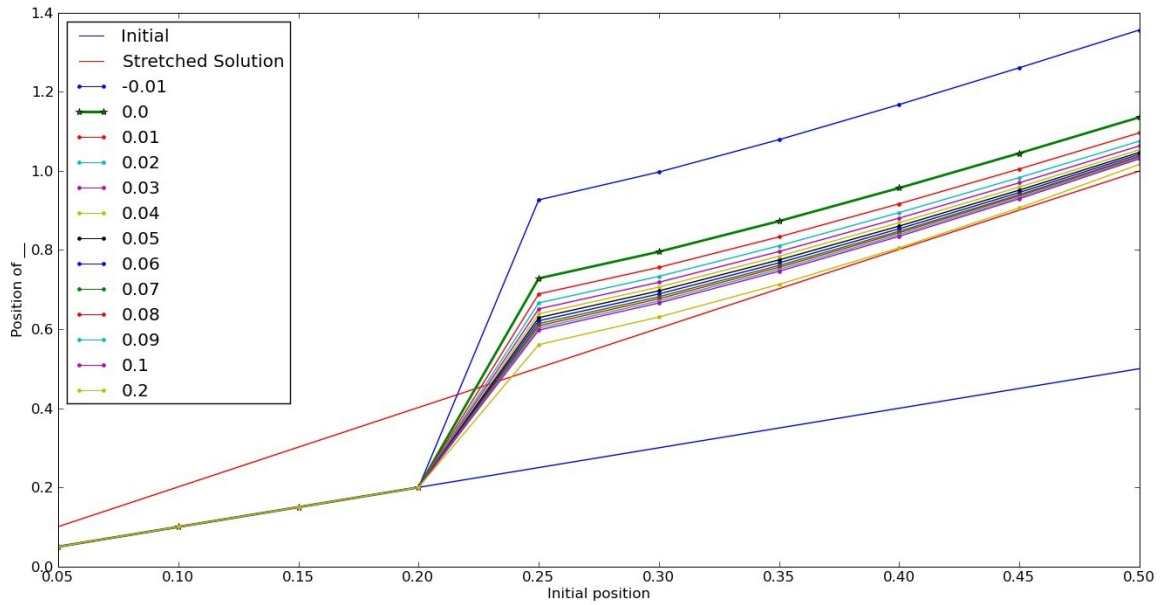


Figure S2.12: Family of Cuts with different nonlinear factors.

The solid blue and red lines as well as the thick green line are the same as in Figure S2.9 . The other curves represent the same simulation but with varying nonlinear factors. The force is carefully adjusted so that all the stretched, uncut solutions match precisely with the red line (to within one part in a thousand). Nonlinear factors outside the range pictured here led to instabilities in the solution and are not pictured. We hope to use this model to compare the experimental data and determine what, if any, nonlinear factor is observed in our hole drilling data.

Supplement 3

S3. Segmenting a wound in 3D+time

Here, we present some sample images from SWS4D while in action.

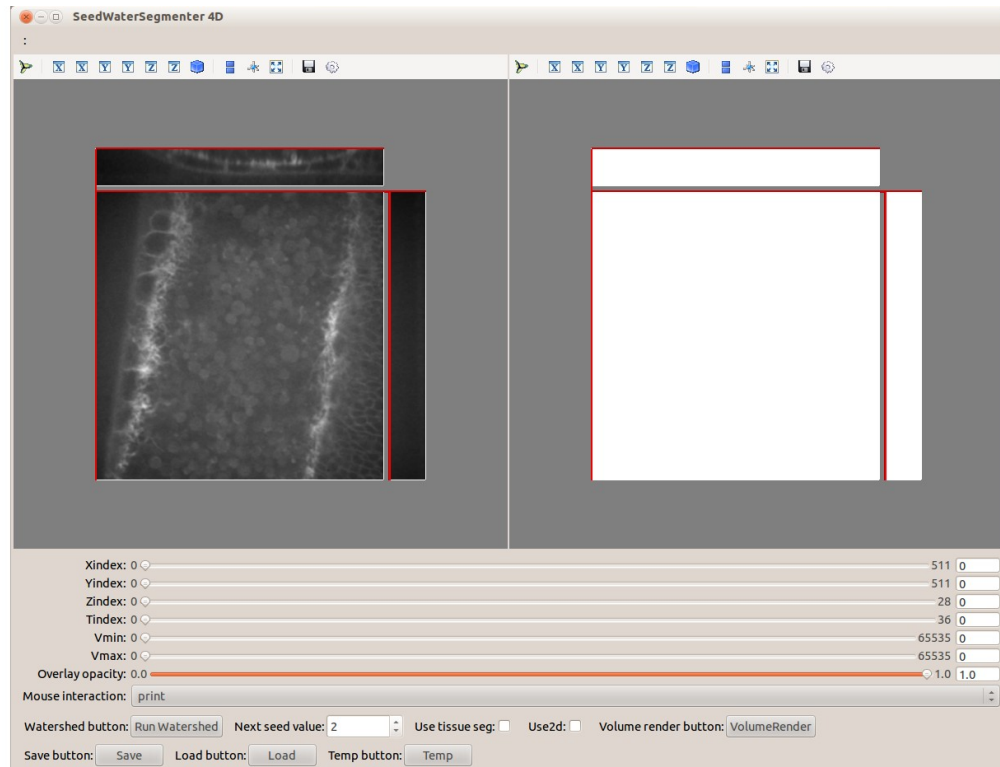


Figure S3.1: Initial view after opening SWS4D.

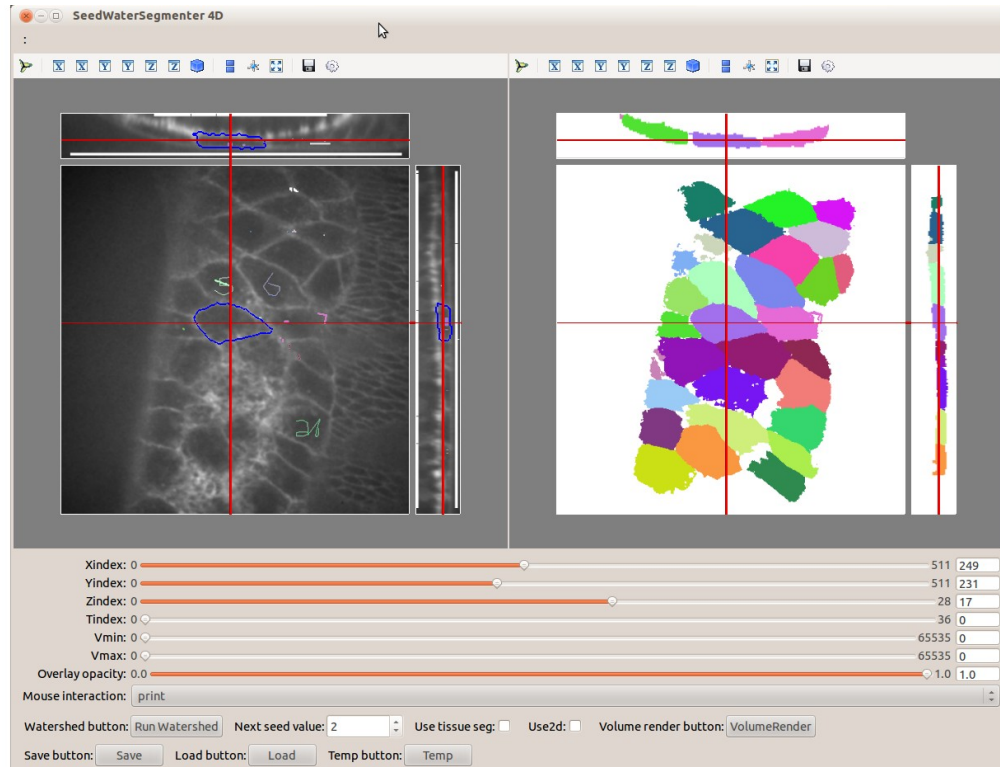


Figure S3.2: Same dataset after segmentation is finished.

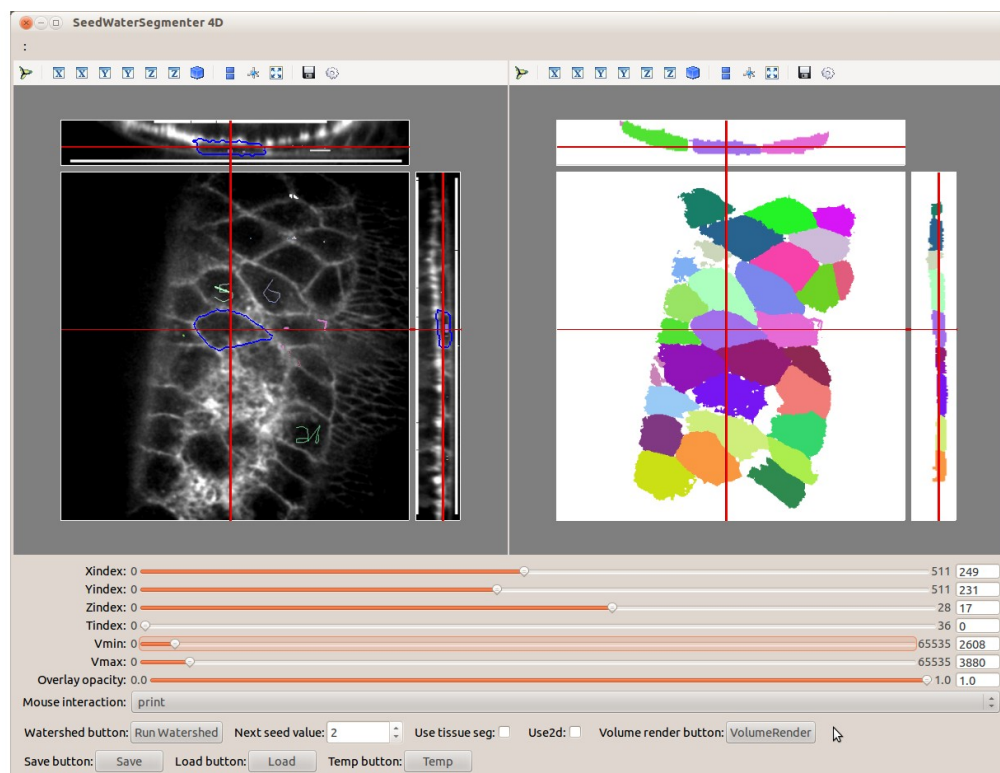


Figure S3.3: Example of changing the contrast using the Vmin/Vmax sliders.

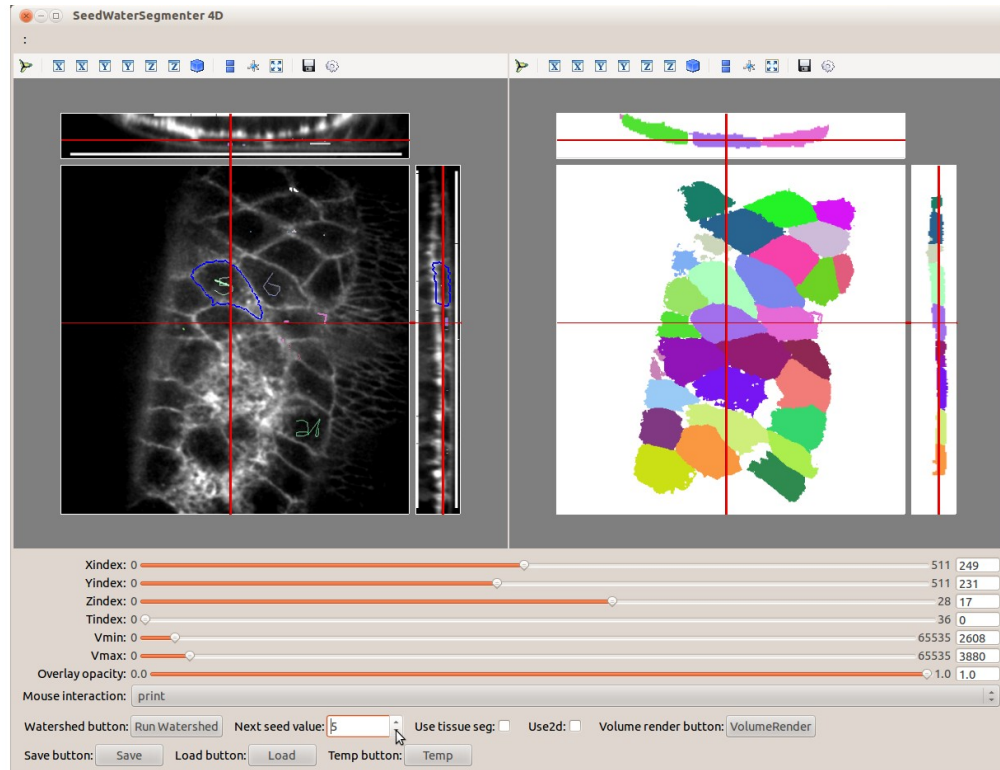


Figure S3.4: Example of changing the current cell to ID 5, which also changes the outlined cell.

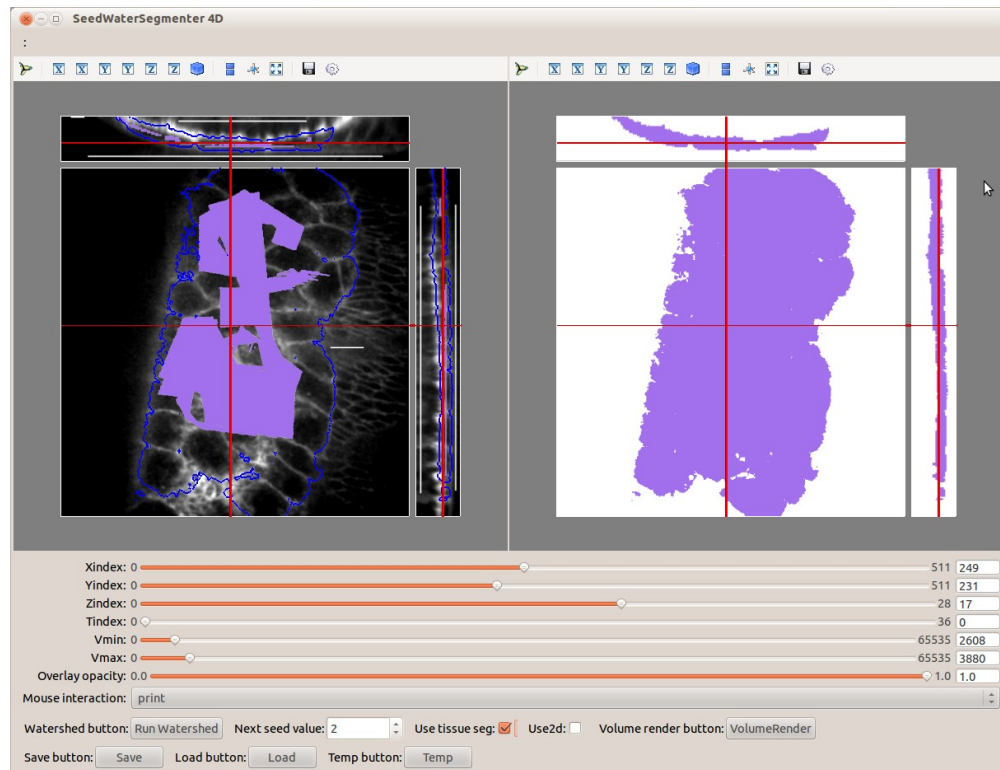


Figure S3.5: Sample use of Tissue Segmentation.

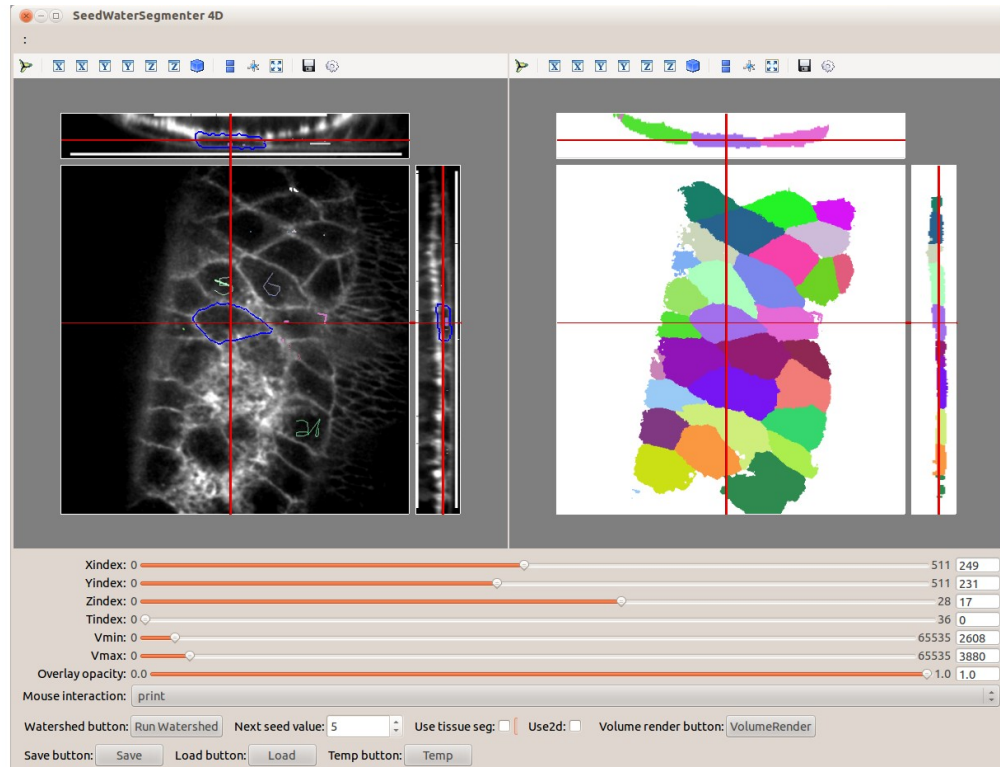


Figure S3.6: Updated segmentation using the Tissue Segmentation from Figure S3.5 as a mask.

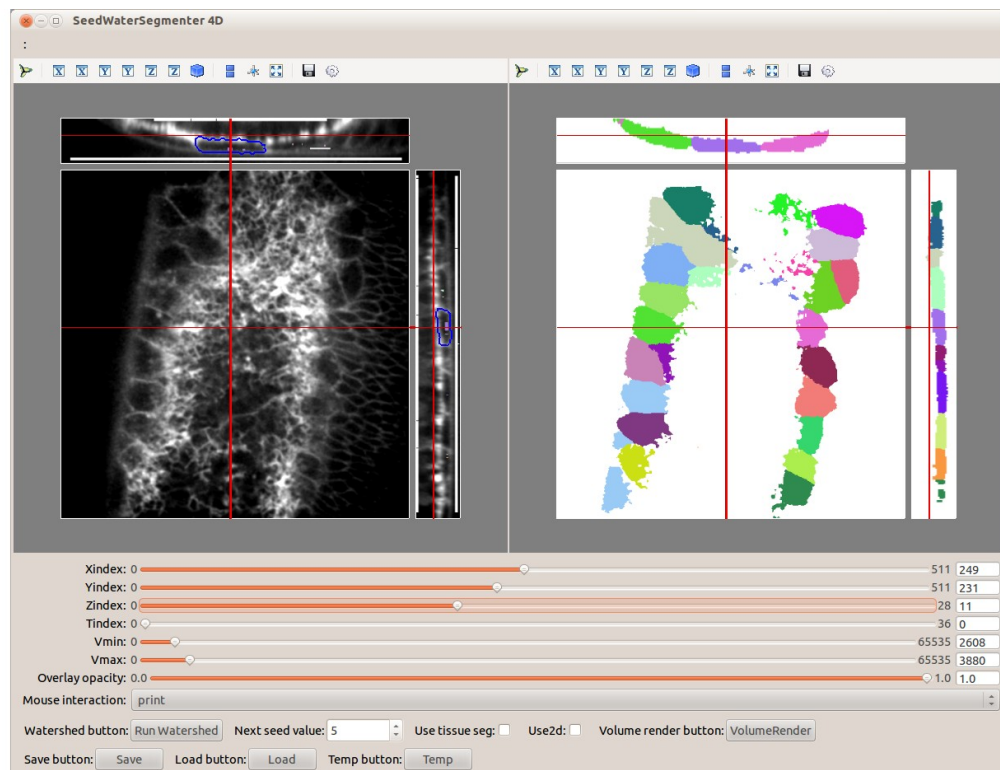


Figure S3.7: Example of changing the Z-index to go deeper into the tissue.

REFERENCES

- [1] G. W. Brodland, V. Conte, P. G. Cranston, J. Veldhuis, S. Narasimhan, M. S. Hutson, A. Jacinto, F. Ulrich, B. Baum, and M. Miodownik, “Video force microscopy reveals the mechanics of ventral furrow invagination in *Drosophila*,” *Proc. Natl. Acad. Sci.*, vol. 107, no. 51, pp. 22111–22116, Dec. 2010.
- [2] D. N. Mashburn, H. E. Lynch, X. Ma, and M. S. Hutson, “Enabling user-guided segmentation and tracking of surface-labeled cells in time-lapse image sets of living tissues,” *Cytometry A*, vol. 81A, no. 5, pp. 409–418, 2012.
- [3] A. K. Jayasinghe, S. M. Crews, D. N. Mashburn, and M. S. Hutson, “Apical Oscillations in Amnioserosa Cells: Basolateral Coupling and Mechanical Autonomy,” *Biophys. J.*, vol. 105, no. 1, pp. 255–265, Jul. 2013.
- [4] S. P. Timoshenko and J. N. Goodier, “Theory of elasticity,” *N. Y.*, pp. 350–360, 1951.
- [5] M. S. Hutson and X. Ma, “Mechanical aspects of developmental biology: perspectives On Growth and Form in the (post)-genomic age,” *Phys Biol*, vol. 5, no. 015001, p. 015001, 2008.
- [6] K. D. Irvine and E. Wieschaus, “Cell intercalation during *Drosophila* germband extension and its regulation by pair-rule segmentation genes,” *Development*, vol. 120, no. 4, pp. 827–841, 1994.
- [7] M. Zajac, G. L. Jones, and J. A. Glazier, “Model of convergent extension in animal morphogenesis,” *Phys. Rev. Lett.*, vol. 85, no. 9, pp. 2022–2025, 2000.
- [8] A. Rodriguez-Diaz, Y. Toyama, D. L. Abravanel, J. M. Wiemann, A. R. Wells, U. S. Tulu, G. S. Edwards, and D. P. Kiehart, “Actomyosin purse strings: renewable resources that make morphogenesis robust and resilient,” *HFSP J.*, vol. 2, no. 4, pp. 220–237, Aug. 2008.
- [9] W. Wood, A. Jacinto, R. Grose, S. Woolner, J. Gale, C. Wilson, and P. Martin, “Wound healing recapitulates morphogenesis in *Drosophila* embryos,” *Nat. CELL Biol.*, vol. 4, no. 11, pp. 907–912, Nov. 2002.
- [10] X. G. Peralta, Y. Toyama, M. S. Hutson, R. Montague, S. Venakides, D. P. Kiehart, and G. S. Edwards, “Upregulation of forces and morphogenic asymmetries in dorsal closure during *Drosophila* development,” *Biophys. J.*, vol. 92, no. 7, pp. 2583–2596, 2007.
- [11] A. Jacinto, S. Woolner, and P. Martin, “Dynamic Analysis of Dorsal Closure in *Drosophila* From Genetics to Cell Biology,” *Dev. Cell*, vol. 3, no. 1, pp. 9–19, 2002.

- [12] N. Harden, “Signaling pathways directing the movement and fusion of epithelial sheets: lessons from dorsal closure in *Drosophila*,” *Differentiation*, vol. 70, no. 4–5, pp. 181–203, 2002.
- [13] J. Solon, A. Kaya-Çopur, J. Colombelli, and D. Brunner, “Pulsed forces timed by a ratchet-like mechanism drive directed tissue movement during dorsal closure,” *Cell*, vol. 137, no. 7, pp. 1331–1342, 2009.
- [14] G. B. Blanchard, S. Murugesu, R. J. Adams, A. Martinez-Arias, and N. Gorfinkiel, “Cytoskeletal dynamics and supracellular organisation of cell shape fluctuations during dorsal closure,” *Development*, vol. 137, pp. 2743–2752, Jul. 2010.
- [15] D. Azevedo, M. Antunes, S. Prag, X. Ma, U. Hacker, G. W. Brodland, M. S. Hutson, J. Solon, and A. Jacinto, “DRhoGEF2 Regulates Cellular Tension and Cell Pulsations in the Amnioserosa during *Drosophila* Dorsal Closure,” *PLoS ONE*, vol. 6, p. e23964, Sep. 2011.
- [16] K. A. Edwards, M. Demsky, R. A. Montague, N. Weymouth, and D. P. Kiehart, “GFP-moesin illuminates actin cytoskeleton dynamics in living tissue and demonstrates cell shape changes during morphogenesis in *Drosophila*,” *Dev. Biol.*, vol. 191, no. 1, pp. 103–117, Nov. 1997.
- [17] Q. Wang, J. J. Feng, and L. M. Pismen, “A Cell-Level Biomechanical Model of *Drosophila* Dorsal Closure,” *Biophys. J.*, vol. 103, no. 11, pp. 2265–2274, 2012.
- [18] C. Meghana, N. Ramdas, F. M. Hameed, M. Rao, G. V. Shivashankar, and M. Narasimha, “Integrin adhesion drives the emergent polarization of active cytoskeletal stresses to pattern cell delamination,” *Proc. Natl. Acad. Sci.*, vol. 108, no. 22, pp. 9107–9112, May 2011.
- [19] F. Schock and N. Perrimon, “Cellular processes associated with germ band retraction in *Drosophila*,” *Dev. Biol.*, vol. 248, no. 1, pp. 29–39, 2002.
- [20] M. S. Hutson, J. Veldhuis, X. Ma, H. E. Lynch, P. G. Cranston, and G. W. Brodland, “Combining Laser Microsurgery and Finite Element Modeling to Assess Cell-Level Epithelial Mechanics,” *Biophys. J.*, vol. 97, no. 12, pp. 3075–3085, Dec. 2009.
- [21] H. E. Lynch, J. Veldhuis, G. W. Brodland, and M. S. Hutson, “Modeling cell elongation during germ band retraction: cell autonomy versus applied anisotropic stress,” *New J. Phys.*, vol. 16, no. 5, p. 055003, May 2014.
- [22] W. T. McCleery, S. M. Crews, D. N. Mashburn, J. Veldhuis, G. W. Brodland, and M. S. Hutson, “Modeling the Morphogenesis of Epidermal Tissues on the Surface of a 3D Last,” *Bull. Am. Phys. Soc.*, 2014.
- [23] C. S. Adams MD and Adams MD, Celniker SE, Holt RA, et al., “The genome sequence of *Drosophila melanogaster*,” *Sci. J.*, vol. 287, no. 5461, pp. 2185–95,

Mar. 2000.

- [24] C. elegans Sequencing Consortium, “Genome sequence of the nematode *C. elegans*: a platform for investigating biology,” *Science*, vol. 282, no. 5396, pp. 2012–2018, Dec. 1998.
- [25] D. F. Poulson, “Histogenesis, organogenesis and differentiation in the embryo of *Drosophila melanogaster* Meigen,” *Biol. Drosoph.*, pp. 168–274, 1950.
- [26] S. P. Bainbridge and M. Bownes, “Staging the metamorphosis of *Drosophila melanogaster*,” *J. Embryol. Exp. Morphol.*, vol. 66, no. 1, pp. 57–80, 1981.
- [27] J. A. Campos-Ortega and V. Hartenstein, *The embryonic development of Drosophila melanogaster*. Springer Berlin, 1997.
- [28] S. F. Gilbert, “Early *Drosophila* Development,” 2000. [Online]. Available: <http://www.ncbi.nlm.nih.gov/books/NBK10081/>. [Accessed: 12-Jun-2014].
- [29] J. Lubliner, “Plasticity theory,” *N. Y. Macmillan C1990*.
- [30] D. Roylance, “Engineering Viscoelasticity,” *311 Mech. OfMaterials*, 2001.
- [31] X. Ma, H. E. Lynch, P. C. Scully, and M. S. Hutson, “Probing embryonic tissue mechanics with laser hole drilling,” *Phys. Biol.*, vol. 6, no. 3, pp. 1–12, 2009.
- [32] A. Y. Malkin and A. I. Isayev, *Rheology: Concepts, Methods, and Applications*. ChemTec Publishing, 2005.
- [33] G. Lenormand, E. Millet, B. Fabry, J. P. Butler, and J. J. Fredberg, “Linearity and time-scale invariance of the creep function in living cells,” *J. R. Soc. Interface*, vol. 1, no. 1, pp. 91–97, 2004.
- [34] N. Desprat, A. Richert, J. Simeon, and A. Asnacios, “Creep function of a single living cell,” *Biophys. J.*, vol. 88, no. 3, pp. 2224–2233, 2005.
- [35] K. Zuiderveld, “Contrast limited adaptive histogram equalization,” in *Graphics gems IV*, P. S. Heckbert, Ed. Boston, MA, 1994, pp. 474–485.
- [36] G. W. Brodland, D. Viens, and J. VELDHUIS, “A new cell-based FE model for the mechanics of embryonic epithelia,” *Comput. Methods Biomech. Biomed. Engin.*, vol. 10, no. 2, pp. 121–128, 2007.
- [37] H. H. Chen and G. W. Brodland, “Cell-level finite element studies of viscous cells in planar aggregates,” *J. Biomech. Eng.*, vol. 122, p. 394, 2000.
- [38] K. K. Chiou, L. Hufnagel, and B. I. Shraiman, “Mechanical Stress Inference for Two Dimensional Cell Arrays,” *PLoS Comput Biol*, vol. 8, no. 5, p. e1002512, May 2012.

- [39] S. Ishihara and K. Sugimura, “Bayesian inference of force dynamics during morphogenesis,” *J. Theor. Biol.*, 2012.
- [40] S. Ishihara, K. Sugimura, S. J. Cox, I. Bonnet, Y. Bellaïche, and F. Graner, “Comparative study of non-invasive force and stress inference methods in tissue,” *Eur. Phys. J. E*, vol. 36, no. 4, pp. 1–13, 2013.
- [41] G. W. Brodland, J. H. Veldhuis, S. Kim, M. Perrone, D. Mashburn, and M. S. Hutson, “CellFIT: A Cellular Force-Inference Toolkit Using Curvilinear Cell Boundaries,” *PLoS ONE*, vol. 9, no. 6, p. e99116, Jun. 2014.
- [42] J. A. Zallen and R. Zallen, “Cell-pattern disordering during convergent extension in *Drosophila*,” *J. Phys. Condens. Matter*, vol. 16, no. 44, p. S5073, Nov. 2004.
- [43] C. Bertet, L. Sulak, and T. Lecuit, “Myosin-dependent junction remodelling controls planar cell intercalation and axis elongation,” *Nature*, vol. 429, no. 6992, pp. 667–671, Jun. 2004.
- [44] R. Rousset, L. Almeida, and S. Noselli, “*Drosophila* morphogenesis: the Newtonian revolution,” *Curr. Biol.*, vol. 13, no. 12, pp. 494–495, 2003.
- [45] M. A. Wozniak and C. S. Chen, “Mechanotransduction in development: a growing role for contractility,” *Nat. Rev. Mol. Cell Biol.*, vol. 10, no. 1, pp. 34–43, Jan. 2009.
- [46] M. von Dassow and L. A. Davidson, “Variation and robustness of the mechanics of gastrulation: The role of tissue mechanical properties during morphogenesis,” *Birth Defects Res. Part C Embryo Today Rev.*, vol. 81, no. 4, pp. 253–269, Dec. 2007.
- [47] G. E. Box and N. R. Draper, *Empirical model-building and response surfaces*. John Wiley & Sons, 1987.
- [48] L. A. Davidson, *How do sea urchins invaginate? Using biomechanics to distinguish between mechanisms of primary invagination*, vol. 121. 1995.
- [49] R. Keller, L. A. Davidson, and D. R. Shook, “How we are shaped: The biomechanics of gastrulation,” *Differentiation*, vol. 71, no. 3, p. 171, 2003.
- [50] J. Alcaraz, L. Buscemi, M. Grabulosa, X. Trepas, B. Fabry, R. Farré, and D. Navajas, “Microrheology of Human Lung Epithelial Cells Measured by Atomic Force Microscopy,” *Biophys. J.*, vol. 84, no. 3, pp. 2071–2079, Mar. 2003.
- [51] P. Kunda, A. E. Pelling, T. Liu, and B. Baum, “Moesin Controls Cortical Rigidity, Cell Rounding, and Spindle Morphogenesis during Mitosis,” *Curr. Biol.*, vol. 18, no. 2, pp. 91–101, Jan. 2008.
- [52] C. Roubinet, B. Decelle, G. Chicanne, J. F. Dorn, B. Payrastre, F. Payre, and S.

- Carreno, “Molecular networks linked by Moesin drive remodeling of the cell cortex during mitosis,” *J. Cell Biol.*, vol. 195, no. 1, pp. 99–112, Oct. 2011.
- [53] A. R. Harris and G. T. Charras, “Experimental validation of atomic force microscopy-based cell elasticity measurements,” *Nanotechnology*, vol. 22, no. 34, p. 345102, Aug. 2011.
- [54] J.-L. Maître, H. Berthoumieux, S. F. G. Krens, G. Salbreux, F. Jülicher, E. Paluch, and C.-P. Heisenberg, “Adhesion Functions in Cell Sorting by Mechanically Coupling the Cortices of Adhering Cells,” *Science*, vol. 338, no. 6104, pp. 253–256, Oct. 2012.
- [55] A. R. Harris, L. Peter, J. Bellis, B. Baum, A. J. Kabla, and G. T. Charras, “Characterizing the mechanics of cultured cell monolayers,” *Proc. Natl. Acad. Sci.*, vol. 109, no. 41, pp. 16449–16454, Oct. 2012.
- [56] E. A. Pukhlyakova, Y. M. Efremov, D. V. Bagrov, N. N. Luchinskaya, D. O. Kiryukhin, L. V. Belousov, and K. V. Shaitan, “Atomic force microscopy as a tool to study *Xenopus laevis* embryo,” *J. Phys. Conf. Ser.*, vol. 345, no. 1, p. 012040, Feb. 2012.
- [57] B. R. Daniels, B. C. Masi, and D. Wirtz, “Probing Single-Cell Micromechanics In Vivo: The Microrheology of *C. elegans* Developing Embryos,” *Biophys. J.*, vol. 90, no. 12, pp. 4712–4719, Jun. 2006.
- [58] P. Kollmannsberger and B. Fabry, “BaHigh-force magnetic tweezers with force feedback for biological applications,” *Rev. Sci. Instrum.*, vol. 78, no. 11, p. 114301, Nov. 2007.
- [59] A. R. Bausch, F. Ziemann, A. A. Boulbitch, K. Jacobson, and E. Sackmann, “Local measurements of viscoelastic parameters of adherent cell surfaces by magnetic bead microrheometry,” *Biophys. J.*, vol. 75, no. 4, pp. 2038–2049, 1998.
- [60] F. J. Alenghat, B. Fabry, K. Y. Tsai, W. H. Goldmann, and D. E. Ingber, “Analysis of Cell Mechanics in Single Vinculin-Deficient Cells Using a Magnetic Tweezer,” *Biochem. Biophys. Res. Commun.*, vol. 277, no. 1, pp. 93–99, Oct. 2000.
- [61] N. Wang, J. P. Butler, and D. E. Ingber, “Mechanotransduction across the cell surface and through the cytoskeleton,” *Science*, vol. 260, no. 5111, pp. 1124–1127, May 1993.
- [62] B. Fabry, G. N. Maksym, J. P. Butler, M. Glogauer, D. Navajas, and J. J. Fredberg, “Scaling the microrheology of living cells,” *Phys. Rev. Lett.*, vol. 87, no. 14, pp. 148102–148102, 2001.
- [63] N. Desprat, W. Supatto, P.-A. Pouille, E. Beaurepaire, and E. Farge, “Tissue Deformation Modulates Twist Expression to Determine Anterior Midgut Differentiation in *Drosophila* Embryos,” *Dev. Cell*, vol. 15, no. 3, pp. 470–477,

Sep. 2008.

- [64] A. Kumar and G. V. Shivashankar, “Mechanical Force Alters Morphogenetic Movements and Segmental Gene Expression Patterns during *Drosophila* Embryogenesis,” *PLoS ONE*, vol. 7, no. 3, p. e33089, Mar. 2012.
- [65] H. Zhang and K.-K. Liu, “Optical tweezers for single cells,” *J. R. Soc. Interface*, vol. 5, no. 24, pp. 671–690, Jul. 2008.
- [66] J. E. Curtis, B. A. Koss, and D. G. Grier, “Dynamic holographic optical tweezers,” *Opt. Commun.*, vol. 207, no. 1–6, pp. 169–175, Jun. 2002.
- [67] K. Bambardekar, R. Clément, O. Blanc, C. Chardès, and P.-F. Lenne, “Direct laser manipulation reveals the mechanics of cell contacts in vivo,” *Proc. Natl. Acad. Sci.*, vol. 112, no. 5, pp. 1416–1421, Feb. 2015.
- [68] J. Wang, “Measurement of residual stress by the hole-drilling method: General stress-strain relationship and its solution,” *Exp. Mech.*, vol. 28, no. 4, pp. 355–358, 1988.
- [69] M. T. Abreu-Blanco, J. M. Verboon, R. Liu, J. J. Watts, and S. M. Parkhurst, “*Drosophila* embryos close epithelial wounds using a combination of cellular protrusions and an actomyosin purse string,” *J. Cell Sci.*, vol. 125, no. 24, pp. 5984–5997, 2012.
- [70] R. Farhadifar, J. C. Röper, B. Aigouy, S. Eaton, and F. Jülicher, “The influence of cell mechanics, cell-cell interactions, and proliferation on epithelial packing,” *Curr. Biol.*, vol. 17, no. 24, pp. 2095–2104, 2007.
- [71] D. Kiehart, Y. Tokutake, M.-S. Chang, M. Hutson, J. Wiemann, and X. G. Peralta, “Ultraviolet laser microbeam for dissection of *Drosophila* embryos,” in *Cell biology: a laboratory handbook*, J. E. Celis, Ed. San Diego, CA: Elsevier, 2006, pp. 87–103.
- [72] J. Colombelli, E. G. Reynaud, and E. H. . Stelzer, “Investigating relaxation processes in cells and developing organisms: from cell ablation to cytoskeleton nanosurgery,” *Methods Cell Biol.*, vol. 82, pp. 267–291, 2007.
- [73] D. P. Kiehart, C. G. Galbraith, K. A. Edwards, W. L. Rickoll, and R. A. Montague, “Multiple forces contribute to cell sheet morphogenesis for dorsal closure in *Drosophila*,” *J. Cell Biol.*, vol. 149, no. 2, pp. 471–490, 2000.
- [74] M. S. Hutson, Y. Tokutake, M.-S. Chang, J. W. Bloor, S. Venakides, D. P. Kiehart, and G. S. Edwards, “Forces for Morphogenesis Investigated with Laser Microsurgery and Quantitative Modeling,” *Science*, vol. 300, no. 5616, pp. 145–149, Apr. 2003.
- [75] M. S. Hutson and X. Ma, “Plasma and cavitation dynamics during pulsed laser

- microsurgery in vivo,” *Phys. Rev. Lett.*, vol. 99, p. 158104, 2007.
- [76] M. S. Hutson and G. S. Edwards, “Advances in the physical understanding of laser surgery at 6.45 microns,” *Phys Rev ST AB*, 2005.
- [77] X. Ma and M. S. Hutson, “Quantifying the Intercellular Forces during *Drosophila* Morphogenesis,” in *American Physical Society, APS March Meeting, March 13-17, 2006, abstract# H29. 003*, 2006.
- [78] X. Ma and M. S. Hutson, “Correlation between Recoil Velocity after Laser Ablation and Cell-edge Orientation,” in *American Physical Society, 74th Annual Meeting of the Southeastern Section, November 8-10, 2007, abstract# CB. 009*, 2007.
- [79] S. Saravanan, C. Meghana, and M. Narasimha, “Local, cell-nonautonomous feedback regulation of myosin dynamics patterns transitions in cell behavior: a role for tension and geometry?,” *Mol. Biol. Cell*, vol. 24, no. 15, pp. 2350–2361, Aug. 2013.
- [80] M. Mayer, M. Depken, J. S. Bois, F. Jülicher, and S. W. Grill, “Anisotropies in cortical tension reveal the physical basis of polarizing cortical flows,” *Nature*, vol. 467, no. 7315, pp. 617–621, Sep. 2010.
- [81] H. E. Lynch, S. M. Crews, B. Rosenthal, E. Kim, R. Gish, K. Echiverri, and M. S. Hutson, “Cellular mechanics of germ band retraction in *Drosophila*,” *Dev. Biol.*, vol. 384, no. 2, pp. 205–213, Dec. 2013.
- [82] Y. Toyama, X. G. Peralta, A. R. Wells, D. P. Kiehart, and G. S. Edwards, “Apoptotic Force and Tissue Dynamics During *Drosophila* Embryogenesis,” *Science*, vol. 321, no. 5896, p. 1683, 2008.
- [83] I. Bonnet, P. Marcq, F. Bosveld, L. Fetler, Y. Bellaïche, and F. Graner, “Mechanical state, material properties and continuous description of an epithelial tissue,” *J. R. Soc. Interface*, vol. 9, no. 75, pp. 2614–2623, Oct. 2012.
- [84] A. K. Jayasinghe, J. Rohner, and M. S. Hutson, “Holographic UV laser microsurgery,” *Biomed. Opt. Express*, vol. 2, no. 9, pp. 2590–2599, 2011.
- [85] M. Rauzi, P. Verant, T. Lecuit, and P. F. Lenne, “Nature and anisotropy of cortical forces orienting *Drosophila* tissue morphogenesis,” *Nat. Cell Biol.*, 2008.
- [86] M. Rauzi, P.-F. Lenne, and T. Lecuit, “Planar polarized actomyosin contractile flows control epithelial junction remodelling,” *Nature*, vol. 468, no. 7327, pp. 1110–1114, Dec. 2010.
- [87] M. Dembo, T. Oliver, A. Ishihara, and K. Jacobson, “Imaging the traction stresses exerted by locomoting cells with the elastic substratum method.,” *Biophys. J.*, vol. 70, no. 4, pp. 2008–2022, Apr. 1996.

- [88] K. A. Addae-Mensah and J. P. Wikswo, “Measurement techniques for cellular biomechanics in vitro,” *Exp. Biol. Med.*, vol. 233, no. 7, p. 792, 2008.
- [89] J. L. Tan, J. Tien, D. M. Pirone, D. S. Gray, K. Bhadriraju, and C. S. Chen, “Cells lying on a bed of microneedles: An approach to isolate mechanical force,” *Proc. Natl. Acad. Sci.*, vol. 100, no. 4, pp. 1484–1489, Feb. 2003.
- [90] P. C. Georges and P. A. Janmey, *Cell type-specific response to growth on soft materials*, vol. 98. Am Physiological Soc, 2005.
- [91] S. Kumar, I. Z. Maxwell, A. Heisterkamp, T. R. Polte, T. P. Lele, M. Salanga, E. Mazur, and D. E. Ingber, “Viscoelastic retraction of single living stress fibers and its impact on cell shape, cytoskeletal organization, and extracellular matrix mechanics,” *Biophys. J.*, vol. 90, no. 10, pp. 3762–3773, 2006.
- [92] V. Maruthamuthu, B. Sabass, U. S. Schwarz, and M. L. Gardel, “Cell-ECM traction force modulates endogenous tension at cell–cell contacts,” *Proc. Natl. Acad. Sci.*, vol. 108, no. 12, pp. 4708–4713, Mar. 2011.
- [93] X. Trepast, M. R. Wasserman, T. E. Angelini, E. Millet, D. A. Weitz, J. P. Butler, and J. J. Fredberg, “Physical forces during collective cell migration,” *Nat. Phys.*, vol. 5, no. 6, pp. 426–430, Jun. 2009.
- [94] M. Fernández-Suárez and A. Y. Ting, “Fluorescent probes for super-resolution imaging in living cells,” *Nat. Rev. Mol. Cell Biol.*, vol. 9, no. 12, pp. 929–943, Dec. 2008.
- [95] Y. Tseng, T. P. Kole, and D. Wirtz, “Micromechanical mapping of live cells by multiple-particle-tracking microrheology,” *Biophys. J.*, vol. 83, no. 6, pp. 3162–3176, Dec. 2002.
- [96] D. T. Chen, E. R. Weeks, J. C. Crocker, M. F. Islam, R. Verma, J. Gruber, A. J. Levine, T. C. Lubensky, and A. G. Yodh, “Rheological Microscopy: Local Mechanical Properties from Microrheology,” *Phys. Rev. Lett.*, vol. 90, no. 10, p. 108301, Mar. 2003.
- [97] O. Campàs, T. Mammoto, S. Hasso, R. A. Sperling, D. O’Connell, A. G. Bischof, R. Maas, D. A. Weitz, L. Mahadevan, and D. E. Ingber, “Quantifying cell-generated mechanical forces within living embryonic tissues,” *Nat. Methods*, vol. 11, no. 2, pp. 183–189, Feb. 2014.
- [98] A.-L. Cost, P. Ringer, A. Chrostek-Grashoff, and C. Grashoff, “How to Measure Molecular Forces in Cells: A Guide to Evaluating Genetically-Encoded FRET-Based Tension Sensors,” *Cell. Mol. Bioeng.*, pp. 1–10, Dec. 2014.
- [99] K. Suffoletto, N. Ye, F. Meng, D. Verma, and S. Z. Hua, “Intracellular forces during guided cell growth on micropatterns using FRET measurement,” *J. Biomech.*, vol. 48, no. 4, pp. 627–635, Feb. 2015.

- [100] D. Cai, S.-C. Chen, M. Prasad, L. He, X. Wang, V. Choesmel-Cadamuro, J. K. Sawyer, G. Danuser, and D. J. Montell, “Mechanical feedback through E-cadherin promotes direction sensing during collective cell migration,” *Cell*, vol. 157, no. 5, pp. 1146–1159, 2014.
- [101] S. P. Arnoczky, M. Lavagnino, J. H. Whallon, and A. Hoonjan, “In situ cell nucleus deformation in tendons under tensile load; a morphological analysis using confocal laser microscopy,” *J. Orthop. Res. Off. Publ. Orthop. Res. Soc.*, vol. 20, no. 1, pp. 29–35, Jan. 2002.
- [102] R. P. Jean, D. S. Gray, A. A. Spector, and C. S. Chen, “Characterization of the Nuclear Deformation Caused by Changes in Endothelial Cell Shape,” *J. Biomech. Eng.*, vol. 126, no. 5, pp. 552–558, Nov. 2004.
- [103] K. N. Dahl, A. J. S. Ribeiro, and J. Lammerding, “Nuclear shape, mechanics, and mechanotransduction,” *Circ. Res.*, vol. 102, no. 11, pp. 1307–1318, Jun. 2008.
- [104] P. Friedl, K. Wolf, and J. Lammerding, “Nuclear mechanics during cell migration,” *Curr. Opin. Cell Biol.*, vol. 23, no. 1, pp. 55–64, Feb. 2011.
- [105] E. A. Jares-Erijman and T. M. Jovin, “FRET imaging,” *Nat. Biotechnol.*, vol. 21, no. 11, pp. 1387–1395, Nov. 2003.
- [106] C. Grashoff, B. D. Hoffman, M. D. Brenner, R. Zhou, M. Parsons, M. T. Yang, M. A. McLean, S. G. Sligar, C. S. Chen, T. Ha, and M. A. Schwartz, “Measuring mechanical tension across vinculin reveals regulation of focal adhesion dynamics,” *Nature*, vol. 466, no. 7303, pp. 263–266, Jul. 2010.
- [107] J. B. Pawley, *Handbook of biological confocal microscopy*, 3rd ed. New York: Springer, 2006.
- [108] M. Held, M. H. . Schmitz, B. Fischer, T. Walter, B. Neumann, M. H. Olma, M. Peter, J. Ellenberg, and D. W. Gerlich, “CellCognition: time-resolved phenotype annotation in high-throughput live cell imaging,” *Nat. Methods*, vol. 7, no. 9, pp. 747–754, 2010.
- [109] B. Aigouy, R. Farhadifar, D. B. Staple, A. Sagner, J.-C. Röper, F. Jülicher, and S. Eaton, “Cell Flow Reorients the Axis of Planar Polarity in the Wing Epithelium of *Drosophila*,” *Cell*, vol. 142, no. 5, pp. 773–786, Sep. 2010.
- [110] M. R. Lamprecht, D. M. Sabatini, and A. E. Carpenter, “CellProfilerTM: free, versatile software for automated biological image analysis,” *Biotechniques*, vol. 42, no. 1, p. 71, 2007.
- [111] K. Li, E. D. Miller, M. Chen, T. Kanade, L. E. Weiss, and P. G. Campbell, “Cell population tracking and lineage construction with spatiotemporal context,” *Med. Image Anal.*, vol. 12, no. 5, pp. 546–566, 2008.

- [112] I. F. Sbalzarini and P. Koumoutsakos, “Feature point tracking and trajectory analysis for video imaging in cell biology,” *J. Struct. Biol.*, vol. 151, no. 2, pp. 182–195, 2005.
- [113] A. Pinidiyaarachchi and C. Wählby, “Seeded watersheds for combined segmentation and tracking of cells,” *Image Anal. Process. 2005*, pp. 336–343, 2005.
- [114] D. P. McCullough, P. R. Gudla, B. S. Harris, J. A. Collins, K. J. Meaburn, M.-A. Nakaya, T. P. Yamaguchi, T. Misteli, and S. J. Lockett, “Segmentation of Whole Cells and Cell Nuclei From 3-D Optical Microscope Images Using Dynamic Programming,” *IEEE Trans. Med. Imaging*, vol. 27, no. 5, pp. 723–734, May 2008.
- [115] A. Cardona, S. Saalfeld, S. Preibisch, B. Schmid, A. Cheng, J. Pulokas, P. Tomancak, and V. Hartenstein, “An integrated micro-and macroarchitectural analysis of the Drosophila brain by computer-assisted serial section electron microscopy,” *PLoS Biol.*, vol. 8, no. 10, p. e1000502, 2010.
- [116] H. K. Hahn and H.-O. Peitgen, “IWT-interactive watershed transform: a hierarchical method for efficient interactive and automated segmentation of multidimensional gray-scale images,” in *Medical Imaging 2003*, 2003, pp. 643–653.
- [117] G. Lin, M. K. Chawla, K. Olson, C. A. Barnes, J. F. Guzowski, C. Bjornsson, W. Shain, and B. Roysam, “A multi-model approach to simultaneous segmentation and classification of heterogeneous populations of cell nuclei in 3D confocal microscope images,” *Cytom. A*, vol. 71, no. 9, pp. 724–736, 2007.
- [118] P. A. Yushkevich, J. Piven, H. C. Hazlett, R. G. Smith, S. Ho, J. C. Gee, and G. Gerig, “User-guided 3D active contour segmentation of anatomical structures: significantly improved efficiency and reliability,” *Neuroimage*, vol. 31, no. 3, pp. 1116–1128, 2006.
- [119] K. Khairy and P. J. Keller, “Reconstructing embryonic development,” *Genesis*, vol. 49, pp. 1–26, 2011.
- [120] K. Miura, “Tracking movement in cell biology,” in *Microscopy Techniques*, vol. 95, Springer Berlin / Heidelberg, 2005, pp. 1304–1307.
- [121] H. Oda and S. Tsukita, “Real-time imaging of cell-cell adherens junctions reveals that Drosophila mesoderm invagination begins with two phases of apical constriction of cells,” *J. Cell Sci.*, vol. 114, no. 3, p. 493, 2001.
- [122] S. Beucher and F. Meyer, “The Morphological Approach to Segmentation: The Watershed Transformation,” in *Mathematical Morphology in Image Processing*, E. Dougherty, Ed. New York: CRC Press, 1993, pp. 433–481.
- [123] L. Vincent and P. Soille, “Watersheds in digital spaces: an efficient algorithm based

- on immersion simulations,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 6, pp. 583–598, 1991.
- [124] L. P. Coelho, “Mahotas: Open source software for scriptable computer vision,” *arXiv:1211.4907*, Nov. 2012.
- [125] D. J. V. David, A. Tishkina, and T. J. C. Harris, “The PAR complex regulates pulsed actomyosin contractions during amnioserosa apical constriction in *Drosophila*,” *Development*, vol. 137, pp. 1645–1655, Apr. 2010.
- [126] R. Fernández-González, A. Muñoz-Barrutia, M. H. Barcellos-Hoff, and C. Ortiz-de-Solorzano, “Quantitative in vivo microscopy: the return from the ‘omics,’” *Curr. Opin. Biotechnol.*, vol. 17, no. 5, pp. 501–510, Oct. 2006.
- [127] 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,” *Nat. Biotechnol.*, vol. 28, pp. 348–353, Mar. 2010.
- [128] E. N. Mortensen and W. A. Barrett, “Interactive segmentation with intelligent scissors,” *Graph. Models Image Process.*, vol. 60, no. 5, pp. 349–384, 1998.
- [129] N. Ben-Zadok, T. Riklin-Raviv, and N. Kiryati, “Interactive level set segmentation for image-guided therapy,” in *IEEE International Symposium on Biomedical Imaging: From Nano to Macro.*, Boston, MA, 2009, pp. 1079–1082.
- [130] Y. Kang, K. Engelke, and W. A. Kalender, “Interactive 3D editing tools for image segmentation,” *Med. Image Anal.*, vol. 8, no. 1, pp. 35–46, 2004.
- [131] H. K. Hahn and H.-O. Peitgen, “The skull stripping problem in MRI solved by a single 3D watershed transform,” in *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2000*, 2000, pp. 134–143.
- [132] J.-M. Kuhnigk, H. Hahn, M. Hindennach, V. Dicken, S. Krass, and H.-O. Peitgen, “Lung lobe segmentation by anatomy-guided 3D watershed transform,” in *Medical Imaging 2003*, 2003, pp. 1482–1490.
- [133] J. Ollion, J. Cochenec, F. Loll, C. Escude, and T. Boudier, “TANGO: a generic tool for high-throughput 3D image analysis for studying nuclear organization,” *Bioinformatics*, vol. 29, no. 14, pp. 1840–1841, Jul. 2013.
- [134] J. Bresenham, “An incremental algorithm for digital plotting,” presented at the Proc. ACM Natl. Conf., 1963.
- [135] J. E. Bresenham, “Algorithm for computer control of a digital plotter,” *IBM Syst. J.*, vol. 4, no. 1, pp. 25–30, 1965.
- [136] V. Boyer and J. J. Bourdin, “A faster algorithm for 3D discrete lines,” in *Proceedings of the European Association for Computer Graphics Conference*,

1998.

- [137] D. Cohen, “Voxel traversal along a 3D line,” *Graph. Gems IV*, vol. 4, p. 366, 1994.
- [138] L. . Dimitrov and M. Sramek, “Using 3D-Bresenham for resampling structured grids,” in *2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings, 2004*, pp. 926–930.
- [139] B. Zalik, G. Clapworthy, and C. Oblonsek, “An Efficient Code-Based Voxel-Traversing Algorithm,” *Comput. Graph. Forum*, vol. 16, no. 2, pp. 119–128, Jun. 1997.
- [140] A. Kaufman, “Efficient algorithms for scan-converting 3D polygons,” *Comput. Graph.*, vol. 12, no. 2, pp. 213–219, 1988.
- [141] D. Prodanov and K. Verstreken, *Automated Segmentation and Morphometry of Cell and Tissue Structures. Selected Algorithms in ImageJ*. INTECH Open Access Publisher, 2012.
- [142] X. Papademetris, M. P. Jackowski, N. Rajeevan, M. DiStasio, H. Okuda, R. T. Constable, and L. H. Staib, “BioImage Suite: An integrated medical image analysis suite: An update,” *Insight J.*, vol. 2006, p. 209, 2006.
- [143] C. R. Johnson, “Biomedical Visual Computing: Case Studies and Challenges,” *Comput. Sci. Eng.*, vol. 14, no. 1, pp. 12–21, Jan. 2012.
- [144] A. Fedorov, R. Beichel, J. Kalpathy-Cramer, J. Finet, J.-C. Fillion-Robin, S. Pujol, C. Bauer, D. Jennings, F. Fennessy, M. Sonka, and others, “3D Slicer as an image computing platform for the Quantitative Imaging Network,” *Magn. Reson. Imaging*, vol. 30, no. 9, pp. 1323–1341, 2012.
- [145] K. Mkrtychyan, D. Singh, M. Liu, V. Reddy, A. Roy-Chowdhury, and M. Gopi, “Efficient cell segmentation and tracking of developing plant meristem,” in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, 2011, pp. 2165–2168.
- [146] R. Fernandez, P. Das, V. Mirabet, E. Moscardi, J. Traas, J.-L. Verdeil, G. Malandain, and C. Godin, “Imaging plant growth in 4D: robust tissue reconstruction and lineaging at cell resolution,” *Nat. Methods*, vol. 7, no. 7, pp. 547–553, 2010.
- [147] Z. Khan, Y.-C. Wang, E. F. Wieschaus, and M. Kaschube, “Quantitative 4D analyses of epithelial folding during *Drosophila* gastrulation,” *Development*, vol. 141, no. 14, pp. 2895–2900, 2014.
- [148] E. Hodneland, T. Kögel, D. M. Frei, H.-H. Gerdes, and A. Lundervold, “CellSegm—a MATLAB toolbox for high-throughput 3D cell segmentation,” *Source Code Biol. Med.*, vol. 8, no. 1, pp. 1–24, 2013.

- [149] A. Körbes, G. B. Vitor, R. de Alencar Lotufo, and J. V. Ferreira, “Advances on watershed processing on GPU architecture,” in *Mathematical Morphology and Its Applications to Image and Signal Processing*, Springer, 2011, pp. 260–271.
- [150] V. Vezhnevets and V. Konouchine, “GrowCut: Interactive multi-label ND image segmentation by cellular automata,” in *proc. of Graphicon*, 2005, pp. 150–156.
- [151] X. Morin, “A protein trap strategy to detect GFP-tagged proteins expressed from their endogenous loci in *Drosophila*,” *Proc. Natl. Acad. Sci.*, vol. 98, no. 26, pp. 15050–15055, Dec. 2001.
- [152] M. D. Abràmoff, P. J. Magalhães, and S. J. Ram, “Image processing with ImageJ,” *Biophotonics Int.*, vol. 11, no. 7, pp. 36–43, 2004.
- [153] K. Michielsen, H. D. Raedt, and J. T. M. Hosson, “Aspects of Mathematical Morphology,” *Order*, vol. 501, p. 2083, 2002.
- [154] E. Somersalo and J. Kaipio, “Statistical and computational inverse problems,” *Appl. Math. Sci.*, vol. 160, 2004.
- [155] A. C. Martin, M. Gelbart, R. Fernandez-Gonzalez, M. Kaschube, and E. F. Wieschaus, “Integration of contractile forces during tissue invagination,” *J. Cell Biol.*, vol. 188, no. 5, pp. 735–749, 2010.
- [156] R. R. Lew, “How does a hypha grow? The biophysics of pressurized growth in fungi,” *Nat. Rev. Microbiol.*, vol. 9, no. 7, pp. 509–518, 2011.
- [157] J. Stefan, “Experiments on the apparent adhesion.,” *Proc. Imp. Acad. Sci. Wiss. Vienna Math Sci. Cl.*, no. 69, pp. 713–735, 1874.
- [158] Z. Tadmor and C. G. Gogos, *Principles of polymer processing*. John Wiley & Sons, 2006.
- [159] Y. Hiramoto, “Mechanical properties of the protoplasm of the sea urchin egg: I. Unfertilized egg,” *Exp. Cell Res.*, vol. 56, no. 2–3, pp. 201–208, Aug. 1969.
- [160] K. Weigmann, R. Klapper, T. Strasser, C. Rickert, G. Technau, H. Jäckle, W. Janning, and C. Klämbt, “FlyMove – a new way to look at development of *Drosophila*,” *Trends Genet.*, vol. 19, no. 6, pp. 310–311, Jan. 2003.