Volumetric Reconstruction of Tissue Structure from Two-Dimensional Microscopy Images

by

Francisco Cruz

Submitted to the
Department of Electrical Engineering and Computer Science

May 30, 2006

In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

Cell morphology of tissue is naturally three-dimensional. Most current methods for tissue analysis use two dimensional histological images of the tissue samples, restricting the analysis to 2D. Existing approaches do not provide essential three-dimensional information such as cell volume, shape and structural orientation of cells within the tissue. This thesis investigates a method to extract three dimensional data using two-dimensional microscopy. We demonstrate that three dimensional cell structure can be acquired using two dimensional fluorescence microscopy and two-photon microscopy and explore the application of the analysis to studies of cardiac tissue.

Thesis Co-Supervisor: Polina Golland
Title: Assistant Professor of Electrical Engineering and Computer Science

Thesis Co-Supervisor: Richard T. Lee
Title: Associate Professor of Medicine, Harvard Medical School

Acknowledgements

I would like to thank my advisors Dr. Richard Lee and Professor Polina Golland in helping me develop and finish my thesis work and thesis. I would especially like to thank Hayden Huang for providing me the opportunity to work with him and starting me off in this project. Hayden provided me with guidance and support through out my lab experience and mentored me during different projects. I would also like to thank the entire Lee Lab for helping me with all the biological information and advice. I additionally would like to thank Ron Rivest for being my academic advisor and consulting with me on my coursework for the 5 years I have been at MIT.

Contents

Biographical Note

I was awarded the degree of Bachelors of Science in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology in June 2004. I enrolled in the Electrical Engineering and Computer Science Department at the Massachusetts Institute of Technology in September of 2005. During my time at MIT, I have been co-supervised by Polina Golland, Assistant Professor of Electrical Engineering and Computer Science and Richard T. Lee, Associate Professor of Medicine, Harvard Medical School. All of the research presented in this master's thesis was performed in Richard Lee's lab at Brigham and Women's Hospital.

Chapter 1 Introduction

This thesis investigates methods and procedures to construct and study three dimensional cell structures of large tissue samples such as an entire heart. Currently there is no standard method for analysis of three dimensional tissue data at subcellular resolution. Standard microscopes or image acquisition instruments in biological research create two dimensional images. Although images acquired are two dimensional, methods can be created to properly use such planar images to extract volumetric representation of cells and visualize them in 3D.

Methods to acquire images for 3D analysis include MRI and PET scans. Unfortunately, these methods produce macro level information and are not suitable for imaging the structure at the cellular level. However, we use the existing methods for registration and segmentation of the MRI images as a basis for developing analysis techniques applicable to microscopy images. Our approach considers separate layers of the 3D object acquired in a series of planar images and reconstructs the original volumetric shape from such a collection of slices.

The experiments in this thesis employ a fluorescent microscope and a two-photon microscope to acquire 2D images and to reconstruct cellular structure of tissues. The specific application that motivated this work is the analysis of heart tissue. The resolution of the microscopy images allowed us to extract cells and visualize them in 3D. To accomplish this, we have developed alignment, segmentation and interpolation algorithms that use a systematic approach to the 3D reconstruction of the cells. In addition to visualizing the cells, the analysis yielded important measurements, such as volume, surface area and orientation of each cell.

1.2 Thesis Organization

The following chapters provide a detailed description of the methods and protocols used to develop the reconstruction of the 3D tissue. Chapter 2 provides an overview of the background on biology, histology and instrumentation used to acquire images. Chapter 3 explains the image processing techniques we developed for cell reconstruction, including image registration, image segmentation, image intensity normalization, reconstruction of 3D data such as calculating volume and surface area. It also discusses the visualization of the extracted cell structure. Chapter 4 discusses Matlab implementation of the proposed methods. Chapter 5 demonstrates the application of the techniques in real biological studies. Chapter 6 discusses possible extensions to other tissue types and presents future plans and further development of the techniques for automation of 3D cell segmentation.

Chapter 2 Background

In this chapter, we review the background on tissue staining and image acquisition techniques. The algorithms presented in the next chapter rely on the acquisition methods described here to generate high contrast images of the cellular structure.

2.1 Fluorescence Labels

Fluorescence labeling is a common technique in biology used to identify certain aspects of tissue and cells. Today there is widespread availability of high-quality fluorescence probes; many labeling protocols have been developed, making fluorescence labeling easy to apply.

Fluorescence is a process in which molecules give off light. Fluorescent molecules or fluorophores are excited by light and consequently emit it during relaxation at a different wavelength [1]. There is a wide variety of fluorophores that are used for specific labeling. The molecules are often antibodies with conjugated chemical fluorophores or proteins that are intrinsically fluorescent such as green fluorescent proteins (GFP). Other fluorophores directly stain cells and tissue. Dyes such as Texas-red maleimide can be used to stain the extracellular matrix of tissue by nucleophilic attack of sulfhydryl groups in the extracellular matrix. For example, Texas Red emits red fluorescence, and DAPI stains are used to stain nuclei and emit blue fluorescence. Figure 1 illustrates tissue staining and how it affects the appearance of the cell.

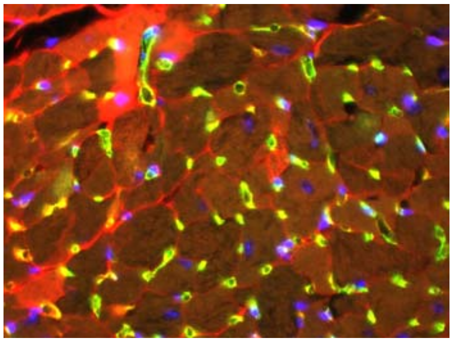


Figure 1. Example of a tissue sample labeled with multiple stains. The extracellular matrix is stained with maleimide-Texas Red (red), the capillaries are stained with isolectin-fluorescein (green) and nuclei are stained with DAPI (blue).

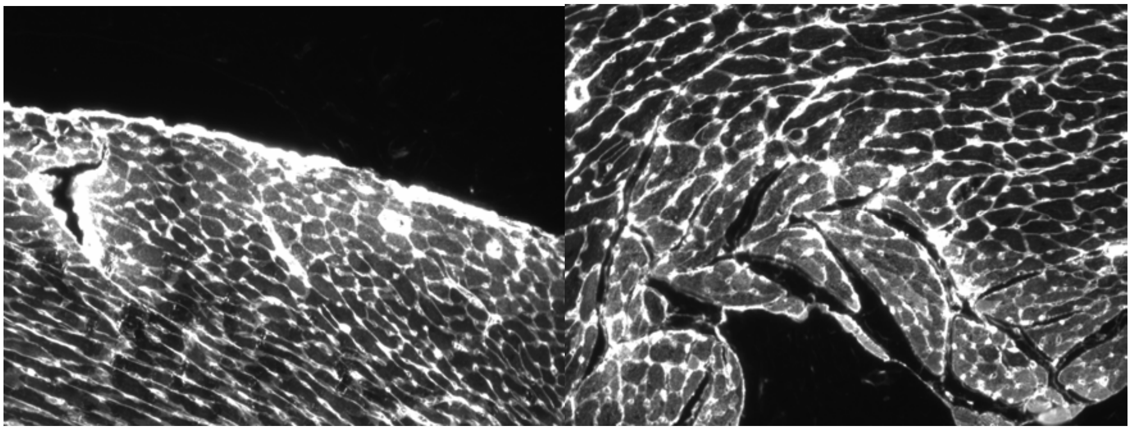


Figure 2. Images stained with maleimide, which targets extra-cellular matrix. The image to the left is of the heart outer wall; the image to the right is of the edge of a ventricle.

2.2 Staining of the Tissue

Preparing the tissue samples correctly for imaging is a crucial step. In order to be able to see cells or other structures within the heart, such as nuclei, the heart must be stained with fluorescence dyes. These dyes, such as maleimide-Texas Red, are used to outline cells or directly stain nuclei. Figure 2 shows a cross-section of a heart stained with Texas red maleimide. A red filter is used to acquire the images. The dye shows up as white in grayscale images.

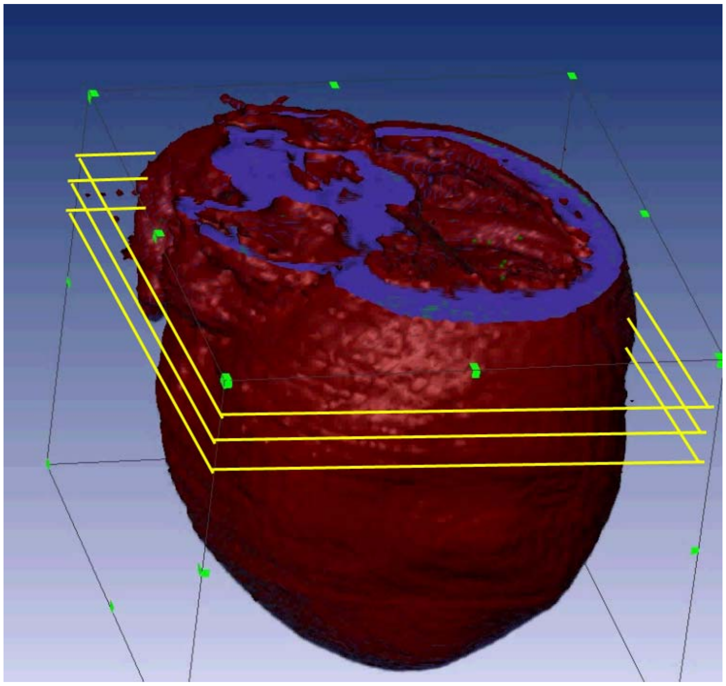


Figure 3. Serial slices of the heart are cut along the direction of the yellow outline.

To obtain these cross-sectional images, mouse hearts were labeled using a Texas Red-maleimide (Molecular Probes, Inc) tail-vein injection, followed by removing the heart, fixing with 4% paraformaldehyde and embedding the heart in paraffin.

Once the staining is complete, the tissue is cut into slices as shown in Figure 3. The z-axis runs from the base to the apex of the heart, and is perpendicular to the serial slices. The slice thickness is 5μm. The length of a heart cell ranges from 70 to130 μm. Thus using 5-10 μm slices are sufficient to obtain accurate volumetric data. The slices are then fixed onto slides for imaging.

2.3 Imaging Equipment: Fluorescence Microscopy and Multi-photon Technology

Fluorescence microscopes acquire images in which constructs are labeled with fluorophores. There are three elements to a microscope filter that allow for visualization of the desired fluorophores: the excitation filter, the beam splitter and the emission filter [1]. The excitation filter is essentially a bandpass filter that regulates the wavelengths that reach the specific fluorophore. The emission filter is a highpass filter that allows wavelengths above a certain threshold to be emitted. The beam splitter regulates the light exciting the fluorophores and the light emitted back to the viewer or the camera [1]. The resulting two-dimensional images a do not provide any volumetric information. Instrumentation has been a large factor in deriving methods to create 3D models. Three-dimensional microscopy approaches are enabled by confocal microscopy and two photon microscopy. Confocal microscopy uses fluorescence illumination and then filters out-of-plane signals in order to create a scan at a certain depth [2]. Two-photon microscopy typically uses a red-to-infrared laser to generate an excitation volume in order to acquire an image. Because of the long wavelength of infrared lasers, greater penetration depth is achievable with two-photon microscopy than with confocal microscopy [3]. A two-photon microscope is more thoroughly explained below.

Two-photon microscopy enables volumetric scanning. Red-to-infrared (> 700 nm wavelengths) generates an excitation volume confined to the area of maximal focus after

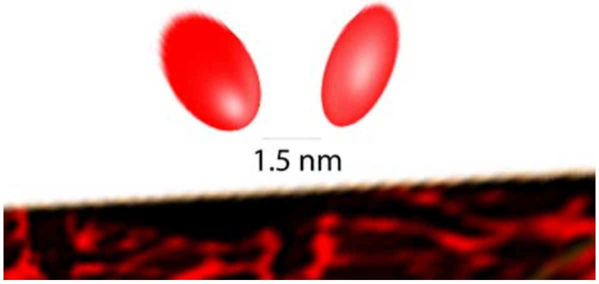


Figure 4 Two photons hitting a sample. The photons must impact the specimen with in 1.5 nm of each other within 1.5 ns.

the objective [4]. Excitation relies on two photons exciting the same molecule nearly simultaneously. Therefore, only the focal region has sufficient photon density to generate a fluorescence signal. Figure 5 demonstrates the steps for two-photon scanning. Two-photon microscopy has two main advantages. First, restricting the excitation to a small volume result in less photobleaching and in reduction in out-of-plane noise. Second, the use of long wavelengths which have greater penetration depth into tissues results in scanning of optically favorable tissues of up to several hundred microns. Disadvantages of two-photon microscopy include heat damage of the specimen, expense of the laser and the microscope maintenance, and in some cases, difficulty in locating the feature of interest due to the scanning nature of the majority of two-photon microscopes [4]. Two-photon microscopy is well-suited for three-dimensional cell and tissue scanning, in part because of its penetration depth [5].

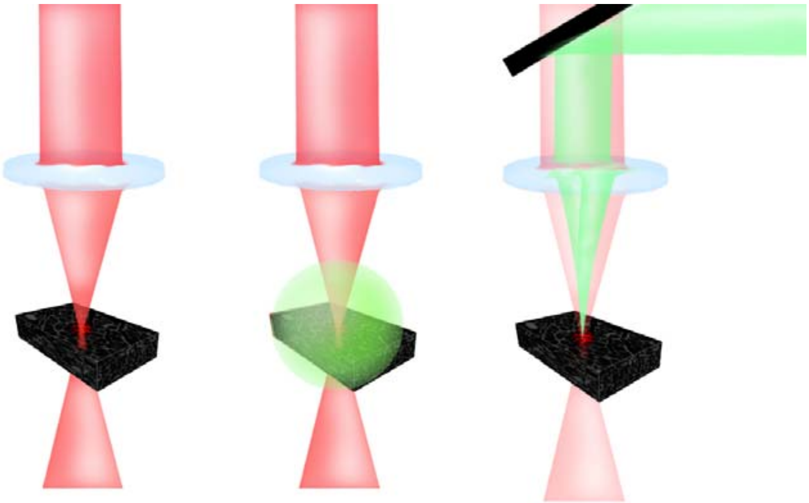


Figure 5. This diagram shows an overview of the two-photon scanning. Within the confined excitation volume two photons target the specimen, which then excite the field and project light back into the computer through the mirror.

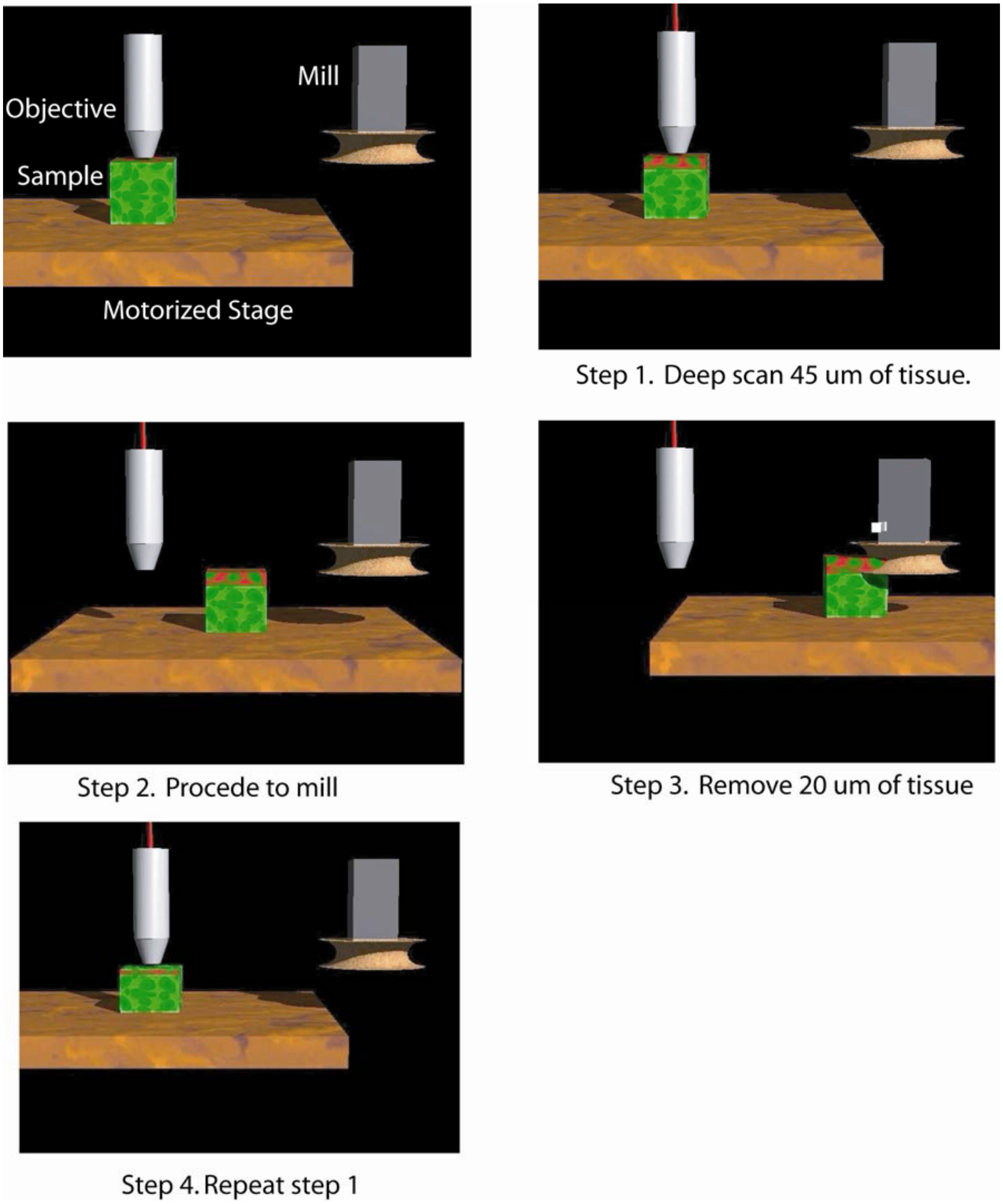
Algorithms presented in this thesis were developed for images acquired by both fluorescence microscopy and two-photon microscopy. Two-photon microscopy allows for faster image acquisition and thus more tissue can be imaged in a manageable time frame.

2.4 Image Acquisition

Two methods are used to image the heart in our experiments. The florescence microscope was used to acquire images manually. In addition, a motorized stage using a multi-photon microscope was used in some experiments.

Images are taken by a person using a fluorescence microscope along with a 10x objective which has a resolution of 1.5536 pixel/µm. Multiple overlapping images are obtained per slide. Figure 2 shows an example image acquired using this method. We can easily see the outlines of cells and also the perimeter of the heart as shown on the left and a ventricle as shown on the right half.

The second method to acquire images after the staining procedure uses a multi-photon microscope instead of a normal fluorescence microscope. The multi-photon microscope has 3D scanning capabilities since it can focus on different layers in the z axis up to 45 µm without having to cut or create serial slices. The distance between the layers is 1.5 µm. Thirty layers are combined to create a volumetric scan. In order to scan more than 45 µm in depth, the microscope stage has a mill that, in real time, removes 20 µm of tissue. The system then continues to image the remaining sample. Figure 6 shows a graphical flow chart of the scanning and cutting procedure.



Step 1. Deep scan 45 um of tissue.

Step 2. Procede to mill

Step 3. Remove 20 um of tissue

Step 4. Repeat step 1

Figure 6. Imaging with motorized microscope. The top left most image shows the motorized stage, mill, sample and objective. Step 1 is to scan 45 µm of the tissue at 1.5 µm between slices. The sample is then moved to the mill where 20 µm of tissue is removed. The sample is then returned to the objective where another 45 µm are scanned.

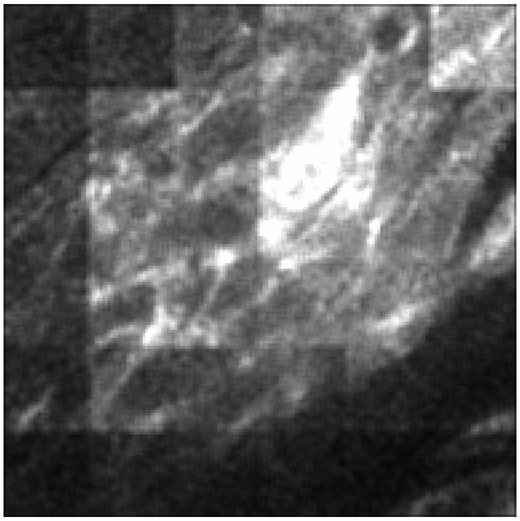


Figure 7. Sample of an output image from multi-photon microscope. The grid-like structure is due to the power distribution of the multi-photon microscope.

An example image from one slice is shown in Figure 7. The grid pattern occurs due to the power distribution of the microscope laser. The multi-photon microscope uses a 6x6 lens array to split the laser so that 36 smaller images can be taken at the same time. This substantially reduces the imaging time. The majority of the laser power is concentrated in the middle of the imaged area. We can see the brighter field where the power is greater.

Chapter 3 Image Analysis

Once the acquisition of the data is complete, the entire image set can be analyzed with the goal of extracting, quantifying and visualizing the 3D structure of the cells. The sections in this chapter explain the analysis steps in detail.

3.1 Registration

Image registration is the process of transforming acquired images into a unified coordinate system through alignment that minimizes the error of the fit. Image registration methods can be grouped into two different classes: area based methods and feature based methods. Both types of methods start with one reference image used as the reference coordinate system which all other target images will be transformed to. Area-based image registration looks at the overall structure of the images and uses correlation methods, Fourier properties and other structural analysis to complete the image alignments. Feature based registration uses image features to perform its mapping using for example lines, curves, points or boundaries.

3.1.1 Slice Mosaic Image – Area Based Registration

The first step in the 3D reconstruction pipeline is to create a large image for each layer that was acquired by either method described in the previous chapter. We developed an image registration program to mosaic individual slices. The registration method has two input parameters: overlap percentage, namely the area fraction by which the images overlap, and a position matrix. If we have images heart1.tiff, heart2.tiff through

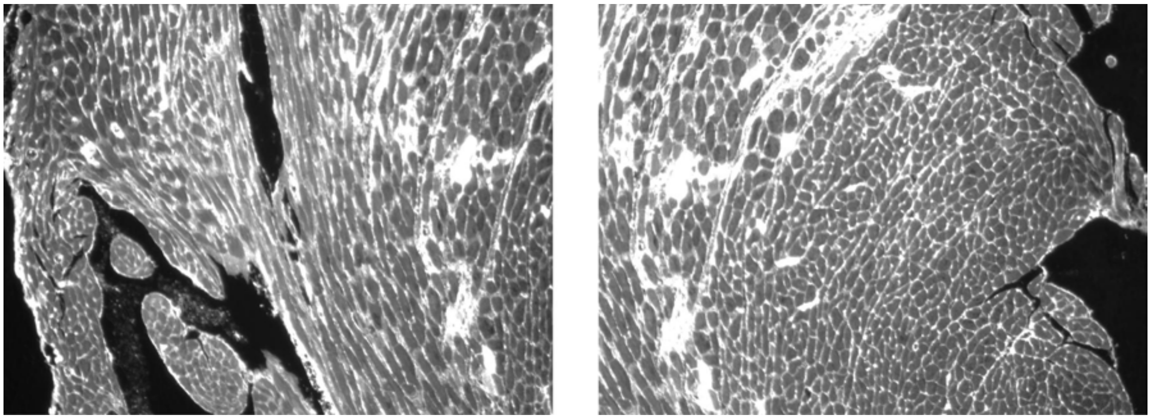


Figure 8. Cross-sectional images of a mouse heart. Shown above are two images which need to be registered and consequently put into a larger mosaic.

heart9.tiff, and they are arranged in a square array, the user would input a matrix of how they will be laid out as shown below:

$$\begin{array}{ccc}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{array} \qquad \text{or} \qquad \begin{array}{ccc}1 & 4 & 7\\ 2 & 5 & 8\\ 3 & 6 & 9\end{array}$$

This preconditioner is used by the algorithm to restrict the registration to images that actually overlap. For example, image 1 and 6 are never compared directly. The registration algorithm employs the normalized cross-correlation between two images to estimate (x,y) offset of the best fit between a pair of images. These offsets are used to place the images appropriately relative to one another. Let I_1 be the reference image and I_2 be the image that needs to be aligned. Normalized cross-correlation [6] is defined as follows:

$$\gamma(u,v) = \frac{\sum_{x,y}[I_1(x,y)-\overline{I_1(u,v)}][I_2(x-u,y-v)-\overline{I_2}]}{\{\sum[I_1(x,y)-\overline{I_1(u,v)}]^2\sum[I_2(x-u,y-v)-\overline{I_2}]^2\}^{1/2}}\,.$$

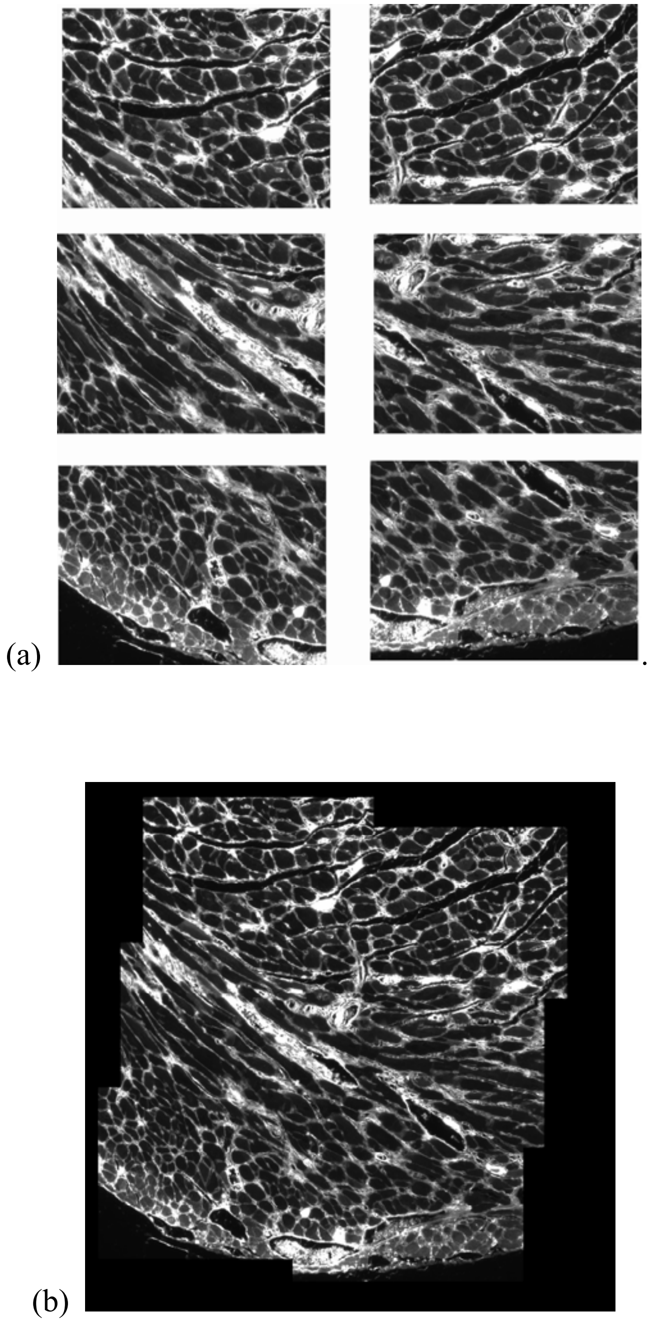


Figure 9. Original images which were imaged at 20x magnification (a) combined into a single mosaic (b).

The user input overlap percentage is used to reduce the area of the images that will be compared leading to faster runtime and lower memory requirements.

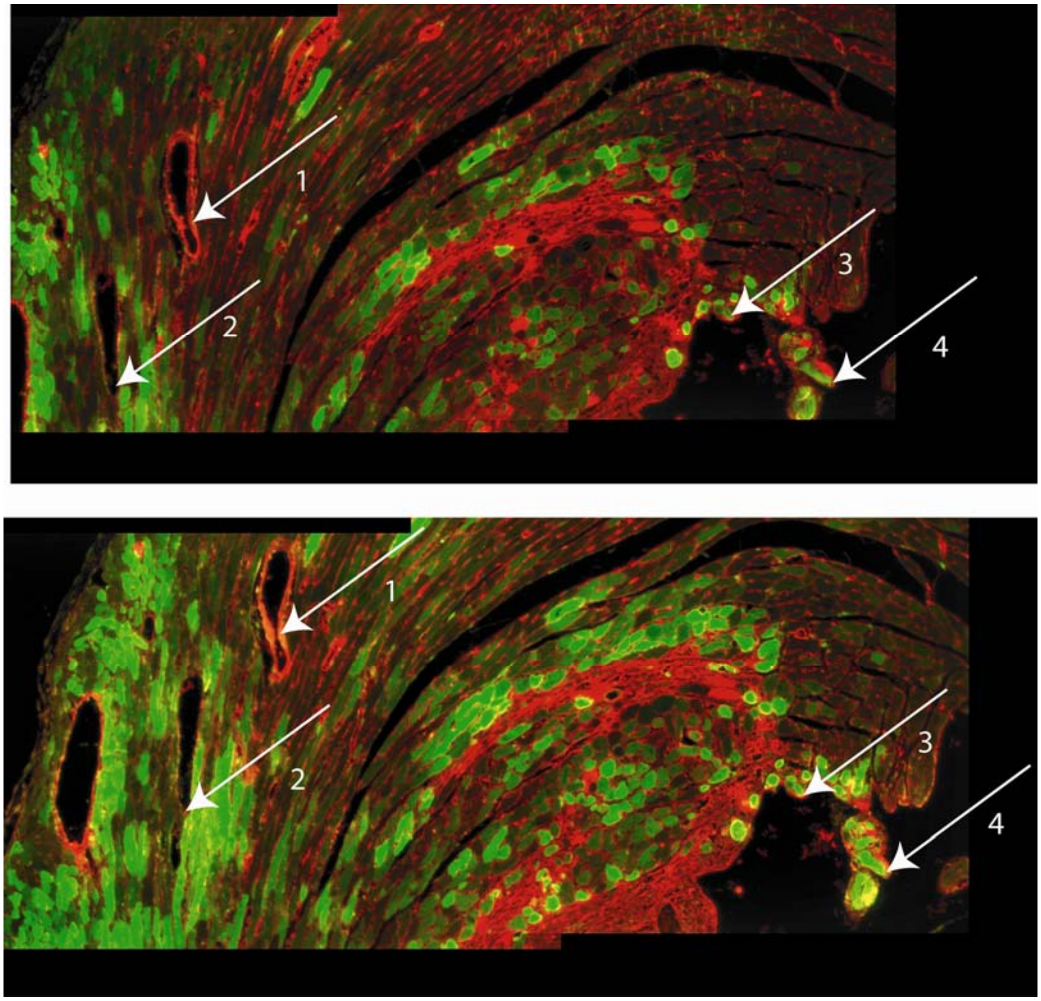


Figure 10. Pick points where vessels and features occur in two consecutive images.

The mosaic image in Figure 9b was created by manual scanning using a fluorescence microscope and then registering the 6 original images, shown in Figure 6a, together. The initial overlap specified by the user was 30%. The size of the images is 1320x1024.

3.1.2 Slice Alignment in 3D – Feature Based Registration

Slide preparation results in rotational misalignment of images in the z-direction. To correct for this misalignment, we register the slice images by optimizing the (x,y) offset

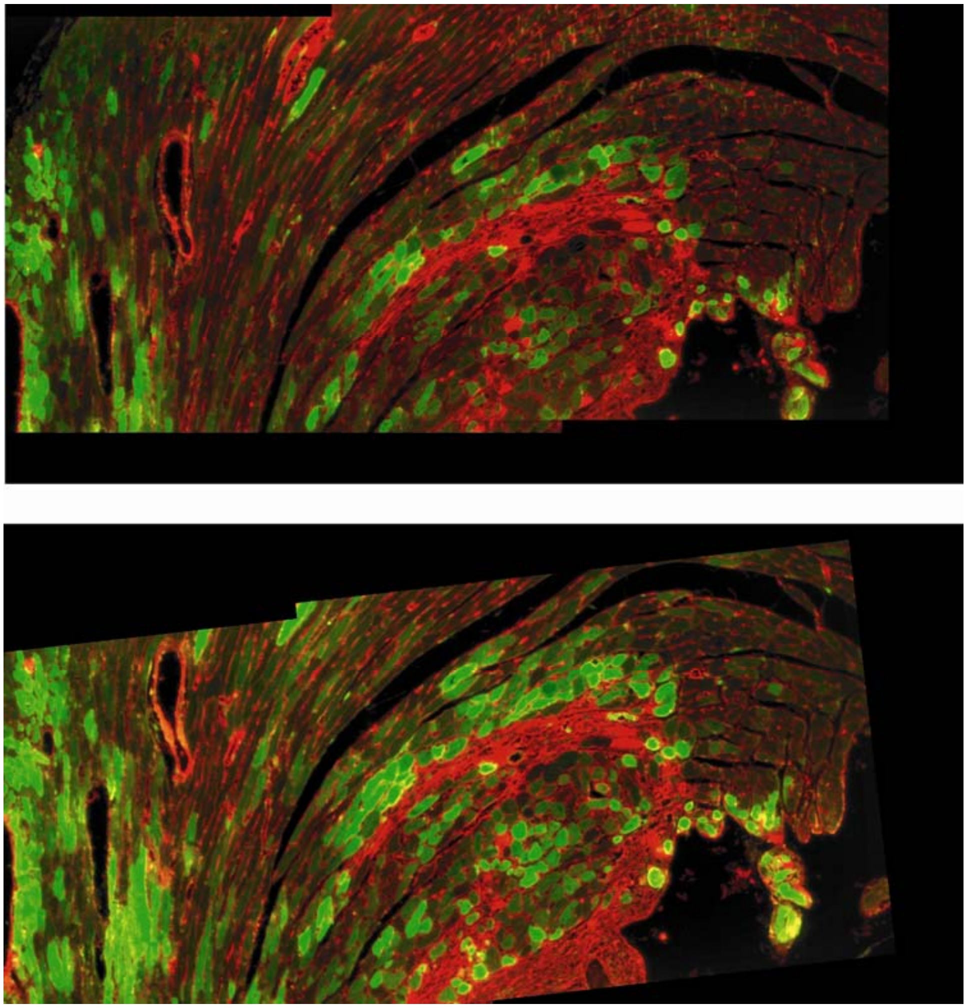


Figure 11. Results of applying linear conformal registration with rotational and translational shifts calculated from the selected features in consecutive slices.

and the angle of rotation. Unlike the problem of creating a large mosaic of smaller images of the same area, the alignment in the depth direction requires registration of images that are not of the same region. Consequently, the area based registration algorithms are not suitable for this application. To best solve this problem, we choose to use a feature based registration algorithm. In this step the user manually picks features on images that occur in consecutive slices.

After selecting these points, we calculate the rigid transformation that aligns the selected features to obtain the rotational and translational shifts [7]. Figure 10 illustrates the feature selection step. Figure 11 shows slices aligned after performing a rigid transformation.

3.1.3 Image Registration for Multi-Photon Microscopy

When using the two photon microscope, the registration problem is simplified tremendously compared to the registration needed for manual imaging. The layout of the microscope and its motorized stage provides specific offsets for the motion of the microscope when taking adjacent images. Registration is used only to correct for the error of the instrumentation.

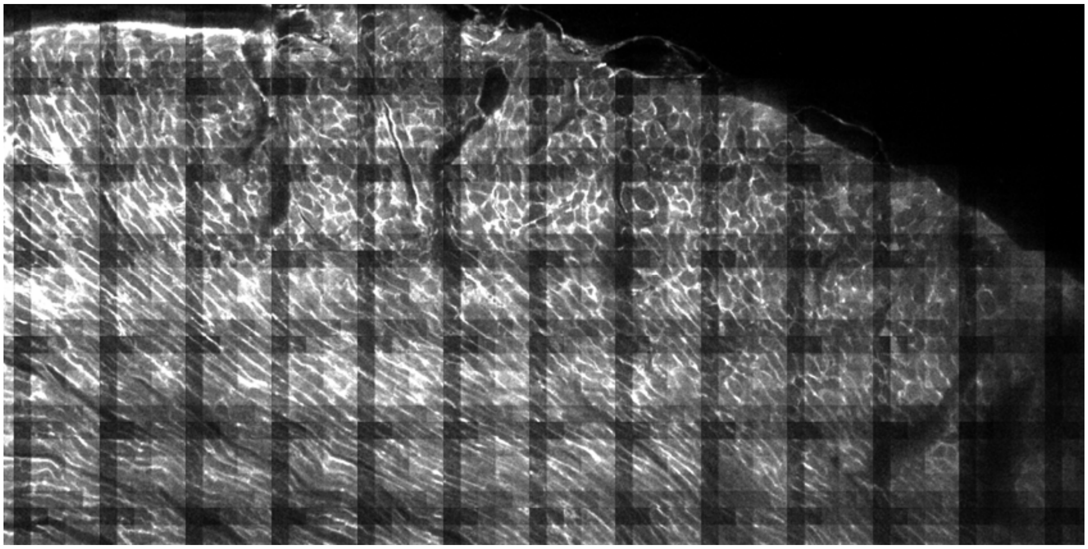


Figure 12. (X ,Y) reconstruction of two-photon microscope captured data

Figure 12 shows a registered image of the images taken from the two-photon microscope. Since the two-photon microscope has a pre-determined method of how the images will be taken and ordered determining offsets of adjacent images is minimal

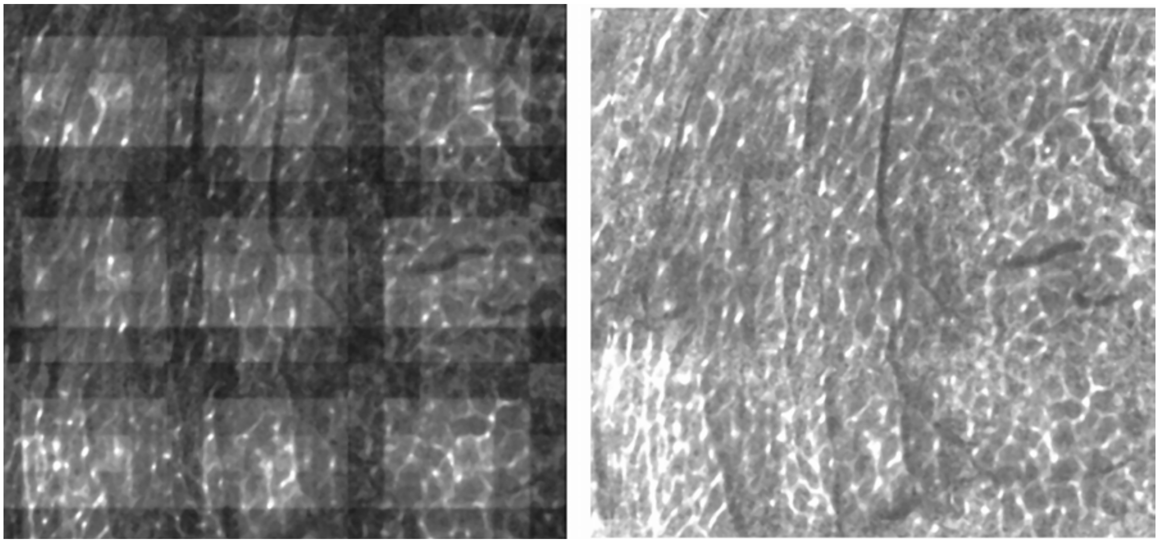


Figure 13. Example image before and after intensity normalization.

3.2 Intensity Normalization

Intensity normalization is needed when the multi-photon microscope is used, due to the varying power distribution of the microscope laser. The grid-like structure in the resulting images is the direct consequence of the-nonuniform power distribution. The laser of the microscope is split into 36 different sections. To smooth the image relative to all the others, an averaged intensity matrix is created which has size 6x6. The values of the matrix range from 0 to 255.

Each sub-block of the image is multiplied by ratio of the average pixel intensity of the entire image and the average intensity for that sub block.

To get a global representation of the average intensity matrix, we average a large sample of images from the entire data set to normalize the data set. The result of normalization is illustrated in Figure 13.

3.3 Cell Segmentation

Image segmentation is the partitioning of an image into multiple regions of interest. The easiest method to segment images is through manual classification, when one edits the image to delineate areas of interest. In our case the areas of interest correspond to cross-sections of cells. Image segmentation is an extensively explored problem in computer vision. Many algorithms and methods have been developed in the past years, which include watershed segmentation and edge detection. We have adapted the methods to cell image segmentation as described in the remainder of this section.

The combination of watershed and edge detection algorithms produced good results in segmenting heart cells from cross-sectional slices. The principle behind watershed detection is to treat the image as a set of basins or wells. Black regions are considered the bottom of the wells called basins and the white the bright areas are treated as tops of the wells called watershed lines [8]. Figure 14 shows a one dimensional example of how an image is treated in watershed segmentation.

One problem with watershed methods is that applying them to noisy images results in over segmentation. To prevent over segmentation, one can impose minima to predetermine where the basins are located [8]. This information provides more control of the segmentation for images that might have large variability in their intensity values.

As we can see in Figure 15a, the staining has already provided a division between each cell. But the division is not consistent. Some borders are dimmer than others and using simple threshholding methods will fail, since some of the borders have the same intensity as some of the cells. Applying watershed and edge detection algorithms directly does not yield desirable results either, as demonstrated in Figure 15.

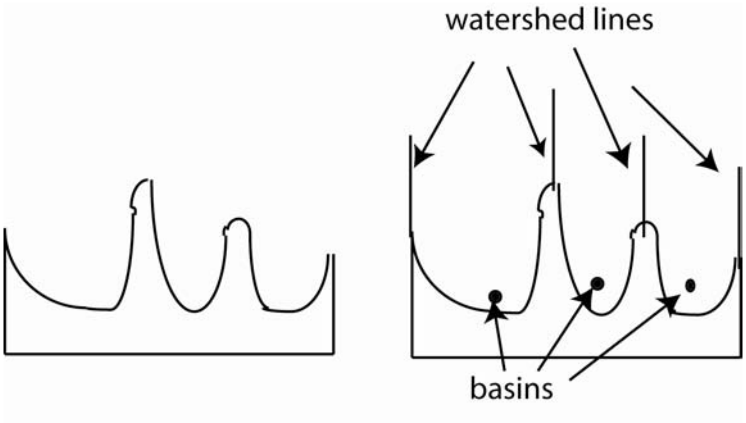


Figure 14. One dimensional example of watershed representation. The watershed lines divide regions based on minima and maxima, minima are basins while maxima are watershed lines.

Our modified algorithm starts by treating a grayscale image as a collection of wells as described above. First we calculate the minima of regions, and then impose those minima as markers so that watershed can create a pre-segmentation of the image. The minima are used as region markers to prevent over segmentation.

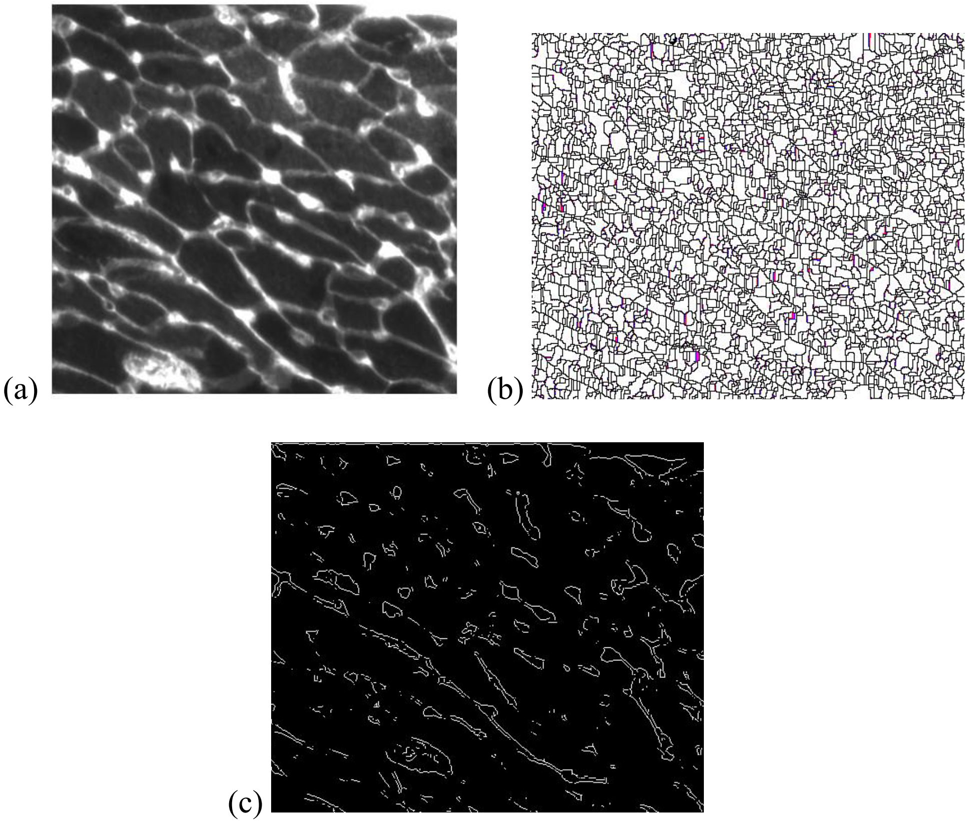


Figure 15. Example of cells which will be segmented (a). Direct implementation of typical watershed (b). Direct implementation of an edge detection algorithm using Prewitt methods with a threshold of .1 (c).

A watershed algorithm is then performed on the image using the minima found as basins. Figure 16 (a) demonstrates the minima imposed on the image before performing a watershed segmentation. The contrast of the image is adjusted to facilitate the watershed propagation. The resulting image is shown in Figure 16 (b). The watershed lines are superimposed on the adjusted contrast image so that edge detection is forced to use the watershed lines. Using a gradient based edge detection algorithm we can obtain the image in figure 17 (a).

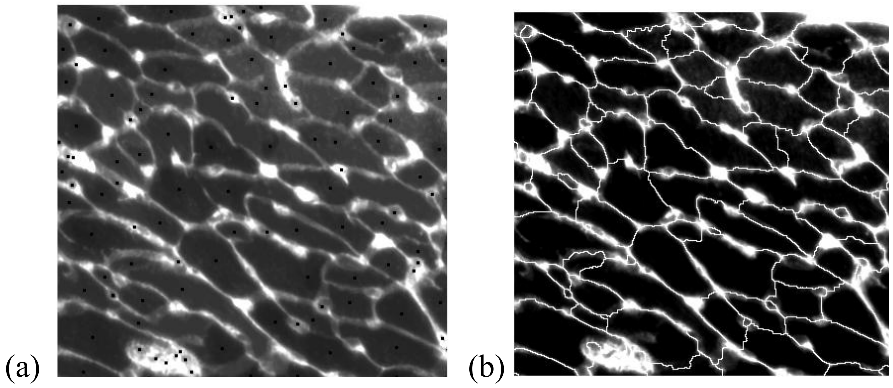


Figure 16. Imposed minima to control over segmentation of watershed algorithms (a) and watershed lines and increased contrast of the image (b).

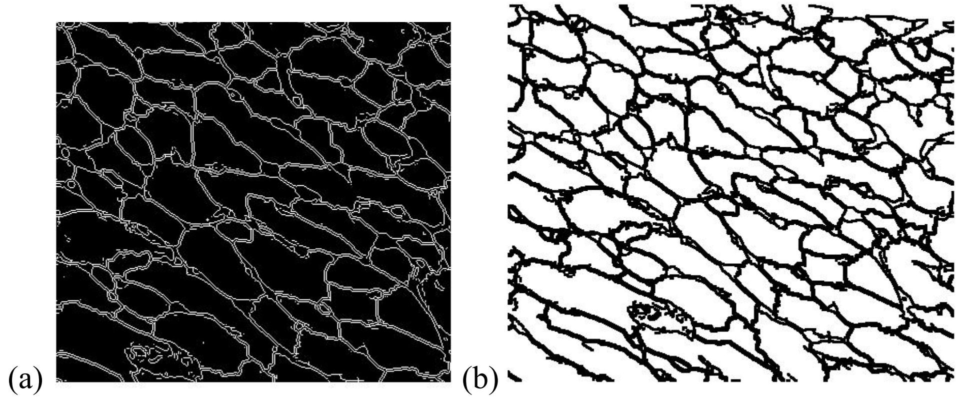


Figure 17. Edge detection results (a). Segmented image (b). White objects correspond to cells. We can now easily extract cell shapes along with other parameters such as centroids and areas.

Vertical and horizontal dilation of two pixels is performed to obtain solid boundary lines. Dilation forces smaller objects such as capillaries in the images to be considered as borders. This is advantageous if one is interested in finding the number of cells in a given cross-section, causing the cells to be the counted and smaller objects to be ignored in the count. The final edge detection results are illustrated in Figure 17 (b).

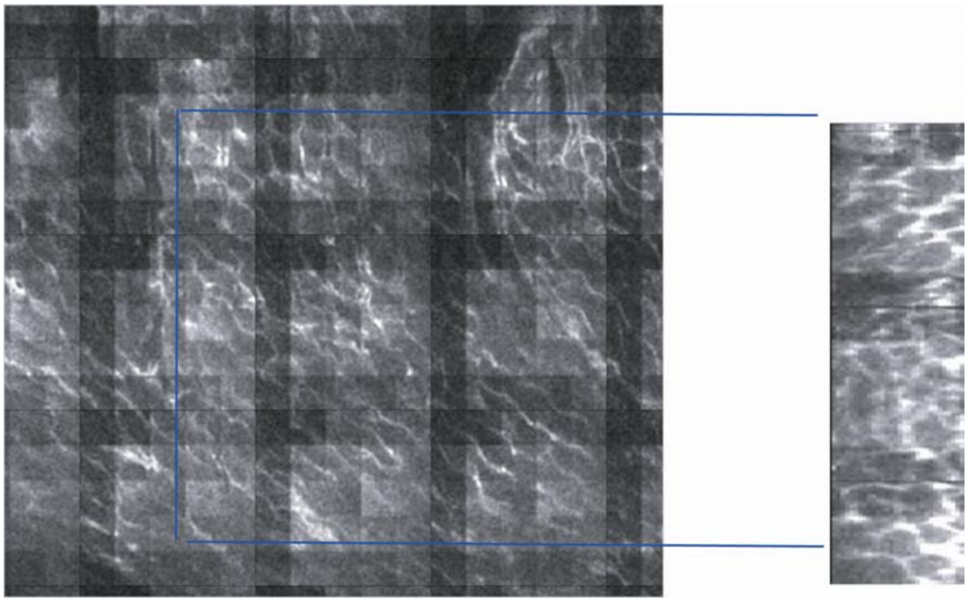


Figure 18. Interpolated cross-section of the yz plane.

3.4 Multi-plane interpolation

Interpolation enables display of different cross-sections of the imaged volume. Because we only cut the tissue sample perpendicular to one direction, imaging cross-sections of different planes is impossible. Using interpolation allows us not only to see different cross-sections but also to ensure the validity of registration results along the z-direction. Multi-photon images allow for higher resolution of 1.5μm between slices and for minimal distortion in the z registration. These two features enable us to interpolate cross-sectional slices in other planes much more accurately and with higher resolution. We can use k-nearest neighbor interpolation algorithms to find cross-sections with the image stacks acquired. Interpolation of any plane can be accomplished as illustrated in Figure 18.

3.5 Whole Cell Reconstruction

After registration and segmentation steps are completed, it is possible to reconstruct the volumetric structure of the cells. Using our larger registered images we now pick a region and track cells from layer to layer. The tracking is done by finding the cell continuation from a previous layer. As cells are tracked, the segmentation program is used to obtain the shape, the centroid and the area measurements. The centroid is used to later identify where the cell is located and to also find the cell orientation. Once we have three or more centroids from different layers of the same cell, a linear regression is used to fit a vector through those points. Characterizing cell orientation throughout the heart tells us how the cells are oriented in the tissue. Using the cross-sectional areas of the tracked cell,

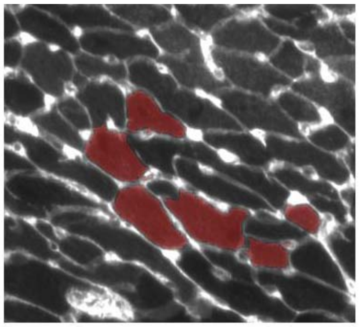


Figure 19. Cell selection after segmentation. Red colored cells are saved in a different file. These files are then imported into software such as Amira for visualization.

we apply a disk integration to obtain the volume which equals $\sum_{i=1}^{n} Area_i \times \Delta z$ where Δz is the thickness or distance between layers (in the two photon case it is 1.5 μm). Cell surface area is calculated by measuring the perimeters of each cross-section and multiplying by the distances between the slices. During cell tracking, the cell in question is colored in, after the segmentation all cells have a pointer to them and are treated as objects. Cells are selected and colored in as in Figure 19; a new image is then created with black background and the cell as all white. This image is used later for visualization of the cell structure in 3D.

3.6 Visualization

To visualize a cell, we use the cell segmentation images. These images have the shape and the location of the cell. They also contain a cross-sections stack of one cell. Using software such as Amira or Matlab we can create a visual 3D model. Figure 20 shows us the interpolation in the z-direction used in Amira to construct the cell model. The regular

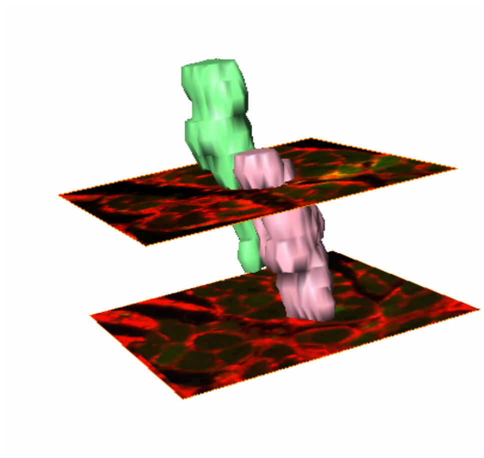


Figure 20. Final visual model. This model was created using manual imaging. The rendering was created through a program called Amira.

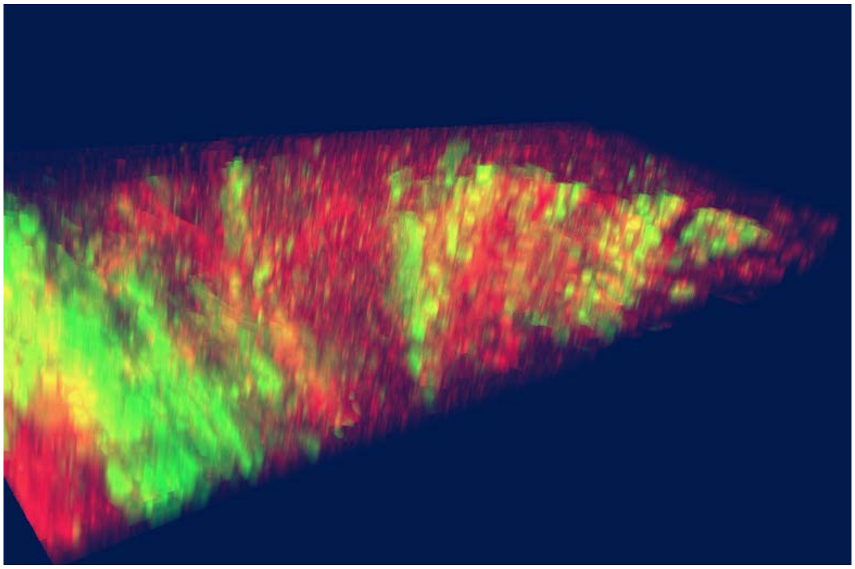


Figure 21. Images imported into Amira allow us to see where cells marked with fluorescence are located relative to the section of the heart.

fluorescent microscope was used to obtain the images used for the models in 20. As one can see, because of the errors in image collection, images from layer to layer are shifted slightly and the cells do not look as smooth as we would expect from the biology of the tissue.

Amira is a three-dimensional visualization program created by Mercury Computer Microsystems. Amira is a dynamic software package that is able to take a variety of different large data sets and construct 3D models such as the one in Figure 21. Another similar available program includes 3D Slicer which is free through www.slicer.org.

Chapter 4 System Implementation

The algorithms described in the previous chapter were implemented in Matlab. Although any of these algorithms and image processing techniques can be implemented in any programming language, Matlab's rich function libraries allow fast prototyping and experiments. The Matlab programs are attached as Appendix A.

4.1 Registration

Appendix A.1 provides the algorithms used for image registration. In particular the most important function for region based registration is *normxcorr2*. This function performs the normalized cross correlation evaluation of two matrices. Using this function and its results, offsets can be calculated and image registration can be implemented as a simple loop.

For the feature based registration of images misaligned in the z-direction, the *cpt2form* function creates a rotation and shift matrix from the input points in two images. The matrix represents the rigid transformation to be applied to an image.

4.2 Segmentation

The two key functions for segmentation are the *watershed* and the *edge* functions. Pre-processing the images as mentioned in section 3.3 is crucial to implementing the watershed algorithm effectively. Appendix A.2 provides the functions that calculate and impose minima on the image of interest before the watershed algorithm is applied.

Matlab also has some options for edge detection such as Prewit or Canny methods to edge detection. Whichever method one chooses should be tested on the type of images used in the experiments. Dilation of images can be performed using the *dilate* function. The user can specify how large the dilation in the x and y directions should be and the shape of the dilation kernel

To obtain cross-sectional area and perimeter measurements, Matlab's *regionprops* function classifies different objects by their connectivity. It then stores properties of those objects such as area, perimeter length and centroids. For more detail, we refer the readers to the Appendix.

Chapter 5 Experimental Validation

This chapter discusses results of the applications of cellular level 3D reconstruction.

5.1 Hypertrophy in Cardiomyocytes

In a recent study it was shown that overexpression of heparin-binding epidermal growth factor (HB-EGF) causes cardiac hypertrophy on cells and its immediate neighbors in vivo [9]. Hypertrophy is an increase in the size of a tissue or organ due to the enlargement of existing cells. Cardiac hypertrophy is a physiological response that enables the heart to adapt to an initial stress. However these increases in sizes can lead to deterioration in cardiac function and an increase in cardiac arrhythmias. In this study we aimed to answer the following question: If a cell produces signals that lead to hypertrophy, how are the neighboring cells affected?

Two dimensional cross-sectional area data showed that cells that produce signals which cause hypertrophy do indeed have larger cross-sectional area and so did their neighboring cells compared to remote cells. Although cross-sectional area increased, it is also important to consider the overall volume of the cells since it is possible that cells became wider but also shorter. Applying 3D visual and quantitative reconstruction proved indeed that the overall volumes of infected cells and their neighbors increased significantly. It also determined that the average volume of cardiac myocytes is about 20,000 cubic micrometers. Figure 22 illustrates results of the study, previously published in [9]. It illustrates that cardiomyocytes almost doubled in volumetric size when infected by HB-EGF, and neighboring cells increased on average by 50 percent.

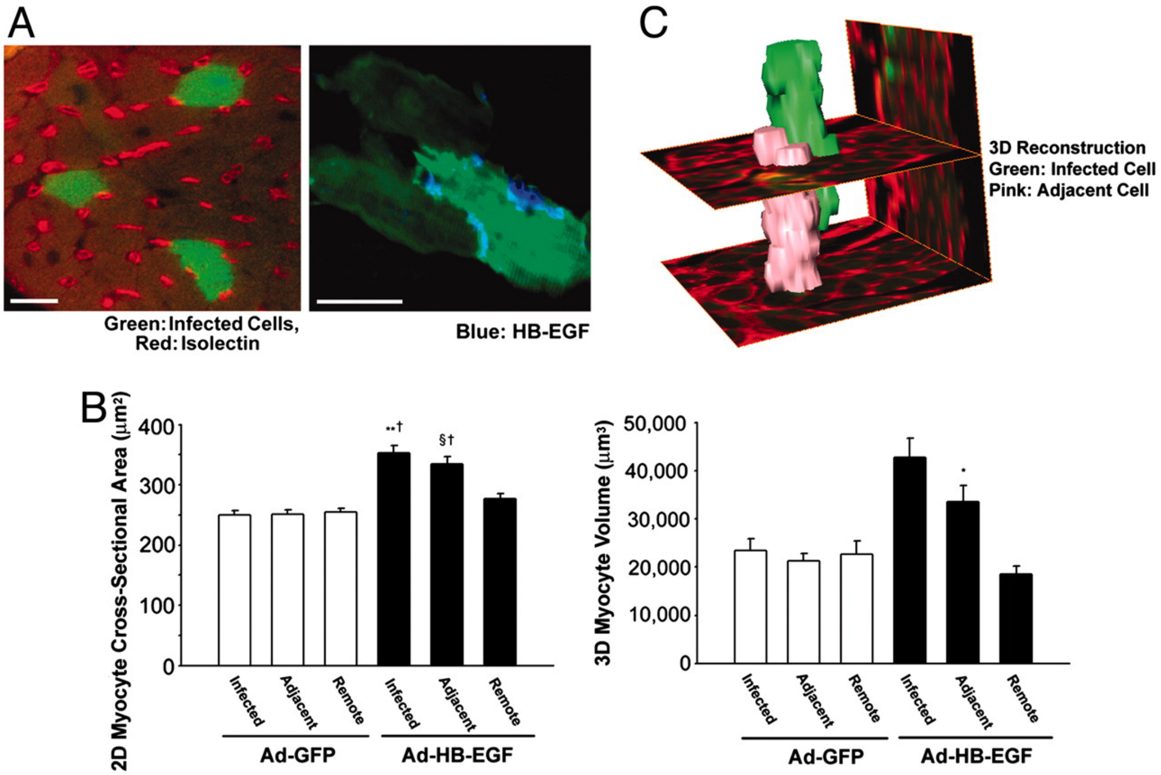


Figure 22. Effects of gene transfer of HB-EGF on cardiomyocyte hypertrophy in vivo. (A) Two-dimensional cross-sectional area of myocytes measured from left ventricle cross sections. Virus infected cells expressing GFP were analyzed, as well as cells adjacent to the GFP expressing cells, and cells remote to the GFP expressing cell in each of the Ad-HB-EGF and Ad-GFP injected hearts. Ad-HB-EGF caused a significant increase in size (41% increase compared to Ad-GFP-infected cells, **P<0.01) and non-infected adjacent cells to Ad-HB-EGF (33% increase of Ad-GFP-adjacent cells, §P<0.01) compared to remote cells. In Ad-GFP injected hearts (n infected=102 cells, n adjacent=92 cells, n remote=97 cells from 5 mice), and in Ad-HB-EGF injected hearts (n infected=119 cells, n adjacent=97 cells, n remote=109 cells from 7 mice). (B) Three-dimensional myocyte volume measured from left ventricle cross-sections registered in the z direction. Virus infected cells expressing GFP were analyzed, as well as cells adjacent to the GFP expressing cells, and cells remote to the GFP expressing cell in each of the Ad-HB-EGF and Ad-GFP injected hearts. Ad-HBEGF induced hypertrophy in the infected cell and its adjacent cell only. Ad-HB-EGF cell volumes were significantly greater than Ad-GFP infected cells (**P<0.01) and remote cells in the Ad-HB-EGF injected heart (†P<0.01). Additionally, cells adjacent to the Ad-HB-EGF cells were significantly larger than remote cells in the Ad-HB-EGF (*P<0.05) injected heart. In Ad-GFP injected hearts n infected=12 cells, n adjacent=10 cells, n remote=9 cells) and in Ad-HB-EGF injected hearts (n infected=19 cells, n adjacent=11 cells, n remote=13 cells). (C) Sample image of extracted myocytes in three dimensions. [9]

5.2 Gene Mutated Mice Exhibit Abnormal Cardiac Cell Growth.

In this study, we examined the structure of cells in mutant mice. Preliminary data using 3D cellular reconstruction have been gathered. In this study (Seidman Lab, Harvard Medical School) mice with specific gene mutations were compared to wild-type mice. The study showed that the cardiomyocyte volume in the mutated mice on average was three times greater than in wild-type mice. The cells also exhibited abnormal shape compared to myocytes from wild-type mice. Figure 23 shows in blue a cell which was reconstructed using our methods. We note that it is not a normal log-shaped myocyte. Figure 24 shows the difference between calculated volumes of cells in mutated and non mutated mice.

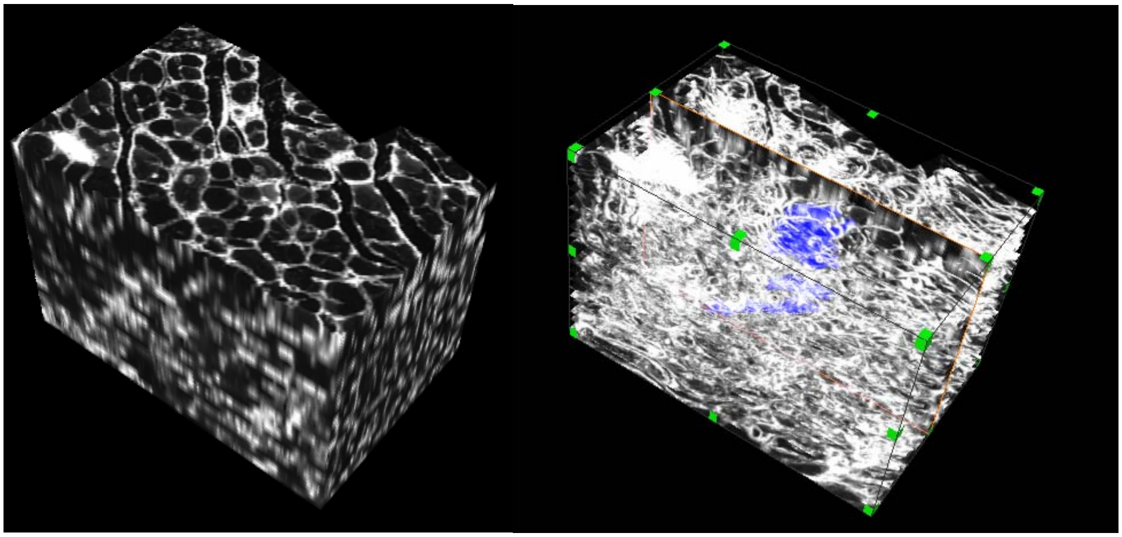


Figure 23. Block of a registered section. On the right only the maleimide is show, the blue is a cell which was tracked over the tissue block.

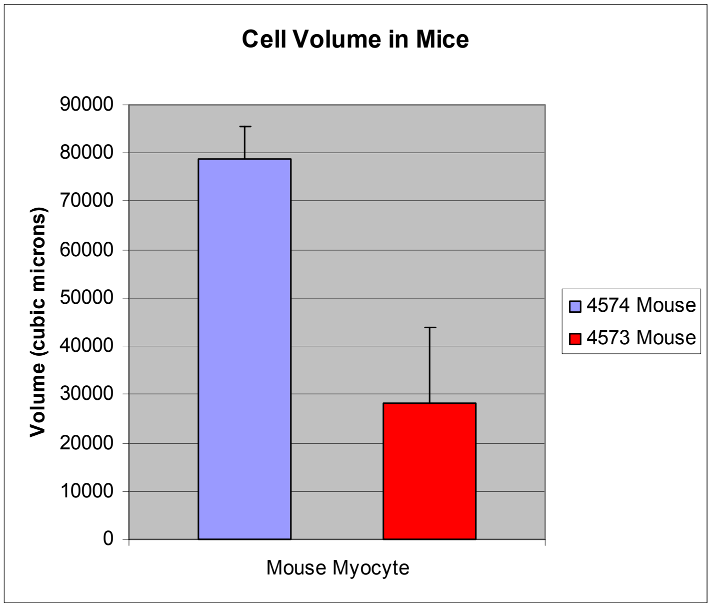


Figure 24. Volumetric data. 4574 mouse had a gene mutation while the 4573 mouse was a wild-type. The samples are 10 heart cells each.

To summarize, the results in this chapter demonstrate that the methods developed in this thesis offer benefits in quantitative analysis of cell structure and its visualization. The software is actively used by the Lee Lab to characterize the cardiac cell structure.

Chapter 6 Discussion and Conclusions

Three dimensional image reconstruction of tissue can only be performed if proper labeling of the tissue is applied. Maleimide staining of the mouse heart through a tail vein injection demonstrated that high quality images can be taken of heart tissue. This staining method was used on the entire heart instead of specific target regions. Using a tail-vein injection for mice avoids tissue damage that occurs if stains are directly injected through a syringe. This is important if the goal is to quantify the shape and any 3D data of healthy cells.

The natural structure of the heart allowed for specific segmentation algorithms to be used. Taking advantage of the spatial structure of the cells allows us to utilize existing segmentation algorithms instead of developing tissue specific methods. Watershed segmentation worked well on cross-sectional slices of heart cells and the use of edge detection algorithms enabled refined classification of cells.

The use of the multi-photon microscope to image tissue has many advantages. The automated setup enables fast scanning. A 1mm x 1mm x 1 mm can be imaged in 6 hours with 1.5 μm sparing between the slices. In addition, since two photon microscopy only excites fluorophores in the regions where the image is taken, photobleaching is not as great a concern. In typical fluorescence microscopy because one exposes more of the tissue than is imaged, excitation photobleaching can cause delays in exposure time to obtain adequate images. Manual imaging is also tedious and the depth resolution is limited to the resolution of the microtome used to create the samples on slides.

Post-processing of registration of images is greatly reduced when images are taken with the multi-photon microscope. Because of the motorized stage, registration is constrained by the fixed offset the stepper motors provide. Only minor correction is needed for registration in the xy plane. In contrast with the manual methods the real time cutting and imaging avoids rotational distortion, making tracking of cells faster. The higher resolution in the z-direction helps in determining where cells begin and end more accurately. In general the multi-photon system enabled faster image acquisition of larger volumes, less registration processes, and easier cell tracking.

Future work should focus on further automation of the three dimensional segmentation of cells using multi-photon images. To achieve this goal, an accurate and efficient method must be created to identify the end points of cells. This in itself is a challenging problem and is worthy of a separate thesis topic. Automating this process will allow for fast calculations of volumes, surface area and cell orientation. Currently the process is tedious, as cells are tracked from image to image by a person. The person has to make a decision on where those cells should end, using the knowledge of the average cell length and taking into account changes to the surroundings of the cell, such as other cells appearing and disappearing. To help with these decisions, nuclear stains could be used to provide extra decision parameters. Using these criteria, among others, automated decision rules or classifiers can be created.

To conclude, the methods developed in this work analyze planar images and recover 3D cell structure from the collection of slices. The algorithms for cell extraction and visualization have been successfully demonstrated in real biological studies of the heart tissue. The experimental validation proved the utility of the methods and highlighted directions for future research. The analysis and visualization techniques described in this thesis represent the first step toward fully automatic volumetric analysis of cell structure.

References

1. Periasamy, A. (2001). Methods in Cellular Imaging, Oxford University Press.

2. McNally J, Karpova T, Conchello J. “Three-Dimensional Imaging by Deconvolution Microscopy.” Methods 19, 373–385 (1999)

3. Centonze V, White J. “ Multiphoton Excitation Provides Optical Sections from Deeper within Scattering Specimens than Confocal Imaging”. Biophysical Journal Vol. 75 Oct. 1998

4. Denk, W. J., Strickler, J. P. and Webb, W. W. Two-photon laser microscopy. United States.(1991)

5. Williams, R. M., Piston, D. W. and Webb, W. W. "Two-photon molecular excitation provides intrinsic 3-dimensional resolution for laser-based microscopy and microphotochemistry." Faseb J. **8**(11): 804-13.(1994)

6. Lewis, J. P. Fast Normalized Cross-Correlation http://www.idiom.com/~zilla/Papers/nvisionInterface/nip.html#eq1:normcc

7. Chen H, Yao D, Li R, Chen W. “A projection-based image registration algorithm and its application.” Brain Topography, Volume 18, Number 1, 47-58 (2005).

8. Sonka M, Hlavac V, Boyle Roger. Image Processing, Analysis and Machine Vision, Brooks/Cole Publishing 1999.

9. Yoshioka J, Prince RN, Huang H, Perkins SB, Cruz FU, MacGillivray C, Lauffenburger DA, Lee RT. “Cardiomyocyte hypertrophy and degradation of connexin43 through spatially restricted autocrine/paracrine heparin-binding EGF.” Proc Natl Acad Sci U S A. 2005 Jul 26;102(30):10622-7.

Appendix

A.1 Registration Code

```
function [xoffset,yoffset,width1]=RLreg(file1,file2,percent)
%Program registers two images together file1 and file2 which have
%horizontal overlap. It outputs the xoffset and yoffset of the second
image
%to the first and the width it starts from. file1 is the image to the
left
%and file2 is the image to the right. This function is used in the
%Plane_register program.
f1=file1;
f2=file2;
f1size=size(f1);  %obtain the size of the image
f2size=size(f2);
xmin1=ceil(f1size(2)*(1-percent)); %overlay dimensions
ymin=1;
width1=f1size(2)-xmin1;
width2=f2size(2)-ceil(f2size(2)*(1-percent));
height=f1size(1);
rect_f1=[xmin1 ymin width1 height];      %crop box for f1
rect_f2=[1 ymin width2 height];         %crop box for f2
sub_f1 = imcrop(f1,rect_f1);            %crop images
sub_f2 = imcrop(f2,rect_f2);
sub_f1 = im2bw(sub_f1,.5);  %turn image to black and white images 0 or
1 for faster computation
sub_f2 = im2bw(sub_f2,.5);
c=normxcorr2(sub_f1(:,:,1),sub_f2(:,:,1));  %calculate normalized cross
correlation
[max_c,imax]=max(c);
xpeak=find(max_c==max(max_c)); %find the maximum value for the x-
coordinate
ypeak=imax(xpeak); %find where the y-coordinate of the max value
occures
corr_offset = [(xpeak-size(sub_f1,2)),(ypeak-size(sub_f1,1))]; %find
the offsets
% total offset
offset = corr_offset;
xoffset = offset(1);
yoffset = offset(2);
```

```
function [xoffset,yoffset,height1]=TBreg(file1,file2,percent)
%registers two images along the top row and gives the overlap of file1
to
%file2 in an x, file1 is the bottom file, percent is the percent
overlap
%you want to check
%created by Francisco Cruz
f1=file1;
f2=file2;
f1s=size(f1);  %gives [y x]
f2s=size(f2);
xmin= 1;%overlay dimensions
ymin2=ceil(f1s(1)*(1-percent));
width=f1s(2);
height1=f1s(1)-ceil(f2s(1)*(1-percent));
height2=f2s(1)-ceil(f2s(1)*(1-percent));
rect_f1=[xmin 1 width height1];      %crop box for f1
rect_f2=[xmin ymin2 width height2];         %crop box for f2
sub_f1 = imcrop(f1,rect_f1);            %crop images
sub_f2 = imcrop(f2,rect_f2);
sub_f1 = im2bw(sub_f1,.5);
sub_f2 = im2bw(sub_f2,.5);
c=normxcorr2(sub_f1(:,:,1),sub_f2(:,:,1)); %find the best correlation!
%find the total offset by correlation
[max_c,imax]=max(c);
xpeak=find(max_c==max(max_c));
ypeak=imax(xpeak);
corr_offset = [(xpeak-size(sub_f1,2)),(ypeak-size(sub_f1,1))];
% total offset
offset = corr_offset;
xoffset = offset(1);
yoffset = offset(2);
```

```
clear;
%This program registers a set of images inputed by the user.
%This program calls on the functions TBreg and RLreg in order to
register
%images. The user inputs a matrix in which the images are laid out. The
%user then inputs a percentage which the images are overlapping. This
%percent is the max overlap that any two images overlay on each other.

%Enter the matrix positions and percent overlap
layout_string=inputdlg('Overlay pattern','overlay pattern');
layout=str2num(layout_string{1});
percentage=inputdlg('Percent of Overlap','Percent Overlap of Images');
percent=str2num(percentage{1})/100;
number_of_images=layout(1)*layout(2);
reference=[1:number_of_images];
reference=reshape(reference,layout(1),layout(2))
%Get and read the images
for k=1:number_of_images
    [filename, pathname]=uigetfile('*.*',['select image' int2str(k)])
    cd(pathname);
    F(:,:,k)=imread(filename);
end
%Create a layout for the images
Fsize=size(F);
dimensions_of_final=[layout(1)*Fsize(1)+200 layout(2)*Fsize(2)+200];
final_image=uint8(zeros(dimensions_of_final));
xrlshift=[0];xrlshiftN=[0];
yrlshift=[0];yrlshiftN=[0];
xtbshift=0;ytbshift=0;
xtb=[0];ytb=[0];xl=[0];yrl=[0];
XShift=[0]; YShift=[0];
%Perform registration on images which overlap vertically
if layout(1)>1
    for j=1
        for i=1:layout(1)-1

[xtb(i),ytb(i),h(i)]=TBreg(F(:,:,reference(i+1,j)),F(:,:,reference(i,j)
),percent);
            XShift(i+1,j)=XShift(i)+xtb(i);
            YShift(i+1,j)=YShift(i)+Fsize(1)+ytb(i)-h(i);
        end
    end
else
end
%Perform registration on images which overlap horizontally
if layout(2)>1
    for j=2:layout(2)
        for i=1:layout(1)
            reference(i,j)
            [xrl(i,j-1),yrl(i,j-1),w(i,j-1)]=RLreg(F(:,:,reference(i,j-
1)),F(:,:,reference(i,j)),percent);
            XShift(i,j)=XShift(i,j-1)+Fsize(2)-xrl(i,j-1)-w(i,j-1);
            YShift(i,j)=YShift(i,j-1)-yrl(i,j-1);
        end
    end
else
end
```

```
XShift
YShift
%Construct the final image
for j=1:layout(2)
    for i=1:layout(1)

final_image(201+YShift(i,j):200+Fsize(1)+YShift(i,j),201+XShift(i,j):20
0+Fsize(2)+XShift(i,j))=F(:,:,reference(i,j));
    end
end
figure(33)
imshow(final_image)
```

Rotate_Image.m

```
[filename, pathname]=uigetfile('*.*',['select image' int2str(k)])
cd(pathname);
untransformed = imread(filename);
Rotation_Shift_Matrix = cp2tform(input_points,base_points, 'linear
conformal');
m=Rotation_Shift_Matrix.tdata.T;
scale=sqrt(m(1,1)^2+m(1,2)^2);
replace=[m(1:2,1:3)/scale; m(3,1:3)];
New_Matrix= maketform('projective',replace);
transformed = imtransform(untransformed, New_Matrix);
imshow(transformed)
```

A.2 Segmentation Code

```
function output = bwcfind(bwimage)
%this function is used in preparation to the watershed program, this
finds
%the centroids of the objects in the image and imposes a minima on them
l = bwlabel(bwimage);
props = regionprops(l, 'Centroid');
Max= size(props, 1);
n=0;
output = zeros(size(bwimage));
while n < Max
    n= n+1;
    x= floor(props(n).Centroid(1));
    y = floor(props(n).Centroid(2));
    if x > (size(bwimage,2)-2) |  y > (size(bwimage,1)-2)
        output(y, x) = 1;
    else
        output(y:y+2, x:x+2) = 1;
    end
end
ouput = uint8(output);
```

```
function masked_image = image_mask(image, mask, mask_value, value)
%this program creates a mask for the images before performing watershed
%this imposes the minima from bwcfind
masked_image = image;
if ndims(image) > 2
    logical_matrix(:,:,1)= ismember(double(mask), mask_value);
    logical_matrix(:,:,2)= logical_matrix(:,:,1);
    logical_matrix(:,:,3)= logical_matrix(:,:,1);
    masked_image(logical_matrix) = value;
else
    logical_matrix = ismember(double(mask), mask_value);
    masked_image(logical_matrix) = value;
end
```

```
clear;
%this program allows the user to select features from an image, such as
%cells, and other objects which are round and are light in color.
[file, pathname]=uigetfile('*.*','Select Image');
cd(pathname)
threshold_value=inputdlg('Threshold value','Threshold value',1,{'25'});
thresh=str2num(threshold_value{1});
slice=imread(file);
figure(1), imshow(slice), title('original');
% % % % % % % % % % % % % % % % % % % % % % % % % %
[Asmall,Rsmall]=imcrop(slice); % Crop the area of interest
% % % % % % % % % % % % % % % % % % % % % % % % % %
Main=Asmall(:,:,1);
Main_a = imadjust(Main,[.2 .9],[]); % If needed adjust the values,
these change the contrast of the images between the black and white
features
% % % % % % % % % % % % % % % % % % % % % % % % % %
Seed_BW = imextendedmin(Main,thresh);
Seeded_BW = bwcfind(Seed_BW);
% Prepare the image for watershed segmentation
Wat_BW = imimposemin(Main, Seeded_BW);
Wat_BW_final = watershed(Wat_BW);
Img_Wat = image_mask(Main_a, Wat_BW_final, 0, ...
       max(Main_a(:)));
% % % % % % % % % % % % % % % % % % % % % % % % % %
%figure(89), imshow(Img_Wat)
BWms=edge(Img_Wat, 'prewitt', .1); %Change the value if the threshold
is too low or high, this calls on edge detection algorithm
se90 = strel('line', 2, 90);
se0 = strel('line', 2, 0);
BWmsdil = ~imdilate(BWms, [se90 se0]); %This connects the edges and
makes whole objects and also inverts the images so the objects are now
white instead of black
BWmdfill = imfill(BWmsdil, 'holes'); %Fill in the holes in the objects
BWmnobord = imclearborder(BWmdfill, 4); %Get rid of objects touching
the borders, so only whole cells are counted
BWmfinal = BWmnobord;
BWgoutline = bwperim(BWmfinal); %Create the outline for the cells
BWmlabel=bwlabel(BWmfinal);
stats=regionprops(BWmlabel);
% % % % % % % % % % % % % % % % % % % % % % % % % % %
h=fspecial('disk',2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
BWgoutline=bwperim(BWmlabel);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%The following produces the code in order to pick out cells from the
figure
figure(1), imshow(BWmdfill),title('water')
Overlap(:,:,1)=Asmall;
Overlap(:,:,2)=Asmall;
Overlap(:,:,3)=Asmall;
segsize=size(Overlap);
figure(4)
imshow(Overlap)
WantedCell=zeros([segsize(1) segsize(2)]);
button=0;
N=0;
```

```
while button~=2
    [x,y,button]=ginput(1);
    Object=BWmlabel(round(y), round(x))
    if Object>0 && button==1
        N=N+1;
        idx2=find(BWmlabel==Object);
        WantedCell(idx2)=BWmfinal(idx2);
        Overlap(:,:,1)=uint8(double(150*WantedCell)/2+double(Asmall));
        imshow(Overlap)
        centroid_of_cell=stats(Object).Centroid;
        area_of_cell=stats(Object).Area;
        centroids(N,:)=[[Rsmall(1) Rsmall(2)] + centroid_of_cell];
        Areas(N,:)=area_of_cell;
    end
end
Areas
```

A.3 Image Normalization for Multi-photon microscope

```
function [S,Mean_S]=ave_192(M)
for i=0:5
    for j=0:5
        Square=M(i*32+1:i*32+32,j*32+1:j*32+32);
        Mean_S(i+1,j+1)=mean(mean(Square));
    end

end
Mean_S;
mean_of_mean=mean(mean(Mean_S(2:5,2:5)));
S=M;
for i=[0 5]
    for j=0:5

S(i*32+1:i*32+32,j*32+1:j*32+32)=M(i*32+1:i*32+32,j*32+1:j*32+32).*(mea
n_of_mean/Mean_S(i+1,j+1));
    end
end

for i=[1:4]
    for j=[0 5]

S(i*32+1:i*32+32,j*32+1:j*32+32)=M(i*32+1:i*32+32,j*32+1:j*32+32).*(mea
n_of_mean/Mean_S(i+1,j+1));
    end
end
```

A.4 Multi-plane interpolation

```
% program will take a line that the user specifies and plot it across
all
% frames of a broken movie to see reconstruct what a z-frame would look
% like.
% select first image, then the image (slightly noise reduced) will
display.
%  Draw a line that you wish to get the z-info from and then hit enter
%  (must click at both beginning and end of the line before hitting
enter.
%  The program will process and display (the transpose) of the
extracted
%  z-sections.
clear;
[filename, pathname]=uigetfile('*.*','Select first frame');
if filename
    imagepath=pathname;
    cd(pathname)
    file=fullfile(pathname,filename);
    [path,fname,ext]=fileparts(filename);
    maxN=CountFrames(filename,pathname);
    disp(['analyzing the files: ', fname, ext])
    disp(['  detected ', num2str(maxN), ' frames'])
    nfile=file;
% display first image and let the user select a line
    nimg=imread(file);
  %  nimg=rgb2gray(nimg);
    [na,nb]=size(nimg);
    % some image processing for reducing noise
    %mim=mean2(nimg);
    for i=1:na
        for j=1:nb
            ntemp=double(nimg(i,j))/255;
            ntemp=uint8(255*ntemp*ntemp);
            nimg(i,j)=ntemp;
%           if nimg(i,j)<mim
%               nimg(i,j)=0;
%           end
        end
    end

    im2an=imshow(nimg);
% let the user draw a line
    [cx,cy,c,xi,yi]=improfile(im2an);
    [ca,cb]=size(c);
% ca is the length of c
    newdata=zeros(ca,maxN);
    newdata(:,1)=c;
    nFile=file;
    for i=1:maxN-1
        file=nFile;
        nFile=FindNextFile(file);
        nimg=imread(file);
        nimg=rgb2gray(nimg);
%       mim=mean2(nimg);
```

```
        for k=1:na
            for j=1:nb
            ntemp=double(nimg(i,j))/255;
            ntemp=uint8(255*ntemp*ntemp);
            nimg(i,j)=ntemp;
 %          if nimg(k,j)<mim
 %              nimg(k,j)=0;
 %          end
            end
        end
        xlabel(['frame ', num2str(i)]);
        d=improfile(nimg,xi,yi,ca);
        newdata(:,i+1)=d;
    end
end
newconstruct=uint8(newdata);
imshow(newconstruct);
fileout=[path,fname,'z',ext];
imwrite(newconstruct,fileout);
```

```
%% Sub-function FindNextFile
function [filename,num]=FindNextFile(currfile)
[path,fnamsh,ext]=fileparts(currfile);
%determine current file number
i=1;
while length(str2num(fnamsh(end-i:end)))
    i=i+1;
end
num=str2num(fnamsh(end-i+1:end));
if num>=0         % only perform analysis if file contains number in
the filename
    % determine the following number
    n=num+1;
    N=length(num2str(n));
    nextfile1=[fnamsh(1:end-N), num2str(n), ext];
    if mod(log10(n),1)
        nextfile2=nextfile1;
    else
        nextfile2=[fnamsh(1:end-N+1), num2str(n), ext];
    end
    if exist(fullfile(path,nextfile1))==2
        filename=fullfile(path,nextfile1);
    elseif exist(fullfile(path,nextfile2))==2
        filename=fullfile(path,nextfile2);
    else
        filename='-1';
    end
else
    filename='-1';
    num=-1;
end
```

```
%% Sub-function CountFrames
function Frames=CountFrames(filename,pathname)
n=0;
nextname=fullfile(pathname,filename);
while (~strcmp(nextname,'-1') & n<10000)
    n=n+1;
    prevname=nextname;
    nextname=FindNextFile(prevname);
end
Frames=n;
```