Winter 12-15-2015

Segmentation and Strain Mapping of a Beating Heart

Kenna Middleton
Washington University in St. Louis

Recommended Citation

Middleton, Kenna, "Segmentation and Strain Mapping of a Beating Heart" (2015). *McKelvey School of Engineering Theses & Dissertations*. 123.
https://openscholarship.wustl.edu/eng_etds/123

Washington University in St. Louis

School of Engineering and Applied Science

Department of Mechanical Engineering

Thesis Examination Committee:
Dr. Guy Genin
Dr. Robert Pless
Dr. Srikanth Singamaneni

Segmentation and Strain Mapping of a Beating Heart

by

Kenna Middleton, B.S. Mechanical Engineering, M.S. Mechanical Engineering

A thesis presented to the School of Engineering and Applied Science
of Washington University in partial fulfillment of the
requirements for the degree of

Master of Science

December 2015
Saint Louis, Missouri

Contents

List of Figures

Acknowledgments

I would like to thank all financial aid and scholarship support that I have received over the years that allowed me to continue my education.

A special thanks goes to the many graduate students and distinguished faculty within my department who have reviewed this thesis and helped support the related research.

Kenna Middleton

Washington University in Saint Louis
December 2015

Dedicated to my parents and my fiance.

ABSTRACT OF THE THESIS

Segmentation and Strain Mapping of a Beating Heart

by

Kenna Middleton

Master of Science in Mechanical and Bio-medical Engineering

Washington University in St. Louis, December 2015

Research Advisor: Professor Guy Genin

Conventional methods of estimating deformation optically often compare image intensities before and after deformation to help estimate the strain fields. However, these current methods often introduce undesirable error that propagates through the system, growing larger through each strain calculation. These errors create false, concentrated regions of strain throughout the image. Additionally, gathering and evaluating images in three- and four-dimensional space creates even more difficulty for conventional methods. Here, by implementing new techniques of gathering four-dimensional ultrasound data sets and calculating deformation fields with traditional errors reduced three-fold, four-dimensional images were evaluated to create a novel visualization tool used for representing the strains in a four-dimensional modeling world. In general, this computer program has applications across a wide range of disciplines.

Chapter 1

Introduction

The problem that was addressed was that strain mapping is difficult to visualize. This thesis solved the problem by creating a visualization of three-dimensional and four-dimensional strain mapping while also segmenting out extraneous parts of an images such as blood swirling in a vein. This visualization technique has many applications because creating techniques to analyze images in order to study their deformation characteristics and strain fields is important in several different fields of study.

In the field of tissue mechanics, stress and strain are the key to understanding formation and prevention of injury. Studies have been conducted on the ability of embryos to heal any wounds rapidly with no evidence of scaring [9]. Strain mapping this wound healing process could lead to many advances in the medical field of injury repair. Studying musculoskeletal tissue is another important aspect of research in order to create noninvasive techniques of evaluating the human body. For example, *in vivo* techniques of strain mapping ligaments are being studied and because “ligaments are complex three-dimensional structures that exhibit highly non-homogeneous strain patterns” [7] three-dimensional models that show the strain mapping can be valuable resources in determining and studying strain fields. Additionally, strain mapping techniques play an important role in understanding intricate cell mechanics such as the “exchange of physical forces in cell-cell and cell-matrix interactions” [5].

Alternatively, stress analysis is not limited to tissue mechanics. Stress mapping extends to structural mechanics and geological features as well. Studying infrastructure such as bridge deflection under the load of a heavy cargo truck using digital image correlation [10] is an important area of research with many more viable options. And finally, applying the following technique to geological structures could prove vital in understanding and evaluating fault zone architecture [3].

1.1 Challenges

Challenges faced during this project included determining ways to segment out the blood from the heart and the heart from the mouse chest cavity. Several automatic processes were tested but the most accurate method determined was manually plotting the sections that were desired. Another challenge was to determine the best implementation for combining the four-dimensional ultrasound data set, the three-dimensional binary mask, and the four-dimensional stress data set into a comprehensive visualization tool.

1.2 My Role

My role within this project involved manually creating a three-dimensional binary mask that removed the blood from the ultrasound images as well as segmenting out the heart from the mouse's chest cavity. This process used a MATLAB GUI that allowed the user to cycle through the z-axis of a three-dimensional data set while drawing a mask at each point on the z-axis. I then used these three-dimensional binary masks of the mouse heart and the blood within the left ventricle to create a three-dimensional data set that contained only the pixels of the mouse heart with no blood. This binary data set was vital in the process of creating a visually stimulating three-dimensional image of a mouse heart that showed the projected strains throughout time. This technique is very useful in creating a visualization tool for a wide range of fields.

1.3 Thesis Overview

This thesis will describe in detail how the ultrasound images were collected, what was done to the ultrasound images to create a collective four-dimensional data set, the strain calculations done on the four-dimensional data set, how the blood from the mouse heart as well as the mouse heart itself was segmented out of the images, and the final display of a three-dimensional mouse heart changing with time in video format.

Chapter 2

Methods

Several methods were used in the process of segmenting and visualizing a beating heart. First, the ultrasound image was collected by using a ultrasound machine. Second, the strains of the heart were calculated using a revolutionary technique of warping images. Then the swirling blood in the ultrasound images were removed by manually creating binary masks. And lastly, the final images were mapped onto a three-dimensional mouse heart figure that changed with time. The following describes these four methods in detail.

Figure 2.1: Vevo2100 system used for a wide range of animal ultrasound imaging and four-dimensional development (www.vevo2100.com).

2.1 Ultrasound Data Acquisition

The ultrasound data used within this research was provided by Purdue University. The heart imaged used was from a C57BL/6 wild-type mouse and the four-dimensional data sets were collected by using a 40 MHz US probe (MS550D, Vevo2100, VisualSonics Inc.) (Figure 2.1).

Within the four-dimensional data sets, the x dimension represented the horizontal position along the ultrasound Transducer Array, the y dimension represented the vertical, in-plane depth of the ultrasound signal, the z dimension represented the through-plane position of the ultrasound probe along the mouse, and the t dimension represented the time position of the three-dimensional volume (x,y,z) in the cardiac cycle.

Specifically, the three-dimensional data sets comprising the four-dimensional volume were obtained by moving a transducer across the heart cavity while gating the heart at 8 ms consecutive delays. The gating occurred after the R-peak in the electrocardiogram (ECG) signal. The ultrasound probe was attached to a motor to traverse the mouse in a linear manner [4].

After all the three-dimensional ultrasound volumes were collected, common image registration techniques were used on each volume to temporarily warp the three-dimensional image to the temporary neighboring volume. This technique allowed the segmented three-dimensional volumes to be projected across the entire four-dimensional data set (Figure 2.2).

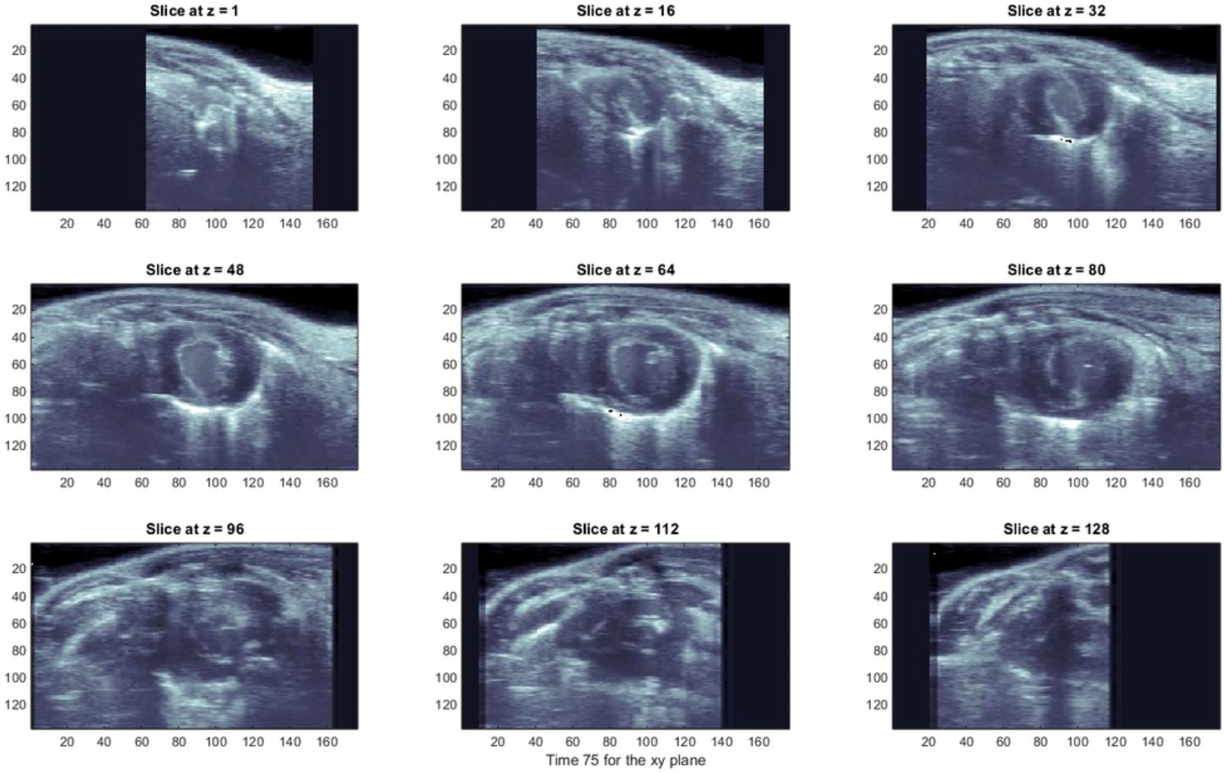


Figure 2.2: Multiple locations through the z direction of the ultrasound data of the mouse heart cavity.

2.2 Strain Mapping in Multiple Dimensions

In several of the areas of research mentioned in the introduction, the strain mapping techniques used most often employs correlating image intensities before and after deformation in order to determine displacement fields [8]. To improve old methods of transferring displacement fields to spatial strain tensors, the Lucas-Kanade algorithm can be applied. The Lucas-Kanade algorithm (LK) utilized an arbitrary warping function, W(x;p), in image alignment technique [1]. By examining the structure of the image difference between the undeformed and deformed images, optimization theory is used to estimate the warping function. This process is applied over and over again until the difference between the deformed image and the warped image is minimized.

$$min\,[warped - deformed]^2 \tag{2.1}$$

$$min \sum_x [I(W(x;p)) - T(x)]^2 \tag{2.2}$$

However, these techniques introduce errors when there are large concentrations of strain. Specifically, when small errors are introduced, the warping and calculating deformation systems carry the errors through the system, creating even larger inaccuracies. This problem creates false strains in the images.

To combat these errors, two algorithms were created: a simple 'Direct Deformation Estimation' (DDE) and 'Strain Inference with Measures of Probable Local Elevation' (SIMPLE) [2]. These algorithms reduce error by "directly incorporating the calculation of deformation [gradient tensor] into the warping function" [2].

$$W(x;p) = \begin{bmatrix} F_{11} & F_{12} & T_1 \\ F_{21} & F_{22} & T_2 \\ 1 & 1 & 0 \end{bmatrix} \tag{2.3}$$

Therefore, estimation of displacement fields are no longer needed before estimating the strain fields. Because of the remarkable accuracy of these newly developed algorithms, DDE and SIMPLE were used in evaluating the strain estimation and concentration for the four-dimensional ultrasound mouse heart data set.

2.3 Segmentation

One of the main problems was to remove the blood located in the left ventricle from the four-dimensional ultrasound image. The reason behind this was that the blood would swirl as time progressed, giving false strain data. In addition to removing the blood, the mouse heart itself needed to be segmented out from the mouse's body in order to better visualize the strain mapping on the heart.

The first attempt was to determine an automatic process of locating the center of the heart and the center of the left ventricle and from there, determining the edges of the heart and left ventricle. To start, different edge detection techniques (Figure 2.3) were evaluated and determined to be insufficient in determining the edges of the heart and left ventricle because there was too much noise within the ultrasound image.

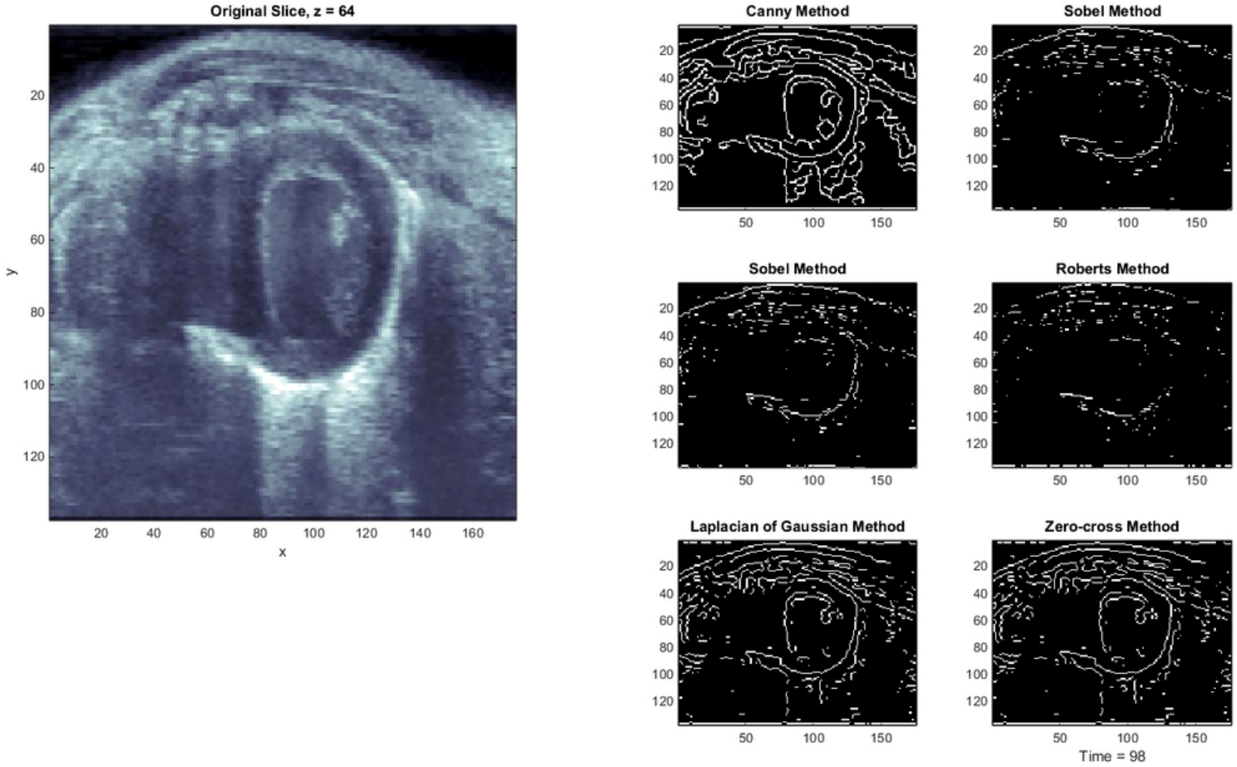


Figure 2.3: Representation of the results of each edge detection tecnique next to their corresponding ultrasound image.

The other method attempted was to evaluate the contour of the ultrasound mouse heart (Figure 2.4). As before with the other edge detection processes, the contour plots throughout the z direction of the mouse heart could not be evaluated for the ventricle edge or the mouse heart edge because of the noise contained within the four-dimensional ultrasound data set.

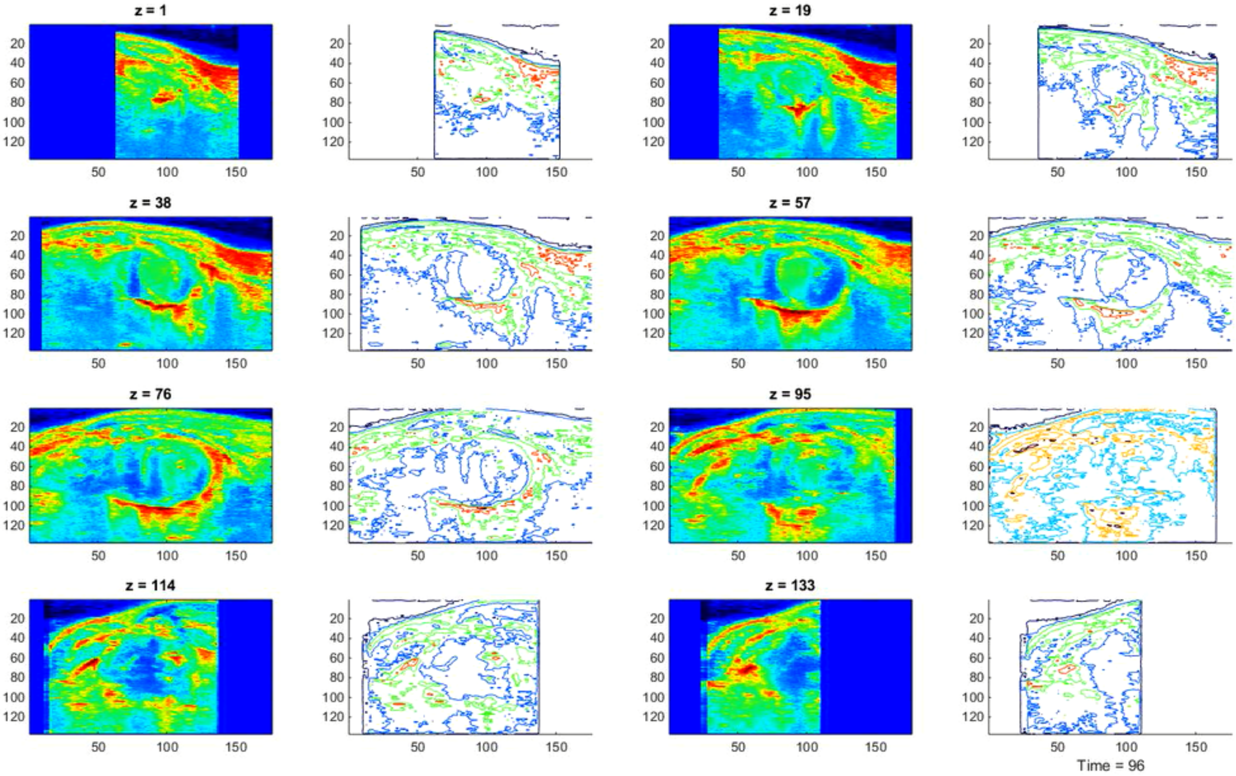
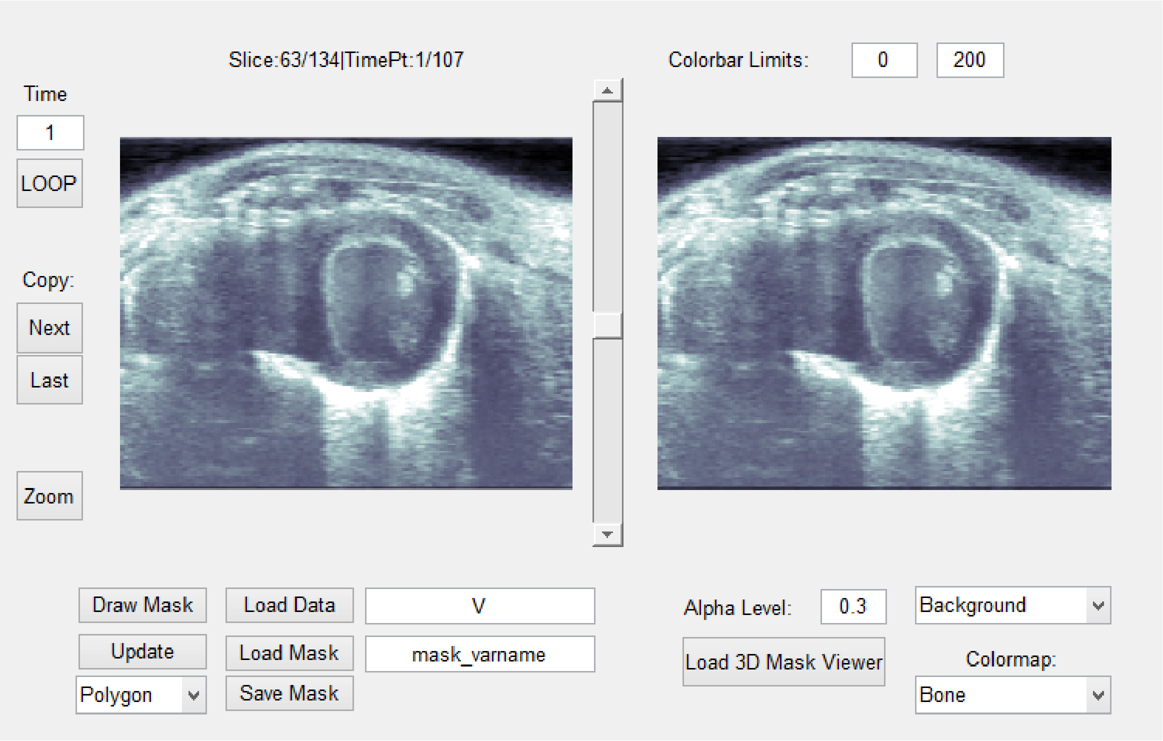


Figure 2.4: Representation of the results of each contour plot next to their corresponding ultrasound image.

The final technique used in determining the edges of the mouse heart and the left ventricle was to manually locate the edges. This was done by creating a GUI in MATLAB that allowed the user to manually plot a binary mask over a specific slice at a specified time (Figure 2.5). The steps are the following.



Slice:63/134|TimePt:1/107

Colorbar Limits: 0 200

Time 1

V

mask_varname

Alpha Level: 0.3

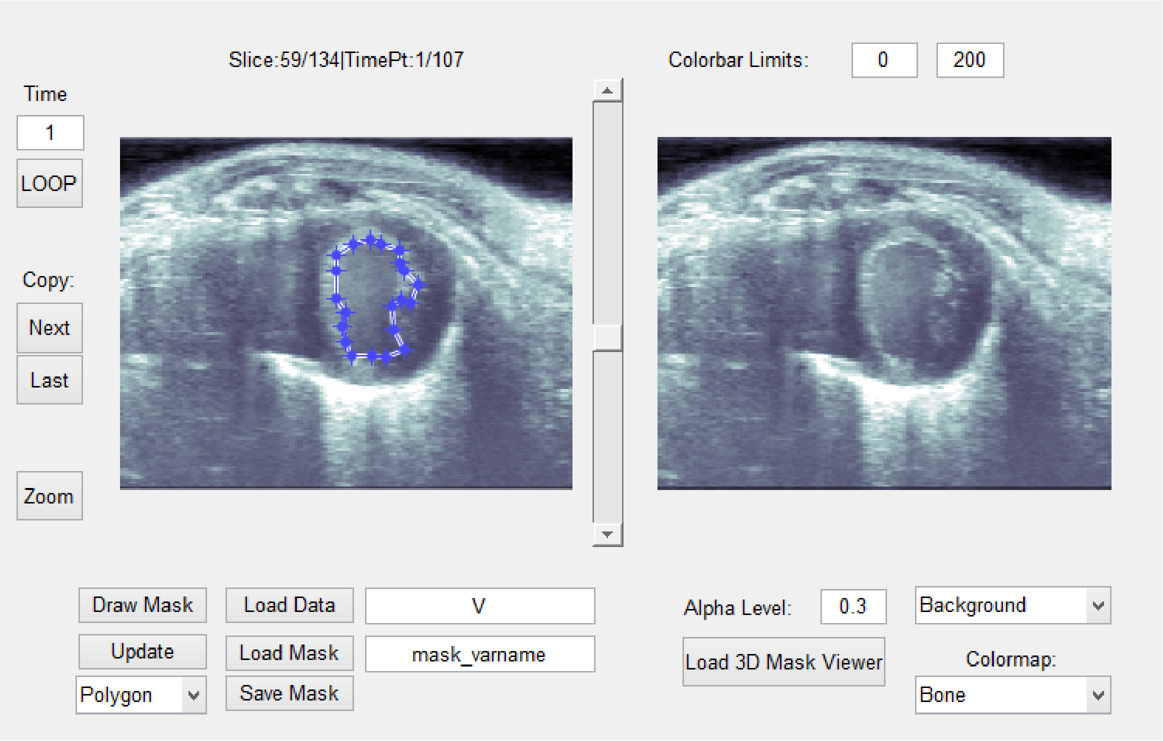
Background

Colormap: Bone

Polygon

Figure 2.5: Initialization of the MATLAB GUI program.

First, the four-dimensional data set is loaded into the define_4DUSmask GUI by using the "Load Data" Button. This then displays the image to be viewed in Gray, Bone, or Jet Colormap. The scroll bar was then used to determine the initial location where the heart could be clearly seen. Once the initial location was found, the "Draw Mask" Button was used to begin drawing the desired segment on the heart (Figure 2.6). Once the mask was drawn on one slice of the mouse heart, the binary mask could be displayed as a red, transparent area on another view of the mouse heart slice in order to determine if the drawn mask was desired (Figure 2.7). The "Draw Mask" process continued until the desired mask was no longer clearly defined. Once completed, the entire mask was viewed by using the "Load 3D Mask View" Button (Figure 2.8).



Slice:59/134|TimePt:1/107

Colorbar Limits: 0 200

Time 1

Alpha Level: 0.3

Background

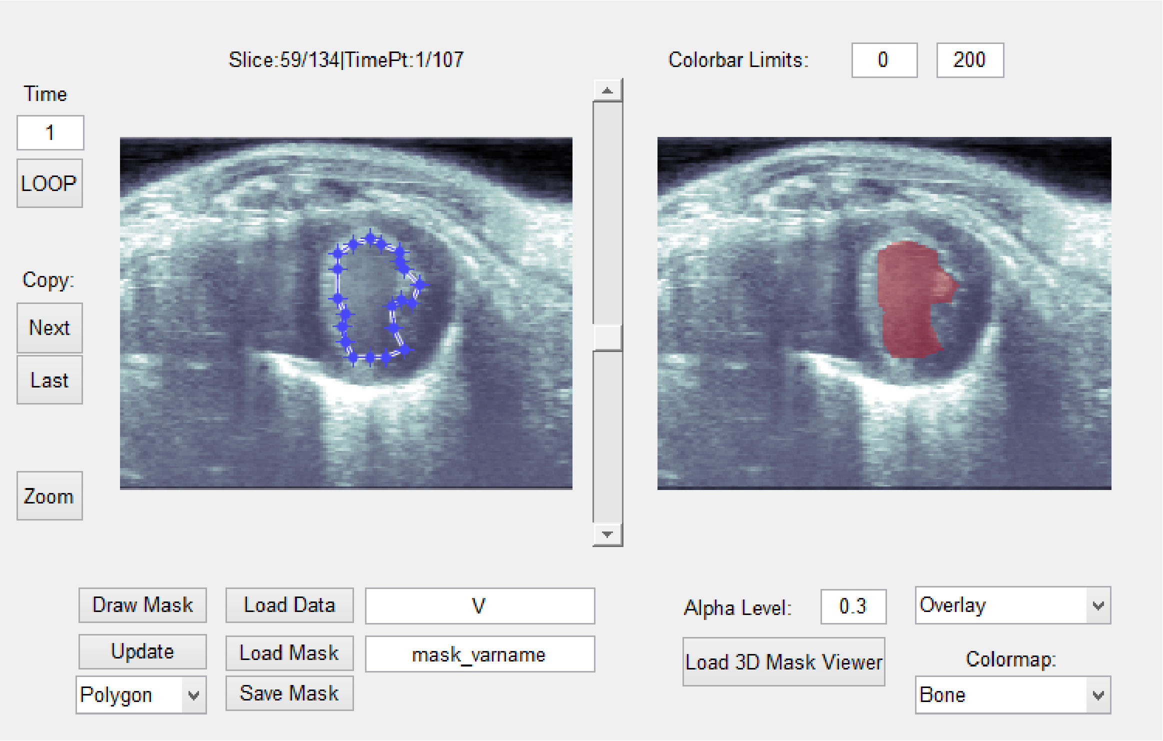
Colormap: Bone

Polygon

V

mask_varname

Figure 2.6: This figure shows the process of manually drawing a binary mask onto the mouse heart image for a specified z-location.



Slice:59/134|TimePt:1/107

Time: 1

Colorbar Limits: 0 200

V

mask_varname

Polygon

Alpha Level: 0.3

Overlay

Colormap: Bone

Figure 2.7: A representation of the Draw Mask button feature.

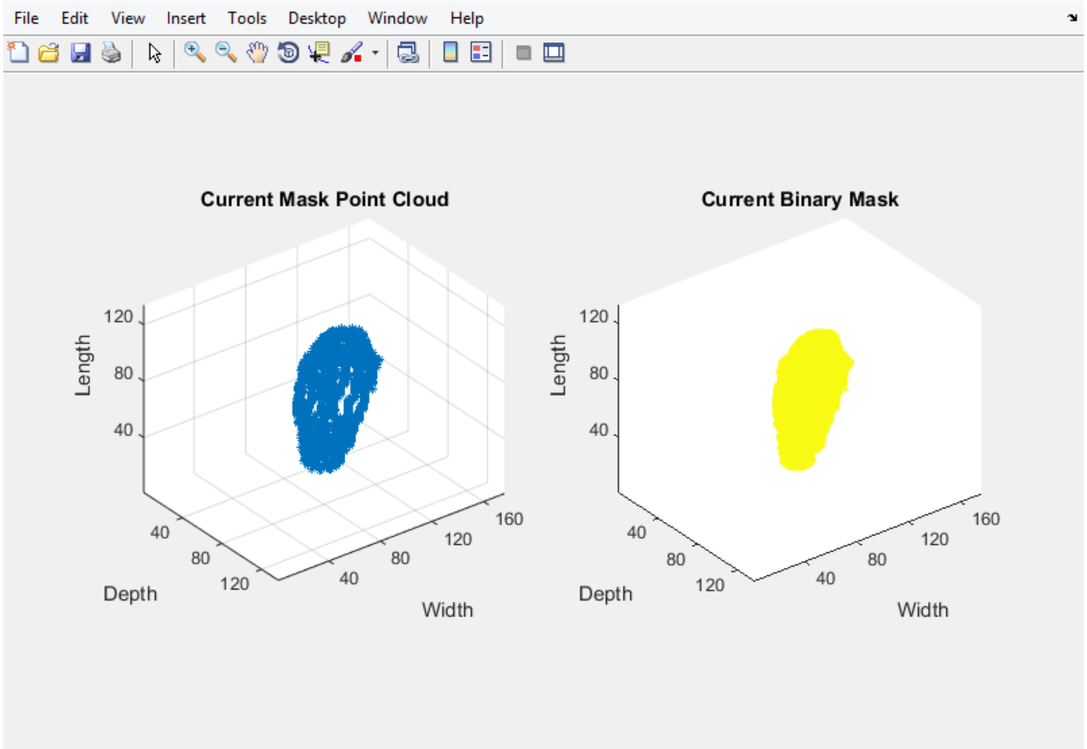


Figure 2.8: A representation of the Overlay button feature.

This manual masking process was completed in the first time frame for the blood inside the left ventricle and the outline of the mouse heart. The segmentation process was vital in implementing the visualization portion.

2.4 Visualization

Once the three-dimensional binary mask of the mouse heart with the blood removed from the left ventricle was complete, it was blended with the first time frame of the three-dimensional ultrasound mouse heart data. This allowed for the creation of an isosurface with end-cap geometry (isocap). The strain of the mouse heart was then assigned to the isosurface vertices that matched its geometric position. This was process was also used for mapping the top and bottom ultrasound mouse heart slice onto the isocaps. The strain was processed with a “jet” colormap and the ultrasound was kept as a “bone” colormap (Figure 2.9). The code used can be found in Appendix A.

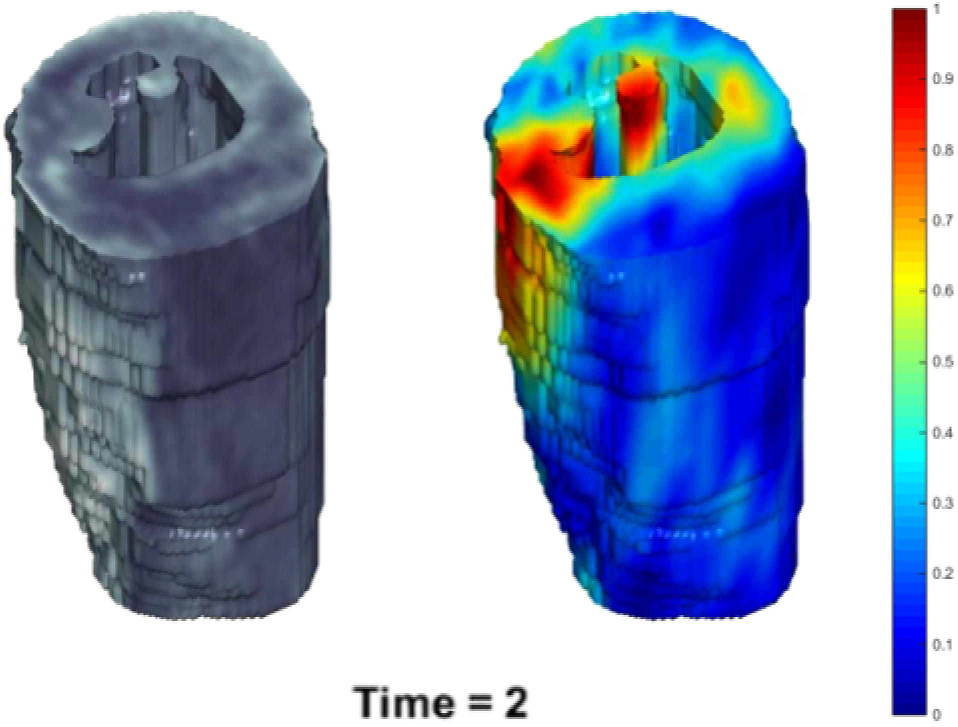


Figure 2.9: Three-dimensional mouse heart with the strains mapped onto the sides of the heart.

Chapter 3

Results

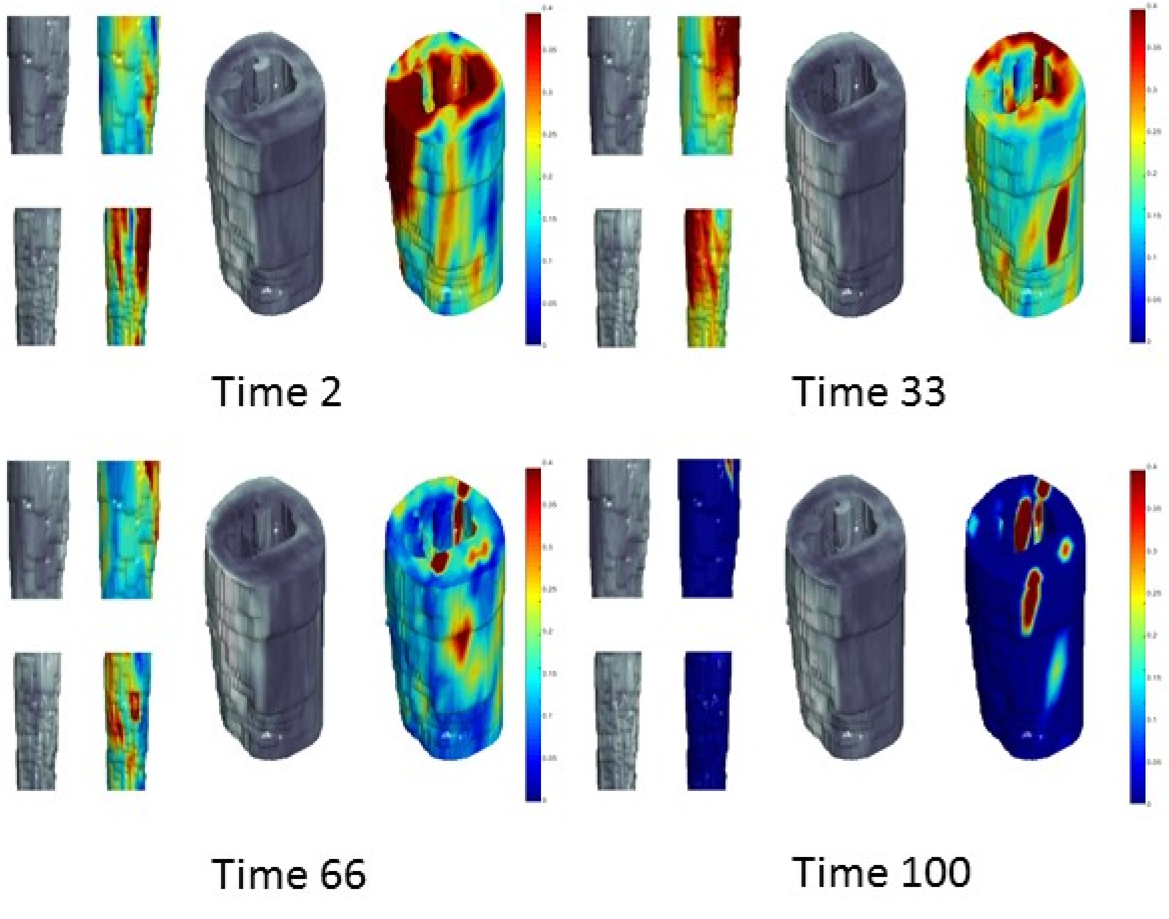


Figure 3.1: Final captured video image of the three-dimensional mouse heart with the strains mapped onto the sides. Each time frame shows three different views of the mouse heart where the ultrasound data is represented in a gray scale coloring and the strain data is represented in a jet scale coloring.

The final result from the implementation of the visualization computer program shows three different views of the heart for four different time frames (Figure 3.1). Each side view is spaced by 120 degrees around the heart. The larger mouse heart in the images shows an angled view of the heart. Thus, the stresses located inside the heart can be visualized. Each view of the mouse heart includes one ultrasound isosurface (the grey scale images) and one strain mapping isosurface (the jet scale images). The final output from the computer program creates a video to show how the strains on the mouse heart change over time.

Additionally, the strains above 0.4 on a 0 to 1 scale were 'cut out' by making the strains from 0.4 to 1 appear red and redistributing the jet colormap equally between 0 and 0.4 (Figure 3.1). This allowed for the lower strains to be better visualized.

Chapter 4

Discussion

The strains located on the heart show a color range, called 'jet' in MATLAB, that go from blue to red. The blue shows that the heart is under no strain and the red shows that the heart is under large amounts of strain. During the video, there are a couple of specific areas that are persistent with high regions of strain. This is due to the fact that the displacement of the heart was used to determine the location of the strains. Thus, the strains located in the figure are most likely external to the heart. This means that the displacement calculations are not taken with respect to the initial time frame. Further implementation and updating of the strain analysis in a four-dimensional world will lead to more accurate results.

Chapter 5

Conclusions

5.1 Main Points

Overall, several techniques were used in the implementation of creating a four-dimensional representation of a mouse heart. By gathering the four-dimensional ultrasound data set of the mouse heart [4], strain fields were able to be calculated by using novel DDE and SIMPLE algorithms [2]. These algorithms reduced previous errors producing more accurate strain mapping and displacement field calculations than other image-processing methods.

Once the strains were calculated for the four-dimensional data set, the first time frame of the four-dimensional data set was used to create a binary mask containing the outer edges of the heart and the edges of the left ventricle. This binary mask was then used to create the body of the heart. The four-dimensional strain fields were mapped to the outside body of the heart and the original ultrasound data was used for the slices through the heart. This resulted in a final video that displayed the heart and its associated strains elapsing through time.

5.2 Future Directions

Further implementations would be to first, update the strain mapping to have the strains remain in their original positions relative to the first frame. This will create a more accurate depiction in the produced video. Second, it would be to use a similar warping technique for mapping strains to map the outer perimeter of the heart and the inner edge of the left ventricle through time. Thus, deforming the heart over time produces a four-dimensional binary mask that would be further implemented in the computer visualization program. The end result would be a computer-generated beating heart with strain mapping. Then, once the program has reached this stage, four-dimensional images from studies such as embryonic wound healing [9], determination of ligament strains [7], and fault zone architecture [3] could be evaluated and presented in an easy to understand visualization technique.

Additionally, a future implementation of the heart visual would be to create a user friendly GUI that would provide features such as a scroll bar that could be used to move through time and a rotation tool allowing user-specified view positions of the heart. As well as an automated program that segments out the heart without having needing a hand drawn binary mask.

Appendix A

MATLAB Code Used to Generate Four-Dimensional Video

Contents

```
function p1 = view4DheartsStationary
```

Load Mask and strain Data

```
MH = load(’MouseHeartNoBlood.mat’);
mask = MH.mhNoBlood; %3D data image
V = matfile(’volume.mat’);
Vh = V.Vh;  %strain images
```

Trim Data with User Specifications

Find Colorless Z-Planes

```
z = [];
for i = 1:size(mask,3)
    if mask(:,:,i) == zeros(size(mask,1),size(mask,2))
        z = [z;i];
    elseif min(mask(:,:,i)) < 0
        z = [z;i];
    end
end
for i = 1:(size(z,1)-1)
    if z(i+1)-z(i) > 1
        zBlankMin = z(i);
        zBlankMax = z(i+1);
        break;
    end
end

%                  EDIT HERE FOR DIFFERENT VOLUME SLICES
xmin = 1;  xmax = size(mask,2);
ymin = 1;  ymax = size(mask,1);
zmin = 20; zmax = 80;
% zmin = 1;  zmax = size(mask,3); % contain all data

% Make Sure Blank Z-Planes are not included
if zmin < zBlankMin
    zmin = zBlankMin;
end
if zmax > zBlankMax
    zmax = zBlankMax;
end
```

```
% Create Color Map Data for both Patches
mapBone = colormap('bone');
mapJet  = colormap('jet');
close(gcf);
```

Trim Data and Create Patches

```
tMax = size(Vh,4);
F(tMax-1) = struct('cdata',[],'colormap',[]);
for time = 2:tMax


    if time == 2
        heart   = mask(ymin:ymax,xmin:xmax,zmin:zmax);
        flat    = isocaps(heart,5);
        surface = isosurface(heart, 5);
    end
    strain  = Vh(ymin:ymax,xmin:xmax,zmin:zmax,time);
    strain(isnan(strain)) = 0;
```

Gather and Turn Facevertexcdata into RGB

```
    % Gather Color Data for Surface from heart
    vSurface = round(surface.vertices);
    cSurface = zeros(size(vSurface,1),1);
    rows = zeros(size(vSurface,1),1); % keep track of rows where this doesn't work
    for i = 1:size(vSurface,1)
        s = strain(vSurface(i,2)-1:vSurface(i,2)+1,vSurface(i,1)-1:vSurface(i,1)+1,vSurf
        if s == zeros(3)
            s = strain(vSurface(i,2)-2:vSurface(i,2)+2,vSurface(i,1)-2:vSurface(i,1)+2,v
            if s == zeros(5)
                s = 0;
                rows(i) = 1;
            else
                s(s == 0) = [];
            end
        else
            s(s == 0) = [];
        end
        cSurface(i) = mean(s(:));
    end

    % Define Color Index for both Patches
    lengthColormaps = size(mapBone,1);
    if time == 2
        minFlatVal = min(flat.facevertexcdata);
        if minFlatVal <= 0
            flat.facevertexcdata = flat.facevertexcdata + (-2 * minFlatVal);
        end

        ciFlat = ceil(lengthColormaps * flat.facevertexcdata / max(flat.facevertexcdata)
    end

    minSurfaceVal = min(cSurface);
    if minSurfaceVal < 0
        cSurface = cSurface + (-2 * minSurfaceVal);
    end

    ciSurface = ceil(lengthColormaps * cSurface / max(cSurface));
    ciSurface(isnan(ciSurface)) = 1;

    % Turn Single Color Data to RGB
    cFlatBone   = mapBone(ciFlat,:);
    cSurfaceJet = mapJet(ciSurface,:);
```

Define Patch Settings

```
if time == 2
    flat.facevertexcdata    = cFlatBone;
    flat.edgecolor          = 'none';
    flat.facecolor          = 'interp';
    flat.facelighting       = 'gouraud';
    flat.ambientstrength    = 0.6;

    surface.facevertexcdata = cSurfaceJet;
    surface.edgecolor       = 'none';
    surface.facecolor       = 'interp';
    surface.facelighting    = 'gouraud';
    surface.specularcolorreflectance = 0;
    surface.specularexponent = 30;
end
```

Draw Patches

```
if time == 2
    pause on
    fig = figure(1);
    fig.Position = [200 74 860 599];

    subplot(3,4,1)
    p2.pSurface  = patch(surface);
    view([0 0])
    lightangle(15,30);
    axis tight
    daspect('auto')
    axis off
    p2.Figure = gcf;
```

```
whitebg(p2.Figure);
p2.Figure.Color = 'k';

subplot(3,4,5)
p3.pSurface  = patch(surface);
view([120 0])
lightangle(135, 30);
axis tight
daspect('auto')
axis off

subplot(3,4,9)
p4.pSurface  = patch(surface);
view([240 0])
lightangle(255, 30);
axis tight
daspect('auto')
axis off

subplot(3,4,[2 3 4 6 7 8 10 11 12])
p1.pSurface  = patch(surface);
if ~isempty(flat.vertices)
    p1.pFlat = patch(flat);
end
view(3);
lightangle(-37.5,30);
axis tight
daspect('auto')
axis off
p1.Title = title('Time = 2');
p1.Colorbar = colorbar;
p1.Colorbar.Parent.Colormap = colormap(jet);
```

```
        F(time-1) = getframe(gcf);

    else
        %Draw New strains
        p1.Title = title(['Time = ' num2str(time)]);
        p1.pSurface.FaceVertexCData = cSurfaceJet;
        p2.pSurface.FaceVertexCData = cSurfaceJet;
        p3.pSurface.FaceVertexCData = cSurfaceJet;
        p4.pSurface.FaceVertexCData = cSurfaceJet;
        p1.Colorbar = colorbar;
        p1.Colorbar.Parent.Colormap = colormap(jet);
        F(time-1) = getframe(gcf);

    end

end
```

Write Video File and Save it to Local Folder

```
d = dir('*Movie4DAttempt*.avi');
numAttempts = size(d,1);
fileName = ['Movie4DAttempt' num2str(numAttempts + 1) '.avi'];
writerObj = VideoWriter(fileName);
writerObj.FrameRate = 5;
open(writerObj);
writeVideo(writerObj,F);
close(writerObj);
pause off


end
```

References

[1] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: a unifying framework. *International Journal of Computer Vision*, 56:221–255, Feb 2004.

[2] John J. Boyle, Maiko Kume, Matthew S. Wyczalkowski, Larry A. Taber, Robert B. Pless, Younan Xia, Guy M. Genin, and Stavros Thomopoulos. Simple and accurate methods for quantifying deformation, disruption, and development in biological tissues. *J. R. Soc Interface*, 11, 2014. http://dx.doi.org/10.1098/rsif.2014.0685.

[3] Jonathan Saul Cain, James P. Evans, and Craig B. Forster. Fault zone architecture and permeability structure. *Geology*, 24:1025–1028, Nov 1996. doi:10.1130/0091-7613(1996)024¡1025:FZAAPS¿2.3.CO;2.

[4] F. W. Damen, Y. T. Delorme, S. H. Frankel, P. P. Vlachos, and C. J. Goergen. 4-dimensional ultrasound imaging of left-ventricular dynamics. *7th World Congress of Biomechanics*, Jul 2014.

[5] Christian Franck, Stacey A. Maskarinec, David A Tirell, and Guruswami Ravichandran. Three-dimensional traction force microscopy: a new tool for quantifying cell-matrix interactions. *PLoS ONE*, 6, Mar 2011. doi:10.1371/journal.pone.0017833.

[6] Myers KM, Courdrillier B, Boyce BL, and Nduyen TD. The inflation response of the psterior bovine sclera. *Acta Biomater*, 6:4327–4335, 2010. doi:10.1016/j.actbio.2010.06.007.

[7] Phatak NS, Sun Q, Kim S-E, Parker DL, Sanders RK, Veress Al, Ellis BJ, and Weiss JA. Noninvasive determination of ligament strain with deformable image registration. *Annals of Biomendical Engineering*, 35:1175–1187, Jul 2007. doi:10.1007/s10439-007-9287-9.

[8] Michael A. Sutton. Computer vision-based noncontacting deformatino measurements in mechanics: a generational transformation. *Applied Mechanics Reviews*, 65, Sep 2013. doi:10.1115/1.4024984.

[9] Matthew A. Wyczalkowski, Victor D. Varner, and Larry A. Taber. Computational and experimental study of the mechanics of embryonic wound healing. *Journal of the Mechanical Behavior of Biomedical Materials*, 28:125–146, Jul 2013. (doi:10.1016/j.jmbbm.2013.07.018).

[10] S. Yoneyama and A. Kitagawa. Bridge deflection measurement using digital image correlation. *Experimental Techniques*, 31:34–40, Jan 2007. doi:10.111/j.1747-1567.2007.00132x.

Vita

Kenna Middleton

Degrees	B.S. Mechanical Engineering, August 2015 B.S. Mathematics/Physics, December 2015 M.S. Mechanical Engineering, December 2015

December 2015

Strain Mapping of a Beating Heart, Middleton, M.S. 2015