



CZECH TECHNICAL UNIVERSITY IN PRAGUE

**Faculty of Biomedical Engineering
Department of Biomedical Technology**

**Evaluation of MR-tagged Images with Respect to the Viability of the
Heart Based on the Strain Analysis**

Master Thesis

Study Programme: Biomedical and Clinical Technology

Field of Study: Biomedical Engineering

Supervisor: doc. Ing. Jiří Hozman, Ph.D.

John T. LaMaster

Kladno, August 2016

Declaration:

I hereby declare that I have completed this thesis with the topic “Evaluation of Dyssynchrony based on MRI tagging“ independently and that I have included a full list of used references. I have no objection to the usage of this work in compliance with the act §60 Zákon č.121/2000 Sb. (Copyright Law).

In Kladno, 19 August 2016



John LaMaster

I dedicate this work to my grandparents, Jean and Jerry Clausius. Growing up, they were the ones who emphasized the importance of getting my education. They have always been there for me, supported me, and pushed me to achieve my goals in life. This would not have been possible without their dedication, their love, and their support. Thank you so much for everything you have done for me. I love you both very much.

ABSTRACT

Title: Evaluation of the MR-tagged images with respect to the viability of the heart based on the strain analysis

Assessing myocardial strain is a particularly challenging task. Of the various methods capable of evaluating myocardial strain, they all have their own advantages and disadvantages. The obstacles faced are still so great that no single method has emerged as an accepted gold standard. MRI tagging is a much easier method to implement and obtain detailed results. However, the post-processing of the data is complex and time-consuming preventing it from being clinically viable. Contrarily, research facilities do not have the same constraints and are actively making use of the technique.

In this pilot study, an algorithm was developed to analyze the strain of the left ventricular myocardium. To overcome the small deformation limitation of HARP, a Gabor filter bank was implemented. This allows the algorithm to find the optimal phase of each pixel and generate a very accurate Angle image. The process of finding the optimal phase for each pixel identifies the optimal filter as well. The parameters used to create the filters in the filter bank were stored in a lookup table. Once the optimal filter is identified, the optimal parameters can then be used to estimate the strain based on the phase. This is a non-tracking strain estimator. This algorithm has a 95% probability of successfully differentiating between physiological data and pathological data. The p-values generated from the Wilcoxon ranksum test indicate enormous certainty in the results.

Keywords:

Cardiac resynchronization therapy, heart failure, dyssynchrony, magnetic resonance imaging, magnetic resonance tagging, HARP, Gabor filtration, strain analysis, non-tracking strain estimation

ABSTRAKT

Název diplomové práce: Vyhodnocení MR snímků z hlediska životaschopnosti srdce na základě analýzy namáhání

Posuzování mechanického namáhání srdečního svalu je obecně velmi náročná a komplexní úloha. Všechny metody, které jsou schopné hodnotit toto namáhání, jsou charakteristické svými výhodami a nevýhodami a neexistuje žádná univerzální. Obecně platí, že existuje mnoho problémů a překážek, které brání tomu, aby existoval jediný tzv. zlatý standard pro toto hodnocení. Metoda založená na prostorové modulaci přídavného magnetického pole (MRI tagging) je relativně dostupný způsob, jak realizovat a získat podrobné výsledky. Nicméně, následné zpracování, analýza a vyhodnocení obrazových dat je velmi složitý a časově náročný proces, který může být komplikací z hlediska používání metody pro standardní klinické využití. Naproti tomu využití ve výzkumu je u této metody velmi značné a v posledních několika letech došlo i k několika zdokonalením.

V rámci této pilotní studie byl vyvinut algoritmus s cílem analyzovat mechanické namáhání levé komory srdce. Za účelem kompenzace nevýhod algoritmu HARP z hlediska hodnocení malých deformací, byla realizována Gaborova filtrace. Ta spočívala v návrhu a realizaci banky filtrů pro jednotlivé směry. To umožňuje algoritmu najít optimální fázi každého pixelu a generovat velmi přesný obraz úhlů (angle image). Proces hledání optimální fáze pro každý pixel určuje právě optimální filtr z uvedené banky. Tyto parametry k vytvoření filtru v rámci banky filtrů byly uloženy v tzv. vyhledávací tabulce (LUT). V případě identifikace optimální filtrace jsou parametry z LUT použity pro odhad namáhání na základě související fáze. Toto je pak tzv. odhad, který není založený na sledování superponované mřížky. Tento algoritmus má 95% pravděpodobnost úspěšného odlišení fyziologických dat a patologických dat. P-hodnoty generované v rámci Wilcoxonova statistického testu naznačují relativně vysokou jistotu v obdržení výsledcích.

Klíčová slova:

Srdeční resynchronizační terapie, srdeční selhání, dyssynchronie, zobrazování magnetickou rezonancí, zobrazování magnetickou rezonancí s prostorovou modulací magnetického pole, HARP, Gaborova filtrace, analýza namáhání, analýza namáhání bez prostorové modulace magnetického pole

Department of Biomedical Technology

Academic year: 2015/2016

D i p l o m a t h e s i s a s s i g n m e n t

(Master project thesis assignment)

Student: **John LaMaster**
Study branch: Biomedical Engineering (CEMACUBE)
Title: **Evaluation of the MR-tagged images with respect to the viability of the heart based on the strain analysis**
Title in Czech: Vyhodnocení MR snímků z hlediska životaschopnosti srdce na základě analýzy namáhání

Instructions for processing:

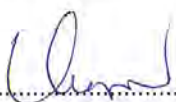
Compute and analyse the radial and circumferential strain within the relevant volume segments of the left ventricle obtained from MR-tagged images in DICOM format with respect to the viability of the heart within the experimental pilot study. Extract the motion vectors within the region of interest for the positional changes of individual points along to the superimposed grid within the segments of the heart, i.e. the apex, the middle, and the base along to the longitudinal axis based on the AHA standard.

References:

- [1] Amir A. Amini, Jerry L. Prince (eds.), Measurement of Cardiac Deformations from MRI: Physical and Mathematical Models, Kluwer Academic Publishers, 2001, 320 s., ISBN 1-4020-0222-X
- [2] El-Sayed H Ibrahim, Myocardial tagging by Cardiovascular Magnetic, Journal of Cardiovascular Magnetic Resonance, ročník 13, číslo 36, 2011, Červenec, 1-40 s., <http://www.jcmr-online.com/content/13/1/36>, doi:10.1186/1532-429X-13-36

Validity of assignment until date: 20.08.2017

Supervisor of diploma thesis: doc. Ing. Jiří Hozman, Ph.D.


.....
Head of Department


.....
Dean

In Kladno, 12.04.2016

ACKNOWLEDGMENTS

I would like to thank doc. Ing. Jiří Hozman, Ph.D. of Faculty of Biomedical Engineering for taking the time to be supervise this master thesis and for his willingness to assist me when I needed it. I would like to thank Ing. Vladimír Petráková, Ph.D. for her sincere motivation and eagerness to help me succeed. Although she was not officially a part of this work, her kind words helped give me the motivation and support I needed to finish. I would also like to thank Mgr. Radim Krupička, Ph.D. and Doc.Mgr. Petr Páta PhD for lending me their time and expertise to help me work through problems when they arose. And finally, I need to thank my friends and family who have continuously supported financially, emotionally, and spiritually. Your motivation and support is what helped me get to where I am today. Thank you all very much.

Contents

Declaration	i
Dedication	ii
Abstract	iii
Abstrakt	iv
Assignment of the Diploma Thesis Topic	iv
Acknowledgments	vi
Contents	ix
List of Figures	x
List of Tables	xi
List of Abbreviations	xii
1 Introduction	2
2 State of the Art	4
2.1 MR Tagging	4
2.2 Motion Analysis	4
2.2.1 Gabor Filters	6
2.3 Strain Analysis	6
3 Research Objectives	8
4 Methods	9
4.1 MRI Acquisition	9
4.2 Segmentation	9
4.3 Motion Analysis	12
4.3.1 Gabor Filter	12
4.4 Strain Quantification	17
4.5 AHA Left Ventricular Segmentation Model	19
4.6 Viability Assessment	21

4.7	Implementation Notes	22
4.7.1	Remarks	22
4.7.2	Program Descriptions	23
	Segment	23
	GaborFilterBank	25
	GaborStrain ** Update Optimize and Adjust (if needed)!!!	25
	combineStack	26
5	Results	27
5.1	Synthetic Data	27
5.2	Physiological Data	27
5.3	Quantitative Comparison	29
6	Discussion	31
6.1	Limitations	31
7	Conclusion	33
8	Future Works	34
8.1	Viability	34
8.2	Improvements	35
8.3	Future Use	38
	References	38
	Appendices	
A	Matlab Code	43
A.1	Segment.m	43
A.1.1	imageCheck.m	44
A.1.2	chooseLabelDialog.m	45
A.1.3	predefined_segmentation_variables.m	45
	FourierPeaks.m	46
A.1.4	ROI.m	47
A.1.5	Segmentation.m	48
	contours.m	49
	Centroid.m	50
	testPoints.m	51
	sortContour.m	52
	resample.m	53
	SegmentFinder.m	53
A.2	GaborFilterBank.m	54
	GaborKernel.m	56
A.3	GaborStrain.m	57

A.3.1	Optimize.m	60
	adjust.m	61
A.4	combineStack.m	62
B	Synthetic Data	64

List of Figures

Figure 1.1	CRT Pacemaker Diagram	2
Figure 2.1	Tag refinement using Fourier expansion coefficients	6
Figure 4.1	Segmentation contours	11
Figure 4.2	Resampled contours and points of interest	11
Figure 4.3	Points-of-Interest separated by AHA segment	12
Figure 4.4	SPAMM-tagged image and FFT	14
Figure 4.5	Gabor filter bank range	15
Figure 4.6	Kernels for the first orientation angle - 2D plot	15
Figure 4.7	Kernels for the first orientation angle - 3D plot	16
Figure 4.8	Kernels for the second orientation angle - 2D plot	16
Figure 4.9	Kernels for the second orientation angle - 3D plot	16
Figure 4.10	Strain Estimation Geometry	18
Figure 4.11	Myocardial strain diagram	19
Figure 4.12	Strain transformation	19
Figure 4.13	Polar Plot of AHA LV Segmentation Model	19
Figure 4.14	AHA LV Segmentation Model	21
Figure 4.15	Filtering effects on Angle images	22
Figure 5.1	Synthetic data Segment.m analysis	28
Figure 5.2	Physiological data Segment.m analysis	28
Figure 8.1	Harmonic peak shape variations	37
Figure 8.2	Points-of-Interest identified by interparc.m	38
Figure B.1	Synthetic data image 01	64
Figure B.2	Synthetic data image 02	64
Figure B.3	Synthetic data image 03	65
Figure B.4	Synthetic data image 04	65
Figure B.5	Synthetic data image 05	66
Figure B.6	Synthetic data image 06	66
Figure B.7	Synthetic data image 07	67
Figure B.8	Synthetic data image 08	67

List of Tables

Table 4.1	Physiological data acquisition trigger times	10
Table 4.2	Pathological data acquisition trigger times	10
Table 5.1	Wilcoxon ranksum hypothesis decisions	29
Table 5.2	Wilcoxon ranksum p-values	30

LIST OF ABBREVIATIONS

- AHA** American Heart Association
- CRT** Cardiac Resynchronization Therapy
- DICOM** Digital Imaging and Communications in Medicine
- ECG** Electrocardiogram
- FFT** Fast Fourier Transform - Matlab function
- FFT2** 2D Fast Fourier Transform - Matlab Function
- FT** Fourier Transform
- HARP** Harmonic Phase Magnetic Resonance
- IFFT** Inverse Fast Fourier Transform - Matlab function
- IFFT2** 2D Inverse Fast Fourier Transform - Matlab Function
- LA** Long-Axis
- LV** Left Ventricle; Left Ventricular
- MR** Magnetic Resonance
- MRI** Magnetic Resonance Imaging
- ROI** Region-of-Interest
- SA** Short-Axis
- SPAMM** Spatial Modulation of Magnetization
- cMR** Cardiac Magnetic Resonance

Collaboration:



**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA BIOMEDICÍNSKÉHO INŽENÝRSTVÍ**



FN MOTOL

FAKULTNÍ NEMOCNICE V MOTOLE

1. INTRODUCTION

Cardiac resynchronization therapy (CRT), also known as biventricular pacing, is the standard treatment method for symptomatic patients suffering from medically refractory heart failure (HF), electrical cardiac dyssynchrony, and depressed left ventricular (LV) function, also referred to as mechanical dyssynchrony and intraventricular dyssynchrony.^[2–4] CRT involves the implantation of a special type of pacemaker. A pacemaker is designed to reset the overall electrical activity of the heart when either irregular activity or no activity is detected. The special kind used in CRT, however, is designed to coordinate the interventricular electrical activity between the right and left ventricles. As can be seen in Fig. 1.1, published in [1], the first two wires are connected to the right atrium and ventricle respectively. Then, the third wire is connected to the left ventricle.

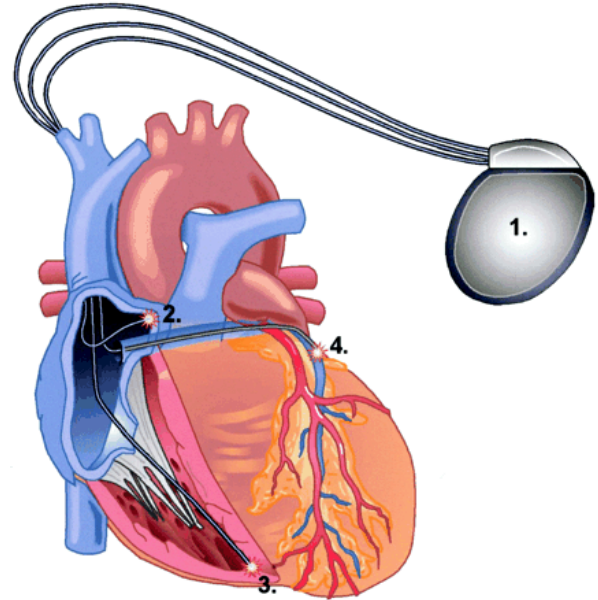


Figure 1.1: CRT: 1. Pacemaker generator; 2. right atrial pacer wire; 3. right ventricular pacer wire; and 4. coronary sinus (“left ventricular”) pacer wire.^[1]

Electrical cardiac dyssynchrony is detected using an electrocardiogram (ECG). This dyssynchrony is used as the primary criterion in selecting patients for CRT. The characteristic evaluated for CRT qualification is the length of the QRS complex. The standard threshold is a QRS duration of 120ms or more.^[4–6] However, this criteria is not particularly effective in predicting patient response to CRT. One study reported 60% to 80% of patients exhibited a favorable response,^[7] while another reported only 20% to 30%.^[8] This has led to a push for finding novel indices and strategies for improving the selection criteria that can more reliably predict the long-term outcome of CRT. One such index is mechanical dyssynchrony, specifically of the left ventricle (LV), which is defined as the delay in the myocardial contraction between the septum and lateral walls,^[8] has shown great promise in more accurately predicting CRT patient response.^[7]

However, further work needs to be completed before mechanical dyssynchrony could be used as a clinical index. Traditionally, echocardiographic measurements using speckle tracking (STE) imaging are used to assess mechanical dyssynchrony.^[8] Most studies that have investigated mechanical dyssynchrony evaluate the left ventricular ejection fraction (LVEF) as one of their selection criteria. LV function is generally considered depressed when LVEF drops to 35% or lower.^[4] However, the optimal modality for assessing intraventric-

ular dyssynchrony has yet to be identified experimentally. Similarly, as Sung and Foster^[4] point out, the benefit of considering the amount of mechanical dyssynchrony, inter- and in-ventricular, has not been investigated and there is currently no threshold used to define the presence of meaningful mechanical dyssynchrony. Studies would have to be carefully designed and carried out to determine not only the sensitivity of using mechanical dyssynchrony to select a patient for CRT, but also the specificity of this criterion in being able to confidently determine that a patient would not benefit from CRT.

A problem that consistently arises when trying to compare results of these studies is that there is no universal definition of CRT response. Some studies use hard end points such as hospitalization, mortality, and morbidity, where as other studies use short-term, surrogate end points such as a change in the LV end-systolic volume (LVESV) or LVEF, LV reverse remodeling, and clinical or symptomatic improvements.^[4] Identifying and establishing an assessment index for evaluating the success of CRT is a crucial step in identifying the optimal index for predicting patient response along with the ideal modality for assessing those characteristics.

Assessing mechanical dyssynchrony is also very problematic. In pediatric cases, it is relatively easy to assess mechanical dyssynchrony using STE because not only is the heart small enough that the entire myocardium can be viewed within one frame, but children also have smaller ribs, which impede the viewing range much less than in adults, and tend to have less fat and other artifact-inducing tissue components for which there is no simple method of compensation. On the contrary, it is much more challenging to use STE for LV strain analysis in adults who tend to have many more artifact-inducing components that simply cannot be compensated. Although STE is a valid method of strain analysis, there are still a sufficient number of challenges that have prevented this modality from being accepted as the gold standard for assessing mechanical dyssynchrony.

Using tagged cardiac magnetic resonance imaging (tMRI) is another method of assessing LV myocardial strain and mechanical dyssynchrony. This modality has many advantages over STE. The myocardium easily fits within the view frame of the MRI, which is known for yielding very high resolution images. Tagging significantly reduces this high resolution, but even with a 1.5T machine the results are still very useful. This method is not clinically viable yet due to the immense and time-consuming post-processing that is required to obtain useful data. The largest hurdles for translating this modality into clinical use are myocardial segmentation, finding an algorithm robust enough for clinical use that can extract the motion information from the tags, and then developing an acceptable algorithm for strain analysis. Validating this method will also be difficult because there is no gold standard for assessing myocardial strain and mechanical dyssynchrony.

MRI has been found to be one of those indices that is useful in selecting candidates for CRT^[8] and potentially for assessing mechanical dyssynchrony; however, it is a very expensive imaging and diagnostic modality. In order to justify using MRI, it must be proven that there is an equivalent diagnostic significance. Although there are many things that need to be accomplished before mechanical dyssynchrony can be translated into a clinically applicable index, this pilot study will focus on the first step: algorithm development.

2. STATE OF THE ART

2.1 MR Tagging

Magnetic resonance tagging was initially developed by Zerhouni *et al.*, in 1988.^[9,10] This initial method could simultaneously generate a maximum of three parallel tagging planes at a time. The following year, in 1989, Axel published an improved tagging methodology based on the idea of spatially modulating the magnetization.^[10,11] This novel method, spatial modulation of magnetization, is most commonly referred to as SPAMM. In just one year some of the major disadvantages were improved upon. The specific absorption rate (SAR) significantly decreased, the spatial resolution increased, and the temporal resolution dramatically increased. Since the development of SPAMM tagging, there have been several other tagging methodologies developed.

Since Zerhouni's original publication, there have been many and significant improvements in the field of MR tagging and the algorithms used today are much more sophisticated. Currently there are several methods available each with their advantages and disadvantages. There are also different versions of these methods that seek to improve upon characteristics such as spatial and temporal resolution, signal-to-noise ratio (SNR), contrast-to-noise ratio (CNR), and three-dimensional capabilities.

2.2 Motion Analysis

Researchers have developed a myriad of techniques and approaches to analyze motion in images. These techniques are all methods of optical flow analysis which seeks to assess motion in a sequence of images. Optical flow methods can be broken down into several categories, of which two will be further elaborated: 1) Tag- and feature-based tracking methods and 2) Phase-analysis methods.

Tag- and feature-based tracking methods take a very direct approach to motion analysis and tend to follow a more traditional, or historical, thinking regarding image processing. Examples of these methods include: active contours, template matching, B-snake grids, and 3D variants among others. Most of these methods are dependent upon the grayscale pixel values of the images. Therefore, they are directly dependent upon the image quality. As the MR scanners are continuously improved using higher magnetic field strengths and developing greater sensitivity, both of which enhance the spatial resolution of the images. So even though tagged MRI has a much lower spatial resolution and image quality, this is not very problematic at the moment and should only get better with time. The main hurdle at the moment is tag fading. Because these techniques rely so heavily on the intensity values of the

tags, it becomes more difficult to extract accurate measurements at later stages of the cardiac cycle.

Phase-based optical flow analyses, the most well-known of which is HARMonic Phase (HARP), attempt to model the tags and their deformations using information extracted from the Fourier space. Osman and Prince developed this novel medical image analysis technique (HARP) in 1999.^[12] This technique has been validated and used in countless studies including: 12–25. SPAMM tagged images are created by modulating the tissue in the field of view (FOV) with k-space spectral peaks at multiples of the tagging frequency. HARP exploits this fact and creates a spatial band-pass filter, based on the central frequency of the first fundamental harmonic peak for each tag line orientation, that isolates this spectral peak containing the tag information. Modulating the filter size can assist in fine tuning the filter's effectiveness. A Fourier transform (FT) of this filtered k-space returns a complex image: the real part being the magnitude, which is a low-pass filtered mask of the myocardium, and the imaginary part being the displacement-encoded phase containing the tag line intensity pattern for the respective harmonic peak. Multiplying these images together produces the HARP image. At this point, myocardial strain estimation is just a matter of tracking the phase of points-of-interest, PoI(s). Although this is a highly robust method, it has a serious disadvantage. Due to being so heavily reliant on a single frequency for each tag orientation, this algorithm is incapable of coping with large deformations and is prone to losing tags.^[22,26]

Many advancements have been made with HARP since its initial publication. Ryf *et al.*^[27] found that by combining the phase of the positive and negative harmonic peaks of CSPAMM images before HARP implementation, what they call "nonidealities", such as B_0 inhomogeneities, and other phase errors are accounted for and corrected. This method also increases the signal-to-noise ratio (SNR). Mehmet Bilgen developed a method called HARP interference.^[28] Bilgen found that more information can be extracted from tMRI by modifying the power and phase of the image. His method follows a similar logic as structured-illumination microscopy by which a structured illumination pattern is projected on the sample and the results of the analysis of the aliasing produces super-resolution. By creating an image with an arbitrary but known phase and then adding the synthetic and acquired phase-wrapped images together, a phase interference image is created. Then the Delaunay algorithm can then be used to automatically locate the tag intersections.

Osman and Prince^[18] developed a method of refining the tag pattern in tMRI. The tag profile coefficients, determined by the Fourier series expansion, can be used to model synthetic tag profiles. By replacing the complex harmonics with real sinusoids, the required number of summations and the computational cost decrease. As can be observed in Fig. 2.1, the more coefficients used, the more narrow the resulting tag lines. These coefficients are then used to generate additional harmonics in the original image. This is done by padding the Fourier space with zeros followed by adding the additional harmonics which results in much crisper, more defined tag profiles and potentially enhancing the results of motion analysis techniques further on.

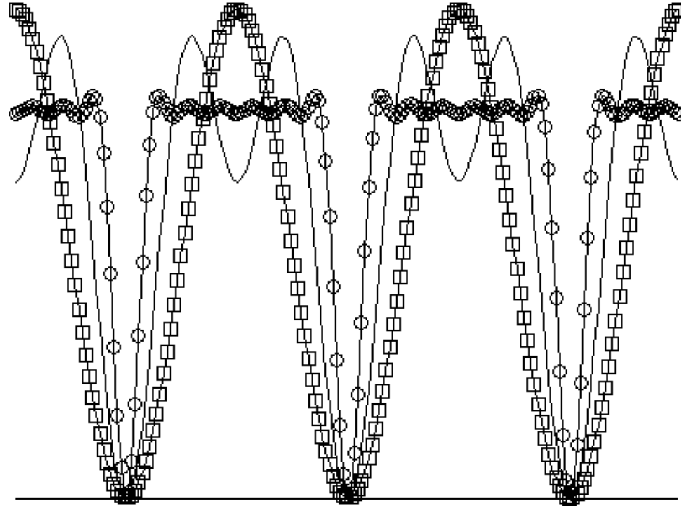


Figure 2.1: Expected tag line profiles given different numbers of Fourier expansion coefficients. The square markers indicate the profile given two coefficients. The solid line represents four coefficients. The circular markers indicate represent the profile given seven coefficients.

2.2.1 Gabor Filters

Smal *et al.* published a quantitative review of four different methods of motion analysis: optical flow methods, HARP, B-snake grids, and non-rigid registration techniques. They concluded that the non-rigid registration techniques were the most effective method. However, there is another method that compliments HARP and appears very promising: Gabor filter bank.

HARP's major limitation is that it assumes a single central frequency is valid throughout the entire cardiac cycle. This restricts the algorithm to only being able to detect small deformations. In order to overcome this restriction, a Gabor filter bank can be implemented.^[22,26,29,30]

Gabor filters, initially developed by John Daugman in 1985^[31], are band-pass filters whose shape can be refined for various situations. The filter envelopes can be anisotropic and have variable orientations meaning that the filters can change for different central frequencies. This implementation creating very precise Angle images by optimizing the filter for each pixel. An Angle image is defined as the phase of a single spectral peak in the Fourier domain of a MR-tagged image.^[32] By creating a bank of these filters and varying the central frequency, a greater range of tag deformation can be detected.

2.3 Strain Analysis

Similar to motion analysis, most strain analysis methods rely on the gradient of the displacement of identifiable landmarks. This approach encounters the same problems as the tag- and feature-based tracking methods used in motion analysis. As the tag lines fade, so

does the accuracy of the computed results.

As Barajas *et al.* explain in [26], it is far more robust to track the phase rather than features. The Gabor filter bank is used to create a much more precise Angle image. This is accomplished by identifying the filter which elicits the maximum magnitude response. From there, it is simple to find and collect the corresponding phases into a single image. Knowing the optimal Gabor filter for each pixel indicates the optimized parameters. By using the parameters optimized for each tag orientation, deformation geometry can be solved and the strain can be estimated in a non-tracking manner. Qian *et al.* developed this method in 2008.^[22]

There are still issues that must be addressed, even with this method. As Qian points out, the radial strain is consistently underestimated while the circumferential strain is overestimated. Potentially this could be taken into account and compensated via correction factors if necessary. Such correction factors would have to be based on synthetic data alone given that there is no standardized method of strain analysis.

3. RESEARCH OBJECTIVES

This is a pilot study to assess the viability of this algorithm with respect to myocardial strain analysis. The primary objective of this thesis is to develop the initial version of an algorithm that can quantify the left ventricular myocardial strain via SPAMM tagged cMR images. If proof-of-concept is achieved, then this will aid in the future steps of establishing a gold standard for evaluating cardiac mechanical dyssynchrony and then refining the candidate selection criteria for CRT.

The fundamental duties of this algorithm can be broken down into four general tasks. Each of these tasks have some recommendations and requirements to help guide the design and development of this work and are explained below.

Fundamental tasks:

- 1. Segmentation:** There are no specific requirements for the segmentation process of the endocardial and epicardial contours of the left ventricle. Manual segmentation is the simplest and most common method. Although this process may be semi-automated, but it is not required. The segmentation process should never rely solely on a fully automated segmentation algorithm because it is necessary for experts, healthcare providers and researchers, to have the opportunity to manually correct the contours. A manual segmentation approach is recommended in the beginning because the main focus should be on tasks 2, 3, and 4.
- 2. Tag deformation tracking:** The traditional method of motion analysis in tagged magnetic resonance sequences is the HARP algorithm. The implementation of a Gabor filter bank has proven to be a beneficial improvement upon the HARP algorithm. Therefore, with HARP as the fundamental concept, it is recommended to use a Gabor filter bank for analyzing the myocardial deformation.
- 3. Strain analysis:** The only requirements for strain analysis are that the radial and circumferential strains must be calculated and returned. No specific method or approach is recommended.
- 4. Viability Assessment:** Validation is the single most important step in most research. The strain analysis and segmentation results, if applicable, obtained from the developed algorithm should be compared to at least one other validated method. A basic statistical analysis should be used to show whether or not the algorithm is capable of differentiating between the control group and the clinical data. Given that this is a pilot study, an in-depth statistical analysis of the algorithm is not required.

4. METHODS

Should the List of Figures [names] list the program name that produced that result, or ***a brief description of the images***?

4.1 MRI Acquisition

The MR examinations were completed using a standard 1.5 T Siemens Magnetom Avanto (Siemens Healthcare Erlangen, Germany) with Syngo MR B17 software. The image parameters were: matrix 208 x 256, in-plane resolution 1.3 x 1.3 mm, slice thickness 6 mm, no inter slice gap, and a flip angle of 14°. The study used spatial modulation of magnetization (SPAMM) as the tagging sequence and the ECG-gating used the R-peak as the trigger.

The cine sequences analyzed in this study came from anonymized data sets and therefore the demographics such as age and sex are unknown. In each examination, three short-axis sequences were obtained corresponding the basal, mid, and apical regions along the long-axis of the heart. Sequences from each region were analyzed from at least five individuals from the two groups: healthy volunteers and clinical patients. The patients were selected according to diagnostic necessity by their physician.

The data sets used in the development of the algorithm contained 22 images. Initially six images were selected in order to accurately represent the entire sequence. Before testing the data sets it was discovered that even cine sequences from the same patient can have varying trigger times and between studies there were even greater variations. The trigger times are shown in Tables 4.1 and 4.2. The first trigger time in most sequences was either 26ms or 35ms. The first image in the sequence must be used because it has the least deformed tag lines. All of the subsequently chosen data sets were selected to match the trigger times of the initial data sets. As a note, even though some of the data sets included more than 22 images, the amount of time between triggers did not change accordingly. This means that some sequences followed the cardiac cycle for a longer period of time. Secondly, this additional information is not included because it is not present in all of the data sets.

4.2 Segmentation

A manual segmentation method was implemented in this study. However, the Segmentation.m program does more than gather user-defined contours. This program is broken down into 3 parts: 1) Finding the contours; 2) Establishing the points-of-interest (PoIs) to be tested; and 3) Separating these PoIs into groups based on the AHA 17-segment model^[33].

The contour detection algorithm follows standard methods for manually segmenting

Table 4.1: Trigger times (ms) for all of the image sequences used in the control group.

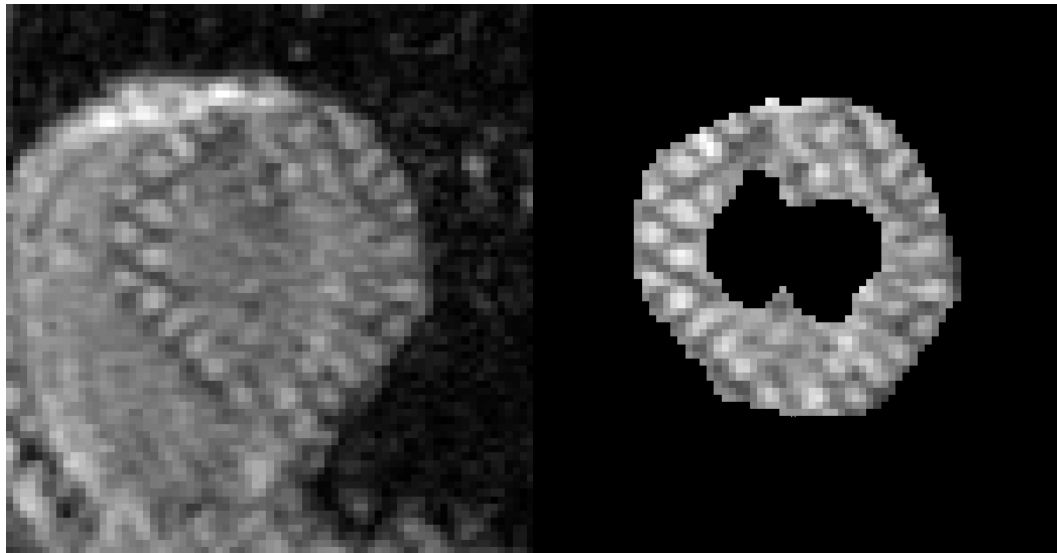
Data Sets	1	2	3	4	5	6
1	35.80	107.40	214.80	322.20	429.60	537.00
2	26.85	107.40	214.80	322.20	429.60	537.00
3	35.80	107.40	214.80	322.20	429.60	537.00
4	35.80	107.40	214.80	322.20	429.60	537.00
5	26.85	107.40	214.80	322.20	429.60	537.00
6	26.85	107.40	214.80	322.20	429.60	537.00
7	26.85	107.40	214.80	322.20	429.60	537.00
8	35.80	107.40	214.80	322.20	429.60	537.00
9	35.80	107.40	214.80	322.20	429.60	537.00
10	26.85	107.40	214.80	322.20	429.60	537.00

Table 4.2: Trigger times (ms) for all of the image sequences used in the pathological data set.

Data Sets	1	2	3	4	5	6
1	34.68	104.04	208.08	312.12	416.16	520.20
2	26.01	104.04	208.08	312.12	416.16	520.20
3	26.01	104.04	208.08	312.12	416.16	520.20
4	26.01	104.04	208.08	312.12	416.16	520.20
5	26.01	104.04	208.08	312.12	416.16	520.20
6	34.68	104.04	208.08	312.12	416.16	520.20
7	26.01	104.04	208.08	312.12	416.16	520.20
8	26.01	104.04	208.08	312.12	416.16	520.20
9	26.01	104.04	208.08	312.12	416.16	520.20
10	26.01	104.04	208.08	312.12	416.16	520.20

images. Manually selected contour points are gathered for both the endocardial and epicardial contours. Those coordinates are returned along with their respective masks and the final contour mask. The plotted contours and the myocardium isolated via the myocardial contour mask are displayed below in Fig. 4.1.

Images****



(a)

(b)

Figure 4.1: a) Original input image. b) The original image masked with the binary mask created from the manually selected contour points.

The next step is to sample the myocardium in a standardized manner. The points that form the boundaries of the endocardium and epicardium are identified and then sorted. In order to maintain the integrity of the contour and sort the points into a usable format, the sorting process has two main sections. The first part calculates the polar angle θ about the centroid of the myocardial mask. The angles are adjusted so that they fall within the range $[0, 2\pi]$. θ is then sorted in ascending order and an index is created that sorts the corresponding contour points into the same order. An For ease of use later on, the points are then rearranged so that the list of points begin with the first point in the first segment of the AHA 17-segment model.

The following section resamples the contours into a set number of points according to specified criteria. The algorithm considers the slice level (basal, mid-cavity, or apical) and guidelines established by the American Heart Association^[33] to determine into how many segments the myocardium should be divided. Based on a similar, freely available software, Segment^[34] by Medviso, each segment will have 10 columns of test points each containing 5 rows spanning between the two contours. These rows of test points are linearly interpolated between the endocardial contour point and the epicardial contour point. One significant difference between Segment and the approach implemented here is that this software only generates test points that fall within the myocardial mask for each echo time.

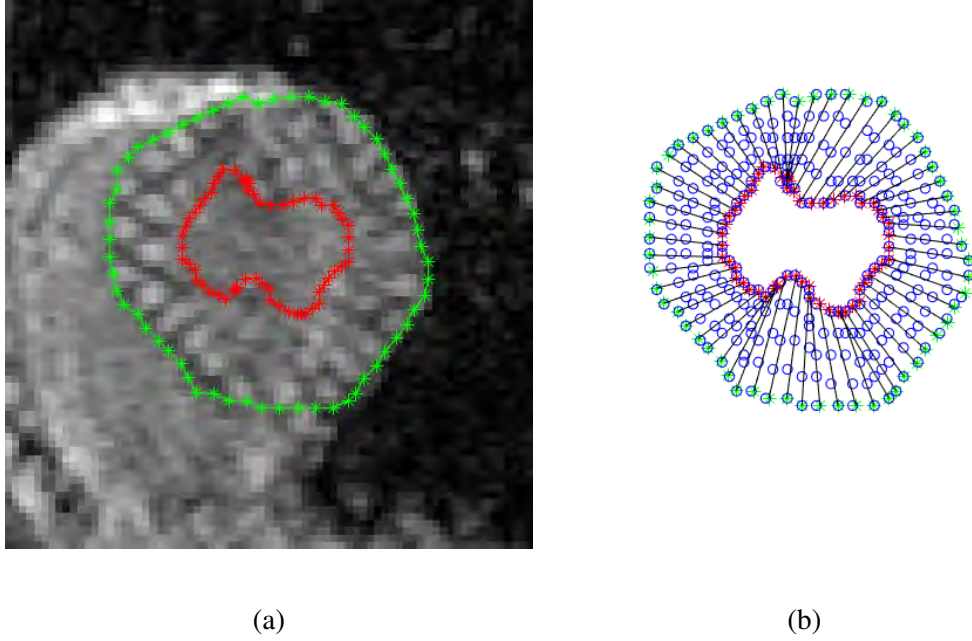


Figure 4.2: a) Manually selected contour points along with the resampled points. Those in red represent the endocardial contour and those in green represent the epicardial contour. b) The points-of-interest chosen for testing are represented by blue circles. The lines connect the endocardial and epicardial points that have been paired together.

Next, the centroid of the myocardial contour is found. Although this is not used further in the segmentation portion of this program, it will be used further on. The final step in the segmentation program is to divide the test points into the AHA segments indicated by the slice label.

4.3 Motion Analysis

4.3.1 Gabor Filter

Osman and Prince^[32] showed that SPAMM-tagged images can be expressed as the sum of several complex images:

$$\psi = \sum_{k=-K}^K \psi_k \quad (4.1)$$

where ψ_k is the complex image resulting from the inverse Fourier transform of one of the harmonic peaks and $2K + 1$ is the number of harmonic peaks. These Angle Images have a linear relation with the true motion of the points within the myocardium meaning that the phase of these points is actually an intrinsic property of the tissue which remains constant over time.^[26,35] Therefore, tracking the phases of these points is equivalent to tracking the

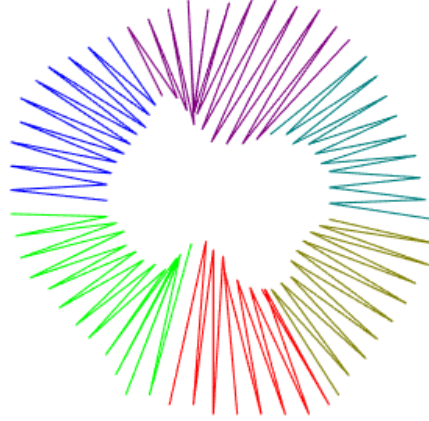


Figure 4.3: Each color represents a different segment. Starting from the top and moving counter-clockwise, the segments are as follows: anterior, anteroseptal, inferoseptal, inferior, inferolateral, and anterolateral.

points themselves. As stated before, robust methods in image processing do not rely solely on grayscale values. The reason this method of motion analysis is so robust is because it tracks the phase values through the cine sequence instead of the fading tag lines.

All methods used to obtain the Angles images have one common feature: at least one ψ_k must be isolated. The most common method to go about this is to use HARP; however, the main disadvantage with HARP is that due to its filter design, it is limited to small local deformations. To get around this limitation, Qian *et al.* proposed in [29] to use a Gabor filter bank which analyzes multiple frequencies to find the optimal filter.

A Gabor filter is a special band-pass filter with an envelope shaped by a Gaussian, g , modulated by a complex sinusoid, s . A 2D Gabor filter in the spatial domain is expressed as:

$$h(x, y) = g(x', y') \cdot s(x, y), \quad (4.2)$$

where the Gaussian is defined by:

$$g(x', y') = \frac{1}{2\pi\sigma_{x'}\sigma_{y'}} e^{-\frac{1}{2}\left[\left(\frac{x'}{\sigma_{x'}}\right)^2 + \left(\frac{y'}{\sigma_{y'}}\right)^2\right]} \quad (4.3)$$

and the complex sinusoid is defined as:

$$s(x, y) = \exp[-i2\pi(Ux + Vy)] \quad (4.4)$$

The Gaussian variables x' and y' are used to establish the rotation of the Gaussian. They are defined as:

$$x' = x \cos\theta + y \sin\theta, \quad \text{and} \quad y' = -x \sin\theta + y \cos\theta. \quad (4.5)$$

whereby x' and y' are the spatial coordinates that have been rotated by angle θ . The shape of the Gaussian envelope is controlled by λ which is defined below in Eqn. 4.6. If $\sigma_{x'}$ and $\sigma_{y'}$ are equal, then $\lambda = 1$ meaning that the envelope will be symmetrical. Depending on the application this may be desirable. However, it can be modified to create an asymmetrical envelope. Fig. 4.4b shows the two-dimensional Fourier transform of a SPAMM-tagged image with two tag orientations. There are harmonics in each direction corresponding to the tag line orientations. Some of these harmonic peaks appear due to having multiple intersecting, orthogonal tag line orientations. These additional peaks must be taken into account when designing the shape of the Gabor filter kernel. The aspect ratio λ , as defined in Eqn. 4.6, is the ratio regarding the length and width of the filters' envelope. HARP generally uses a symmetrical envelope with a radius of half the central frequency. In order to avoid accidentally including information from the second harmonic peak, an ellipsoid is generally chosen for the overall shape. In this study, as in Barajas *et al.*'s 2005 publication, an aspect ratio of $\lambda = 2$ was chosen.^[26]

$$\lambda = \frac{\sigma_{x'}}{\sigma_{y'}}, \quad \phi = \arctan\frac{V}{U} \quad (4.6)$$

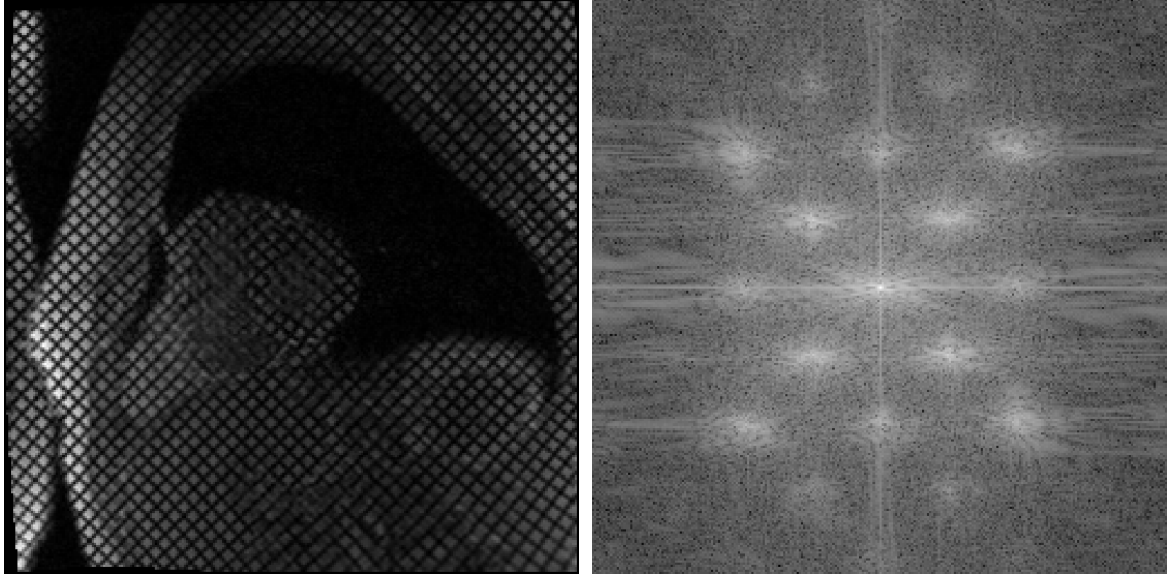
where $\sigma_{x'}$ and $\sigma_{y'}$ are the standard deviations that dictate the shape and the symmetry of the Gaussian envelope, ϕ is the orientation of the envelope, and U and V are the 2D frequencies of the complex sinusoid. The envelope was designed so that the orientations of the Gaussian and the sinusoid align, $\theta = \psi$.^[26]

The shape of the filter, the envelope, is very important in designing the band-pass filter. There are three variables- λ , σ_x , and σ_y -that are used to define the bandwidth of the filter. As stated above, this algorithm uses a λ of 2. σ_x and σ_y are then defined:

$$\sigma_x = 2\sigma_y = \frac{1}{\sqrt{U^2 + V^2}} \quad (4.7)$$

such that (U, V) are frequencies of ψ_1 of the input image. (U, V) are also the frequencies about which the Fourier transform $H(u, v)$ of $h(x, y)$ is centered which indicates that the Gabor filter is capable of achieving optimal resolutions in both the spatial and frequency domains. It should be noted that even though Barajas *et al.* defined the Gaussian and the complex sinusoid on the x - and y -axes, the same equations still hold true when the tag orientations are shifted because of ϕ . The envelope is still defined according to a horizontal and vertical geometry and then it is rotated about the center.

According to Daugman *et al.*^[31], 2D Gabor filters have eight degrees of freedom. The first two are the coordinates of the filter in the spatial domain, (x, y) . Next are the modulation coordinates (U, V) specifying the location of the filter in the frequency domain. These are the parameters that are modulated in order to create the filter bank. Then there is the phase



(a)

(b)

Figure 4.4: a) Original SPAMM-tagged image with two tag line orientations; b) The logarithmic spectrum of the Fourier transform of the SPAMM-tagged image

modulation component controlling the filter's mixture of symmetry-antisymmetry and the relative angle between the 2D Gaussian axes and the orientation of the modulating sinusoid. And finally there is σ_x and σ_y which define the width and length of the elliptical Gaussian envelope. Of these, the first four are the independent variables: x , y , U , and V . The previous equations can be combined and simplified, as shown in Eqn. 4.8, so that the Gabor filter can be defined in terms of two parameters: U and V .

$$h(x,y) = \frac{U^2 + V^2}{2\pi} e^{\left\{ -\left[\frac{3}{2} \frac{(x^2+y^2)(U^2+V^2)}{2} + j2\pi(Ux+Vy) \right] \right\}} \quad (4.8)$$

Gabor filters behave much the same way as HARP does in that they are both band-pass filters centered about the central frequency ψ_k . This fact has two big implications: 1) the phase response of the filtered images are also Angle Images and can be dealt with as such, and 2) applying a single Gabor filter has no benefit over tradition HARP because the filter is again assuming a single frequency for the entire image which causes the algorithm to be incapable of dealing with large local deformations. In order to overcome the second implication, a set of Gabor filters can be defined by a range of frequencies surrounding the central frequency.

[30] and [36] build upon this idea of of creating a bank of filters. Looking back to Eqn. 4.8, the Gabor filters can be defined using but two input variables, (U, V) . Therefore, defining a bank of filters can be done by modulating both U and V . In their work, they defined one frequency range for each tag orientation by varying the central frequency according to the

following equations:

$$\begin{aligned} U' &= \Re\{(U + i \cdot V) \cdot m \cdot \exp(i \cdot \Delta\phi + \omega)\}, \\ V' &= \Im\{(U + i \cdot V) \cdot m \cdot \exp(i \cdot \Delta\phi + \omega)\} \end{aligned} \quad (4.9)$$

where m , $\Delta\phi$, and ω are used to define the frequency range centered about (U, V) of ψ_k . The range of these parameters were set to:

$$m \in [0.85, 1.3], \quad \Delta\phi \in [-\pi/12, \pi/12], \quad \omega \in [-\pi, \pi]$$

A frequency range defined by these parameters and applied to the type of data used in this study is shown in Fig. 4.5a. Certainly all of the possible frequencies are analyzed and the true motion could be obtained. However, this is impractical for several reasons. First, the computational cost of assessing so many frequencies is extremely high. And many of the values tested will not return useful data as they are too far from the central frequency and outside the range of possible tag deformation. Secondly, this method includes data from multiple harmonics which is undesirable. A much more reasonable range was prescribed in [26]. These ranges are much smaller, meaning they are computationally more efficient, and have been narrowed so that only the first harmonic peak for each orientation is included. The equations for the new ranges are:

$$\begin{aligned} U' &= \Re\{(U + i \cdot V) \cdot m \cdot \exp(i \cdot \Delta\phi)\}, \\ V' &= \Im\{(U + i \cdot V) \cdot m \cdot \exp(i \cdot \Delta\phi)\} \end{aligned} \quad (4.10)$$

where m is a linear variation of ψ_1 and $\Delta\phi$ is the range of angle orientations. It is useful to consider these parameters in terms of polar coordinates where m modulates the radius, r and $\Delta\phi$ modulates the angle, θ . The parameters modulating the new frequency range have the following ranges:

$$m \in [0.81, 1.2], \quad \Delta\phi \in [-\pi/12, \pi/12]$$

These ranges, visible in Fig. 4.5b, were chosen with a specific goal in mind: to create a more precise Angle Image. A more accurate Angle Image provides more accurate information regarding local deformation. The first step is to create Gabor filter bank. The initial filter is centered at the first harmonic peak, (U_1, V_1) . The remaining filters map all of the possible frequencies that could be present in the motion of the myocardium throughout the heart cycle. These filters are then convolved with a local image patch, I , that is centered at pixel (x, y) . In this algorithm, only certain points are chosen. These are the the points-of-interest that were set during segmentation. As shown in Eqn. 4.11, the maximal response for each pixel is then used to determine the optimal Gabor filter $h_o(U_o, V_o)$. The phase of this response is treated as a local Angle Image. From here, the local deformation information can be extracted. As long as the parameters m and $\Delta\phi$ are tuned well, this should create a very precise Angle Image.

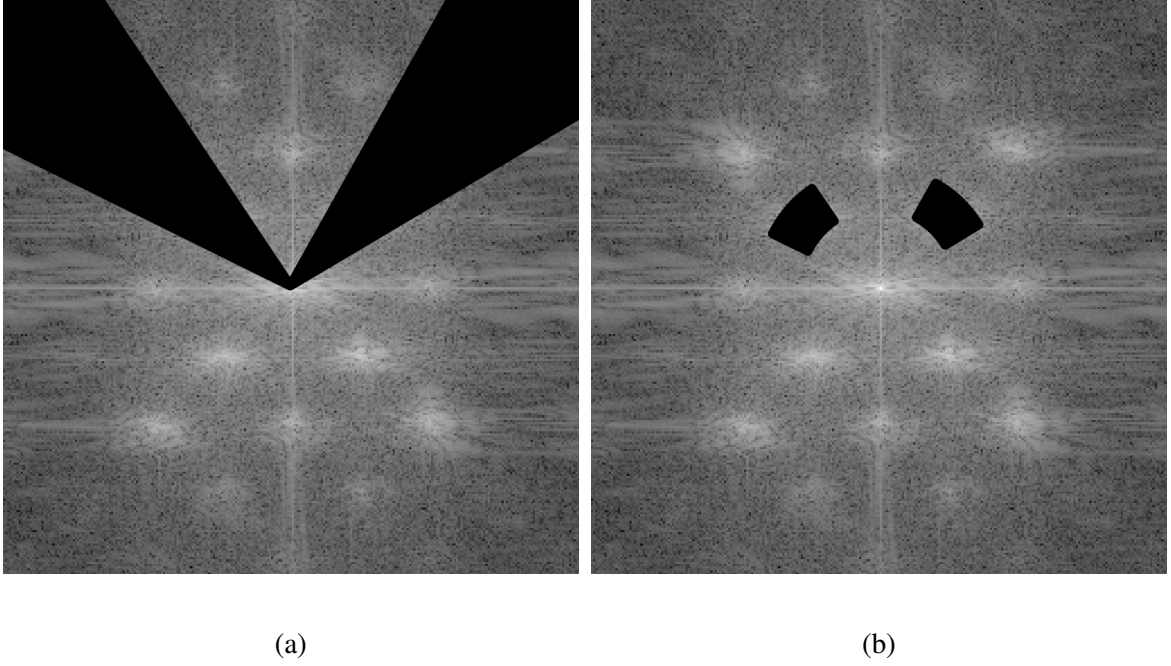


Figure 4.5: Range of frequencies spanned in the original method and the proposed method. a) Previous range of the definition of the Gabor filter bank; b) Current range of the Gabor filter bank.

$$(U_o, V_o) = \underset{U, V}{\operatorname{argmax}} (|h(U, V) * I|) \quad (4.11)$$

4.4 Strain Quantification

Tagged cMR images have two inherent characteristics that prove challenging to overcome: 1) They have tag lines that significantly interfere with LV segmentation; and 2) These tag lines fade over time making it difficult to track them. Due to these two characteristics, semi-automated and fully-automated segmentation of the LV are still active research topics. When using intensity-based approaches with this imaging modality, many problems arise due for various reasons including the problems inherent to tagging such as lower resolution. In order to avoid these problems, it is better to use more robust, intensity-independent properties such as phase.^[26]

By analyzing the local tag deformation, tracking the tag pattern can be avoided. This methods requires some simplifying assumptions: 1) That the myocardial tissue is incompressible; 2) The myocardium is subject to three types of deformations: compressive strain, tensile strain, and local torsion; 3) The first image contains the initial, undeformed, conditions. The first two assumptions are simply for the sake of simplicity. The third assumption references initial conditions-the initial tag orientations and spacings-which are identified by the central frequencies of the first harmonic peaks. This provides the basis for determining the myocardial LaGrangian strains because the subsequent echo times $(t_2, t_3, \dots, t_{i-1}, t_i)$ can

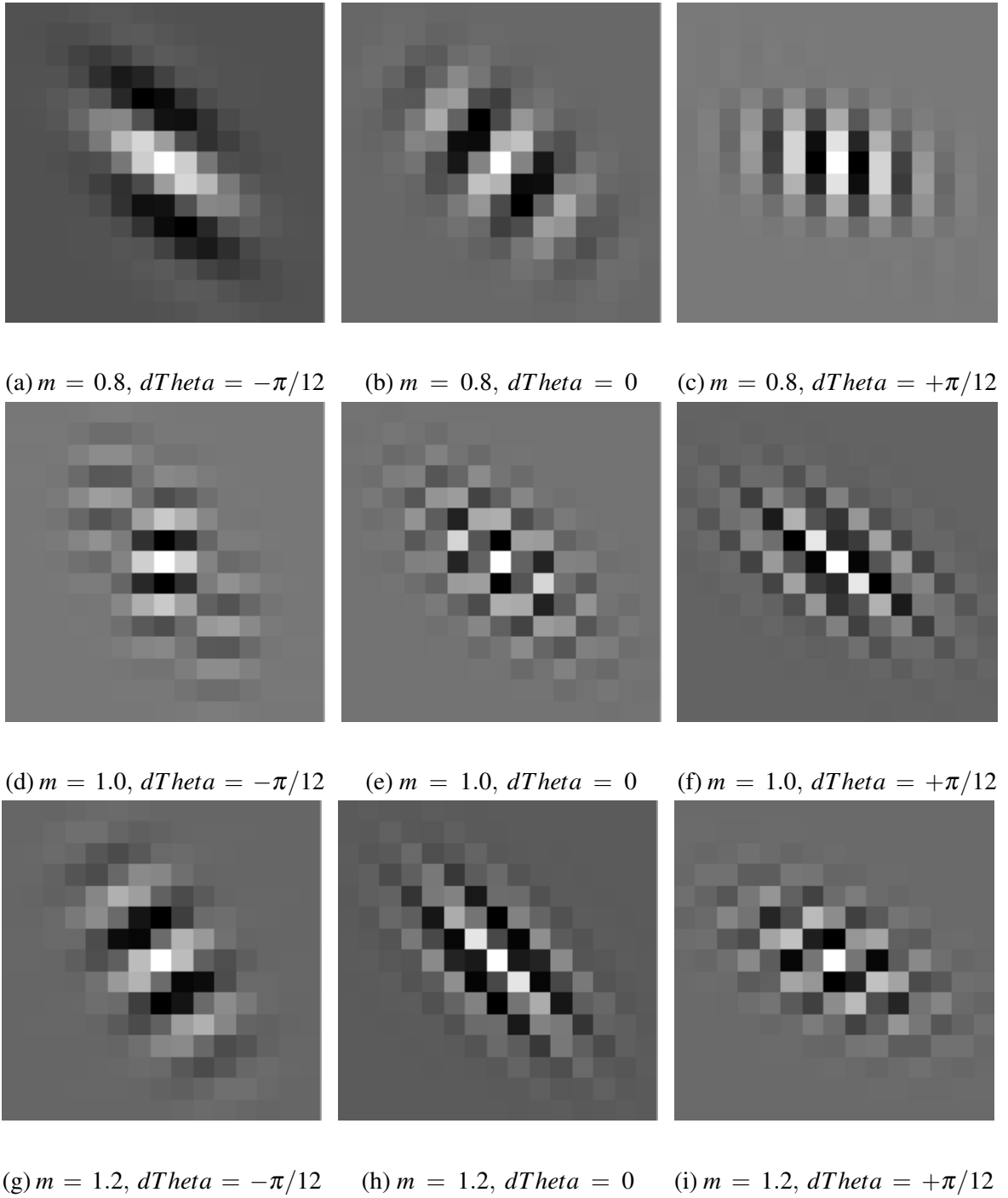
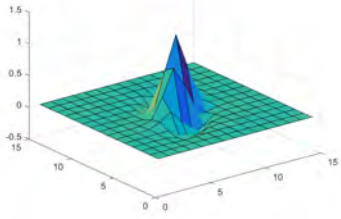
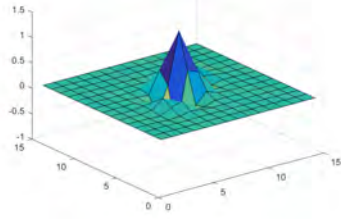


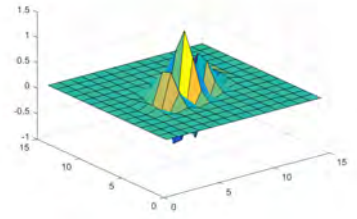
Figure 4.6: 2-dimensional visualization of the Gabor kernels for the second orientation angle, $\omega(1)$. The kernel size is 15×15 pixels, the aspect-ratio $\lambda = 2$, and the coefficient of $\sigma_x = 3$.



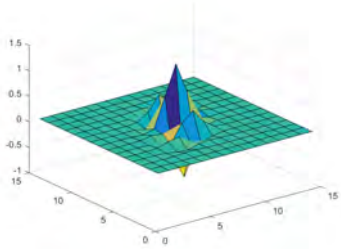
(a) $m = 0.8, d\theta = -\pi/12$



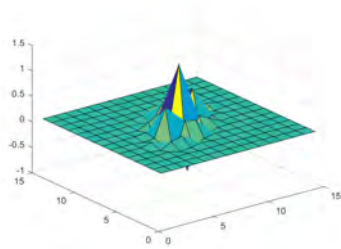
(b) $m = 0.8, d\theta = 0$



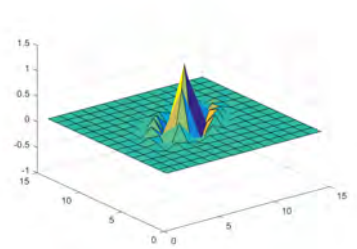
(c) $m = 0.8, d\theta = +\pi/12$



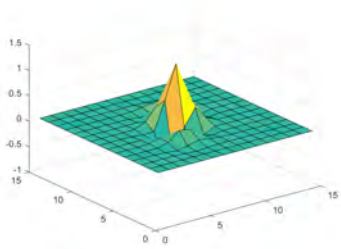
(d) $m = 1.0, d\theta = -\pi/12$



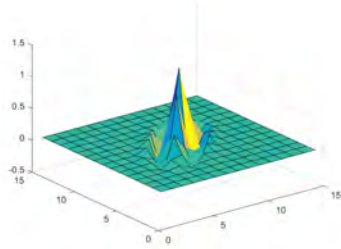
(e) $m = 1.0, d\theta = 0$



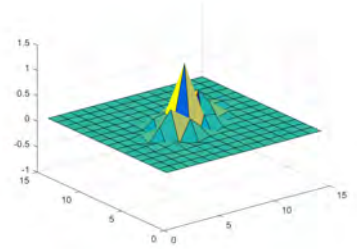
(f) $m = 1.0, d\theta = +\pi/12$



(g) $m = 1.2, d\theta = -\pi/12$

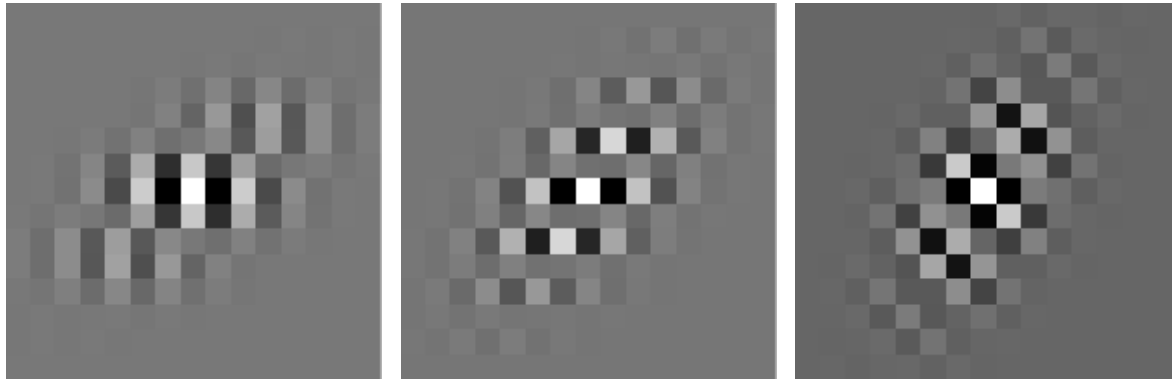


(h) $m = 1.2, d\theta = 0$



(i) $m = 1.2, d\theta = +\pi/12$

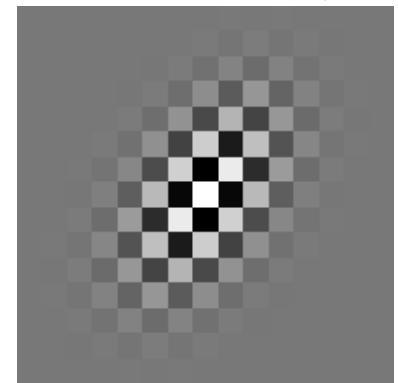
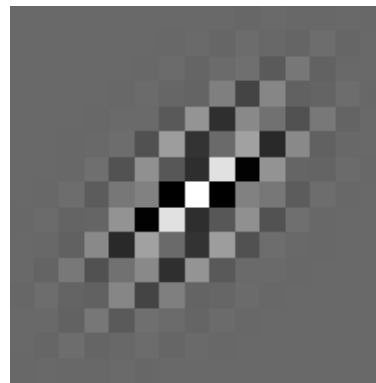
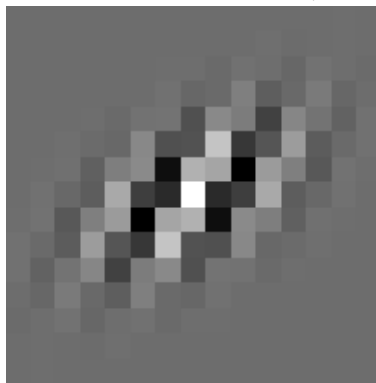
Figure 4.7: 3-dimensional visualization of the Gabor kernels for the first orientation angle, $\omega(1)$. The kernel size is 15x15 pixels, the aspect-ratio $\lambda = 2$, and the coefficient of $\sigma_x = 3$.



(a)
 $m = 0.8, d\theta = -\pi/12$

(b)
 $m = 0.8, d\theta = 0$

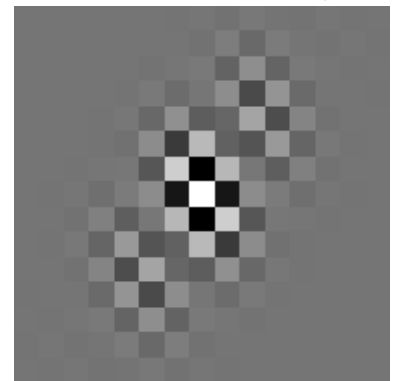
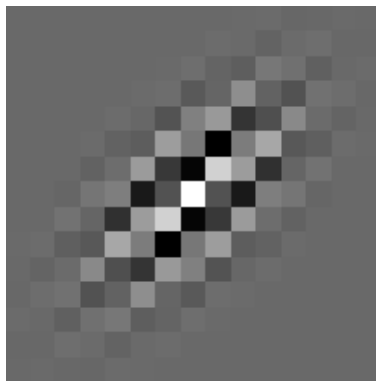
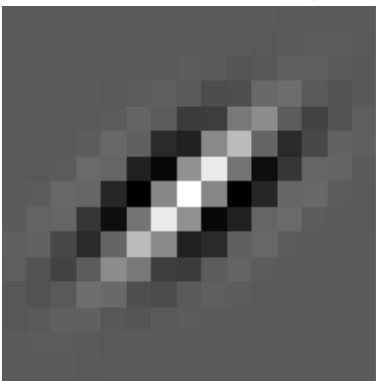
(c)
 $m = 0.8, d\theta = +\pi/12$



(d)
 $m = 1.0, d\theta = -\pi/12$

(e)
 $m = 1.0, d\theta = 0$

(f)
 $m = 1.0, d\theta = +\pi/12$

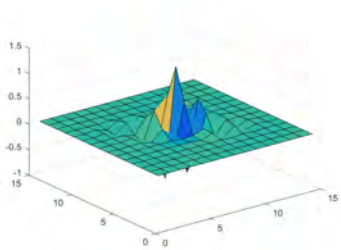


(g)
 $m = 1.2, d\theta = -\pi/12$

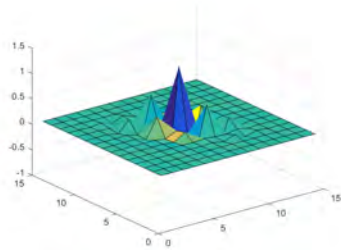
(h)
 $m = 1.2, d\theta = 0$

(i)
 $m = 1.2, d\theta = +\pi/12$

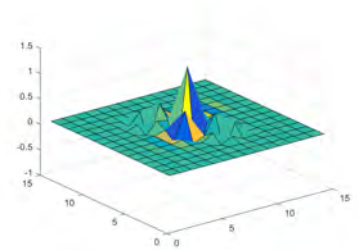
Figure 4.8: 2-dimensional visualization of the Gabor kernels for the second orientation angle, $\omega(2)$. The kernel size is 15×15 pixels, the aspect-ratio $\lambda = 2$, and the coefficient of $\sigma_x = 3$.



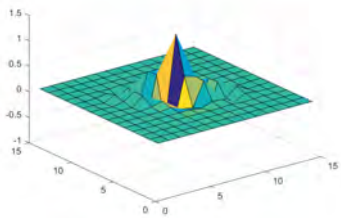
(a)
 $m = 0.8, dTheta = -\pi/12$



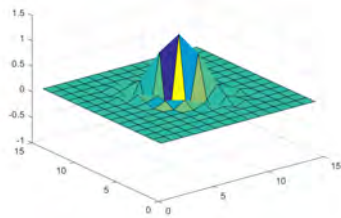
(b)
 $m = 0.8, dTheta = 0$



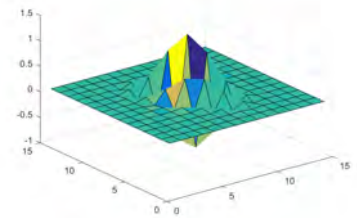
(c)
 $m = 0.8, dTheta = +\pi/12$



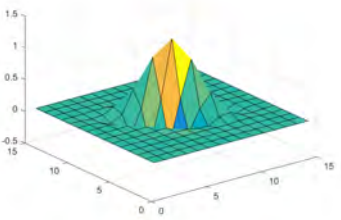
(d)
 $m = 1.0, dTheta = -\pi/12$



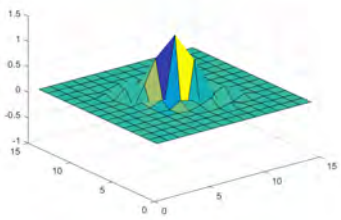
(e)
 $m = 1.0, dTheta = 0$



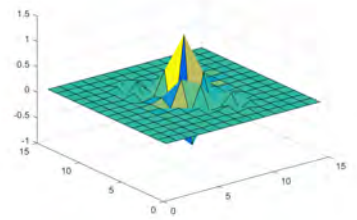
(f)
 $m = 1.0, dTheta = +\pi/12$



(g)
 $m = 1.2, dTheta = -\pi/12$



(h)
 $m = 1.2, dTheta = 0$



(i)
 $m = 1.2, dTheta = +\pi/12$

Figure 4.9: 3-dimensional visualization of the Gabor kernels for the second orientation angle, $\omega(2)$. The kernel size is 15x15 pixels, the aspect-ratio $\lambda = 2$, and the coefficient of $\sigma_x = 3$.

be compared with the initial echo time, t_1 .

This method, although very much rooted in simple geometry, frames the problem in terms of a finite element. The following description is illustrated in Fig. 4.10 taken from [22]. This finite element, also referred to as a myocardial element, is situated in an arbitrary coordinate system \mathbf{X} and has an initial length of $d\mathbf{X}$. At echo times subsequent to the initial frame, the myocardial element deforms. The deformed element's length in coordinate system \mathbf{X} is now $d\mathbf{x}$. At this new position, \mathbf{x} , the strain, which will be referred to as the deformation gradient, is defined as $\mathbf{F} = d\mathbf{x}/d\mathbf{X} = \nabla\mathbf{x}$.

The tMRI images used in this study have two tag orientations. Therefore, it is necessary to use a 2D deformation gradient. The description of the model in Qian *et al.* [22], is based on the initial state having undeformed horizontal and vertical tag lines. The exact initial tag orientations are irrelevant when it comes to the fundamental concept because the orientation of the filter envelopes changes with the location of the filter. This means that they can still be defined in terms of horizontal and vertical tag lines. Justification for this claim will be provided shortly.

Based on the previous concept, the horizontal and vertical tagging have spacings D_x and D_y respectively. After the deformation, the new deformed spacings are S_x and S_y . The change in tag orientations are $\Delta\phi_x$ and $\Delta\phi_y$. This leads to the new two-dimensional deformation gradient tensor \mathbf{F} , which is derived from the following equation:

$$\mathbf{F} = \begin{bmatrix} \frac{S_x \cos\Delta\phi_y}{D_x \sin\phi} & \frac{S_y \cos\Delta\phi_x}{D_x \sin\phi} \\ \frac{S_x \cos\Delta\phi_x}{D_y \sin\phi} & \frac{S_y \cos\Delta\phi_y}{D_y \sin\phi} \end{bmatrix} \quad (4.12)$$

where $\phi = \frac{\pi}{2} - \Delta\phi_x - \Delta\phi_y$. This equation can be solved using the geometry outlined in Fig. 4.10 and the optimized parameters identified from the Gabor filter which elicited the greatest response. There are two additional definitions that are useful in solving this geometry. The first is the peak-to-peak distance of the sinusoid S , which happens to be defined the same as the standard deviations of the Gaussian. This spacing and the tag orientations change throughout the cardiac cycle and are represented by m and $\Delta\phi$ respectively:

$$S = 1/\sqrt{U^2 + V^2} \quad m = S/D, \quad \text{and} \quad \Delta\phi = \phi - \phi_i$$

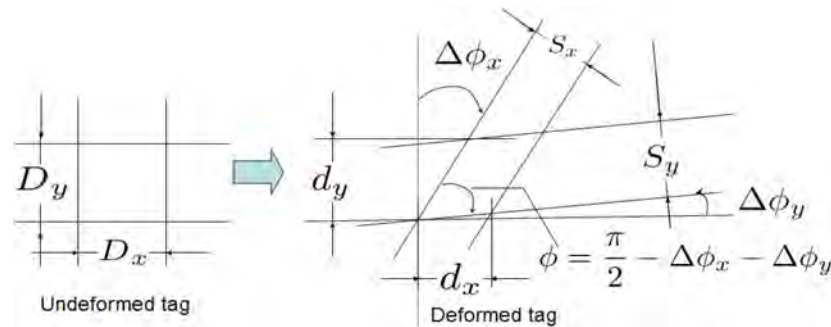


Figure 4.10: This is the graphical illustration of Eqn. 4.12. After the tag deforms, $d_x = S_x \cos\Delta\phi_y / \sin\phi$ and $d_y = S_y \cos\Delta\phi_x / \sin\phi$. [22]

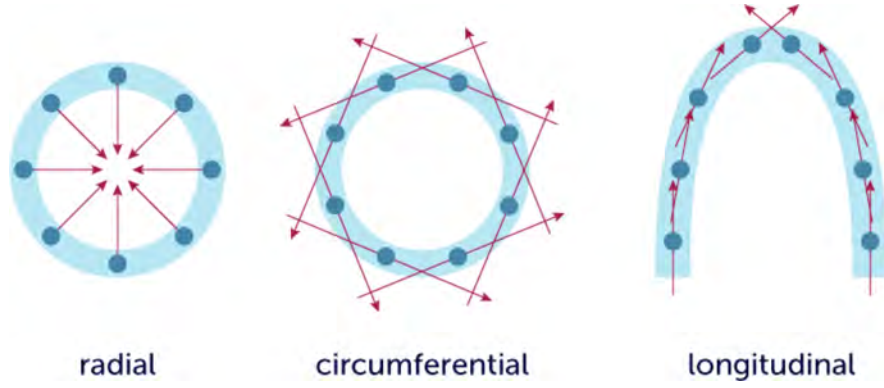


Figure 4.11: There are three important types of strain in myocardial analysis: radial, circumferential, and longitudinal which is also referred to as axial.^[39]

Once parameters (U, V) have been optimized and \mathbf{F} has been solved, then the Lagrangian finite strain tensor \mathbf{E} and the local rotation matrix \mathbf{R} can be calculated by the following equations:

$$\mathbf{E} = \frac{1}{2} (\mathbf{F}^T \cdot \mathbf{F} - \mathbf{I}) \quad (4.13)$$

$$\mathbf{R} = \mathbf{F} (\mathbf{F}^T \cdot \mathbf{F})^{-1/2} \quad (4.14)$$

where \mathbf{I} is an identity matrix and the exponent T indicates the transpose of the matrix. In myocardial deformation research when assessing the myocardial function and wall motion, the most useful strains are the radial and circumferential strains, and in three-dimensional studies, the axial strain. In order to determine the radial and circumferential strains, the 2D horizontal-vertical Lagrangian finite strain tensor can be transformed into a radial-circumferential strain tensor $\hat{\mathbf{E}}$ using a rotation matrix^[19,22,37,38] $\mathbf{Q}(\theta)$:

$$\hat{\mathbf{E}} = \mathbf{Q}\mathbf{E}\mathbf{Q}^T \quad (4.15)$$

where the rotation matrix \mathbf{Q} is defined by an angle θ about the centroid of the myocardium and is considered positive in the counter-clockwise direction.

Because the Lagrangian tensor is a horizontal-vertical strain tensor, this does not provide information about the required strains. As shown in Fig. 4.11, taken from [39], myocardial radial strains are oriented about the centroid of the myocardium. In order to calculate this radial strain for each PoI, \mathbf{E} must be rotated by angle θ to align with the angle of the particular PoI about the centroid. Fig. 4.12 illustrates how a rotation of the tensor by θ changes the outcome from the horizontal-vertical value of ϵ_x to the radial value of $\epsilon_{x'}$. Similarly for the circumferential strains, Fig. 4.11 shows that these strains are tangential to the PoIs about the centroid. So following the same logic, by rotating \mathbf{E} by $\theta + 90^\circ$ will produce the circumferential strain. Therefore, Eqn. 4.15 can be expanded and understood to be Eqn. 4.16.

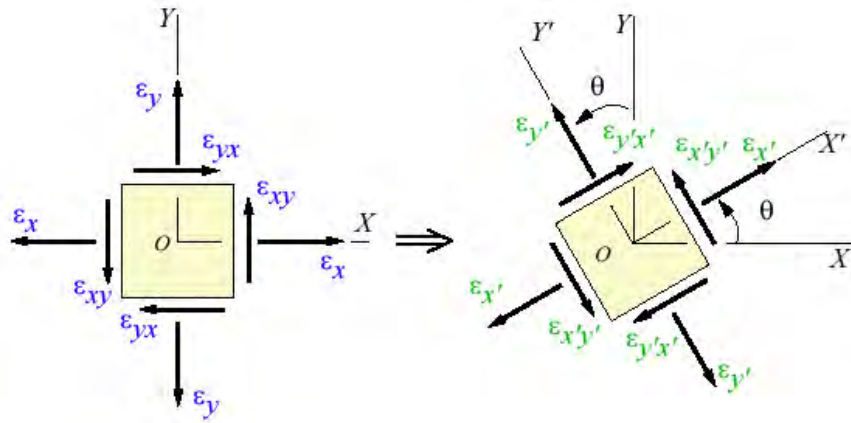


Figure 4.12: In-plane strain diagrams: the original coordinate system is on the left and the rotated coordinate system is on the right. Modified from [40].

$$\hat{\mathbf{E}}(x,y) = \mathbf{Q}(x,y)\mathbf{E}(x,y)\mathbf{Q}^T(x,y), \text{ where}$$

$$\mathbf{Q}(x,y) = \tilde{\mathbf{R}}(\theta_{R,C|(x,y)}) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.16)$$

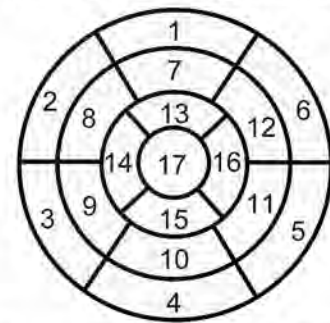
where $\theta_{R|(x,y)}$ is the angle of the point-of-interest (x,y) about the centroid and is used to calculate the radial strain, while $\theta_{C|(x,y)} = \theta_{R|(x,y)} + 90$ is used for the circumferential strain.

4.5 AHA Left Ventricular Segmentation Model

The Cardiac Imaging Committee of the Council on Clinical Cardiology of the American Heart Association released a statement in 2002, written by the AHA Writing Group on Myocardial Segmentation and Registration for Cardiac Imaging, establishing a standardized model and its corresponding nomenclature, to be used in tomographic imaging of the heart, for the segmentation of the left ventricular myocardium.

When evaluating the myocardium and the LV cavity, it is recommended to use the 17-segment model. An ideal model should have an appropriate number of segments to adequately represent the anatomical data.

Left Ventricular Segmentation



- | | | |
|------------------------|-----------------------|---------------------|
| 1. basal anterior | 7. mid anterior | 13. apical anterior |
| 2. basal anteroseptal | 8. mid anteroseptal | 14. apical septal |
| 3. basal inferoseptal | 9. mid inferoseptal | 15. apical inferior |
| 4. basal inferior | 10. mid inferior | 16. apical lateral |
| 5. basal inferolateral | 11. mid inferolateral | 17. apex |
| 6. basal anterolateral | 12. mid anterolateral | |

Figure 4.13: Circumferential polar plot representation of the 17-segment myocardial model and the recommended nomenclature. [33]

Initially a 20-segment model was the standard; however, a more recent evaluation concluded that this model overrepresented the apex and apical cap when compared to the anatomical data. This model was reduced to 16 segments, but still lacked a true apical myocardial segment devoid of the ventricular cavity. Therefore, the 17-segment model was chosen as it accurately represents each myocardial segment.

The recommended naming convention dictates that the segment name should define both the relative location of the segment along the long-axis of the heart and its circumferential position. Keeping the recommended nomenclature in mind, the 17-segment model divides the LV myocardium into four sections along the long-axis: basal, mid, apical, and apex. The two most superior levels are each subdivided into six equal segments: anterior, anteroseptal, inferoseptal, inferior, inferolateral, and anterolateral. Each segment spans 60° . As shown in Fig. 4.13 from [33], segments 1 and 7, basal anterior and mid anterior respectively, both begin at 60° and the subsequent sections start at $n \cdot 60^\circ$.

The apical layer is divided into four equal segments of 90° . These segments are called: apical anterior, apical septal, apical inferior, and apical lateral. The apical anterior segment is defined from $[45^\circ, 135^\circ]$. The remaining segments are defined to begin at $45^\circ + n \cdot 90^\circ$. The seventeenth and final segment is the apex which is completely devoid of the left ventricular cavity. These segmentation guidelines are clearly illustrated in Fig. 4.14 from³³.

When displaying the the results of LV myocardial analyses, the heart should be divided into three equal segments perpendicular to the long-axis of the left ventricle. The resulting regions are circular SA views of the basal, mid-cavity, and apical regions of the LV myocardium. As shown in the long-axis illustrations in Fig. 4.14, the apical short-axis view excludes the true apex. Representative slices of each region should be selected. The slice for the apical SA region should fall within the range between the papillary muscles and the end of the LV cavity with the apical cap falling just outside that range. Anatomical landmarks should be considered when selecting slices. In general, slices should have a thickness of $< 1\text{cm}$, but this can vary based on the modality-specific resolution and the clinical relevance. Only slices containing the complete myocardium in all 360° should be accepted.

The guidelines recommend that the apical cap be evaluated from the vertical and horizontal LA planes for all imaging modalities. However, due to the nature of the investigations in this study, it may prove interesting and useful to examine the apical cap in the short-axis plane as well. These guidelines will be especially important in the future steps of the main research project that will be utilizing this algorithm.

4.6 Viability Assessment

Given that there is no gold standard for assessing myocardial strain, there is a greater amount of freedom in developing assessment protocols. In this study, three methods were used to assess the ability of the algorithm to evaluate myocardial strain. The first method employs a software called Segment^[34], Version 2.0 R4596. This program was designed and developed by Medviso AB and Lund Cardiac MR Group at Lund University and is freely

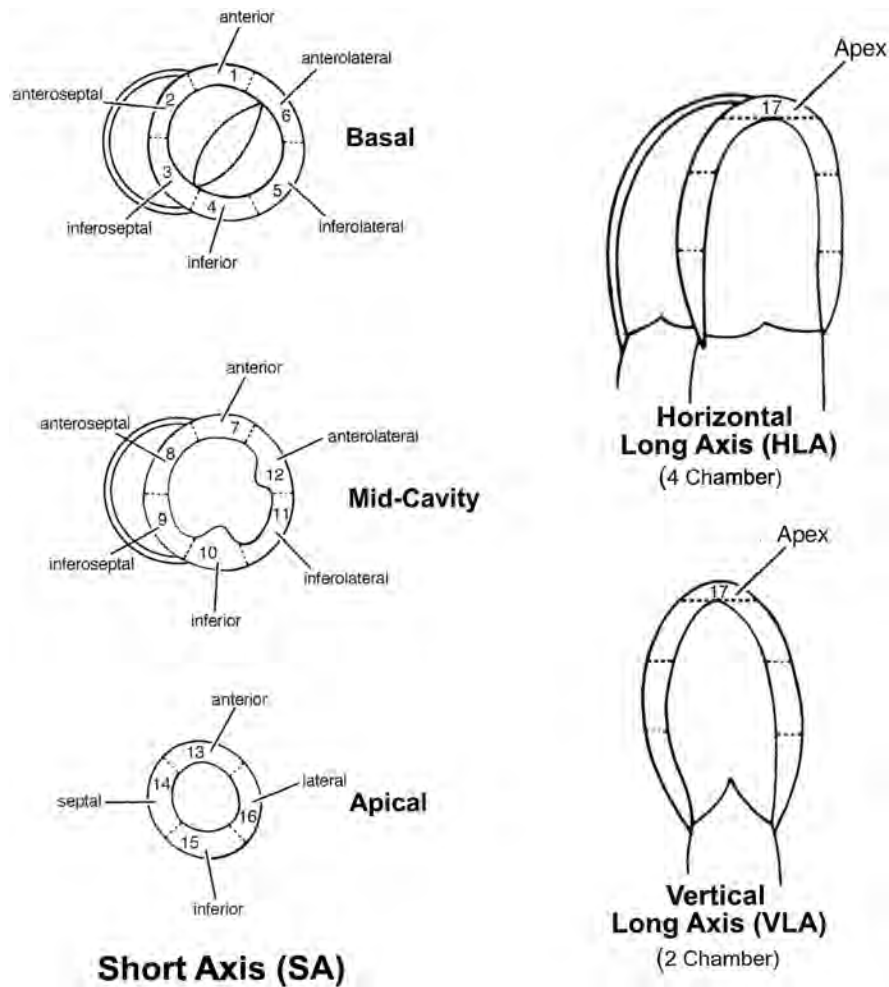


Figure 4.14: On the left, anatomical landmarks can be seen in the short-axis segmentation regions for each of the three regions orthogonal to the long-axis. On the right, the long-axis views show the location of the 17th segment, the apex. This segment is completely devoid of the left ventricular cavity. It encompasses the remainder of the apical cap not included in the apical region of the LV cavity wall.^[33]

available for academic research and has been validated in [34]. This software was used to analyze a synthetic data set. The results were then used to evaluate the new algorithm. This follows the convention for assessing new algorithms. This is considered a very robust method of evaluation because synthetic data is analogous to a controlled environment. Free of undesired artifacts, synthetic data allows one to test whether or not a theoretical concept works by allowing it to test only the parameters of interest.

The second method uses Segment to analyze physiological data sets, the control group, along with pathological data sets, the test group, and then to compare these results with those from the algorithm. Actual data sets have a great range of variability from image quality spatial resolution to contrast of the tags. This protocol, similar to the first, assess the accuracy of the new algorithm when it comes to using real data. This is important in order to show a high level of robustness.

The last protocol is less about accuracy and focused more on being able to differentiate the results of healthy patients from those of the test group. The goal of this protocol is to

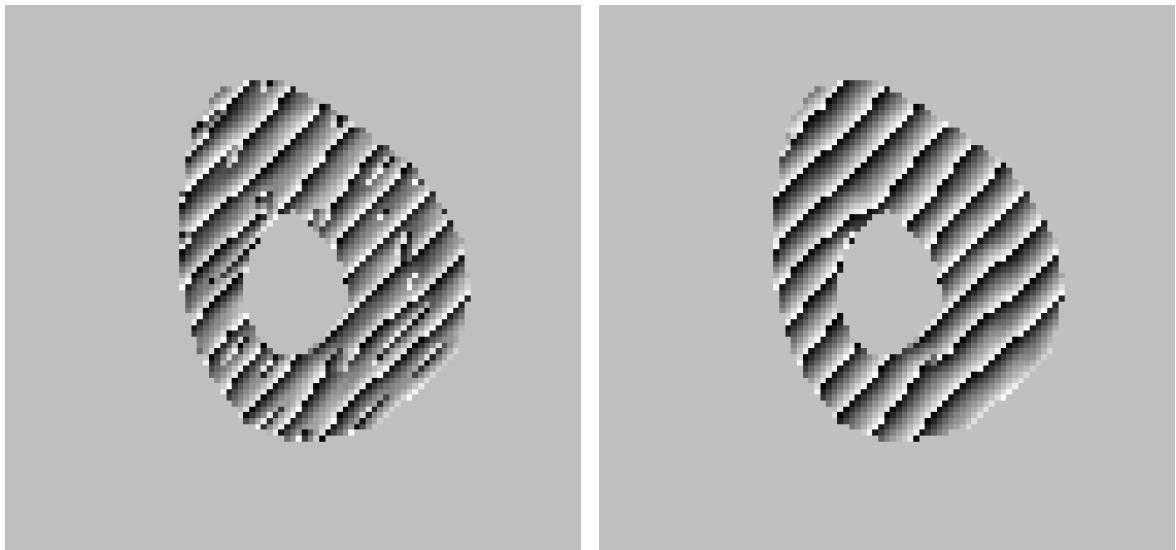
quantitatively analyze the results by firstly determining the type of the data distribution and then implementing the corresponding statistical test.

4.7 Implementation Notes

4.7.1 Remarks

As noted in Qian *et al.* [22], it is necessary to filter the input images. As can be seen in Fig. 4.15a, there is some noise in the form of additional points and lines that do not correspond to the tagging pattern. This is caused by the slight variations in proton density within the tissue which changes the signal intensity between pixels in the image. To start, the input image was smoothed using a Gaussian smoothing filter in order to eliminate these random spikes and to make the homogenous regions more uniform. This smoothing also makes the edges softer, which is highly undesirable. In order to compensate for this effect, the image is then sharpened with an unsharp mask. The results are significant. As Fig. 4.15b shows, this filtering protocol is a great improvement.

The infrastructure for some future features and options has already been included.



(a)

(b)

Figure 4.15: a) The Angle Image of the original, unfiltered input image. b) The Angle Image of the filtered input image.

This is only the initial stage of development. Therefore, it appears that some options are available, but may elicit an error. For example, the testing was performed only on basal and mid-cavity slices, but there is the option to select "Apical". This feature will certainly be necessary in the future, but was not a priority during this work. Other features are in the way arguments are passed between functions. The end-user might not know it is there, but it

should make it easier for the next person to continue developing the algorithm.

4.7.2 Program Descriptions

Segment

Segment.m

Name = Segment

Segment is the parent file for segmenting the desired files and saving the results in a .mat file whose name is returned.

Segment allows the user to select a DICOM file via an interactive GUI. The image is checked to make sure the format is correct before proceeding. The slice label is selected through a dialog. The first harmonic peaks are interactively identified before guiding the user through the manual segmentation of the endocardium and epicardium. The output variables of this file are:

angles: Vector containing the angles of the first harmonic spectral peaks about the center

centroid: Centroid of the myocardial contour

contourMask: Contour of the myocardium

Im: Original SPAMM-tagged MR image

name: Name of the original input file

omega: Central frequency or frequencies of the first harmonic peak(s)

rect: Vector specifying [xmin ymin width height] of the chosen ROI

sector: Column vector indicating which PoIs in test_points are in which AHA 17-segment model section

test_points: Chosen PoIs between the endocardial and epicardial contours

time: Trigger time of the image within the cine sequence

imageCheck.m

image = imageCheck(image)

imageCheck ensures that the input image from the DICOM file is in the correct format for processing. It will convert 1- and 3-layer non-double class images to grayscale images of class double.

chooseLabelDialog.m

label = chooseLabelDialog

chooseLabelDialog is an interactive dialog that requires the user to identify the long-

axis region that the slice represents. There are three options: Basal, Mid-Cavity, and Apical.

ROI.m

`varargout = ROI(varargin)`

ROI helps the user specify the region-of-interest. The first input must be an image. If the region-of-interest has already been specified then it can be entered as the second input argument. If only one output is specified, then the cropped image will be returned. If two are specified, the second output will be 'rect' which was returned from `imcrop` and has been rounded to the nearest whole integer.

FourierPeaks.m

`[omega, angles] = FourierPeaks(image)`

FourierPeaks is a semi-automated approach to identify the central frequencies of desired spectral harmonic peaks in the 2D Fourier space. It also returns the angles of the peaks about the center. Both of these values are used in creating the Gabor filter bank.

Segmentation.m

`[segmentLabels, contourMask, test_points, sector, members, centroid] = Segmentation(image, AHA, nPoints, rows, sectionAngles, segmentLabels)`

Segmentation finds the contours and the centroid of the myocardium, resamples the contours, and assigns the PoIs to their respective segments according to the AHA 17-segment model.

contours.m

`[contourMask, endo_cont, epi_cont] = contours(image)`

contours.m does 1 of 2 things depending on the number of inputs. If only the image is given, then the user will be prompted to select points along the endocardial and epicardial contours. The more points selected, the better the contours will be. If the contours have already been identified and given as inputs, first the endocardial contour followed by the epicardial contour, then the contour masks will be created and the program terminates.

Centroid.m

`centroid = Centroid(contourMask)`

This program uses the polygon masks created in `contours.m` and finds the centroids of the myocardial contour.

testPoints.m

`[strain_pts, sector, combined, segmentLabels] = testPoints(centroid, epi_cont, endo_cont, AHA, nPoints, rows, sectionAngles, segmentLabels)`

Define the coordinates of the points between the contours to be analyzed. There will be `nPoints` number of points in AHA number of sections depending on the slice location, i.e. base, mid, or apical. The contours will be resampled at uniform arc lengths.

sortContour.m

Contour = sortContour(contour, centroid, sectionAngles)

sortContour sorts the points of a contour first according to its angle (in radians) about the centroid, and then corrected by choosing the nearest point spatially with the lowest angle.

resample.m

Resampled = resample(contour, noPts)

resample returns noPts number of points along the given contour spread equidistantly along the perimeter of the contour.

SegmentFinder.m

[angNum, segmentLabels] = SegmentFinder(point, centroid, newPts, sectionAngles, segmentLabels)

SegmentFinder assigns the POIs to the appropriate segment according to the AHA 17-segment model.

GaborFilterBank

GaborFilterBank.m

Name = GaborFilterBank(varargin)

GaborFilterBank builds a bank of Gabor filters for data sets with either 1 or 2 tagging directions. The segmented file (*.mat) can either be passed as an input or if no inputs are provided, the it will be selected interactively. The output variables stored in the .mat file are the:

kernels1: Kernel values for the first orientation angle, $\omega(1)$

lookup1: Lookup table with the parameters for every kernel in kernels1

kernels2: Kernel values for the second orientation angle, $\omega(2)$

lookup2: Lookup table with the parameters for every kernel in kernels2

GaborKernel.m

[kernel, varargout] = GaborKernel(matrixsize, theta, omega, centerpoint, m)

GaborKernel creates a single Gabor filter. The outputs are the Gabor kernel itself along with the U_{prime} , U' , and V_{prime} , V' , values associated with that kernel.

GaborStrain ** Update Optimize and Adjust (if needed)!!!

GaborStrain.m

Name = GaborStrain(GaborFilterBank)

GaborStrain uses the Gabor filter bank specified in the input to analyze the selected *.mat files and saves the results for each in a separate output file. Each input file is

filtered and the optimized filter parameters and Angle Image are returned. Then the strain is calculated for the test_points for each image.

Optimize.m

```
[image, optiParameters, IND, contourMask, AngleImage, Filtered, FilteredMag, FilteredPhase]  
= Optimize(Im, contourMask, Rect, rect, kernels, test_points, lookuptable)
```

Optimize.m filters the points of interest, defined in testPoints, of the input image with the Gabor filters contained in kernels. The results are then evaluated searching for the greatest magnitude response. The parameters that generated the maximum response are then returned stored in a lookup table, optiParameters, and an Angle Image is created by storing the phase responses of each pixel elicited by the optimized filters.

adjust.m

```
[image, test_points, contourMask] = adjust(Im, Rect, test_points, contourMask)
```

adjust will modify the ROIs, contours, and PoIs that were selected during the segmentation process so that all of the images in the sequence use the same ROI as the first image in the sequence. Therefore, the dicom images are cropped according to the original region-of-interest, Rect. The coordinates in the contour mask and the test_points vector will be adjusted by the change between Rect and rect, the original ROI of the current image. The contour mask is also redefined to match the size of the original ROI as indicated by Rect.

combineStack

combineStack.m

```
Name = combineStack(directory)
```

combineStack loads the desired *analyzed.mat files for an image sequence and combines the results into single variables. The outputs can then be displayed in plots or tables to visualize and assess the results.

5. RESULTS

To assess the validity of the developed algorithm, it was tested according to three different protocols. The first protocol required synthetic data to be analyzed by Medviso’s Segment and the algorithm in question and then to compare the results. The second protocol called for comparing the results of actual physiological data with that of Segment. And finally, the results of analyzing physiological and pathological data with the algorithm were to be compared to ensure that the algorithm is capable of differentiating between the two. The results are described below.

5.1 Synthetic Data

As convention calls for assessing algorithms via synthetic data, the eight images in Appendix B were assessed in Segment to establish ground truth values for comparison. The results of this analysis are inconclusive at best and unfit for use in assessing the developed algorithm. As can be seen in Fig. 5.1, the contours and points-of-interest that were automatically generated by Segment are unsuitable for accurate analysis. Although the epicardial and endocardial contours were drawn manually for each image before strain analysis, for some reason they are only used when an image is selected as the initial time frame. The greater the deformation of the synthetic myocardium, the more radical the contours become. Likewise, one row of the chosen PoIs falls along the endocardial contour and the fifth row falls along the epicardial contour while the remaining three rows fall in-between. This is the same methodology as what is employed in the new algorithm as well. However, due to the abnormal behavior of the contours, many of these points begin to fall outside of the myocardial boundaries where no data exists. Whether this error is due to the deformations of the synthetic myocardium being so large and inconsistent, it is unclear. The subsequent strain data produced by Segment has been omitted and will not be used. This data set is not used in any further analyses.

5.2 Physiological Data

This data set consists of 22 SPAMM-tagged cine sequences. In this data set, the myocardial volume changes rather significantly throughout the cine sequence. Fig. 5.2 accurately represents the results of this sequence by displaying every seventh image from the sequence. Similar to the results from the Synthetic Data set, Segment does not adhere to the manually drawn contours. Although more of the PoIs are inside the myocardium, there are too many outside of the myocardium for this to be use for comparison.

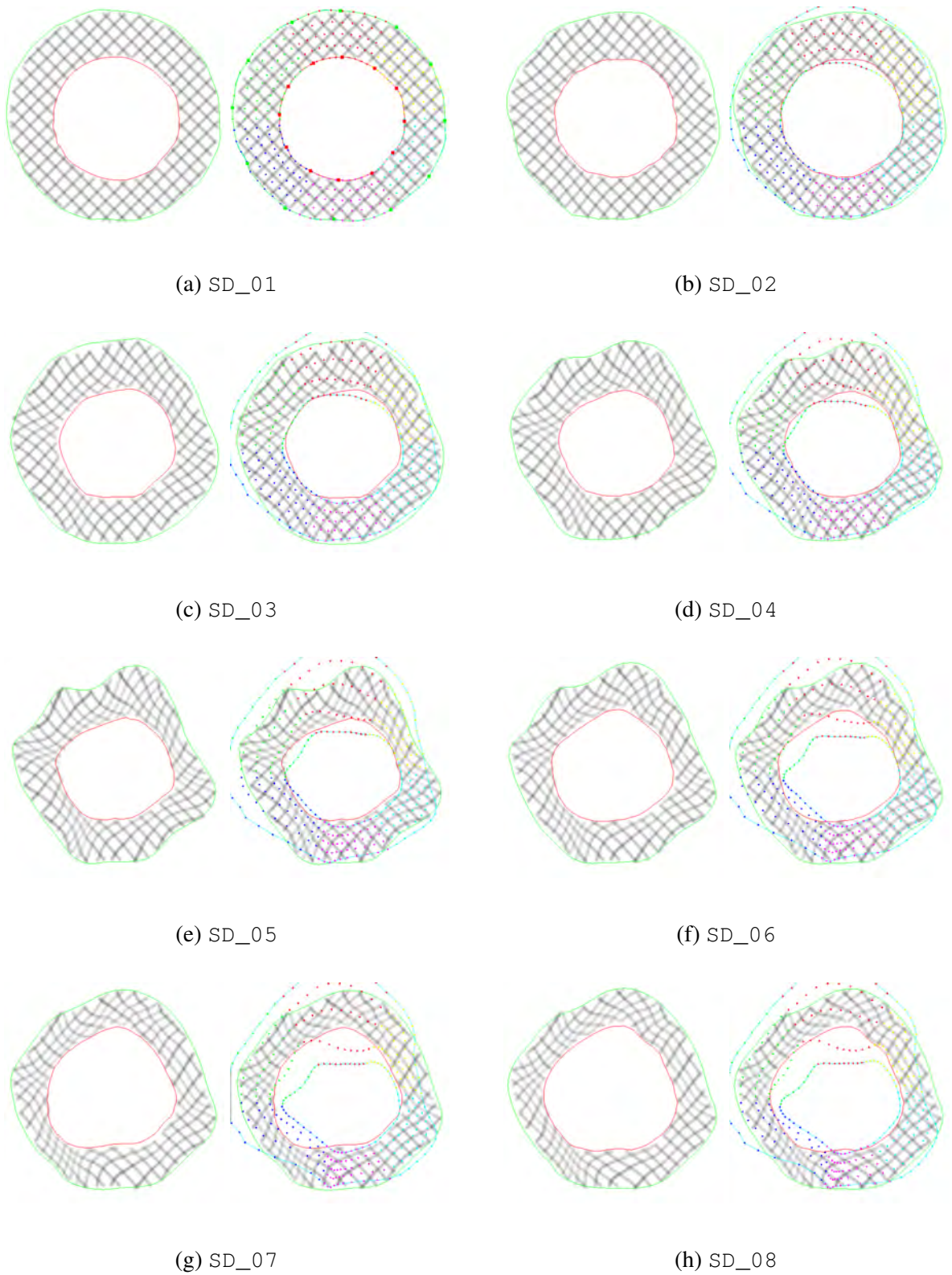
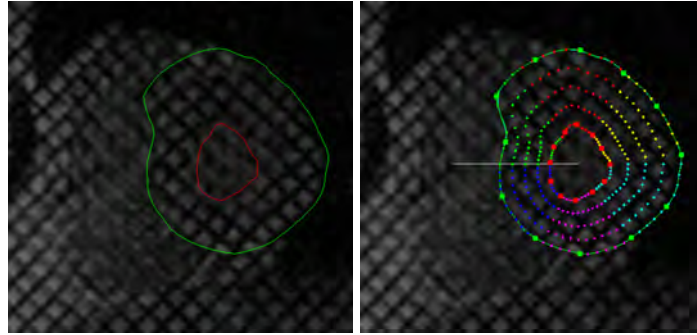
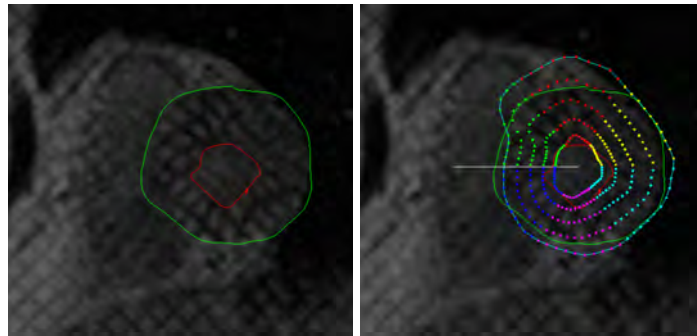


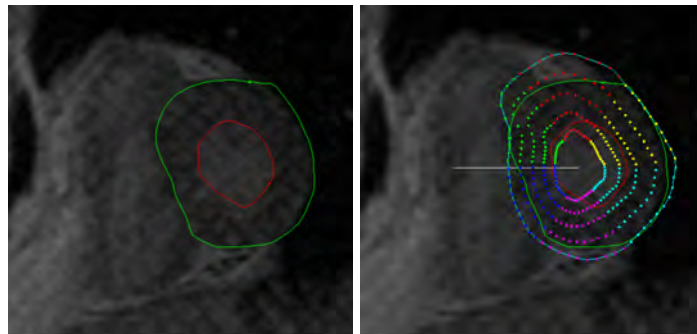
Figure 5.1: These are the analysis results of the Synthetic Data set using Segment. The images on the left are the manually drawn contours. The images on the right have both the manually drawn contours, the contours created by Segment, and the PoIs automatically selected by Segment for analysis.



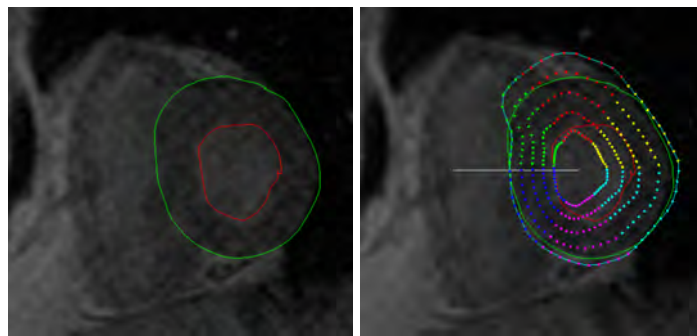
(a) Normal01B_01



(b) Normal01B_08



(c) Normal01B_15



(d) Normal01B_22

Figure 5.2: These are the analysis results of one of the healthy data sets using Segment. The images on the left are the manually drawn contours. The images on the right have both the manually drawn contours, the contours created by Segment, and the PoIs automatically selected by Segment for analysis.

5.3 Quantitative Comparison

Given the relatively small amount of data, it was not possible to determine the type of distribution. The Lilliefors test, the one-sample Kolmogorov-Smirnov test, and two distribution fitting programs, including allfitdist^[41] from the MathWorks File Exchange, that test the data for a wide variety of distribution types were unable to indicate a distribution type. Therefore, a non-parametric statistical analysis was required. The Mann-Whitney U-test was selected as the appropriate statistical test. The hypotheses are as follows:

Null hypothesis: The two samples come from distributions with equal medians

Alternative hypothesis: The two samples come from distributions with unequal medians

Ten data sets, each with six trigger times, were used in the physiological group and in the pathological group. The data sets were separated according to trigger times. The ten physiological data sets were combined for each trigger point while the pathological data was kept separate. For each trigger, each of the ten pathological data sets were compared to the physiological data set for the given trigger. The results are shown below: H = 1 indicates that the

Table 5.1: Hypothesis decisions of the Wilcoxon ranksum test

		Data Set									
		1	2	3	4	5	6	7	8	9	10
Trigger 1		1	1	1	1	1	1	1	1	1	1
Trigger 2		1	1	1	1	1	1	1	1	1	1
Trigger 3		1	1	1	1	1	1	1	0	1	1
Trigger 4		1	1	1	1	1	0	1	0	1	1
Trigger 5		1	1	1	1	1	0	1	1	1	0
Trigger 6		1	1	1	1	1	0	1	1	0	1

null hypothesis that the two data sets have an equal median is rejected while H = 0 indicates that the null hypothesis was not rejected. The p-values, shown in Table 5.2, represent the probability of the current data being observed if the null hypothesis were accepted. P-values of $p \leq 0.05$ indicate that the null hypothesis can safely be rejected.

Table 5.2: P-values of the Wilcoxon ranksum test

Data Set					
	1	2	3	4	5
Trigger 1	$1.3713E^{-39}$	$4.8019E^{-127}$	$1.1371E^{-60}$	$2.7333E^{-79}$	$5.8066E^{-144}$
Trigger 2	$4.2263E^{-46}$	$7.8448E^{-229}$	$3.0927E^{-74}$	$1.0591E^{-56}$	$6.4646E^{-47}$
Trigger 3	$8.3865E^{-65}$	0	$5.9651E^{-233}$	$3.9624E^{-63}$	$2.3633E^{-16}$
Trigger 4	$5.2535E^{-73}$	0	$1.9179E^{-62}$	$1.2534E^{-127}$	$2.1423E^{-95}$
Trigger 5	$7.6317E^{-75}$	0	$6.885E^{-57}$	$4.246E^{-237}$	$7.824E^{-178}$
Trigger 6	$4.3801E^{-75}$	0	$7.0895E^{-67}$	$8.2287E^{-155}$	$1.9753E^{-75}$
	6	7	8	9	10
Trigger 1	$2.3522E^{-4}$	$1.6956E^{-18}$	$2.5606E^{-26}$	$3.0319E^{-150}$	$5.5905E^{-74}$
Trigger 2	$7.8203E^{-4}$	$4.5242E^{-17}$	$2.7328E^{-26}$	$6.0148E^{-162}$	$1.8971E^{-254}$
Trigger 3	$9.1003E^{-4}$	$7.1602E^{-06}$	0	$1.4303E^{-16}$	$1.4192E^{-144}$
Trigger 4	0.60346	$2.1473E^{-2}$	0	$7.4374E^{-06}$	$7.916E^{-202}$
Trigger 5	0.6503	$2.722E^{-2}$	$1.6523E^{-09}$	$6.3399E^{-73}$	0
Trigger 6	0.79162	$5.0393E^{-3}$	$5.9348E^{-10}$	0	$4.2433E^{-123}$

6. DISCUSSION

As can be seen in Table 5.1, there are seven instances in which the null hypothesis is accepted. It was found that the data sets in which the null hypothesis was accepted and the p-value is equal to zero elicited an error during processing and were therefore excluded. Of the 55 remaining data sets, 52 of them rejected the null hypothesis indicating a significant statistical difference between the physiological and pathological data regarding the radial strain within the myocardium. The developed algorithm successfully differentiates between physiological and pathological radial strain data with a probability of about 95%.

Looking at Table 5.2, the p-values indicate the level of certainty of the hypothesis decision to reject the null hypothesis. All but six of the non-zero values, only three of which are unable to reject the null hypothesis, range in orders of magnitude from -4 to -245 . 9 of the 55 values have single-digit orders of magnitude. 25 of the 55 values have double-digit orders of magnitude. And 14 of the 55 values have triple-digit orders of magnitude. $p = 0.05$ is the required value to accept the hypothesis decision. 4 data sets produced p-values so small, they were displayed as zero. To say these results are strong indicators is a great understatement. Only about 5% of the pathological data was undistinguishable from the physiological data. These results indicate great success for algorithm.

6.1 Limitations

In the end, there are more limitations to this study than expected. The two of the three protocols for assessing this algorithm failed. the protocols that failed would have been able to compare accepted values with the results of the algorithm. This means that the accuracy of the algorithm could not be assessed. The remaining protocol was a preliminary statistical analysis.

Data from the entire pathological image, including segments that may or may not exhibit any signs of mechanical dyssynchrony, were compared to a larger set of data from a similar trigger time. Segments with noticeable deformation were not isolated and compared to the physiological data. And as can be seen in Tables 4.2 and ??, the trigger times within each collection of data sets is relatively uniform. However, that uniformity is not matched when comparing the two. It would better if the trigger times were matched for the control and test group data sets to ensure accuracy. The duration of the cardiac cycle should also have been taken into account to ensure that the strain values are being compared from the same part of the cardiac cycle.

The statistical analysis only addresses radial myocardial strain. The rotation angle, circumferential strain, and the axial strain have yet to be analyzed. These values are necessary to accurately evaluate the success of the algorithm.

Another limitation is the relatively small sample size. Several evaluation methods

were used on various grouping of the data sets, but no distribution type could be determined. Significantly increasing the number of segmented data sets is necessary for a true evaluation.

Increasing the sample size requires more segmented data sets. Aside from the cost of gathering MRI data sequences, a significant amount of time is required to segment each image in a given sequence. It should be noted that the data sets used in this study were not segmented by a healthcare professional. Because the only viable method of statistical analysis was to compare the two samples to each other, the precision of the segmented contours was not as important as consistency. In future studies this may become more important.

7. CONCLUSION

The primary objective of this work was to create an algorithm that accepts tagged cardiac MR images stored in DICOM format and is able to differentiate between physiological strain and pathological strain. Typically, HARP is one of the most common methods for analyzing MR-tagged images. To overcome HARP's limitation of small deformation, a bank of Gabor filters was created to analyze local deformation centered at a wide variety of frequencies. By finding the greatest magnitude response, a very precise Angle image is developed using the phase response of the optimal Gabor filters. The parameters of the optimized Gabor filters, (U', V') , are identified and can then be used to estimate the radial strain based on a non-tracking approach. The algorithm developed here was able to correctly distinguish between physiological and pathological strain data with 95% accuracy. This is a pilot study and is only the first step of a much larger research project. There is still much work to be done to complete this algorithm. Further testing is required accompanied by detailed statistical analyses.

8. FUTURE WORKS

8.1 Viability

Assessing this algorithm was a great obstacle and not all of the hurdles were overcome. As previously mentioned, there is no gold standard for assessing myocardial strain. Although ultrasound can measure myocardial strain, there are a number of factors that render it unsuitable, at least at this point in time, to be able to assess the results from a tagged-MR methodology. In this study, it was attempted to use the software Segment to analyze synthetic images as well as images from the physiological control group because this program has been used and validated in a variety of peer-reviewed publications. Perhaps with further study, the problems encountered with Segment could be resolved.

Another method would be to create customized synthetic data. In this instance, the ground truth values should be known. Alternatively, the current synthetic images could be analyzed using a more direct method for strain analysis that has already been validated such as B-splines or non-rigid registration. If another method can be used to establish the "knowns" for the synthetic data, then that data set can be used to validate this algorithm. A second benefit is that the synthetic data set can also be used to calibrate the algorithm. Qian *et al.* mentioned in their 2008 publication, [22], that the non-tracking strain estimator most accurately calculates the rotation angle, but consistently over estimates the circumferential strain while underestimating the radial strain. Using a large set of synthetic data with known strains and deformations would help with the development of correction factors to account for this over- and underestimations. Potentially, improvements in tag refinement and image quality might help reduce the estimation errors as well.

Given that this was a pilot study, only a basic statistical analysis was necessary to show that the algorithm can differentiate between physiological and pathological data. Now that this stage has been completed, it will be necessary to complete an in-depth statistical analysis.

It would also be fruitful display the benefits of the chosen methodologies over other comparable methods. Barajas *et al.* applied the concept of the Gabor filter bank to MR tagging.^[26] This idea modifies the implementation of the HARP algorithm by analyzing a variety of frequencies that might contain information about local deformations. By analyzing a greater number of frequencies, a more accurate Angle image can be developed. More accurate Angle images represent more precise deformation information. Although generating these Angle images was a means to estimating more accurate strain values. This is a crucial step for the success of the algorithm. Monitoring the Angle images and comparing them to standard methods and previous versions of this algorithm will show the progress of the algorithm over time and superiority of the methodology.

8.2 Improvements

Currently the algorithm assumes a uniform orientation of all input images when the contour is segmented into the 17-segment model. This should be modified so that user-input can redefine the orientation of the image. This issue was not addressed in this pilot study because the current object is essentially "proof-of-concept". Although this is critical in providing an accurate and detailed assessment of myocardial strain, it is not critical in proving that the algorithm can assess myocardial strain and differentiate between physiological and pathological data.

There are several aspects of the algorithm that can potentially be improved upon. One such aspect is the filtering of the input images. These inputs must be filtered to eliminate erroneous spikes in the Angle Images caused by non-homogenous proton densities in the contractile tissue. Currently this is done by smoothing the image with a very small low-pass Gaussian filter and then increasing the contrast using an unsharp mask. This unsharp mask creates a low-pass filtered image of the smoothed image and then subtracts it to create the sharpened image. However, it could be beneficial to create the low-pass filtered image of the original input image, then subtract that from the smoothed image.

As the algorithm was developed, consideration was given to future improvements. Some features that will be needed in the future have been planned out and some of them are in various stages of implementation. For example, when choosing the label of the slice to be analyzed the user can choose apical. A .mat file has been included for the apical label to load just like the files corresponding to the labels "Basal" and "Mid-Cavity". It was simpler to develop, debug, and test the algorithm using the basal and mid regions because they both have six segments according to the 17-segment model. In the future, the implementation of the "Apical" label as well as including an option to analyze the apex itself will need to be finished.

Initially it was intended that the algorithm would calculate the radial and circumferential strains. The infrastructure to calculate this data and to handle passing it along from one step to the next has already been built. However, the circumferential strain is not currently working as expected. After several attempts and much consideration regarding the primary objective, it was decided that the radial strain would suffice for this work.

Another improvement could be refining the tag lines before processing. As discussed in the Sect. 2.2, by computing the Fourier expansion coefficients and adding additional harmonics to Fourier spectrum, the resolution of the tag lines can increase significantly. This would be especially beneficial when analyzing images with later trigger times. Refining the tag lines reduces the contrast-to-noise ratio (CNR) and could improve the strain estimation. Given the unique method of strain estimation in this algorithm, it would be very interesting to see how this tag line refinement method affects the algorithm.

Sect. ?? explored the parameters used to create the bank of Gabor filters. While processing various datasets, it was observed that some images, when viewed in the Fourier domain, have the harmonic peaks that are skewed or stretched along a particular direction.

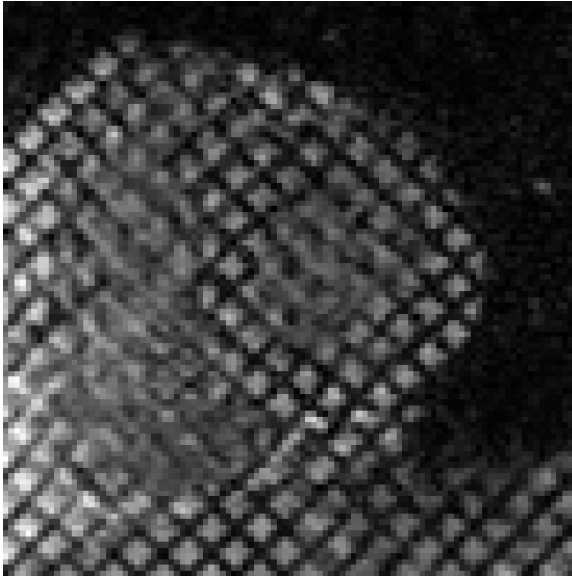
Fig. 8.1 shows two very similar images with the same anatomical and tag line orientations. Figures 8.1c and 8.1d show that even similar images can have significant variations with regard to the ranges of frequencies that could possibly contain relevant motion information. Further research is needed to determine if creating custom ranges for the Gabor filter banks is beneficial or if the information that falls outside of the bank range for ψ_k is even relevant.

Initially, the code was designed to select the points-of-interest so that they are spread out uniformly according to an arc length determined by the required number of points in each segment. This was accomplished using a program called `interparc`^[42] (Version 1.3) published by John D’Errico on the MathWorks FileExchange. This is a very nice approach because there will be a uniform distribution of the points along the circumference of the contours as they become larger and larger between the endocardium and the epicardium. The implementation is relatively straightforward as well. The two contours are sampled at the default arc length of $\pi/(10 * \text{number of AHA segments})$ which yields points that fall upon straight rays emanating from the centroid. Then points are linearly interpolated between those two points. A disadvantage of this method, and also the method use in the algorithm, is that there is a decreasing gradient of the density of POIs from the endocardium to the epicardium as can be seen in Fig. 8.2. The reason this was replaced in the algorithm is that for reasons yet to be determined, the function began to produce inconsistent and unusable results. Often, most of the POIs were concentrated inside the LV cavity and sometimes even fell outside of the ROI. While this did not happen with every image, it happened enough that it had to be replaced. Further work might should be able to resolve this issue.

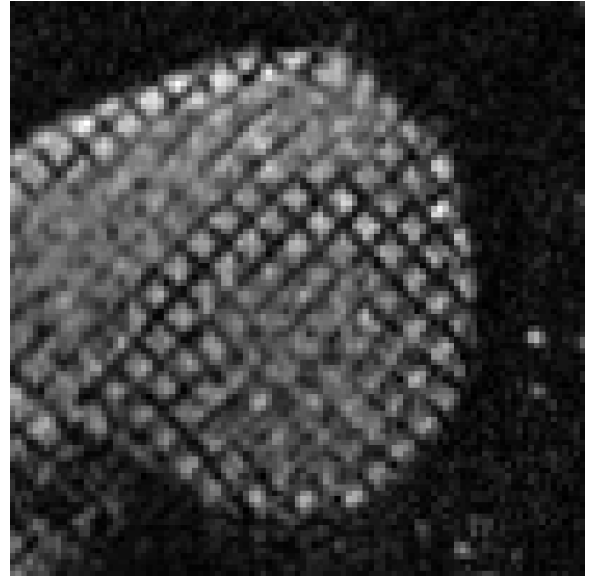
A better approach, however, would be to maintain the density of the points-of-interest throughout the entire myocardium. That could be done by deriving a relationship between the length of the perimeter of the contour and the number of points along said contour in order to calculate the arc length necessary to maintain the point density established along the endocardial contour. Maintaining a set number of rows that are evenly, although non-uniformly, spaced between the two contours versus using a grid of points, makes it easier to evaluate a strain gradient across myocardium.

Further development of an algorithm such as this one can help make MR-tagging a clinically viable diagnostic tool. As previously stated, the post-processing required to obtain useful information from tMRI is so extensive and cumbersome that it is not used clinically. Developing a semi-automated segmentation method would make such an algorithm significantly more robust by more consistent results that are user-independent. Given the nature of the tags in tMRI, it would be more practical to use texture analysis coupled with active contours to identify the myocardium.

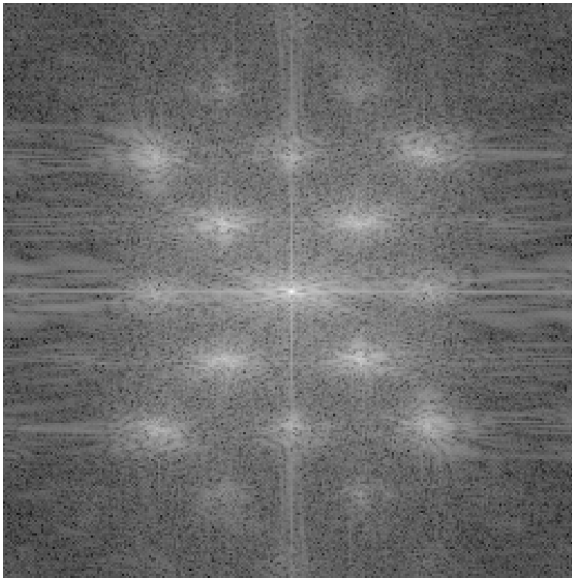
The current algorithm provides results for 2D images. It is not equipped to evaluate the entire heart all at once. It would be beneficial to develop the algorithm to work with a 3-dimensional image stack. This would allow for a more detailed localization of the strain. Because the relationship between myocardial strain and CRT response has not been researched, there is a possibility that location, in addition to the amount of strain, might play an important role in predicting CRT response. This could also reduce the personnel workload by automating the processing of multiple slice sequences.



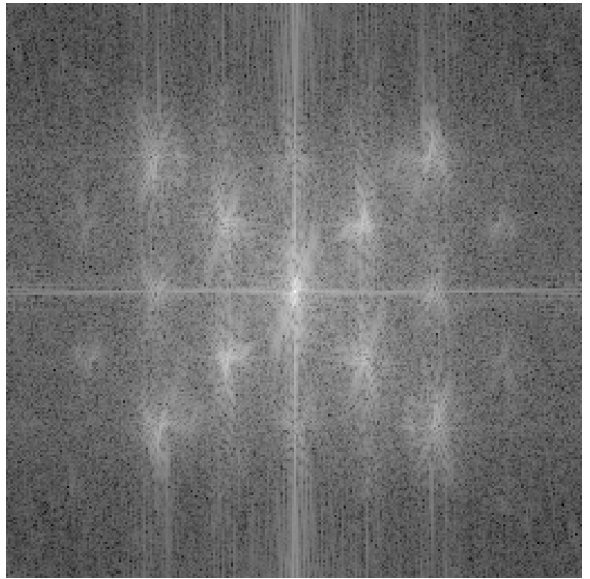
(a)



(b)



(c)



(d)

Figure 8.1: Differences in the shape of the harmonic peaks, ψ_k , even between similar images. a,b) First image in a cine sequence of a healthy patient. c,d) The 2D Fourier transforms of the original images in a) and b) respectively.

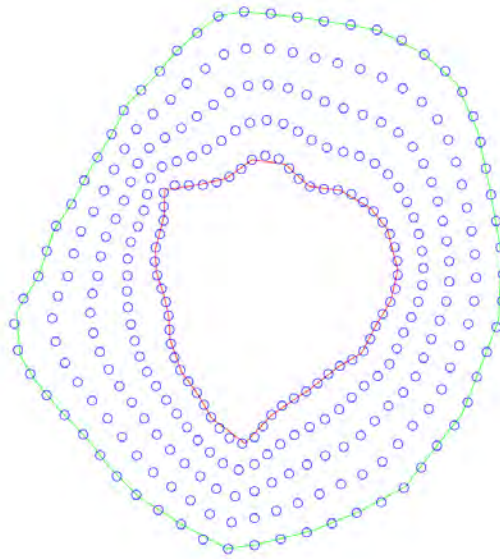


Figure 8.2: Points-of-interest identified by interparc.m

Finally, a graphical user interface, or GUI, should be developed if this is to be used as a research tool. Currently files are called from the commandline, which works sufficiently well in the initial stages of development. However, a GUI would streamline the working environment.

8.3 Future Use

The goal of this thesis is to design an algorithm that can assess and differentiate between physiological strain and pathological strain. Once this algorithm has been perfected, it will be used to analyze data from patients suffering from left ventricular mechanical dyssynchrony. The hope is to be able to use these results to identify indicators that can be used to diagnose LV dyssynchrony and predict the patients' response to CRT. If this research proves fruitful, it could potentially save time, suffering, and money for patients who more than would not benefit from CRT.

REFERENCES

- [1] S. Iyengar and W. T. Abraham, “Cardiac resynchronization therapy,” *Circulation*, vol. 112, pp. e236–e237, 2005.
- [2] N. A. Chatterjee and J. P. Signh, “Cardiac resynchronization therapy: past, present, and future,” *Heart Failure Clinics*, vol. 11, no. 2, pp. 287–303, 2015.
- [3] J. S. Gottdiener, “Review - the pathology of ventricular dyssynchrony and the role of cardiac resynchronization therapy.” [Online]. Available: <http://www.medscape.org/viewarticle/493887>
- [4] R. K. Sung and E. Foster, “Assessment of systolic dyssynchrony for cardiac resynchronization therapy is not clinically useful,” *Circulation*, vol. 123, no. 6, pp. 656–663, 2011.
- [5] N. M. Hawkins, M. C. Petrie, M. R. MacDonald, K. J. Hogg, and J. J. McMurray, “Selecting patients for cardiac resynchronization therapy: electrical or mechanical dyssynchrony?” *European Heart Journal*, vol. 27, no. 11, pp. 1270–1281, 2006. [Online]. Available: <http://eurheartj.oxfordjournals.org/ehj/27/11/1270.full.pdf><http://eurheartj.oxfordjournals.org/content/ehj/27/11/1270.full.pdf>
- [6] S. Ghioa, C. Constantina, C. Klersyb, A. Serioa, A. Fontanaa, C. Campanaa, and L. Tavazzia, “Interventricular and intraventricular dyssynchrony are common in heart failure patients, regardless of qrs duration,” *European Heart Journal*, vol. 25, pp. 571–578, 2004.
- [7] V. Delgado and J. J. Bax, “Assessment of systolic dyssynchrony for cardiac resynchronization therapy is clinically useful,” *Circulation*, vol. 123, no. 6, pp. 640–655, 2011.
- [8] J. J. M. Westenberg, H. J. Lamb, R. J. v. d. Geest, G. B. Bleeker, E. R. Holman, M. J. Schalij, A. d. Roos, E. E. v. d. Wall, J. H. C. Reiber, and J. J. Bax, “Assessment of left ventricular dyssynchrony in patients with conduction delay and idiopathic dilated cardiomyopathy,” *Journal of the American College of Cardiology*, vol. 47, no. 10, pp. 2042–2048, 2006.
- [9] E. A. Zerhouni, D. M. Parish, W. J. Rogers, A. Yang, and E. P. Shapiro, “Human heart: tagging with mr imaging—a method for noninvasive assessment of myocardial motion,” *Radiology*, vol. 169, no. 1, pp. 59–63, 1988. [Online]. Available: <http://pubs.rsna.org/doi/abs/10.1148/radiology.169.1.3420283>
- [10] E.-S. H. Ibrahim, “Myocardial tagging by cardiovascular magnetic resonance: Evolution of techniques—pulse sequences, analysis algorithms, and applications,” *Journal of Cardiovascular Magnetic Resonance*, vol. 13, pp. 1–40, 2011. [Online]. Available: <http://www.jcmr-online.com/content/13/1/36>

- [11] L. Axel and L. Dougherty, “Mr imaging of motion with spatial modulation of magnetization,” *Radiology*, vol. 171, no. 3, pp. 841–845, 1989. [Online]. Available: <http://pubs.rsna.org/doi/abs/10.1148/radiology.171.3.2717762>
- [12] N. F. Osman, W. S. Kerwin, E. R. McVeigh, and J. L. Prince, “Cardiac motion tracking using cine harmonic phase (harp) magnetic resonance imaging,” *Magnetic Resonance in Medicine*, vol. 42, no. 6, pp. 1048–1060, 1999.
- [13] V. C. Aymeric Histace, Christine Cavaro-Menard and M. Menard, “Analysis of tagged cardiac mri sequences,” in *Functional Imaging and Modeling of the Heart*, A. F. Frangi, P. I. Radeva, A. Santos, and M. Hernandez, Eds. Springer, 2005, Conference Proceedings, pp. 404–413.
- [14] N. F. Osman and J. L. Prince, “On the design of the bandpass filters in harmonic phase mri,” in *Image Processing, 2000. Proceedings. 2000 International Conference on*, vol. 1, 2000, Conference Proceedings, pp. 625–628 vol.1. [Online]. Available: <http://ieeexplore.ieee.org/ielx5/7221/19490/00901036.pdf?tp=&arnumber=901036&isnumber=19490>
- [15] V. Parthasarathy, M. Ness-Aiver, and J. L. Prince, “Dynamic range of harmonic phase magnetic resonance imaging (harp-mri),” in *Biomedical Imaging: Nano to Macro, 2004. IEEE International Symposium on*, 2004, Conference Proceedings, pp. 884–887 Vol. 1.
- [16] P. Li, J. L. Prince, J. A. C. Lima, and N. F. Osman, “Fast tracking of cardiac motion using 3d-harp,” *Biomedical Engineering, IEEE Transactions on*, vol. 52, no. 8, pp. 1425–1435, 2005. [Online]. Available: <http://ieeexplore.ieee.org/ielx5/10/31465/01463331.pdf?tp=&arnumber=1463331&isnumber=31465>
- [17] N. F. Osman and J. L. Prince, “Visualizing myocardial function using harp mri,” *Phys. Med. Biol.*, vol. 45, pp. 1665–1682, 2000.
- [18] —, “Regenerating mr tagged images using harmonic phase (harp) methods,” *Biomedical Engineering, IEEE Transactions on*, vol. 51, no. 8, pp. 1428–1433, 2004. [Online]. Available: <http://ieeexplore.ieee.org/ielx5/10/29163/01315866.pdf?tp=&arnumber=1315866&isnumber=29163>
- [19] Z. Qian, Q. Liu, D. N. Metaxas, and L. Axel, “Identifying regional cardiac abnormalities from myocardial strains using nontracking-based strain estimation and spatio-temporal tensor analysis,” *IEEE Transactions on Medical Imaging*, vol. 30, no. 12, pp. 2017–2029, 2011.
- [20] A. M. Khalifa, A. B. M. Youssef, and N. F. Osman, “Improved harmonic phase (harp) method for motion tracking a tagged cardiac mr images,” in *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, 2005, Conference Proceedings, pp. 4298–4301.
- [21] E. Bayram, C. A. Hamilton, and W. G. Hundley, “Mr tagging from a signal process-

- ing perspective,” in *23rd Annual EMBS International Conference of the IEEE*, vol. 3. IEEE, 2001, Conference Proceedings, pp. 2488–2491.
- [22] Z. Qian, D. Metaxas, and L. Axel, “Non-tracking-based 2d strain estimation in tagged mri,” *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 528 – 531, 2008.
- [23] K. Z. Abd-Elmoniem, S. Sampath, N. F. Osman, and J. L. Prince, “Real-time monitoring of cardiac regional function using fastharp mri and region-of-interest reconstruction,” *Biomedical Engineering, IEEE Transactions on*, vol. 54, no. 9, pp. 1650–1656, 2007. [Online]. Available: <http://ieeexplore.ieee.org/ielx5/10/4291649/04291650.pdf?tp=&arnumber=4291650&isnumber=4291649>
- [24] M. A. AlAttar, N. F. Osman, and A. S. Fahmy, “Segmentation of left ventricle in cardiac mri images using adaptive multi-seeded region growing,” in *Biomedical Engineering Conference (CIBEC), 2010 5th Cairo International*, 2010, Conference Proceedings, pp. 25–28. [Online]. Available: <http://ieeexplore.ieee.org/ielx5/5710819/5716039/05716057.pdf?tp=&arnumber=5716057&isnumber=5716039>
- [25] N. F. Osman, E. R. McVeigh, and J. L. Prince, “Imaging heart motion using harmonic phase mri,” *Medical Imaging, IEEE Transactions on*, vol. 19, no. 3, pp. 186–202, 2000. [Online]. Available: <http://ieeexplore.ieee.org/ielx5/42/18324/00845177.pdf?tp=&arnumber=845177&isnumber=18324>
- [26] J. Barajas, J. Garcia-Barnés, F. Carreras, S. Pujadas, and P. Radeva, “Angle images using gabor filters in cardiac tagged mri,” *Proceedings of the 2005 conference on Artificial Intelligence Research and Development*, no. IOS Press, pp. 107–114, 2005.
- [27] S. Ryf, J. Tsao, J. Schwitter, A. Stuessi, and P. Boesiger, “Peak-combination harp: A method to correct for phase errors in harp,” *Journal of Magnetic Resonance Imaging*, vol. 20, no. 5, pp. 874–880, 2004.
- [28] M. Bilgen, “Harmonic phase interference for the detection of tag line crossings and beyond in homogeneous strain analysis of cardiac tagged mri data,” *Australasian Physical & Engineering Sciences in Medicine*, vol. 33, no. 4, pp. 357–366, 2010.
- [29] Z. Qian, A. Montillo, D. N. Metaxas, and L. Axel, “Segmenting cardiac mri tagging lines using galbor filter banks,” *Proceedings of the 25th Annual International Conference of the IEEE EMBS*, vol. 1, pp. 630–633, 2003.
- [30] Z. Qian, X. Huang, D. N. Metaxas, and L. Axel, “Robust segmentation of 4d cardiac mri-tagged images via spatio-temporal propagation,” *Proceedings of SPIE: Medical Imaging*, pp. 580–591, 2005.
- [31] J. G. Daugman, “Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters,” *Journal of the Optical Society of America*, vol. 2, no. 7, pp. 1160–1169, 1985.

- [32] N. F. Osman and J. L. Prince, "Angle images for measuring heart motion from tagged mri," *Proceedings of the International Conference on Image Processing*, vol. 1, pp. 704–708, 1998.
- [33] M. D. Cerqueira, N. J. Weissman, V. Dilsizian, A. K. Jacobs, S. Kaul, W. K. Laskey, D. J. Pennell, J. A. Rumberger, T. Ryan, and M. S. Verani, "Standardized myocardial segmentation and nomenclature for tomographic imaging of the heart," *Circulation*, vol. 105, no. 4, pp. 539–542, 2002.
- [34] E. Heiberg, J. Sjogren, M. Ugander, M. Carlsson, H. Engblom, and H. Arheden, "Design and validation of segment - a freely available software for cardiovascular image analysis," *BioMed Central Medical Imaging*, vol. 10, no. 1, pp. 1–13, 2010. [Online]. Available: <http://www.biomedcentral.com/1471-2342/10/1>
- [35] N. F. Osman and J. L. Prince, "Visualizing myocardial function using harp mri," *Physics in Medicine and Biology*, vol. 45, no. 6, pp. 1665–1682, 2000.
- [36] D. N. Metaxas, L. Axel, Z. Qian, and X. Huang, "A segmentation and tracking system for 4d cardiac tagged mr images," *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, pp. 1541 – 1544, 2006.
- [37] Z. Qian, W.-N. Lee, E. E. Konofagou, D. N. Metaxas, and L. Axel, "Ultrasound myocardial elastography and registered 3d tagged mri: Quantitative strain comparison," *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2007 Lecture Notes in Computer Science*, vol. 10, pp. 800–808, 2007.
- [38] W. M. Lai, D. Rubin, and E. Krempl, *Introduction to Continuum Mechanics*, 3rd ed. Oxford: Pergamon Press Ltd., 1993.
- [39] T. Binder and G. Goliash, "Left ventricular function," pp. 30–31, 2015.
- [40] C. Steele and P. Pinsky, "Mechanics of materials: Plane strain." [Online]. Available: http://www.efunda.com/formulae/solid_mechanics/mat_mechanics/images/StrainTransform.gif
- [41] M. Sheppard, "allfitdist," p. Fit all valid parametric probability distributions to data, 2012. [Online]. Available: <https://www.mathworks.com/matlabcentral/fileexchange/34943-fit-all-valid-parametric-probability-distributions-to-data/content/allfitdist.m>
- [42] J. D'Errico, "interparc," p. Distance based interpolation along a general curve in space, 2012.

A. MATLAB CODE

A.1 Segment.m

```
1 function Name = Segment(varargin)
2 % Segment is the parent file for segmenting the desired files and saving
3 % the results in a .mat file whose name is returned.
4 % Segment allows the user to select a DICOM file via an interactive GUI.
5 % The image is checked to make sure the format is correct before
6 % proceeding. The slice label is selected through a dialog. The first
7 % harmonic peaks are interactively identified before guiding the user
8 % through the manual segmentation of the endocardium and epicardium.
9
10 id = 'MATLAB:colon:nonIntegerIndex';
11 warning('off',id);
12
13 %% Load the files and generate the image stack
14 if isempty(varargin)
15     [fileName, pathName] = uigetfile('*.dcm','Select a DICOM file');
16     name = fullfile(pathName,fileName);
17 else
18     name = varargin{1};
19     [pth, fileName, ~] = fileparts(name);
20     pathName = strcat(pth,'\');
21 end
22 dinfo = dicominfo(name);
23 time = dinfo.TriggerTime;
24 Im = imageCheck(dicomread(dinfo));
25
26 %% Create name for output file
27 [~,filename,~] = fileparts(name);
28 Name = strcat(pathName,filename,'_segmented.mat');
29 name = filename;
30
31 %% Select the label - 'base', 'mid', 'apical'
32 label = chooseLabelDialog;
33
34 if strcmp(label,'Basal')
35     load('base.mat')
36 elseif strcmp(label,'Mid-Cavity')
37     load('mid.mat')
38 elseif strcmp(label,'Apical')
39     load('apical.mat')
40 end
41
```

```

42 %% Set key variables
43 prompt = {'nPoints'; 'rows'};
44 dlg_title = 'Alter Default Variables';
45 num_lines = 1;
46 defaultans = {'10', '5'};
47 answer = inputdlg(prompt, dlg_title, num_lines, defaultans);
48 nPoints = str2num(answer{1}); rows = str2num(answer{2});
49
50 %% Fundamental Harmonic Peaks
51 % % Define the variables to be used in generating the filter bank
52 % Identify the central frequencies of the first harmonic peaks
53 [omega, angles] = FourierPeaks(Im);
54
55 %% Select the region-of-interest and crop the stack
56 [image, rect] = ROI(Im);
57
58 %% Image Segmentation
59 [segmentLabels, contourMask, test_points, sector, members, centroid] ...
    = Segmentation(image, AHA, nPoints, rows, sectionAngles, ...
    segmentLabels); %segm, mx); %, status);
60
61 %% Save Output
62 save(Name, 'name', 'time', 'Im', 'rect', 'omega', 'angles', ...
    'segmentLabels', 'contourMask', 'test_points', 'sector', ...
    'members', 'centroid');
63
64 end

```

A.1.1 imageCheck.m

```

1 function image = imageCheck(image)
2 % imageCheck ensures that the input image from the DICOM file is in
3 % the correct format for processing. It will convert 1- and 3-layer
4 % non-double class images to grayscale images of class double.
5
6 if ~isa(image, 'double') && size(image, 3) == 3
7     image = double(rgb2gray(image));
8 elseif ~isa(image, 'double') && (size(image, 3) == 1)
9     image = double(image);
10 else
11     error('IMAGE.TYPE::', 'Invalid Image: Image should be either RGB ...
        or a 2D matrix')
12 end
13
14 end

```

A.1.2 chooseLabelDialog.m

```
1 function label = chooseLabelDialog
2 % chooseLabelDialog is an interactive dialog that requires the user to
3 % identify the long-axis region that the slice represents.
4 % There are three options: Basal
5 %                               Mid-Cavity
6 %                               Apical
7
8     d = dialog('Position',[300 300 250 150],'Name','Select One');
9     txt = uicontrol('Parent',d,...
10                  'Style','text',...
11                  'Position',[20 80 210 40],...
12                  'String','Select slice level');
13
14     popup = uicontrol('Parent',d,...
15                    'Style','popup',...
16                    'Position',[75 70 100 25],...
17                    'String',{'Basal';'Mid-Cavity';'Apical'},.....
18                    'Callback',@popup_callback);
19
20     btn = uicontrol('Parent',d,...
21                  'Position',[89 20 70 25],...
22                  'String','Select',...
23                  'Callback','delete(gcf)');
24
25     label = 'Basal';
26
27     % Wait for d to close before running to completion
28     uiwait(d);
29
30     function popup_callback(popup,callbackdata)
31         idx = popup.Value;
32         popup_items = popup.String;
33         label = char(popup_items(idx,:));
34     end
35
36 end
```

A.1.3 predefined_segmentation_variables.m

```
1 label = 1;
```



```

2 AHA = 6;
3 segmentNames = {'Basal Anterior', 'Basal Anteroseptal', 'Basal ...
    Inferoseptal', 'Basal Inferior', 'Basal Inferolateral', 'Basal ...
    Anterolateral'};
4 sectionAngles = [pi/3 2*pi/3 pi 4*pi/3 5*pi/3 2*pi pi/3];
5 segmentLabels = {'basalAnterior', {}, 'basalAnteroseptal', {}, ...
    'basalInferoseptal', {}, 'basalInferior', {}, ...
    'basalInferolateral', {}, 'basalAnterolateral', {}};
6 save('base.mat', 'label', 'AHA', 'segmentNames', 'sectionAngles', ...
    'segmentLabels')
7
8 label = 2;
9 AHA = 6;
10 segment_names_mid = {'Mid Anterior', 'Mid Anteroseptal', 'Mid ...
    Inferoseptal', 'Mid Inferior', 'Mid Inferolateral', 'Mid ...
    Anterolateral'};
11 sectionAngles = [pi/3 2*pi/3 pi 4*pi/3 5*pi/3 2*pi pi/3];
12 segmentLabels = {'midAnterior', {}, 'midAnteroseptal', {}, ...
    'midInferoseptal', {}, 'midInferior', {}, 'midInferolateral', {}, ...
    'midAnterolateral', {}};
13 save('mid.mat', 'label', 'AHA', 'segmentNames', 'sectionAngles', ...
    'segmentLabels')
14
15 label = 3;
16 AHA = 4;
17 segment_names_apical = {'Apical Anterior', 'Apical Septal', 'Apical ...
    Inferior', 'Apical Lateral'};
18 sectionAngles = [pi/4 3*pi/4 5*pi/4 7*pi/4 pi/4];
19 segmentLabels = {'midAnterior', {}, 'midAnteroseptal', {}, ...
    'midInferoseptal', {}, 'midInferior', {}, 'midInferolateral', {}, ...
    'midAnterolateral', {}};
20 save('apical.mat', 'label', 'AHA', 'segmentNames', 'sectionAngles', ...
    'segmentLabels')

```

FourierPeaks.m

```

1 function [omega, angles] = FourierPeaks(image)
2 % FourierPeaks is a semi-automated approach to identify the central
3 % frequencies of desired spectral harmonic peaks in the 2D Fourier
4 % space. It also returns the angles of the peaks about the center.
5 % Both of these values are used in creating the Gabor filter bank.
6
7 % % Implementing the Gabor Filter Bank
8 % Find the first harmonic peak for m = [-1 +1]
9 log_F = log(1 + fftshift(fft2(image)));
10 s = size(log_F);

```

```

11
12 figure('Visible','off'), imshow(real(log_F),[])
13 set(gcf, 'Position', get(0,'Screensize')); % Maximize the figure
14 title({'Fourier Transform of the input image';...
15     'Please select the first harmonic peak for both the positive and ...
16     'negative m';...
17     'Double click the second point to submit'})
18 set(gcf, 'Visible', 'on')
19 [pts_x, pts_y] = getpts(gcf);
20
21 % Creates a window around the selected peaks and finds the maximum,
22 % central, frequencies of those peaks
23 window1 = log_F((pts_x(1) - 20):(pts_x(1) + 20), (pts_y(1) - ...
24     20):(pts_y(1) + 20));
25
26 % % Organize omega
27 % QII and QIV = -theta      QI and QIII = +theta
28 test_x = (pts_x>round(s(2)/2)); % Is x positive?
29 test_y = (pts_y<round(s(1)/2)); % Is y positive?
30 if (length(pts_x)==1)
31     omega = max(max(window1));
32     angles = atan(-(pts_y(1) - round(s(1)/2))/(pts_x(1) - ...
33         round(s(2)/2)));
34 close(gcf)
35 return
36 else
37     window2 = log_F((pts_x(2) - 20):(pts_x(2) + 20), (pts_y(2) - ...
38         20):(pts_y(2) + 20));
39     A1 = atan(-(pts_y(1) - round(s(1)/2))/(pts_x(1) - round(s(2)/2)));
40     A2 = atan(-(pts_y(2) - round(s(1)/2))/(pts_x(2) - round(s(2)/2)));
41     if ((test_x(1)==1 && test_y(1)==1) || (test_x(1)==0 && test_y(1)==0))
42         omega = [max(max(window1)) max(max(window2))];
43         angles = [A1 A2];
44     elseif ((test_x(1)==1 && test_y(1)==0) || (test_x(1)==0 && ...
45         test_y(1)==1))
46         omega = [max(max(window2)) max(max(window1))];
47         angles = [A2 A1];
48     end
49 end
50 close(gcf)
51
52 end

```

A.1.4 ROI.m

```

1 function varargout = ROI(varargin)
2 % ROI helps the user specify the region-of-interest. The first input
3 % must be an image. If the region-of-interest has already been
4 % specified then it can be entered as the second input argument. If
5 % only one output is specified, then the cropped image will be
6 % returned. If two are specified, the second output will be 'rect'
7 % which was returned from imcrop and has been rounded.
8 %   rect = [xmin ymin width height];
9
10 image = varargin{1};
11 if nargin==2
12     rect = varargin{2};
13     varargout{1} = image(rect(2):rect(2) + rect(4), rect(1):rect(1) + ...
14         rect(3));
15 elseif nargin==1
16     figure('Visible','off');
17     iptsetpref('ImshowAxesVisible','off');
18     imshow(image,[])
19     title({'Please select a region of interest centered about the ...
20         left ventricular cavity.';...
21         'Be sure to leave a reasonable border around the myocardium ...
22         and double click when finished.'})
23     % Maximize the figure and make it visible
24     set(gcf, 'Position', get(0,'Screensize'), 'Visible','on');
25
26     [-, rect] = imcrop;
27     rect = round(rect);
28     varargout{1} = image(rect(2):rect(2) + rect(4), rect(1):rect(1) + ...
29         rect(3));
30 end
31
32 if nargout==2
33     varargout{2} = rect;
34 end
35
36 close(gcf)
37
38 end

```

A.1.5 Segmentation.m

```

1 function [segmentLabels, contourMask, test_points, sector, members, ...
2     centroid] = Segmentation(image, AHA, nPoints, rows, sectionAngles, ...
3     segmentLabels)
4 % Segmentation finds the contourMasks and the centroid of the myocardium,
5 % resamples the contourMasks, and assigns the PoIs to their respective

```

```

4 % segments according to the AHA 17-segment model.
5
6 [contourMask, ~, ~] = contours(image);
7 centroid = Centroid(contourMask);
8 [test_points, sector, segmentLabels] = testPoints(centroid, AHA, ...
    nPoints, rows, sectionAngles, segmentLabels, contourMask);
9
10 m = length(sectionAngles) - 1;
11 members = arrayfun(@(x)find(sector == x), 1:m, 'uniform', 0);
12
13 end

```

contours.m

```

1 function [contourMask, endo_cont, epi_cont] = contours(image)%, mx)
2 % contours.m does 1 of 2 things depending on the number of inputs. If
3 % only the image is given, then the user will be prompted to select
4 % points along the endocardial and epicardial contours. The more points
5 % selected, the better the contours will be. If the contours have
6 % already been identified and given as inputs, first the endocardial
7 % contour followed by the epicardial contour, then the contour masks
8 % will be created and the program terminates.
9
10 % Evaluate the number of inputs and collect the contour coordinates
11 s = size(image);
12 figure('Visible','off');
13 imshow(image,[])
14 set(gcf, 'Position', get(0,'Screensize')); % Maximize the figure
15 title({'Please select points along the endocardium first and then ...
    select points along the epicardium.';...
16     'Please start at the top, where the blue line intersects the ...
    contour';...
17     'Double click the final point to submit'})
18 hold on
19 plot([0.5*s(2) 0.5*s(2)], [0 s(1)], 'b'), hold off, set(gcf, 'Visible', 'on')
20
21 [endo_contx, endo_conty] = getpts;
22 endo_cont = [endo_contx, endo_conty];
23 % Ensure the entire contour was selected
24 prompt = {'Did you finish the endocardial contour? y/n'};
25 dlg_title = 'Endocardial Contour';
26 num_lines = 1;
27 answer = inputdlg(prompt,dlg_title,num_lines);
28 test_check = strcmp(answer,'y');
29 while ~test_check
30     [endo_contx, endo_conty] = getpts;

```

```

31     endo_cont = [endo_contx, endo_conty];
32     answer = inputdlg(prompt,dlg_title,num_lines);
33     test_check = strcmp(answer,'y');
34 end
35
36 [epi_contx, epi_conty] = getpts;
37 epi_cont = [epi_contx, epi_conty];
38 % Ensure the entire contour was selected
39 prompt = {'Did you finish the epicardial contour? y/n'};
40 dlg_title = 'Epicardial Contour';
41 num_lines = 1;
42 answer = inputdlg(prompt,dlg_title,num_lines);
43 test_check = strcmp(answer,'y');
44 while ~test_check
45     [epi_contx, epi_conty] = getpts;
46     epi_cont = [epi_contx, epi_conty];
47     answer = inputdlg(prompt,dlg_title,num_lines);
48     test_check = strcmp(answer,'y');
49 end
50
51 % Create the contour masks
52 epicardium = poly2mask(epi_cont(:,1),epi_cont(:,2),s(1),s(2));
53 endocardium = poly2mask(endo_cont(:,1),endo_cont(:,2),s(1),s(2));
54 contourMask = epicardium - endocardium;
55
56 close(gcf)
57
58 end

```

Centroid.m

```

1 function centroid = Centroid(contourMask)
2 % This program uses the polygon masks created in contours.m and finds
3 % the centroids of the myocardial contour. If only one input and
4 % output are specified, then only the myocardial contour's centroid
5 % will be found. If 3 inputs and outputs are specified, then the
6 % myocardial, endocardial, and epicardial centroids will be returned.
7
8 CC_cont = bwconncomp(contourMask);
9 centroid = cell2mat(struct2cell(regionprops(CC_cont,'Centroid')));
10
11 end

```

testPoints.m

```
1 function [strain_pts, sector, combined, segmentLabels] = ...
    testPoints(centroid, AHA, nPoints, rows, sectionAngles, ...
        segmentLabels, contourMask)%epi_cont, endo_cont, AHA, nPoints, ...
        rows, sectionAngles, segmentLabels, contourMask)
2 % Define the coordinates of the points between the contours to be
3 % analyzed. There will be nPoints number of points in AHA number of
4 % sections depending on the slice location, i.e. base, mid, or apical.
5 % The contours will be resampled at uniform arc lengths.
6
7 % Resample the contours at uniform dist (arc length)
8 % There will be AHA number of sections, nPoints points per section
9 noPts = AHA*nPoints;
10
11 % Separate the endocardial and epicardial contours
12 [b,~,N] = bwboundaries(contourMask);
13 for k=1:length(b),
14     boundary = b{k};
15     if(k > N)
16         endo_C = [boundary(:,2), boundary(:,1)];
17     else
18         epi_C = [boundary(:,2), boundary(:,1)];
19     end
20 end
21
22 %% Reorder the contours
23 Endo = sortContour(endo_C, centroid, sectionAngles);
24 Epi = sortContour(epi_C, centroid, sectionAngles);
25
26 % Resample the contours
27 sampledEndo = resample(Endo, noPts);
28 sampledEpi = resample(Epi, noPts);
29
30 % Create testing points between the resampled contour points
31 combined = []; strain_pts = []; sector = zeros([noPts*rows 1]);
32 for n = 1:length(sampledEndo)
33     start = length(strain_pts) + 1;
34     % Linearly interpolate between endocardial and epicardial points
35     x = linspace(sampledEndo(n,1), sampledEpi(n,1), rows);
36     y = linspace(sampledEndo(n,2), sampledEpi(n,2), rows);
37     strain_pts = [strain_pts; x' y'];
38     stop = length(strain_pts);
39     [angNum, segmentLabels] = SegmentFinder(sampledEndo(n,:), ...
        centroid, strain_pts(start:stop,:), sectionAngles, segmentLabels);
40     sector(start:stop,1) = angNum(1);
41 end
42 strain_pts = round(strain_pts);
43
```

44 end

sortContour.m

```
1 function Contour = sortContour(contour, centroid, sectionAngles)
2 % sortContour sorts the points of a contour first according to its
3 % angle (in radians) about the centroid, and then corrected by
4 % choosing the nearest point spatially with the lowest angle.
5
6 %% Sorting theta
7 % Sort the endocardial contour points by theta
8 [theta, ~] = cart2pol(contour(:,1) - centroid(1), contour(:,2) - ...
9     centroid(2));
10
11 % Correct for theta to be within the range [0, 2*pi)
12 n = find(theta >= 2*pi); theta(n) = theta(n) - 2*pi;
13 n = find(theta < 0); theta(n) = theta(n) + 2*pi;
14
15 % Sort theta in ascending order in counter-clockwise direction
16 [b, ind] = sort(theta); b = abs(b);
17 % Sort the contour points according to the order of the angles
18 sorted = contour(ind, :);
19
20 % Let the first point start in the first AHA segment
21 wrp = find(b < sectionAngles(1));
22 sorted = [sorted(max(wrp)+1:end, :); sorted(wrp(1):end, :)];
23 B = [b(max(wrp)+1:end, :); b(wrp(1):end, :)];
24
25 %% Looking for the next nearest point
26 % Sort according to minimal distance between points
27 L = length(sorted); Contour = sorted(1, :);
28 for n = 1:L-1
29     exclude = []; d = zeros(1, length(sorted)); %L-n);
30     % Measure the distance between the current pt and all others
31     for m = 1:length(sorted)
32         % Because the previous value is always erased, the next value
33         % will always be (1, :)
34         d(m) = sqrt((sorted(m,1) - sorted(1,1))^2 + (sorted(m,2) - ...
35             sorted(1,2))^2);
36         % Exclude the PoI and any duplicate
37         if isequal(d(m), 0)
38             exclude(length(exclude)+1) = m;
39         end
40     end
41     excluded = d; excluded(exclude) = [];
42     % Find the nearest points
```

```

41     if ~isempty(excluded)
42         f = find(min(excluded)==d,5);
43     end
44     % Find value with the smallest polar theta
45     [~, minT] = min(B(f));
46     Contour(n+1,:) = sorted(f(minT),:);
47     B(f(minT)) = []; sorted(1,:) = [];
48 end
49
50 end

```

resample.m

```

1 function Resampled = resample(contour, noPts)
2 % resample returns noPts number of points along the given contour
3 % spread equidistantly along the perimeter of the contour.
4
5 % Cumulative arclength along the contour's perimeter
6 t = cumsum(sqrt([0,diff(contour(:,1)')).^2] + ...
7             [0,diff(contour(:,2)')).^2]));
8
9 % Total distance = t(end)
10 tmax = t(end);
11
12 % create a piecewise linear spline for each of px and py as a dunction
13 % of the cumulative chordwise arclength.
14 splx = mkpp(t,[diff(contour(:,1))./diff(t'),contour(1:(end-1),1)]);
15 sply = mkpp(t,[diff(contour(:,2))./diff(t'),contour(1:(end-1),2)]);
16
17 % Interpolate the polygon splines, splx and sply.
18 % noPts is the number of points to generate along the contour
19 % The first and last points should be replicates
20 tint = linspace(0,tmax,noPts);
21
22 qx = ppval(splx,tint);
23 qy = ppval(sply,tint);
24 Resampled = [qx', qy'];
25
26 end

```

SegmentFinder.m


```

1 function [angNum, segmentLabels] = SegmentFinder(point, centroid, ...
    newPts, sectionAngles, segmentLabels)
2 % SegmentFinder assigns the POIs to the appropriate segment according
3 % to the AHA 17-segment model.
4
5 % Compute polar coordinates
6 [thetaPt, ~] = cart2pol(point(1) - centroid(1), point(2) - centroid(2));
7 % Correct so that theta is in (0, 2pi]
8 ind = (thetaPt<=0);
9 thetaPt(ind) = thetaPt(ind) + 2*pi;
10 m = length(sectionAngles) - 1;
11 sectionAngles = -sectionAngles;
12
13 % Identify which sector it falls into by checking theta against one
14 % sector and then the next. Should be greater than one and less than
15 % the other
16 membership = zeros([1 m]);
17 for n = 1:m
18     membership(n) = (thetaPt>=sectionAngles(n) && ...
        thetaPt<sectionAngles(n+1));
19     if (n==m)
20         if ~ (m==4)
21             membership(n) = (thetaPt>=0 && thetaPt<sectionAngles(1));
22         elseif (m==4)
23             membership(n) = ((thetaPt>=0 && thetaPt<sectionAngles(1)) ...
                || (thetaPt>=sectionAngles(n) && thetaPt<0));
24         end
25     end
26 end
27
28 angNum = find(membership~=0);
29
30 for n = 1:m
31     segmentLabels{2*n} = {[segmentLabels{2*n}(:, :); newPts]};
32 end
33
34 end

```

A.2 GaborFilterBank.m

```

1 function Name = GaborFilterBank(varargin)
2 % GaborFilterBank builds a bank of Gabor filters for data sets with
3 % either 1 or 2 tagging directions. The segmented file (*.mat) can
4 % either be passed as an input or if no inputs are provided, the
5 % it will be selected interactively. The output variables stored in

```

```

6 % the .mat file are the:
7 %   kernels1 - kernel values for the first orientation angle, omega(1)
8 %   lookup1  - lookup table with the parameters for every kernel in
9 %               kernels1
10 %   kernels2 - kernel values for the second orientation angle, omega(2)
11 %   lookup2  - lookup table with the parameters for every kernel in
12 %               kernels2
13
14 % Load the image to be processed with or w/out an input file
15 if isempty(varargin)
16     [fileName, pathName] = uigetfile('*segmented.mat','Select a ...
17         segmented file');
18     file = fullfile(pathName,fileName);
19 else
20     file = varargin{1};
21     [pathName, ~, ~] = fileparts(file);
22     pathName = strcat(pathName, '\');
23 end
24 load(file)
25 Name = strcat(pathName,name, '_GaborFilterBank.mat');
26
27 % Set key variables
28 prompt = {'m'; 'dTheta'; 'MatrixSize'; 'sigma_x coeff'; 'lambda'};
29 dlg_title = 'Alter Default Variables';
30 num_lines = 1;
31 defaultans = {'0.8:0.01:1.2', '[pi/12 pi/96]', '11', '1.5', '2'};
32 answer = inputdlg(prompt,dlg_title,num_lines,defaultans);
33 m = str2num(answer{1}); dTheta = str2num(answer{2});
34 matrixsize = str2num(answer{3}); sigx = str2num(answer{4});
35 lambda = str2num(answer{5});
36
37 % Initialize variables
38 ltheta = dTheta(1)/dTheta(2) + 1;
39 kernels = zeros([matrixsize matrixsize length(m)*ltheta]);
40 lookup = zeros([(length(m)*length(ltheta)) 4]);
41 x = round(matrixsize/2); y = x;
42 centerpoint = [x y]; counter = 0;
43
44 %% Create the filter bank
45 for o = 1:length(omega)
46     % Define the frequency range of the filter bank
47     theta = angles(o)-dTheta(1):dTheta(2):angles(o)+dTheta(1);
48     for n = 1:length(m)
49         for th = 1:length(theta)
50             counter = counter + 1;
51             % Create the Gabor kernels to be used in the filter bank
52             [kernels(:, :, counter), U_prime, V_prime] = ...
53                 GaborKernel(matrixsize, theta(th), omega(o), ...
54                     centerpoint, m(n), sigx, lambda);
55             % Generate a lookup table of the optimized parameters
56             lookup(counter, :) = [U_prime V_prime m(n) theta(th)];

```

```

54     end
55 end
56
57 if (o==1)
58     counter = 0;
59     kernels1 = kernels;
60     lookup1 = lookup;
61     theta = angles(2)-dTheta(1):dTheta(2):angles(2)+dTheta(1);
62     clear kernels lookup
63 elseif (o==2)
64     kernels2 = kernels;
65     lookup2 = lookup;
66 end
67
68 end
69
70 save(Name, 'kernels1', 'lookup1', 'kernels2', 'lookup2', 'name');
71
72 end

```

GaborKernel.m

```

1 function [kernel, varargout] = GaborKernel(matrixsize, theta, omega, ...
2     centerpoint, m, sigx, lambda)
3 % matrixsize - Size of matrix (odd values are desirable -> center point)
4 % theta      - orientation angle (in radians)
5 % omega      - circular frequency of sin part (in spatial domain)
6 % centerPoint - center point of filter [x y]
7
8 pointX = centerpoint(1);
9 pointY = centerpoint(2);
10
11 maxX = matrixsize - pointX;
12 maxY = matrixsize - pointY;
13
14 % Ensure that the matrix size is odd
15 if(mod(matrixsize, 2)==0)
16     maxX = maxX - 1;
17     maxY = maxY - 1;
18 end
19
20 [x, y] = meshgrid(-maxX :1: maxX, -maxY :1: maxY);
21 x = double(x); y = double(y);
22
23 % % Create Kernel
24 e = exp(1);

```

```

24
25 % % % R1 and R2 are actually x and y from download_Gabor.pdf
26 x_prime = x.*cos(theta(1)) + y.*sin(theta(1));
27 y_prime =-x.*sin(theta(1)) + y.*cos(theta(1));
28 U = cos(omega);
29 V = sin(omega);
30 U_prime = real((U + 1i*V) * m(1) * exp(1i * theta(1)));
31 V_prime = imag((U + 1i*V) * m(1) * exp(1i * theta(1)));
32
33 % Define sigma
34 sigy = sigx / lambda;
35 sigma = [sigx*1/sqrt(U^2+V^2) sigy*1/sqrt(U^2+V^2)];
36
37 expFactor = -1/2 * ((x_prime/sigma(1)).^2 + (y_prime/sigma(2)).^2);
38
39 gauss = 1 / (sqrt(lambda*pi)*sigma(1)*sigma(2));
40 gauss = (gauss + 1) .* e.^expFactor;
41
42 % Complex sinusoid used in generating the gabor filter
43 sinusoid = exp(-1i*2*pi.*(U_prime.*x + V_prime.*y));
44
45 % kernel = gaborReal + gaborImag*1i;
46 kernel = gauss .* sinusoid;
47 kernel = kernel'; % Set kernel(y,x); y=row; x=column;
48
49 varargout{1} = U_prime;
50 varargout{2} = V_prime;
51
52 end

```

A.3 GaborStrain.m

```

1 function Name = GaborStrain(varargin)
2 % GaborStrain uses the Gabor filter bank specified in the input to
3 % analyze the selected *.mat files and saves the results for each in a
4 % separate output file. Each input file is filtered and the optimized
5 % filter paramters and Angle Image are returned. Then the strain is
6 % calculated for the test_points for each image.
7
8 % Load the files and generate the image stack
9 if isempty(varargin)
10     [fn, pth] = uigetfile('*GaborFilterBank.mat','Select the Filter ...
11         Bank');
12     GaborFilterBank = fullfile(pth,fn);
13 else

```

```

13     GaborFilterBank = varargin{1};
14 end
15 load(GaborFilterBank)
16
17 %% Choose segmented files to be analyzed
18 if ~isempty(varargin)
19     [pth, ~, ~] = fileparts(varargin{1});
20     pth = strcat(pth, '\');
21 end
22 [fileName, pathName] = uigetfile(strcat(pth, '*segmented.mat'), ...
    'MultiSelect', 'on', 'Select the segmented files to be processed');
23
24 %% Process the selected files
25 if ~iscell(fileName)
26     l = 1;
27 elseif iscell(fileName)
28     l = length(fileName);
29 end
30 for n = 1:l
31     %     n = 2;
32     if ~iscell(fileName)
33         load(fullfile(pathName, fileName));
34         FileName = fileName;
35     else
36         load(fullfile(pathName, fileName{n}));
37         FileName = fileName{n};
38     end
39     if n == 1
40         Rect = rect;
41     end
42
43     %% Apply the Gabor filters
44     [image, optimized1, IND, contourMask, AngleImage1, Filtered1, ...
        FilteredMag1, FilteredPhase1] = Optimize(Im, contourMask, ...
        Rect, rect, kernels1, test_points, lookup1);
45     [ ~ , optimized2, ~ , ~ , AngleImage2, Filtered2, ...
        FilteredMag2, FilteredPhase2] = Optimize(Im, contourMask, ...
        Rect, rect, kernels2, test_points, lookup2);
46
47     %% Analyze the strain for values inside the myocardium
48     F = zeros(2,2,length(optimized1)); E = F; R = F;
49     for i = 1:length(IND)
50         % Calculate F
51         % % Parameters along the x-direction
52         U_maxx = optimized1(IND(i),1);          V_maxx = ...
            optimized1(IND(i),2);
53         S_x = 1/sqrt(U_maxx^2 + V_maxx^2);
54         dPhi_x = atan(V_maxx/U_maxx);
55
56         % % Parameters along the y-direction
57         U_maxy = optimized2(IND(i),1);          V_maxy = ...

```

```

        optimized2(IND(i),2);
58     S_y = 1/sqrt(U_maxy^2 + V_maxy^2);
59     dPhi_y = atan(V_maxy/U_maxy);
60
61     ind = ceil(length(lookup1)/2) + 1;
62     D = 1/sqrt(lookup1(ind,1)^2 + lookup1(ind,2)^2);
63     psi = pi/2 - dPhi_x - dPhi_y;
64     F(:, :, i) = [(S_x*cos(dPhi_y)/(D*sin(psi))) ...
        (S_y*sin(dPhi_x)/(D*sin(psi))); ...
        (S_x*sin(dPhi_y)/(D*sin(psi))) ...
        (S_y*cos(dPhi_x)/(D*sin(psi)))]];
65
66     % E is the Lagrangian finite strain tensor
67     E(:, :, i) = 0.5*(F(:, :, i)'.*F(:, :, i) - eye(size(F(:, :, i))));
68
69     % R is the local rotation matrix
70     R(:, :, i) = F(:, :, i)*(F(:, :, i)'.*F(:, :, i))^-0.5;
71
72     % Radial-circumferential strain tensor rotation matrices
73     [theta, ~] = cart2pol(IND(i,1) - centroid(1), IND(i,2) - ...
        centroid(2));
74     Q_R = [cos(theta) -sin(theta);
75            sin(theta)  cos(theta)];
76     Q_C = [cos(theta+90) -sin(theta+90);
77            sin(theta+90)  cos(theta+90)];
78
79     % Radial and circumferential strain tensors
80     E_dotR(:, :, i) = Q_R*E(:, :, i)*(Q_R.>');
81     E_dotC(:, :, i) = Q_C*E(:, :, i)*(Q_C.>');
82
83     end
84
85     %% Combine all of the data
86     % Find the magnitude of the strain
87     for i = 1:length(E_dotR)
88         EdR(i) = norm(E_dotR(:, :, i));
89         EdC(i) = norm(E_dotC(:, :, i));
90     end
91
92     % Combine the PoI data for each segment
93     for j = 1:max(sector)
94         index = find(sector==j);
95         for k = 1:length(index)
96             strainR(k) = EdR(:, index(j));
97             strainC(k) = EdC(:, index(j));
98         end
99         StrainRseg(j) = {strainR};
100        StrainCseg(j) = {strainC};
101        StrainR(j) = mean(strainR);
102        StrainC(j) = mean(strainC);
103     end

```

```

104
105 % Combine data for grand totals
106 TotalR = mean(EdR);
107 TotalC = mean(EdC);
108
109 %% Save the results variables for the current image to a .mat file
110 filename = strcat(pathName,name,'_analyzed.mat');
111 Name{n} = {filename};
112 save(filename,'image','segmentLabels','test_points',...
113       'sector','members','centroid','contourMask','time','Im',...
114       'rect','EdR','EdC','StrainR','StrainC','TotalR',...
115       'TotalC','AngleImage1','AngleImage2','Filtered1',...
116       'Filtered2','FilteredMag1','FilteredMag2',...
117       'FilteredPhase1','FilteredPhase2','StrainRseg',...
118       'StrainCseg');
119
120 %% Clear dynamic variables to prevent probelms
121 clear image optimized1 optimized2 U_maxx V_maxx S_x dPhi_x U_maxy ...
122       V_maxy S_y dPhi_y ind D psi F E R theta Q_R Q_C E_dotR E_dotC ...
123       strain avg_strainF avg_strainE avg_strainR avg_strainE_dotC ...
124       avg_strainE_dotR EdR EdC StrainR StrainC TotalR TotalC ...
125       AngleImage1 AngleImage2 name
126
127 end
128
129 end

```

A.3.1 Optimize.m

```

1 function [image, optiParameters, IND, contourMask, AngleImage, ...
2         Filtered, FilteredMag, FilteredPhase] = Optimize(Im, contourMask, ...
3         Rect, rect, kernels, test_points, lookuptable)
4 % Optimize.m filters the points of interest, defined in testPoints, of
5 % the input image with the Gabor filters contained in kernels. The
6 % results are then evaluated searching for the greatest magnitude
7 % response. The parameters that generated the maximum response are then
8 % returned stored in a lookup table, optiParameters, and an Angle Image
9 % is created by storing the phase responses of each pixel elicited by
10 % the optimized filters.
11
12 % lookuptable = lookup1; kernels = kernels1;
13 % % Adjust the inputs according to the change in ROI
14 [image, test_points, contourMask1] = adjust(Im, Rect, rect, ...
15       test_points, contourMask);
16 [IND(:,1), IND(:,2)] = find(contourMask1==1);

```

```

15 %% Apply the Gabor filters
16 for n = 1:length(kernels)
17     % Smooth to eliminate erroneous peaks, and then sharpen
18     im = imsharpen(imgaussfilt(image,1), 'Radius',2, 'Amount',10);
19     Filtered(:,:,n) = conv2(im, kernels(:,:,n), 'same');%image, ...
        kernels(:,:,n), 'same');
20     FilteredMag(:,:,n) = abs(Filtered(:,:,n));
21     FilteredPhase(:,:,n) = angle(Filtered(:,:,n));
22 end
23
24 % Create the "more precise" Angle Image
25 [~, index] = max(FilteredMag,[],3);
26 s = size(FilteredMag);
27 for r = 1:s(1)
28     for c = 1:s(2)
29         AngleImage(r,c) = FilteredPhase(r,c,index(r,c));
30         % Store the optimized parameters
31         n = sub2ind(s,r,c);
32         optiParameters(n,:) = lookuptable(index(r,c),:);
33     end
34 end
35
36 % Mask the Angle Image to only display the myocardial information
37 AngleImage = AngleImage .* contourMask1;
38
39 % Improve image display with a gray background
40 ind = find(contourMask1==0);
41 AngleImage(ind) = max(max(AngleImage))/2;
42
43 end

```

adjust.m

```

1 function [image, test_points, contourMask1] = adjust(Im, Rect, rect, ...
        test_points, contourMask)
2 % adjust will modify the ROIs, contours, and PoIs that were selected
3 % during the segmentation process so that all of the images in the
4 % sequence use the same ROI as the first image in the sequence.
5 % Therefore, the dicom images are cropped according to the original
6 % region-of-interest, Rect. The coordinates in the contour mask and
7 % the test_points vector will be adjusted by the change between Rect
8 % and rect, the original ROI of the current image. The contour mask
9 % is also redefined to match the size of the original ROI as indicated
10 % by Rect.
11
12 % Crop the input image

```



```

13 image = Im(Rect(2):Rect(2) + Rect(4), Rect(1):Rect(1) + Rect(3));
14
15 % Correct the test_points for the shift of rect(1:2)
16 test_points = [test_points(:,1) + rect(1), test_points(:,2) + rect(2)];
17 dc = Rect(1) - rect(1); dr = Rect(2) - rect(2);
18 test_points = [test_points(:,1) - Rect(1), test_points(:,2) - Rect(2)];
19
20 % Correct the displacement of the contour
21 [X, Y] = find(contourMask==1);
22 X = X + dr; Y = Y + dc;
23 X = round(X); Y = round(Y);
24
25 % Redefine the size of the contour mask to fit the region-of-interest
26 contourMask1 = zeros(size(image));
27 for n = 1:length(X)
28     if (X(n) ≤ 0), X(n) = 1; end
29     if (Y(n) ≤ 0), Y(n) = 1; end
30     contourMask1(X(n), Y(n)) = 1;
31 end
32
33 end

```

A.4 combineStack.m

```

1 function Name = combineStack(varargin)
2 % combineStack loads the desired *analyzed.mat files for an image
3 % sequence and combines the results into single variables. The
4 % outputs can then be displayed in plots or tables to visualize
5 % and assess the results.
6
7 if isempty(varargin)
8     [fileName, pathName, ~] = uigetfile( ...
9     files = uigetfilename('extension', '*analyzed*.mat');
10     if ~ischar(files)
11         l = length(files);
12     else
13         l = 1;
14     end
15     if isequal(files,0)
16         disp('User pressed cancel')
17         return
18     end
19 else
20     files = varargin{1};
21     l = length(files);

```

```

22 end
23
24 Time = []; Strain_R = []; Strain_C = []; Total_R = []; Total_C = [];
25 for n = 1:l
26     load(files{n}, 'StrainR', 'StrainC', 'TotalR', 'TotalC', 'time', ...
27           'StrainRseg', 'StrainCseg')
28     % Mean strains for each segment
29     Strain_R = [Strain_R; StrainR];
30     Strain_C = [Strain_C; StrainC];
31
32     Time = [Time time];
33
34     % Combine data for grand totals
35     Total_R(n) = TotalR;
36     Total_C(n) = TotalC;
37     segStrainR(n) = {cell2mat(StrainRseg)};
38     segStrainC(n) = {cell2mat(StrainCseg)};
39
40 end
41 Strain_R = Strain_R'; Strain_C = Strain_C';
42 exportr = [Time;Strain_R;Total_R]; exportc = [Time;Strain_C;Total_C];
43
44 %% Save results
45 [pathName, fileName{1}, ~] = fileparts(files{1});
46 [~, fileName{2}, ~] = fileparts(files{n});
47 Name = fullfile(pathName, strcat(char(fileName{1}), '_to_', ...
48     char(fileName{2}), '_stacked.mat'));
49 save(Name, 'Total_R', 'Total_C', 'Strain_R', 'Strain_C', 'n', ...
50     'Time', 'exportr', 'exportc', 'segStrainR', 'segStrainC');
51
52 end

```

B. SYNTHETIC DATA

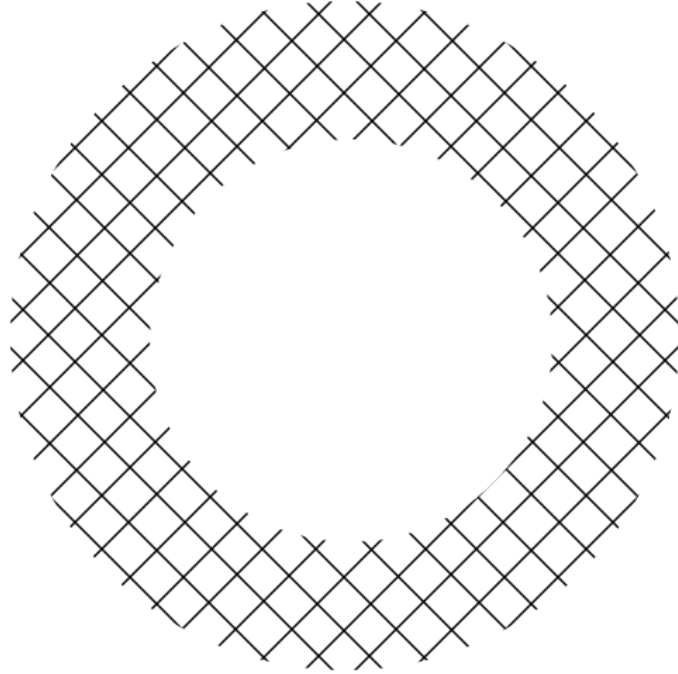


Figure B.1: Synthetic data image 01

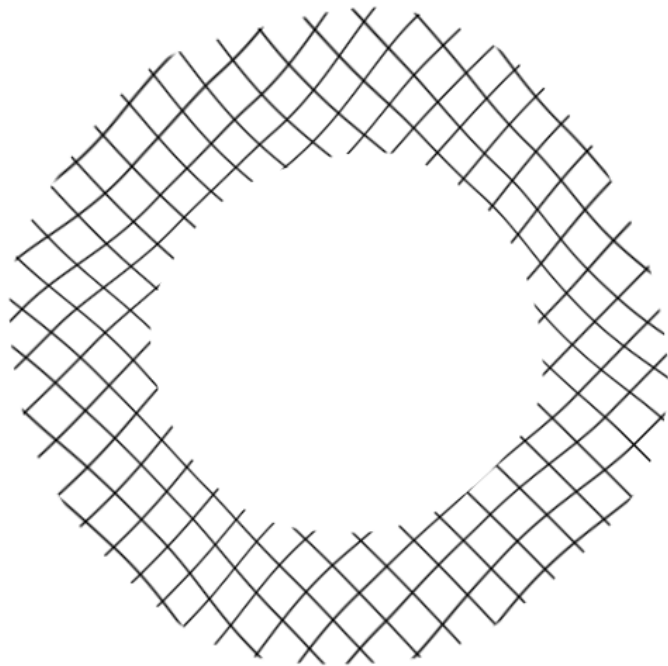


Figure B.2: Synthetic data image 02

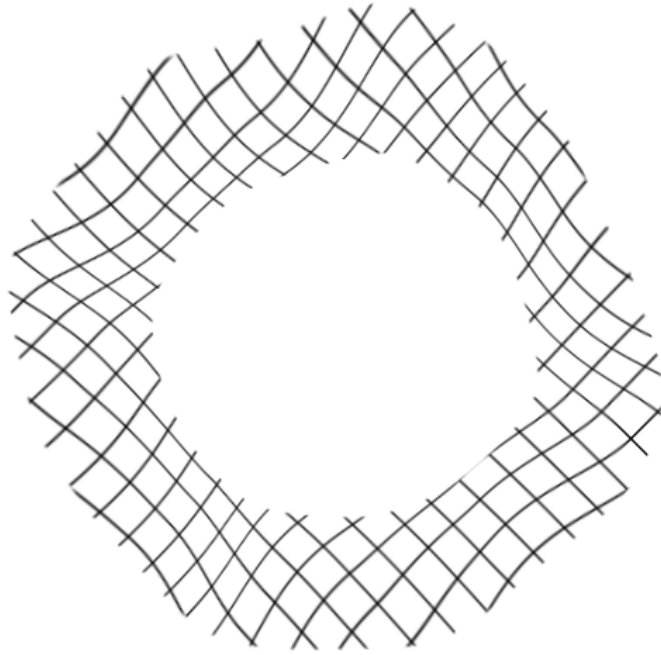


Figure B.3: Synthetic data image 03

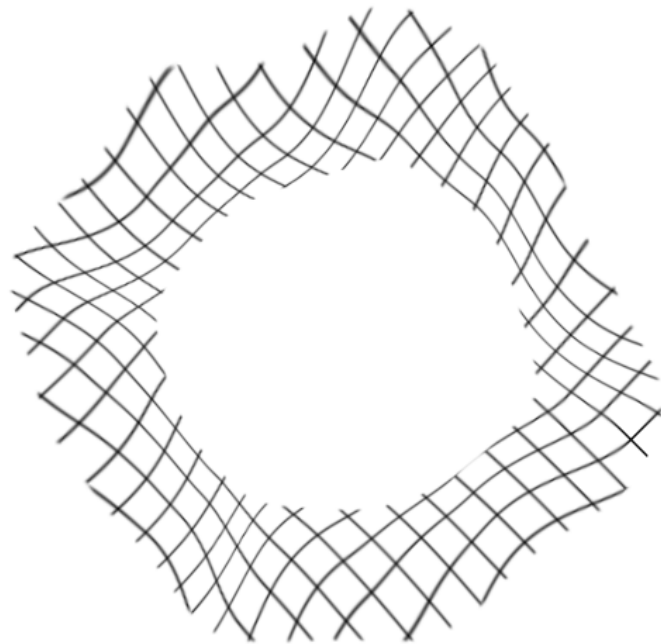


Figure B.4: Synthetic data image 04

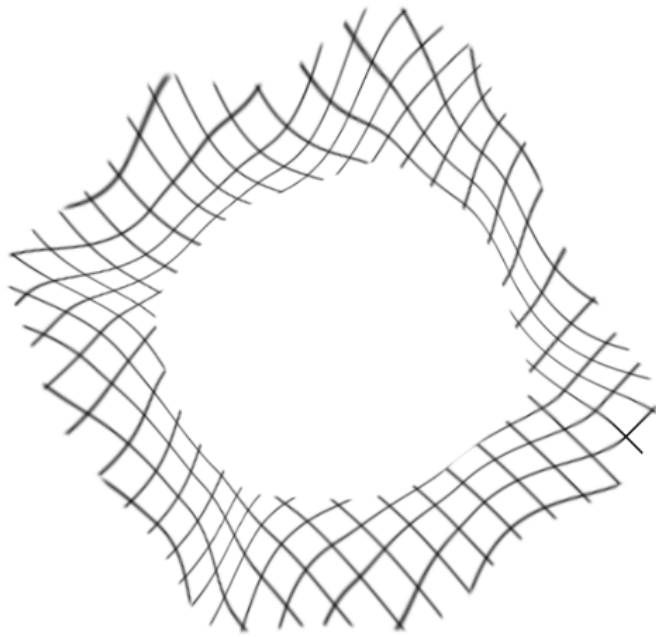


Figure B.5: Synthetic data image 05

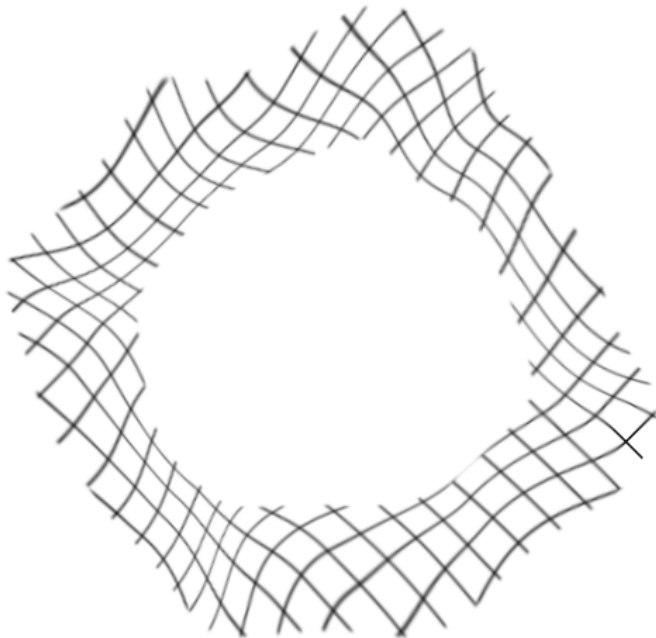


Figure B.6: Synthetic data image 06

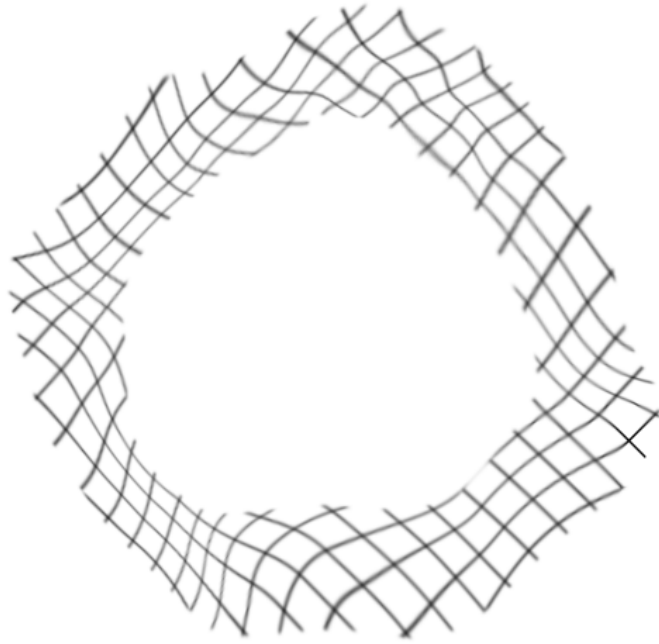


Figure B.7: Synthetic data image 07

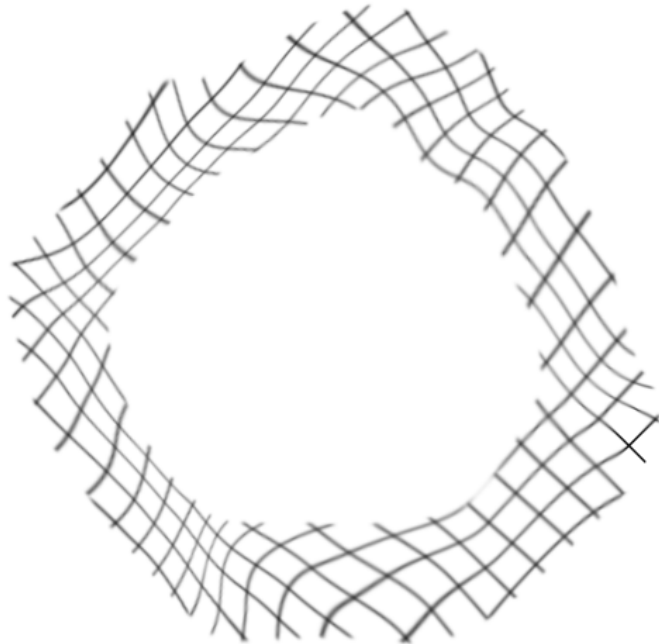


Figure B.8: Synthetic data image 08