2020

Design of a Novel Wearable Ultrasound Vest for Autonomous Monitoring of the Heart Using Machine Learning

Garrett G. Goodman
Wright State University

DESIGN OF A NOVEL WEARABLE ULTRASOUND VEST FOR AUTONOMOUS MONITORING OF THE HEART USING MACHINE LEARNING

A Dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

by

GARRETT G. GOODMAN
M.S., Wright State University, 2018
B.S.C.S., Wright State University, 2016

2020
Wright State University

Wright State University
GRADUATE SCHOOL

Nov 20, 2020

I HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER MY SUPERVISION BY GARRETT G. GOODMAN ENTITLED DESIGN OF A NOVEL WEARABLE ULTRASOUND VEST FOR AUTONOMOUS MONITORING OF THE HEART USING MACHINE LEARNING BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Doctor of Philosophy.

Nikolaos G. Bourbakis, Ph.D.
Dissertation Director

Yong Pei, Ph.D.
Director, Computer Science
and Engineering Ph.D. Program

Barry Milligan, Ph.D.
Interim Dean of the Graduate School

Committee on
Final Examination

Nikolaos G. Bourbakis, Ph.D.

Soon M. Chung, Ph.D.

Yong Pei, Ph.D.

Iosif Papadakis Ktistakis, Ph.D.

Konstantina Nikita, Ph.D.

Anthony Pothoulakis, M.D. (non-voting)

ABSTRACT

Goodman, Garrett G. Ph.D., Department of Computer Science and Engineering, Wright State University, 2020. *DESIGN OF A NOVEL WEARABLE ULTRASOUND VEST FOR AUTONOMOUS MONITORING OF THE HEART USING MACHINE LEARNING*

As the population of older individuals increases worldwide, the number of people with cardiovascular issues and diseases is also increasing. The rate at which individuals in the United States of America and worldwide that succumb to Cardiovascular Disease (CVD) is rising as well. Approximately 2,303 Americans die to some form of CVD per day according to the American Heart Association. Furthermore, the Center for Disease Control and Prevention states that 647,000 Americans die yearly due to some form of CVD, which equates to one person every 37 seconds. Finally, the World Health Organization reports that the number one cause of death globally is from CVD in the form of either myocardial infarctions or strokes.

The primary ways of assisting individuals affected with CVD are from either improved treatments, monitoring research, or primary and secondary prevention measures. In the form of cardiovascular structural monitoring, there are multiple ways of viewing the human heart. That is, Magnetic Resonance Imaging (MRI), Positron Emission Tomography (PET), Computed Tomography (CT), and Ultrasonography are the four fundamental imaging techniques. Though, continuous monitoring with these imaging techniques is far from currently possible. Large financial cost and size (MRI), radiation exposure (PET and CT), or necessary physician assistance (Ultrasonography) are the current primary problems. Though, of the four methodologies, Ultrasonography allows for multiple configurations, is the least expensive, and has no detrimental side effects to the patient.

Therefore, in an effort to improve continuous monitoring capabilities for cardiovascular health, we design a novel wearable ultrasound vest to create a near 3D model of the heart in real-time. Specifically, we provide a structural modeling approach specific to this sys-

tem's design via a Stereo Vision 3D modeling algorithm. Similarly, we introduce multiple Stochastic Petri Net (SPN) models of the heart for future functional feature extraction as well. Finally, this system also includes an individualized prediction methodology via our novel Machine Learning algorithm called the Constrained State Preserved Extreme Learning Machine (CSPELM) for heart state prediction. Thus, the wearable vest will not require continuous medical professional assistance and will allow for real-time autonomous monitoring of the heart. Furthermore, our wearable vest could be the entry point to objective monitoring for patients utilizing telemedicine.

Contents

List of Figures

List of Tables

Acknowledgments

To Dr. Nikolaos Bourbakis, thank you for giving me the opportunity to be your student, a part of your team, and an organizing member of your conferences. Your life lessons and valuable opportunities you have provided me with have not only made me a better computer scientist, but also a better person as well. Even after graduation, I will continue to view you as a lifelong advisor, both in and out of work.

I also want to thank the members of my Ph.D. committee, Dr. Soon Chung, Dr. Yong Pei, Dr. Iosif Papadakis Ktistakis, Dr. Konstantina Nikita, and Dr. Anthony Pothoulakis. Their input and guidance in the final work presented here was very beneficial. Furthermore, I would like to acknowledge my fellow colleagues in the Center of Assistive Research Technologies for their discussions on this topic as well.

Next, I would like to thank my friends Cogan Shimizu, Aikaterini Britzolaki, and Quinn Hirt for their valuable discussions over the multiple topics discussed throughout this dissertation. These discussions solved more problems that I faced with this work than I could have possibly imagined. Finally, I would like to thank my friends Randy Eid, Justin St. Pierre, and Jake Zurface for their continuous support and fun distractions throughout my time as a Ph.D. student. The stress relief is more than a necessary factor for completing this program.

This dissertation was supported in part by the Office of Naval Research (ONR) under grant ONR-CART-N00014-18-1-2144. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the ONR.

This dissertation is dedicated to my family, friends, and Tom Hanks, the greatest actor to ever live. If you see this Mr. Hanks, I would love a signed copy of Saving Private Ryan.

1

Introduction

Cardiovascular health is essential to life as every organ in the human body relies on a functioning heart. Even with this knowledge, many Americans have cardiovascular health concerns. The American Heart Association (AHA) reports that approximately 2,303 Americans die of some form of Cardiovascular Disease (CVD) daily [8]. Furthermore, they estimate by the year 2035, there will be close to 130 million American adults with some form of CVD. Following, according to the Center for Disease Control and Prevention (CDC), approximately 647,000 Americans die yearly due to some form of CVD [36]. As cardiovascular disease is the top leading cause of mortality in America, it incurs a significant financial coast. An estimation by the AHA is that by 2035, the healthcare costs for CVD will be approximately $1.1 trillion. The CDC also reports an estimate of $219 billion spent on cardivascular disease in America yearly. Worldwide, according the World Health Organization (WHO), the number one cause of death globally is CVD where 85% of deaths are from myocardial infarctions (heart attacks) or strokes [104]. The reports from the AHA, CDC, and WHO show the need for further research into cardiovascular treatment and monitoring methodologies.

In the forms of monitoring cardiovascular health, we consider three primary areas.

First, the hospital setting, whether that be the emergency room or inpatient wards. The second monitoring area of interest is in the public sector. Individuals involved in transportation of many people such as bus drivers, pilots, taxi drivers, etc. We are concerned in this area as if the driver has a myocardial infarction, the result could be the injury or worse of the passengers. In the United States of America, the Federal Aviation Administration requires a medical certificate in order to be a commercial pilot [2]. Even with this medical certificate, it does not completely rule out the possibility of a medical emergency during flight. For example, in 2013, a pilot suffered a myocardial infarction during a commercial flight with 161 passengers on board [7]. Another case similar to this happened in 2015 where a pilot suffered a myocardial infarction during a commercial flight with 147 passengers on board [12]. Finally, the third monitoring area involves patients in their home such as older adults of those utilizing telemedicine. Non-invasive medical assistance, outside of emergencies, can be beneficial to older adults to allow them to remain in their comfortable environment while still able to be effectively monitored.

Therefore, due to the growing and aging population of humans and the widespread nature of CVD, we believe that it is important to continue research in both treatment and monitoring methodologies. Our research will be focused on the monitoring aspect as an attempt to further improve the monitoring capabilities of the heart. Specifically, on non-invasive cardiovascular monitoring in both the hospital and public sector settings. Thus, we propose a new medical imaging methodology with the flexibility to cover both the hospital setting and public sector driver/pilot setting. That is, a low-cost, flexible, and real-time imaging system capable of producing a near 3D model of the heart with autonomous heart state prediction capabilities. Our system is specifically a novel wearable ultrasound transducer vest capable of creating a near 3D model of the heart while accounting for the obstructions from the thoracic cage. The methodology uses between 8 and 12 ultrasound transducers in two different orientations and synthesizes the views together via a stereo vision 3D modeling algorithm to produce a near 3D model of the heart in real-time. The

primary benefit to our system alongside the 3D aspect is the ability of continuous unassisted monitoring as Ultrasonography generally requires a medical professional present at all times. Furthermore, the data that is gathered from the 3D model will be used to train a Machine Learning algorithm for cardiovascular state changing predictions to further assist the autonomy of the system. To justify Ultrasonography and stereo vision over other methods, we perform a literature review of state-of-the-art medical imaging techniques and 3D modeling algorithms, respectively. The justification aspect comes from our subjective maturity metric which shows the most suitable medical imaging and 3D modeling algorithm that meets our research needs. Finally, we also create multiple Stochastic Petri Nets (SPN) to introduce a possible method for extracting functional features for future work as the near 3D model of the heart covers the structural features. The flowchart of these contributions is shown in Figure 1.1. This chart details our overall contributions as well as how each individual contribution is interconnected (i.e., which portion of the work utilizes another piece).

Though, the question arises, why 3D model the heart and what are the benefits? An article by Gripari et al. discusses the benefits of 3D echocardiography over the 2D version [52]. Specifically, for patients with mitral valve disease, the 3D echocardiography allows for better modeling of the mitral valve's anatomy and pathology which helps in the prognosis of the disease as well as when surgery should be performed. Another article by Tanis et al. discusses how 2D echocardiography has potential problems detecting the signs of prosthetic heart valve endocarditis due to the acoustic shadowing that occurs from the prosthetic heart valve and how 3D echocardiography may help avert this issue [130]. While these are only two examples, there are many more in the literature. Finally, while the use of continuous objective monitoring, in the case of Ultrasonography, is not the current state-of-the-art in CVD management, we present this work as a new possibility to be used in the field. That is, we hypothesize that having objective monitoring in the homes of older

adults or for patients using telemedicine, it can provide the medical professionals with better prognosis abilities as the data can be transmitted in real-time. Also, with the addition of autonomy to our system, it can be used in a hospital setting to allow more distribution of work between medical professionals or in the public sector for pilots and drivers as they are static during work.

Chapter Overview

The remainder of this dissertation is organized as follows.

Chapter 2: Literature Review details the literature review and corresponding maturity metric for the two main components used in this dissertation. That is, in Section 2.1, we examine work related to Medical Imaging. Then, in Section 2.2, we examine work related to Stereo Vision, both independently and with respect to medical imaging. In each of these sections, we perform a subjective maturity evaluation to determine which of of the medical imaging and 3D modeling algorithms best aligns with our research goals. Once a selection is made, we examine more of the literature surrounding the respective methodologies.

Chapter 3: Wearable Ultrasound Transducer Vest examines the methodology for the Wearable Ultrasound Transducer Vest. That is, we examine two transducer orientations in the form of 4x2 and 4x3 arrays, respectively. We segregate the methodology into multiple pieces which form the overall system. Before this though, we begin the chapter with a brief overview of the macroscopic anatomy and physiology of the human heart in Section 3.1 for general understanding before moving forward. Then, Section 3.2 provides the average specifications of a human heart and thoracic cage used for both orientations. Next, Section 3.3 shows the 4x2 transducer orientation. Specifically, it discusses the vest location and transducer description in Section 3.3.1, the produced stereo overlap in Section 3.3.2, and the model and operation specifications in Section 3.3.3 which shows the generalized

mathematical formulations to produce the real-time 3D image from the vest. Following, Section 3.4 shows the second transducer orientation of 4x3. This section contains similar information to the previous in that Section 3.4.1 shows the new vest location, Section 3.4.2 discusses the new stereo overlap by adding another column of transducers, and Section 3.4.3 gives the new mathematical model and operation specifications. Finally, Section 3.5 provides a brief comparison between both the 4x2 and 4x3 ultrasound transducer orientations.

Chapter 4: 3D Modeling Workflow discusses the example workflow used to create a near 3D model of the heart. Specifically, in Section 4.1, we provide a representative literature review of 3D modeling using Ultrasonography. Following, in Section 4.2, we explain the workflow specifications. Finally, Section 4.3 provides an example of using this process.

Chapter 5: Modeling of the Heart with Stochastic Petri Nets presents three SPN models of a healthy functioning heart and one SPN model of a heart with a CVD to introduce a method for gathering functional features of the heart, with respect to the wearable vest, in the future.

Chapter 6: Design of a Novel Machine Learning Algorithm with Real-Time Training Capabilities examines the Machine Learning algorithm we have created to use in conjunction with the wearable ultrasound transducer vest that has considerably faster training time as compared to traditional backpropagation. That is, Section 6.1 discusses the Machine Learning algorithm that we used as a basis and improved upon. We provide a representative literature review about this algorithm in Section 6.1.1 as well as a technical description in Section 6.1.2. Then, Section 6.2 shows our improved state-of-the-art algorithm in detail while Section 6.2.1 presents comparative results on multiple publicly available datasets. Finally, Section 6.3 provides prediction optimization methods for our

state-of-the-art algorithm and Section 6.3.1 shows experimental results of said optimization methods on multiple publicly available datasets as well.

Chapter 7: Predicting the Change in State of the Heart with Synthetic Heart Chamber Surface Area Data provides the framework for utilizing the algorithm created in Chapter 6 to predict the change in state of the human heart with respect to synthetic heart chamber volume data. First, a representative literature review on CVD predictions is performed in Section 7.1. Next, Section 7.2 discusses the features that can be extracted from the vest and the corresponding prediction possibilities. Then, in Section 7.3, we detail how the synthetic datasets are generated. Following, we provide the reasoning on why we used the algorithm created previously for this task in Section 7.4. Finally, Section 7.5 shows the numerical experiments conducted to verify the validity of the system.

Chapter 8: Discussion provides analysis, use-cases, and limitations of the Wearable Ultrasound Vest. Specifically, Section 8.1 discusses the application possibilities of the wearable vest as well as limitations and expected future challenges. Following, Section 8.2 discusses the Machine Learning algorithm we have created to stay within the real-time constraint and still outperform similar algorithms. Finally, Section 8.3 discusses the results of the numerical experiments from the heart state predictions as well as interesting discoveries from the algorithm created in Chapter 6.

Chapter 9: Conclusion summarizes the contributions of this dissertation.

Chapter 10: Future Work discusses potential future work opportunities in multiple directions creating a working prototype vest, a fuzzy inference system for dynamic setting adjustments for the transducer placements with respect to an individual's anatomy, dynamically adjusting transducer depth's based on an individuals breathing, and heart volume extraction for further real-time insight.

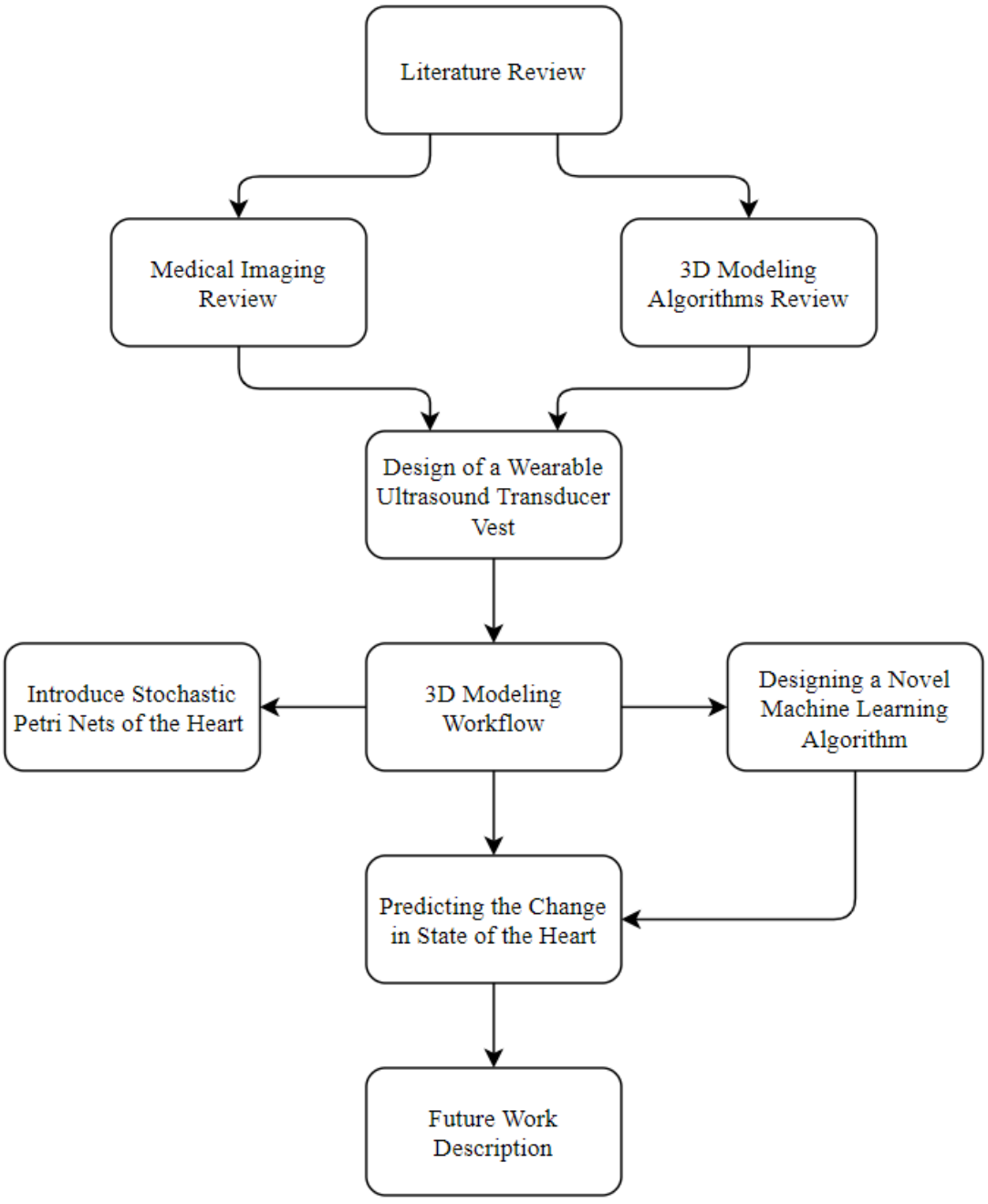


Figure 1.1: The flowchart to show the overall contributions to the topic of this dissertation as well as how each contribution is interconnected.

2

Literature Review

This chapter is to not only examine the state-of-the-art, but also determines the most suitable methodologies for our goal of near 3D modeling the heart in real-time. First, we will examine the primary Medical Imaging practices and perform a subjective maturity evaluation as discussed in Section 2.1. This will allow us to choose the most effective means of collecting images of the heart for 3D modeling with respect to our research goal. Following, we will discuss various popular 3D Modeling Algorithms in Section 2.2 which utilize real world images. We are excluding computer graphics algorithms as they specifically do not utilize real world imagery. We will once again choose the 3D reconstruction algorithm that is most suitable for our goal based off of our subjective maturity evaluation which will be used for this work.

2.1 Medical Imaging

We first need to examine the current medical imaging practices to determine the best course of action to gather the imaging data for reconstructing the 3D model of the heart [46]. There are four primary technologies with 3D medical imaging capabilities which are examined here. First, Positron Emission Tomography (PET) Imaging is discussed in Section 2.1.1.

Next, Magnetic Resonance Imaging (MRI) in Section 2.1.2. Then, Computed Tomography (CT) Imaging in Section 2.1.3. Following, Ultrasonography Imaging in Section 2.1.4. For each of these imaging methodologies, we produce a representative search rather than an exhaustive search for the sake of choosing a methodology. Using these four medical imaging methodologies, we produce a subjective maturity evaluation in Section 2.1.5 with respect to our research goal to decide which imaging methodology we will choose. Finally, in Section 2.1.6, we examine the chosen imaging methodology in more detail.

2.1.1 Positron Emission Tomography Imaging

A Positron Emission Tomography scan is a type of nuclear medicine functionality imaging which produces images by detecting radioactive tracers from a dye that is injected into the patient [70]. The dye, when affected by the metabolic process, emits gamma rays that are detected by the scanner. The PET scanner can produce 2D images on its own but is often combined with a CT scanner to produce 3D images of tissue, organ, and blood flow functionality. The primary strength of PET imaging is that it can show changes in metabolic behavior in organs where the physical appearance has not changed.

Kumar et al. used PET scans on pediatric patients with Autoimmune Neuropsychiatric Disorders Associated with Streptococcal infection (PANDAS) and Tourette syndrome for evaluation of Basal Ganglia and Thalamic Inflammation [77]. There are many dyes or tracers that are effective with PET scans, though they utilized a specific one called ^{11}C-[R]-PK11195 (PK) which they hypothesized would interact with the Basal Ganglia more than other currently available tracers. The authors found that the PK tracer PET scans did indeed find neuroinflammatory changes as expected but with interesting notes. Patients with PANDAS had both bilateral caudate and lentiform nuclei found while Tourette syndrome patients only had bilateral caudate present.

In an article by Johnson et al., PET scans were utilized in both elderly patients with

mild cognitive impairment or mild Alzheimer's Disease and elderly clinically healthy individuals [67]. The authors hypothesize that the tracer ^{18}F T807 bindings would be elevated in neocortical areas of impaired patients versus clinically healthy patients. Furthermore, they examined if it would correlate with the standard ^{11}C Pittsburgh compound B tracer. The authors found that ^{18}F T807 bindings were indeed higher in impaired patients rather than clinically healthy patients.

Similarly, in an article by Gomperts et al., the authors examined how Tau aggregation in both Dementia with Lewy Bodies (DLB) and Parkinson Disease (PD) effect these diseases as compared to clinically non-impaired individuals [42]. This was done by using the PET scanner with the ^{18}F AV1451 tracer. The author's findings were that the Tau aggregation was greater in impaired DLB and PD patients in the inferior temporal gyrus and precuneus as compared to clinically non-impaired individuals.

These articles relating to PET research show that the technique of PET scanning is a well-known practice, but the nuclear portion of the system regarding the tracer used during testing is the primarily researched area. The PET scanner is not a readily available device and has a high financial cost. Furthermore, the training required to operate PET scanners is extensive.

2.1.2 Magnetic Resonance Imaging

Magnetic Resonance Imaging is a medical imaging technique which uses magnetic fields and radio waves to produce images of the internals of a human, such as organs and soft tissue [92]. Specifically, MRI utilizes large spinning magnets to send radio waves through the body to produce the desired functional and structural images by detecting subtle changes in the magnetism in the nucleus of atoms that the radio waves pass through [92]. Currently, MRI scanners are widely used in large hospitals and clinics and have the benefit of not exposing the patient to radiation as done in PET and CT scans. Though, due to their large

size and initial purchase cost, they may not be available in all locations.

In an article by Lurz et al., the authors used Cardiac Magnetic Resonance (CMR) imaging, a sub-method of MRI [96], as a diagnostic tool and compared the results to an endomyocardial biopsy (EMB) [84]. The authors used two different CMR scanners, a 1.5 and 3.0 Tesla scanner (Tesla is the strength of the magnetic field created by the MRI machine). The results using the CMR imaging technique outperformed the Lake Louise criteria, which is a widely used approach for diagnosing Myocarditis, when presented with acute symptoms and performed slightly below when presented with chronic symptoms.

Nagle and Eichinger compared numerous MRI methods to the traditional chest X-ray and CT scan for monitoring Cystic Fibrosis (CF) [98]. The idea of using MRI instead of X-ray or CT scans is due to the major drawback of radiation exposure from these two sources, which are especially detrimental to young children. From the functional MRI method set, the authors compared contrast enhanced 3D MRI, hyperpolarized gas ventilation MRI, oxygen enhanced MRI, and more to chest X-ray and CT scans. Then, structural MRI scans were compared using proton-MRI sequences with both T1 and T2 weights. These comparisons showed promise in allowing for an alternative imaging process for patients with CF that does not have radiation exposure, even if it is in small doses.

Waugh et al. utilized dynamic contrast-enhanced breast MRI to perform texture analysis on recently diagnosed women with breast cancer [138]. The goal was to determine if texture analysis from the MRI could be used as a non-invasive method of lesion (histological and immunohistochemical) subtype classification as treatment is tailored depending on the lesion subtype. A model was created from features produced from a co-occurrence matrix. The authors utilized the Area Under the Receiver Operating Characteristic (AUROC) curves and overall percentage accuracy to show performances of 0.816 and 75%, respectively. The authors state that using contrast-enhanced breast MRI is still evolving but can be useful for cases in which multiple lesions are present and biopsies become increasingly

difficult.

The MRI technique provides a very useful imaging methodology as there are multiple sub-methods of MRI, each with its own strengths and weaknesses. The primary strength to an MRI scan over a PET scan is that no radiation exposure occurs. Though, due to their large size and initial cost of purchase, it is not always common for an MRI machine to be available at all locations.

2.1.3 Computed Tomography Imaging

CT scans, an extension of X-ray imaging, is a form of structural X-ray imaging that produces 3D images by using a rotational X-ray beam to capture all angles of the body part in question [14]. The X-ray that is utilized in the CT scan produces electromagnetic radiation which is sent through a specified portion of the body. The final image produced from the CT scan is a cross-sectional image of multiple slices constructed with the assistance of a computer. Medical uses range from examining bone structures to entire organ imaging. The major drawback to the CT scan is the radiation exposure from the X-rays.

Hou et al. used an extended form of the CT scanner called a dual-energy spectral CT (DESCT) scanner, which rapidly switches between high and low energy data during each rotation [56]. This allows for two different images to be produced which are monochromatic spectral images and material decomposition images, respectively. The authors used this DESCT scanner to examine how the produced images can differentiate between pulmonary malignant mases and inflammatory masses as these two masses have different potential treatments. Their findings showed that a combination of the iodine concentration, slope of the spectral attenuation curves, and the differences between the central and peripheral regions of the masses produced from the DESCT scanner could be used in the differentiation of pulmonary malignant mases and inflammatory masses.

An article by Ehling et al. used micro-CT for examining the vascular alterations of

progressive kidney injury [31]. The authors used mice as their test subjects rather than human participants. The in vivo (mice subjects are alive) micro-CT imaging allowed for continuous monitoring of the blood vessels and showed decline in blood volume as the disease progressed. Then, ex vivo (mice subjects are deceased) micro-CT imaging allowed for quantitative 3D modeling of the blood vessel networks.

The CT scanner is also used in cardiac care. Danad et al. discussed the various techniques of CT scanning that can be utilized by cardiologists [27]. Their article discussed the use of dual-energy CT (DECT), spectral CT, and CT-based molecular imaging. They discuss the technical side of each CT sub-imaging method and the corresponding clinical applications. For DECT, myocardial perfusion imaging and coronary atherosclerotic plaque characterization. As for Spectral CT, it can assist with atherosclerosis evaluation. Finally, CT-based molecular imaging can assist in detection of non-occlusive microthrombus-associated ruptured plaques within the coronary arteries.

These studies show the multiple uses and sub-methods of CT imaging. As CT imaging is an improvement upon normal X-ray imaging, it still contains the radiation exposure problem. Though, compared to MRI and PET scans, it is less expensive and more readily available.

2.1.4 Ultrasonography Imaging

Ultrasound imaging, otherwise known as Ultrasonography, is a real-time non-invasive medical imaging technique used in the diagnostic process for numerous different medical fields such as Cardiology, Radiology, Obstetrics, and more [95]. Ultrasonography operates by utilizing a small ultrasound transducer that contains one or more piezoelectric crystals. When an electric current is applied to these crystal, they rapidly vibrate and produce a sound wave that is emitted outward to the desired location. That same sound wave then reflects off of some object, the heart for example (or anything beneath the tissue layer [22]), and is

reciprocated by the same piezoelectric crystal(s). Once the sound wave is reciprocated, the piezoelectric crystal(s) produce an electrical current which is read by a CPU and processes the data to produce the medical images.

Ultrasonography has found success in the Point of Care (POC) sensing, otherwise known as bedside testing, in which diagnostic sensing and testing happens at the literal POC rather than confined to a medical laboratory. An article by Dr. Kimura discussed how the use of POC Ultrasonography can improve the accuracy of a standard physical cardiac examination [74]. The considerable improvements from the inclusion of ultrasound were not restricted by the experience of an individual physician. Instead, even a new inexperienced physician showed improved examination accuracy. Specifically, the ultrasound examination allowed the viewing of pulmonary congestion, right heart functionality and central venous pressures, and more.

An article by Tsakalakis and Bourbakis describes a multi-transducer system for 3D modeling an abdominal organ in real-time [133]. The proposed system's architecture allows for a wearable vest containing 6 ultrasound transducers which is placed over the desired abdominal organ, in this case the liver. This eliminates the need for a physician to physically hold an ultrasound probe over the area of interest for real-time imaging. The article shows how the 6 transducers are oriented (in a 3x2 array) and the overlapping Field of View (FOV) at multiple depths. Finally, a preliminary mathematical model is created to show how the system is able to capture the entire region of interest.

Similar to the 3x2 ultrasound transducer orientation [133], an article by Gemmeke and Ruiter discusses an ultrasound system formatted in a CT (USCT) fashion (i.e., many transducers oriented in a circle facing inward) [40]. This system would also eliminate the constant attention of holding an ultrasound probe needed by a physician. The 3D imaging USCT setup has some negatives associated with it though. Specifically, a large number of ultrasound sensors are needed, which increases likelihood of side lobe artifacts (leaked or

extra energy returning outside of the main ultrasound wave). Also, to obtain accurate sub-millimeter 3D resolution, activation time for all transducers has to be as short as possible as patient movement is inevitable. The article describes both 2D and 3D versions of their USCT system in which the transducers are assigned to be either a sender or a receiver of a sound wave. The 3D USCT for example contains 2304 virtual senders and 9216 virtual receivers. Unfortunately, image reconstruction takes approximately 8 to 16 days using several computers in parallel but is able to image structures down to 0.15mm in size.

These studies show the flexibility and usefulness of ultrasound imaging in the diagnostic process. Ultrasonography allows for real-time imaging of abdominal organs [133], cancer and cysts [40], and more. There is also no radiation as is used in PET and CT scans. Furthermore, the ultrasound transducers are cheaper in the financial sense and the amount of training necessary to use this method is low as compared to MRI, CT, or PET scanning technology.

2.1.5 Medical Imaging Maturity Evaluation

We now present the subjective maturity evaluation for the four discussed medical imaging methodologies of PET, MRI, CT, and Ultrasonography. For the maturity metric, we have created 12 features which represent different aspects that are important to us in our evaluation presented in Table 2.1. We calculate a maturity metric based upon these 12 presented features and their corresponding weights to describe how mature a methodology is from two perspectives. That is, with respect to the patient's perspective and medical professional's perspective. The calculation of the maturity metric is shown in Equation 2.1 below.

$$M_i = \frac{\sum_{j=1}^{12} w_j F_{ij}}{\sum_{j=1}^{12} w_j} \tag{2.1}$$

In Equation 2.1, we calculate the maturity M_i for methodology i, or more specifically study i. The variable w_j is the weight of feature j with respect to either the patient or medical professional's perspective. Finally, F_{ij} is the feature value j of methodology i. Using this equation, we calculate the maturity rating for all 12 presented studies. Table 2.2 presents the feature weights associated with both perspectives. The weights range from 0.1 to 1 where 1 indicates that the feature has a significant role in the overall maturity. Following, Table 2.3 shows feature scores from 1 to 5 and the calculated maturity. A feature with a score of 5 shows high performance while a feature with a score of 1 shows low performance.

Table 2.1: The list of features to subjectively evaluate the four medical imaging methodologies of PET, MRI, CT, and Ultrasonography. The abbreviation IFn next to each feature stands for "Imaging Feature n".

Feature	Description
Ease of Use (IF1)	The level of training needed to operate the imaging equipment.
Portability (IF2)	How portable the imaging equipment is.
Real-Time Application (IF3)	Whether the equipment is able to produce real time results (binary feature).
Resolution (IF4)	The level of detail the imaging equipment can produce.
Decision Support (IF5)	The level of support the equipment gives to making a final diagnoses.
Discomfort (IF6)	The amount of discomfort the individual interacting with the equipment feels.
Power Consumption (IF7)	The amount of energy required for the equipment to function.
Security (IF8)	Level of security of the data transfer from the equipment.
Cost (IF9)	The initial financial cost of the equipment.
Size (IF10)	The physical dimensions of the equipment.
Flexibility (IF11)	The ability for the system to be modified into different configurations for operation.
Continuous Imaging (IF12)	The ability for the system to continuously image a patient over an extended period of time.

Finally, we present Figure 2.1, a histogram of maturity ratings of all 12 reviewed studies. Examining the figure, we can see that no methodology has a perfect maturity rating in either presented perspective. Though, this is an expected outcome. It is nearly impossible

Table 2.2: The list of feature weights for all 12 features. Feature weights range from 0.1 to 1 where 1 means that the feature has a significant role in the maturity of the methodology. Both the patient's perspective and the medical professional's perspective are accounted for as well as the average between the two.

Feature	**Weights**		
	Patient's Perspective	Medical Professional's Perspective	Average
IF1	0.4	0.9	0.65
IF2	0.6	0.7	0.65
IF3	0.9	0.9	0.9
IF4	0.5	0.8	0.65
IF5	0.8	1	0.9
IF6	0.9	0.5	0.7
IF7	0.2	0.6	0.4
IF8	0.4	1	0.7
IF9	0.8	0.8	0.8
IF10	0.4	0.8	0.6
IF11	0.4	0.7	0.55
IF12	0.7	0.8	0.75

to create a methodology that is able to perfectly account for each of the features without any negatives associated with it. There are also a few interesting notes from both Figure 2.1 and Table 2.3. Starting with the PET scanning studies [70, 67, 42], the only noticeably different approach these studies provided was the type of dye used. That is, the chemical makeup of the dye changed with the hypothesis that it would allow for more prominent imaging of a particular area or organ. Thus, there is a consistent maturity rating across the PET scanning studies. Next, for the MRI studies [84, 98, 138], there are small differences between the flexibility in the studies as well as power consumption. Overall though, the MRI systems are still large, expensive, and require extensive training. Following, for the CT studies [56, 27, 31], the most notable differentiation of these studies comes from the Micro-CT used on the mice in [31]. This increased the portability, size, and power consumption feature ratings. Though, radiation still plays a large negative factor in this particular methodology. Finally, for the Ultrasonography studies [74, 133, 40], these showed the highest overall maturity ratings, specifically from [133]. This is because the studies were able to capture 3D images, with low discomfort and high flexibility, in real-time. These

Table 2.3: The assigned feature values for all 12 features and the corresponding maturity calculation for both the patient's and medical professional's perspectives. The feature values range from 1 to 5 where a value of 5 shows the highest performance of said feature.

Features	**Scores**											
	[77]	[67]	[42]	[84]	[98]	[138]	[56]	[27]	[31]	[74]	[133]	[40]
IF1	2	2	2	2	2	2	2	2	2	4	3	2
IF2	2	2	2	1	1	1	2	2	3	3	4	2
IF3	0	0	0	0	0	0	0	0	0	1	1	0
IF4	4	4	4	4	4	4	4	4	4	2	4	4
IF5	4	4	4	4	4	4	4	4	4	2	2	3
IF6	2	2	2	3	2	3	2	2	2	4	4	3
IF7	2	2	2	1	2	2	2	2	3	4	4	3
IF8	4	4	4	4	4	4	4	4	4	3	3	4
IF9	2	2	2	2	2	2	2	2	3	4	4	3
IF10	1	1	1	1	1	1	1	1	3	4	4	3
IF11	2	2	2	3	4	2	3	4	2	2	4	3
IF12	1	1	1	2	2	2	1	1	1	2	3	4
Patient's Perspective	2.07	2.07	2.07	2.24	2.2	2.21	2.13	2.19	2.41	2.79	3.17	2.7
Medical Professional's Perspective	2.23	2.23	2.23	2.31	2.39	2.30	2.31	2.38	2.62	2.84	3.22	2.82
Maturity Average	2.15	2.15	2.15	2.27	2.30	2.26	2.22	2.28	2.52	2.81	3.20	2.76

features mimic the research goal of 3D modeling the heart in real-time. Therefore, we will choose Ultrasonography as our medical imaging methodology.

2.1.6 Further Examination of Ultrasonography Imaging

As Ultrasonography was chosen from the maturity evaluation, we examine more in-depth the various approaches in recent literature. Fortunately, a survey has been performed by Huang and Zeng on real-time 3D ultrasound imaging [61]. With their work and the following articles, we can obtain a good insight of the current research in Ultrasonography. An article by Maresca et al. examined Coronary Vasculature imaging with ultrafast ultrasound [89]. Ultrafast ultrasound, while conceptually old, is functionally a newer form of ultrasound imaging that increases the speed in which an image is created by utilizing unfocused transmit beams rather than multiple focused transmit beams as is used in conventional ul-

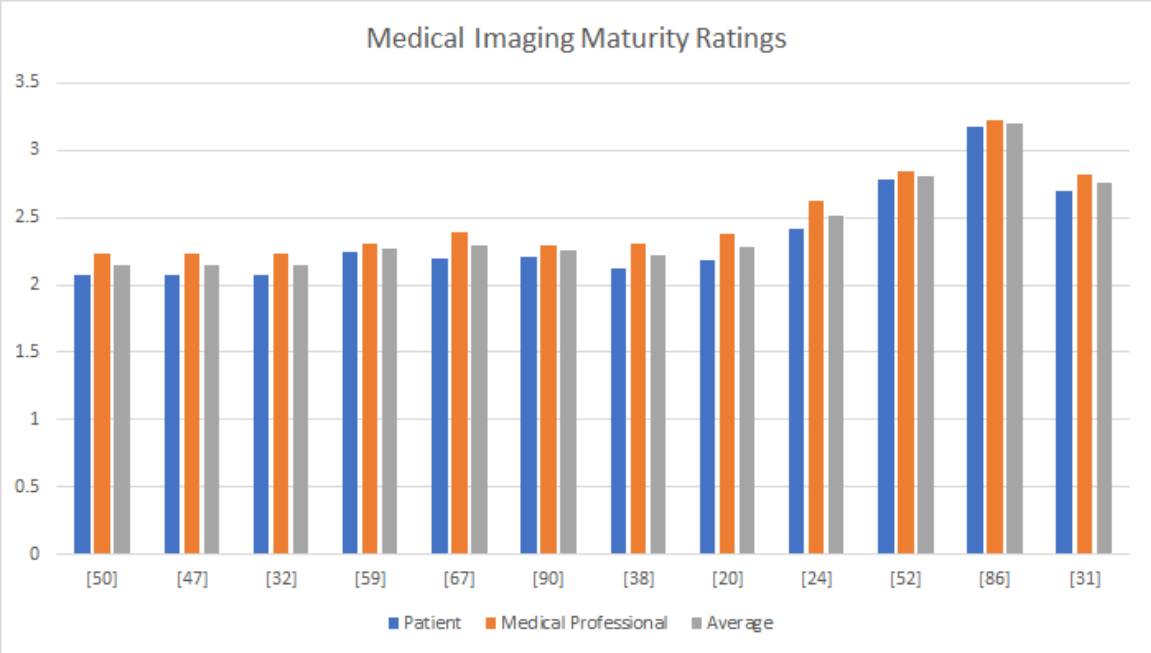


Figure 2.1: Maturity scores for all 12 medical imaging methodologies. The maximum maturity rating is 5 while the lowest maturity rating is 0.

trasound transducers [13, 54]. These unfocused beams allow for parallel scan line creation and thus the number of ultrasound emissions is reduced drastically. The downside of this is generally the spatial resolution and contrast of the produced image is worse than the conventional method. The article by Maresca et al. specifically uses Coronary Ultrafast Doppler Angiography (CUDA) on 9 open-chest in vivo swine. CUDA allowed for visualization of ultrahigh-sensitive cine loops of venous and arterial intramyocardial blood flows within a single cardiac cycle. Furthermore, feasibility of CUDA was shown on 4 human volunteers.

The emergency room also includes POC Ultrasonography. In an article by Gaspari et al., they used POC Ultrasonography imaging to improve the outcomes of Advanced Cardiac Life Support (ACLS) [39]. A total of 793 patients with cardiac arrest were examined with ultrasound and the authors found that cardiac activity on ultrasound was the variable most associated with survival based on a multivariate regression model. Furthermore, no cardiac activity during the ultrasound imaging corresponded highly with non-survival, though 0.6%

survived to hospital discharge. The multivariate model contained variables such as gender, rhythm during initial ultrasound imaging, cardiac activity, shockable rhythm, etc.

An article by Crosby et al. utilized POC ultrasound imaging for diagnosis of Acute Mesenteric Ischaemia (sudden severe abdominal pain) [24]. The real-time imaging ability of ultrasound is important in this case as the treatment often requires emergent medical or surgical operations. The POC ultrasound imaging was utilized to find a mobile mass in the left ventricle of the heart. While ultrasound imaging in this case did not give a complete diagnosis, it did narrow the scope of examination to quickly find a solution to save this individuals life.

Continuing, POC Ultrasonography in conjunction with a traditional clinical evaluation for infection source identification in septic patients has been researched by Cortellaro et al. [23]. The inclusion of the POC ultrasound examination improved both the accuracy and more importantly the speed of diagnosis. Out of a cohort of 200 patients, only 1 false positive occurred. The authors stated that it took 6 hours to confirm the location of the infection in 60 out of the 200 patients.

Following, an article by Taylor and Moore showed how an ultrasound examination can be conducted on the thoracic aorta to obtain useful images for prognosis and diagnosis efforts [131]. The article shows different positioning of the ultrasound probe for multiple views. For example, suprasternal position for the aortic arch and branches, parasternal short-axis position for a cross-section and longitudinal descending view of the aortic root, and more.

Further examinations of Ultrasonography show the multiple areas of interest in which this type of imaging can be effectively utilized. In most cases, POC ultrasound imaging is the most useful due to its real-time applications such as in [39] and [24]. Also, research has been done in "Ultrafast Ultrasound" to improve the speed of image creation drastically with the downside of poorer spatial resolution and contrast of the image [13, 54]. This

imaging process though is certainly the most effective for our original goal of 3D modeling of the heart in real-time.

2.2 3D Modeling Algorithms

Now that we have chosen the Ultrasonography medical imaging technique as it best accommodates our needs, we need to examine the current state-of-the-art 3D reconstruction algorithms. Fortunately, there is plentiful research into 3D modeling techniques as pointed out in the survey conducted by Kordelas et al. [75]. In this survey, they discuss multiple algorithms from which we will cover two here. These include Stereo Vision in Section 2.2.1 and Laser Scanning 3D Reconstruction in Section 2.2.2. We also include the Orthographic Projections algorithm in Section 2.2.3 and the Single Image Pixel Depth Reconstruction algorithm in Section 2.2.4 which are not discussed in the survey by Kordelas et al. Similar to the medical imaging methodology selection process, we once again produce a representative search rather than an exhaustive search for choosing the 3D modeling technique. From this, we conduct a subjective maturity evaluation in Section 2.2.5 of these algorithms with respect to our research goal to choose which algorithm will be utilized for the remainder of this work. Finally, in Section 2.2.6, we examine our chosen method in further detail.

2.2.1 Stereo Vision

Stereo vision is the process of extracting 3D information from two or more 2D overlapping images. Emphasis is placed on the overlapping of the images. This is also known as binocular vision (for two images) as it replicates the depth perception and registration of humans and animals. The depth of the two images is obtained via the disparity, otherwise known as the horizontal shift between image one (left) and image two (right) [26]. In order to calculate the point correspondence between two images, we only need to know the parameters of

the setup, i.e., the camera's orientation and focal length (in pixels). The points correspond to vectors emanating from its respective camera. Then, we can triangulate the intersections from the two vectors via epipolar geometry to obtain the final point coordinates on a 3D plane.

Finding correspondence points between two images is very difficult and rapidly increases in complexity as the image resolution increases. This is due to any single pixel looking similar to many pixels in the second image. So, we apply the epipolar constraint [26], which is a 2D line in the image where we can limit our search of correspondence points. The epipolar constraint is created by forming a plane between camera center point one, camera center point two, and a camera point in which the point ray from camera one intersects. With this plane, we can form the epipolar constraint, or line on the camera lens, on both cameras to drastically limit our point correspondence search. If the images are not aligned, they can be warped to be realigned so that the epipolar constraints are parallel. Even with the limited pixel search, some ambiguity remains. The common method of bypassing the ambiguity is to use a correlation approach. Given a small pixel window from the left image, take the correlation between that window and a set of pixel windows from the right image and the highest correlation is the corresponding pixel. With the depth of the correspondence points calculated, we can now plot the set of point in 3D space.

With a brief overview of stereo vision presented, the following literature examines approaches to improve the current technique or discusses related application fields. Hernandez et al. proposed a method for finding illumination directions and the corresponding intensity of Lambertian objects (objects with reflective surfaces capable of reflecting light in multiple directions) using silhouettes [55]. The setup consisted of a single camera, vertically adjustable light source, and a 360° turntable for the object to sit and rotate for multiple stereo overlapping images. The algorithm proposed in this study utilizes the silhouette and shading cues of the object in question. The authors estimate the illumination direction and

intensities by constructing a predicted visual hull from contours of the silhouettes from camera movement. Then, a mesh of vertices, triangular faces, and albedo (incident light surface reflections) is created which represents the surfaces of the 2D images of the object to solve the Lambertian model. The algorithm then optimizes the mesh with a cost function until the predicted mesh sufficiently models the actual surface. Specifically, the vertices and light direction vectors (called photometric normals by the authors) are optimized one at a time while keeping the others fixed.

An interesting application study by Okada et al. uses stereo vision in conjunction with laparoscopic surgery (surgery in the abdomen or pelvis by small incisions aided by one or more cameras) for image depth calculation [103]. The article creates a new stereo vision matching algorithm based off of their previous work for determining the depth from the stereoscopic laparoscope. Their previous work created the initial system for calculating the depth of the laparoscope while this study improves the algorithms computational complexity. Specifically, it utilizes only local-region based registration rather than registration of the entire stereo pair.

In an article by Sengupta et al., the authors propose a method for reconstructing urban large scaled 3D environments [118]. This method utilizes semantic labels of pixels and calibrated stereo cameras on a moving vehicle. The moving vehicle captures multiple stereo images over time which are semantically mapped during the 3D reconstruction phase to form the semantic 3D stereo model. Camera pose estimation is done by matching correspondence features from stereo image frames. Following, the 3D surface reconstruction is performed by generating a depth map from the disparity image on a frame by frame basis. Then, each computed depth map is fused together using a volumetric approach consisting of a truncated signed distance function (TSDF) that differentiates between points behind objects as negative values and visible points as positive values. This extends to an online version as the car containing the stereo camera rig can cover long distances. A mesh is

applied using the marching tetrahedra algorithm for extracting triangulations based on the zero-crossing of the TSDF algorithm. Finally, an aggregate of semantic pixel labels is used to determine the final label by majority rule of all areas within the 3D model. Stereo vision is an older method but to this day is still studied as it is one of the most consistent methods for reconstructing 3D images of scenes from one or more views without the assistance of lasers or external devices.

2.2.2 Laser Scanning 3D Reconstruction

Laser scanning 3D reconstruction is a methodology which utilizes lasers as the primary source of depth tracking for producing a 3D point cloud [75]. This method allows for a very fast, accurate, and high resolution 3D image reconstruction. Generally, this methodology is used for large scale scene reconstruction, building interior and exterior mapping, or archaeological site preservation. There are three steps that are required for laser scanning 3D reconstruction. First, an adequate number of color laser range scans are needed which cover the entirety of the scene in question. Second, registration of the range scans that are within the same coordinate system are performed to produce an accurate point cloud. This step is where research is predominately happening. Therefore, two of the three following articles will be covering registration methods. Finally, step three consists of data processing to remove redundant information and to construct the final 3D model from the point cloud using methods such as polygonal facets obtained from the produced point cloud or curve fitting.

In a study by Bienert and Maas, they utilize a terrestrial laser scanner and gather multiple recordings of trees from different angles for forestry applications [17]. The goal of the study was to create an automatic registration process without pre-assigned artificial tie points (a feature clearly visible in two or more images which can serve as a reference point). In the forestry application, it is often the case that artificial tie points must be placed

which is time consuming. So, the authors present first a registration method using multiple tie points of white spheres placed in the scenery in which they are detected by the color intensity after each pixel is put through a white and black threshold. Corresponding point patterns are calculated by iteratively finding all distances between tie points. Finally, the automatic registration from tree axes is presented by using only one tie point and extracting multiple planes angled with respect to multiple corresponding trees surrounding the tie point. This creates multiple new virtual tie points rather than having to place real tie points in the scene.

Following, an article by Poncet et al. examines a method of laser optical scanning a human torso to reproduce the topography and estimate the 3D spine parameters [109]. Specifically, this is aimed towards individuals with scoliosis as the traditional method is multiple full spinal X-rays which has a large amount of radiation exposure that is very detrimental to them. The authors constructed a custom laser scanning system consisting of four laser cameras on a symmetrical steel ring. The ring is connected to a large tubular-steel frame that has vertical axis movement with a stabilizing device to hold the patient in place. After an x-ray scan, the four laser cameras mapped the topography and estimated an accurate 3D reconstruction of the spine and rib cage. This allows for minimizing the radiation exposure to the patients.

An article by Stamos et al. discusses another registration method for 3D laser scanning reconstruction [127]. Specifically, the study examines automated registration of urban large scale structures with 2D-to-3D and 3D-to-3D registration techniques. First, the 3D-to-3D registration technique is performed after an adequate set of laser scans is gathered which covers the scene in question. Then, edge detection is performed followed by circular feature extraction. This feature extraction produces literal circles, and sometimes ovals depending on the viewing angle, from the images that serve as technical tie points. This leads to feature matching that is performed by computing the surface consistency and average point

distances. The 2D-to-3D registration is then shown that uses 2D photographs to generate a 3D model of the scene via point cloud generation.

While laser scanning 3D reconstruction is a fast and accurate methodology, it is generally used for large scenery such as forests or buildings (interior and exterior). Furthermore, the study by Poncet et al. utilized an invasive, expensive, and rigid methodology consisting of a steel frame and stabilizing device to hold the patient standing upright and still utilized an X-ray image [109].

2.2.3 Orthographic Projections

Orthographic projections are an older methodology used to create a 3D object from at minimum three 2D views [91]. These three views are the front, top, and side views, respectively. Generally, these views are used in engineering applications and engineering modeling software. Orthographic projections are created with the property stating that all the 2D projection lines are orthogonal to the corresponding projection plane. This results in the planar scene representing an Affine space, i.e., a 3D generalized Euclidean space in which only properties of parallelism are kept. Following, affine transformations can be done which preserve the parallel line segment's properties (distance ratios from points on a straight line) of the now 3D object.

As an object becomes more sophisticated in design, more 2D views and angles are needed to sufficiently model the object in 3D. Such an object could be a specific machine part with multiple angles and protrusions that the simple three view orthographic projection cannot handle. We can expand this methodology into a 6-tuple orthographic projection to allow for more precise views of the object in question [91]. The 6-tuple consists of left, right, top, bottom, near, and far 2D plane views (similar to the sides of a cube) that describe the object in detail. With the six views created, the 6-tuple, which represents a cube in Affine space, is centered at the origin and scaled to [-1, 1]. All following transformations

can be calculated with a predetermined set of homogeneous matrix coordinates.

Following the orthographic projection high-level methodology description, we present three related articles. First, an article by Aldefeld created an automated recognition system of orthographic projections from three views (front, side, and top) [3]. The process consists of 4 primary components. First is Segmentation of Primitives, which means to divide 2D input primitives that have structural contact with another primitive. Second is Attributes and Relationships, which checks each new data primitive to see if a new entity has been added, then proceeds to assign labels to all previous primitives as needed. Third, is Substructure Selection which finds all possible substructures of the given primitives using a heuristic search. Finally, object recognition is performed. This consists of Model Guided Recognition, which in brief finds all uniform-thickness objects that consists of a loop in a given view that represents the base silhouette of the object. Then, if the objects found comply with the predetermined engineering drawing rules, the algorithm terminates and outputs the final result. While this is an older article, it was very beneficial to the mechanical Computer Aided Design (CAD).

Next, an article by Shin and Shin discuss an improved algorithm for reconstructing solid 3D objects from 2D orthographic projections [121]. Similar to Aldefeld's study [3], three orthographic views are used here as well. These are the front, top, and side views. A block diagram is provided which describes the linear algorithm for 3D reconstruction in 8 steps: input checking, vertex generation, edge generation, edge segmentation, face construction, cutting edge insertion, candidate block construction, and decision making. First, input checking simply decides if the input views are valid. Then, vertex generation is performed which classifies each 3D vertex into two classes. Class I, an intersection point of non-collinear line segments in two or more views. Class II, an intersection point of non-collinear line segments in only one view. Next, candidate edges are chosen for three subtypes: standard, silhouette, and tangency. Each having their own corresponding property

and proof. Edge segmentation is then performed. If two non-collinear edges intersect, they are split into four edges and a class II vertex is created for the intersection. Following, face construction is done for both curved and planar faces. In this step, all possible faces are constructed. Cutting edges are then inserted in between two faces that may intersect one another. Now, candidate block construction is done, similar to [3] where a set of smaller sub-objects are made from the available faces that can be used to reconstruct the final larger object, or other possible objects as well. Finally, decision making is performed such that a final object is a union of candidate blocks, so a validation check is done for which subset of blocks can be glued together to form the final object.

The last orthographic projection article presented for this report is by Liu et al. who used a matrix based approach for reconstruction 3D images [82]. The proposed algorithm consists of a wire-frame for recovering the 3D object from three 2D orthographic views. The algorithm is broken into three steps: Preprocessing the 2D orthographic engineering drawings, reconstructing a wire-frame, then reconstructing the 3D solid object. Besides input validation, the preprocessing step covers a few cases. If two 2D curves from separate views intersect, a new point in the view-space must be created to represent this. Also, center lines and axes are added to each view. Following, reconstruction of the wire-frame is done in which all possible 3D vertices and edges are constructed from the three views. Finally, the reconstruction of the 3D solid object is done using the now constructed wire-frame. First, a set of candidate faces are found with respect to the wire frame using a maximum turning angle method. This includes all surfaces, faces, and edge loops. Then, a search is performed to find the 3D solid. Specifically, all possible solids that can be constructed from the generated set of faces are found. This is done based on the Moebius rule, which dictates that each edge in a manifold belongs to two faces. The implementation of the authors algorithm shows that it is possible for a wireframe to give a single 3D object, or multiple solutions for a valid 3D object.

Orthographic projections as a form of 3D modeling is an older method that helped greatly improve CAD software. While it is a useful tool that allows for taking multiple angled images to recreate a 3D image, the assumptions it follows makes it difficult to utilize outside of strict engineering applications. Also, it cannot account for moving objects in real-time as in the case of the heart.

2.2.4 Single Image Pixel Depth 3D Reconstruction

Traditionally, measuring depth from 2D images is done with stereoscopic images [26]. Though, that requires a stereo camera setup and corresponding measurements. Furthermore, if the stereo camera setup is not pre-constructed, calibration needs to be done. So, using a single image to measure depth can be very beneficial. There have been many articles written to solve this problem with respect to real world images.

First, an article by Liu et al. calculates the depth of a single image from semantically labeling regions on the image [80]. The proposed algorithm is divided into two steps. First, predict the semantic class of every pixel in the image as well as the horizon (as this is focused on outdoor images). Second, estimate the depth in the image with respect to the semantic labels. To predict the pixel wise semantic labels, a standard pairwise Markov Random Field (MRF) is implemented. This predicts the pixels on the scene that are not corresponding to the horizon and the sky. The horizon prediction is done with an unnamed algorithm produced from Gould et al. [50]. With the image semantically labeled and the horizon found, the depth estimation now occurs. A different local depth predictor is learned for each semantic class (sky, ground, etc.). Specifically, the pointwise log-depth is estimated as this is more accurate for relative depth in scene understanding. The features used for estimation of the depth are the 17 raw filter responses of the semantic model, the log of the 17 raw filter responses, the geometric constraints of the scene (prior log depth and horizon adjusted vertical coordinates), and the square of each feature. The log-depths are

finally estimated via linear regression. Lastly, 3D reconstruction is performed with MRF as well.

An article by Saxena et al. used a supervised Machine Learning approach to estimate the depth of a single monocular image [116]. Specifically, they utilize a MRF as the algorithm for this problem. A 3D distance scanner is used to collect a ground truth labeled dataset which is used to train the MRF. Each image is sub-divided into a matrix and each cell is depth measured individually. The feature vector generated for this problem consisted of texture variations, texture gradients, and haze. For Texture variations, the Laws' mask was applied to the image intensity channel. The intensity channel is convolved with six oriented edge filters. Finally, Haze is calculated by applying a local average filter to the color channels. Then, the authors delve deeper and calculate these separately for absolute depth of a single cell and relative depth, which is the difference in magnitude of two cells. The absolute depth has the cell specific 34 features, then the features of its immediate, far, and very far neighbors (depending on the scale of prediction). This results in a 646 dimensional feature vector for absolute depth. Relative depth features are then computed with a histogram of each of the 17 filter features, producing a vector of 170 features for each patch. Finally, these are applied to the MRF model and also a Laplacian model as well.

Similar to the Machine Learning approach taken by Saxena et al. in [116], Eigen et al. used a multi-scale deep network for predicting the depth in a single image [32]. The authors utilize two networks for this task; one to predict depth globally and another to refine that prediction locally. The datasets used are the NYUDepth and KITTI datasets, where they were transformed randomly based on 5 criteria: scale, rotation, translation, color, and flips. Both networks used Rectified Linear Units for activation except for the outputs, which are linear. The global network uses the entire image (after being downscaled by 2) as an input and has 5 convolutional feature extraction layers, and 1 output layer. The output is of 1/4 the resolution of the input image. For the local network, there are 3 convolutional layers

where the first layer consists of a pooling stage for edge features. Interestingly, layer two in the fine network has the output of the global network fed into it as well. This allows for the refinements to occur on a local level. Lastly, a scale-invariant error is minimized in this problem. The authors define the scale-invariant mean squared error in log space as the performance evaluation. For training, they used a similar scale-invariant error as training loss. The authors showed state of the art results on both datasets.

Utilizing a single image for 3D reconstruction on a pixel by pixel level is impressive and useful in many scenarios. Though, the major drawback is while it is 3D and a point cloud is produced, when the image is turned 90°, then it is not comprehensible. Therefore, more views are needed in order to create a full 3D reconstruction of the object or scene.

2.2.5 3D Modeling Algorithms Maturity Evaluation

We now present the subjective maturity evaluation for the four discussed 3D modeling methodologies of stereo vision, laser scanning 3D reconstruction, orthographic projections, and single image pixel depth 3D reconstruction. For the maturity metric, we have accumulated 11 features which represent different aspects that are important to us in our evaluation presented in Table 2.4. We calculate a maturity metric based upon these 11 presented features and their corresponding weights to describe how mature a methodology is from two perspectives. That is, with respect to the user's perspective and the methodology developer's perspective. The calculation of the maturity metric utilizes Equation 2.1, the same equation used in the medical imaging maturity evaluation in Section 2.1.5. Table 2.5 presents the feature weights associated with both perspectives. The weights range from 0.1 to 1 where 1 indicates that the feature has a significant role in the overall maturity. Following, Table 2.6 shows features scores from 1 to 5 and the calculated maturity. A feature with a score of 5 shows high performance while a feature with a score of 1 shows low

Table 2.4: The list of features to subjectively evaluate the four 3D modeling methodologies of stereo vision, laser scanning 3D reconstruction, orthographic projections, and single image pixel depth 3D reconstruction. The abbreviation VFn next to each feature stands for "Computer Vision Feature n".

Feature	Description
Accuracy (VF1)	The ability of the method to produce accurate 3D models.
Ease of Use (VF2)	The ability to utilize the method in a non-complicated manner.
Robustness (VF3)	The ability of the method to operate in differing conditions.
Cost (VF4)	How much the method costs to recreate.
Reliability (VF5)	The ability of the method to consistently produce and reproduce 3D models.
Scalability (VF6)	The ability of the method to create models on a growing scale outside its original design.
Complexity (VF7)	The overall complexity of the proposed method.
Completeness (VF8)	How much information is missing from the produced model.
Resolution (VF9)	The overall quality of the produced 3D image.
Multiple Images (VF10)	The ability to ingest multiple images.
Flexibility (VF11)	The ability for the algorithm to ingest images with unknown a priori.

performance.

Finally, we present Figure 2.2, a histogram of maturity ratings of all 12 reviewed 3D modeling algorithm studies. Examining the figure, we can see that no methodology has a perfect maturity rating in either presented perspective. Though, once again, this is an expected outcome. There are also a few interesting notes from both Figure 2.2 and Table 2.6. Starting with the stereo vision studies [55, 103, 118], we can see they performed the highest as compared to the other three methodologies. Most notably due to the flexibility and low cost of the system (simply two cameras at most). Next, for the laser scanner 3D reconstruction studies [17, 109, 127], they had some of the lowest ratings. Particularly, the study by Poncet et al. [109] due to its high complexity and low flexibility. Though, with this study, it is highly situational. Following, for the orthographic projection studies [3, 121, 82], these performed quite will comparatively. This is due to the low complexity of the algorithms expected for their ages. Yet, the scalability of the algorithms hold them back unfortunately.

Table 2.5: The list of feature weights for all 11 features. Feature weights range from 0.1 to 1 where 1 means that the feature has a significant role in the maturity of the methodology. Both the user's perspective and developer's perspective are accounted for as well as the average between the two.

Feature	**Weights**		
	User's Perspective	Developer's Perspective	Average
VF1	1	1	1
VF2	0.8	0.3	0.55
VF3	0.9	0.8	0.85
VF4	0.7	0.2	0.45
VF5	0.8	0.8	0.8
VF6	0.5	0.3	0.4
VF7	0.2	0.8	0.5
VF8	0.8	0.6	0.7
VF9	1	0.8	0.8
VF10	0.8	0.6	0.7
VF11	0.9	0.6	0.75

Finally, for the single image pixel depth 3D reconstruction [80, 116, 32], these algorithms underperformed our expectations. This is due to the two Machine Learning studies having low scalability and flexibility. Outside of the trained networks expected operating data, the systems will have trouble with accuracy performance. Though, within the area of operation that the networks were trained on, the results are very promising. Of the four algorithms, stereo vision performed the highest and therefore we will choose this algorithm to use for the remainder of this work.

2.2.6 Further Examination of Stereo Vision

With stereo vision being the chosen 3D image reconstruction algorithm for this work, we continue with a more in-depth examination of the methodology. Fortunately, there is a plethora of research in the field of stereo vision ranging in both application and theory. A survey about stereo vision on low resource systems was created by Tippetts et al. [132]. In this survey, they discuss stereo vision research published into three categories: real-time

Table 2.6: The assigned feature values for all 11 features and the corresponding maturity calculation for both the user's and developer's perspectives. The feature values range from 1 to 5 where a value of 5 shows the highest performance of said feature.

Features	**Scores**											
	[55]	[103]	[118]	[17]	[109]	[127]	[3]	[121]	[82]	[80]	[116]	[32]
VF1	4	4	4	4	3	4	4	4	4	3	4	4
VF2	4	3	3	3	2	3	4	4	4	3	3	3
VF3	2	3	4	3	2	4	2	2	2	3	2	2
VF4	5	4	4	3	2	3	4	4	4	3	4	4
VF5	4	4	3	3	3	3	4	4	4	4	4	4
VF6	3	2	4	4	2	4	2	2	2	2	2	2
VF7	3	3	3	2	2	2	3	3	2	3	3	2
VF8	4	4	4	3	4	4	4	4	4	4	4	4
VF9	4	4	4	3	3	4	4	4	4	4	4	4
VF10	4	4	4	3	3	3	4	4	4	1	1	1
VF11	3	3	4	2	2	3	2	2	2	4	2	2
User's Perspective	3.67	3.54	3.78	3.05	2.61	3.46	3.42	3.42	3.39	3.15	3.02	3.00
Developer's Perspective	3.54	3.54	3.72	2.99	2.65	3.40	3.38	3.38	3.27	3.19	3.07	2.96
Maturity Average	3.61	3.54	3.75	3.02	2.63	3.43	3.40	3.40	3.33	3.17	3.05	2.98

or near real-time results on standard hardware, real-time results on specialized hardware (Graphics Processing Unit (GPU), Field Programmable Gate Array (FPGA), etc.), and those that do not have real-time results. We will examine some interesting examples of the first two categories now as they pertain to our research goal.

We first examine an article by Kalarot and Morris that reached impressive real-time performance in two different manners of a FPGA and GPU, respectively [68]. We note that it is common for medical imaging devices to utilize one of more GPUs to process the gathered data. Starting with the FPGA implementation, the authors built a custom FPGA containing three separate blocks. First, a distortion removal and rectification block to assist camera alignment as real lenses distort images which makes configuration difficult. Next, disparity calculators are used for creating the disparity map with calculations down to 15ns (nanoseconds). Finally, a predecessor array and backtrack module is included which stores data and allows for a backwards depth profile construction. Additionally, a GPU implementation was also created on a GTX 280 using nVidia's Compute Unified Device

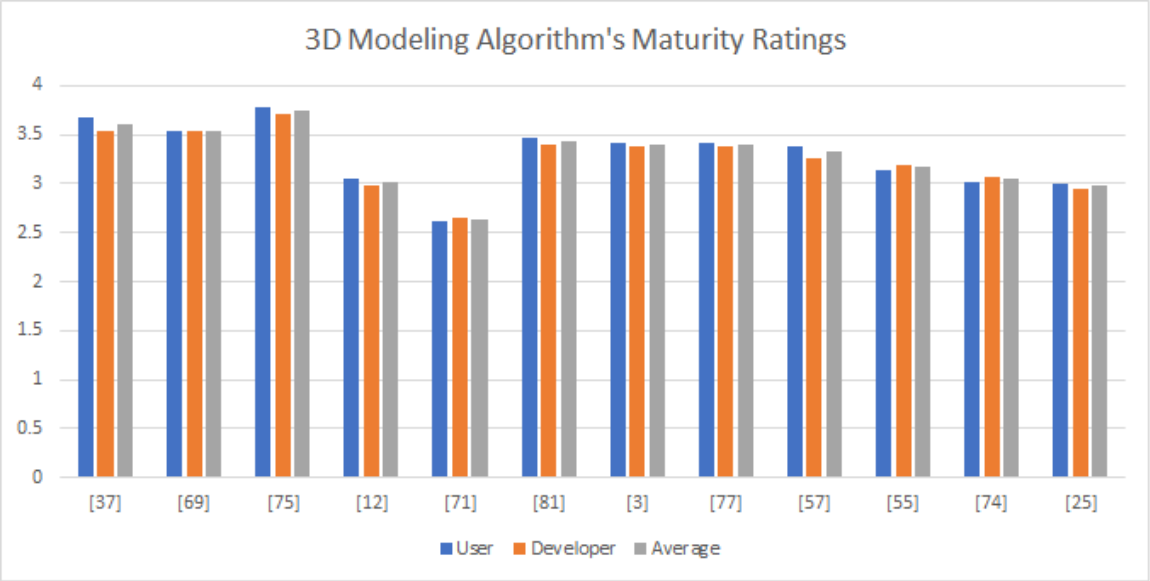


Figure 2.2: Maturity scores for all 12 3D modeling methodologies. The maximum maturity rating is 5 while the lowest maturity rating is 0.

Architecture (CUDA) language. This language allows for direct control on memory access and functionality of the GPU. The comparison shows the FPGA out-performing the GPU in their problem. Though, the authors note that even state-of-the-art FPGAs have space limitations while GPUs have significant internal overheads which slow it down but can handle double the size of disparity ranges.

Following, an article by Cuadrado et al. also utilizes an FPGA to obtain real-time stereo vision results [25]. Specifically, the authors propose the Real-Time Stereo Vision Processor (RTSVP). This digital architecture contains a matching algorithm based on the correlation between the two windows as the sum of absolute intensity difference for every pixel. First, the FPGA implementation contains a rectification block, similar to [68]. Second, a correlation window delayer with First In First Out (FIFO) memory and registers is used to correctly order the pixel intensities for use with the correlation measure. Next, a block is designated for the stereo disparity correlators, which takes two intensity values in the reference image and two in the disparity image to compute the correlation of each. Finally, a disparity comparator and check block is included to take in as input the correlation

values from the previous block and output the corresponding disparity value. All of these can be placed on one device.

Another real-time stereo vision GPU implementation is produced by Yang et al. [143]. In this article, the authors discussed using the existing Hierarchical Belief Propagation algorithm and creating an extension called Constant-Space Belief Propagation (CSBP). Their CSBP algorithm computes the data term at each level of disparity which costs time but saves on memory storage such that the data term no longer depends on the maximum disparity. This is beneficial for higher resolution images. Furthermore, their algorithm gradually decreases the number of disparity levels as the spatial resolution increases. Their CSBP algorithm is thus memory independent of the total number of disparity levels allowing for larger resolution images to be handled without speed decrease via a GPU.

Another study related to laparoscopic surgery is performed by Kang et al. [69]. Here, the authors use the stereoscopic laparoscope in conjunction with ultrasound imaging to create an Augmented Reality (AR) view. The 3D images are applied on a plane over the currently visible images in real-time. This new AR view allows surgeons to see within organs and tissue without having to make deeper incisions. While the laparoscopic surgery is still invasive, this system reduces the depth of incisions needed due to the addition of ultrasound imaging and avoids the time consuming imaging of MRI and the radiation doses of PET and CT scans.

Vergauwen et al. created an interesting stereo vision approach to planetary exploration for use by the European Space Agency [136]. The main contribution by this study comes from the calibration system as the planetary entry flight almost guarantees misalignment of the stereo camera setup. The authors use the concept of the essential matrix to calculate the baseline distance between both cameras and the epipolar geometry. Features are applied to each image by using the Harris corner detector. These features assist in the epipolar geometry calculation. The corners are matched between both images using cross-correlation.

Random sampling consensus is used to iteratively update the epipolar geometry. Finally, a matrix F is computed to represent the epipolar geometry. This F matrix is used in the calculation of the relative transformation between both stereo cameras. Pan and tilt calibration is also needed. The same algorithm for computing the epipolar geometry is used here as well. The overlapping images are slightly different as the angles differ due to the camera angles of the pan and tilt, respectively. Finally, the 3D reconstruction of the scene was similar to other stereo vision studies. That is, a disparity map is computed to obtain image correspondence. Though, one image may be transformed in order to decrease disparity ranges. The authors devised a way to calculate the calibration of the stereo camera setup and compute a 3D elevation map for the remote planetary rover to navigate successfully.

Further examinations of stereo vision show the multiple areas of interest with respect to application domains (space exploration [136], laparoscopic surgery augmented reality [143], etc.) as well as algorithmic and physical architecture speed increases for stereo vision [68, 25, 143]. It is obvious that stereo vision is still being improved to this day and reinforces our choice of utilizing it for this work.

3

Wearable Ultrasound Transducer Vest

In this chapter, we present the wearable ultrasound transducer vest with both the 4x2 and 4x3 transducer orientations. We begin with a brief overview of the anatomy and physiology of the heart in Section 3.1 to understand the human heart. Then, we discuss the specifications of an average human heart for the measurements needed to design the vest and the corresponding operating characteristics in Section 3.2. Next, we examine the 4x2 transducer orientation in Section 3.3. This section contains the vest location and transducer description in Section 3.3.1, the produced stereo overlap from the transducers in Section 3.3.2, and the generalized mathematical model and operation specifications in Section 3.3.3. Following, Section 3.4 discusses the 4x3 transducer orientation. Similarly, this section includes the new vest location while including an additional column of transducers in Section 3.4.1, the new produced stereo overlap in Section 3.4.2, and the mathematical modeling and operation specifications with 12 instead of 8 transducers in Section 3.4.3. Finally, Section 3.5 provides a brief comparison between both the 4x2 and 4x3 transducer orientations.

3.1 Macroscopic Anatomy and Physiology of the Heart

As discussed in Chapter 1, the heart is one of the most important organs within the human body as it is essential to life. Thus, it is important to understand the heart before attempting to model it. Beginning with the macroscopic anatomy, the heart is contained within the Pericardium sac which also contains a small amount of Pericardial fluid for lubrication that allows the heart to function freely [100, 53]. Next, as shown in Figure 3.1, the heart is broken into 4 chambers which consist of the right atrium, right ventricle, left atrium, and left ventricle. There are 4 valves as well located within the heart. The Tricuspid valve connects the right atrium to the right ventricle. The Mitral valve connects the left atrium to the left ventricle. The Pulmonary valve connects the right ventricle with the Pulmonary artery. Lastly, the Aortic valve connects the left ventricle to the Aorta. Externally, the Superior (upper) and Inferior (lower) vena cava connect from the organs and extremities to the right atrium. The Pulmonary artery connects from the right ventricle to the lungs. The Pulmonary veins connect from the lungs to the left atrium. Lastly, the Aorta connects from the left ventricle outwards to the rest of the organs and extremities. As for external obtrusions to the heart, as shown in Figure 3.2, 7 ribs of the thoracic cage (i.e., the rib cage), the sternum, and 12 thoracic vertebrae protect the entirety of the heart [15].

Next, the physiology of the heart (i.e., the blood flow, contraction and relaxation, etc.) is presented. The hearts primary function is to circulate blood throughout the body to the organs and the extremities [53, 15]. To do this, the oxygenated blood is pumped to the organs and extremities where they take the oxygen from the blood before it is sent back to the heart deoxygenated. The cardiovascular system acts as a cyclical system where they cycle works as follows.

- The oxygenated blood is pumped from the left ventricle to the Aorta which is then distributed throughout the organs and extremities of the body.

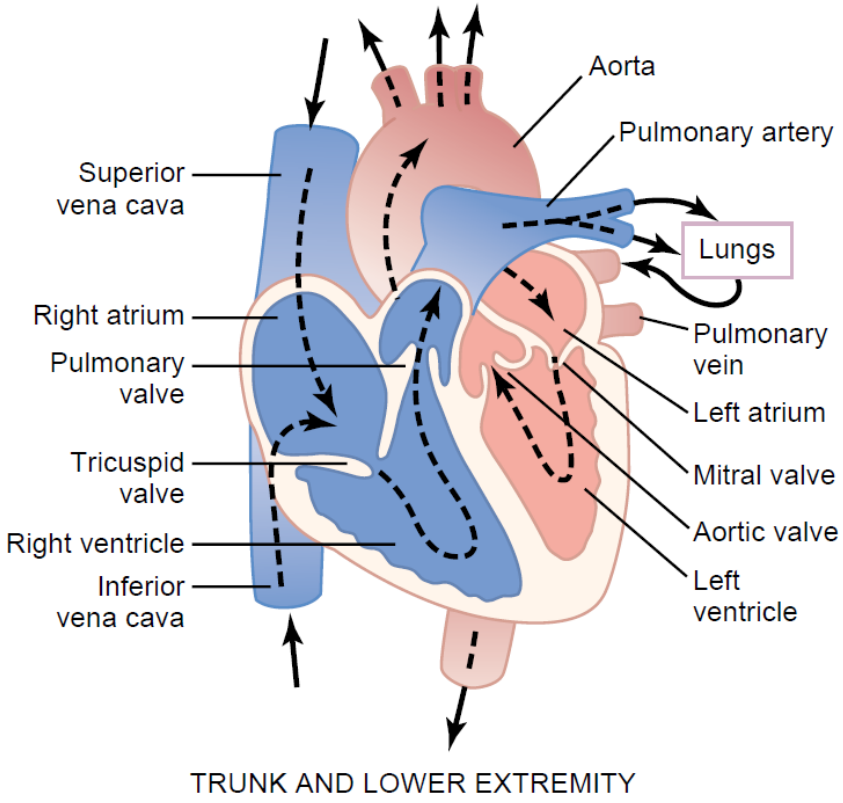


Figure 3.1: The labeled anatomy of the heart. Figure courtesy of Guyton and Hall [53].

- The organs and extremities take the oxygen from the blood, deoxygenating it, where it is then pumped back to the heart. The deoxygenated blood goes through the Superior or Inferior vena cava (depending if it came from the top or bottom of the body) into the right atrium.
- The deoxygenated blood is then moved to the right ventricle to be pumped through the Pulmonary artery to the lungs to be reoxygenated.
- The newly oxygenated blood is moved back through the Pulmonary veins to the left atrium, then finally to the left ventricle for the cycle to be repeated.

The entire cardiovascular cycle is done by contractions and relaxations, which is what is commonly known as the heart beat. The heart begins in the relaxation phase where both the left and right ventricles expand and receive the blood stored in the left and right atrium, respectively. This is also referred to as diastole. Following is the contraction phase

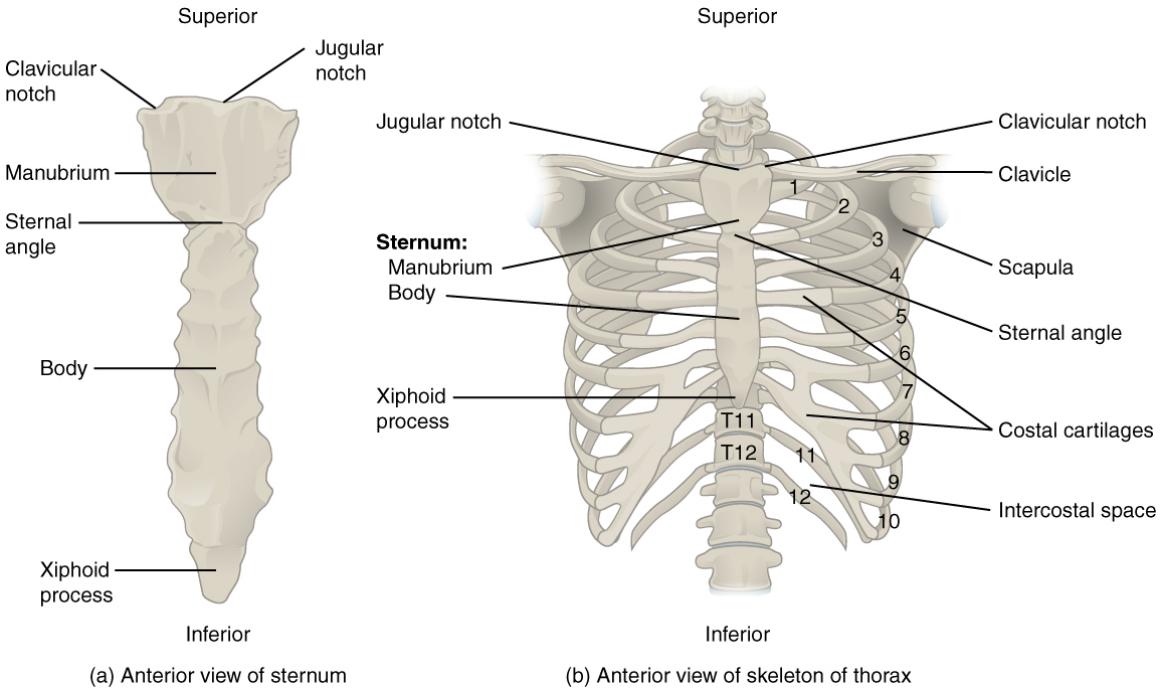


Figure 3.2: The labeled anatomy of the thoracic cavity. Figure courtesy of Betts et al. [15].

in which both the left and right ventricle contract and rapidly pump the blood stored in the ventricles through the Aorta and Pulmonary artery that connect to the left and right ventricles, respectively. This is also referred to as systole.

3.2 Specifications of a Human Heart & Thoracic Cage

There are many measurements of the human heart we need to take into account when designing the wearable ultrasound transducer vest. Specifically, we are most interested in measurements pertaining to the three viewpoints of parasternal (the top-front of the chest angled behind the sternum), frontal (45° to the right of the parasternal view between the direct front of the chest and left arm), and apical (under the left arm). The necessary measurements are as follows.

- The number of ribs and intercostal spaces (the space between two rib bones) which directly obstruct the heart.

- The average size of each intercostal space.
- The average height of individual ribs and their corresponding downward angles.
- The average Body Mass Index (BMI) of a healthy adult.
- The average distance from the skin to the heart.
- The average breaths per minute of a healthy adult.
- The average Beats Per Minute (BPM) of the heart of a healthy adult.
- The average dimensions of a human heart.

We begin with stating the hearts location within the chest and the number of ribs and intercostal spaces that obstruct the heart. The heart is located in the center of the chest with a slight left offset. There are a total of 12 ribs in the thoracic cage and 11 intercostal spaces [87]. As stated in Section 3.1, of these rib bones, true ribs (ribs that connect directly to the sternum) 2 through 6, inclusive, are directly obstructing the heart [87]. Therefore, intercostal spaces 2 through 5, inclusive, are the spaces which will give us the most direct visibility of the heart. We note that the intercostal spaces are ordered craniocaudally (from head to toe). So, intercostal space 2 is between true ribs 2 and 3, intercostal space 3 is between true ribs 3 and 4, etc.

Following, we examine the average size of each intercostal space that provides visibility of the heart. A study performed by Dewhurst et al. measured the intercostal spaces of 50 patients [30]. Their findings for intercostal spaces 4 and 5 showed a mean width and standard deviation of 12.90 ± 2.64mm and 15.40 ± 2.96mm, respectively. Dewhurst et al. also mentioned an important observation that the intercostal space's widths increased craniocaudally. Following, a study by Kim et al. measured intercostal spaces from 466 patients via a sonographic examination [73]. Their findings showed mean widths and standard deviations of 18.3 ± 3.4mm, 17.55 ± 3.6mm, 15.95 ± 3.8mm, and 17.2 ± 3.7mm for intercostal spaces 7 through 10, respectively. Unfortunately, we are unable to find informa-

tion regarding the measurements of intercostal spaces 2 and 3. Though, from the remarks by Dewhurst et al. stating that intercostal spaces increase in width craniocaudally [30], we can make worst case scenario assumptions. That is, we take the measurement of intercostal space 4 and round this to 13.00mm to use as the measurement for intercostal space 2 and 3.

Now, we need to examine the height of the individual rib bones and their corresponding downward angles to know where in the vest the ultrasound transducers should be placed. Beginning with the heights, Abrams et al. performed a study in which they measured the heights of rib bones 3 through 9 at four different cross sections [1]. Of the four cross sections, we are interested in 2 (apical), 3 (frontal), and 4 (parasternal). The authors state that the average heights and standard deviations are 13.5 $\pm$ 3.1mm, 12.1 $\pm$ 2.9mm, and 12.2 $\pm$ 3.1mm for cross sections 2, 3, and 4, respectively. As for the angles of the rib bones, they differ depending on the view of the thoracic cage. That is, looking at 3 different views gives us three different angles. A study conducted by Dansereau and Stokes showed the 3D measurements of the thoracic cage, which included the average angles and lengths of the rib bones [28]. So, from the parasternal view, the connections of the ribs to the sternum do not have an angle which will effect transducer placement. The downward frontal view angle (the angle of the rib viewed from the front of the thoracic cage before connecting to the sternum) for ribs 2 through 6 are 19.5 $\pm$ 9.2°, 16.9 $\pm$ 8.4°, 13.6 $\pm$ 7.0°, 12.1 $\pm$ 6.5°, and 10.4 $\pm$ 6.0°, respectively. Then, the downward lateral view angle (the apical angle of the rib) for ribs 2 through 6 are 35.2 $\pm$ 10.3°, 34.5 $\pm$ 9.1°, 35.2 $\pm$ 8.8°, 36.1 $\pm$ 7.8°, and 37.2 $\pm$ 7.7°, respectively. Finally, the lengths of ribs 2 through 6 are 203 $\pm$ 28.5mm, 254.8 $\pm$ 25.6mm, 289.1 $\pm$ 24.8, 304.8 $\pm$ 28.9mm, and 313.8 $\pm$ 31.7mm, respectively. With these measurements, we can effectively place the transducers correctly in the vest to match the intercostal spaces.

Next, one of the important aspects of ultrasound is knowing the depth for correctly imaging. So, we need to find average measurements for the distance from the skin to the

heart. To do this, we make the assumption that we will be dealing with a healthy adult with an average BMI. The CDC states that a healthy BMI is between 18.5 and 24.9 [35]. With this BMI range, the average and standard deviation skin to heart distance as shown in an article by Rahko is 32.1 $\pm$ 7.9mm and 31.3 $\pm$ 11.3mm for the parasternal and apical views, respectively [113]. Unfortunately, we were unable to find measurements for the frontal view. Though, as the thoracic cage is rounded, it is reasonable to take the average between the parasternal and apical views to be 31.7mm. It is also important to take into account respiration as it is not only the heart that is moving but also the entire chest cavity as well. Barrett et al. states that an individual on average takes 12 to 18 breaths per minute and their chest expands between 50.8 and 127mm from the resting state [10]. Because it is not reasonable to have an individual hold their breath consistently, we slightly mitigate this extra distance by averaging out the skin to heart distances to 40mm for all three views.

Finally, for the heart itself, we first examine the average BPM as our goal is to be able to 3D model it in real-time. So, we need to account for the systolic and diastolic movements and states of the heart. From an article by Nanchen, we see an average BPM of between 50 and 90 [99]. For the sake of this work, we will be taking the worst case scenario of 90 BPM for future modeling. As for the dimensions of the heart, Dr. Gray states that an average size of the heart for an adult is approximately 120mm in length, 90mm in breadth at the broadest area (i.e., close to the middle), and 60mm thick [51].

3.3 4x2 Transducer Orientation

In this section, we discuss the first of two potential ultrasound transducer orientations. That is, a 4x2 ultrasound transducer orientation is presented in Section 3.3.1 as well as the specific ultrasound transducer design [45]. Following, we provide the stereo overlap produced from the FOV of the 8 ultrasound transducers in Section 3.3.2. Finally, we provide a generalized mathematical model in Section 3.3.3 to show the proof of the operating theory such

that the transducers will fire individually without conflict in the time necessary to 3D model the different states of the heart.

3.3.1 Vest Location & Transducer Description

For the wearable vest and its location, it is important to note the approximate position of where the vest lies around the chest as well as where the transducers are placed within the vest in order to have vision through the intercostal spaces. For the 4x2 transducer orientation, the vest will be composed of two compartments which are positioned over the parasternal and apical views, respectively, as shown in Figure 3.3. The vest itself will be elastic based and wrap entirely around the chest of the individual and tightened to fit. The two mentioned compartments will contain a vertical column of four transducers each and an internal placement for the ultrasound transmission gel.

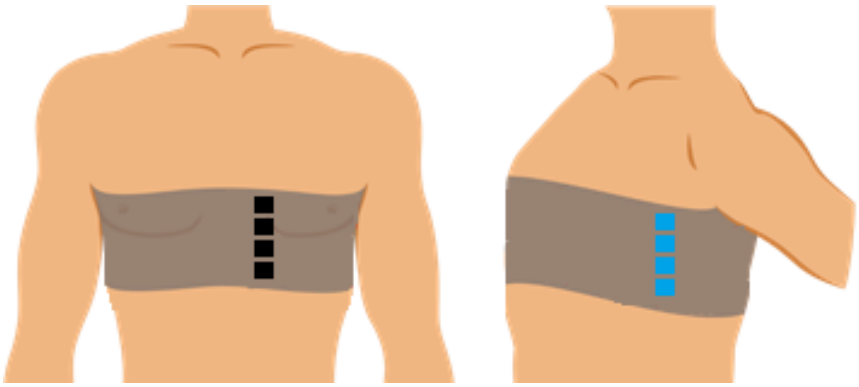


Figure 3.3: The approximate position of the wearable ultrasound transducer vest showing the 4x2 transducer orientation. The squares represent the transducers location within the vest over the intercostal spaces 2 through 5, inclusive.

As the vest and transducer positions have been visually shown, we now need to know specifically what ultrasound transducer design we will utilize moving forward. An article by Tsakalakis and Bourbakis, discussed previously in the literature review, presents an ultrasound transducer design which we will utilize [133]. That is, a 2D circular phased array ultrasound transducer. This transducer was of dimensions 14.4x14.4mm and is comprised

of 256 0.9x0.9mm piezoelectric elements. The FOV of this transducer is 45° in both the azimuthal and elevation directions. We make one small modification to this design in that instead of a circular transducer, we utilize a square transducer. The entire design is presented in Figure 3.4 A. There is an immediate problem of the size posed by this design though. The smallest size of an intercostal space as mentioned in Section 3.2 is 13.00mm. Therefore, we need to adjust the design slightly to account for this. Instead of physically modifying the number of piezoelectric elements, we instead choose to filter them as shown in Figure 3.4 B. That is, by simply not activating the top and bottom rows of the transducer which limits the height to 12.6mm and the number of piezoelectric elements to 224. This filtered variation of the transducer is just small enough to have vision through the intercostal space. We choose this route over physical modification as this work is based upon the assumption of human average. This includes average height as well. Given an individual that is significantly taller than the average, their thoracic cage dimensions will be larger as well. So, for these special cases, we can simply reactivate one or both piezoelectric element rows as needed.

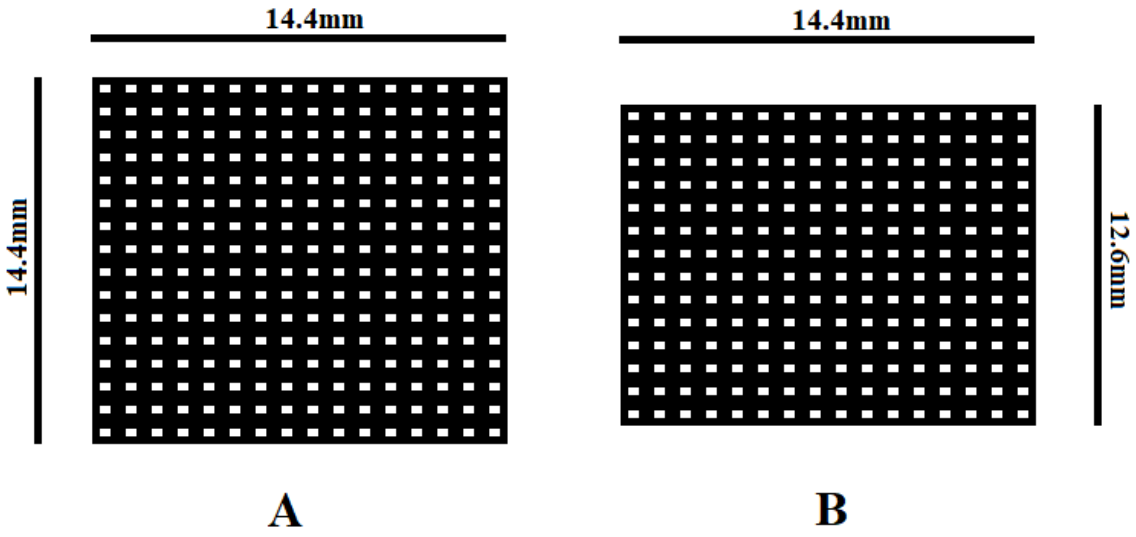


Figure 3.4: A) The 14.4x14.4mm 2D square phased array transducer with 256 piezoelectric elements. B) The filtered 2D square phased array transducer down to a 14.4x12.6mm size with 224 active piezoelectric elements.

Furthermore, from applying the filter, the amount of information lost is low. From Equation 3.1, we have two variables $filterHeight$ and $height$ which equal 12.6mm and 14.4mm, respectively. Therefore, $Loss$ is equal to 12.5%.

$$Loss = 100 - (100 * (filterHeight/height)) \tag{3.1}$$

This loss is negligible in our case though as we can recover this information. Due to the FOV of 45° in both azimuthal and elevation directions, the middle piezoelectric elements of the transducers have visibility behind the rib bones above and below the transducer as shown in Figure 3.5. This realization allows us to apply the filter to the ultrasound transducer and still recover most of, if not all, missing information.

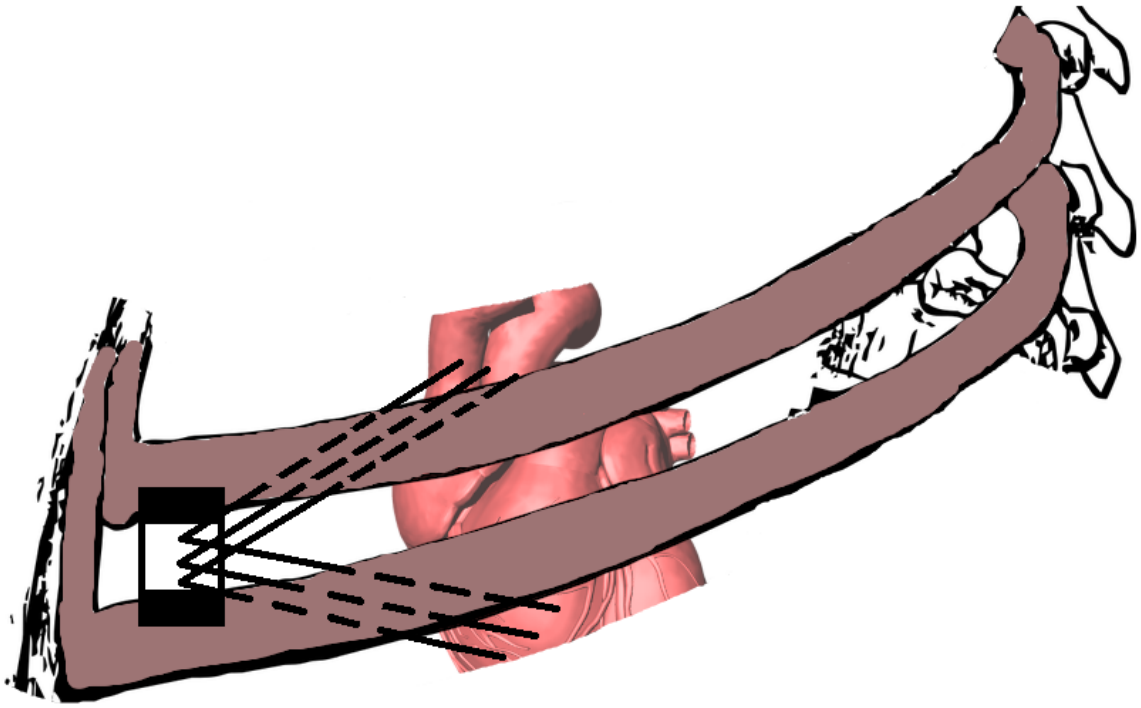


Figure 3.5: The FOV produced from the middle row's piezoelectric elements. The 45° azimuthal and elevation projections allow for data capturing behind the rib bones that would otherwise be blocked from the top and bottom filtered elements.

There are other problems associated with the ultrasound transducers though that we

do acknowledge. Specifically, in the forms of three main artifacts that can possible be obtained from a single ultrasound activation. First, speckle noise, which is the multiplicative constructive and destructive interferences of the returning scattered signal [64]. This shows literal specks on the produced image. Second, attenuation mismatch artifacts, which is when the attenuation of the transducer is different than what is expected. For example, an object such as a bone interferes with the ultrasound wave [33]. Third, side lobe artifacts, which is when energy leaks from the sides of the main ultrasound beam and form what are called "side lobes" [33]. These leaked energies can either be absorbed into the tissue or, if the object in which the energy interacts with has a strong reflector in it (a bone perhaps), it will reflect back to the ultrasound transducer. The reason this is considered an artifact is because ultrasound functions under the assumption that the reciprocated ultrasound echoes come from only the main beam. Finally, artifacts can also be produced from implanted medical devices such as prosthetic valves, ventricular assist devices, and more [112]. Fortunately, there are already known ways of avoiding or compensating for these known artifacts [33].

3.3.2 Produced Stereo Overlap

In order to know the produced stereo overlap of the eight ultrasound transducers, we need to first know what the measurements of the produced FOV from a single transducer is. As we are using 2D square phased array transducers, we can examine this problem to be similar to a truncated pyramid as shown in Figure 3.6 A. The base of the pyramid represents the projected FOV we are interested in. So, we must solve for $r2$, the radius of the base, which is $r1$, the radius of the transducer, plus h. Fortunately, we know that the projected FOV is linearly scaled from the transducer and that the FOV in the azimuthal and elevation directions is 45°. This means that the triangle produced from the figure is a 45-45-90 triangle (Figure 3.6 B). The variable h in the figure is the height of the truncated pyramid, or in our case, the average distance from the skin to the heart which is 40mm. Now,

we simply apply the filter of removing the top and bottom piezoelectric element rows of 0.9mm each. The corresponding FOV of the base from this removal is 6.25% from the top and bottom of the square or 11.8mm total. Thus, the radius of the base $r2$ is 40mm plus 7.2mm (the radius of the transducer). This gives a final projected rectangular FOV for a single ultrasound transducer of 94.4mm by 82.8mm.

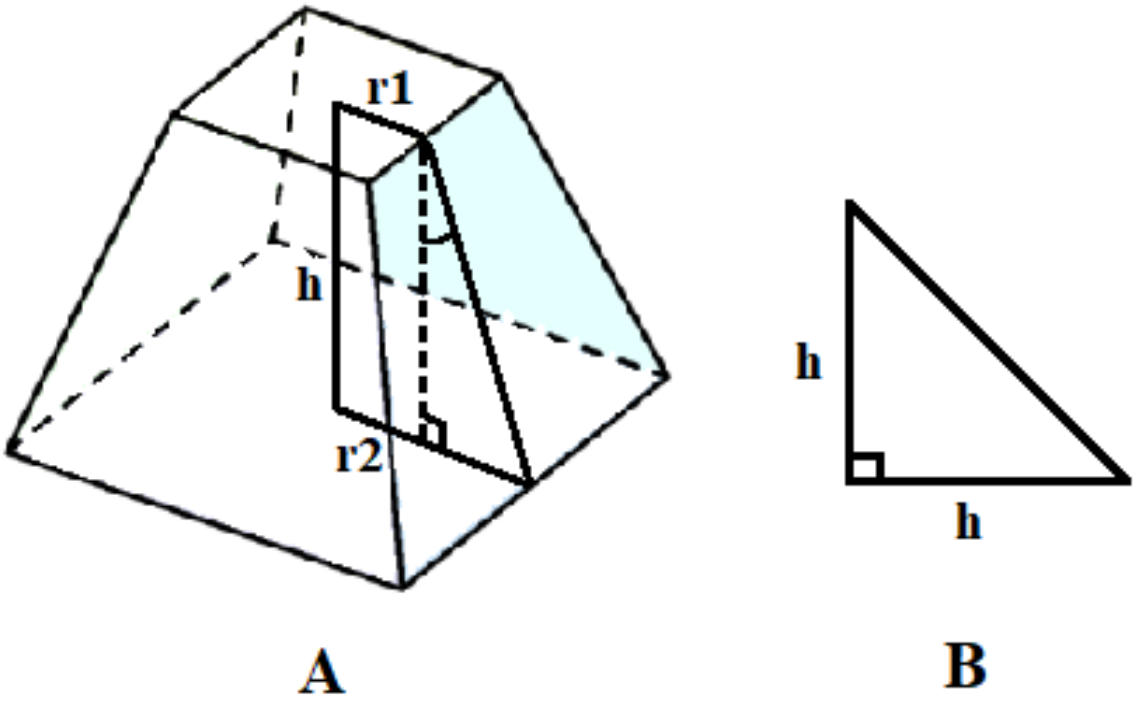


Figure 3.6: A) An example of the ultrasound transducer's projected FOV in the form of a truncated pyramid. B) The 45-45-90 triangle extracted to find the radius of the FOV.

Following, we also discussed in Section 3.2 that there are downward angles associated with the rib bones. This will affect the placements of the ultrasound transducers within the vest. Specifically, their orientation, as if we placed them directly in the vest without turning them some degree, we would receive many artifacts from the rib bones. Therefore, the transducers must be turned to account for these obstructions. For the apical view, we know that rib bone 2 has a downward lateral angle of 35.2°. Therefore, the transducer which is placed over intercostal space 2 will also be turned 35.2° to the left. From here, we simply turn the remainder of the apical transducers 35.2° to the left as well due to them

being placed in a vertical pattern. Between each transducer there is a vertical 13.5mm shift to account for each rib bone as they are thicker in the apical view. Next, for the parasternal view, the connections of the ribs to the sternum have negligible angles and thus we can orient each transducer in a vertical column with no degree turns. The downward vertical shift for the parasternal view is 12.2mm as the rib bones are slightly thinner in this location.

Now, with placement and orientation of the transducers discussed, we must calculate the produced stereo overlap from each column separately and the overlap between each column together as well for finding correspondence points. That is, we calculate parasternal overlap, apical overlap, and the overlap between the parasternal and apical views. We begin with the parasternal overlap. Figure 3.7 shows the total view produced from the four transducers as well as the extracted stereo overlap. From the left image, we can see there is significant stereo overlap for calculating stereo points. Each individual rectangle is 94.4x82.6mm in size. So, from the right image, we know that line segment **AB** is 94.4mm while line segment **AC** is 94.8mm to account for the downward vertical shifts. Therefore, the approximate stereo overlap surface area is 8959.12mm^2. Furthermore, the total parasternal vertical view is approximately 119.2mm.

Following, we have the apical stereo overlap calculations. Instead of being a rectangle area calculation, the angles of the turned transducers changes the overlapping area into a composite object. Recall that each transducer in the apical column is turned 35.2° to the left and have a 13.5mm vertical downward shift in-between each transducer. We can see from Figure 3.8 that the left image shows the total apical viewpoint coverage while the right image is just the stereo overlapping portion. As the stereo overlapping area is a composite object, we only need the lengths of line segments **AB**, **AC**, **BD**, and **CE** which are 86.6mm, 71.6mm, 11mm, and 7.8mm, respectively. From this, we calculate the area of the composite object to be approximately 9047.7mm^2 of stereo overlap. Furthermore, the

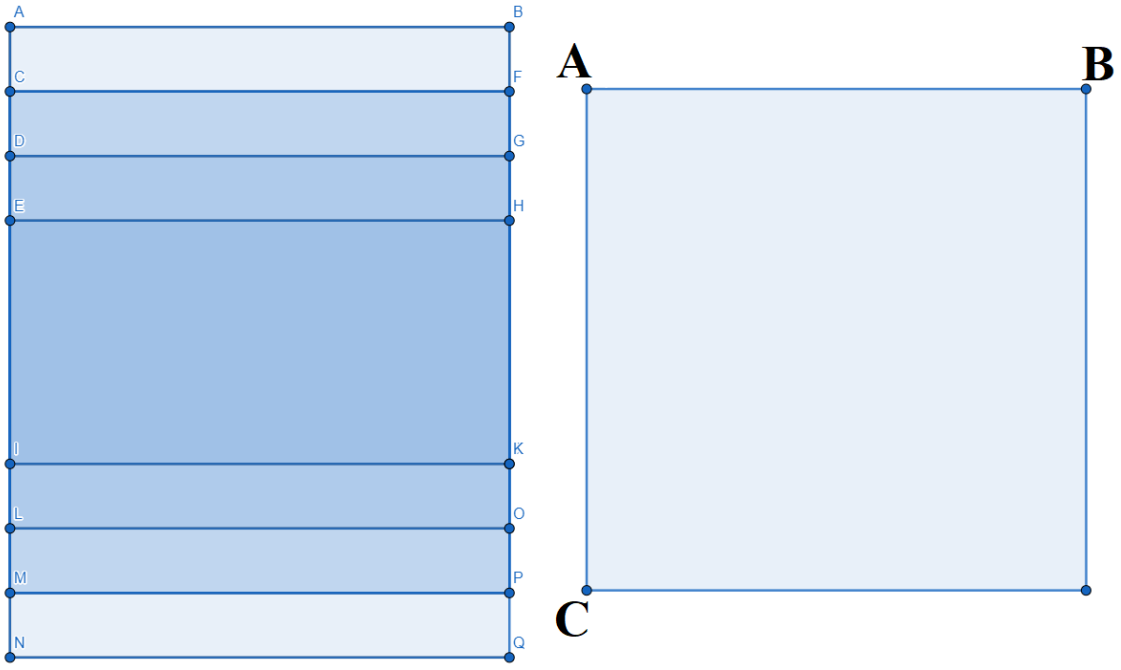


Figure 3.7: The stereo overlap produced from the parasternal column of ultrasound transducers. The left image shows the total view of the four transducers while the right image is only the stereo overlap area minus the non-overlapped areas (the top and bottom lightest rectangles).

total apical vertical view is 161.9mm.

Finally, we have the stereo overlap between the parasternal and apical views. This calculation is the most complex of the three. That is, we must take into account the shape of the heart. To do this, the shape which best represents the heart is a prolate spheroid. This shape, and its corresponding known formulas [153], allow us to approximate the produced stereo overlap between both the parasternal and apical views. Furthermore, as we are dealing with ultrasound waves, we know the projected FOV continues over the curvature of the heart slightly from both views. With this knowledge, we present Figure 3.9 to visually represent this idea. From the figure, there is approximately $1/4$ of the prolate spheroid which is represented from both views. Therefore, we must estimate the total surface area of a prolate spheroid and take $1/4$ of this final value which will be our approximation of the stereo overlap between the parasternal and apical views. We utilize Equations 3.2 and 3.3 to perform the surface area calculations. From Equation 3.2, S is the surface area of

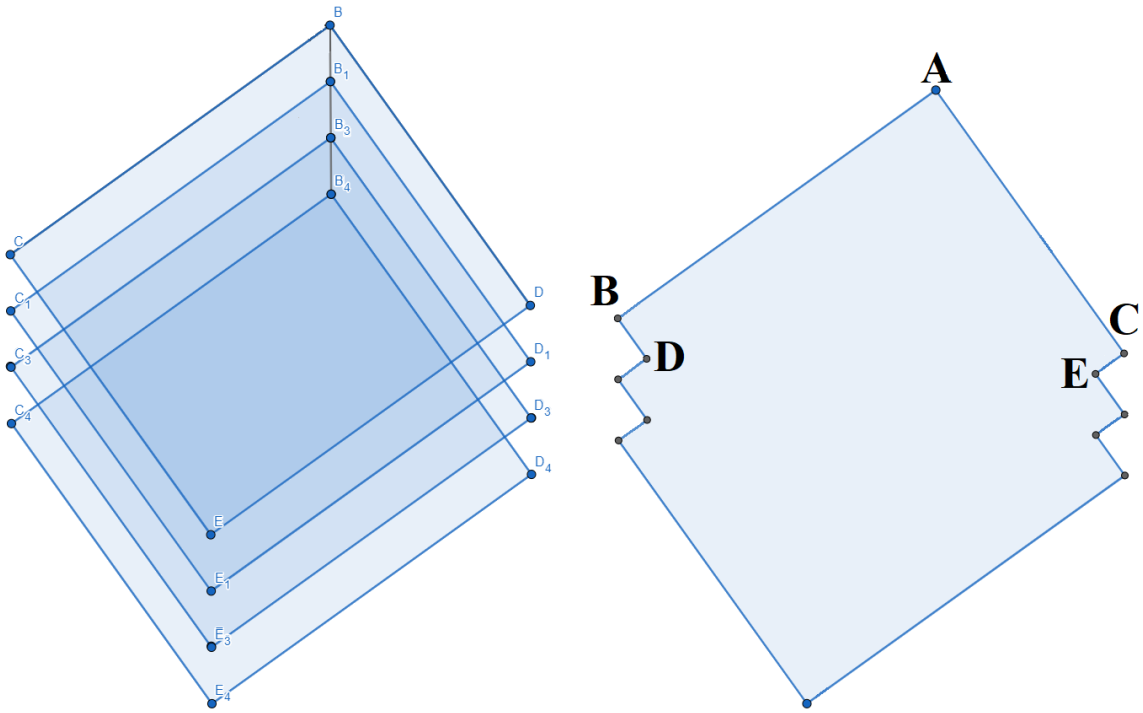


Figure 3.8: The stereo overlap produced form the apical column of ultrasound transducers. The left image shows the total view of the four transducers while the right image is only the stereo overlap minus the non-overlapped areas (the lightest top and bottom portions).

the prolate spheroid, variables a and b are the two semi-axes of the prolate spheroid, and e is the eccentricity of the prolate spheroid. The eccentricity is calculated from Equation 3.3. Variables a and b have values of 45mm and 60mm, respectively, from the measurements of the heart we found in Section 3.2. The total surface area of the prolate spheroid representing the heart is 31260.19mm^2. Then, taking $1/4$ of this value gives a final stereo overlap approximation of 7815.05mm^2 between the parasternal and apical views.

$$S = 2\pi(a^2 + (a * b * e)/sin(e)) \tag{3.2}$$

$$e = arccos(a/b) \tag{3.3}$$

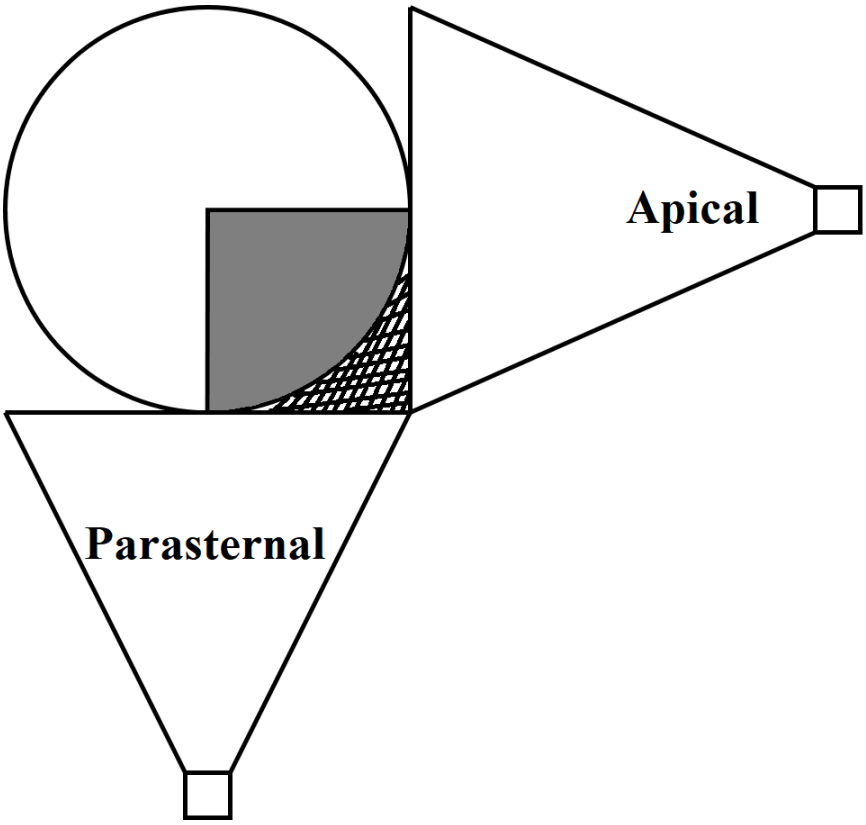


Figure 3.9: The stereo overlap produced from both the parasternal and apical views. This is visualized as a prolate spheroid to represent the human heart. The stereo overlap is then approximately $1/4$ of the surface area of the prolate spheroid.

3.3.3 Model & Operation Specifications

In order to determine if our design is feasible, we have created a set of generalized mathematical formulas to ensure that there are no overlapping activations of transducers and that we are able to get at minimum of 2 or more frames per systolic and diastolic stage. If all eight transducers activated simultaneously, then each transducer would reciprocate ultrasound echoes that were not sent from the original transducer, causing a large amount of noise. Thus, we have formulated Equations 3.4, 3.5, and 3.6 from [66] to ensure our system is capable of both non-overlapping transducer activation as well as frame coverage for both the systolic and diastolic stages.

Beginning with Equation 3.4, the Frame Speed (FS) is how fast a single ultrasound transducer can produce one viewable frame. For this, we need the calculated FOV pro-

jected by a single transducer which is 94.4x82.6mm. The value 1540 is the average speed of sound in meters per second while passing through human flesh [66]. Therefore, we convert the measurements to meters so $xLength$ and $yLength$ in the equation are 0.0944m and 0.0826m, respectively. From this equation, we get FS to equal $5.06e^{-6}$. Following, Equation 3.5 represents the inequality which calculates the Frames Per Second (FPS) that a single transducer is capable of achieving. For this, we require the number of scan lines ($\#ScanLines$) which we retrieve once again from the design of [133] to be 2025. That is, 45 scan lines from both the azimuthal and elevation directions, respectively. With this value and FS from the previous equation, we calculate the FPS to be 97 for a single transducer. Finally, from Equation 3.6, we can calculate the Frame Rate Per Beat ($FRPB$). This equation uses the FPS, the number of transducers in the system ($\#Transducers$), the BPM described previously in Section 3.2, and an aggregate activation interval t. Of these variables, only t is not immediately intuitive. This variable allows for some user specified time between transducer activations to allow for the non-overlapping of activations per minute. The aggregation of each interval equals t. Also, the entire equation is divided by 2 to represent each state the heart can be in. So, we input 97, 8, 2, and 90 for FPS, $\#Transducers$, t, and BPM, respectively. The 2 for variable t allows for approximately 250 milliseconds between each transducer activation. The result of this equation gives us a $FRPB$ of 3.91 which we round to 4.

$$FS = (xLength * yLength)/1540 \tag{3.4}$$

$$FS * \#ScanLines * FPS \leq 1.00 \tag{3.5}$$

$$FRPB = \frac{(FPS/\#Transducers) * ((60 - t)/BPM)}{2} \tag{3.6}$$

We also acknowledge that the breathing of an individual is an issue. Resolution loss will be affected in practice such that the worst case of 18 breaths per minute causes approximately 18/60% (per minute) resolution degradation, or 30% degradation total during the breathing intervals.

3.4 4x3 Transducer Orientation

In this section, we discuss the second potential ultrasound transducer orientation. That is, a 4x3 ultrasound transducer orientation is presented in Section 3.4.1 with one additional column of transducers placed between the parasternal and apical views. We utilize the same ultrasound transducer design as previously shown in Section 3.3.1. Following, in Section 3.4.2, the stereo overlap of this additional ultrasound transducer column is presented as well as the overlap of all three columns. Finally, in Section 3.4.3, we provide the results of the mathematical model shown previously in Section 3.3.3 with the addition of four more ultrasound transducers.

3.4.1 Vest Location

For the 4x3 ultrasound transducer orientation, the only noticeable difference in the approximate vest location and transducer placement comes from the inclusion of the third column of transducers themselves. That is, the third column, which we call the frontal column, is located precisely in the middle between the parasternal and apical views as shown in Figure 3.10. Therefore, we include another compartment to accommodate the third transducer column. We also utilize the same intercostal spaces as the 4x2 orientation as well as the same ultrasound transducer design. The vest will still be made of elastic material to tighten around the patient to fit.

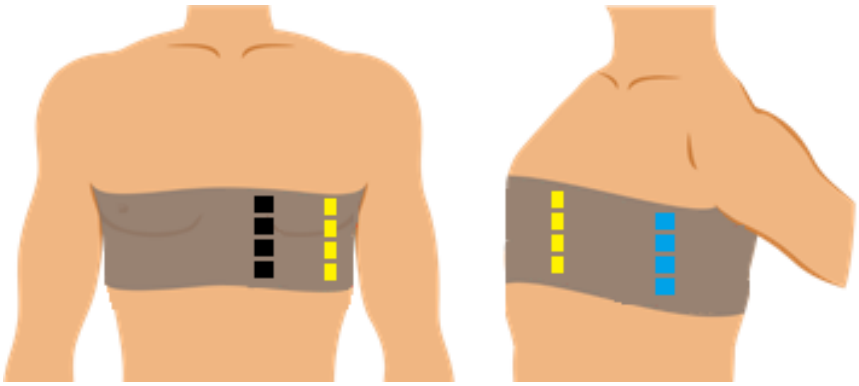


Figure 3.10: The approximate position of the wearable ultrasound transducer vest showing the 4x3 transducer orientation. The squares represent the transducers location within the vest over the intercostal spaces 2 through 5, inclusive.

3.4.2 Produced Stereo Overlap

With the addition of the frontal column for the 4x3 transducer orientation, we get one new additional stereo overlap column and more stereo surface area coverage of the heart between all three views. First, we begin with the frontal column's stereo overlap. This problem is very similar to the calculation of the apical column of stereo overlap such that we have to turn the transducers 19.5° to account for the downward angle at the frontal location. Furthermore, the height of the rib bones at this location is approximately 12.1mm. Therefore, the vertical down shift of each transducer will be 12.1mm. As the produced FOV of a single transducer is still 94.4x82.6mm, the stereo overlap of the frontal transducer column can be seen in Figure 3.11. The left image shows the total view of the frontal transducer column and the right image is the extracted stereo overlap between the transducers. We know the distances of line segments **AB**, **AC**, **BD**, and **EC** to be 94.4mm, 82.6mm, 11.4mm, and 4mm, respectively. Calculating the surface area of this composite object equals 8975.9mm^2 of stereo overlap. Furthermore, the total frontal vertical view is 143.2mm.

Following, we provide the new stereo overlap for the prolate spheroid between the parasternal, frontal, and apical views as shown in Figure 3.12. Once again, we utilize the

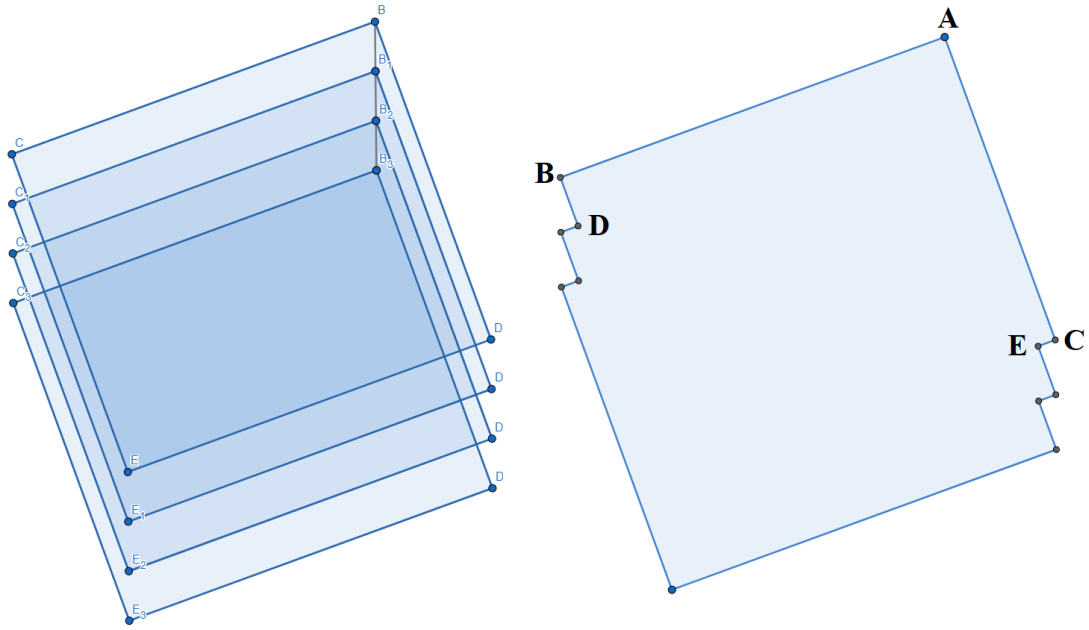


Figure 3.11: The stereo overlap produced form the frontal column of ultrasound transducers. The left image shows the total view of the four transducers while the right image is only the stereo overlap minus the non-overlapped areas (the lightest top and bottom portions).

formulas for the surface area of a prolate spheroid shown in Section 3.3.2. The addition of the frontal column adds more surface area coverage of the prolate spheroid, giving us a better overall view of the heart at a higher resolution as well. That is, instead of only $1/4$ of the coverage between just the parasternal and apical views, the addition of the frontal view increases the coverage to approximately $1/3$ of the surface area. Using Equations 3.2 and 3.3, we still get 31260.19mm^2. Then, taking $1/3$ of this value gives a final stereo overlap approximation of 10420.06mm^2 between the parasternal, frontal, and apical views.

3.4.3 Model & Operation Specifications

Now, we examine the same mathematical formulas presented in Section 3.3.3 to view the feasibility of adding another four transducers to the system. Once again, we want to avoid overlapping activations of transducers and to also obtain at least 2 or more frames per systolic and diastolic stage. Using Equations 3.4, 3.5, and 3.6, we insert the same val-

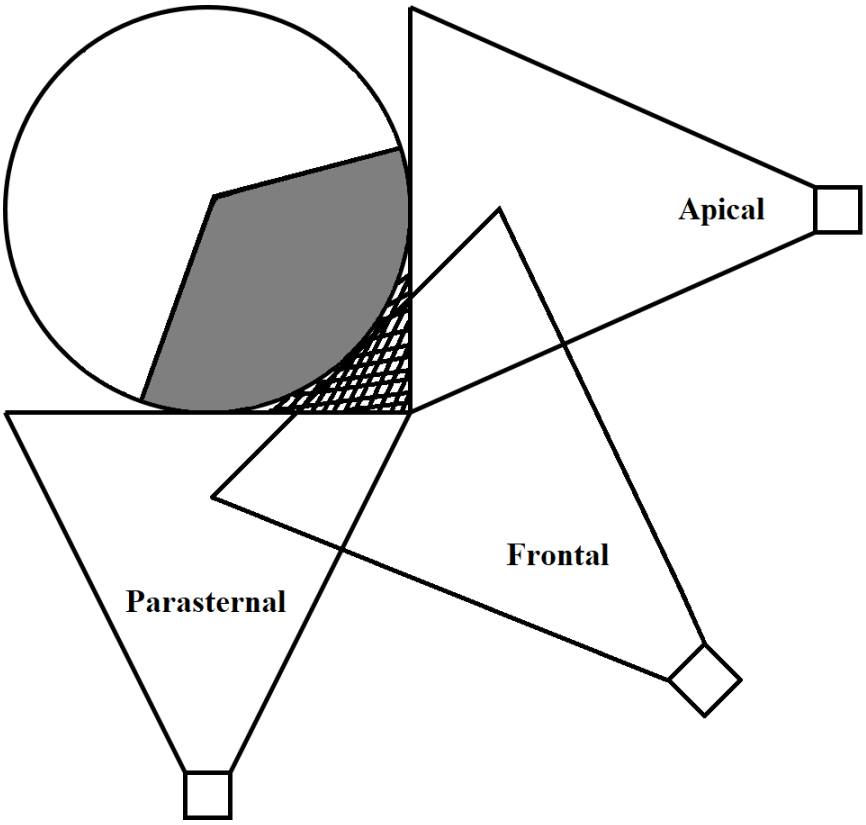


Figure 3.12: The stereo overlap produced from the parasternal, frontal, and apical views. This is visualized as a prolate spheroid to represent the human heart. The stereo overlap is then approximately $1/3$ of the surface area of a prolate spheroid.

ues as previously described. For Equation 3.4, $xLength$ and $yLength$ in the equation are 0.0944m and 0.0826m, respectively. The result is $5.06e^{-6}$. Next, for Equation 3.5, FS and $\#ScanLines$ are $5.06e^{-6}$ and 2025, respectively, resulting in a FPS of 97. Finally, for Equation 3.6, the real notable difference is displayed. That is, for the variable $\#Transducers$, we use 12 instead of 8 as in Section 3.3.3. So, we input 97, 12, 2, and 90 for FPS, $\#Transducers$, t, and BPM, respectively. We now input 3 for t to accommodate the additional 4 transducers. The resulting $FRPB$ is now 2.56 in which we round to 3.

3.5 Comparison Between the 4x2 and 4x3 Transducer Orientations

There are a few notable advantages and disadvantages between the 4x2 and 4x3 ultrasound transducer orientations. Beginning with the 4x2 orientation, we can see from the mathematical calculations that we get approximately 4 FRPB for both the systolic and diastolic states. This allows for a fast system capable of viewing the movement of the heart in real-time. The vision provided by the 4x2 orientation of the heart allows for viewing the four chambers and therefore their activations as well. Though, the resolution might suffer slightly on the overlap between the parasternal and apical views.

Conversely, for the 4x3 transducer orientation, the added four additional transducers lowers the FRPB to approximately 3. This is still more than the minimum we want, which is 2. Though, we are very close to the minimum. This shows a higher complexity system as the trade-off where a higher resolution, specifically in the overlap between the parasternal and apical views, is the improvement. The frontal viewpoint gives an explicit view of the halfway point between the parasternal and apical views which also will increase the resolution slightly of the other two views as well. The addition of the four transducers also will affect the computations of the stereo vision algorithm. It will need to ingest 4 more images per ultrasound vest activation, perform all of the necessary calculations to create a near 3D model, and display the results on a monitor in real-time.

We know the advantages and disadvantages of both setups. The 4x2 orientation is computationally faster, less complex, produces a higher FRPB, but at a lower resolution. The 4x3 orientation is computationally slower, more complex, produces a lower FRPB, but at a higher resolution. Though, both orientations still provide the most important capability which is autonomous modeling. Both orientations will produce the 3D model of the heart on a monitor without the continuous assistance from a medical professional. This allows

for the medical professional to focus more on diagnosing suspected problems rather than imaging the necessary areas. In order to fully understand which model is best, future work will be performing a study with both transducer orientations to gather comparative results to come to a more comprehensive conclusion.

4

3D Modeling Workflow

In this chapter, we begin by providing a representative literature review on Ultrasonography 3D modeling in Section 4.1. Next, we provide the specifications for the general 3D modeling workflow in Section 4.2. Following, we present a rudimentary example using the described workflow in Section 4.3.

4.1 Representative Ultrasonography 3D Modeling Literature Review

We begin this chapter with a representative literature review on Ultrasonography 3D modeling. That is, we examine articles that detail different methodologies of creating a 3D model of a human organ using Ultrasonography. There has been previous work in the field of 3D modeling via Ultrasonography as detailed in and article by Kwon and Gopal [78]. This article details the current progress and future perspectives of 3D and 4D Ultrasound. Using this article, as well as other articles, we can understand the current capabilities of Ultrasonography 3D modeling and how our proposed methodology is unique as compared to other methodologies.

We begin with an article by Soler et al., which designed two methods of creating a real-time 3D model of the left ventricle of the heart [124]. The authors use two different Ultrasonography views aligning with the apical side of the heart to gather volumetric data. Registration is then performed for finding the geometric transformation T that aligns the two views the best. Finally, a comparison between two novel fusion techniques of generalized averaging and multiview deconvolution is presented to create the final model.

Similarly, an article by Rajpoot et al. showed a multiview real-time method for creating a visualization of a beating heart in 3D [114]. More specifically, by taking four single views at the left ventricle apex, the lateral wall of the left ventricle, 1cm from the lateral wall to the interventricular septum, and one intercostal space above and below the optimal apical position. The authors used the same technique as [124] in that they performed registration for finding an optimal T transformation for view alignment. Then, to fuse the images, they proposed using wavelet transformations. Their experimental results showed that using wavelet transformations as the fusion technique provides more features, contrast, and Signal to Noise Ratio (SNR).

An article by Yastrebov et al. examined a more invasive methodology of 3D modeling the heart in the form of intracardiac echocardiography (a method where the ultrasound probe is inserted directly into the heart) [145]. Specifically, they used this method on sheep for implanting a left ventricular assistance device called the Impella CP. Their method is related to use in human patients as sheep have a similar blood clotting time as humans.

Following, Yu et al. proposed a multiview reconstruction using freehand Ultrasonography sparse 2D scanning planes [148]. The authors developed novel methods for image segmentation and registration. First, a hybrid geometric level-set methodology is proposed and optimization techniques are discussed. Then, for registration, the authors create a novel coarse to fine automatic multview registration methodology. Their methodology showed promising results with 1, 2, and 3 views based on a relative volume estimation error. Fur-

thermore, it can handle both end-systolic and end-diastolic phases as well. Though, these results are limited to working with a single ventricle rather than the entire heart.

Next, an article by Legget et al. showed a system for 3D modeling the left ventricle via echocardiography based on magnetic-field position and orientation [79]. The authors proposed using a magnetic-field to track the 3D echocardiography imaging planes. Then, they used the piecewise smooth subdivision method to reconstruct the surface of the left ventricle. Their method is unique as it has the ability to show complex anatomy such as the left ventricle's blood outflow track. Their in-vitro validation showed a high correlation for volume estimation and good shape modeling based on subjective visual inspection.

An article by Song et al. showed integrated surface model optimization using freehand 3D echocardiography of the left ventricle [125]. The authors used a Bayesian framework to address the problem of low image quality by including prior shape knowledge to the equation. The authors utilized 45 total case studies to validate their approach and showed 2-3mm epicardial and endocardial surface projection distance errors.

Ultrasonography is not exclusive to cardiovascular health, it is also used for pregnancy monitoring. Thus, the article by Prager et al. reconstructed a 3D model of the infant for volume measurement by creating voxels from Ultrasonography images [111]. Their results on a 16-week fetus showed high accuracy when compared to an experts hand traced boundaries of B-scan images.

Another methodology for creating a 3D model of the heart is by using electro-mechanical modeling which captures the electrophysiological aspects of the heart using electrodes from an ECG or any other method to capture the electrical potential of the heart [128]. An article by Sermesant et al. uses an ECG to create a simple electro-mechanical model of the heart and provides a framework for extracting quantitative parameters from their model [119]. These parameters can assist in better understanding ventricular functioning and cardiac motion. Following, in a more recent article by Santiago et al., they created a fluid-electro-

mechanical model of the heart using software compatible with supercomputers [115]. Their model's improvement shows active fluid dynamics throughout the interpolated geometry. While impressive, the downside is the computational inefficiency of this method which, according to [119], still needs 4D (a 3D model over time) echocardiographic images for a ground truth comparison.

Finally, an article by Gooding et al. performed volume segmentation and 3D reconstruction for Ovarian Follicle measurement from 3D Ultrasonography data [43]. The authors workflow consisted of segmentation using region competition that uses a dot-product similarity measure for intensity comparisons within regions. Then, surface reconstruction is performed using spatial information. This includes interpolation of the surface to accommodate missing data due to the data being gathered freehand. The final model is reconstructed using voxel space. The results showed that their volumetric measurements from the 3D reconstruction had a lower variance as compared to the clinical measurements. These articles all showcase 3D modeling of a human organ using a volumetric approach, an invasive procedure, of an electro-mechanical approach. That is, an inside-out modeling method for the volumetric or invasive approaches or an extremely time consuming interpolated approach via the electro-mechanical model. Our work details an outward-in methodology that focuses on the specifics of the surface area of the heart rather than the volume. Though, future work can indeed include the volume as an input to the model as our design from Chapter 3 allows for this. Furthermore, for the articles that focus on the human heart, they limit their approach to specific chambers of the heart, or rely on a complex mathematical and time consuming method to interpolate surface areas, while our approach encompasses the entirety of the heart.

4.2 Workflow Specifications

We now provide the workflow for creating the near 3D model of the heart. First, we note that this process requires a free software called Visual Structure From Motion (VSFM) [142] as well as MATLAB R2018a or later with the *Curve Fitting* and *Video and Image Blockset* toolboxes. The VSFM software is used to create the initial point cloud and MATLAB is used for the extraction and 3D modeling. That is, given either the 4x2 or 4x3 orientations described in Sections 3.3 and 3.4, respectively, we give the images as input to the VSFM software. Then, the VSFM software matches the images, creates a sparse reconstruction, and finishes with a dense reconstruction under the assumption that there are enough correspondence points available between images. Once completed, an output file is saved which is then opened in MATLAB. Here we take three major steps to polish the point cloud and produce a final near 3D model. Specifically, automated outlier detection and removal, interpolation of points to increase the density of the point cloud, and empirical Region Of Interest (ROI) creation when necessary. First, for the automated outlier detection and removal, we execute the following steps.

- Calculate the centroid of the point cloud assuming a Gaussian distribution. Therefore, we calculate the mean of the x, y, and z, coordinates, respectively.
- Calculate the distance from the centroid to every point in the cloud.
- Calculate the standard deviation of the centroid distances to use as part of the outlier detection threshold.
- Remove all points in the cloud that have distances greater than the standard deviation plus an empirically determined constant.

Now, the point cloud has been reduced to only contain points that are corresponding to the object in focus. Although, it is possible the point cloud is still sparse and could benefit from additional interpolated points that are constrained to the underlying shape of the point

cloud. Thus, we move onto primary step two which is interpolating points that follow the steps below.

- Create a distribution of values between 0 and 0.05 (empirically determined), inclusive, in intervals of 0.01.
- For every point in the cloud, generate 4 new points around each point in the $+x$, $-x$, $+y$, and $-y$ directions, respectively, of a random value from the previously created distribution. More specifically, create each point using the coordinates of the original point but with the addition of $+x$, $-x$, $+y$, or $-y$ in the respective direction where x or y is the randomly chosen value from the distribution. We disregard the z coordinate as to not corrupt the underlying shape of the point cloud.
- Update the corresponding Red Green and Blue (RGB) value matrix where every interpolated point has the same RGB values as the original point for which the interpolation was performed.

After the interpolation procedure, we increase the density of the point cloud by a factor of 4. This will help with the curve fitting and surface area generation. Unfortunately, for any outlier that escaped the removal process, this exacerbates the outlier problem in that specific area. Although, it would be a problem with or without this interpolation process regardless.

Finally, we complete the third primary step which is empirical ROI detection and further outlier removal. This step helps reduce any outliers that were not caught in the first step and gives us the final point cloud for curve fitting and surface area generation. Also, recall that this step may not be necessary in all cases which is why we still include the first outlier removal procedure. Fortunately, this step only requires a creation of an array of size 6 formated as $xMin$, $xMax$, $yMin$, $yMax$, $zMin$, and $zMax$ which represent ranges within the ROI. Then, we simply remove any points that are not within this ROI. Thus, after completing these three steps, we are left with the final point cloud for creating the

near 3D model. The last step is to simply utilize the MATLAB toolboxes for curve fitting to output the near 3D model.

4.3 3D Modeling Example

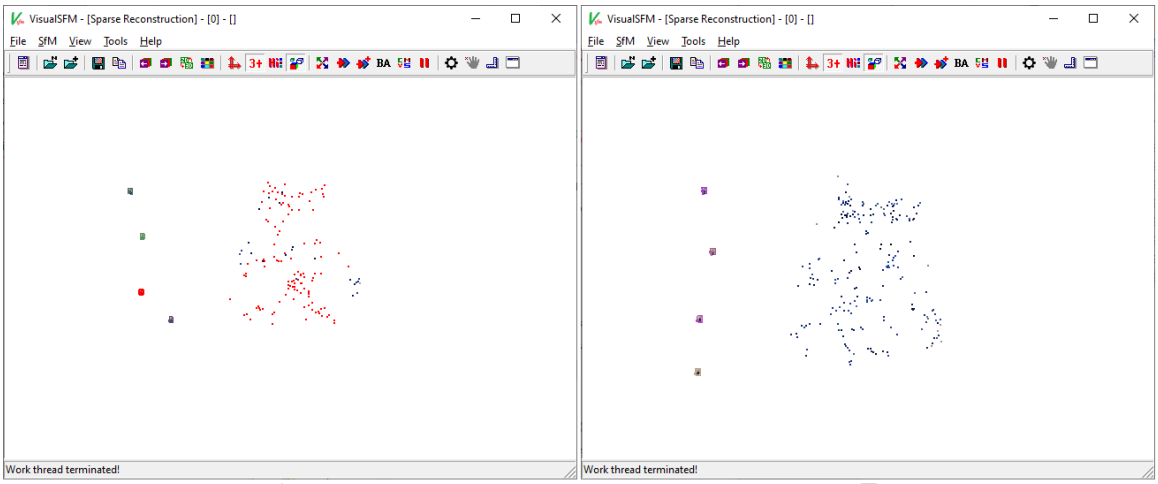
In this section, we provide two examples. The first example is of a near ideal situation utilizing both the 4x2 and 4x3 orientations where we have images of a 3D printed human heart to scale. Unfortunately, we were unable to gather real ultrasound data of the orientations described previously. Thus, 3D printing a heart was adequate for showing this proposed near 3D modeling workflow. This first example is not successful as not enough correspondence points are found. The reasoning will be discussed later. The second example is of a less ideal situation which includes a background of items for better correspondence point matching but fully represents the entirety of the workflow. For both examples, the camera on a Motorola Moto Z^2.

Beginning with the first example, we conduct an experiment by capturing images which mimic the parasternal, frontal, and apical views of our 3D printed heart. These images can be seen in Figure 4.1 which shows the three rows of four images corresponding to each of the views. Specifically, rows one, two, and three correspond to the parasternal, frontal, and apical views, respectively, for the 4x3 orientation. For the 4x2 orientation, we simply remove row two from the set of images for the experiments. With the images captured, we begin with inputting them into the VSFM software for both the 4x2 and 4x3 orientations separately. From the reconstructions, the VSFM software creates a 4x2 and 4x3 point cloud as shown in Figure 4.2 A and 4.2 B, respectively. As we can see, there is no underlying visible structure of the 3D printed heart to extract. Instead, we see effectively a flat surface, which is incorrect. Therefore, this particular example is considered a failure. We hypothesize that this cause is due to the software not being able to make matches between images. That is, the 3D printed heart is homogeneous in color and also has no

imperfections which removes the unique details that we would expect to see from a real heart.

Figure 4.1: The images used for example one. Rows one, two, and three correspond to the parasternal, frontal, and apical views, respectively.

We now move on to the successful example two. First, we present the images used for the modeling process in Figure 4.3. Columns one, two, and three in the figure correspond to the parasternal, frontal, apical views, respectively. For the 4x2 orientation experiment, we simply remove column two from the images. After inputting the images into the VSFM software, we received the point clouds for the 4x2 and 4x3 orientations as shown in Figures 4.4 A and 4.4 B, respectively. Now, we can see a quite obvious underlying structure to the point cloud. Therefore, we continue with this set of images for the remainder of the workflow. Moving forward, all remaining images will have the 4x2 orientation represented as



A **B**

Figure 4.2: The first example results of the VSFM software point cloud output for the 4x2 (A) and 4x3 (B) orientations.

Figure A and the 4x3 orientation represented as Figure B. Once the point cloud is saved, it is first opened in MATLAB directly as shown in Figure 4.5. Following, the automated outlier detection and removal procedure is executed which removes the background noise from the point cloud (Figure 4.6). Then, we add more points to the point cloud to make it more dense using the interpolation method. The result is shown in Figure 4.7. Next, we must identify the heart from the keyboard in the images. Thus, the empirical ROI method is used for this example. That is, we create the following ROI of $[21.4, 23.4, 14.75, 15.8, 14.75, 15.8]$ and $[0.8, 1.37, 1.01, 1.72, 2, 2.5]$ for the 4x2 and 4x3 orientations, respectively. Note that these values are specific to this example and should not be used for other examples. The resulting ROI with respect to the point clouds is shown in Figure 4.8. Continuing, we remove all points outside the ROI resulting in the point clouds in Figure 4.9. Finally, we use the *Curve Fitting* toolbox in MATLAB to create the near 3D model of the heart shown in Figure 4.10. As we can see from the included graphical 3D model of the heart in the figure, the comparison shows that our preliminary results need further improvement. Continuing, we can also calculate the surface area of this model to be 4.06 and 0.81 for the 4x2 and 4x3 orientations, respectively. Note that these values are not necessarily in centimeters or millimeters, rather they are represented by arbitrary units. This is acceptable though as the machine

learning algorithm discussed in Chapter 6 is indifferent to this fact. Finally, we make two important notes. First, as expected from the orientation comparison in Section 3.5, the 4x3 orientation does indeed have a higher resolution from examining these figures. Although, this does not discredit the 4x2 orientation as the underlying structure is still visible and thus usable. Therefore, either orientation is acceptable for future work. Second, this 3D modeling approach is indeed relevant for a moving heart as this workflow will simply be executed per image set to show multiple near 3D models over time in a video format.

Figure 4.3: The images used for example two. Columns one, two, and three correspond to the parasternal, frontal, and apical views, respectively.

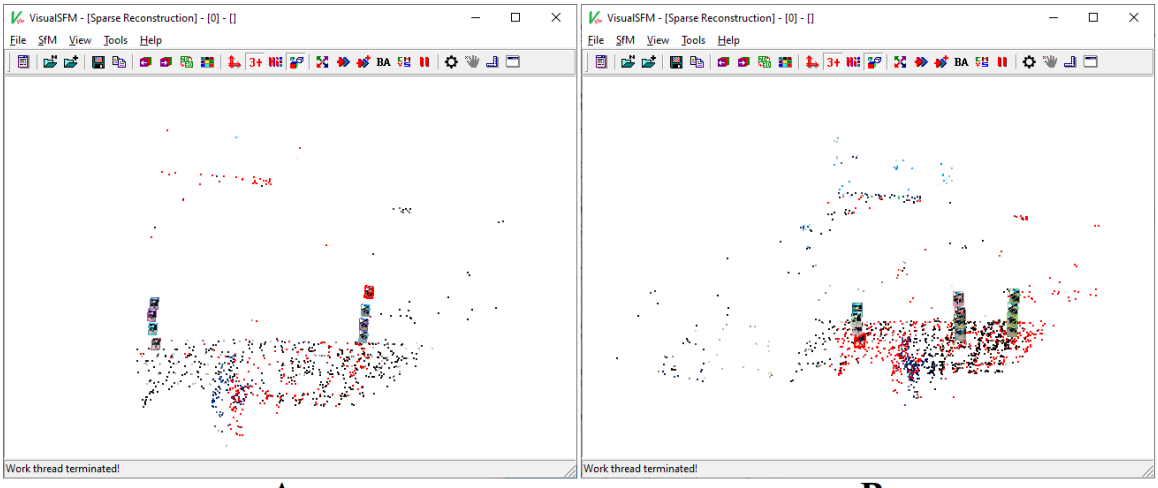


Figure 4.4: The second example results of the VSFM software point cloud output for the 4x2 (A) and 4x3 (B) orientations.

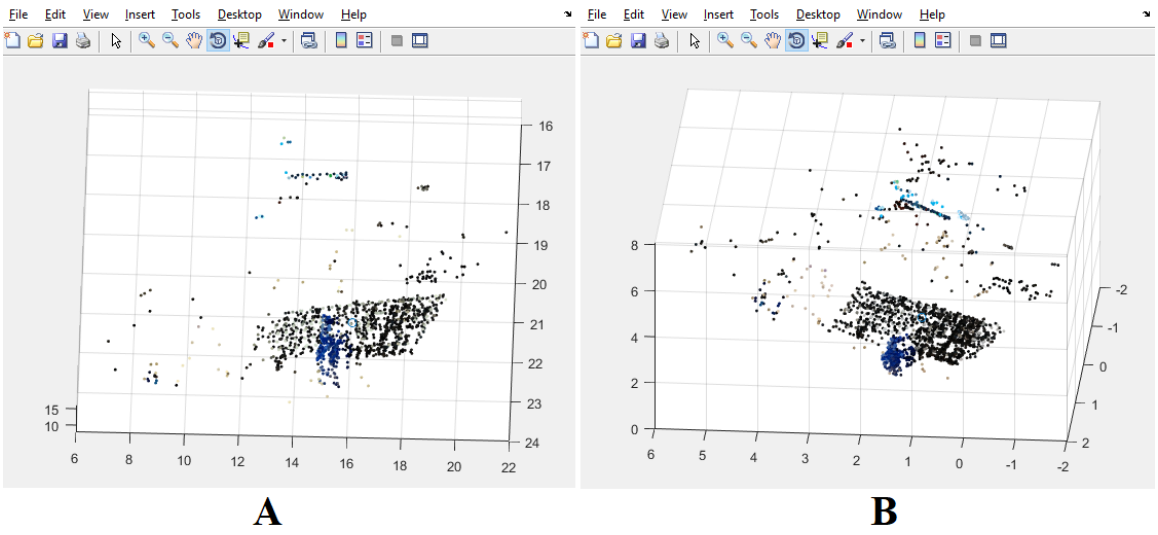


Figure 4.5: The point clouds after being loaded directly into MATLAB. The 4x2 and 4x3 orientations are represented in A and B, respectively.

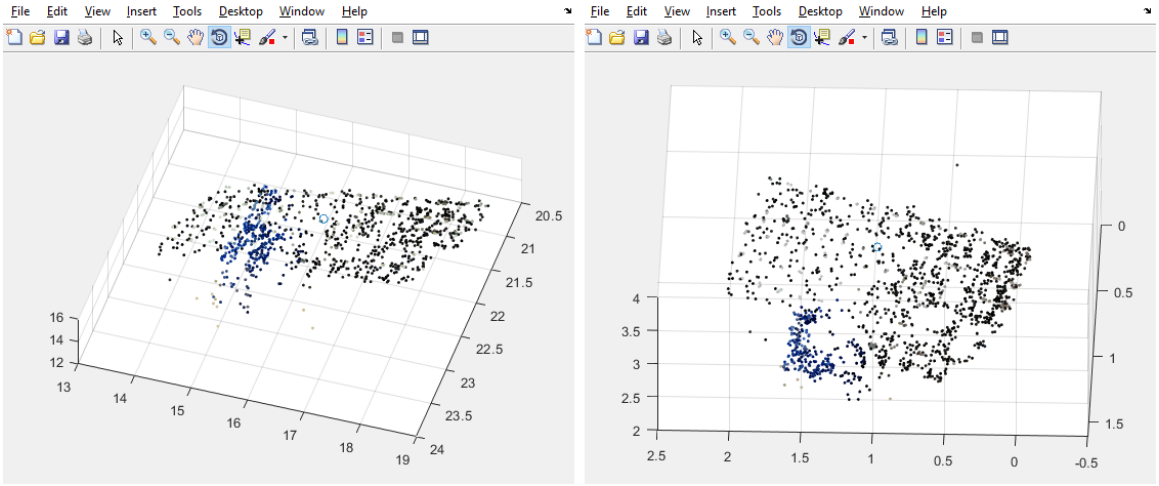


Figure 4.6: The point clouds after the automated outlier detection and removal procedure. The 4x2 and 4x3 orientations are represented in A and B, respectively.

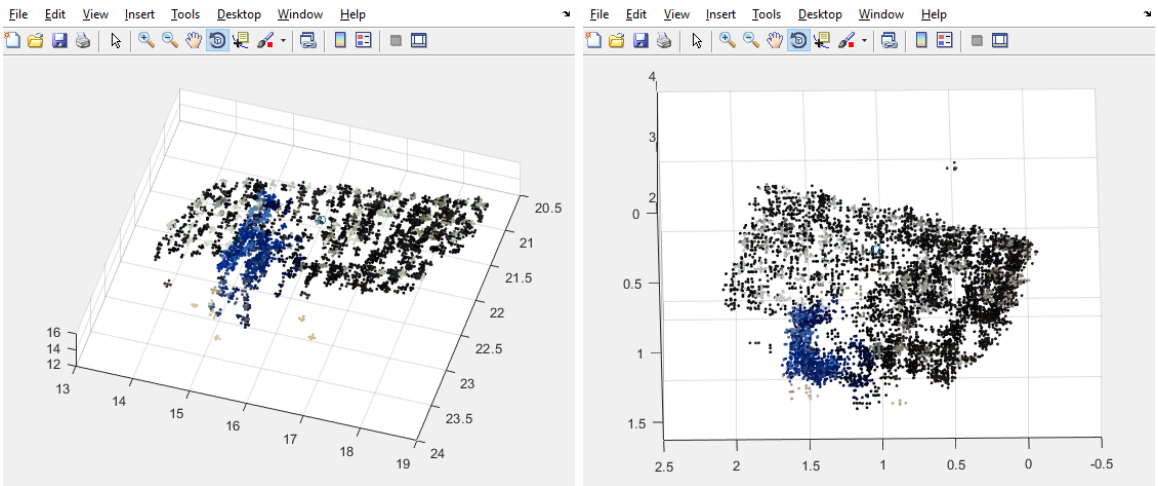


Figure 4.7: The point clouds after the additional point interpolation process. The 4x2 and 4x3 orientations are represented in A and B, respectively.

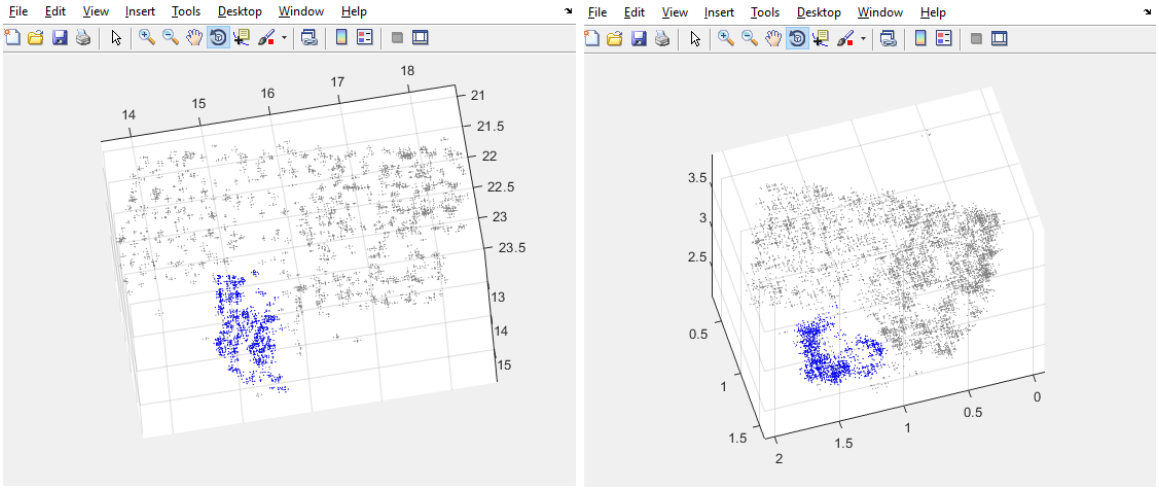


Figure 4.8: The point cloud empirical ROI detection to extract the heart from the background. The 4x2 and 4x3 orientations are represented in A and B, respectively.

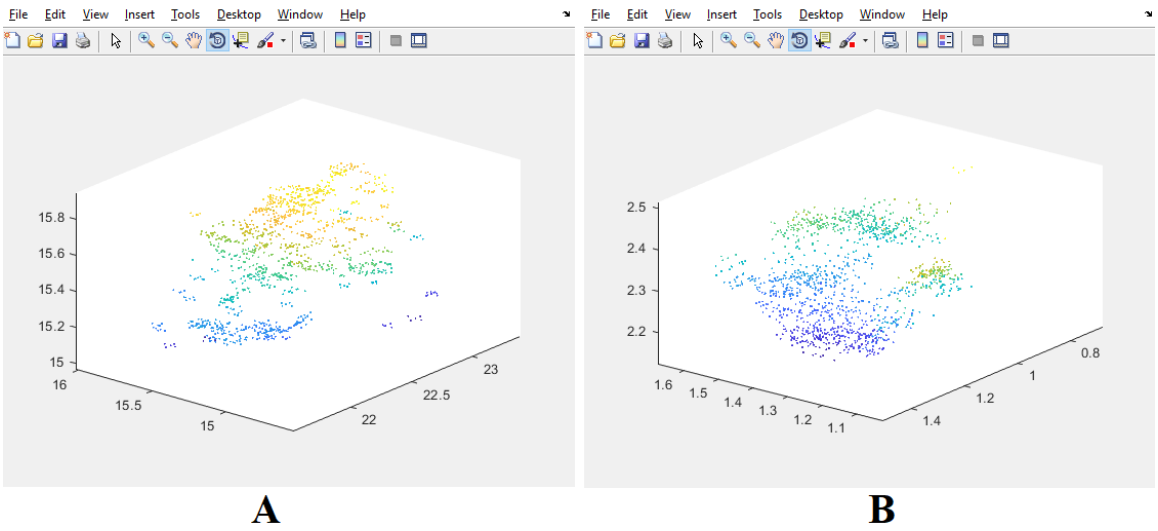


Figure 4.9: The point cloud after removal of points outside of the ROI. The 4x2 and 4x3 orientations are represented in A and B, respectively.

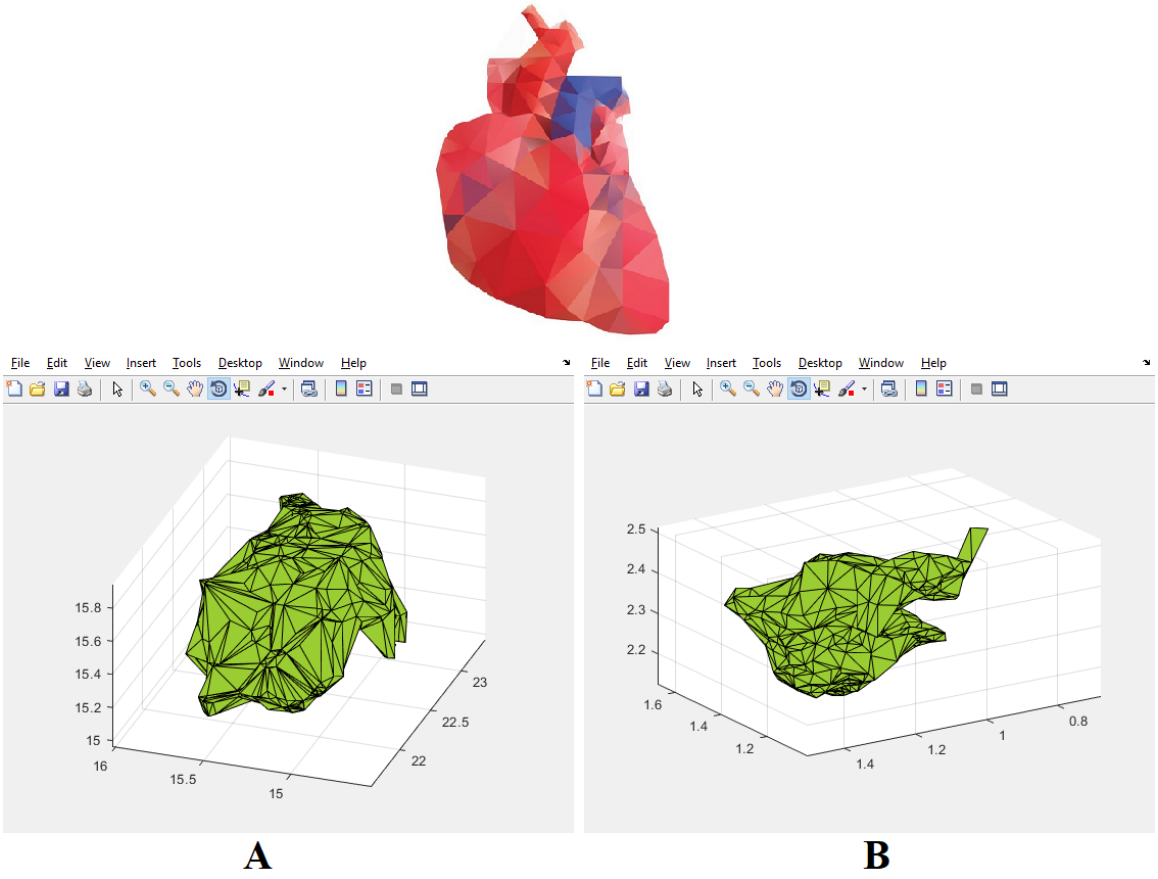


Figure 4.10: The results of the curve fitting and surface area generation of the point cloud to create the near 3D model of the heart. The 4x2 and 4x3 orientations are represented in A and B, respectively. We include a graphical 3D model of the heart to compare our preliminary results against.

5

Modeling of the Heart with Stochastic Petri Nets

For further understanding and functional modeling potential, we move to create a SPN of the overview of a healthy heart's cyclic functionality (with respect to two different signal models) as well as a focused model of systolic and diastolic functionality. Then, we also create a model focusing on a specific CVD. We introduce these SPN models as a possible path to extracting functional features of the heart with respect to the wearable vest, which can be further expanded in the future. First, for the modeling system the SPN is built from, a Petri Net (PN), created by Carl Petri in 1962 [108], is a diagrammatic tool that models the concurrency and synchronization of a system [117]. In our case, this is the healthy human heart. The PN is relatively simple mathematically but excels at providing a human readable model of a system. That is, in this bipartite network, it consists of a set of Places, Tokens, Transitions, and Arcs. A Place or a set of Places represents a state, condition, or resource the system is currently in or needs. Next, Tokens are seen within Places and represent that the corresponding condition is met or the resource is available. Following, Transitions represent an action that occurs to move from one Place to another. Finally, Arcs

simply show the direction of the network and connect Places to Transitions or Transitions to Places, but never Places to Places or Transitions to Transitions. Mathematically, these four sets can be represented as follows [117].

- $PN = \{P, T, I, O, M\}$
- $P = \{p_1, p_2, ..., p_n\}$ is a finite set of Places where $n \geq 0$.
- $T = \{t_1, t_2, ..., t_m\}$ is a finite set of Transitions where $m \geq 0$ with P and T being disjoint (i.e., no elements in common).
- $I : P^{\infty} \rightarrow T$ is the input function which is the mapping from a bag of Places to Transitions.
- $O : T \rightarrow P^{\infty}$ is the output function which is the mapping from Transitions to a bag of Places.
- $M_0 : P \rightarrow n$ is the initial marking of the PN.

Graphically, these four sets are shown in Figure 5.1 as a circle for the Place, dot for the Token, line for the transition, and arrow for the Arc. There are also multiple different types of PN. These include SPN [11], Colored Petri Nets [107], Timed Petri Nets [137], and more. Though, while most are usable, we will choose to continue with the SPN as it satisfies the Markov property, which means the conditional probability distribution of any future state in the SPN only depends on the current state [90]. Furthermore, it allows for variability within the model which can extend to interesting new models. That is, the SPN adds a Λ variable which is a set of firing rates [11]. Effectively, Λ contains multiple random variables associated with every transition in the network which can be written as $\lambda(M)$ where M is the current marking or as $\lambda(T)$ for every transition. Therefore, the final SPN after including Λ is $SPN = \{P, T, I, O, M, \Lambda\}$. By using a SPN to model a specific functionality of the heart, this being either healthy or in certain disease states, it will provide us a hierarchical model with structural components that have probabilistic properties. The

following models are for both a healthy heart and a heart with a CVD. That is, we create three models of the healthy heart and one model for a CVD. Specifically, we will extend the first healthy heart model to show Atrial Fibrillation (AFb). Note that for each CVD, a new model should ideally be made to represent it rather than creating one overarching model for all CVD.

For the first of the three healthy heart models, we showcase the overview of the full healthy heart cycle with the bottom half of Figure 5.2. Here, we have five places, five transitions, and an M_0 of $(1, 0, 0, 0, 0)$. This SPN shows the cycle of blood movement through the atria and ventricles as well as the corresponding necessary contractions and relaxations. Furthermore, this model directly corresponds to one traditional QRS signal cycle of an Electrocardiogram (ECG) reading as seen in the top half of Figure 5.2 [147]. The mathematical formulation of the SPN in Figure 5.2 is as follows.

- $P = \{p_0, p_1, p_2, p_3, p_4\}$
- $T = \{t_0, t_1, t_2, t_3, t_4\}$
- I :
  - $I(t_0) = \{p_0\}$
  - $I(t_1) = \{p_1\}$
  - $I(t_2) = \{p_2\}$
  - $I(t_3) = \{p_3\}$
  - $I(t_4) = \{p_4\}$
- O :
  - $O(t_0) = \{p_1\}$
  - $O(t_1) = \{p_2\}$

- $O(t_2) = \{p_3\}$
- $O(t_3) = \{p_4\}$
- $O(t_4) = \{p_0\}$

- $M_0 : (1, 0, 0, 0, 0)$
- $\Lambda : \lambda(t_{0-4} = 1.0)$

As we are dealing with a healthy heart, we choose to utilize a value of 1.0 for each of the λ firing rates. For the places $p_0, p_1, p_2, p_3,$ and p_4, they are the isovolumetric contraction, ventricular ejection, isovolumetric relaxation, diastasis, and atrial contraction as well as ejection, respectively. For the transitions $t_0, t_1, t_2, t_3,$ and t_4, they are the first phase of ventricular systole, second phase of ventricular systole, early ventricular diastole, late ventricular diastole, and atrial systole, respectively. Note that atrial relaxation and its corresponding transition of diastole are not explicitly presented here as they occur concurrently with $p_0, p_1, p_2,$ and p_3 as well as $t_0, t_1, t_2,$ and t_3, respectively. The resulting model explicitly and succinctly models a single cycle of a healthy heart [72]. The direct correspondence to the ECG signal is due to the places matching the functionality of the P wave, QRS Complex, and T wave of the ECG. That is, for the P wave, QRS Complex, and T wave, they relate to the functionality of atria depolarization (p_1 atrial contraction and ejection), ventricular depolarization (p_2, p_3, and p_4 ventricular contraction, ventricular ejection, and atrial relaxation), and ventricular repolarization (p_0 ventricular relaxation), respectively [72].

Future modifications of this model for any CVD can include additional places and transitions representing the individual valves as well as adjusting the λ firing rates to model uncertainties of functionality as caused by a CVD. For example, we can extend this model to show a heart's functionality with AFb. Before presenting the model, AFb is when the two atria beat irregularly (i.e., out of sync with the ventricles) which equates to either going to fast. More specifically, electrical signals do not originate solely from the the Sinoatrial (SA)

node, the hearts natural pacemaker. Instead, they also originate from the pulmonary veins which cause the atria to fibrillate. The extra electrical signals flood the Atrioventricular (AV) node causing the ventricles to beat fast as well, regardless if they contain enough blood [101]. This condition can lead to blood clots forming due to the loss of regular mechanical atrial contraction and possibly a stroke if the clot breaks off and leads through the Common Carotid Artery into the brain.

As for the AFb model, we extend the first healthy QRS heart model, shown in Figure 5.2, by adding six additional places and eight additional transitions. The initial marking of M_0 is $(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$ for this SPN. These additional places and transitions model the contractions and relaxations of either the left or right atrium as well as the extra ventricular contractions and relaxations. This is graphically shown in Figure 5.3 where the original healthy model is shown in blue with the AFb model extending from it. The mathematical formulation of the AFb SPN is as follows.

- $P = \{p_0, p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}\}$
- $T = \{t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}\}$
- I :
  - $I(t_0) = \{p_0\}$
  - $I(t_1) = \{p_1\}$
  - $I(t_2) = \{p_2\}$
  - $I(t_3) = \{p_3\}$
  - $I(t_4) = \{p_4\}$
  - $I(t_5) = \{p_1\}$
  - $I(t_6) = \{p_5\}$

- $I(t_7) = \{p_6\}$
- $I(t_8) = \{p_7\}$
- $I(t_9) = \{p_4\}$
- $I(t_{10}) = \{p_8\}$
- $I(t_{11}) = \{p_9\}$
- $I(t_{12}) = \{p_{10}\}$

- O :
  - $O(t_0) = \{p_1\}$
  - $O(t_1) = \{p_2\}$
  - $O(t_2) = \{p_3\}$
  - $O(t_3) = \{p_4\}$
  - $O(t_4) = \{p_0\}$
  - $O(t_5) = \{p_5\}$
  - $O(t_6) = \{p_6\}$
  - $O(t_7) = \{p_1\}$
  - $O(t_8) = \{p_3\}$
  - $O(t_9) = \{p_8\}$
  - $O(t_{10}) = \{p_9\}$
  - $O(t_{11}) = \{p_4\}$
  - $O(t_{12}) = \{p_1\}$

- $M_0 : (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$

- $\Lambda : \lambda(t_{0,1,2,3,4,6,7,8,10,11,12} = 1.0), \lambda(t_{5,9} = 0.8)$

Because we extend the first healthy QRS heart model, places p_0, p_1, p_2, p_3, and p_4 as well as transitions t_0, t_1, t_2, t_3, and t_4 remain the same. The additional places p_5, p_6, p_7, p_8, p_9, and p_{10} are additional atrial fibrillated relaxation, additional atrial fibrillated contraction, additional ventricular contraction, additional atrial fibrillated contraction, additional atrial fibrillated relaxation, and additional ventricular relaxation, respectively. Next, the additional transitions $t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}$ and t_{12} are pulmonary vein electrical signal activation, minor fibrillated atrial systole, minor fibrillated atrial diastole, additional ventricular systole first phase, pulmonary vein electrical signal activation, minor fibrillated atrial diastole, minor fibrillated atrial systole, and additional late ventricular diastole, respectively. In this model, we include two additional atrial loops at each of the ventricular ejection and atrial contraction and ejection places with starting transitions t_5 and t_9 having a modified firing rate of 0.8 to simulate the fibrillation process. Moreover, these loop starting transitions also lead to additional ventricular ejections as well as diastasis to simulate how the electrical signal from the pulmonary vein can interact with the AV node and cause such interactions and to show the possibility of clot formations.

Another method at examining a healthy heart is via pressure rather than electrical signals as from an ECG. That is, by using the Wigger's cycle [94] which shows the pressure of the heart. This SPN and the corresponding Wigger's cycle signal can be seen in Figure 5.4. The initial marking of M_0 is $(1, 0, 0, 0, 0, 0)$ for this SPN. The difference between this model and the standard QRS complex model is that there is an additional place and transition to fully show the pressure from contractions, blood ejections, relaxations, and blood fillings. Here, atrial relaxation is not directly acknowledged as it directly overlaps places 2-5. The mathematical formulation of this SPN is as follows.

- $P = \{p_0, p_1, p_2, p_3, p_4, p_5\}$
- $T = \{t_0, t_1, t_2, t_3, t_4, t_4\}$

- I :
  - $I(t_0) = \{p_0\}$
  - $I(t_1) = \{p_1\}$
  - $I(t_2) = \{p_2\}$
  - $I(t_3) = \{p_3\}$
  - $I(t_4) = \{p_4\}$
  - $I(t_5) = \{p_5\}$
- O :
  - $O(t_0) = \{p_1\}$
  - $O(t_1) = \{p_2\}$
  - $O(t_2) = \{p_3\}$
  - $O(t_3) = \{p_4\}$
  - $O(t_4) = \{p_5\}$
  - $O(t_5) = \{p_0\}$
- $M_0 : (1, 0, 0, 0, 0, 0)$
- $\Lambda : \lambda(t_{0-5} = 1.0)$

Once again, as this model represents a healthy heart, we utilize a value of 1.0 for each λ firing rate. For the places p_0, p_1, p_2, p_3, p_4 and p_5, they are atrial contraction, isovolumetric ventricular contraction, ventricular ejection, isovolumetric ventricular relaxation, rapid ventricular filling, and diastasis (slow ventricular filling), respectively. For the transitions t_0, t_1, t_2, t_3, t_4 and t_5, the are mitral valve closes, aortic valve opens, aortic valve closes, mitral valve opens, ventricular pressure stabilization, and diastolic suction, respectively. The

resulting model represents another way of viewing a cardiac cycle of the heart, but from the Wigger's pressure view. Depending on the user's desired expected features, they may choose to use either the QRS complex model, the Wigger's pressure model, or both.

For the third healthy heart model, we showcase a more focused model of the systolic and diastolic phases of the individual chambers as displayed in Figure 5.5. Here, we have eight places, two transitions, and an M_0 of $(1, 1, 1, 1, 0, 0, 0, 0)$. This SPN shows systolic and diastolic actions via the two transitions and the contraction and relaxation states of each chamber via the places. The mathematical formulation of the SPN in Figure 5.5 is as follows.

- $P = \{p_0, p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$
- $T = \{t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$
- I :
  - $I(t_0) = \{p_0, p_1, p_2, p_3\}$
  - $I(t_1) = \{p_4, p_5, p_6, p_7\}$
- O :
  - $O(t_0) = \{p_4, p_5, p_6, p_7\}$
  - $O(t_1) = \{p_0, p_1, p_2, p_3\}$
- $M_0 : (1, 1, 1, 1, 0, 0, 0, 0)$
- $\Lambda : \lambda(t_{0,1} = 1.0)$

Similar to the first healthy heart model, we choose a firing rate of 1.0 for each of the transition's λ functions. For the places $p_0, p_1, p_2, p_3, p_4, p_5, p_6$, and p_7, they are the right atrium relaxed, left atrium relaxed, right ventricle contracted, left ventricle contracted, right atrium contracted, left atrium contracted, right ventricle relaxed, and left ventricle

relaxed, respectively. For the transitions t_0 and t_1, they are the systolic and diastolic actions, respectively. Here we provide a model showing the two states each of the four chambers can be in, relaxed or contracted. With a firing rate of 1.0 for each transition, it shows that a healthy heart will reach each of these states as expected. Though, by modifying the firing rate and perhaps including places and transitions for individual valve functionality, we can model multiple types of CVD similarly to the previous model. With these models, they act as an initial approach to creating functional features from the SPN with respect to the wearable vest. Future work can create more or extend these existing models for such purposes.

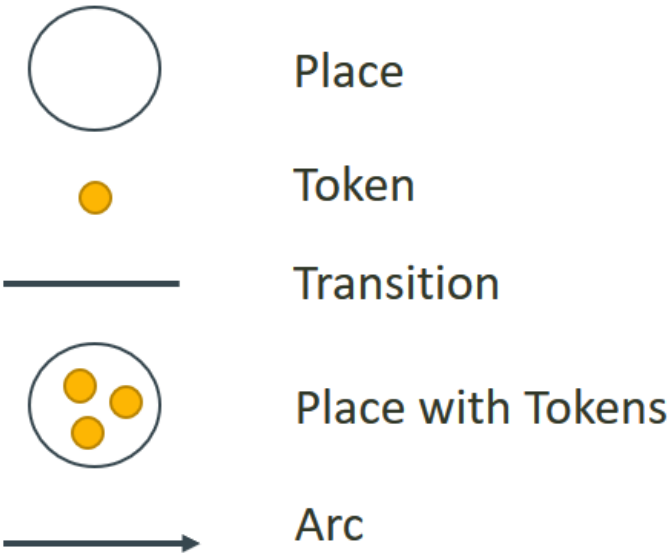


Figure 5.1: The four items that a PN consists of. The Place, Token, Transition, and Arc are represented as a circle, dot, line, and arrow, respectively. Furthermore, a Place with Tokens is shown as a circle with one or more dots inside.

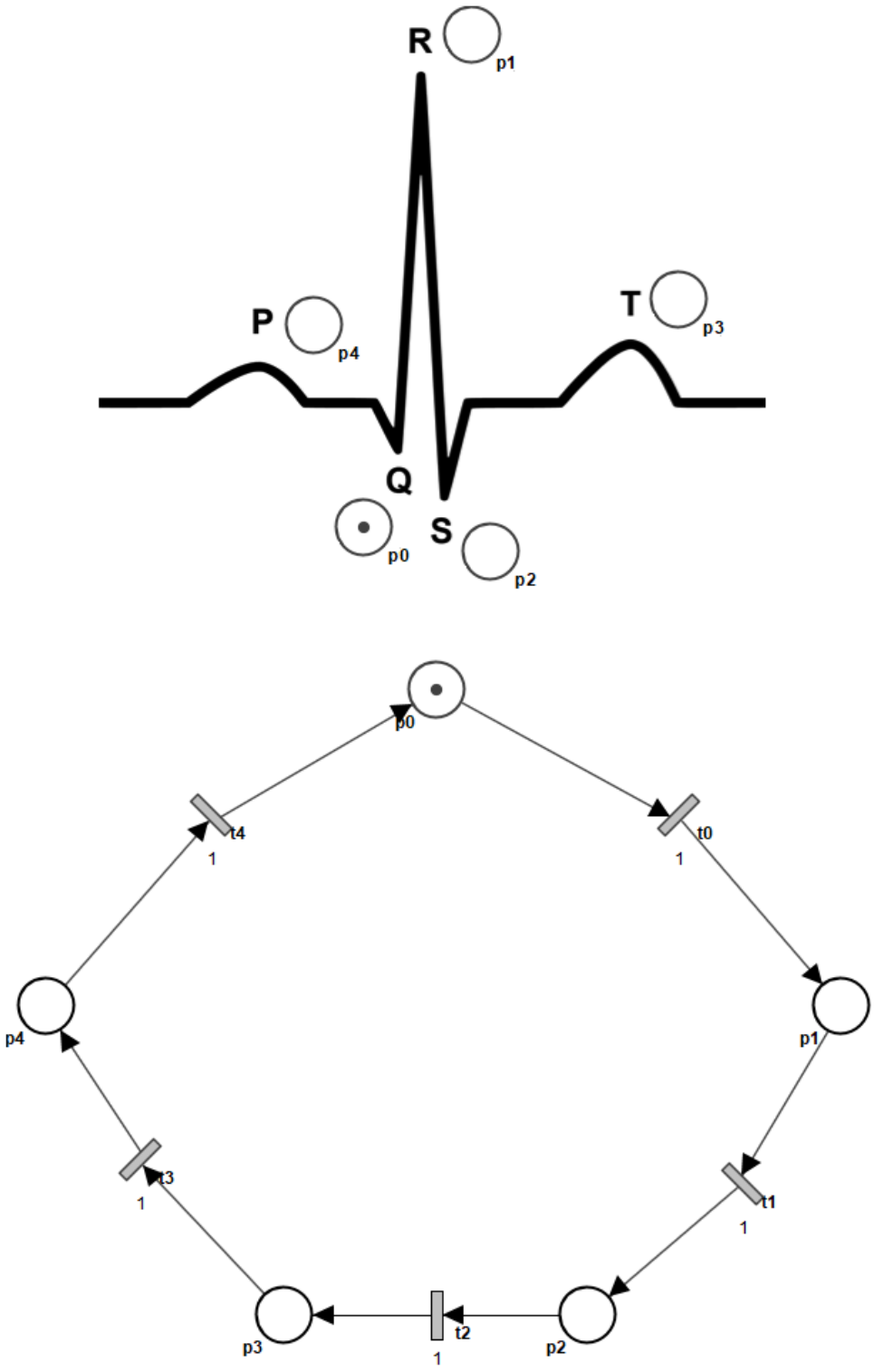


Figure 5.2: The QRS complex SPN overview showing the full systolic and diastolic cycle of a healthy heart as shown in the bottom half of the figure. There are five places and five transitions all of which require one token per transition. The model corresponds to a single signal cycle of an ECG as shown in the top half of the figure.

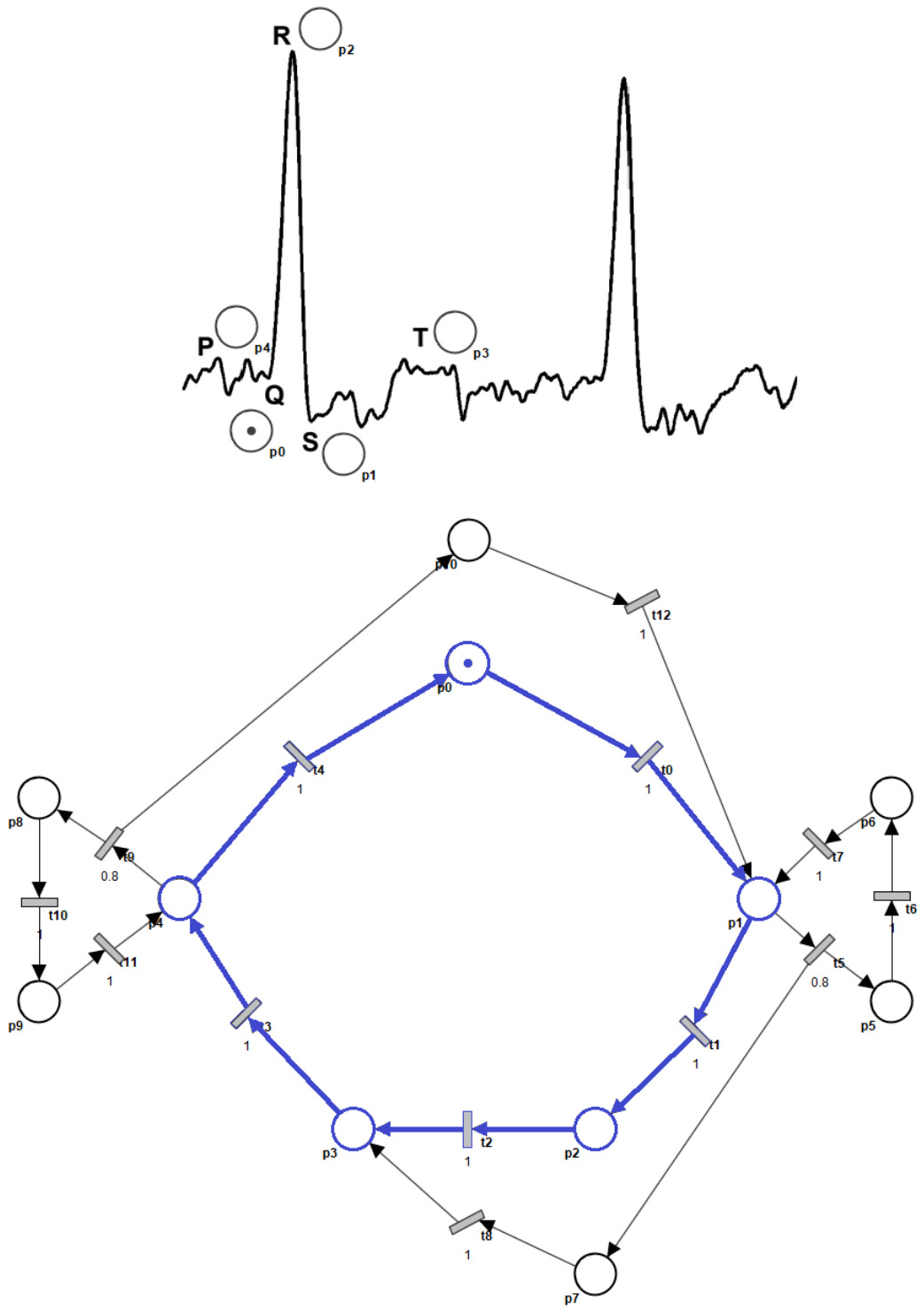


Figure 5.3: The SPN model of AFb which is an extended model of the first healthy heart SPN (colored in blue) as seen in Figure 5.2. There are eleven places and thirteen transitions all of which require one token per transition. Furthermore, transitions 5 and 9 have a firing rate of 0.8 to represent the possibility of activating a fibrillation loop. Finally, each fibrillation loop leads to an additional ventricular contraction or relaxation as well. The AFb signal is courtesy of [41].

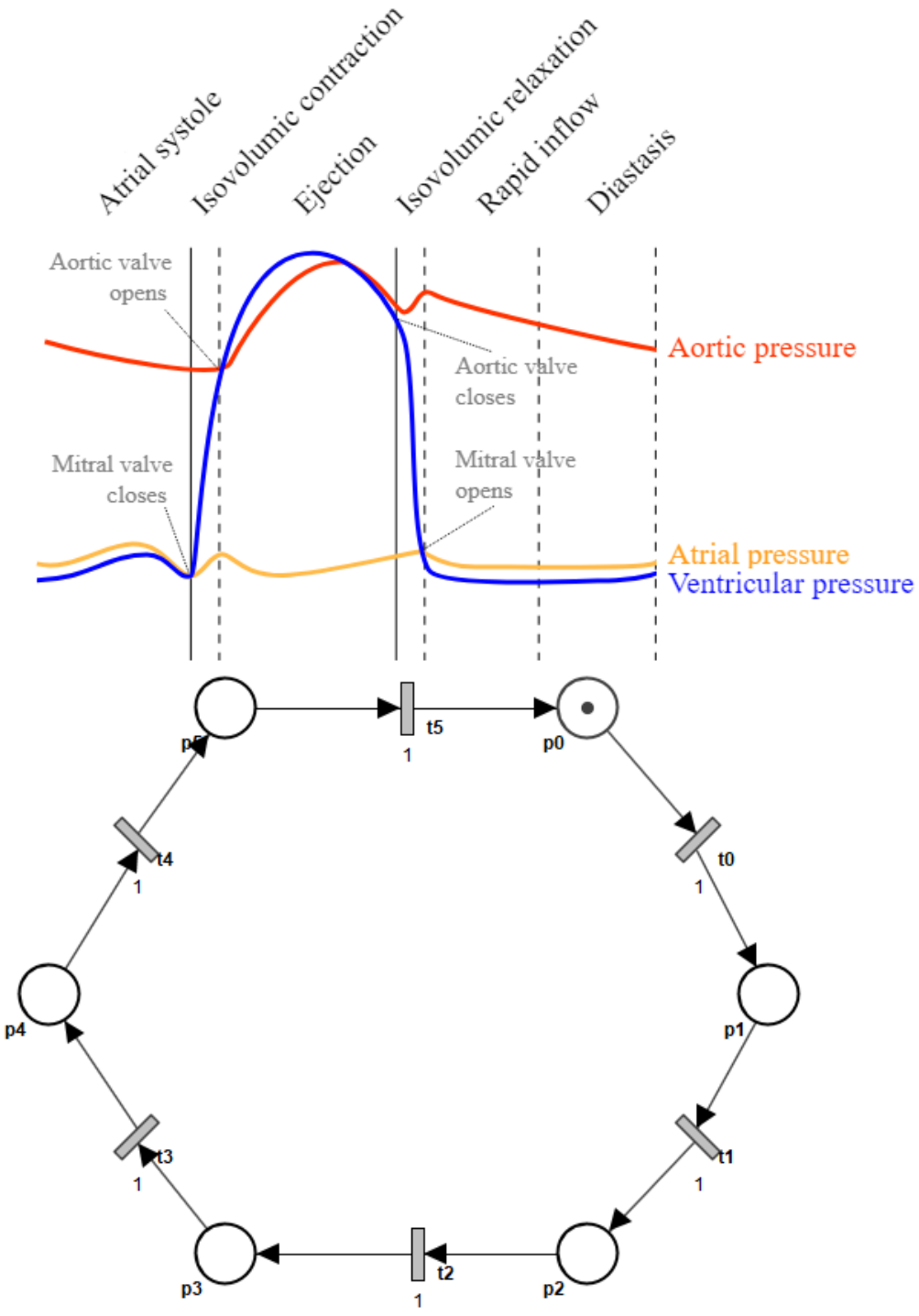


Figure 5.4: The SPN overview using the Wigger's cycle showing the heart pressure as shown in the bottom half of the figure. There are six places and six transitions all of which require one token per transition. The corresponding Wigger's cycle signal is shown in the top half of the figure courtesy of [140].

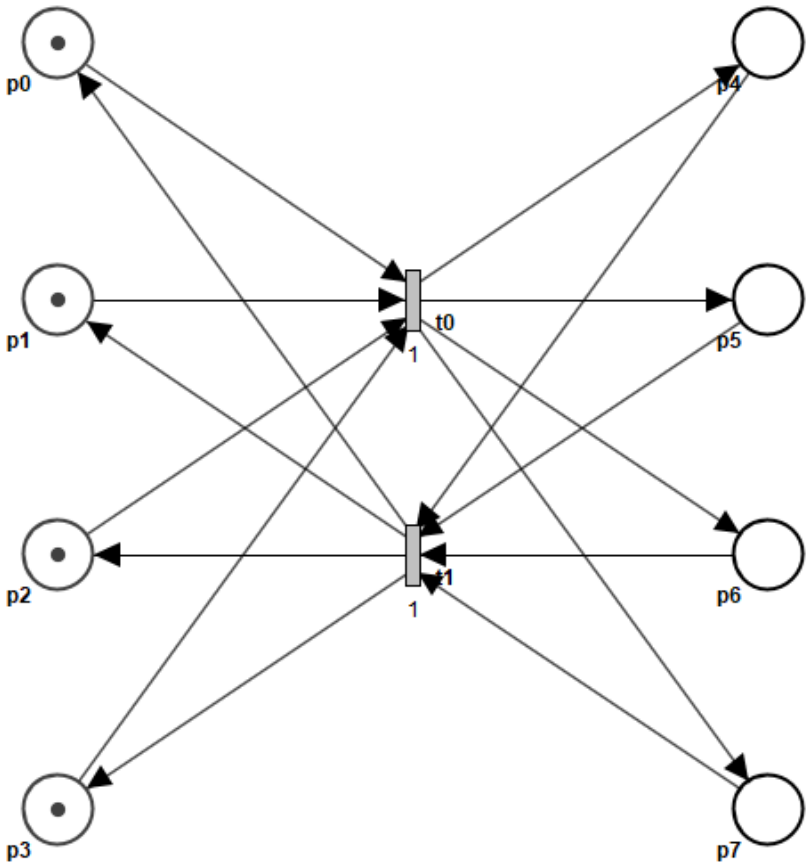


Figure 5.5: The SPN showing the systolic and diastolic functions of a healthy heart. There are eight places and two transitions which each require four tokens.

6

Design of a Novel Machine Learning Algorithm with Real-Time Training Capabilities

In this chapter, we discuss the algorithm we will use for learning from the wearable ultrasound vest and performing the cardiovascular error predictions. This system will use the data that is gathered in real-time from the vest to train a neural network on a single individual's cardiovascular health. We begin by discussing the base training methodology we improve upon in Section 6.1. We provide a representative literature review for this methodology in Section 6.1.1 and a technical description in Section 6.1.2. Following, our state-of-the-art algorithm based from this methodology is presented in Section 6.2. We then show comparative results of our methodology using multiple public datasets in Section 6.2.1. Finally, we provide methods for prediction optimization for our state-of-the-art methodology in Section 6.3 and experimental results of these optimized methods in Section 6.3.1.

6.1 Extreme Learning Machine

We now discuss the precursor to our training methodology that will be used on the neural network included with the wearable ultrasound transducer vest. The methodology must include increasingly fast training times without sacrificing accuracy. This will allow us to continue working with our real-time research perspective. Thus, we do not utilize the traditional backpropagation training methodology and instead look towards a more recent training methodology to work from called the Extreme Learning Machine (ELM). This methodology, proposed by Huang et al. for a Single Layer Feedforward Network (SLFN) in 2005, was intended to be an alternative to backpropagation with highly reduced training times [60]. The training time improvement over traditional backpropagation's gradient descent was significant and the accuracies were competitive as well. The ELM functions by randomizing the input weight matrices and biases, then analytically calculating the output weight matrix by using the Moore-Penrose (MP) [120] generalized inverse of a matrix [60]. Thus, treating the problem of training as a linear system of equations. We first present a representative literature review of ELM on what the current state-of-the-art is in Section 6.1.1. Then, we discuss the technical description of an ELM and how it functions mathematically in Section 6.1.2.

6.1.1 Representative Extreme Learning Machine Literature Review

As previously mentioned, Huang et al. introduced the ELM as an alternative to traditional gradient descent backpropagation [60]. The conventional gradient descent methods iteratively adjust each of the weights and biases causing the overall process to be slow and cumbersome. Though, iteratively generating sets of random values for the weights and biases then calculating the MP inverse is trivial in comparison. The experiments performed by Huang et al. showed that their proposed ELM was 170 times faster than the traditional backpropagation and also provided better generalization.

Zhu et al. proposed an extension to the original ELM in that instead of pure randomization for the input weights and biases of a SLFN, the Differential Evolutionary (DE) algorithm could be utilized [151]. That is, this Evolutionary ELM (E-ELM) shows a higher generalization result over the original ELM though at the cost of slower training times, which was the original main contribution of the ELM. The authors also note that by using the DE algorithm, the E-ELM is supposedly easier to use with nonlinear hidden nodes.

Huang et al. continued their work by introducing an extension of their original ELM to show that the SLFN can be considered a universal approximator in an incremental fashion called the Incremental Extreme Learning Machine (I-ELM) [59]. That is, a user only needs to randomly generate the hidden weights and biases when a new node is added incrementally to the hidden layer, then recalculating the output weights. They provide both a proof that this I-ELM is a universal approximator as well as show a special Two Hidden Layer Feedforward Network (TLFN) case using this system. Furthermore, Huang and Chen continued work on the I-ELM by utilizing a convex optimization methodology on the recalculation of the output weights [57]. This Convex Incremental Extreme Learning Machine (CI-ELM) showed improved generalization over the I-ELM. Lastly, Huang and Chen proposed one final improvement to the I-ELM and CI-ELM by checking the residual error at each learning iteration to decide which node will be added to the final network [58]. That is, the author's hypothesis stated that some nodes in the hidden layer had low impact on the result and increased complexity unnecessarily. Therefore, by examining the residual error at each step, the hidden node with the largest residual error decreasing will be placed in the final network.

Working on the incremental addition of hidden nodes idea, Feng et al. proposed another extension to automatically determine the number of hidden nodes that should be present in a SLFN that do not need to be neural alike [34]. This methodology adds hidden nodes to the SLFN either one at a time or in groups by checking the corresponding

error. This extension, termed the Error Minimized Extreme Learning Machine (EM-ELM) showed results that outperformed previous incremental methods in training time.

Similar to iteratively choosing the SLFN architecture, an article by Miche et al. discussed an Optimally Pruned Extreme Learning Machine (OP-ELM) [93]. This OP-ELM constructed an SLFN using a standard ELM, ranked the hidden nodes using a multiresponse sparse regression algorithm, then used leave-one-out to optimally select the number of hidden nodes to be used in the final SLFN. Their result showed faster training time over the original ELM despite the simplistic approach.

Similar to our state-of-the-art methodology presented in Section 6.2, an article by Zhu et al. discussed their Constrained Extreme Learning Machine (CELM) which constrained the randomization process [152]. That is, the randomization of the weights and biases is done from choosing from a constrained set of difference vectors rather than sampling from a uniform distribution. The results showed higher stability for discriminative tasks while still retaining the fast training speeds of ELM. Furthermore, the results were comparable to those of Support Vector Machines (SVM) and Reduced Support Vector Machines (RSVM).

An article by Weng et al. showed the development of a model for land-use classification from high resolution imagery collected remotely [139]. The authors utilized the CELM in an interesting fashion. That is, they removed the fully connected layers of a Convolutional Neural Network (CNN), normalized the deep convolutional features, then fed this to the CELM. This created a hybrid CNN-CELM model. They used the proposed methodology on high resolution benchmark datasets and showed increased generalization and decreased training time. Though, the current model is incapable of ingesting High Resolution Remote Sensing (HRRS) images.

Tang et al. hypothesized that the ELM may not be suitable for natural signals or with a large amount of hidden nodes and proposed a Hierarchical Extreme Learning Machine (H-ELM) to solve this problem [129]. The primary suspect to these problems comes from the

shallow architecture of a SLFN. Therefore, their main contribution comes from utilizing a Multilayer Feedforward Network (MFN). The hidden weights begin randomly initialized per ELM standard. Then, an unsupervised multilayer encoding is used to extract features to develop an autoencoder which better represents the features over the traditional ELM. Finally, the hierarchical encoded outputs are randomly projected before the decision making process occurs. This provides better generalization and a faster learning speed for the MFN.

The Artificial Neural Network ELM (NELM) and Fuzzy ELM (FELM) were created by Beyhan et al. for modeling U-tube steam generators [16]. Performance of the models were measured by number of neurons in the NELM and number of rules in the FELM, respectively. The performance metrics used were the Root Mean Square Error (RMSE) and Minimum Descriptive Length (MDL) criteria. The authors found that in the offline prediction case, both the NELM and FELM performed similarly. Though, in the online prediction case, the NELM RMSE showed an order of magnitude higher performance than FELM.

Yaseen et al. proposed an Enhanced ELM (EELM) which uses Complete Orthogonal Decomposition (COD) instead of the traditional ELM method of the MP inverse [144]. The benefits from using the COD method comes from increased stability, efficiency, and scalability over using the MP inverse. When comparing to a standard tool in the hydrology field, Support Vector Regression (SVR), the authors found that their EELM had a non-trivially increased RMSE and Mean Absolute Error (MAE).

An article by Zhang et al. presented the development of the Online Sequential Outlier Robust ELM (OSORELM) that they combine with Hybrid Mode Decomposition (HMD) [149]. The authors created this methodology for predicting wind speeds with power grid management using wind power. The input data was preprocessed with HMD to reduce the impact of outliers before training with the OSORELM. Finally, the authors present

an optimized version of the OSORELM using a Criss-Cross Optimization (CSO). They showed that their OSORELM with preprocessed data from HMD and optimized with CSO outperformed the current benchmarks.

Bui et al. utilized an ELM to compute the susceptibility of a flash flood for any given area [21]. In this study, the authors choose to stray from the standard approach of randomizing weights and biases for the ELM and instead choose to implement Particle Swarm Optimization (PSO) to generate these values. The PSO combined ELM iteratively refines the neural network to find the best performance. The authors compared their method to the Multilayer Perceptron-Artificial Neural Network (MLP-ANN), SVM, and C4.5 Decision Tree and showed significantly better results.

Finally, Alom et al. presented a modification of the ELM which examines the generation of weights and biases as they could be potentially part of unstable outputs during both the training and testing phases [5]. Therefore, during the iterative training of the ELM, the authors saved the best performing model's weights and biases for the final network. This State Preserved Extreme Learning Machine (SPELM) showed increased accuracy as the original ELM did not consider that a previous iteration $n-1$ could have better performance over the following iteration n.

6.1.2 ELM Technical Description

As previously discussed, the ELM is an alternative method of training a neural network's weights and biases. The ELM is capable of training a SLFN and a MFN. As a reminder, according to [59], an ELM trained SLFN can be shown as a universal approximator under certain conditions, which makes it an excellent choice to proceed with. When working with the ELM training methodology, it is capable of training any neural network connection architecture. That is, the network need not be fully connected. So, given any neural network architecture, we randomize the hidden weights and biases and treat the network as a system

of linear equations as shown in Equation 6.1.

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T} \tag{6.1}$$

The three variables in Equation 6.1 are as follows. First, $\mathbf{H} = \{h_{ij}\}$ where h_{ij} is the activation function of the hidden node and where i iterates from $1, ..., N$ and j iterates from $1, ..., K$. The variables N and K are the number of data points and number of hidden nodes, respectively. More specifically, the activation function h_{ij} can be further represented as $g(w_j \cdot x_i + b_j)$. The variables within this representation of the activation function are w_j which is the weight vector $[w_{j1}, ..., w_{jn}]^T$ that connects the j^{th} input node and corresponding hidden node. Following, $w_j \cdot x_i$ calculates the inner product of the weight vector w_j and a single data point x_i. Finally, the activation function adds the bias vector b_j. Any activation function chosen is feasible, though generally the literature uses the Sigmoid function. Continuing, $\boldsymbol{\beta}$ denotes the matrix of output weights represented by $[\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_n]^T$. Within this variable, a single $\boldsymbol{\beta}_j$ is represented by the weight vector $[\boldsymbol{\beta}_{j1}, ..., \boldsymbol{\beta}_{jn}]^T$ that connects the j_{th} hidden node and corresponding output node. Finally, $\mathbf{T}$ denotes the matrix of target outputs represented by $[t_1, ..., t_N]^T$. Using this equation, the ELM finds the least-square solution which is then represented by Equation 6.2.

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger}\mathbf{T} \tag{6.2}$$

From this equation, $\hat{\boldsymbol{\beta}}$ is the output weight matrix calculated by $\mathbf{H}^{\dagger}\mathbf{T}$. That is, the MP generalized inverse of the matrix $\mathbf{H}^{\dagger}$ multiplied by the target matrix $\mathbf{T}$. The symbol † denotes the MP generalized inverse calculation on the matrix $\mathbf{H}$. With these two equations, we have a neural network with randomly generated hidden weights and biases with an output layer calculated using the MP generalized inverse. This process is simply one iteration of multiple iterations. It is unlikely with just one iteration that the ELM will find a satisfac-

tory randomization. Thus, the ELM is inherently an iterative process where multiple low complexity iterations must occur.

With a trained neural network, we can finally utilize classical feedforward to produce results in both regression and classification instances. That is, we simply use linear algebra matrix multiplications as seen in Equation 6.3 [97] to feed in data and have the network produce a prediction.

$$\mathbf{x_n} \xrightarrow{\mathbf{W}} \mathbf{a_n} \xrightarrow{\mathbf{g}} \mathbf{z_n} \xrightarrow{\hat{\boldsymbol{\beta}}} \mathbf{b_n} \xrightarrow{\mathbf{h}} \hat{\mathbf{y}}_\mathbf{n} \tag{6.3}$$

This equation shows the flow of data from input to the predicted target output. That is, the single data point $\mathbf{x_n}$ is multiplied by the weight matrix $\mathbf{W}$, which is randomized by the ELM to produce $\mathbf{a_n}$. Then, $\mathbf{a_n}$ is passed through the activation function $\mathbf{g}$ to calculate $\mathbf{z_n}$. Following, $\mathbf{z_n}$ is multiplied by the least-square solution $\hat{\boldsymbol{\beta}}$ which produces $\mathbf{b_n}$. Finally, $\mathbf{b_n}$ is passed through the activation function $\mathbf{h}$ to produce the prediction $\hat{\mathbf{y}}_\mathbf{n}$.

6.2 Constrained State-Preserved Extreme Learning Machine

The ELM and literature surrounding it have been very beneficial to neural network research. Though, besides the article by Zhu et al. [152], the idea of constraining how the ELM functions has not been thoroughly examined. Furthermore, Zhu et al. only constrained the randomization of the weights based on a distance vector approach. We believe that the weights of the hidden node connections per iteration provides an approximation of a local optimum and that we can improve upon this. That is, by perturbing this initial approximation, we hypothesize that it is possible to achieve a measurably better overall model without sacrificing considerable training time. Thus, we present the Constrained State-Preserved

Extreme Learning Machine (CSPELM) [49] that is based off of the constraining premise discussed in [152] and the state preserving aspect presented in [5].

The CSPELM functions by including an ε constraint on the actual randomization process itself for both the weights and biases. The idea is that by using the state-preserving aspect from [5], we can find a local optimum from an initial set of SPELM iterations, then use the preserved weights and biases from this network as a starting point to perturb upon. With this initial starting state, we include the constraint ($\pm\varepsilon$) to the randomization search of the uniform distribution. This can be seen in Algorithm 1.

Algorithm 1: The CSPELM algorithm that takes in as input a trained SPELM, corresponding dataset, and the user specified ε. The weight and bias matrices from the trained SPELM are used as the basis of the constrained randomization search.

input : A trained SPELM, dataset, and constraint ε
output: A trained CSPELM
Initialization:
$Best\_Result$:= int;
for $i \leftarrow 0$ **to** Num_Nodes **do**
  $bias[i]$ = rand($b[i]$ - ε, $b[i]$ + ε);
  for $j \leftarrow 0$ **to** $Num\_Features$ **do**
    $weight[i][j]$ = rand($w[i][j]$ - ε, $w[i][j]$ + ε);
  end
end
H = zeros($Num\_Samples$, Num_Nodes);
for $i \leftarrow 0$ **to** Num_Nodes **do**
  for $j \leftarrow 0$ **to** $Num\_Samples$ **do**
    $Inner\_Product$ = in($weight[i,:]$, $x[j,:]$);
    $H[j,i]$ = g($Inner\_Product$ + $bias[i]$);
  end
end
$H^{\dagger}$ = MP_Inverse(H);
$\hat{\beta}$ = $H^{\dagger}T$;
$Model.Result$ = feedforward($weight$, $bias$, $\hat{\beta}$);
if $Best\_Result$ < $Model.Result$ **then**
  w = $weight$;
  b = $bias$;
  $Best\_Result$ = $Results$;
end

From this algorithm, the CSPELM takes in as input a trained SPELM, the corresponding dataset, and the user specified ε constraint. At the beginning of the algorithm, we can see that the ε variable is added to the randomization process in which it alters the inputs to the "rand" function. Instead of sampling from a uniform distribution, it now samples randomly from the interval of $\pm\varepsilon$ of $bias[i]$ and $weight[i][j]$, respectively. So, we take into account the values of these weights and biases that were present from the local optimum created by the SPELM instead of completely re-randomizing them. There are five functions used in Algorithm 1. First, function "rand" denotes the randomization function that samples a value between the upper and lower bounds presented as parameters one and two, respectively. Next, the function "zeros" simply generates an m x n sized matrix containing only zeros. Following, the function "in" calculates the inner product of two matrices. Then, the function "MP_Inverse" calculates the MP generalized inverse on the specified matrix. Finally, the function "feedforward" takes the $weight$ matrix, $bias$ matrix, and calculated $\hat{\beta}$ matrix to produce a prediction.

Another contribution the CSPELM provides is the homing mechanism which can be seen in Algorithm 2. This homing mechanism iteratively shrinks the upper and lower bounds that the ε provides on the randomization of the weights and biases. The variables ε, ε_Mult, and $CSPELM\_Iter$ are user defined. For example, suppose ε is set to 0.01, ε_Mult is set to 10, and $CSPELM\_Iter$ is set to 5. Therefore, the CSPELM will generate random values for the weights and biases at an interval of 0.10, shrinking by 0.01 per inner loop iteration, 5 times. This allows there to be a number of iterations at each varying constraint until the best performing model is found. This state-of-the-art CSPELM allows for higher performing models as compared to the standard ELM and SPELM with a minor increase in training time. Specifically, the training time is effectively the training time of a standard ELM doubled. Furthermore, we outperform the most popular Levenberg-Marquardt (LM) backpropagation in many instances.

Algorithm 2: By including the ε_Mult variable to the CSPELM, we can vary the constraint sizes for the random search by starting large and homing into a better performing optimum.

```
Initialization:
ε := float;
ε_Mult := int;
CSPELM_Iter := int;
Best_Result := float;
Best_Model := list();
for i ← 0 to CSPELM_Iter do
    for j ← ε_Mult to 1 do
        Model = CSPELM(SPELM, data, ε * j);
        if Best_Result < Model.Result then
            Best_Model = Model;
            Best_Result = Model.Result;
        end
    end
end
```

6.2.1 Public Dataset Results

To show the improvements that the CSPELM provides, we examine the accuracy over a variety of datasets for both classification and regression. These datasets vary in size, number of training and test data, number of features, and number of target classes. Specifically, we examine 13 different datasets in total. The publicly available datasets used in these experiments are as follows. Classification datasets are labeled as "Class" while regression datasets are labeled as "Reg". Also, the datasets are abbreviated to capital letters A-M for simplicity.

A: (Class) Diabetes [123]

B: (Class) Ionosphere [122]

C: (Class) Heart Disease [65]

D: (Class) Wisconsin Breast Cancer [141]

E: (Class) Conditional Grammatical Facial Expressions [37]

F: (Class) Satellite [126]

G: (Class) Image Segmentation [19]

H: (Class) Optical Recognition of Handwritten Digits [6]

I: (Reg) Weather in Szeged [20]

J: (Reg) Concrete Compressive Strength [146]

K: (Reg) Parkinson's Telemonitoring with the total_UPDRS outcome variable [134]

L: (Reg) Combine Cycle Power Plant [135]

M: (Reg) Airfoil Self-Noise [83]

The precise attributes for each dataset can be seen in Table 6.1. The number of classes, features, training data, and test data are shown. The "–" in the #Classes column signifies that the dataset is for regression. We take the standard approach in training and testing the ELM, SPELM, CSPELM, and LM backpropagation algorithms. That is, unless the dataset came provided with a pre-split training and test set, we shuffled and split the data into 80% training and 20% test sets. Then, all data were normalized from [-1, 1]. We choose the sigmoid function of $1/(1 + e^{-x})$ to be the activation for all four algorithms. Next, the number of hidden nodes of the neural network is calculated using the rule of thumb $\lfloor(2/3 * \#InNodes) + \#OutNodes\rfloor$ [29]. The only dataset which does not utilize this calculation is dataset E which simply had to many features and caused the computer used for the experiments to run out of memory. Therefore, we choose to use 40 hidden nodes for this particular network. The regression neural networks used the RMSE for determining the performance. All 13 datasets used a fully connected neural network. Finally, for the user defined variables, $CSPELM\_Iter$, ε_mult, and ε were set to 5, 10, and 0.01, respectively. All four algorithms were trained 50 times each to calculate the mean and standard deviation performances.

The hardware utilized for these experiments consisted of an Intel Core I7-4790K CPU at 4.00 GHz, an AMD Radeon R9 380 Series GPU, and 16.00 GB of DDR3 RAM. The ELM, SPELM, and CSPELM were coded in Python 3.7.1 using the random seed of 101 for reproducibility. The LM backpropagation was coded in MATLB R2018a using the pattern recognition neural network toolbox.

Dataset	#Classes	#Features	#Training Data	#Test Data
A	2	8	613	155
B	2	34	279	72
C	2	13	240	63
D	2	30	445	124
E	2	301	1524	383
F	6	36	3542	893
G	7	19	2100	210
H	10	64	3823	1797
I	–	6	77162	19291
J	–	8	825	205
K	–	16	4700	1175
L	–	4	7654	1914
M	–	5	1202	301

Table 6.1: The datasets used for the experiments of the ELM, SPELM, CSPELM, and LM backpropagation algorithms. The "–" signifies a regression dataset.

We separate the results of the classification and regression datasets. First, Table 6.2 shows the classification results. This table shows the mean $\pm$ standard deviation accuracy percentages, the final/best accuracy, mean $\pm$ standard deviation training time for the 50 training iterations, the total time of all 50 training iterations, and the number of hidden nodes in the neural network. Similarly, for the regression datasets, Table 6.3 shows the same information but uses RMSE instead of accuracy percentages. Both tables have the column for final/best performance of the algorithm. The "final" is presented as the original ELM has no state preserving features. Thus, the last iteration of training is what would be used in the final neural network. In both tables, the bold values represent the best result for the corresponding dataset. For the classification datasets, our CSPELM algorithm

outperformed the other three algorithms in 5 of the 8 datasets. Furthermore, our CSPELM algorithm had the best accuracy over both the ELM and SPELM in all 8 instances. Similarly, for the regression datasets, our CSPELM algorithm outperformed the other three algorithms in 3 of the 5 datasets. Moreover, our CSPELM algorithm had the best RMSE over both the ELM and SPELM in all 5 instances.

Dataset	Algorithm	Accuracy %	Final/Best Accuracy %	Training Time (s)	Total Time (s)	Hidden Nodes
A	ELM	71.07 ± 3.15	70.32	0.02 ± 2.8x10(-3)	1.19	7
	SPELM	71.07 ± 3.15	78.96	0.02 ± 0.01	1.22	
	CSPELM	**74.39 ± 2.22**	**79.94**	0.03 ± 0.01	2.51	
	LM	71.06 ± 2.22	78.07	0.24 ± 0.29	12.2	
B	ELM	85.28 ± 4.05	81.94	0.03 ± 3.1x10(-3)	1.54	24
	SPELM	85.28 ± 4.05	91.39	0.03 ± 0.01	1.57	
	CSPELM	**87.64 ± 1.12**	93.19	0.03 ± 0.01	3.2	
	LM	84.78 ± 5.16	**94.44**	0.59 ± 0.31	29.49	
C	ELM	73.05 ± 4.58	73.02	0.01 ± 9.4x10(-4)	1.43	10
	SPELM	73.05 ± 4.58	87.50	0.01 ± 2.1x10(-3)	0.59	
	CSPELM	**79.08 ± 1.48**	**88.33**	0.01 ± 2.8x10(-3)	1.25	
	LM	74.32 ± 4.52	82.54	0.24 ± 0.29	12.32	
D	ELM	94.29 ± 2.16	94.36	0.05 ± 0.01	2.34	22
	SPELM	94.29 ± 2.16	97.53	0.05 ± 0.01	2.42	
	CSPELM	91.97 ± 1.56	98.65	0.05 ± 0.01	4.7	
	LM	**95.84 ± 7.10**	**99.19**	0.76 ± 0.43	37.81	
E	ELM	82.83 ± 9.20	89.03	0.27 ± 0.01	13.87	40
	SPELM	82.83 ± 9.20	96.65	0.27 ± 0.02	13.56	
	CSPELM	86.96 ± 1.17	**97.38**	0.29 ± 0.02	28.37	
	LM	**92.94 ± 7.75**	97.13	1141.23 ± 606.48	57061.44	
F	ELM	88.28 ± 2.06	89.36	0.47 ± 0.03	23.57	30
	SPELM	88.28 ± 2.06	91.59	0.45 ± 0.02	22.39	
	CSPELM	**89.91 ± 0.28**	**92.52**	0.45 ± 0.01	44.77	
	LM	82.82 ± 0.74	84.55	18.02 ± 2.84	900.98	
G	ELM	92.84 ± 1.05	92.38	0.19 ± 0.01	9.24	19
	SPELM	92.84 ± 1.05	93.38	0.18 ± 0.01	9.11	
	CSPELM	93.69 ± 0.87	94.09	0.19 ± 0.01	18.37	
	LM	**94.84 ± 3.65**	**98.09**	4.15 ± 1.53	207.64	
H	ELM	99.49 ± 0.23	99.33	0.83 ± 0.02	41.28	52
	SPELM	99.49 ± 0.23	99.37	0.79 ± 0.01	39.83	
	CSPELM	**99.68 ± 0.05**	**99.43**	0.82 ± 0.02	80.74	
	LM	95.64 ± 0.79	97.16	256.56 ± 27.91	12828.08	

Table 6.2: Experimental results of the ELM, SPELM, CSPELM, and LM backpropagation algorithms for the classification datasets. The bold values show the highest performing algorithm in both the average accuracy and overall best accuracy cases. Our CSPELM algorithm outperformed the other three algorithms in 5 out of 8 datasets. Furthermore, our CSPELM algorithm had the best accuracy over both the ELM and SPELM algorithms in all 8 instances.

Dataset	Algorithm	RMSE	Final/Best RMSE	Training Time (s)	Total Time (s)	Hidden Nodes
I	ELM	0.16 ± 0.01	0.15	1.98 ± 0.03	99.11	5
	SPELM	0.16 ± 0.01	0.14	1.95 ± 0.06	97.58	
	CSPELM	0.18 ± 0.01	0.14	1.95 ± 0.06	195.10	
	LM	**0.14 ± 0.02**	**0.13**	4.62 ± 3.63	230.74	
J	ELM	18.04 ± 2.87	16.39	0.03 ± 3.8x10(-3)	1.35	6
	SPELM	18.04 ± 2.87	9.38	0.03 ± 0.01	1.29	
	CSPELM	**13.41 ± 0.38**	**9.21**	0.03 ± 0.01	2.81	
	LM	17.95 ± 1.99	13.05	0.17 ± 0.23	8.57	
K	ELM	11.82 ± 0.32	12.34	0.25 ± 0.01	12.29	11
	SPELM	11.82 ± 0.32	10.09	0.25 ± 0.02	12.48	
	CSPELM	12.77 ± 0.21	9.97	0.25 ± 0.02	24.74	
	LM	**9.98 ± 0.38**	**9.79**	0.35 ± 0.31	17.34	
L	ELM	16.89 ± 10.98	28.14	0.15 ± 0.01	7.28	3
	SPELM	16.89 ± 10.98	4.64	0.14 ± 0.01	7.02	
	CSPELM	**5.01 ± 0.23**	**4.39**	0.15 ± 0.02	11.41	
	LM	15.35 ± 1.66	14.09	0.25 ± 0.26	12.38	
M	ELM	8.58 ± 2.51	9.69	0.03 ± 4.1x10(-3)	1.43	4
	SPELM	8.58 ± 2.51	5.47	0.01 ± 2.6x10(-3)	1.32	
	CSPELM	7.32 ± 0.55	**5.18**	0.03 ± 4.1x10(-3)	2.70	
	LM	**6.79 ± 1.08**	5.60	0.21 ± 0.23	10.71	

Table 6.3: Experimental results of the ELM, SPELM, CSPELM, and LM backpropagation algorithms for the regression datasets. The bold values show the highest performing algorithm in both the average RMSE and overall best RMSE cases. Our CSPELM algorithm outperformed the other three algorithms in 3 of the 5 datasets. Furthermore, our CSPELM algorithm had the best RMSE over both the ELM and SPELM algorithms in all 5 instances.

6.3 Prediction Optimization Methods for the Constrained State-Preserved Extreme Learning Machine

As seen in Section 6.2, we created the CSPELM, a state-of-the-art SLFN training algorithm. With default hyper-parameter settings for the user defined variables $CSPELM\_Iter$, ε_mult, and ε being set to 5, 10, and 0.01, respectively, we showed impressive results in Section 6.2.1 when compared to the original ELM, SPELM, and LM backpropagation. Now, we introduce prediction optimization methods for the CSPELM in the form of a forest optimization technique, ε constrain range-finder, and dataset specific CSPELM hyper-parameter settings for the best possible prediction results [48]. Note that in these scenarios, we are strictly optimizing for prediction accuracy and RMSE rather than training time as the CSPELM default hyper-parameter settings showed a good balance between the two.

First, for the forest optimization technique, we know that the SPELM contributes a satisfactory starting set of hidden weights and biases for the CSPELM to further perturb. Though, we hypothesize that due to the general lack of explainability associated with neural network weights and biases that any number of SPELM starting points in the n-dimensional gradient space could produce the same or better results. That is, creating a forest for which each root consists of a separately trained SPELM could give starting positions to the CSPELM that are in different places in the n-dimensional gradient space. Thus, one of the SPELM root's starting position could be close to the global minimum. We term this the CSPELM Forest (CSPELMF).

The CSPELMF can be seen in Algorithm 3. We utilize the the same CSPELM training theory as seen in Algorithm 1 but change the generation of the SPELM. Specifically, instead of using a single SPELM as input, we train a number of SPELM equal to the $\#Roots$ user defined variable and store them in the $Roots$ variable. So, suppose that $\#Roots$ equals 5. Therefore, 5 SPELM are trained with the number of iterations used per SPELM training session varying at model one being 1 iteration, model two being 2 iterations, etc. This gives us 5 SPELM models trained with differing amounts of training iterations to assist in lowering the overall training time and while still retaining the benefits of the SPELM. Following, a CSPELM is trained per SPELM root. Finally, from Algorithm 3, the user defined hyperparameters are ε, ε_Mult, $\#Roots$, and $CSPELM\_Iter$. The two functions used in the algorithm are the SPELM and CSPELM functions which both simply train their respective models and returns the results.

Next, for the ε constraint range-finder, we consider the problem of escaping local minimums. As previously discussed, the SPELM produces a point in gradient space giving satisfactory results. It is possible in the gradient space that this point is a local minimum that we can escape. Therefore, we hypothesize that by gradually increasing the ε constraint size, it can potentially escape this local minimum and produce better results. This method

Algorithm 3: The CSPELM forest optimization technique. The user defined hyper-parameters are ε, ε_Mult, $\#Roots$, and $CSPELM\_Iter$. An SPELM is trained at a varying number of training iterations for each root in the forest. Then, each root is used to train an individual CSPELM.

```
Initialization:
ε := float;
ε_Mult := int;
Best_Result := float;
Best_Model := list();
Roots := list();
#Roots := int;
CSPELM_Iter := int;
for i ← 0 to #Roots do
    Train_Iterations = i + 1;
    Roots.append(SPELM(Train_Iterations, data));
end
for Root in Roots do
    for i ← 0 to CSPELM_Iter do
        for j ← ε_Mult to 1 do
            Model = CSPELM(Root, data, ε * j);
            if Best_Result < Model.Result then
                Best_Model = Model;
                Best_Result = Model.Result;
            end
        end
    end
end
```

effectively mimics taking larger steps in gradient descent backpropagation. We term this the CSPELM Range-finder (CSPELMR).

The CSPELMR can be seen in Algorithm 4. This range-finder methodology functions by creating a list of both ε and ε_Mult variables which allows for gradually increasing size combinations between the two. Thus, if the SPELM starting point is within a local minimum, then by increasing the ε constraint, we increase the probability of escaping the local minimum. For example, say that the ε and ε_Mult lists are initialized to $[3, 6]$ and $[0.02, 0.03]$, respectively. Then, Algorithm 4 will create combinations of $[3, 0.02]$, $[3, 0.03]$, $[6, 0.02]$, and $[6, 0.03]$ for training the CSPELM. Finally, from Algorithm 4, the user defined

hyper-parameters are ε_List, ε_Mult_List, and $CSPELM\_Iter$. The only function used is the CSPELM function which trains the model and returns the results.

Algorithm 4: The CSPELM ε constraint range-finder. The user defined hyper-parameters are ε_List, ε_Mult_List, and $CSPELM\_Iter$. This creates multiple ε and ε_Mult combinations to increase the probability of escaping local minimums by gradually increasing the ε constraint size. This effectively mimics taking larger steps in gradient descent.

```
Initialization:
ε_List := list();
ε_Mult_List := list();
CSPELM_Iter := int;
Best_Result := float;
Best_Model := list();
for ε_Mult in ε_Mult_List do
    for ε in ε_List do
        for i ← 0 to CSPELM_Iter do
            for j ← ε_Mult to 1 do
                Model = CSPELM(SPELM, data, ε * j);
                if Best_Result < Model.Result then
                    Best_Model = Model;
                    Best_Result = Model.Result;
                end
            end
        end
    end
end
```

6.3.1 Public Dataset Results

We now perform experiments with the CSPELMF and CSPELMR using the same datasets as used for the CSPELM experiments (Table 6.1). We compare these with LM backpropagation as well as dataset specific hyper-parameter optimized CSPELM. We choose not to include the ELM and SPELM as we have already shown that the CSPELM with default hyper-parameter settings consistently outperforms them.

The experimental setup used in this section is similar to the previous experiments

in Section 6.2.1 with a few changes. The data splits are still 80% and 20% for training and test, respectively, unless a training and test split was provided. The number of fully connected hidden nodes is once again dictated by $\lfloor(2/3 * \#InNodes) + \#OutNodes\rfloor$ [29]. Dataset E continues to prove difficult to train with limited hardware. Thus, we again choose 40 hidden nodes for this dataset. All data were normalized to $[-1, 1]$. The sigmoid activation function of $1/(1 + e^{-x})$ was used for all models. The following are the hyper-parameter settings needed for each of the algorithms. For the CSPELM, the ELM_Iter, $CSPELM\_Iter$, ε, and ε_Mult hyper-parameters are set. Then, for the CSPELMF, they are the $\#Roots$, $CSPELM\_Iter$, ε, and ε_Mult hyper-parameters. Following, for the CSPELMR, the hyper-parameters ELM_Iter, ε_List, and ε_Mult_List are set. Lastly, for LM backpropagation, the only hyper-parameter is the $LMIterations$ variable. The values of these variables differ per dataset and can be seen in Table 6.4. The only exception is the ε_Mult variable which is set to 10 for all cases.

The hardware used for these experiments was an Intel Core i5-9600K CPU at 3.70 GHz, NVIDIA GeForce GTX 1070 Ti FTW Edition, and 16.00 GB of DDR4 RAM. The CSPELM, CSPELMF, and CSPELMR were coded in Python 3.7.1 with the random seed 2998322090 for reproducibility. The LM backpropagation was coded using MATLAB R2018a with the pattern recognition toolbox.

We present the experimental results in Tables 6.5 and 6.6 for the classification and regression datasets, respectively. Similar to how the results were presented in Section 6.2.1, we show the mean $\pm$ standard deviation results over multiple training attempts, final/best results, mean $\pm$ standard deviation training time in seconds, total time in seconds, and number of hidden nodes in the SLFN. The bold values represent the highest performance for each dataset. From the classification results in Table 6.5, we can see that our three methods further outperformed the LM backpropagation as compared to the previous testing. As for the CSPELMF and CSPELMR, either one of the two, or both, matched or outperformed

the hyper-parameter optimized CSPELM. The most impressive result is from Dataset H in which all three of our methods produced a 100% test accuracy. Next, for the regression results in Table 6.6, while the optimized hyper-parameter settings of the CSPELM showed further improvements over previous testing, the CSPELMF and CSPELMR appear to have difficulties with regression datasets. The only instances which either the CSPELMF or CSPELMR outperformed LM backpropagation are datasets L and M.

Table 6.4: The hyper-parameter settings for each algorithm per dataset. The "—" means that the algorithm does not require that hyper-parameter. The only hyper-parameter missing is the ε_Mult variable which is set to 10 for all cases.

Dataset	Algorithm	$\#Roots$	ε_Mult_List	ε_List	ELM_Iter	CSPELM_Iter	ε	LM Iterations
A	CSPELM	—	—	—	400	30	0.05	—
	CSPELMF	19	—	—	—	5	0.01	—
	CSPELMR	—	2, 4, 6, 8	0.01, 0.02, 0.03, 0.04	50	5	—	—
	LM	—	—	—	—	—	—	35
B	CSPELM	—	—	—	250	30	0.04	—
	CSPELMF	25	—	—	—	5	0.01	—
	CSPELMR	—	1, 5, 10, 20	0.01, 0.02, 0.03, 0.04	50	5	—	—
	LM	—	—	—	—	—	—	30
C	CSPELM	—	—	—	175	10	0.02	—
	CSPELMF	25	—	—	—	5	0.01	—
	CSPELMR	—	1, 5, 10	0.01, 0.02, 0.03, 0.04, 0.05	50	5	—	—
	LM	—	—	—	—	—	—	50
D	CSPELM	—	—	—	50	5	0.01	—
	CSPELMF	12	—	—	—	5	0.01	—
	CSPELMR	—	1, 5	0.01, 0.02, 0.03, 0.04, 0.05	50	5	—	—
	LM	—	—	—	—	—	—	40
E	CSPELM	—	—	—	500	30	0.03	—
	CSPELMF	50	—	—	—	5	0.01	—
	CSPELMR	—	1, 5, 10	0.01, 0.02, 0.03, 0.04, 0.05	50	5	—	—
	LM	—	—	—	—	—	—	30
F	CSPELM	—	—	—	1000	50	0.05	—
	CSPELMF	20	—	—	—	5	0.01	—
	CSPELMR	—	10, 20	0.01, 0.02, 0.03, 0.04, 0.05	50	5	—	—
	LM	—	—	—	—	—	—	45
G	CSPELM	—	—	—	300	5	0.03	—
	CSPELMF	10	—	—	—	5	0.01	—
	CSPELMR	—	1, 5, 10	0.01, 0.02, 0.03, 0.04, 0.05	50	5	—	—
	LM	—	—	—	—	—	—	30
H	CSPELM	—	—	—	500	5	0.02	—
	CSPELMF	35	—	—	—	5	0.01	—
	CSPELMR	—	1, 5, 10	0.01, 0.02, 0.03, 0.04, 0.05	50	10	—	—
	LM	—	—	—	—	—	—	50
I	CSPELM	—	—	—	25	5	0.01	—
	CSPELMF	10	—	—	—	5	0.01	—
	CSPELMR	—	1, 5	0.01, 0.02	50	5	—	—
	LM	—	—	—	—	—	—	5
J	CSPELM	—	—	—	50	5	0.02	—
	CSPELMF	20	—	—	—	5	0.01	—
	CSPELMR	—	1, 5, 10	0.01, 0.02, 0.03, 0.04	50	5	—	—
	LM	—	—	—	—	—	—	25
K	CSPELM	—	—	—	100	5	0.02	—
	CSPELMF	15	—	—	—	5	0.01	—
	CSPELMR	—	1, 5, 10	0.01, 0.02, 0.03, 0.04	50	5	—	—
	LM	—	—	—	—	—	—	5
L	CSPELM	—	—	—	45	5	0.01	—
	CSPELMF	20	—	—	—	5	0.01	—
	CSPELMR	—	1, 5	0.01, 0.02, 0.03, 0.04, 0.05	50	5	—	—
	LM	—	—	—	—	—	—	20
M	CSPELM	—	—	—	45	5	0.01	—
	CSPELMF	10	—	—	—	5	0.01	—
	CSPELMR	—	1, 5, 10	0.01, 0.02, 0.03	50	5	—	—
	LM	—	—	—	—	—	—	15

Table 6.5: The classification results for the CSPELM, CSPELMF, CSPELMR, and LM Backpropagation. The bold values represent the best results for each dataset. The most notable result is the 100% classification of dataset H with the CSPELMF, CSPELMR, and dataset specific hyper-parameter setting CSPELM algorithms.

Dataset	Algorithm	Accuracy %	Final/Best Accuracy %	Training Time (s)	Total Time (s)	#Hidden Nodes
A	CSPELM	75.66 ± 2.61	**80.65**	0.03 ± 1*10(-3)	20.39	7
	CSPELMF	74.34 ± 2.39	**80.65**	4.09 ± 2.83	155.61	7
	CSPELMR	**75.99 ± 1.79**	78.71	0.81 ± 0.38	13.73	7
	LM	70.34 ± 3.29	78.07	0.27 ± 0.35	9.42	7
B	CSPELM	**94.19 ± 2.53**	**98.61**	0.04 ± 2*10(-3)	22.76	24
	CSPELMF	92.00 ± 2.20	97.22	1.31 ± 0.79	65.41	24
	CSPELMR	91.38 ± 3.09	**98.61**	1.93 ± 1.49	32.73	24
	LM	85.23 ± 5.28	94.44	0.54 ± 0.42	16.22	24
C	CSPELM	**81.64 ± 3.03**	**87.3**	0.02 ± 1*10(-3)	4.36	10
	CSPELMF	80.03 ± 2.47	**87.3**	0.52 ± 0.33	25.82	10
	CSPELMR	78.58 ± 3.45	85.71	0.45 ± 0.29	7.23	10
	LM	73.97 ± 4.49	80.95	0.25 ± 0.29	12.23	10
D	CSPELM	**97.37 ± 1.06**	98.39	0.06 ± 4*10(-3)	5.81	22
	CSPELMF	96.59 ± 0.96	98.39	1.73 ± 1.36	41.48	22
	CSPELMR	96.15 ± 2.05	98.39	7.91 ± 2.97	86.98	22
	LM	95.63 ± 8.22	**99.19**	0.68 ± 0.51	27.35	22
E	CSPELM	89.13 ± 6.94	96.61	0.40 ± 4*10(-3)	305.85	40
	CSPELMF	92.63 ± 3.23	**97.65**	13.82 ± 6.14	1381.47	40
	CSPELMR	84.70 ± 13.74	97.13	11.15 ± 7.37	178.44	40
	LM	**93.52 ± 6.59**	97.39	707.68 ± 410.81	21230.32	40
F	CSPELM	90.96 ± 1.39	93.17	0.63 ± 0.01	942.39	30
	CSPELMF	**92.28 ± 0.94**	93.17	24.41 ± 16.31	976.26	30
	CSPELMR	91.17 ± 1.06	**93.51**	58.56 ± 20.16	644.17	30
	LM	82.01 ± 4.21	85.11	14.18 ± 2.59	638.09	30
G	CSPELM	94.83 ± 1.19	96.67	0.25 ± 4*10(-3)	85.45	19
	CSPELMF	94.11 ± 1.01	96.67	12.88 ± 10.41	257.66	19
	CSPELMR	94.70 ± 1.06	96.67	6.74 ± 4.49	107.82	19
	LM	**95.64 ± 2.99**	**98.57**	2.43 ± 0.60	73.01	19
H	CSPELM	**99.86 ± 0.09**	**100.00**	1.18 ± 0.01	641.59	52
	CSPELMF	99.67 ± 0.17	**100.00**	29.77 ± 15.37	2083.96	52
	CSPELMR	99.79 ± 0.12	**100.00**	62.82 ± 42.09	1005.14	52
	LM	95.64 ± 0.83	97.16	205.58 ± 29.65	10278.75	52

Table 6.6: The regression results for the CSPELM, CSPELMF, CSPELMR, and LM Back-propagation. The bold values represent the best results for each dataset. As compared to the default hyper-parameter settings, these dataset specific settings show impressive improvements.

Dataset	Algorithm	RMSE	Final/Best RMSE	Training Time (s)	Total Time (s)	#Hidden Nodes
I	CSPELM	**0.14 - 4*10(-3)**	**0.13**	2.67 - 0.02	199.59	5
	CSPELMF	0.16 - 0.01	0.16	74.65 - 60.17	1492.91	5
	CSPELMR	0.15 - 4*10(-3)	0.15	59.19 - 44.40	295.95	5
	LM	0.14 - 0.02	**0.13**	2.79 - 0.71	13.94	5
J	CSPELM	**12.29 - 0.56**	**11.55**	0.03 - 2*10(-3)	3.36	6
	CSPELMF	18.05 - 2.84	17.27	1.03 - 0.69	41.14	6
	CSPELMR	18.05 - 4.45	13.16	0.96 - 0.63	12.53	6
	LM	17.25 - 2.70	12.33	0.19 - 0.31	4.91	6
K	CSPELM	10.68 - 0.29	10.33	0.32 - 0.01	47.52	11
	CSPELMF	11.99 - 0.49	11.72	9.34 - 6.86	280.28	11
	CSPELMR	12.20 - 0.69	10.96	9.18 - 6.06	119.35	11
	LM	**10.00 - 0.30**	**9.79**	0.65 - 0.66	3.24	11
L	CSPELM	**5.44 - 0.78**	**4.96**	0.18 - 0.01	17.37	3
	CSPELMF	17.49 - 8.89	11.39	5.62 - 3.78	224.96	3
	CSPELMR	14.96 - 8.91	6.28	3.38 - 2.54	37.16	3
	LM	15.65 - 1.76	14.09	0.27 - 0.35	5.29	3
M	CSPELM	**3.81 - 0.73**	**2.77**	0.04 - 2*10(-3)	3.41	4
	CSPELMF	10.40 - 3.64	11.12	1.01 - 0.81	20.15	4
	CSPELMR	7.74 - 2.16	4.78	1.03 - 0.66	10.27	4
	LM	6.48 - 1.05	5.62	0.28 - 0.37	4.18	4

7

Predicting the Change in State of the Heart with Synthetic Heart Chamber Surface Area Data

Previously, in Chapter 3, we devised the novel wearable ultrasound vest to create a near 3D model of the heart in real-time. The system was shown to be capable of modeling the end-systolic and end-diastolic stages. From these stages, we can extract features and use them with our CSPELM prediction algorithm, as discussed in Chapter 6, which can assist medical professionals by adding the actual autonomy to the wearable vest. Now, we continue to build upon the individualized prediction scheme by providing the workflow for creating multiple synthetic datasets, extracting features from the 3D model of the heart, and predicting the change in state of the human heart [47]. Note that we define the "state" of the human heart to simply be the current condition it is in, whether that be healthy or abnormal. So, a prediction does not necessarily imply a good or bad state.

We begin by first providing a representative literature review on Cardiovascular Dis-

ease prediction methodologies in Section 7.1. Then, Section 7.2 details the features that we can extract from the wearable ultrasound transducer vest and the corresponding prediction possibilities. Next, Section 7.3 discusses how the synthetic datasets are generated. Following, Section 7.4 provides the justification for using the algorithm we created in Chapter 6. Finally, Section 7.5 shows the numerical experiments performed on the synthetic datasets.

7.1 Representative Cardiovascular Disease Prediction Literature Review

We begin with an article by Park et al. which described a system that is capable of monitoring the arterial pulse [105]. The authors utilize a self-powered piezoelectric pressure sensor that is placed on the wrist. This sensor measures radial and carotid pulses via the pressure sensor from the arteries near the surface of the skin. Their design of the sensor consists of a small patch containing the pressure sensor and wireless transmission capabilities to transfer data to a smartphone for processing and continuous monitoring. The system's goal is to assist in detecting early onset CVD.

An article which also creates a sensor patch is written by Imani et al. that introduces a biochemical sensor patch to measure lactate which is used in conjunction with an electrocardiogram [63]. The device is placed over the fourth intercostal space to assist the electrocardiogram's detection capabilities. The inclusion of the biochemical sensor patch for additional measurements is seen to be lacking in medicine and current research. The results from their testing showed that their additional patch with the electrocardiogram allows for measuring both biochemical and physiological signals simultaneously to assist in further inference during the diagnosis process. This warrants further investigation in including additional non-invasive sensors in medical devices to increase information intake.

Following, a real-time continuous monitoring system consisting of sensors integrated

into a shirt was created by Park et al. [106]. The authors are interested in long term monitoring but find difficulties with doing this using the traditional wet electrode for monitoring arrhythmia. Therefore, they designed a shirt which houses dry electrodes that obtain electrocardiogram signals to detect cardiac arrhythmia in adults between the ages of 40 and 60. Testing showed that their system was capable of both mobile as well as immobile detecting and showed a cardiac arrhythmia classification accuracy of 98.2%.

An article by Oh et al. shows the design of an automated system that can assist medical professionals in diagnosing five forms of arrhythmia [102]. The five forms are the normal sinus rhythm, left bundle branch block, right bundle branch block, atrial premature beats, and premature ventricular contraction. The authors created a single CNN combined with Long Short-Term Memory (LSTM) capabilities for this monitoring task. The LSTM portion is implemented before the fully connected layer of the CNN. This adds a memory state aspect to the network allowing the network to "remember" or "forget" data as needed temporally. From testing, the authors showed a 98.10% classification accuracy of the five arrhythmia states.

Another arrhythmia classification article by Alickovic and Subasi showed their system using a Random Forest (RF) classifier [4]. Electrocardiogram signal classification is the current primary method for detecting arrhythmia. Therefore, improving the classification accuracy of this task is critical. So, the authors create a workflow which begins by performing signal de-noising using Multiscale Principal Component Analysis. Then, feature extraction is performed by taking electrocardiogram signals of heart beat cycles and breaking them down into multiple frequencies using discrete wavelet transformations. These multiple frequencies are then used as inputs into the RF classifier. Their results showed 99.33% and 99.95% classification accuracies for the MIT-BIH and St. Petersburg Institute of Cardiological Technics 12-lead Arrhythmia Database, respectively.

In an attempt to improve explainability of a heart disease classification system, Kasbe

and Pippal used Fuzzy Logic as their classifier of choice [71]. That is, the authors focus on classification of heart disease risk in patients. The authors used 10 input features, 29 membership functions, and 86 fuzzy rules in their design. Testing showed a classification accuracy of 93.33% as compared to seven other methodologies in which the previous highest performing model was 93.27%.

An article by Manogaran et al. took a hybrid approach for heart disease diagnosis by utilizing an Adaptive Neuro-Fuzzy Inference System (ANFIS) with Multiple Kernel Learning (MKL) [88]. The authors used MKL as a differentiation technique between healthy patients and patients with some form of heart disease. Next, the ANFIS is used for classification by using the outputs from the MKL as inputs to train from. The testing results showed 98% and 99% for sensitivity and specificity, respectively.

Finally, similar to our conjecture on detecting abnormality during the hearts operation, Potes et al. used an ensemble classifier for detecting abnormal heart sounds from phonocardiogram data [110]. The authors used a combination of Adaptive Boosting (AdaBoost) and a CNN for this task. Following, a rule for deciding if either the AdaBoost of CNN algorithms outputs were larger than an empirically determined threshold was created. If this threshold was crossed, then the phonocardiogram input was abnormal. Their results showed a sensitivity, specificity, and overall score of 0.9424, 0.7781, and 0.8602, respectively.

7.2 Individualized Learning and Prediction Scheme

The wearable ultrasound vest, as discussed in Chapter 3, will create a near 3D model of the heart in real-time. This will of course assist the medical professional in prognosis and diagnosis as they are more focused on the actual produced 3D image rather than the imaging process. Though, our design is capable of continuous autonomous monitoring as

the medical professional will not always be present to examine the monitor. Therefore, an individualized learning and prediction scheme is needed to assist the medical professional in prognosis and diagnosis efforts as well as to alert them if anomalies occur while they are not present. Thus, during the 3D modeling process of the heart from the wearable ultrasound vest, it will also gather data from the near 3D model which will be used to train a neural network in real-time. This neural network will then be used for heart state prediction with respect to that individual. Specifically, we will be extracting the following features listed in Table 7.1.

Table 7.1: The features to be extracted from the wearable ultrasound transducer vest during the near 3D modeling of the heart. These features will be used as inputs to the individualized learning and prediction scheme.

Feature	Description
Left ventricular end-diastolic dimension	The approximate surface area of the left ventricle in the end-diastolic state.
Left ventricular end-systolic dimension	The approximate surface area of the left ventricle in the end-systolic state.
Right ventricular end-diastolic dimension	The approximate surface area of the right ventricle in the end-diastolic state.
Right ventricular end-systolic dimension	The approximate surface area of the right ventricle in the end-systolic state.
Left atrium end-diastolic dimension	The approximate surface area of the left atrium in the end-diastolic state.
Left atrium end-systolic dimension	The approximate surface area of the left atrium in the end-systolic state.
Right atrium end-diastolic dimension	The approximate surface area of the right atrium in the end-diastolic state.
Right atrium end-systolic dimension	The approximate surface area of the right atrium in the end-systolic state.
Left ventricle motion score	The measure of the left ventricle's difference between the end phases of both systolic and diastolic dimensions.
Right ventricle motion score	The measure of the right ventricle's difference between the end phases of both systolic and diastolic dimensions.
Left atrium motion score	The measure of the left atrium's difference between the end phases of both systolic and diastolic dimensions.
Right atrium motion score	The measure of the right atrium's difference between the end phases of both systolic and diastolic dimensions.
Average overall motion score	The average measure of the four chamber's walls motion scores.

During the execution of the 3D modeling by the wearable vest, we can gather measurements of surface areas (SA) for each of the hearts four chambers. Recall that the system is modeling the physical heart and the chamber activations in real-time. So, from Table 7.1, we can see that the first eight features will record the surface area of the end-diastolic state and end-systolic state for all four chambers, respectively. We only need the end states for each chamber as the end state of the diastolic state is in fact the start state of the systolic state and vice versa. Furthermore, there is a motion score feature for each chamber as well.

This motion score is the difference between the end systolic and diastolic dimensions. Finally, there is the average overall motion score which is simply the average calculated from the four chamber motion scores. These 13 features will be recorded per full cardiovascular cycle. That is, starting with diastolic state n, once the heart reaches diastolic state $n + 1$, a single data point will have been collected. By utilizing the 90 BPM as the worst case scenario, discussed in Section 3.2, we thus obtain an approximate 90 data points per minute and approximately 5,400 data points per hour.

With these features, we can use them as inputs into a neural network to be trained with respect to a single individual's cardiovascular state. It will be an unsupervised regression network as during data gathering we are unable to label the data without the constant assistance from a medical professional which is counter intuitive to our research goals. Note that abnormal predictions with respect to the individual's trained neural network does not necessarily mean the heart is improving or degrading. It simply shows that there is a change in functionality as opposed to the previous state. This would notify the medical professionals to examine the current state of the heart to make further inferences. There are four potential use-cases that this prediction scheme can be utilized with that corresponds to the research goals of the wearable ultrasound vest as shown below.

1. The vest is placed on a patient admitted to the hospital. Therefore, the data the neural network will be trained on will be of a heart not functioning properly. Thus, when the network produces a prediction that is not aligned with what it was trained on, then the heart is functioning differently from when the patient was admitted.

2. The vest is placed on a patient that has normal cardiac functionality but is suspected of cardiovascular problems eminently or in the near future.

3. The vest is placed on an individual working in the public sector and obtains a considerable amount of normal heart functioning data. The predictions from this use-case should be more differentiable due to the abundance of data the network has been

trained on.

4. The neural network is pre-trained on a dataset of multiple individuals with normal cardiovascular functionality.

Of these four use-cases, number four is seen as a worst-case scenario as it obviously does not pertain to an individual patient's cardiovascular health. Though, it can be used as a comparative measure during the implementation phase. The other three use-cases focus on individualized data which allows for a more personalized prognosis and diagnosis experience.

7.3 Generating Synthetic Data

As this dissertation is not inherently clinical and as no dataset currently exists, to the best of our knowledge, we move to generate a synthetic dataset. That is, we could not find literature detailing the external surface areas of the four heart chambers. Though, there is literature detailing the internal volumes of the four chambers. With this information and a small workflow, we can create our own synthetic dataset.

From Aune et al., we can extract reference ranges for the Left and Right Atria at both end-systolic and end-diastolic internal volumes [9]. Next, Maceira et al. details the Right Ventricle's end-systolic and end-diastolic volume reference ranges [85]. Finally, Maceira et al. published a separate article with the ranges for the Left Ventricle's end-systolic and end-diastolic volumes [86]. Now, we have the required information to create internal chamber volume ranges for all four chambers. Note that these articles report volumes of healthy individuals. Also, we took the liberty to create averages between age groups and sexes as reported in the literature as well as the reported average Body Surface Areas (BSA). Table 7.2 presents the mean ± standard deviation volume values for the Left Ventricle End-Diastolic (LVED), Left Ventricle End-Systolic (LVES), Right Ventricle End-

Diastolic (RVED), and Right Ventricle End-Systolic (RVES) states, respectively. Also, Table 7.2 presents the reference volume values for the Left Atrium End-Diastolic (LAED), Left Atrium End-Systolic (LAES), Right Atrium End-Diastolic (LAED), and Right Atrium End-Systolic (LAES) states, respectively. As some values were reported as ranges adjusted for the standard deviation and others as simply the mean ± standard deviation, we will take the mean ± standard deviation and create our own upper and lower bounded ranges from the mean using the standard deviation.

Table 7.2: The reported healthy internal heart volumes for the four heart chambers. Note, the values are reported as both strict ranges as well as mean ± standard deviation.

LVED Volume	**LVES Volume**	**RVED Volume**	**RVES Volume**
77.5 ± 8.8ml/m^2	25.7 ± 5.1ml/m^2	78 ± 11ml/m^2	27 ± 7ml/m^2
LAED Volume	**LAES Volume**	**RAED Volume**	**RAES Volume**
15-41ml/m^2	5-19ml/m^2	18-47ml/m^2	5-20ml/m^2

Now, as mentioned previously, we must construct a short workflow to change these volume values to external surface area values. To do this, we estimate the internal shape of each heart chamber to be a sphere. This is a rough estimation and we will compensate for it in the following workflow. We take the provided values in Table 7.2 and input them into Equation 7.1, the surface area A of a sphere given the volume V, and the six steps below to approximate the external surface area of each heart chamber.

$$A = \pi^{1/3}(6V)^{2/3} \tag{7.1}$$

1. Take the volumes from Table 7.2 reported as ml/m^2 (it is common practice for medical professionals to correct these metrics for BSA) and multiply by the average BSA in the corresponding articles to remove the m^2.

2. Convert milliliters to liters by dividing by 1000.

3. Input these new values into Equation 7.1 to calculate the approximate surface area of

the sphere.

4. Convert from m^2 to cm^2 by multiplying by 10,000. Unfortunately, this gives unrealistically large numbers. For example, the LVED SA from this step results in the range 12,274.94cm^2-14,093.11cm^2. Thus, we include the next step to adjust these values accordingly.

5. Scale the previous results by dividing by 100.

6. Multiply the result by $1/3$ to compensate for the rough shape estimation. Also, as we are only using a portion of the external surface area of the heart chamber.

With the provided workflow and the assumptions of using a sphere to represent the internal structure of each heart chamber while compensating with the multiplication of $1/3$, we produce the approximated external healthy heart surface areas in Table 7.3.

Table 7.3: The approximated ranges of healthy external SA for all four heart chambers.

LVED SA	**LVES SA**	**RVED SA**	**RVES SA**
40.9-46.9cm^2	18.2-23.8cm^2	39.9-48.2cm^2	17.8-25.4cm^2
LAED SA	**LAES SA**	**RAED SA**	**RAES SA**
15.1-29.6cm^2	7.3-17.7cm^2	17.1-32.4cm^2	7.3-18.3cm^2

Now, to have comparison heart states to judge our systems effectiveness against, we must create a set of abnormal data as well. Therefore, we follow the same workflow methodology we created for the healthy heart volume data and use it for abnormal heart data. Specifically, we examine an article for AFb by Zhang et al. [150], Primary Chronic Mitral Regurgitation (PCMR) by Borg et al. [18], and Post-Myocardial Infarction (PMI) by Gardner et al. [38]. Of these three abnormal heart conditions, AFb is associated with the Left Atrium and both PCMR and PMI are associated with the Left Ventricle. The pre-conversion and post-conversion values for these these three abnormal heart conditions are seen in Tables 7.4, 7.5, and 7.6 for AFb, PCMR, and PMI, respectively. Using these three abnormal heart condition values, we can generate three abnormal test datasets. Note that

we utilize the normal heart SA for the missing chambers of these tables.

Table 7.4: The ranges of volumes and approximated external surface areas for the Left Atrium from the AFb abnormality.

LAED Volume	LAES Volume	LAED SA	LAES SA
27.4 ± 1.5ml/m^2	22.3 ± 1.2ml/m^2	19.5-21.0cm^2	17.1-18.3cm^2

Table 7.5: The ranges of volumes and approximated external surface areas for the Left Ventricle from the PCMR abnormality.

LVED Volume	LVES Volume	LVED SA	LVES SA
74.8 ± 21.3ml/m^2	26.1 ± 8.8ml/m^2	34.2-50.6cm^2	16.1-25.8cm^2

Table 7.6: The ranges of volumes and approximated external surface areas for the Left Ventricle from the PMI abnormality.

LVED Volume	LVES Volume	LVED SA	LVES SA
86 ± 28ml/m^2	44 ± 22ml/m^2	37.7-59.4cm^2	19.8-41.2cm^2

With the external heart chamber SA data ranges, we can now create multiple synthetic datasets using statistical bootstrap sampling of the mean. That is, we create a distribution with respect to each of the heart chamber SA ranges and perform the bootstrap sampling. For all the generated datasets, the distribution is the minimum to the maximum of the ranges provided in the tables showing the SA ranges. The value increments within the distribution are set to 1 (min:1.0:max). Finally, we bootstrap n number of data points to create either normal or abnormal datasets. The process of bootstrapping uses the mean statistic for the sampling process as all the literature utilizes the mean as well.

For choosing n, we take two separate approaches for healthy and abnormal data. First, for the healthy data, we take the estimated 90 data points produced per minute from the ultrasound transducer vest in Section 3.2. In order to determine how much data is needed to effectively train the heart state prediction system, we will generate five separate datasets. That is, dataset one will contain 90 data points to represent one minute of elapsed time,

dataset two will contain 180 data points to represent two minutes of elapsed time, etc. Also, each dataset will build off of each other to simulate a continuous data gathering process. So, dataset two will contain the exact same 90 data points from dataset one. This process allows us to closely simulate the actual use-case of the vest. This process will create five datasets representing one to five minutes of data.

Next, for the abnormal data, we choose to sample 360 data points for each of the abnormal heart conditions. This will allow us to test each heart problem individually which will show use if the system is capable of detecting a different state as compared to the one it was trained on. Finally, we also generate one extra normal heart dataset of 360 data points to test if the system will predict false positives. All healthy and abnormal test data is generated with the same process as previously described.

7.4 Choosing a Machine Learning Algorithm

For the differentiation between healthy and abnormal heart states, we choose the neural network as our classifier. To train the network, we use our algorithm created in Chapter 6, the CSPELM. Though, we do not include the optimization techniques of the CSPELMF or CSPELMR here. During experimentation, we found cases of over generalization with the default settings. By implementing the CSPELMF, CSPELMR, or increasing the complexity of the CSPELM hyper-parameters, we only exacerbate the over generalization. Furthermore, as shown in previous testing, the time complexity for these three methods generally loses the real-time capabilities. Thus, we choose to use the default CSPELM to prove the theory of predicting the change in state of the heart. Future work is to utilize real data and retest our three accuracy optimized methods. The default hyper-parameter CSPELM though showed real-time capabilities in many cases and also high performance in the regression cases.

As stated in Section 7.2, we will be using the neural network as a regression model. In practice, it is not feasible to consistently label the data during usage. Therefore, we choose to compute a target variable from our features in Table 7.1. This target variable is the length of the feature vector as shown in Equation 7.2. In the equation, T is the target variable, F is a feature in the feature vector, and n is the number of features in the vector which in this case is 13. With this equation, we can now have fully generated synthetic datasets which we will use in conjunction with the CSPELM to train a neural network for heart state predictions.

$$T = \sqrt[n]{\sum_{i=1}^{n} F_i^2} \tag{7.2}$$

7.5 Numerical Experiments

For the numerical experiments, we take the following steps in the model training process. We will train five models on the one to five minute datasets to determine how well the CSPELM trains with the varying amount of data. The large healthy and abnormal datasets will be used as the final test case for this determination. The comparison of the results will help us decide how much data is necessary in order to train the network during practice. The less data needed allows for faster recognition of the current heart state and thus allowing for detecting changes in said state sooner. Finally, we will test each model with the three abnormal heart datasets to examine how effective each model is at differentiating heart states.

In order for a model to determine a change in state, we compute the 95% confidence interval of the test data outputs from the training network. Note that the one to five minute datasets are all split into 80% training and 20% test sets, respectively. This interval will act as the state change threshold classifier. With this threshold, when the healthy and abnormal

test data are passed through each model, we can examine how well each model predicts whether a data point is abnormal or not. Specifically, we want to compare how many abnormal data points are predicted outside of the threshold as compared to healthy data points. Once the comparison is performed, we can empirically determine an amount of data that needs to be predicted outside the threshold to warrant a detection of a change in heart state and notification of a medical professional.

The synthetic data was generated using MATLAB R2018a. The CSPELM was coded in Python 3.7.6 with the random seed of 101 for reproducibility purposes. We use the same rules for network architecture and training as done in Chapter 6. That is, we use the rule of thumb of $\lfloor(2/3 * \#InNodes) + \#OutNodes\rfloor$ [29] for deciding the number of hidden nodes in the SLFN. The sigmoid activation function of $1/(1 + e^{-x})$ was used for all nodes. All data were normalized to [-1, 1]. For the CSPELM hyper-parameter settings, we used the default settings of 5, 10, and 0.01 for the $CSPELM\_Iter$, ε_mult, and ε variables, respectively. Finally, the hardware used for training was an Intel(R) Core i5-9600K CPU at 3.70GHz, an NVIDIA GeForce GTX 1070 Ti FTW Edition GPU, and 16.00 GB of DDR4 RAM.

The results of our experiments are presented in Tables 7.7 and 7.8 showing the results of the training process and the testing of the abnormal datasets, respectively. The regression results are calculated using the RMSE. As we can see from Table 7.7, all five models performed similarly with respect to the best RMSE. Though, there was a large amount of test data predicted outside of the 95% confidence interval. This occurred for every model except the one minute model. Specifically, the amount of test data predicted outside the confidence interval was 81.94%, 75.28%, 99.72%, and 98.61% for the 2, 3, 4, and 5 minute models, respectively. Furthermore, we can see a slight rise in the mean RMSE of 0.0225, 0.0321, 0.0306, and 0.0196 for the 2, 3, 4, and 5, minute models, respectively. Interestingly though, the relative RMSE is still quite similar between all five models. Finally, the training

time never reached over 2 seconds for any of the models. This shows that each model stays within the real-time constraint we imposed for the system.

Table 7.7: Results of the SLFN training using the CSPELM algorithm. Results are reported as mean ± standard deviation RMSE and best RMSE. Training time is reported in seconds (s) per training iteration. All results are of the 20% test splits.

Model	Mean RMSE	Best RMSE	Mean Training Time (s)	Total Time (s)	Amount of Test Data Outside 95% Interval	Percentage of Test Data Outside 95% Interval
1 Minute	0.0169 ± 0.0079	0.0032	0.0042 ± 0.0007	0.4219	121/360	**33.61%**
2 Minutes	**0.0225** ± 0.0027	0.0037	0.0080 ± 0.0011	0.8009	295/260	**81.94%**
3 Minutes	**0.0321** ± 0.0063	0.0036	0.0118 ± 0.0012	1.1747	271/360	**75.28%**
4 Minutes	**0.0306** ± 0.0039	0.0032	0.0152 ± 0.0010	1.5239	359/360	**99.72%**
5 Minutes	**0.0196** ± 0.0043	0.0032	0.0192 ± 0.0019	1.9204	355/360	**98.61%**

Table 7.8: Results of the models with respect to the abnormal heart test datasets. For each abnormal dataset, the results show the number of points lying outside the 95% confidence interval as well as the exact percentage of points.

Model	AFb Points	AFb %	PCMR Points	PCMR %	PMI Points	PMI %
1 Minute	257/360	**71.39%**	318/360	**88.33%**	217/360	**60.28%**
2 Minutes	288/360	80.00%	277/360	76.94%	303/360	84.17%
3 Minutes	330/360	91.67%	297/360	82.50%	264/360	73.33%
4 Minutes	341/360	94.72%	359/360	99.72%	300/360	83.33%
5 Minutes	356/360	98.89%	359/360	99.72%	351/360	97.50%

8

Discussion

In this chapter, we provide further analysis about both the wearable ultrasound transducer vest and CSPELM in Sections 8.1 and 8.2, respectively. Specifically, we will discuss the application possibilities and analyze the wearable vest's positives and limitations in more detail. Following, we will present use-cases of our CSPELM algorithm and further examine the results of the classification and regression datasets. Then, in Section 8.3, we discuss the numerical experiments from the heart state predictions as well as an interesting finding of the CSPELM.

8.1 Wearable Ultrasound Transducer Vest

The wearable ultrasound transducer vest has many discussion points. First, from both the 4x2 and 4x3 transducer orientations, the calculated FOV has near complete visual coverage of the heart. That is, the parasternal, frontal, and apical views have a vertical FOV of 119.2mm, 143.2mm, and 165.1mm, respectively. Recall that the average human heart has a height dimension of approximately 120mm. Both the frontal and apical views have more coverage than needed while the parasternal view is only missing 0.8mm total of the heart. In a worst case scenario, it is possible to impute this 0.8mm of missing FOV data. Though,

it is unlikely that this will be necessary. As previously discussed, the goal of the system is to create a near 3D model of the heart to examine chamber activation in real-time. So, this missing 0.8mm of data from one view is negligible.

This follows for the frame rate of the heart as well. The calculations show a FRPB of approximately 4 with the 4x2 orientation and a FRPB of approximately 3 for the 4x3 orientation. Emphasis is placed on the visualization of the heart's movement and not necessarily the maximum speed in which we see it. The goal is to capture the systolic and diastolic states to examine the chamber activations. The calculations show that it is indeed possible for either of the transducer orientations. Moreover, in the calculations to produce the FRPB, we were conservative with the user defined t activation interval variable as we are producing calculations for worst case scenarios. By lowering the value of t, we can increase the FRPB.

For the stereo vision 3D model recreation, there will of course need to be preprocessing steps before the model can be viewed on the ultrasound monitor. For both the frontal and apical views, we must consider the rotation of the transducers during the image gathering process. Recall that the frontal and apical views have a downward angle of 19.5° and 35.2°, respectively, to account for the rib bones at those locations. Therefore, image transformations for both of these views will be necessary to match the alignment of the parasternal view. Fortunately, as we know the angle in which to transform the produced images, this process is trivial. Though, the negative comes from the processing time addition to the system. In a hospital setting, the corresponding computer for processing should be able to handle this. Though, our second use case of public sector transportation employees could prove problematic depending on the computational power available.

Finally, the primary two use-cases for this wearable ultrasound transducer vest is of course the hospital and public sector transportation areas. Though, another interesting use-case is of older adults with known cardiovascular problems in nursing homes or being

cared for by a family caregiver at home. The mobility of these individuals is generally limited and having a non-invasive continuous cardiovascular monitoring system could be very beneficial. Research has been performed which shows older adults are interested in technology which can assist their daily living [62]. Furthermore, research has been done using Machine Learning to assist caregivers of older adults with dementia [44]. This suggests our system could assist the caregivers, potentially avoid taking the patient out of their comfortable environment, and prevent the need for medical professionals to directly intrude on their privacy.

8.2 Design of a Novel Machine Learning Algorithm with Real-Time Training Capabilities

We created the CSPELM algorithm to train a neural network for cardiovascular error predictions. The SLFN is proven to be a universal approximator and can fit our needs for cardiovascular state change detection. With the amount of data expected to be collected, backpropagation can be slow. Recall that the wearable ultrasound transducer vest is expected to collect approximately 90 data points per minute which equates to 5,400 data points per hour. For long term usage, such as perhaps the third use-case with older adults, there will be an abundance of data.

The speed at which the CSPELM trains is very impressive while providing results that outperform the other three algorithms. On average, the CSPELM has approximately double the training time of the ELM and SPELM but still significantly less than LM backpropagation and is still within the real-time constraint in many cases, which we require. As seen from Table 6.2, there are two specific cases that show this phenomenon best. That is, datasets E and H. The total training time of LM backpropagation is 57061.11 seconds and 12828.08 seconds for datasets E and H, respectively. Conversely, for the CSPELM algo-

rithm, the total training time was 28.37 seconds and 80.74 seconds for datasets E and H, respectively. In both cases, the CSPELM produced the highest performing model as well.

The primary potential use-case for the CSPELM training algorithm is when datasets contain a considerable amount of features. For example, in the healthcare setting, it is possible to obtain a large amount of data from various sources due to the continuous monitoring techniques used at the bedside. Keeping information in this realm is important as it may help with inferences at a later date, so dimensionality reduction [81] may be viewed as a negative here. So, if LM backpropagation takes 57061.11 seconds as in the case of dataset E, then the CSPELM is a competitive choice to consider. Especially, if there is a time constraint to produce the results such as a medical emergency. Moreover, it is possible to further reduce the training time of the CSPELM at the cost of potentially missing a perturbed optimum. As the training of the CSPELM is directly dictated by the user defined variables for number of training iterations, lowering these will also lower the training time.

As for the ε and ε_Mult variables, because these are user defined, it is possible to assign values which will cause the CSPELM to function improperly. That is, the CSPELM functions by utilizing a pre-trained local optimum SPELM. If one were to assign very large values for these two variables, you would effectively retrain the system as a SPELM. For example, assigning a ε constraint of 0.7, you are starting with a range that is almost the size of the original randomization sampling range. Therefore, we no longer preserve the constraining aspect.

Continuing with the CSPELM optimization techniques in the form of the CSPELMF and CSPELMR, they showed considerable improvements in the classification datasets. Though, in the regression datasets, both novel optimization techniques did not perform as expected. We consider that the lack of features in the regression datasets could cause this phenomenon. Examining Table 6.1 shows us the number of features in each regression data set being between 4-16 features. For classification datasets, the number of features are

between 8-301 and the mean number of features is considerably higher.

The use-cases for the optimization techniques are two-fold. The CSPELMF can be applied in situations in which the user encounters multiple starting weights that all train to the same local minimum. The forest optimization allows a large variance in starting positions in the n-dimensional gradient space which can assist in finding different minimums. As for the CSPELMR, this method is tailored to assist in escaping local minimums if traditional methods are ineffective.

Finally, we note that from our representative ELM literature review in Section 6.1.1, that every method outperforms the original ELM. Though, they do not appear to achieve as impressive of results as the CSPELM, CSPELMF, or CSPELMR. Furthermore, we are interested in [152] as it is the closest to our methodology with the constraint mentality. We believe it is possible to combine our CSPELM and their CELM to create a methodology which outperforms both existing ones. That is, applying our ε constraint to their method of selecting weights and biases of between class samples.

8.3 Predicting the Change in State of the Heart with Synthetic Heart Chamber Surface Area Data

Beginning with the training results of the one to five minute models, we can see from Table 7.7 that the CSPELM training tends to over generalize the data for more than one minute. The overfitting is interesting to see with the CSPELM as the algorithm showed no sign of overfitting in previous testing. We hypothesize that this phenomenon is likely caused by the target variables lack of significant variance. Recall that we use the mean bootstrap sampling method to generate the dataset. Thus, taking the length of the produced vector can possibly have a low variance. Although, from the one minute model, we see promising performance with respect to the models differentiability of heart states. That is, only 33.61% of healthy

data points were predicted outside the 95% confidence interval.

As for the abnormal datasets, we see from Table 7.8 that the one minute model performs well for the three types of heart conditions. Specifically, the results were 71.39%, 88.33%, and 60.28% of data correctly predicted outside of the 95% confidence interval for the AFb, PCMR, and PMI conditions, respectively. Although, the two to five minute models technically perform better than the one minute model. This is of course due to the over generalization of these models. Therefore, we continue to present the one minute model as the highest performing model.

Lastly, recall that we generated synthetic data for predicting heart state changes. We specifically set out to provide a theoretical foundation for an automated prediction system to work in conjunction with the wearable ultrasound transducer vest. We did not attempt to provide a foolproof classification system. Furthermore, as this is synthetic data, we find it inappropriate to design a system with explicit conditions to be used in practice. Instead, we show the plausibility of heart state differentiation using a SLFN trained by the CSPELM. From the one minute model, it is possible now to empirically determine a threshold of data points needed to be predicted outside the 95% confidence interval which would notify the medical professional as a "change in state" occurring. We hypothesize that the difference between an individual's data and an averaged group's data with respect to differentiability is considerable. That is, the sensitivity to abnormal data should improve if the model is only trained on one person's data as intended. Furthermore, only needing one minute of data is beneficial in that situations where a person is in a critical condition and changes can occur rapidly could help the medical professional make a quicker diagnosis.

9

Conclusion

In this chapter, we present the conclusion of our overall methodology. We began by providing a literature review of the state-of-the-art for medical imaging systems and 3D modeling techniques. From this review, we performed a subjective maturity rating to determine which methodologies we would choose from the two topics. The methodologies which were seen to be the most mature with respect to our research goals were Ultrasonography and Stereo Vision, respectively. Next, we created the wearable ultrasound transducer vest which is a system that will be capable of creating a near 3D model of the human heart in real-time. The system will utilize a stereo vision 3D modeling algorithm for the modeling. We present two possible transducer orientations, both of which account for the obstruction from the thoracic cage, that can be placed within the vest. The 4x2 orientation is computationally faster, overall less complex, produces a higher Frame Rate per Beat (FRPB), but with lower resolution in comparison to the 4x3 orientation. Conversely, the 4x3 orientation is capable of achieving a higher resolution at the cost of complexity and FRPB. Both orientations though have the Field of View (FOV) and stereo overlap to create a near 3D model of the heart. We also show the stereo overlap surface area coverage from all three possible views independently, the overlap between the parasternal and apical views, and the overlap between all three views. Furthermore, if we were to find smaller regions of interest

within the larger FOV, it is possible to split the FOV into smaller areas if needed. Finally, we provide generalized mathematical formulas to show the feasibility of such a system to work in real-time while not producing noise in the images from simultaneous transducer activations. That is, the systems allows for an approximately 250ms delay between each transducer activation. Then, to coincide with the wearable vest design, we provide a theoretical 3D modeling workflow specific for this system. While the preliminary data gathered using a 3D printed heart showed difficulties, we hypothesize that real Ultrasonography data will provide better results as the 3D printed heart was homogeneous in creation and color. The last feature gathering aspect we presented is from four Stochastic Petri Net (SPN) models. These include the blood flow for a healthy heart via the QRS complex cycle, a heart inflicted with Atrial Fibrillation (AFb), a healthy heart using the Wigger's pressure cycle, and of the systolic and diastolic states for each of the four chambers, respectively. These models can be used in the future for extracting functional features whereas the near 3D model handles structural features.

As for the individualized learning and prediction scheme, we created a state-of-the-art neural network training algorithm called the Constrained State Preserved Extreme Learning Machine (CSPELM). The CSPELM functions by using a baseline randomization of the hidden weights and biases of a neural network produced from a State Preserved Extreme Learning Machine (SPELM). Then, it perturbs the weights and biases using an ε constraint that is user defined to create a new sampling range rather than sampling from a uniform distribution. Furthermore, there is an additional ε_Mult variable that creates a homing aspect to the CSPELM which initializes a starting constraint range and shrinks over time. The results on the 13 publicly available datasets show that our CSPELM algorithm outperforms the standard Extreme Learning Machine (ELM), SPELM, and Levenberg-Marquardt (LM) backpropagation in 8 of the 13 datasets. Moreover, our CSPELM algorithm outperforms the ELM and SPELM algorithms in all 13 datasets. This algorithm has the capabilities of real-time training with very promising prediction capabilities. Therefore, it is

very suitable to be used as the algorithm to train the neural network for the individualized learning and prediction scheme. We also provided two prediction optimization techniques for the CSPELM in the form of a Forest optimization (CSPELMF) and ε Range-finder (CSPELMR). These two techniques disregard training time in an attempt to maximize prediction accuracy. The CSPELMF operates under the notion that the n-dimensional gradient space could have multiple minimums and that the SPELM only finds one. Therefore, a forest where each root is a trained SPELM could provide a better starting point for the CSPELM. Finally, for the CSPELMR, it operates by gradually increasing the size of the ε constraint in an attempt to escape a local minimum that the algorithm may be trapped in.

Finally, for the theoretical heart state prediction system, we first presented a method for synthetically generating multiple healthy and abnormal datasets of heart chamber surface areas. That is, for the healthy data, we generate datasets for one, two, three, four, and five minutes of 90 data points each. As for the abnormal data, we generate a dataset for Atrial Fibrillation (AFb), Primary Chronic Mitral Regurgitation (PCMR), and Post-Myocardial Infarction (PMI) conditions of 360 data points each. Finally, we generate one large healthy dataset of 360 data points as well for testing. A Single Layer Feedforward Network (SLFN) was trained using the CSPELM due to its real-time capabilities and high performance with default hyper-parameter settings. Our results show that the one minute model has high differentiability between healthy and abnormal data while using the 95% confidence interval as a threshold. Specifically, the system correctly predicted abnormal data of 71.39%, 88.33%, and 60.28% for AFb, PCMR, and PMI, respectively. This model acts as a theoretical foundation for adding automation to the wearable ultrasound transducer vest.

10

Future Work

The future work for this dissertation has both primary and secondary paths. The primary path is to create a prototype ultrasound vest. This prototype will allow us to conduct authentic experiments and gather real data to test our proposed autonomous prediction algorithm as well as the 3D modeling approach. More specifically, we hypothesize that real data will benefit the proposed 3D modeling workflow as there are more unique features with a moving human heart.

For the secondary paths, we first begin with measuring multiple FOV depths for each transducer to calculate the volume of the heart for both the systolic and diastolic stages. This will allow for more data to be gathered for the individualized learning and prediction scheme such as Blood Volume Pulse (BVP) and valve activation. Second, we acknowledge that this methodology functions on averages of humans. There is a noticeable anatomical difference between a 5'2" individual and a 6'5" individual. Thus, we will create a fuzzy inference system, optimized via a genetic algorithm [76], which can automatically produce a new arrangement (such as degrees to turn the transducers on the frontal and apical views or shifting them vertically) on certain individuals. Finally, we will examine the alterations necessary to dynamically change the depths of the FOV for each transducer to account for

the breathing cycles of an individual.

Bibliography

[1] Eduard Abrams, Marcus Mohr, Christine Engel, and Michael Bottlang. Cross-sectional geometry of human ribs. *cortex*, 80:2, 2003.

[2] Federal Aviation Administration. Medical certification, 2019.

[3] B Aldefeld. On automatic recognition of 3d structures from 2d representations. *Computer-Aided Design*, 15(2):59–64, 1983.

[4] Emina Alickovic and Abdulhamit Subasi. Medical decision support system for diagnosis of heart arrhythmia using dwt and random forests classifier. *Journal of medical systems*, 40(4):108, 2016.

[5] Md Zahangir Alom, Paheding Sidike, Vijayan K Asari, and Tarek M Taha. State preserving extreme learning machine for face recognition. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2015.

[6] Ethem Alpaydin and Kaynak Cenk. Optical recognition of handwritten digits data set, 1995.

[7] Dave Alsup. Pilot's deadly in-flight heart attack threatens to renew age debate, 2013.

[8] American Heart Association. Heart disease and stroke statistics-2019 update, 2019.

[9] Erlend Aune, Morten Baekkevar, Jo Roislien, Olaf Rodevand, and Jan Erik Otterstad. Normal reference ranges for left and right atrial volume indexes and ejection fractions obtained with real-time three-dimensional echocardiography. *European Journal of Echocardiography*, 10(6):738–744, 2009.

[10] Kim E. Barrett, Susan M. Barman, Scott Boitano, and Heddwen L. Brooks. *Ganong's review of medical physiology. 25*. McGraw-Hill Education, USA, 2015.

[11] Falko Bause and Pieter S Kritzinger. *Stochastic petri nets*, volume 1. Citeseer, 2002.

[12] BBC. American airlines pilot died mid-flight from 'heart attack', 2015.

[13] Jeremy Bercoff. Ultrafast ultrasound imaging. *Ultrasound imaging-Medical applications*, pages 3–24, 2011.

[14] J Gordon Betts, Peter Desaix, Eddie Johnson, Jody E Johnson, Oksana Korol, Dean Kruse, Brandon Poe, James Wise, Mark D Womble, and Kelly A Young. *Anatomy & Physiology*. OpenStax College, Rice University, Houstan, Texas, 2013.

[15] J Gordon Betts, Peter DeSaix, Eddie Johnson, Jody E Johnson, Oksana Korol, Dean H Kruse, Brandon Poe, James A Wise, Kelly A Young, et al. *Anatomy and physiology*. OpenStax, 2014.

[16] Selami Beyhan and Kadir Kavaklioglu. Comprehensive modeling of u-tube steam generators using extreme learning machines. *IEEE Transactions on Nuclear Science*, 62(5):2245–2254, 2015.

[17] Anne Bienert and Hans-Gerd Maas. Methods for the automatic geometric registration of terrestrial laser scanner point clouds in forest stands. *ISPRS Int. Arch. Photogramm. Rem. Sens. Spat. Inf. Sci*, pages 93–98, 2009.

[18] Alexander N Borg, James L Harrison, Rachel A Argyle, and Simon G Ray. Left ventricular torsion in primary chronic mitral regurgitation. *Heart*, 94(5):597–603,

2008.

[19] Carla Brodley. Image segmentation data set.

[20] Norbert Budincsevity. Weather in szeged 2006-2016, 2016.

[21] Dieu Tien Bui, Phuong-Thao Thi Ngo, Tien Dat Pham, Abolfazl Jaafari, Nguyen Quang Minh, Pham Viet Hoa, and Pijush Samui. A novel hybrid approach based on a swarm intelligence optimized extreme learning machine for flash flood susceptibility mapping. *Catena*, 179:184–196, 2019.

[22] Robert W Cootney. Ultrasound imaging: principles and applications in rodent research. *Ilar Journal*, 42(3):233–247, 2001.

[23] Francesca Cortellaro, Laura Ferrari, Francesco Molteni, Paolo Aseni, Marta Velati, Linda Guarnieri, Katia Barbara Cazzola, Silvia Colombo, and Daniele Coen. Accuracy of point of care ultrasound to identify the source of infection in septic patients: a prospective study. *Internal and emergency medicine*, 12(3):371–378, 2017.

[24] Jill Crosby, Dagmawi Werku, Tigist Zewdu, Grace Wanjiku, and Jessica Schmidt. Acute mesenteric ischaemia: A case of expedited diagnosis and management using point-of-care ultrasound. *African Journal of Emergency Medicine*, 8(4):164–166, 2018.

[25] Carlos Cuadrado, Aitzol Zuloaga, Jose L Martin, Jesus Laizaro, and Jaime Jimenez. Real-time stereo vision processing system in a fpga. In *IECON 2006-32nd Annual Conference on IEEE Industrial Electronics*, pages 3455–3460. IEEE, 2006.

[26] Boguslaw Cyganek and J Paul Siebert. *An introduction to 3D computer vision techniques and algorithms*. John Wiley & Sons, 2011.

[27] Ibrahim Danad, Zahi A Fayad, Martin J Willemink, and James K Min. New applications of cardiac computed tomography: dual-energy, spectral, and molecular ct

imaging. *JACC: Cardiovascular Imaging*, 8(6):710–723, 2015.

[28] Jean Dansereau and Ian AF Stokes. Measurements of the three-dimensional shape of the rib cage. *Journal of biomechanics*, 21(11):893–901, 1988.

[29] Howard B Demuth, Mark H Beale, Orlando De Jess, and Martin T Hagan. *Neural network design*. Martin Hagan, 2014.

[30] Catherine Dewhurst, Siobhan O'Neill, Kevin O'Regan, and Michael Maher. Demonstration of the course of the posterior intercostal artery on ct angiography: relevance to interventional radiology procedures in the chest. *Diagnostic and interventional radiology*, 18(2):221, 2012.

[31] Josef Ehling, Janka Bábíčková, Felix Gremse, Barbara M Klinkhammer, Sarah Baetke, Ruth Knuechel, Fabian Kiessling, Jürgen Floege, Twan Lammers, and Peter Boor. Quantitative micro-computed tomography imaging of vascular dysfunction in progressive kidney diseases. *Journal of the American Society of Nephrology*, 27(2):520–532, 2016.

[32] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[33] Myra K Feldman, Sanjeev Katyal, and Margaret S Blackwood. Us artifacts. *Radiographics*, 29(4):1179–1189, 2009.

[34] Guorui Feng, Guang-Bin Huang, Qingping Lin, and Robert Gay. Error minimized extreme learning machine with growth of hidden nodes and incremental learning. *IEEE Transactions on Neural Networks*, 20(8):1352–1357, 2009.

[35] Center for Disease Control and Prevention. National health and nutrition examination survey.

[36] Center for Disease Control and Prevention. Heart disease facts, 2019.

[37] Fernando A Freitas, Felope V Barbosa, and Sarajane M Peres. Grammatical facial expressions data set, 2014.

[38] Blake I Gardner, Scott E Bingham, Marvin R Allen, Duane D Blatter, and Jeffrey L Anderson. Cardiac magnetic resonance versus transthoracic echocardiography for the assessment of cardiac volumes and regional function after myocardial infarction: an intrasubject comparison using simultaneous intrasubject recordings. *Cardiovascular Ultrasound*, 7(1):38, 2009.

[39] Romolo Gaspari, Anthony Weekes, Srikar Adhikari, Vicki E Noble, Jason T Nomura, Daniel Theodoro, Michael Woo, Paul Atkinson, David Blehar, Samuel M Brown, et al. Emergency department point-of-care ultrasound in out-of-hospital and in-ed cardiac arrest. *Resuscitation*, 109:33–39, 2016.

[40] H Gemmeke and NV Ruiter. 3d ultrasound computer tomography for medical imaging. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 580(2):1057–1065, 2007.

[41] Shima Gholinezhadasnefestani, Omar Escalona, Kimia Nazarzadeh, Vivek Kodoth, Ernest Lau, and Ganesh Manoharan. Qrst cancellation in ecg signals during atrial fibrillation: Zero-padding versus time alignment. In *2012 6th International Conference on Signal Processing and Communication Systems*, pages 1–7. IEEE, 2012.

[42] Stephen N Gomperts, Joseph J Locascio, Sara J Makaretz, Aaron Schultz, Christina Caso, Neil Vasdev, Reisa Sperling, John H Growdon, Bradford C Dickerson, and Keith Johnson. Tau positron emission tomographic imaging in the lewy body diseases. *JAMA neurology*, 73(11):1334–1341, 2016.

[43] Mark J Gooding, Stephen Kennedy, and J Alison Noble. Volume segmentation and reconstruction from freehand three-dimensional ultrasound data with application to

ovarian follicle measurement. *Ultrasound in medicine & biology*, 34(2):183–195, 2008.

[44] Garrett Goodman, Tanvi Banerjee, William Romine, Cogan Shimizu, and Jennifer Hughes. Caregiver assessment using smart gaming technology: A feasibility study. In *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–6. IEEE, 2019.

[45] Garrett Goodman and Nikolaos Bourbakis. A wearable ultrasound methodology for creating a real time near 3d model of the heart. In *2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE)*, pages 539–546. IEEE, 2019.

[46] Garrett Goodman and Nikolaos Bourbakis. A comparative survey on state-of-the-art medical imaging methodologies for heart monitoring. In *Advances in Medicine and Biology*, volume 168, pages 183–204. New York: Nova Science, 2020.

[47] Garrett Goodman and Nikolaos Bourbakis. Predicting the change in state of the human heart based on synthetic heart chamber volume data. In *2020 IEEE 20th International Conference on Bioinformatics and Bioengineering (BIBE)*. IEEE, 2020.

[48] Garrett Goodman, Quinn Hirt, Cogan Shimizu, Iosif Papadakis Ktistakis, Miltiadis Alamaniotis, and Nikolaos Bourbakis. Methods for prediction optimization of the constrained state-preserved extreme learning machine. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2020.

[49] Garrett Goodman, Cogan Shimizu, and Iosif Papadakis Ktistakis. Constrained state-preserved extreme learning machine. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 752–759. IEEE, 2019.

[50] Stephen Gould, Richard Fulton, and Daphne Koller. Decomposing a scene into geometric and semantically consistent regions. In *2009 IEEE 12th international*

conference on computer vision, pages 1–8. IEEE, 2009.

[51] HENRY GRAY. Antomy of the human body. *Annals of Surgery*, 68(5):564–566, 1918.

[52] Paola Gripari, Manuela Muratori, Laura Fusini, Gloria Tamborini, and Mauro Pepi. Three-dimensional echocardiography: advancements in qualitative and quantitative analyses of mitral valve morphology in mitral valve prolapse. *Journal of cardiovascular echography*, 24(1):1, 2014.

[53] John E Hall. *Guyton and Hall textbook of medical physiology e-Book*. Elsevier Health Sciences, 2015.

[54] Hideyuki Hasegawa and Chris de Korte. Special issue on ultrafast ultrasound imaging and its applications, 2018.

[55] Carlos Hernandez, George Vogiatzis, and Roberto Cipolla. Multiview photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):548–554, 2008.

[56] Wei Shu Hou, Hua Wei Wu, Yan Yin, Jie Jun Cheng, Qing Zhang, and Jian Rong Xu. Differentiation of lung cancers from inflammatory masses with dual-energy spectral ct imaging. *Academic radiology*, 22(3):337–344, 2015.

[57] Guang-Bin Huang and Lei Chen. Convex incremental extreme learning machine. *Neurocomputing*, 70(16-18):3056–3062, 2007.

[58] Guang-Bin Huang and Lei Chen. Enhanced random search based incremental extreme learning machine. *Neurocomputing*, 71(16-18):3460–3468, 2008.

[59] Guang-Bin Huang, Lei Chen, Chee Kheong Siew, et al. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans. Neural Networks*, 17(4):879–892, 2006.

[60] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006.

[61] Qinghua Huang and Zhaozheng Zeng. A review on real-time 3d ultrasound imaging technology. *BioMed research international*, 2017, 2017.

[62] Jennifer C Hughes, Tanvi Banerjee, Garrett Goodman, and Larry Lawhorne. A preliminary qualitative analysis on the feasibility of using gaming technology in caregiver assessment. *Journal of Technology in Human Services*, 35(3):183–198, 2017.

[63] Somayeh Imani, Amay J Bandodkar, AM Vinu Mohan, Rajan Kumar, Shengfei Yu, Joseph Wang, and Patrick P Mercier. A wearable chemical–electrophysiological hybrid biosensing system for real-time health and fitness monitoring. *Nature communications*, 7(1):1–7, 2016.

[64] Michael F Insana, Robert F Wagner, Brian S Garra, David G Brown, and Thomas H Shawker. Analysis of ultrasound image texture via generalized rician statistics. *Optical Engineering*, 25(6):256743, 1986.

[65] Andras Janosi, William Steinbrunn, Matthias Pfisterer, and Robert Detrano. Heart disease data set, 1989.

[66] Jørgen A Jensen. Ultrasound imaging and its modeling. In *Imaging of Complex Media with Acoustic and Seismic Waves*, pages 135–166. Springer, 2002.

[67] Keith A Johnson, Aaron Schultz, Rebecca A Betensky, J Alex Becker, Jorge Sepulcre, Dorene Rentz, Elizabeth Mormino, Jasmeer Chhatwal, Rebecca Amariglio, Kate Papp, et al. Tau positron emission tomographic imaging in aging and early a lzheimer disease. *Annals of neurology*, 79(1):110–119, 2016.

[68] Ratheesh Kalarot and John Morris. Comparison of fpga and gpu implementations of real-time stereo vision. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pages 9–15. IEEE, 2010.

[69] Xin Kang, Mahdi Azizian, Emmanuel Wilson, Kyle Wu, Aaron D Martin, Timothy D Kane, Craig A Peters, Kevin Cleary, and Raj Shekhar. Stereoscopic augmented reality for laparoscopic surgery. *Surgical endoscopy*, 28(7):2227–2235, 2014.

[70] Vibhu Kapoor, Barry M McCook, and Frank S Torok. An introduction to pet-ct imaging. *Radiographics*, 24(2):523–543, 2004.

[71] Tanmay Kasbe and Ravi Singh Pippal. Design of heart disease diagnosis system using fuzzy logic. In *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, pages 3183–3187. IEEE, 2017.

[72] Arnold M Katz. *Physiology of the Heart*. Lippincott Williams & Wilkins, 2010.

[73] Young-Sun Kim, Min Jung Park, Hyunchul Rhim, Min Woo Lee, and Hyo Keun Lim. Sonographic analysis of the intercostal spaces for the application of high-intensity focused ultrasound therapy to the liver. *American Journal of Roentgenology*, 203(1):201–208, 2014.

[74] Bruce J Kimura. Point-of-care cardiac ultrasound techniques in the physical examination: better at the bedside. *Heart*, 103(13):987–994, 2017.

[75] Georgios Kordelas, JD Perez-Moneo Agapito, J Vegas Hernandez, and P Daras. State-of-the-art algorithms for complete 3d model reconstruction. *Proceedings of the Engage Summer School, Zermatt, Switzerland*, 1315:115, 2010.

[76] Iosif Papadakis Ktistakis, Garrett Goodman, and Cogan Shimizu. Methods for optimizing fuzzy inference systems. In *Advances in Data Science: Methodologies and Applications*, pages 97–116. Springer, 2020.

[77] Ajay Kumar, Mitchel T Williams, and Harry T Chugani. Evaluation of basal ganglia and thalamic inflammation in children with pediatric autoimmune neuropsychiatric disorders associated with streptococcal infection and tourette syndrome: a positron

emission tomographic (pet) study using 11c-[r]-pk11195. *Journal of child neurology*, 30(6):749–756, 2015.

[78] Susan H Kwon and Aasha S Gopal. 3d and 4d ultrasound: current progress and future perspectives. *Current Cardiovascular Imaging Reports*, 10(12):43, 2017.

[79] Malcolm E Legget, DF Leotta, EL Bolson, JA McDonald, RW Martin, Xiang-Ning Li, CM Otto, and FH Sheehan. System for quantitative three-dimensional echocardiography of the left ventricle based on a magnetic-field position and orientation sensing system. *IEEE transactions on Biomedical Engineering*, 45(4):494–504, 1998.

[80] Beyang Liu, Stephen Gould, and Daphne Koller. Single image depth estimation from predicted semantic labels. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1253–1260. IEEE, 2010.

[81] Huan Liu and Hiroshi Motoda. *Feature extraction, construction and selection: A data mining perspective*, volume 453. Springer Science & Business Media, 1998.

[82] Shi-Xia Liu, Shi-Min Hu, Chiew-Lan Tai, and Jia-Guang Sun. A matrix-based approach to reconstruction of 3d objects from three orthographic views. In *Proceedings the Eighth Pacific Conference on Computer Graphics and Applications*, pages 254–261. IEEE, 2000.

[83] Roberto Lopez, Thomas Brooks, Stuart Pope, and Michael Marcolini. Airfoil self-noise data set, 1989.

[84] Philipp Lurz, Christian Luecke, Ingo Eitel, Felix Föhrenbach, Clara Frank, Matthias Grothoff, Suzanne de Waha, Karl-Philipp Rommel, Julia Anna Lurz, Karin Klingel, et al. Comprehensive cardiac magnetic resonance imaging in patients with suspected myocarditis: the myoracer-trial. *Journal of the American College of Cardiology*, 67(15):1800–1811, 2016.

[85] Alicia M Maceira, Sanjay K Prasad, Mohammed Khan, and Dudley J Pennell. Reference right ventricular systolic and diastolic function normalized to age, gender and body surface area from steady-state free precession cardiovascular magnetic resonance. *European heart journal*, 27(23):2879–2888, 2006.

[86] AM Maceira, SK Prasad, M Khan, and DJ Pennell. Normalized left ventricular systolic and diastolic function by steady state free precession cardiovascular magnetic resonance. *Journal of Cardiovascular Magnetic Resonance*, 8(3):417–426, 2006.

[87] Plonsey Malmivuo, Jaakko Malmivuo, and Robert Plonsey. *Bioelectromagnetism: principles and applications of bioelectric and biomagnetic fields*. Oxford University Press, USA, 1995.

[88] Gunasekaran Manogaran, R Varatharajan, and MK Priyan. Hybrid recommendation system for heart disease diagnosis based on multiple kernel learning with adaptive neuro-fuzzy inference system. *Multimedia tools and applications*, 77(4):4379–4399, 2018.

[89] David Maresca, Mafalda Correia, Olivier Villemain, Alain Bizé, Lucien Sambin, Mickael Tanter, Bijan Ghaleh, and Mathieu Pernot. Noninvasive imaging of the coronary vasculature using ultrafast ultrasound. *JACC: Cardiovascular Imaging*, 11(6):798–808, 2018.

[90] Andrei Andreevich Markov. The theory of algorithms. *Trudy Matematicheskogo Instituta Imeni VA Steklova*, 42:3–375, 1954.

[91] Patrick Maynard. *Drawing distinctions: the varieties of graphic expression*. Cornell University Press, 2018.

[92] Donald W McRobbie, Elizabeth A Moore, Martin J Graves, and Martin R Prince. *MRI from Picture to Proton*. Cambridge university press, 2017.

[93] Yoan Miche, Antti Sorjamaa, Patrick Bas, Olli Simula, Christian Jutten, and Amaury Lendasse. Op-elm: optimally pruned extreme learning machine. *IEEE transactions on neural networks*, 21(1):158–162, 2009.

[94] Jamie R Mitchell and Jiun-Jr Wang. Expanding application of the wiggers diagram to teach cardiovascular physiology. *Advances in physiology education*, 38(2):170–175, 2014.

[95] Christopher L Moore and Joshua A Copel. Point-of-care ultrasonography. *New England Journal of Medicine*, 364(8):749–757, 2011.

[96] Shaine A Morris and Timothy C Slesnick. Magnetic resonance imaging. *Visual Guide to Neonatal Cardiology*, pages 104–108, 2018.

[97] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[98] Scott K Nagle, Michael Puderbach, Monika Eichinger, and Talissa A Altes. Magnetic resonance imaging of the lung: cystic fibrosis. In *MRI of the Lung*, pages 277–291. Springer, 2017.

[99] David Nanchen. Resting heart rate: what is normal? *Heart*, 104(13):1048–1049, 2018.

[100] University of Minnesota. Atlas of human cardiac anatomy.

[101] Department of Surgery. Atrial fibrillation.

[102] Shu Lih Oh, Eddie YK Ng, Ru San Tan, and U Rajendra Acharya. Automated diagnosis of arrhythmia using combination of cnn and lstm techniques with variable length heart beats. *Computers in biology and medicine*, 102:278–287, 2018.

[103] Yoshimichi Okada, Takeshi Koishi, Suguru Ushiki, Toshiya Nakaguchi, Norimichi Tsumura, and Yoichi Miyake. A fast stereo matching algorithm for 3d reconstruction of internal organs in laparoscopic surgery. In *Medical Imaging 2008: Visualization,*

Image-Guided Procedures, and Modeling, volume 6918, page 69182K. International Society for Optics and Photonics, 2008.

[104] World Health Organization. Cardiovascular diseases (cvds), 2017.

[105] Dae Yong Park, Daniel J Joe, Dong Hyun Kim, Hyewon Park, Jae Hyun Han, Chang Kyu Jeong, Hyelim Park, Jung Gyu Park, Boyoung Joung, and Keon Jae Lee. Self-powered real-time arterial pulse monitoring using ultrathin epidermal piezoelectric sensors. *Advanced Materials*, 29(37):1702308, 2017.

[106] Ji-Ae Park, Hee-Jeong Han, Jin-Chul Heo, and Jong-Ha Lee. Computer aided diagnosis sensor integrated outdoor shirts for real time heart disease monitoring. *Computer Assisted Surgery*, 22(sup1):176–185, 2017.

[107] James L Peterson et al. A note on colored petri nets. *Inf. Process. Lett.*, 11(1):40–43, 1980.

[108] Carl Adam Petri. Kommunikation mit automaten. 1962.

[109] Philippe Poncet, Sebastien Delorme, Janet L Ronsky, Jean Dansereau, GEORGE CLYNCH, JAMES HARDER, RICHARD D DEWAR, Hubert Labelle, PEI-HUA GU, and Ronald F Zernicke. Reconstruction of laser-scanned 3d torso topography and stereoradiographical spine and rib-cage geometry in scoliosis. *Computer methods in biomechanics and biomedical engineering*, 4(1):59–75, 2001.

[110] Cristhian Potes, Saman Parvaneh, Asif Rahman, and Bryan Conroy. Ensemble of feature-based and deep learning-based classifiers for detection of abnormal heart sounds. In *2016 Computing in Cardiology Conference (CinC)*, pages 621–624. IEEE, 2016.

[111] Richard Prager, Andrew Gee, Graham Treece, and Laurence Berman. Freehand 3d ultrasound without voxels: volume measurement and visualisation using the stradx system. *Ultrasonics*, 40(1-8):109–115, 2002.

[112] Mary M Quien and Muhamed Saric. Ultrasound imaging artifacts: How to recognize them and how to avoid them. *Echocardiography*, 35(9):1388–1401, 2018.

[113] Peter S Rahko. Evaluation of the skin-to-heart distance in the standing adult by two-dimensional echocardiography. *Journal of the American Society of Echocardiography*, 21(6):761–764, 2008.

[114] Kashif Rajpoot, J Alison Noble, Vicente Grau, Cezary Szmigielski, and Harald Becher. Multiview rt3d echocardiography image fusion. In *International Conference on Functional Imaging and Modeling of the Heart*, pages 134–143. Springer, 2009.

[115] Alfonso Santiago, Jazmín Aguado-Sierra, Miguel Zavala-Aké, Ruben Doste-Beltran, Samuel Gómez, Ruth Arís, Juan C Cajas, Eva Casoni, and Mariano Vázquez. Fully coupled fluid-electro-mechanical model of the human heart for supercomputers. *International journal for numerical methods in biomedical engineering*, 34(12):e3140, 2018.

[116] Ashutosh Saxena, Sung H Chung, and Andrew Y Ng. Learning depth from single monocular images. In *Advances in neural information processing systems*, pages 1161–1168, 2006.

[117] Stephen R Schach. *Object-Oriented and Classical Software Engineering*. McGraw Hills Publications, 2011.

[118] Sunando Sengupta, Eric Greveson, Ali Shahrokni, and Philip HS Torr. Urban 3d semantic modelling using stereo vision. In *2013 IEEE International Conference on robotics and Automation*, pages 580–585. IEEE, 2013.

[119] Maxime Sermesant, Yves Coudière, Herve Delingette, Nicholas Ayache, and Jean-Antoine Désidéri. An electro-mechanical model of the heart for cardiac image anal-

ysis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 224–231. Springer, 2001.

[120] Denis Serre. *Matrices Theory and Applications*, volume 2. Springer, 2010.

[121] Byeong-Seok Shin and Yeong Gil Shin. Fast 3d solid model reconstruction from orthographic views. *Computer-Aided Design*, 30(1):63–76, 1998.

[122] Vince Sigillito. Ionosphere data set, 1989.

[123] Jack W Smith, JE Everhart, WC Dickson, WC Knowler, and RS Johannes. Pima indians diabetes database, 1988.

[124] P Soler, O Gerard, P Allain, E Saloux, E Angelini, and I Bloch. Comparison of fusion techniques for 3d+ t echocardiography acquisitions from different acoustic windows. In *Computers in Cardiology, 2005*, pages 141–144. IEEE, 2005.

[125] Mingzhou Song, Robert M Haralick, Florence H Sheehan, and Richard K Johnson. Integrated surface model optimization for freehand three-dimensional echocardiography. *IEEE transactions on medical imaging*, 21(9):1077–1090, 2002.

[126] Ashwin Srinivasan. Statlog (landsat satellite) data set.

[127] Ioannis Stamos, Lingyun Liu, Chao Chen, George Wolberg, Gene Yu, and Siavash Zokai. Integrating automated range registration with multiview geometry for the photorealistic modeling of large-scale scenes. *International Journal of Computer Vision*, 78(2-3):237–260, 2008.

[128] Sándor M Szilágyi, László Szilágyi, and Zoltán Benyó. A patient specific electromechanical model of the heart. *Computer methods and programs in biomedicine*, 101(2):183–200, 2011.

[129] Jiexiong Tang, Chenwei Deng, and Guang-Bin Huang. Extreme learning machine for multilayer perceptron. *IEEE transactions on neural networks and learning sys-*

tems, 27(4):809–821, 2015.

[130] Wilco Tanis, Arco J Teske, Lex A Van Herwerden, Steven Chamuleau, Folkert Meijboom, Ricardo PJ Budde, and Maarten-Jan Cramer. The additional value of three-dimensional transesophageal echocardiography in complex aortic prosthetic heart valve endocarditis. *Echocardiography*, 32(1):114–125, 2015.

[131] R Andrew Taylor and Christopher L Moore. Point-of-care ultrasonography of the thoracic aorta. *Emergency Point-of-Care Ultrasound*, pages 32–38, 2017.

[132] Beau Tippetts, Dah Jye Lee, Kirt Lillywhite, and James Archibald. Review of stereo vision algorithms and their suitability for resource-limited systems. *Journal of Real-Time Image Processing*, 11(1):5–25, 2016.

[133] Michail Tsakalakis and Nikolaos Bourbakis. A wearable ultrasound multi-transducer array system for abdominal organs monitoring. In *13th IEEE International Conference on BioInformatics and BioEngineering*, pages 1–5. IEEE, 2013.

[134] Athanasios Tsanas, Max A Little, Patrick E McSharry, and Lorraine O Ramig. Accurate telemonitoring of parkinson's disease progression by noninvasive speech tests. *IEEE transactions on Biomedical Engineering*, 57(4):884–893, 2009.

[135] Pınar Tüfekci. Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. *International Journal of Electrical Power & Energy Systems*, 60:126–140, 2014.

[136] Maarten Vergauwen, Marc Pollefeys, and Luc Van Gool. A stereo-vision system for support of planetary surface exploration. *Machine Vision and Applications*, 14(1):5–14, 2003.

[137] Jiacun Wang. *Timed Petri nets: Theory and application*, volume 9. Springer Science & Business Media, 2012.

[138] SA Waugh, CA Purdie, LB Jordan, S Vinnicombe, RA Lerski, P Martin, and AM Thompson. Magnetic resonance imaging texture analysis classification of primary breast cancer. *European radiology*, 26(2):322–330, 2016.

[139] Qian Weng, Zhengyuan Mao, Jiawen Lin, and Xiangwen Liao. Land-use scene classification based on a cnn using a constrained extreme learning machine. *International journal of remote sensing*, 39(19):6281–6299, 2018.

[140] Wikipedia. Wiggers diagram. https://en.wikipedia.org/wiki/Wiggers_diagram.

[141] William H Wolberg, Nick W Street, and Olvi L Mangasarian. Breast cancer wisconsin (diagnostic) data set, 1993.

[142] Changchang Wu. Visualsfm: A visual structure from motion system. http://ccwu.me/vsfm/.

[143] Qingxiong Yang, Liang Wang, and Narendra Ahuja. A constant-space belief propagation algorithm for stereo matching. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1458–1465. IEEE, 2010.

[144] Zaher Mundher Yaseen, Sadeq Oleiwi Sulaiman, Ravinesh C Deo, and Kwok-Wing Chau. An enhanced extreme learning machine model for river flow forecasting: State-of-the-art, practical applications in water resource engineering area and future research direction. *Journal of Hydrology*, 569:387–408, 2019.

[145] Konstantin Yastrebov, Laurencie Brunel, Hugh S Paterson, Zoe A Williams, and Paul G Bannon. Three-dimensional intracardiac echocardiography and pulmonary embolism. *Cardiovascular Ultrasound*, 18(1):1–4, 2020.

[146] I-C Yeh. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research*, 28(12):1797–1808, 1998.

[147] Yun-Chi Yeh and Wen-June Wang. Qrs complexes detection for ecg signal: The difference operation method. *Computer methods and programs in biomedicine*, 91(3):245–254, 2008.

[148] Honggang Yu, Marios S Pattichis, Carla Agurto, and M Beth Goens. A 3d freehand ultrasound system for multi-view reconstructions from sparse 2d scanning planes. *Biomedical engineering online*, 10(1):7, 2011.

[149] Dan Zhang, Xiangang Peng, Keda Pan, and Yi Liu. A novel wind speed forecasting based on hybrid decomposition and online sequential outlier robust extreme learning machine. *Energy conversion and management*, 180:338–357, 2019.

[150] Qian Zhang, Ju-fang Wang, Qing-qing Dong, Qing Yan, Xiang-hong Luo, Xue-ying Wu, Jian Liu, and Ya-ping Sun. Evaluation of left atrial volume and function using single-beat real-time three-dimensional echocardiography in atrial fibrillation patients. *BMC medical imaging*, 17(1):44, 2017.

[151] Qin-Yu Zhu, A Kai Qin, Ponnuthurai N Suganthan, and Guang-Bin Huang. Evolutionary extreme learning machine. *Pattern recognition*, 38(10):1759–1763, 2005.

[152] Wentao Zhu, Jun Miao, and Laiyun Qing. Constrained extreme learning machine: a novel highly discriminative random feedforward neural network. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 800–807. IEEE, 2014.

[153] Daniel Zwillinger. *CRC standard mathematical tables and formulae*. Chapman and Hall/CRC, 2002.