

***RASTER*: Robot Assistant for Sonographic Tissue Examination and Reconstruction**

Benjamin Mart



A Thesis submitted to the Faculty of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Masters of Science in
Robotics Engineering
April 2024

APPROVED

Professor Haichong Zhang, *Advisor*, Worcester Polytechnic Institute

Professor Vincent Aloï, *Committee Member*, Worcester Polytechnic Institute

Professor Yihao Zheng, *Committee Member*, Worcester Polytechnic Institute

1 Abstract

While the exact prevalence of thyroid nodules in the wider population is not known, studies have shown them to be prevalent in between 2% and 65% of the the human population. The vast majority of these tumors are benign. However, the rate of thyroid cancer has at least doubled since 1980 while a corresponding increase in mortality has not occurred, leading researchers to conclude the rise is a measure of how many cancerous nodules are being found. Physicians have turned to ultrasonography as a relatively low-cost method of imaging thyroid nodules in a non-invasive way. These exams, when conducted by physicians and trained ultrasound technicians, are prone to human error which can cause inconsistent results between exams. This thesis proposes a novel robotic system for assisting physicians in conducting thyroid ultrasound exams. The proposed system uses a seven degree of freedom robotic manipulator mounted with a wireless ultrasound scanner to capture ultrasound images at known locations and a two-stage image segmentation framework to track the region of interest identified within the ultrasound image. The first stage is a real-time, non-machine learning image segmentation algorithm, initialized by the physician, that tracks the region of interest throughout the examination assisted by a force, scanner orientation, and position control framework. The second stage of the framework is a non-real time, non-machine learning image segmentation algorithm which, combined with the positional data from the robot, generates an approximation of the region of interest as a volume and displays a 3D representation of this volume. Testing completed on a training phantom demonstrated the ability of the system to track the patient profile accurately and in real-time. From the experimental data, a volume was repeatedly generated and visualized properly.

2 Acknowledgements

I would like to sincerely thank my advisor, Professor Zhang, for his guidance and encouragement throughout my thesis process as well as further fueling my interest in medical robotics. I would also like to thank Xihan Ma for his support and help with any and all technical questions I had.

Contents

1	Abstract	i
2	Acknowledgements	ii
	List of Figures	v
	List of Tables	vii
3	The Thyroid Gland	1
3.1	Nodules in the Thyroid	1
3.2	Motivation for Completing Thyroid Ultrasound Examinations	2
3.3	The Thyroid Examination Procedure and Calculating Thyroid Volumes	3
3.4	Shortcomings of the Current Ultrasound Examination Procedure	5
4	Previous Work on Robotizing Ultrasound Examinations	6
4.1	Previous Robotic Ultrasound Systems	6
4.2	Common Components of Robotic Ultrasound Systems	8
4.3	Ultrasound Image Segmentation Techniques	10
4.4	Further Explanation of the GrabCut Method	11
5	Introduction to the <i>RASTER</i> System	14
6	Materials and Methods	15
6.1	Integrating the Ultrasound Scanner	17
6.2	Mounting the Ultrasound Scanner	17
6.3	Selecting the Robotic Manipulator	19
6.4	Designing the User Interface	21
6.5	Developing a Real-time Image Segmentation Framework	24

6.6	Creating an Image Centering Algorithm	29
6.7	Creating an Image Balancing Algorithm	29
6.8	Implementing a Trajectory Generation Algorithm	34
6.9	Implementing a Force control Algorithm	36
6.10	Registering Corresponding Image and Position Data	36
6.11	Developing a Non-real Time Image Segmentation Framework	37
6.12	Estimating the Volume of the Region of Interest	39
6.13	Controlling the Robotic Manipulator	40
7	Testing and Validation	41
7.1	Evaluation of the Force Control Framework	42
7.2	Evaluation of Trajectory Generation Framework	44
7.3	Evaluation of Image Balancing Framework	45
7.4	Evaluation of Image Centering Framework	47
7.5	Accuracy of the Non-real Time Image Segmentation Framework	49
7.6	Repeatability of the Volume Reconstruction Framework	51
8	Discussion	53
9	Conclusion	55
	References	57

List of Figures

1	The anatomy of the thyroid gland [1].	2
2	An ultrasound image of a thyroid with a nodule.	4
3	The generalized workflow of RUSS [2].	9
4	The complete <i>RASTER</i> system.	15
5	A system diagram of the <i>RASTER</i> system.	16
6	A hardware diagram of the <i>RASTER</i> system.	16
7	The <i>Clarius</i> L7 Linear scanner (a) and the <i>Clarius</i> app (b).	17
8	The hardware for mounting the ultrasound scanner to the robot.	19
9	The <i>Franka</i> robotic manipulator used by <i>RASTER</i>	20
10	The landing page of the <i>Franka Desk</i> application.	21
11	The 'Exam Setup' tab of <i>RASTER</i> 's user interface.	23
12	The 'Thyroid Exam' tab of <i>RASTER</i> 's user interface.	24
13	A step-by-step illustration of the image segmentation process.	28
14	The possible locations of the image centering goal (white line) and error bands (green area).	29
15	A comparison between a high-quality ultrasound image (a) and a low-quality ultrasound image (b).	30
16	A step-by-step illustration of the creation of sectors from a raw ultrasound image.	31
17	An example of the image balancing approximation line generated from bound- ary points.	33
18	The definition of the axes of <i>RASTER</i> 's ultrasound scanner.	34
19	A theoretical plane-based trajectory with d millimeters between each image.	35
20	The 'Experimentation' tab of <i>RASTER</i> 's user interface.	42
21	The measured force applied to the patient with and without active force control.	43

22	The measured pose of the scanner with respect to the origin of the robot with and without active force control.	44
23	The measured cartesian position of the ultrasound scanner relative to the robot origin over the length of a trajectory where each black diamond represents when a waypoint was reached.	45
24	The measured slope of the approximation line with and without image balancing.	46
25	The measured orientation of the ultrasound scanner about the scanner's X axis with and without image balancing.	46
26	Ultrasound images captured at the start (a) and end (b) of the trajectory when image balancing was not active and at the start (c) and end (d) of the trajectory when image balancing was active.	47
27	The measured distance of the centroid of the region of interest from the center of the image with and without image centering active.	48
28	Ultrasound images captured at the start (a) and end (b) of the trajectory when image centering was not active and at the start (c) and end (d) of the trajectory when image centering was active.	49
29	The calculated accuracy of the non-real-time segmentation throughout five trajectories.	50
30	Comparison between ground truth segmentation (left) and automatic segmentation (right) early in the trajectory.	50
31	Comparison between ground truth segmentation (left) and automatic segmentation (right) in the middle of the trajectory.	51
32	Comparison between ground truth segmentation (left) and automatic segmentation (right) late in the trajectory.	51
33	The visualization of a generated 3D volume.	52

List of Tables

1	List of control schemes and the axes of motion which they control.	41
2	The result of the volume estimation module over seven trials.	52

3 The Thyroid Gland

The thyroid is a gland of the endocrine system in the neck straddling the trachea. Figure 1 illustrates how the two distinct lobes of the thyroid are connected by an isthmus of tissue creating a butterfly shape [3]. Hormones produced by the thyroid regulate the function of several major organ systems, such as the central nervous system, the cardiac system, and the skeletal system [4]. The exact size of the thyroid varies significantly based on the age, sex, weight, and iodine intake of any given person. However, it was found that in healthy adults without an iodine deficiency, the gland is approximately 4.4 x 1.4 x 1.2 centimeters in size, or between 7 and 10 milliliters in volume when measured by ultrasound [5].

3.1 Nodules in the Thyroid

A thyroid nodule is a clump of abnormal tissue found within the thyroid. While the exact prevalence of thyroid nodules in the wider population is not known, studies have shown them to be prevalent in between 2% and 65% of the human population [6]. Notably, women are found to have nodules at a rate four times greater than men. The leading factors affecting the incidence of thyroid nodules are sufficient iodine intake [7], followed by age and body mass index [8].

While the general incidence rate for thyroid nodules is high, the vast majority of these tumors are benign [9]. The rate of thyroid cancer has at least doubled since 1980 but this has not corresponded to an increase in mortality, leading researchers to conclude the rise is a measure of how many cancerous nodules are being found [10]. As the use of medical imaging technology has increased, so has the incidence of thyroid nodules found during radiologic examinations conducted for other reasons [11].

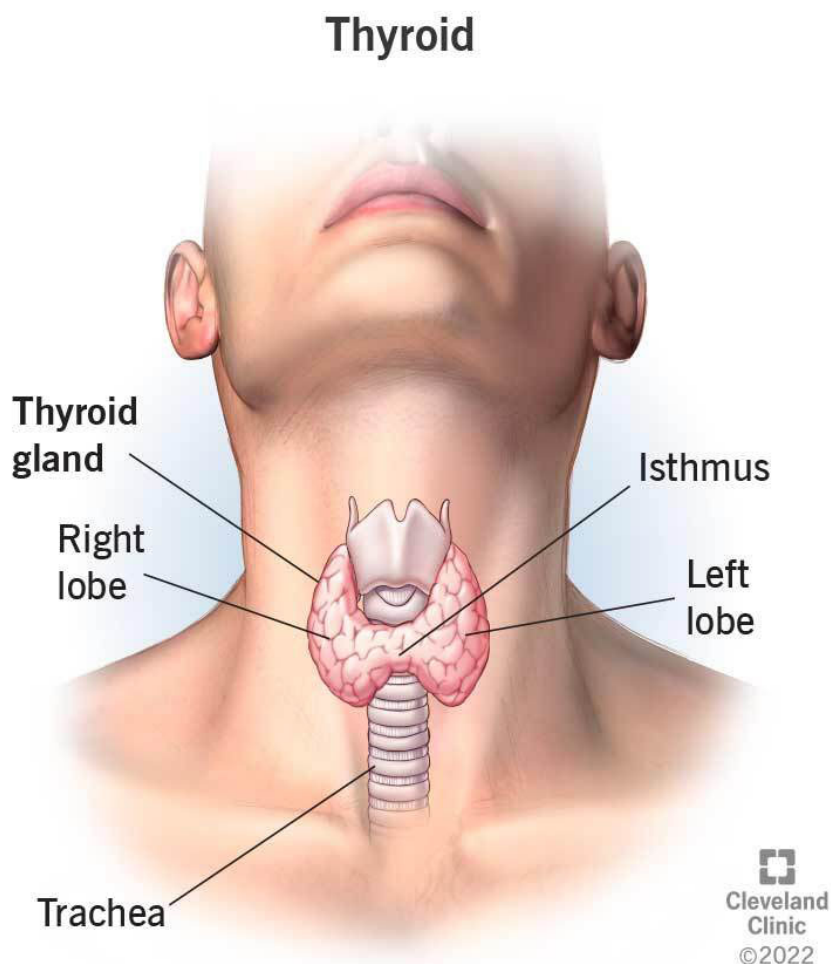


Figure 1: The anatomy of the thyroid gland [1].

3.2 Motivation for Completing Thyroid Ultrasound Examinations

Physicians have turned to ultrasonography as a non-invasive, relatively low-cost method of imaging the thyroid gland and any nodules found within it. Ultrasonographic examinations are more accurate than traditional physical examinations and allow the detection of smaller nodules [12], [6]. These examinations, in conjunction with measurements of the patient's thyroid stimulating hormone, are the first diagnostic step when a thyroid nodule has been found [8]. According to the American Thyroid Association, ultrasound examinations should gather quantitative and qualitative data about the gland as well as information

about any nodules found within the gland. For the gland, the examination should determine the homogeneity of the functional thyroid tissue and the overall size of the gland. For each nodule, the size in three dimensions, overall shape, sonographic characteristics such as echogenicity, composition, and vascularity, and the nodules position within the thyroid should be captured. Based on the size of each nodule and its sonographic characteristics, physicians can categorize the risk of malignancy and determine if a fine needle aspiration biopsy should be completed [13].

3.3 The Thyroid Examination Procedure and Calculating Thyroid Volumes

In addition to nodules that were discovered incidentally by other imaging methods, patients receiving an ultrasound examination of the thyroid will have been identified as at risk for having a thyroid condition either due to a previously recognized thyroid issue, a new symptom plausibly related to a thyroid issue, or a standard screening exam [9].

Regardless of the reason for the examination, the steps in the procedure remain consistent. Patients lay down in a supine position or sitting upright in a chair with their head tilted back. The key for either position is that the head is tilted back so that the neck is fully extended and exposed [14]. Ultrasound gel is applied to the neck to improve the quality of the image and the ultrasound scanner is then placed in the center of the neck to locate the thyroid gland. Once each lobe of the gland is visible, the scanner is moved to each lobe to measure their respective diameter and overall length. The scanner is moved along the length of each thyroid lobe to find any nodules that may exist [12]. Figure 2 demonstrates how a nodule may appear within a thyroid gland using a thyroid phantom. From the measurements taken, the volume of the lobe can be calculated by assuming its shape as a rotational ellipsoid and then using Equation 1 to estimate the volume [15].

$$V = \frac{\pi}{6} * height * width * length \approx 0.524 * height * width * length \quad (1)$$

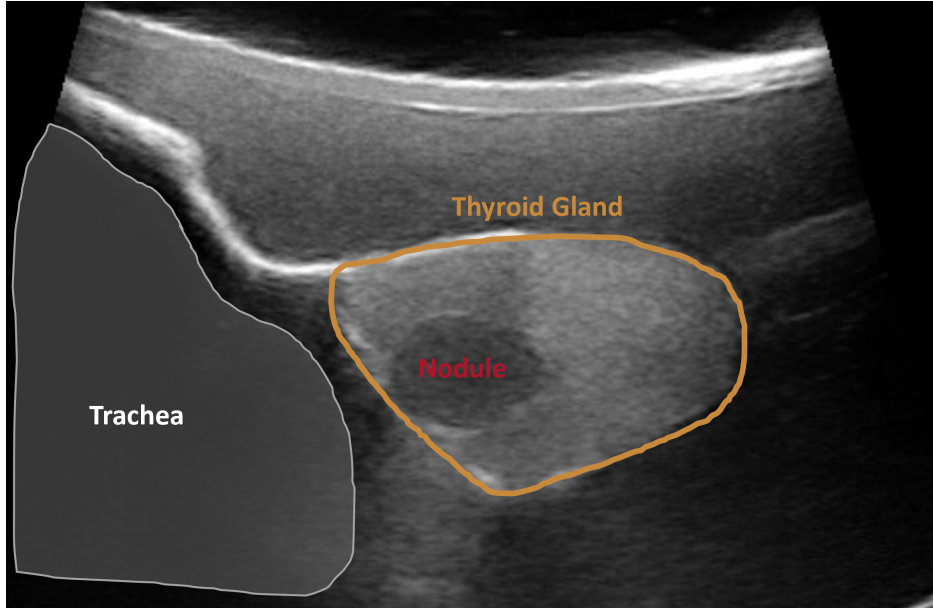


Figure 2: An ultrasound image of a thyroid with a nodule.

However, through experimentation conducted on cadavers where the volume of the gland was measured with both ultrasound imaging and water displacement, Equation 2 was developed as the accepted method of calculating volume [16].

$$V = 0.479 * height * width * length \quad (2)$$

Researchers have conducted further work using CT imaging techniques to determine if a more accurate correction factor should be used resulting in the proposal of a new correction factor of 0.529 [17].

Depending on the results of the patient's ultrasound examination or biopsy, their physician may recommend that they return for another ultrasound examination between 6 and 24 months later [8]. At each subsequent examination, the size of the gland and any individual nodules will be compared with previous examinations. Significant changes in size of the nodules are an indication that the further testing should be completed [13]

3.4 Shortcomings of the Current Ultrasound Examination Procedure

While ultrasound examinations have been shown to be effective at screening for nodules in the thyroid gland, the current procedure has several areas where it could be improved. Foremost, the procedure requires a trained sonographer or physician to conduct the examination limiting its availability from the start [18]. One study found that in the United States, while 88.56% of metropolitan counties had access to point-of-care ultrasound, only 38.84% of rural counties had access to point-of-care ultrasound. Furthermore, while ultrasounds are not harmful to the patient, sonographers who conduct these examinations are likely to develop long-term musculoskeletal issues from their work [19].

Since a key reason for conducting these examinations is monitoring the thyroid and any nodules found within it over time, it is critical that the results of any singular examination are consistent with the results from any other examination. However, the examination results depend on the skill and repeatability of the sonographer, two studies that examined the growth of nodules overtime found a variation of 14.6% and 11.7% respectively between the sonographers [20], [21]. This known issue has directly caused the American Thyroid Association to select a higher volume change cutoff to ensure statistically accurate results [13].

4 Previous Work on Robotizing Ultrasound Examinations

Researchers have investigated how to apply robotics to ultrasound procedures, both of the thyroid and other areas of the body, to improve the results of the examinations while reducing the requirements for conducting them.

4.1 Previous Robotic Ultrasound Systems

In recent years, significant research has been conducted on robotic ultrasound systems (RUSS) and has led to the development of many systems utilizing a variety of imaging systems, hardware designs, and software implementations to reach various levels of autonomy. Literature reviews of previous work on this topic reveal that research has shifted focus from developing systems designed to be teleoperated towards semi- and fully autonomous RUSS [22], [2], [23].

Teleoperated RUSS has become a mature research field as multiple studies have been conducted in clinical settings over large distances. One study conducted with a tele-operated RUSS, operated from fifty kilometers away, was conducted on 300 patients within one year on various anatomical structures, including the thyroid. This study found that the information gathered from the tele-sonographic exam provided equivalent information to that of a traditional exam while reducing costs and reducing patient wait times [24]. A second study, conducted on 22 COVID-19 patients, demonstrated that a tele-operated RUSS can provide adequate diagnostic information with no statistical difference from data captured in traditional ultrasound methods [25]. Both studies demonstrate that robotizing ultrasound examinations could increase their availability by removing the need for trained personnel to be in the same location as the patient while providing equivalent levels of care. Additionally, these systems reduce the physical strain placed on the sonographer or physician during

the examinations, thereby reducing their likelihood of developing long-term musculoskeletal injuries.

As research moved towards developing more autonomous solutions, multiple systems have been proposed that add levels of autonomy to tele-operated systems to increase their accuracy and effectiveness. One such system proposed allowing the operator and robot to co-control all degrees of freedom of the ultrasound scanner while automatically ensuring that the organ of interest remains in the ultrasound image, regardless of the other scanner motions [26]. Another proposed system used four laser sensors to compute the normal vector for the skin surface around the scanner and automatically orient the scanner as the operator moved it around the patient without human input [27]. Semi-autonomous systems like these help physicians and sonographers capture consistent results and lower the skill threshold required to conduct ultrasound examinations.

Research towards creating fully autonomous solutions has explored how to remove human interaction from every step in the exam process. Hennersperger et al. proposed a system that utilizes pre-operative MRI imaging to create trajectories for the robot to follow and then a structured-light 3D scanner to register the location of the patient relative to the robot so that the pre-planned trajectories could be followed autonomously [28]. However, requiring the patient to undergo MRI imaging prior to completing an ultrasound examination nullified the positive aspects of ultrasound imaging systems. A similar proposed system instead used a depth camera to estimate the surface of the patient's skin and then develop trajectories from this data [29]. A similar proposed system used multiple structured-light scanners in a large-scale multi-robot scanner to automatically conduct breast ultrasound examinations [30]. While effective, the size, cost, and complexity of this system mean that it cannot be used in many ultrasound examination scenarios.

4.2 Common Components of Robotic Ultrasound Systems

From the wide field of research available on RUSS, a framework of common components required for a semi- or fully autonomous RUSS can be determined. These components can be split between hardware and software components. A further division can be made by examining the necessary operations that must be completed during the pre-operative, intra-operative, and post-operative steps.

From a hardware perspective, a semi-autonomous system includes at minimum a robotic manipulator, a method of measuring the forces applied by the robot, such as a multi-axis force sensor, an ultrasound scanner for recording data, and some form of user interface for the operator to provide control input to the robot [22]. In some proposed systems, the method of measuring force is replaced with a mechanical mechanism that can apply a constant force [31][32]. For systems where full autonomy is desired, a camera system is added to provide information on the location of the patient relative to the robot [22].

From a software perspective, a semi-autonomous system includes a framework for controlling the forces the robot applies to the patient and can include a method of properly orienting the ultrasound scanner. Force control is critical to ensure the safety of the patient. Force control prevents the robot from hurting the patient as the robot moves around the patient's body or when the patient moves unexpectedly relative to the robot. However, the force control framework may not be included if force control is completed through a purely mechanical mechanism that can account for these motions. Schemes for controlling the scanner orientation within the imaging plane have been previously proposed using ultrasound images as the source of error for the orientation. Further scanner orientation has been proposed for the out-of-plane direction as well as skin plane estimation to completely control the scanner orientation. For fully autonomous systems, an additional control framework must be included for position control. This includes both generating a trajectory for the robot to traverse that properly moves across the patient as well as a control scheme that minimizes

the error between the actual robot position and the desired trajectory. Notably, several control schemes have been proposed to address the conflicting requirements that force, scanner orientation, and pose control create using impedance-based controllers [25].

The differences between tele-operative versus semi-autonomous versus fully autonomous RUSS can be further characterized by reviewing the workflow of an examination throughout the pre-operative, intra-operative, and post-operative stages. Figure 3 summarizes the workflow used by previously proposed systems of varying levels of autonomy. A fully autonomous system must be able to complete the full workflow autonomously whereas semi-autonomous or tele-operative systems can rely on human input to accomplish some number of these steps [2].

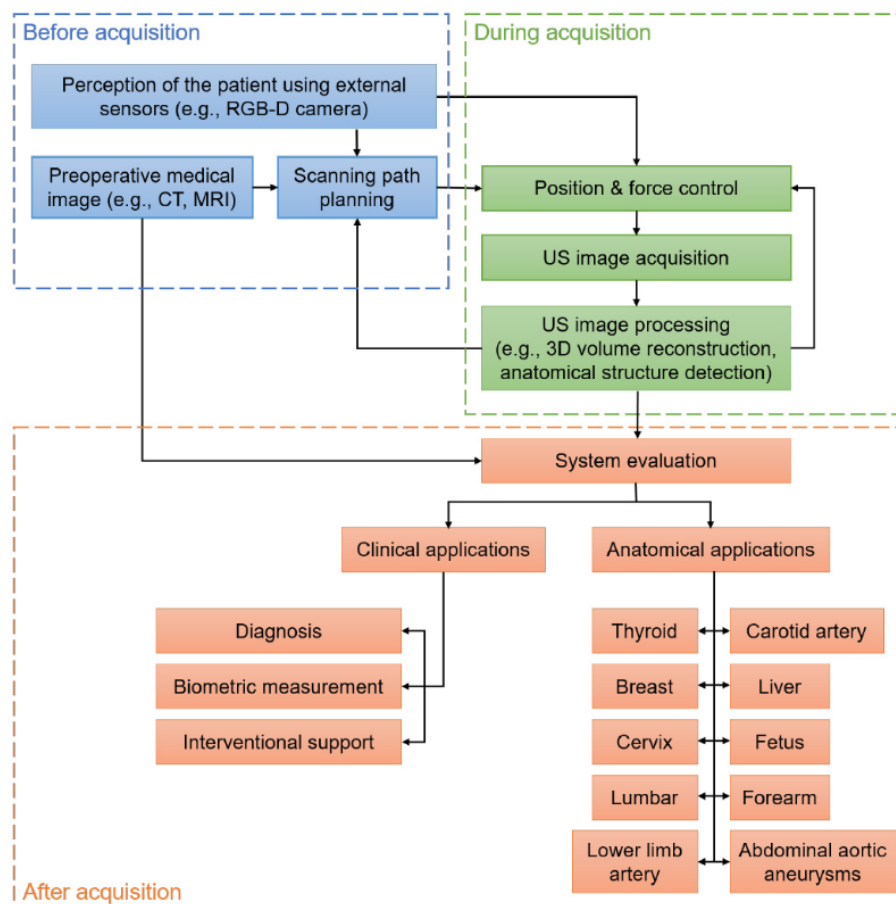


Figure 3: The generalized workflow of RUSS [2].

4.3 Ultrasound Image Segmentation Techniques

Every semi- or fully autonomous RUSS must have a method to identify the region of interest, such as the thyroid gland or nodules within it, within the ultrasound image. Chen et al. categorizes several algorithms implemented to identify the thyroid, and nodules within it, into three broad categories: contour and shape-based methods, region-based methods, and machine and deep learning methods [33].

Contour and shape-based segmentation methods, such as active contour without edges (ACWE), localized region based active contour (LRAC), distance regularized level set evolution (DRLSE), and geodesic active contours (GAC), are categorized as using geometric information and visually distinct edges to isolate areas of the image [33]. The ACWE model is an evolution of the existing active contour model improved through the inclusion of Mumford-Shah segmentation techniques and a level set formulation [34]. The GAC model, which was also based on the active contour model, uses minimal distance curves, geodesics, in a Riemannian space developed from the image data to find boundaries in the image [35]. The LRAC model, derived from the active contour model, redefines the segmentation problem as an issue of minimizing many local region energy models rather than a singular global image models. This model can then be implemented in conjunction with other segmentation models to improve their performance [36]. The DRLSE model proposes a level set formulation with two additional terms, one that maintains the desired shape of the level set and a second that pushes the contour towards the desired boundaries, improves the computational efficiency and reduces the accuracy requirements for the initial level set over previous level set models [37]. Notably, each of these models require some form of initialization to identify the region of interest and are subject to errors caused by poor initialization to varying degrees.

Region based segmentation methods, such as GrabCut (GC) and similar reflective region (SRR), use statistical analysis of the image to differentiate between regions [33]. The GC method expands upon the graph cut method, a graph-based method that maps pixels

together based on similarity, to reduce the amount of user input required and improve the behavior of the segmentation around the border of the area [38]. While the GC method was created as a general method for segmenting a broad range of images, the SSR method was made specifically for segmenting ultrasound images. The method relies on the differences in echogenicity (the brightness of tissue in an ultrasound image due to sound waves) between different tissue types found around the thyroid gland. A key limitation of this model is that it assumes the ultrasound images were captured in one of two orientations and therefore contain certain anatomical features in each image. With this assumption, the segmentation method then matches the number and approximate locations of these similar reflective regions with patterns created from the two known orientations [39].

Machine learning segmentation methods rely on machine and deep learning algorithms to train classifiers that can properly label each pixel in the ultrasound images. While generally these machine learning approaches are proficient at segmenting the thyroid gland and nodules found within, the large amount of labelled data and long periods of time required to train these algorithms limit their usefulness [33]. One proposed system for identifying the thyroid gland in ultrasound images used the 3D U-net architecture to classify each pixel in an ultrasound image as a member of the thyroid gland. Their training data consisted of 10 data sets comprising 1416 images and the generated segmentations had an average DICE score of 0.876 [40]. Another proposed system framed the problem of identifying nodules within the thyroid gland as a patch classification problem and then used a deep convolution neural network to identify the nodules. The proposed system was trained on 22,123 images collected from 6,242 patients and their model generated segmentations with an average DICE score of 0.922 [41].

4.4 Further Explanation of the GrabCut Method

In the paper “*GrabCut*” — *Interactive Foreground Extraction using Iterated Graph Cuts*, Rother et al. designed a graph-based image segmentation strategy for iteratively segmenting

non-uniform shapes from complex multi-color images [38].

The foundation of the GrabCut algorithm is the graph cut algorithm proposed by Boykov and Jolly which uses an initial trimap, denoting the locations of foreground, background, and uncertain areas, to segment a monochrome image [42]. From the image which will be segmented and the trimap provided, a graph is created where the nodes are pixels from the image and the edges between the nodes are assigned a weight based on their similarity. The trimap is used to define the set of nodes that must be separated and is created through user input. The user is expected to draw several marks identifying the background and foreground. To then separate the nodes that were marked as the foreground and background, the graph is split between dissimilar nodes by cutting across low-cost edges with the objective of finding the minimum possible cut cost. This algorithm was shown to be effective at segmenting a variety of images, including CT images and slices from a 3D MRI exam [42]. However, the Graph Cut algorithm was limited by the amount of input that the user had to give to the algorithm and attempted to create a perfect segmentation based only on the input of the user in one attempt [38].

Rother et al. improved this algorithm further and developed the GrabCut algorithm which sought to reduce the amount of user input and improve the behavior of the algorithm around the border to reduce the number of leftover artifacts. Rather than building a traditional graph, Gaussian Mixture Models (GMM) are used to represent all the pixels in the image and their data. An energy function is then created that accounts for the pixel value similarity and the relative location of each pixel in the image. To segment the image, the parameters of the GMMs for the foreground and background are modified to minimize the energy function. Once the minimization has occurred, the first major improvement to the algorithm occurs where it iterates on the minimization process until a global minimum of energy is reached. This iteration feature then allows the user to specify partial trimaps because the GrabCut algorithm does not attempt to completely segment the image on the first iteration. Rather uncertain areas are carried over until more information is known on

future iterations. Finally, a border matting operation is added to the algorithm to improve the border of the segmented area and reduce artifacts leftover. This operation smooths out the border of the segmentation by taking the hard segmentation border, created from the iterations, and applying a soft step function whose arguments is the given pixels distance from the hard edge [38].

5 Introduction to the *RASTER* System

In this thesis, a robot assistant for sonographic tissue examination and reconstruction (*RASTER*) is proposed that augments the capabilities of physicians and sonographers in an easy-to-use and strain-free package while reducing human-induced error and maintaining the flexibility that ultrasound imaging offers. The key contribution of this system is the use of a region-based segmentation method that can be used to identify any visually distinct region, such as the thyroid gland, with no prior information about that region. *RASTER* then uses this information to estimate the volume of the identified region and provide a visualization of the volume to the operator, whether that be a physician, sonographer, or other trained personnel, for further analysis.

With *RASTER*, the standard thyroid ultrasound examination procedure could be simplified without any significant changes to the spaces where these examinations are conducted. Patients would continue to be positioned such that their neck is exposed, whether that be sitting or lying down, in a standard exam room. Once ultrasound gel is applied to the neck, the ultrasound scanner of the *RASTER* system would be placed on the patients neck such that the thyroid gland is at least partially visible in the ultrasound image. The operator would then initialize the control systems of *RASTER* and identify the thyroid gland within the ultrasound image. Once fully initialized, the operator would command *RASTER* to scan the patient and when complete, generate a 3D volume estimate and visualization. Throughout the scans the operator can view the live ultrasound image and when finished scanning, move *RASTER* to any location on the neck for further examination.

The remainder of this thesis is laid out as follows. Section 6 details how the *RASTER* system was designed and built. Section 7 explains the experiments conducted to validate the *RASTER* system. Section 8 discusses the results of the validation experiments and the lessons learned from the design of this system. Section 9 offers directions in which this work could be expanded.

6 Materials and Methods

The *RASTER* system uses a seven degree of freedom robotic arm to move a wireless ultrasound scanner in a controlled motion on the patient’s neck while applying a constant force and maintaining a proper orientation of the scanner relative to the patient’s skin. Concurrently a region-based segmentation pipeline running in real-time, initialized by the operator, is responsible for keeping the region of interest in the ultrasound image. A user interface allows the operator to complete all steps of the examination seamlessly from one device. A secondary non-real time region-based segmentation pipeline is used to generate a 3D representation of the region of interest and provide volumetric information back to the physician to assist in a diagnosis. Figure 4 shows the system in full and Figure 5 breaks down how each functional module is connected, including which information is shared between them.



Figure 4: The complete *RASTER* system.

The *RASTER* system was designed using a ROS architecture to break up the different

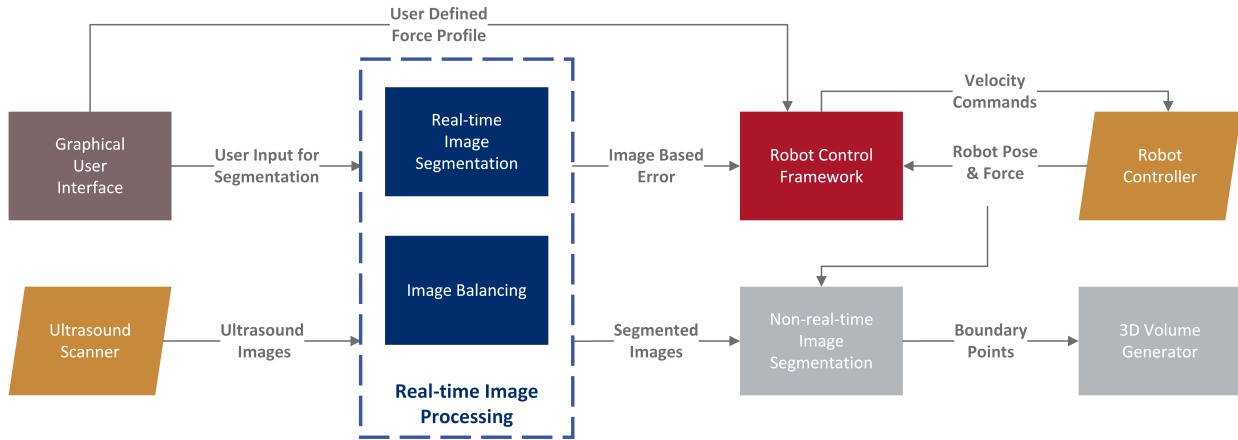


Figure 5: A system diagram of the *RASTER* system.

functions of the system into distinct modules that could run concurrently in real-time. To ensure that the real-time image processing would not be slowed down by the requirements of the robot control framework, two laptops were used in the *RASTER* system. Additionally, a tablet computer was required to connect with the wireless ultrasound scanner and the robot controller was needed to be hardwired to one of the laptops. Figure 6 shows how a local wireless network was used to facilitate communication between the hardware devices of *RASTER*.

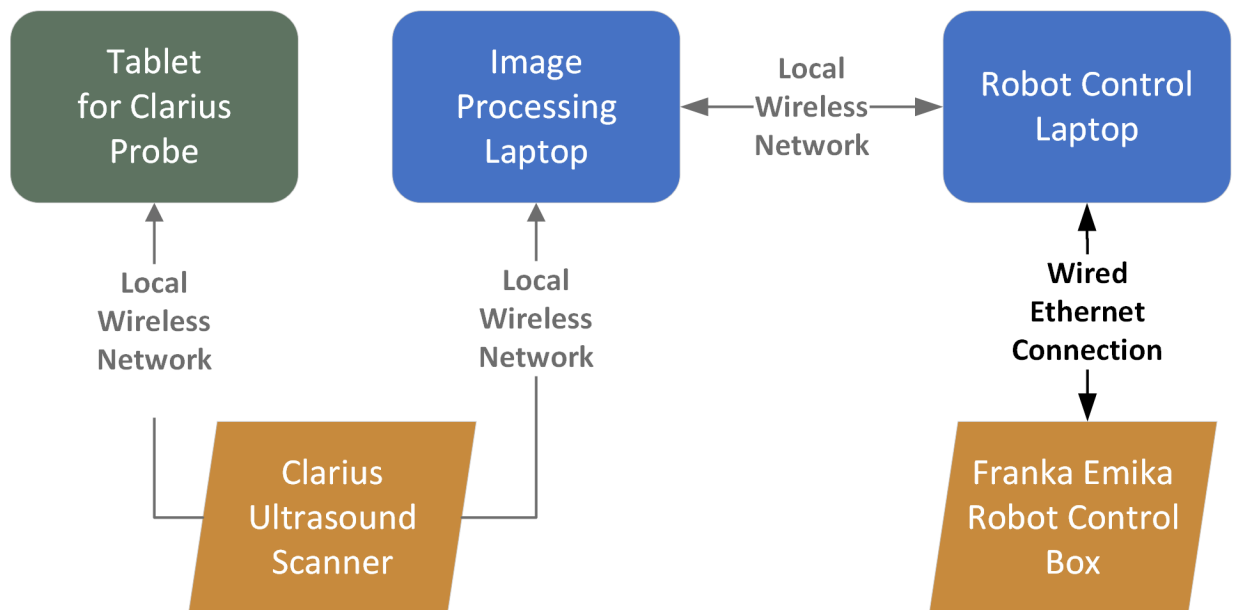


Figure 6: A hardware diagram of the *RASTER* system.

6.1 Integrating the Ultrasound Scanner

The core of the ultrasound image capturing module is the ultrasound scanner, a L7 Linear Scanner made by *Clarius*, shown in Figure 7a. This scanner was chosen for its wide view angle, smaller imaging depth, and wireless configuration. By selecting a wireless scanner, the system does not rely on a full-size ultrasound machine, further reducing the footprint of the final product. To use the *Clarius* scanner, the scanner must be connected to a mobile device, for this project an iPad Pro made by *Apple*, running the *Clarius* mobile application, shown in Figure 7b. Because *Clarius* devices are designed to integrate into research applications, the scanner and mobile device were connected over a local wireless network over which the ultrasound images were then broadcast. *Clarius* releases a package for receiving these images called *cast* which was used to read the ultrasound images into a Python script and rebroadcast them on a ROS topic for the image segmentation pipeline to receive.

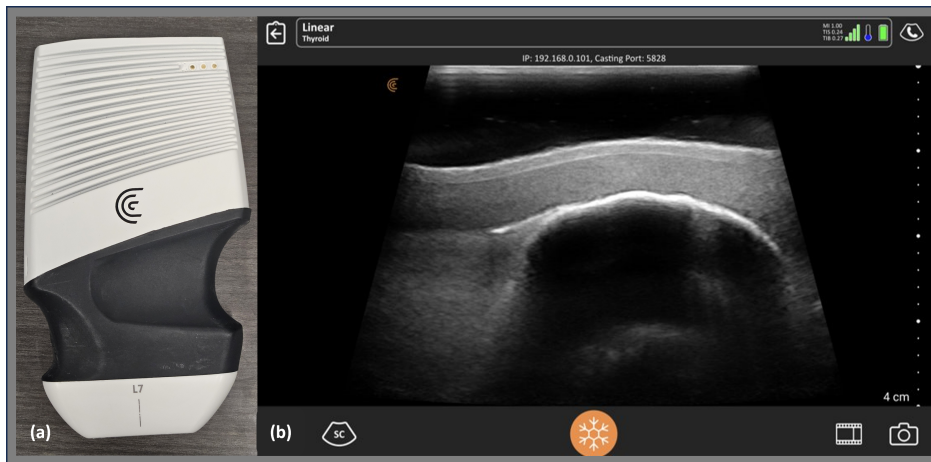


Figure 7: The *Clarius* L7 Linear scanner (a) and the *Clarius* app (b).

6.2 Mounting the Ultrasound Scanner

To mount the ultrasound scanner to the robot, a 3D printed four-piece design, shown in Figure 8d, was developed that allowed the scanner to be easily taken off the robot while remaining durable and stiff. To the end flange of the robot arm, a universal mount, shown in

Figure 8a, was mounted that could be used to repeatedly and accurately attach different end-effectors quickly and without tools. While not strictly necessary for the *RASTER* system, the flexible design allows room for future iterations of the *RASTER* system as well as the possibility of using multiple styles of ultrasound scanners with *RASTER*. A scanner-specific mount, shown in Figure 8b, was then attached to the universal mount. To ensure that the scanner-specific mount was properly oriented, a matching hole for the dowel pin in the universal mount was included. The scanner-specific mount was included in the design so that the scanner could be removed to replace the battery or allow the operator to use the scanner with minimal impact from the mounting system. The third and fourth piece of the design, shown in Figure 8c, were two blocks that mounted to the scanner itself. They were manufactured to match the shape of the handhold on the scanner and then screw together creating a strong hold on the scanner without requiring any modifications to the scanner itself. The scanner with mounting blocks attached was then mounted to the scanner-specific mount using four screws, of which only one needed to be tightened to ensure a strong connection, the remaining three acted purely as locational supports. A key criterion for the mounting of the scanner was that it would not interfere in the use of the scanner in tight spaces. Therefore, a complex convex shape was used to reduce the overall size of the mounting blocks at the bottom near the scanner while providing strong structural support at the top.

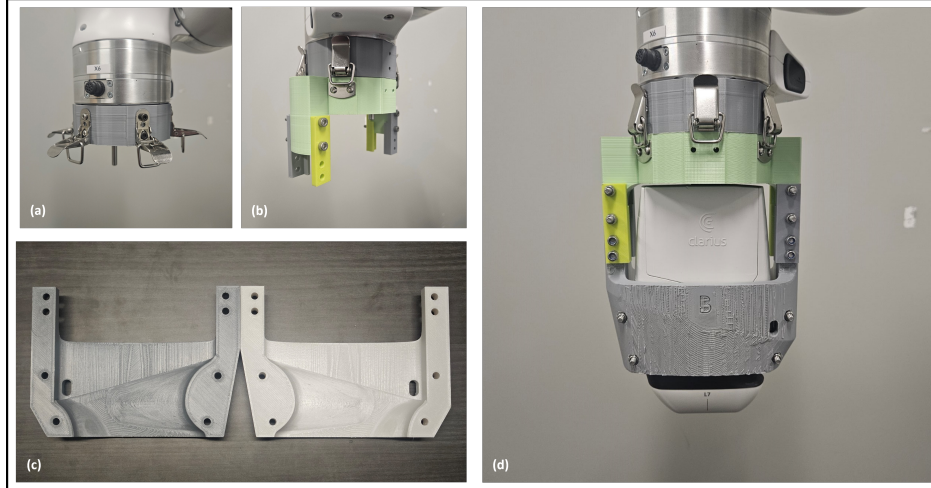


Figure 8: The hardware for mounting the ultrasound scanner to the robot.

6.3 Selecting the Robotic Manipulator

The *RASTER* system uses a seven degree of freedom force-sensitive robotic manipulator to safely move the scanner around the patient’s neck. The *Franka Research 3* manipulator by *Franka Robotics*, shown in Figure 9, was chosen because it is force compliant by design, which prevents the robot from applying excessive force to the patient, providing an additional layer of safety for the patient. Furthermore, the robot includes an emergency stop switch, that when pressed, will stop all motion of the robot which provides another layer of safety within the system. The robot can be moved into any pose by manually manipulating it at any point along its body which simplifies the setup for the operator. The robot is directly connected to a control box where the low-level control of the robot is done. The control box is connected to a Dell G15 5520 laptop, running Ubuntu 20.04.6 LTS on a Linux 5.9.1-rt20 kernel, via an ethernet cable. A real-time kernel version of Linux was used because *libfranka*, the control library for the *Franka* manipulator, requires real-time priority to maintain a 1 kHz communication rate between the control box and any external workstation. High-level control of the manipulator is done in two ways, the first being through a browser-based application built into the *Franka* control box and the second being a third-party software package.

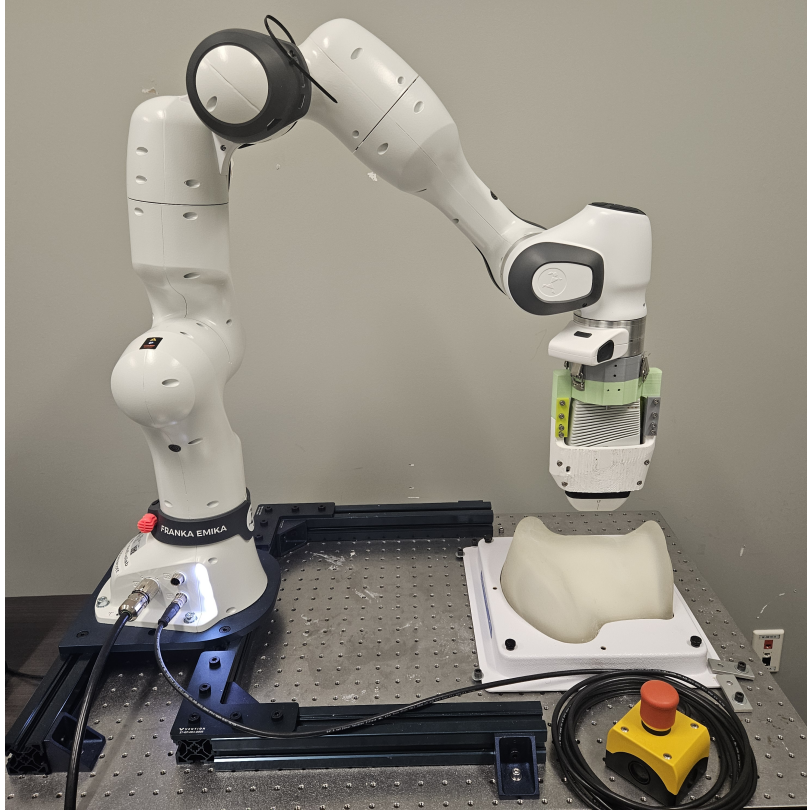


Figure 9: The *Franka* robotic manipulator used by *RASTER* .

Franka Desk is a browser-based application, shown in Figure 10, required to prepare the robot for use. Through *Franka Desk* the connection between the laptop and the robot is verified, the joints of the robot are unlocked, the physical characteristics of the end-effector are loaded, and the *Franka Control Interface* is activated. The *Franka Control Interface* must be active before any commands can be sent to the robot from the third-party software package. Crucially, *Franka Desk* allows the operator to switch between “execution” mode and “programming” mode. In “programming” mode, the robot can be freely moved around by the operator. With *RASTER*, the operator would use this operating mode to place the ultrasound scanner in approximately the correct position using the ultrasound image as a guide. Once placed, the operator would then switch the robot into “execution” mode to complete the remainder of the examination.

The third-party package used for high-level control of the robot is called *armer_panda* and is released by the *QUT Center for Robotics*. This package internally handles all logic to move

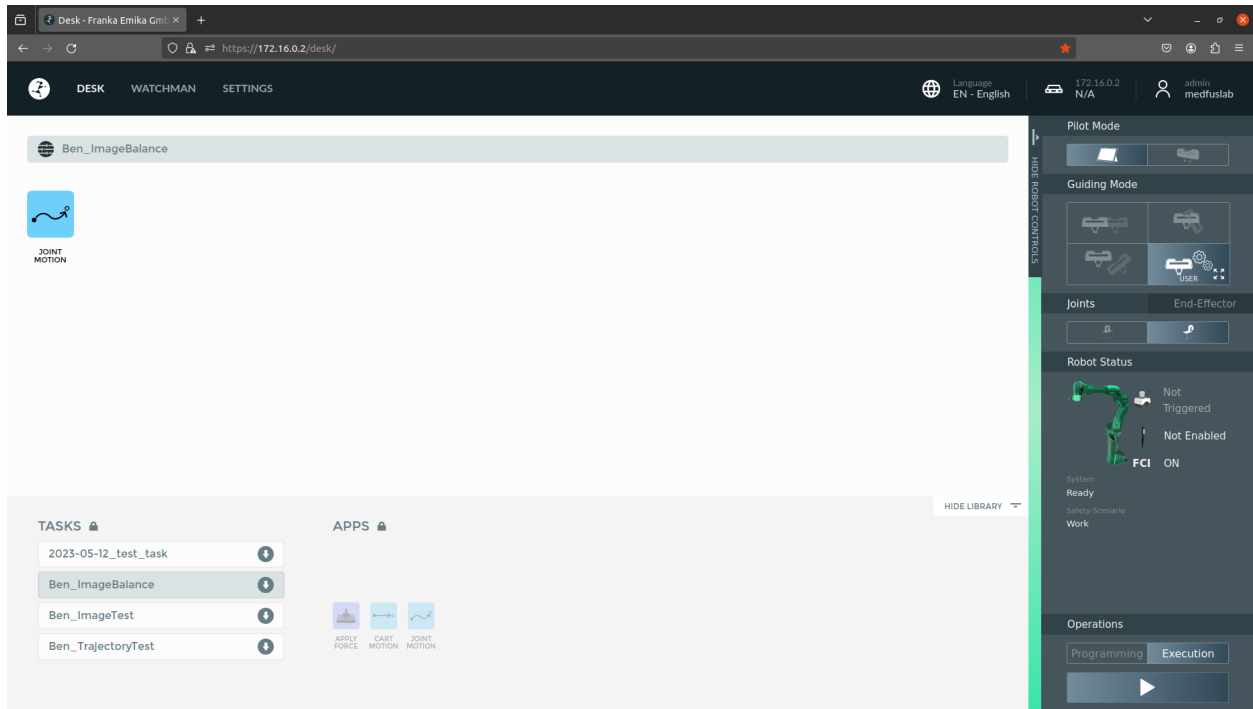


Figure 10: The landing page of the *Franka Desk* application.

the robot and provides several simplified topics to see data about and send commands to the robot. Once the robot has been placed in the approximate position and put in “execution” mode, the *armer_panda* package is launched from the command line using a single *roslaunch* command at which point the robot is ready to use.

When setting up *RASTER* for the first time, the *armer_panda* package had to be given information about the end-effector of the robot. This was in the form of a URDF file that described the physical characteristics of the end-effector, including size, weight, and moment of inertia.

6.4 Designing the User Interface

The user interface allows the operator to prepare for, supervise, and review the results of an examination in a single, easy-to-use interface. The user interface contains two sections that are always visible, one containing the manual robot controls and the second containing the control scheme selection buttons. The operator can use the manual robot controls to

adjust the robot's pose without using the *Franka Desk* to move the robot directly. The control scheme selection buttons allow the operator to enable and disable each form of control at any time giving them final control over how the system operates.

The 'Exam Setup' tab of the user interface, shown in Figure 11, is used by the operator to prepare for the examination. The top section displays to the operator the force that the robot is applying to the patient and provides a method for the operator to set a new desired force for the robot to apply. Additionally, two adjustment buttons allow the operator to make minor adjustments in the desired force quickly and easily. Below that, the operator can select to crop the raw ultrasound image coming into the system. By default, the system uses a saved image cropping defined previously but a new image cropping can be created if a new ultrasound scanner is being used. To create a new image cropping, the operator interactively selects two points on the raw ultrasound image where the image cropping will start and end. Afterwards, because the current imaging depth cannot be retrieved directly from the scanner through the *cast* package, the operator must input the imaging depth of the scanner which is later used in the volume estimation. Finally, the operator must select the region of interest that will be tracked in the ultrasound image, such as the thyroid gland, a specific nodule, or some other visually distinct aspect of the anatomy. To select this region, the operator is presented with a single frame of the current ultrasound image and asked to select the region of interest by placing points around the perimeter of the region. The operator must select at least three points to create an enclosed area and lines are drawn between each consecutive point to display their progress. From the list of points, a convex polygon is generated which is then transformed into a series of triangles. From these triangles, a set of bounding shapes is generated that are then used to find all the pixels within the region selected by the user. From this information, a mask is built that initializes the real-time segmentation process.

Once ready to start the examination, the 'Thyroid Exam' tab, shown in Figure 12, is used to control the examination and select where the data saved from the examination should be placed. To conduct an examination, the operator selects the linear distance for the system to

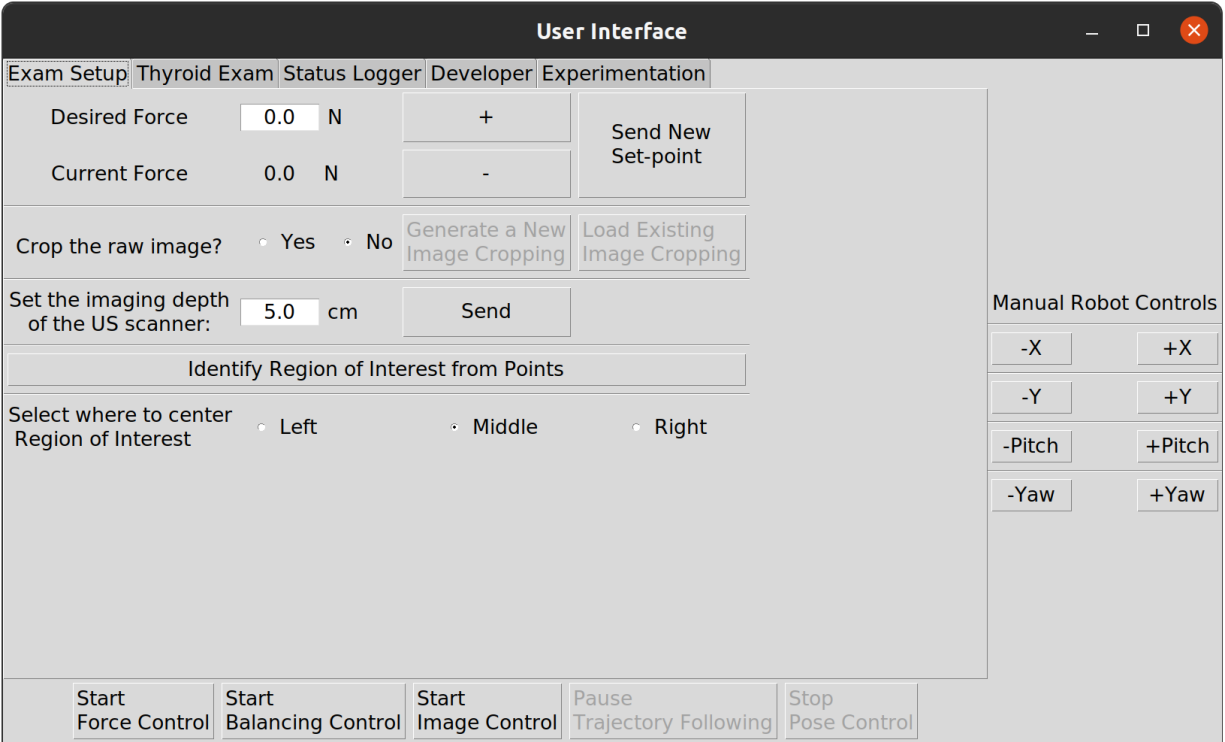


Figure 11: The 'Exam Setup' tab of *RASTER*'s user interface.

scan and selects which direction to scan. Notably, the user interface will not let the operator conduct a scan until both force control and image balancing control have been activated. Prior to starting the exam, the operator selects where the data gathered in the exam will be stored if the default location is not appropriate. After completing an exam or if the operator wants to review the results of a previous exam, they select the data they would like to view and regenerate the results.

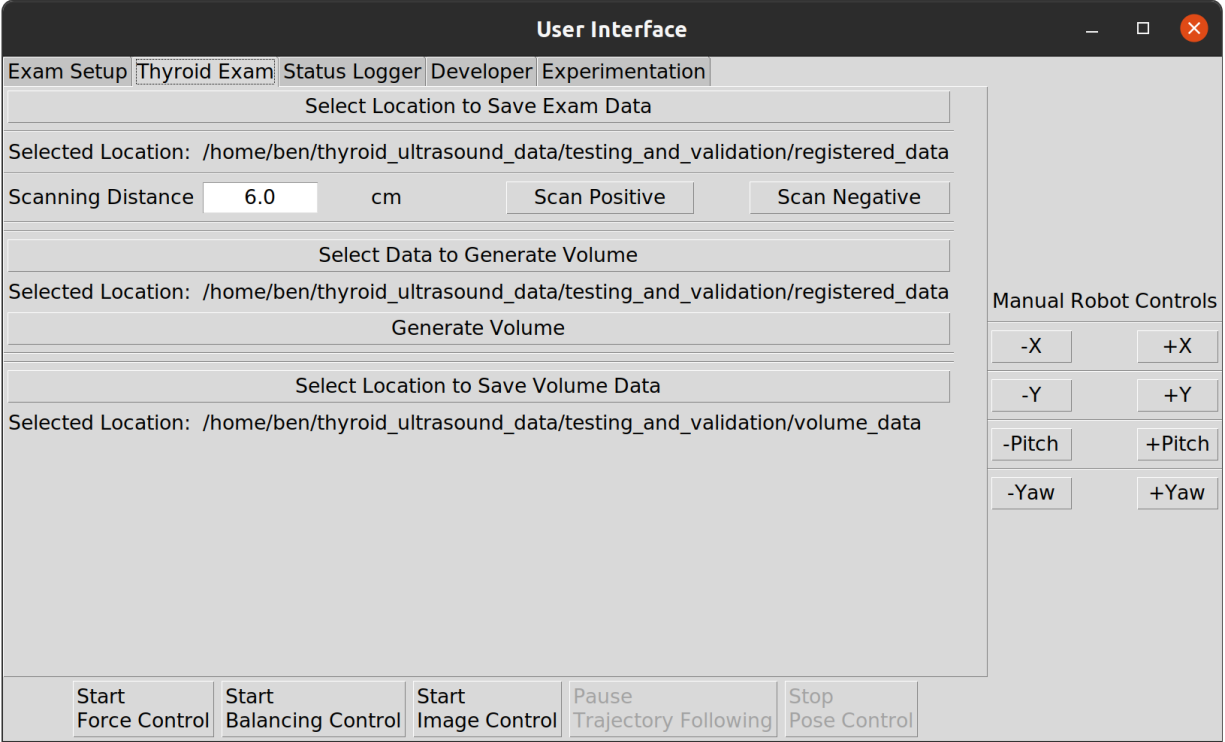


Figure 12: The 'Thyroid Exam' tab of *RASTER*'s user interface.

6.5 Developing a Real-time Image Segmentation Framework

RASTER uses a real-time image segmentation framework that tracks visually distinct areas selected by the operator using the *GrabCut* algorithm. A non-machine learning segmentation method was chosen because, as an assistive device, an operator is present that can provide the initial information required to start a segmentation and this choice removed the requirement that large data sets of labelled ultrasound images be available to be used as training data for a learning-based segmentation. Additionally, by selecting a non-machine learning method, less computing resources are required for the system to function and allows *RASTER* the flexibility to work with different ultrasound scanners and new anatomical regions without retraining the segmentation framework.

RASTER's image segmentation framework functions based on an assumption of image similarity between any pair of consecutive images. Specifically, the image segmented at time i must be reasonably similar to the image segmented at time $i - 1$. If those two images are

reasonably similar, then it is assumed that the result of the segmentation of the image at time $i - 1$ is a reasonable guess for how the image at time i should be segmented. Because this assumption is not guaranteed to be true, several design choices were made to ensure that assumption held true for *RASTER*. Foremost, the real-time image segmentation framework was designed to work in as close to real-time speed as possible. The parameters of the *GrabCut* algorithm were tuned to prioritize speed over accuracy and the overall speed of the robot was limited to prevent large, sudden movements. Additionally, through implementing other control schemes, such as force control and image balancing that track the movement of the patient in additional axes, the overall picture quality and similarity between pictures was further improved. Despite these safeguards, it is known that *RASTER* cannot handle large, fast motions that the patient may unexpectedly complete.

Before the images can be segmented automatically, the operator must provide input to initialize the framework. First the operator selects how the image should be cropped, through the user interface, to remove any artifacts at the top or bottom of the image that may cause errors in the segmentation process. Once an image cropping has been developed it can be saved and reused in future scans. Next, the operator must enter the scanning depth being used by the ultrasound scanner. This information cannot currently be retrieved directly from the scanner and must be entered through the user interface. Finally, the operator must create the seed for the first segmentation through the user interface. Notably, this seed segmentation does not have to be completely accurate as the segmentation will expand or contract to fit the boundaries of the region it detects.

Real-time image segmentation is handled by the real-time image segmentation module. New images are saved in a first-in last-out queue, to ensure the newest message is always segmented first. Additionally, any stored data that is too old is removed to prevent any errors. Once at least one image has been received by the module, the newest image is removed from the queue and the module attempts to crop the image. If the image cannot successfully be cropped, the original image is passed on to the next step in the process

and any image cropping that had been applied is reset. Then the image is recolored from grayscale to a three channel image as is required by the *GrabCut* algorithm. The cropped and colorized images are then displayed to the operator so the progress of the segmentation framework can be monitored.

Further segmentation of the received images is not completed until the operator has seeded the segmentation framework and has commanded the framework to begin segmenting images. To prevent the segmentation framework from giving false information, the module cannot segment any images unless it has been notified that the patient is in contact with the ultrasound scanner.

A custom *ImageFilterGrabcut* class is responsible for segmenting the images. To allow for future development and improve the modularity of the module, this class extends a custom *ImageFilter* class which defines common functions that any image filter would need to implement to be used in this segmentation framework. Additionally, error catching is built into every layer of this process, including into the custom classes, to ensure that the image segmentation module will not crash at any point. If the image segmentation process does fail, the module will not segment any more images until the filter has been properly reinitialized.

The *ImageFilter* class defines the process for segmenting every ultrasound image received by the module. After the image is cropped and recolored appropriately, the image is down-sampled by a factor of 2.5 to improve the speed of the segmentation significantly. The image is then segmented by applying the *GrabCut* algorithm. To further improve the performance of the segmentation, the *GrabCut* algorithm only performs one iteration on each image. Although this decreases the accuracy of the segmentation on any given image, the rate at which new ultrasound images are received and the similarity between any two consecutive ultrasound images artificially increase the number of iterations performed on any single ultrasound image recovering some of the accuracy loss. Once a mask has been generated by the segmentation algorithm, a list of the contours contained within the image, and the

centroid of each contour, is extracted. After the successful segmentation of each image, the full results of the segmentation, including images from each step of the image segmentation process, are published for other modules to use and for the operator to view if desired. Figure 13 shows the segmentation process applied to a single ultrasound image.

Once the segmentation of the current image has been completed, the image segmentation module must prepare to segment the next image. Since this image segmentation framework assumes image similarity, the resulting image mask from the image just segmented is modified to build the seed segmentation of the next image. The objective of this process is to define masks that contain the areas of the next image where the region of interest is most likely not going to be (the sure-background), where the region of interest might be (the probable-foreground), and where the region of interest most likely will be (the sure-foreground). Because it is possible that the boundaries of the region of interest may shrink in the next image or the region of interest may be in a slightly different location in the image, the sure-foreground area should be smaller than the area identified as the region of interest in the current image. Therefore, the sure-foreground mask is found by applying an erode operation to the result mask. For this operation a kernel of size 3x3, anchored at the center, was used and the operation was applied to the mask three times using the *morphologyEx* function of the *cv2* package. The sure-background mask is found by applying a dilate operation to the result mask and then taking the inverse. For this operation, a kernel of size 3x3, anchored at the center, was used and the operation was applied eight times, to encourage the segmentation to expand rapidly, using the *morphologyEx* function of the *cv2* package. The probable-foreground is then defined as the area not contained in the sure-foreground or in the sure-background and was generated by inverting the sum of the two masks. A final seed array is generated by applying the correct value to each mask and then adding them together. This seed is saved within the image filter to be used on the next image unless the operator generates a new seed through the user interface.

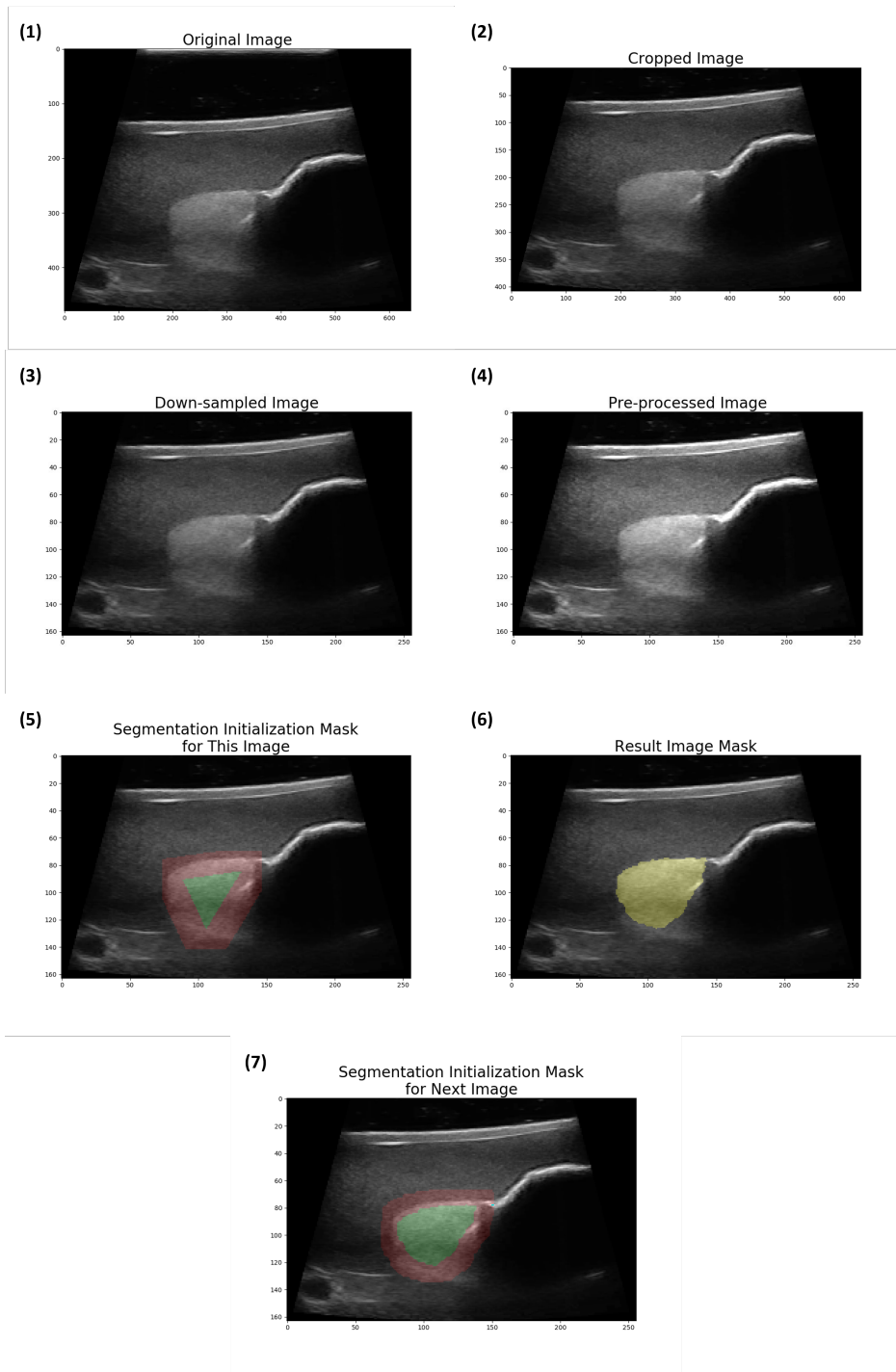


Figure 13: A step-by-step illustration of the image segmentation process.

6.6 Creating an Image Centering Algorithm

RASTER implements an image centering control algorithm that is responsible for determining the error of the location of the region of interest within the image. While the default objective of the control algorithm is to keep the centroid of the region of interest on a line down the center of the image, the operator can choose to shift this goal line to the left or right side of the image when scanning a singular lobe. The centering error is calculated as the horizontal distance, in pixels, between the centroid and the goal line. To prevent excessive movement by the robot, a wide error band around the goal line was instituted. Figure 14 shows the three possible positions of the goal line as well as the error band for each goal line.

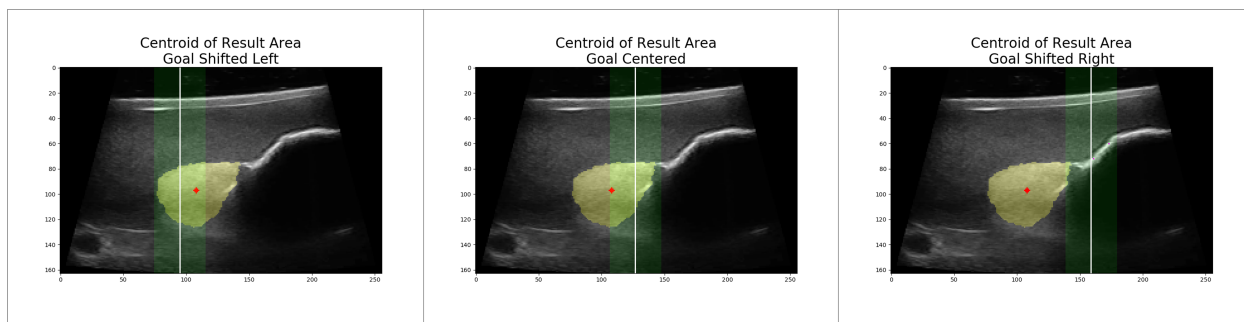


Figure 14: The possible locations of the image centering goal (white line) and error bands (green area).

6.7 Creating an Image Balancing Algorithm

RASTER implements a pipeline to maintain consistent scanner contact with the patient and a perpendicular scanner orientation relative to the local skin plane based on the ultrasound image. Figure 15 shows an example of a high-quality ultrasound image, which has an evenly distributed intensity throughout the whole image, as well as a low-quality ultrasound image, which contains significant dark areas. Dark areas of an ultrasound image occur when the scanner is imaging air. While not all dark areas are an issue, such as when imaging near air-filled anatomical features like the esophagus, they can be used as an indicator that a portion of the scanner is not in contact with the patient. Therefore, the quality of the scanner

contact is estimated by looking at the intensity of different areas of the image. In the case where the ultrasound is sufficiently intense, the definition of a high-quality ultrasound image is expanded to include an ultrasound image where the bright feature visible in the top of the image is approximately horizontal, which correlates to the scanner being perpendicular to the local skin surface in the image plane. On the ultrasound image, the bright feature is approximated as the boundary between the dark area at the top of the image, excluding any artifacts left by the scanner at the top of the image, and the first light area moving down the image.

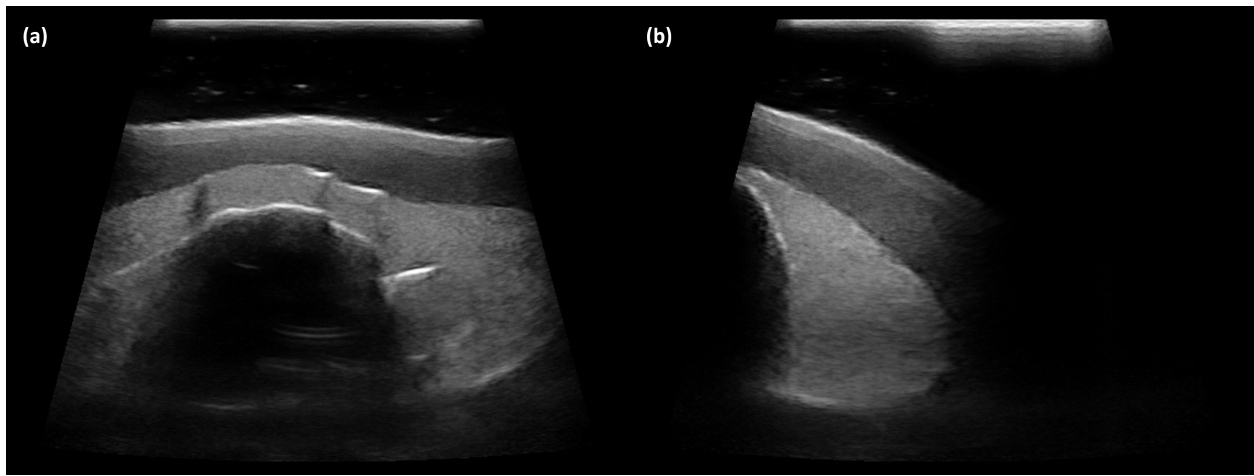


Figure 15: A comparison between a high-quality ultrasound image (a) and a low-quality ultrasound image (b).

Before determining the intensity of individual sections of the ultrasound image, the area of the ultrasound image needs to be separated from the areas filled in with zeros to create a square image. Figure 16a shows an original image received from the scanner and Figure 16b shows the region of that image that contains the ultrasound image. The portion of the image containing the ultrasound image is a wave-shaped area in the center of the full image. This area is approximated as two intersecting lines closed by a horizontal line at the top and closed at the bottom by an arc. The intersecting lines have an angle between them, in this case approximately 30 degrees, and an intersection point that is some distance above the top of the image. The horizontal line at the top is placed at the top of the ultrasound

image and the closing arc at the bottom is placed tangent to the bottom of the image. Once the ultrasound image has been mapped within the larger image, it is broken into individual sectors to be analyzed independently, denoted as bright or dark sections in Figure 16c. Each sector is defined by four bounding sets: a line representing the left edge of the sector, a line representing the right edge of the sector, a line representing the top edge of the sector, and an arc representing the bottom edge of the sector. To improve performance, these sectors are generated when the module is started and then only regenerated if the ultrasound image changes size or shape during operation. At startup two lists are generated, one containing the sectors on the left-half of the image and one containing the sectors on the right-half of the image.

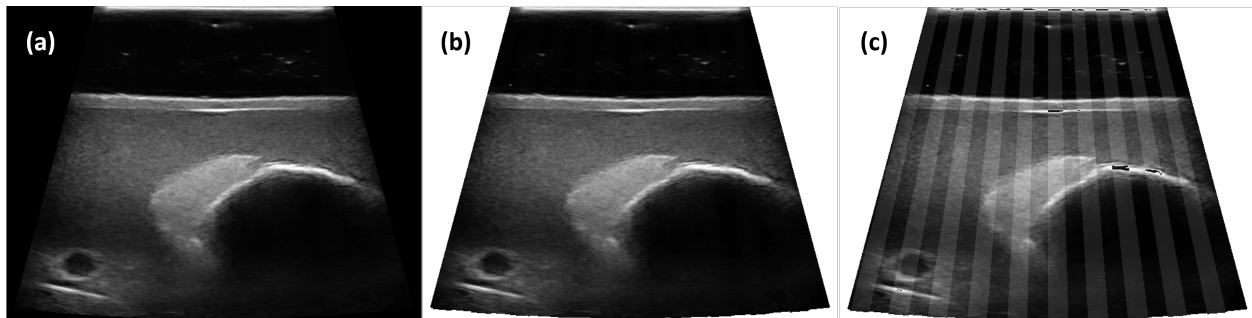


Figure 16: A step-by-step illustration of the creation of sectors from a raw ultrasound image.

Once an image is received by the image balancing module, it is placed in a first-in last-out queue so that the most recently received image is the first image to be analyzed. Similar to the real-time image segmentation module, data that is too old is removed to prevent errors. To analyze the image, the module must iterate through every pixel contained within the ultrasound image and determine which image sector contains that pixel. To minimize the amount of time required to complete this process, several algorithmic design choices were made to prioritize efficiency. First, the image is down sampled to reduce the number of pixels visited by a factor of ten which improved the performance of the node with no cost to the accuracy of the algorithm. Then, rather than iterating through the entire width of the full image, the algorithm benefited from the closed form solutions of the sector boundaries

to iterate through only the ultrasound image area. Finally, since the sectors were generated sequentially from left to right in the ultrasound image and the pixels of each image row are iterated through from left to right, when the sector is found which contains a given point, j , the first sector checked to find point $j + 1$ is the sector that contained the previous point. The search then checks each remaining sector moving to the right of the image which ensures that, at most, the location of any pixel is found within two calculations.

Once a pixel has been matched with the sector that contains it, the number of pixels found that are contained within the sector is incremented and the average intensity of every pixel found is calculated. Once the iteration of all pixels is complete, the average pixel intensity is used to determine if each sector is bright compared to a baseline intensity level. The number of sectors considered dark in the left-half and right-half of the ultrasound image are counted and the difference between these two numbers is calculated. If less than 25% of the image is dark, then a suitable amount of the scanner is contacting the patient, and an error value is calculated as the difference in the number of dark sectors in each half of the image divided by ten.

However, if less than 10% of the image is dark, then the error will be calculated by approximating the bright feature. To find the bright feature in the ultrasound image, a ray-based search was conducted on each bright sector in the ultrasound image. Since each image sector is defined by a linear set on both the left and right side, these bounding sets are used to search along a line without iterating through every pixel in the image. Because each bounding set has a closed-form solution in the form of $y = ax + b$, the search calculates the corresponding column of the image array, the x value in the above equation, given any row, or y value, for the image array. Therefore, for each sector, a search is conducted from the top of the image to the bottom of the image along the line defining the left edge of the set. To prevent the artifact left by the scanner at the top of the image from causing errors, the search starts at a distance away from the top of the image. For each row, the corresponding column that falls on the sector boundary is calculated. If the pixel at that row and column

position is twice the minimum intensity, a counter is incremented counting how many bright points along the ray have been found. The bright feature is considered to have been found along that ray once five points have been found. The last bright point found is added to a list of points representing the boundary of the bright feature. The search then moves to the next sector and repeats until all sectors have been searched. Figure 17 shows the boundary points as orange dots for points found on the left-hand side of the image and yellow dots for points found on the right-hand side of the image. By implementing a counter on the number of bright pixels found along a ray, small bright artifacts that exist above the boundary of the bright feature are ignored. While this creates an error in the depth of the bright feature, all points found on the bright feature will contain the same level of depth error. If bright feature boundary points are found on both the left-half and right-half of the image, a single straight line is approximated from these points using the *polyfit* library, the green line in Figure 17. The slope of this approximated line is then used as the balance error for the image. The equation for the approximated line is also published so that the result can be visualized by other modules.

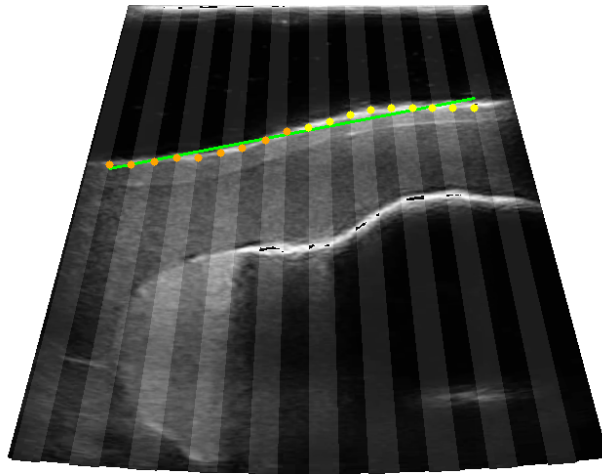


Figure 17: An example of the image balancing approximation line generated from boundary points.

6.8 Implementing a Trajectory Generation Algorithm

RASTER uses a trajectory generation framework that ensures that each ultrasound imaging is captured at an equal distance from one another and that the imaging plane for each image is parallel to the initial imaging plane of the trajectory. As shown in Figure 18, the imaging plane of the ultrasound scanner is the YZ plane. Therefore, the trajectory is only responsible for controlling the robot motion such that the scanner reaches each imaging plane by travelling along the X axis while maintaining the same angular position about the scanner's Y and Z axes. Importantly, this requires the trajectory to allow free linear motion of the scanner relative its own Y and Z axes as well as free angular motion about the scanner's X axis.

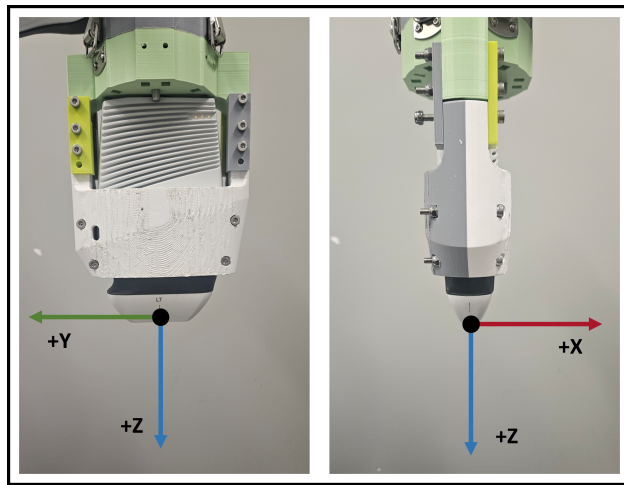


Figure 18: The definition of the axes of *RASTER*'s ultrasound scanner.

Once *RASTER* receives a command to complete a scan, which must include the direction and distance in which to scan, from the user interface, the trajectory generation process is started. The starting pose of the ultrasound scanner serves as the origin of a new coordinate system in which the trajectory will be defined and is shown as the coordinate system on the green plane in Figure 19. The end point of the trajectory is defined as a plane offset from the YZ plane of the of the origin by the distance and direction included in the command. Waypoints for the trajectory are then generated at standard intervals, d millimeters apart

as shown in Figure 19, between the start and end points. Once the endpoint and each waypoint are defined, the trajectory is created as a series of planes stretching along the X axis that the end effector must pass through as shown in Figure 19. Every time the trajectory module receives an updated pose of the robot, the trajectory error is calculated as the distance between the current position of the robot and the destination plane measured along the X axis of the trajectory coordinate system. Additional error values are calculated as the difference between the current and starting angular position of the scanner about the Y and Z axes. To ensure that consistent data is being collected throughout the exam, the trajectory module publishes when a waypoint has been reached so that an ultrasound image can be captured before the robot moves to the next waypoint. Furthermore, the trajectory will not update the goal waypoint until it has been confirmed that an ultrasound image has been saved at the current waypoint. To ensure that the trajectory module cannot behave unexpectedly, the module cannot calculate an error value unless the robot pose is known to the module and the ultrasound scanner is in contact with the patient, as determined by the image balancing module.

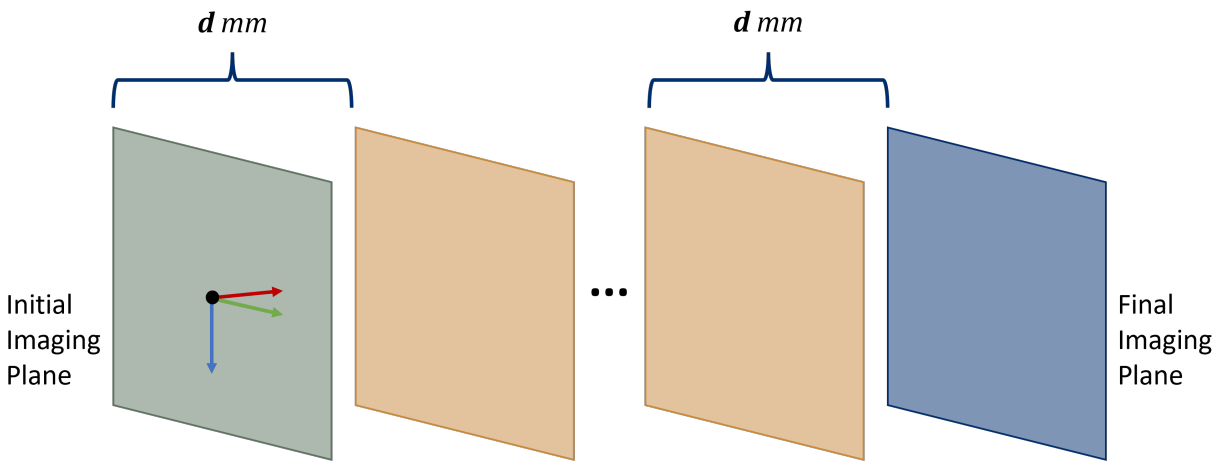


Figure 19: A theoretical plane-based trajectory with d millimeters between each image.

6.9 Implementing a Force control Algorithm

RASTER implements a force control framework that regulates the amount of force the robot applies to the patient to ensure patient safety, improve image quality and similarity, and account for small patient movements caused by breathing or other similar motions. The operator is given final control over the system's behavior by specifying the exact force applied by the robot to the patient for the force control module. This value is used as the setpoint for the force control module and is compared with the force exerted on the robot's end-effector which is estimated at over 300 Hz by the *armer_panda* package based on the torque applied to each joint. Because this data is noisy, the last 100 data points are averaged to find the current force value. To further ensure patient safety and prevent unexpected motion, the force control module cannot command the robot to move unless the ultrasound scanner is in contact with the patient, as determined by the image balancing module.

6.10 Registering Corresponding Image and Position Data

Because ultrasound images do not contain information about the pose of the ultrasound scanner when the images were taken, a module was created to register ultrasound images with the pose of the robot and the force applied to the scanner when the image was taken.

To accomplish this, each data point for each type of data, ultrasound image, robot pose, and force, are stamped with the time at which that data was published by its source module. The image and position registration module listens to each of these data sources and records data for each one. When a new message containing data is received, the data point is placed in its respective dictionary. The keys for these dictionaries are the time at which the data was captured measured in seconds since the epoch. The value for each key is a lower-level dictionary. In the lower-level dictionary, the keys are the time the data was captured measured in nanoseconds since the capture time in seconds. To prevent a build-up of data slowing down the system, any upper-level key which is more than ten seconds older

than the newest key is removed whenever new data is added. Additionally, when adding a new ultrasound image data point, new data is only be added when the trajectory following module is active, the current waypoint has been reached, and an ultrasound image has not already been registered for the current waypoint. This prevents the module from saving an excessive amount of data while ensuring that one data point is captured at each waypoint.

While the recording of data points occurs as callbacks whenever a new message is received, the main functionality of the module occurs constantly in the main loop of the module until the module is shut down. To prevent data loss due to old data being deleted in a callback while the main loop is using the same data, the dictionaries containing the data points are deep copied at the start of every main loop. Furthermore, the main loop will not try to register data unless there are data points in all three dictionaries. Once these conditions have been met, the newest upper-level key is retrieved from the ultrasound image dictionary. This key is checked against the remaining two dictionaries to ensure that potential data matches exist for the ultrasound image data. Additionally, the lower-level dictionary is checked to ensure that it is not empty. Then each nanosecond key for the ultrasound images is used to find the corresponding robot pose and force data points by finding the data point with closest nanosecond key. After the corresponding data has been found, it is saved to the computer in a location set by the operator, published for other modules to see, and then the individual data points are deleted from the dictionaries.

6.11 Developing a Non-real Time Image Segmentation Framework

While real-time image segmentation is required to sacrifice accuracy for performance, the non-real time segmentation focuses on accurately segmenting the selected area of the ultrasound image regardless of time required to accomplish this task. Regardless of the different objectives, the process of segmenting images in non-real time is almost identical to that of the real-time segmentation. Both processes use the *GrabCut* algorithm to segment the images and even share many code structures. The key differences lie in the selection of

the tuning parameters for the *GrabCut* algorithm. The first parameter choice was to remove the down-sampling step from the process thereby retaining the full amount of information from the image. The second choice was to increase the number of segmentation iterations completed on each image, giving the segmentation more time to fully expand or contract as needed. Finally, the size of the area defined as possibly containing the background was expanded to allow the segmentation to find the true extent of the selected region. To improve the usability of this module, the ultrasound images segmented by this module are not the live-streamed images from the ultrasound scanner but rather the saved data captured by the image registration module, thus allowing the operator to view data captured at any time without having to actively be scanning the patient.

The non-real time segmentation process is started when the operator has requested to view the volumetric data from a recorded examination. Once the operator has selected the data to load through the user interface, the data is loaded from the save-location into an array in chronological order based on the timestamp of the ultrasound images. Contained in the saved data is the raw ultrasound image and the full set of information from how the real-time image segmentation was completed. This information is then used to set the cropping parameters of the non-real time image segmentation. Additionally, the resulting mask of the real-time image segmentation is modified to seed the segmentation of every image for the non-real-time image filter. Once initialized, the loaded data is fed sequentially into the image segmentation until all the images have been processed. Throughout the segmentation, updates on how many images have been segmented and how many total images must be segmented are published by the module. These messages, while helpful for debugging purposes, serve as the notification for the volume reconstruction module that the image segmentation process has completed.

6.12 Estimating the Volume of the Region of Interest

Once a full set of data has been collected, the volume reconstruction module estimates the volume contained within the data set and builds a single 3D shape to visualize the result. For each image, the non-real time image segmentation found the boundary of the region of interest in terms of pixels within the image. From this boundary, the area contained within it is found using the image resolution, as calculated from the imaging depth of the ultrasound scanner and denoted as IR , to find the area of the region of interest in each slice in square meters as shown in Equation 3.

$$A_{mm^2}^i = A_{px^2}^i * IR^2 \quad (3)$$

Using the distance between any two slices, denoted as x_i^{i+1} , and the area of two consecutive slices, denoted as $A_{mm^2}^i$ and $A_{mm^2}^{i+1}$, Equation 4 is used to approximate the volume between any two sequential ultrasound images.

$$V_i^{i+1} = \frac{(A_{mm^2}^i * \Delta x_i^{i+1}) + (A_{mm^2}^{i+1} * \Delta x_i^{i+1})}{2} \quad (4)$$

The total volume of the region of interest across all of the ultrasound images is then be found using Equation 5 where n is the number of ultrasound images.

$$V_T = \sum_{i=0}^n V_i^{i+1} \quad (5)$$

To visualize the volume, the resolution of the ultrasound image is used to convert the location of each boundary pixel to a location relative to the to the end of the ultrasound scanner measured in meters. Then by applying the corresponding robot pose as a homogeneous transformation matrix to each boundary point, the points from each ultrasound image are mapped to the origin frame of the robot. Once all the points have been transformed, they are plotted on a 3D plot and a 3D Mesh, from the *plotly* package, is plotted on top of

them. This package uses an alpha parameter to fit a concave 3D shape to the data points and therefore is not completely accurate. However, it provides enough information to complete the visualization.

6.13 Controlling the Robotic Manipulator

To create a more robust system, one module in *RASTER* is exclusively responsible for sending velocity commands to the robot, the robot control module. This module publishes commands to move the end-effector of the robot at a given speed in each cartesian direction relative to the end-effector's coordinate system in meters per second. Additionally, this node is responsible for denoising the estimated force signal since that signal also comes from the robot.

Contained within this module are six PID controllers responsible for converting the error values calculated by the other modules into control inputs for each degree of freedom in space. Each time a new error message is received by the module, it is stored to be used for calculating the control input in the main loop of the module. Within the main loop of the module, a control input of zeros is generated by default to prevent the robot from moving unless directly commanded. Each error value is then used to calculate the corresponding control input using the appropriate PID controller. Then, based on which control schemes the operator has activated, the control inputs generated from each control scheme, listed in Table 1, are added to the control input sent to the robot. The manual controls are the only exception to this control scheme where each button pressed by the operator causes a constant control input to be sent to the robot in that axis. The architecture of this module was chosen because the control inputs must be sent to the *armer_panda* package at over 100 Hz. Additionally, all control inputs were calculated relative to the end-effector to allow for the greatest flexibility for the user and to off-load computation on to the *armer_panda* package.

Axis	Movement Type	Controller
X	linear	Trajectory
Y	linear	Image Centering
Z	linear	Force
X	angular	Image Balancing
Y	angular	Trajectory
Z	angular	Trajectory

Table 1: List of control schemes and the axes of motion which they control.

7 Testing and Validation

To validate *RASTER*, a series of tests were conducted that isolated each module and tested the effect each module had on the success of the whole system. All the experiments to validate *RASTER* were conducted on a model 074 thyroid ultrasound training phantom manufactured by CIRS Inc. This phantom was placed near the base of the robot and mounted to the table to prevent it from moving unexpectedly during the experiments. To ensure that each experiment conducted was consistent, the *Franka Desk* interface was used to return the robot to the same position at the start of each trial. To facilitate debugging and testing, an additional interface was implemented to monitor the state of each controller in real-time compared to their set-point. To facilitate testing and capture data for each experiment, an additional tab, show in Figure 20, was added to the user interface that allowed the safety logic, such as the requirement that the patient be visible in the ultrasound image, built into the various nodes to be overridden which was required to complete several of the experiments. On this tab, the data to be saved for each experiment was also be selected. Data would automatically be recorded while a trajectory was being completed and would stop once the trajectory was completed. The amount of data to save would also be displayed since generally the time required to complete the test was less than that required to save all the data from the test.

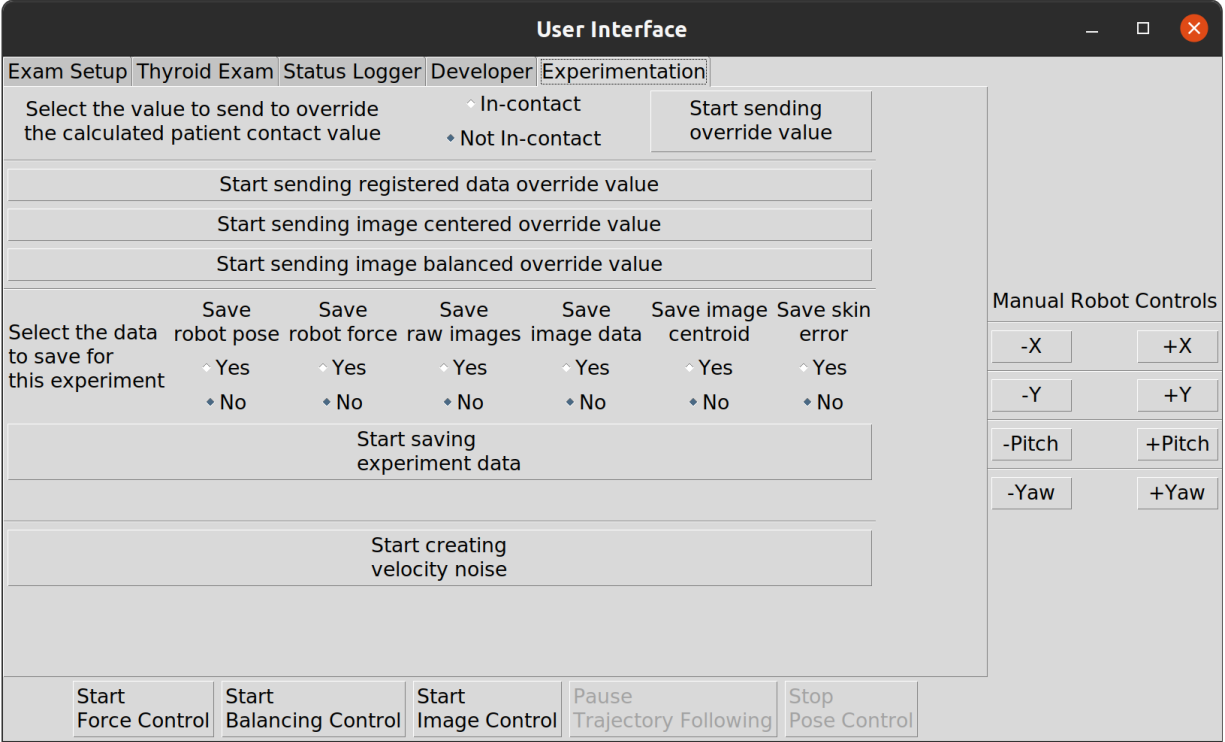


Figure 20: The 'Experimentation' tab of *RASTER*'s user interface.

7.1 Evaluation of the Force Control Framework

An experiment was conducted to verify the efficacy of the force control module at maintaining a constant scanning force regardless of patient motion, scanner motion, or skin profile. For this experiment, the robot was placed in a position such that the ultrasound scanner was fully touching the phantom and applying a non-zero force on the phantom. Critically, the scanner was oriented such that when it was moved in a straight-line trajectory along the X axis of the scanner, the scanner would move out of contact with the patient in the Z axis of the scanner. This starting pose was then saved within the *Franka Desk* application so that the trajectory could be repeated multiple times.

The first set of three trials conducted served as the control and did not include any corrections from the force control framework. Once the experimental setup procedure was completed, the robot was commanded to follow a straight-line trajectory. Throughout the trajectory, the force measured by the robot and the pose of the ultrasound scanner were mea-

sured continuously. These data points were recorded in a CSV file to be analyzed afterwards. The second set of three trials conducted measured the effect of the force control framework on the end-effector force. The exact same procedure was followed with one exception, prior to starting the trajectory, the robot was commanded to maintain a force of 2.0 newtons and the force control framework was activated.

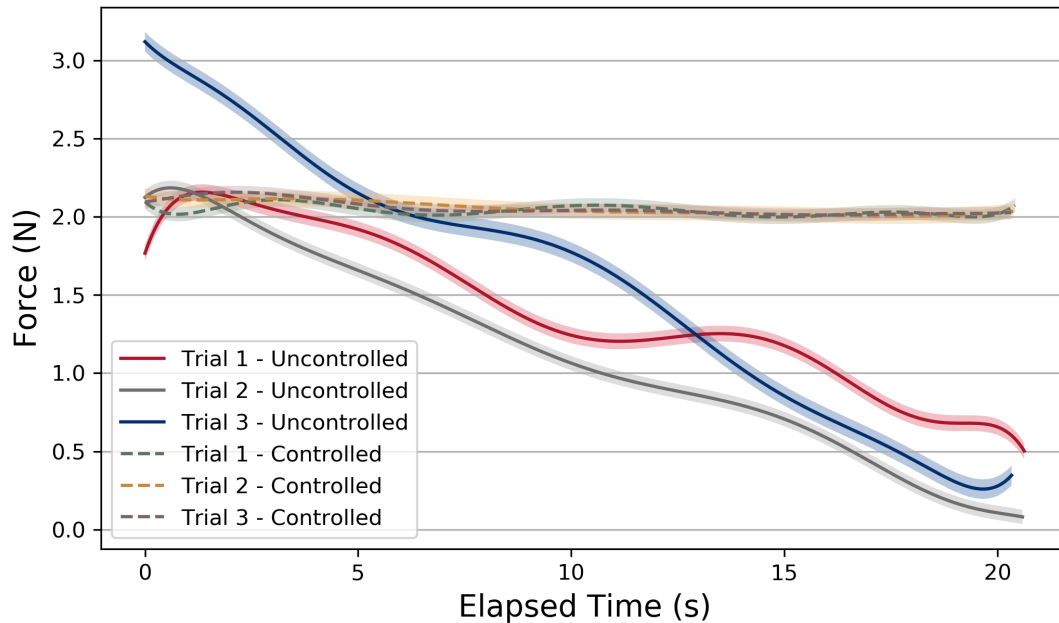


Figure 21: The measured force applied to the patient with and without active force control.

Figure 21 shows how the force applied by the ultrasound scanner on the phantom stayed constant, with a mean of 2.05 newtons and a standard deviation of 0.06 newtons, during the three trials where the force control framework was active. However, on the three trials where the force control framework was not active, the force applied by the scanner dropped by an average of 2.07 newtons with a standard deviation of 0.741 newtons. Figure 22 confirms that a drop in measured force should be expected for the first three trials as the robot did not follow the profile of the skin. However, for the second set of trials, the profile of the skin is seen as the position of the robot moved in Z to compensate for the profile of the skin.

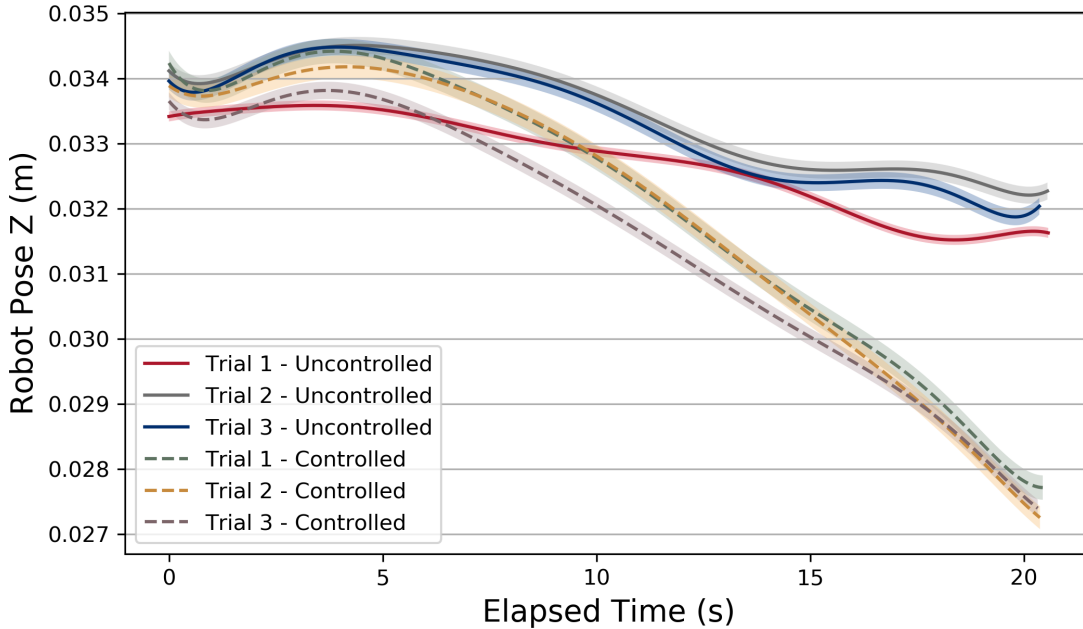


Figure 22: The measured pose of the scanner with respect to the origin of the robot with and without active force control.

7.2 Evaluation of Trajectory Generation Framework

An experiment was conducted to demonstrate the effectiveness of the trajectory module at ensuring that each ultrasound imaging is captured at an equal distance from one another, and the imaging planes are parallel to one another. The experimental setup mirrored that of the force control experiment except that the location of the scanner was selected such that a large displacement in the Z axis of the scanner would need to be compensated for. For each trial, of which there were three total, once the experimental setup was completed, the force control module was activated and a trajectory of six centimeters was generated. Throughout the trajectory the robot pose was measured continuously and recorded in a CSV file and it was noted whenever a waypoint was reached.

Figure 23 shows how the distance travelled along the X axis is linear despite the non-linear nature of the motion along the Y and Z axes. More importantly, the desired distance between each waypoint was two millimeters and the mean distance between each waypoint

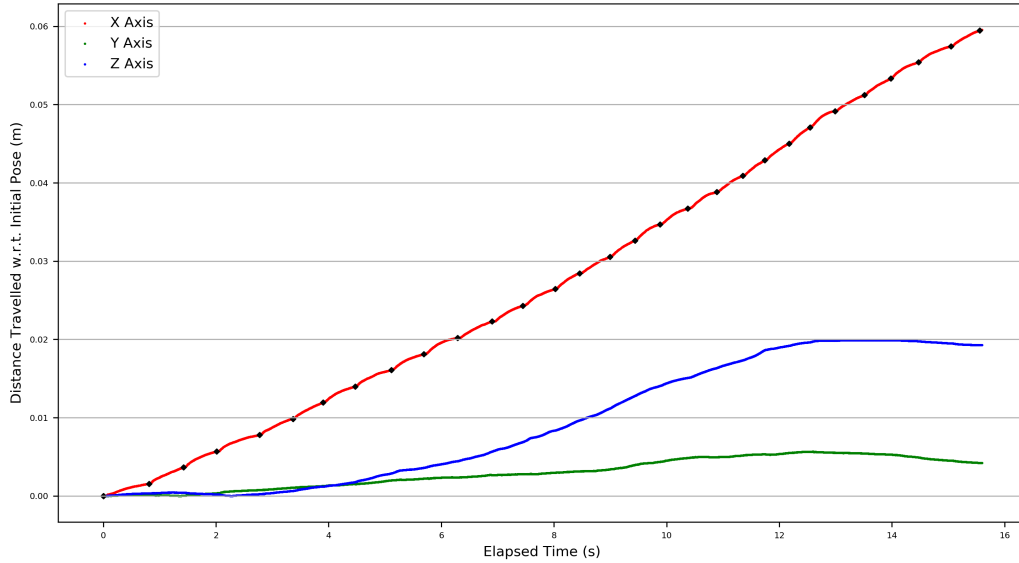


Figure 23: The measured cartesian position of the ultrasound scanner relative to the robot origin over the length of a trajectory where each black diamond represents when a waypoint was reached.

was 2.05 millimeters with a standard deviation of 0.1 millimeters. The starting orientation of the scanner about the Y axis was -0.822 degrees and the scanner maintained an average orientation of -0.748 degrees with a standard deviation of 0.093 degrees. The starting orientation of the scanner about the Z axis was -91.598 degrees and the scanner maintained an average orientation of -91.568 degrees with a standard deviation of 0.023 degrees.

7.3 Evaluation of Image Balancing Framework

An experiment was conducted to demonstrate the image balancing module’s ability to maintain an optimal pose relative to the bright feature. The experimental setup mirrored that of the force control experiment. Notably, the exact starting position of the scanner was chosen such that it was perpendicular to the skin at the start of the trajectory but would not remain so over the trajectory. The force control module was also activated prior to completing any trials to ensure patient contact was maintained throughout the trajectory

regardless of any motion used to balance the image. For the first set of three trials, the image balancing module was not activated and then a trajectory was generated. While following the trajectory, the slope of the line approximating the bright feature was captured in a CSV file continuously. The raw ultrasound images were also captured as individual PNG files. For the second set of three trials, the image balancing module was activated prior to generating the trajectory and data was recorded throughout the trajectory again.

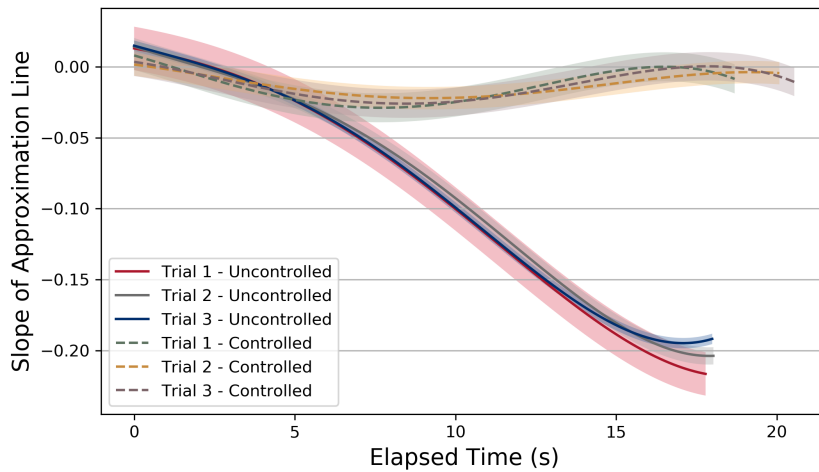


Figure 24: The measured slope of the approximation line with and without image balancing.

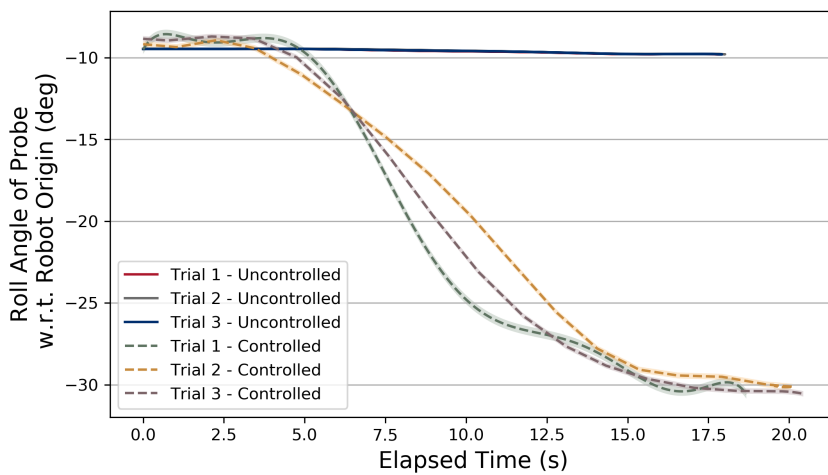


Figure 25: The measured orientation of the ultrasound scanner about the scanner's X axis with and without image balancing.

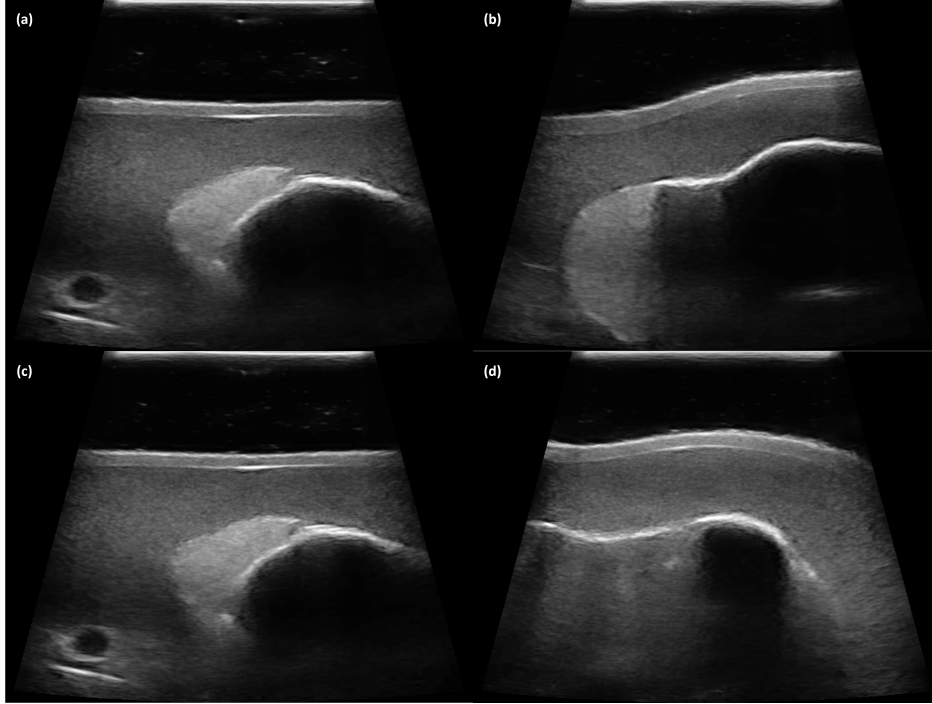


Figure 26: Ultrasound images captured at the start (a) and end (b) of the trajectory when image balancing was not active and at the start (c) and end (d) of the trajectory when image balancing was active.

Figure 24 compares the slope of the approximation line over the length of the trajectory for the six trials conducted. For the three trials where the image balancing module was active, the average slope of the line was -0.013 with a standard deviation of 0.013 . Meanwhile for the three tests where image balancing was not active, the slope of the line decreased by a mean of -0.211 with a standard deviation of 0.005 . Figure 25 shows how the image balancing control caused the robot to change its roll angle over the length of the trajectory to remain in contact with the phantom. Figure 26 shows qualitatively how the ultrasound images captured while the image balancing module was active were more similar throughout the trajectory than those captured while the image balancing module was inactive.

7.4 Evaluation of Image Centering Framework

An experiment was conducted to demonstrate that the image centering module keeps the region of interest centered in the ultrasound image throughout a trajectory regardless of the

patient profile. The experimental setup mirrored that of the image balancing experiment. Besides activating the force control module prior to the start of the trials, the image balancing module was also activated to ensure that the scanner was properly contacting the phantom throughout the trajectory. For the first set of three trials, the image centering module was not activated prior to following the trajectory. Throughout the trajectory, the location of the centroid of the region of interest was recorded in a CSV file and the raw ultrasound images were captured as PNG files. For the second set of three trials, the image centering module was activated, and the data was again recorded throughout the trajectory.

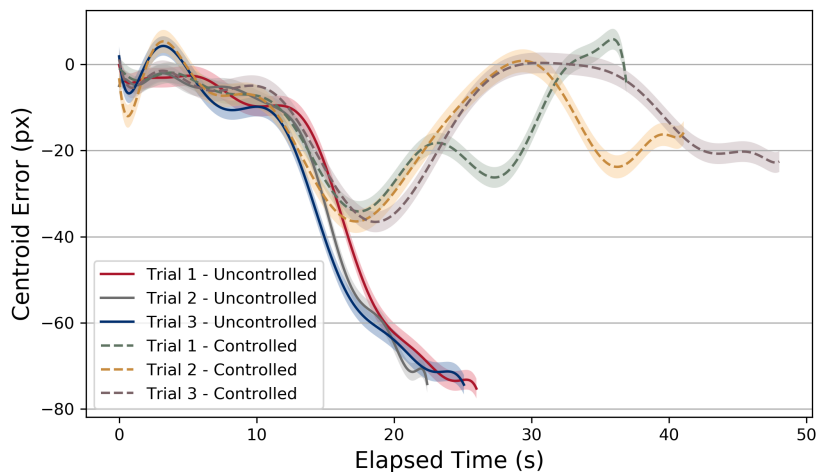


Figure 27: The measured distance of the centroid of the region of interest from the center of the image with and without image centering active.

Figure 27 shows how the image centering module kept the centroid of the region of interest at an average of -11 pixels away from the center of the image with a standard deviation of 11.6 pixels. However, when the module was inactive during the first three trials, the centroid of the region of interest moved away from the center of the image by an average of -70 pixels with a standard deviation of 1.7 pixels. Figure 28 shows the same trend as the region of interest remains centered throughout the entire path in the second three trials whereas it progressively moves to the left throughout the first three trials.

As part of the experimentation, the rate at which the image segmentation ran was moni-

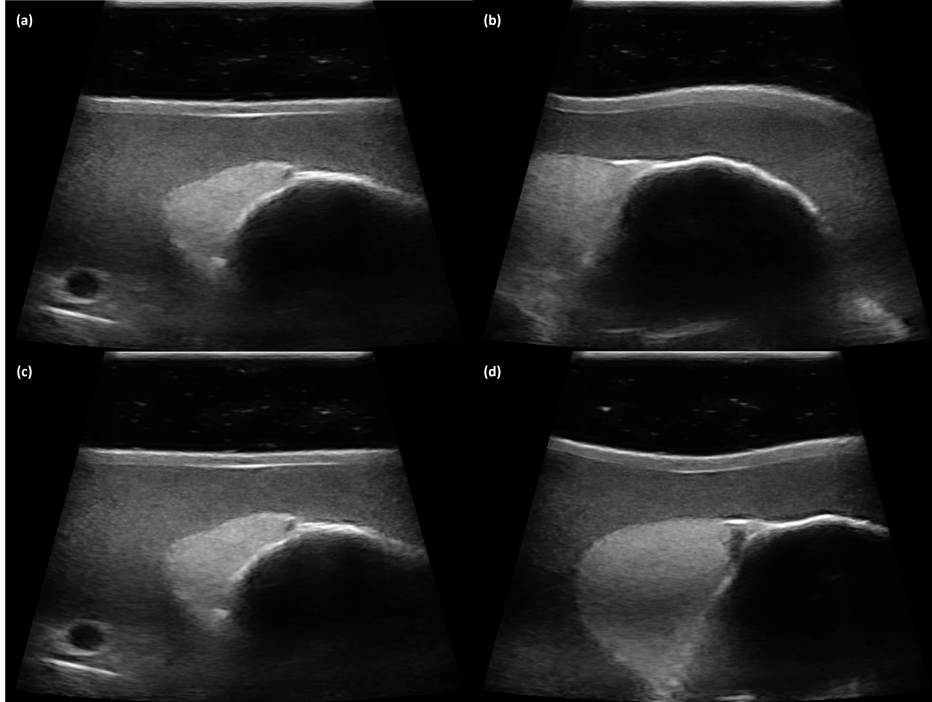


Figure 28: Ultrasound images captured at the start (a) and end (b) of the trajectory when image centering was not active and at the start (c) and end (d) of the trajectory when image centering was active.

tored throughout these tests. While in testing where the real-time segmentation module was the only module running, the segmentation ran speeds of up to 30 Hz. However, when the full system was activated, the performance slowed down to approximately 18 Hz with new ultrasound images being produced by the scanner at 25 Hz.

7.5 Accuracy of the Non-real Time Image Segmentation Framework

To verify the accuracy of the image segmentation, five trajectories were accomplished with the full *RASTER* system active and capturing images at each waypoint. Each waypoint image was then segmented both by hand and then by using the non-real time segmentation module. For each image, points were selected to create a polygon enclosing the thyroid in the image and the resulting mask from the selected points was saved locally as a *NumPy* array to be used as the ground truth. The result of the automatic segmentation of each image

was saved as a *NumPy* array and then compared against the ground truth segmentation for that image. The metric used to quantify the effectiveness of the image segmentation was the DICE score, calculated using Equation 6 where N_A is the number of pixels in mask A, N_B is the number of pixels in mask B, and N_C is the number of pixels common to each image.

$$DICE = \frac{2 * N_C}{N_A + N_B} \tag{6}$$

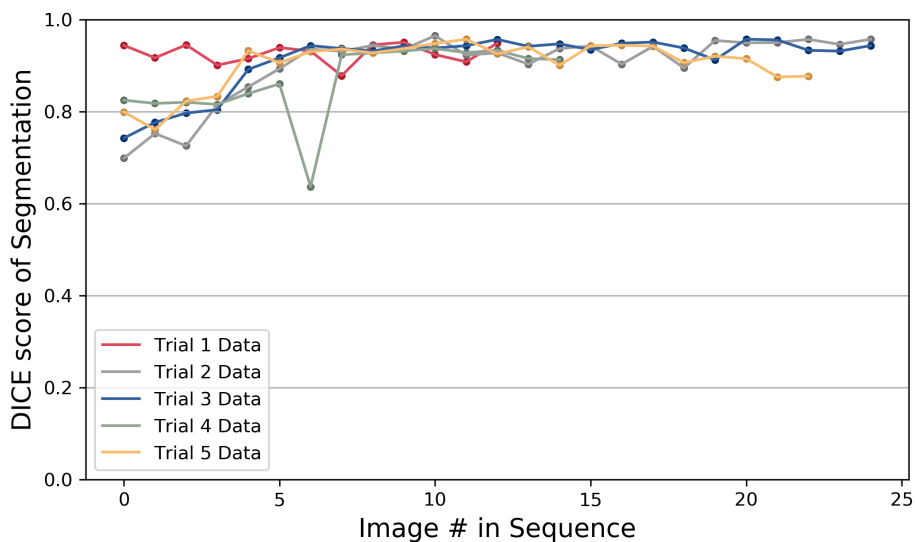


Figure 29: The calculated accuracy of the non-real-time segmentation throughout five trajectories.

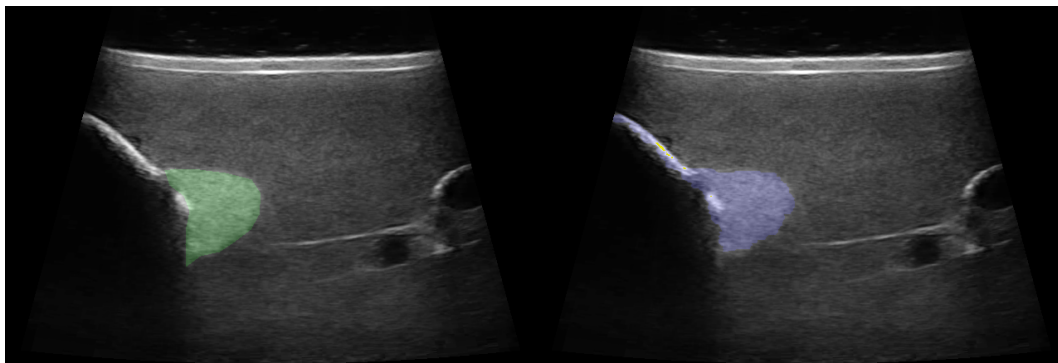


Figure 30: Comparison between ground truth segmentation (left) and automatic segmentation (right) early in the trajectory.

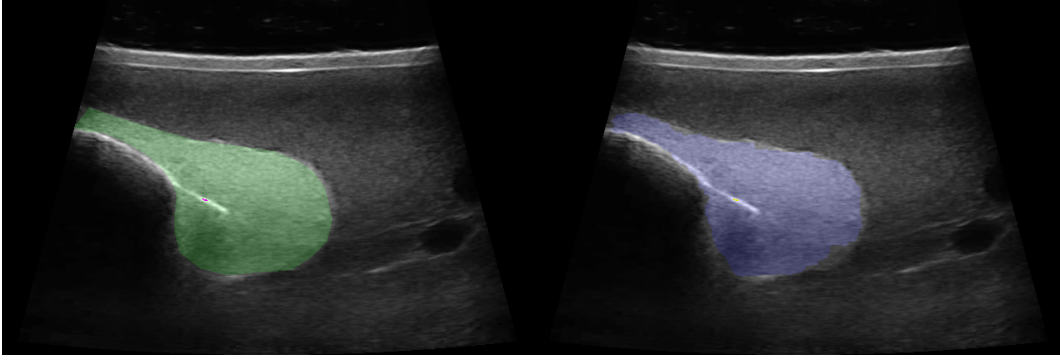


Figure 31: Comparison between ground truth segmentation (left) and automatic segmentation (right) in the middle of the trajectory.

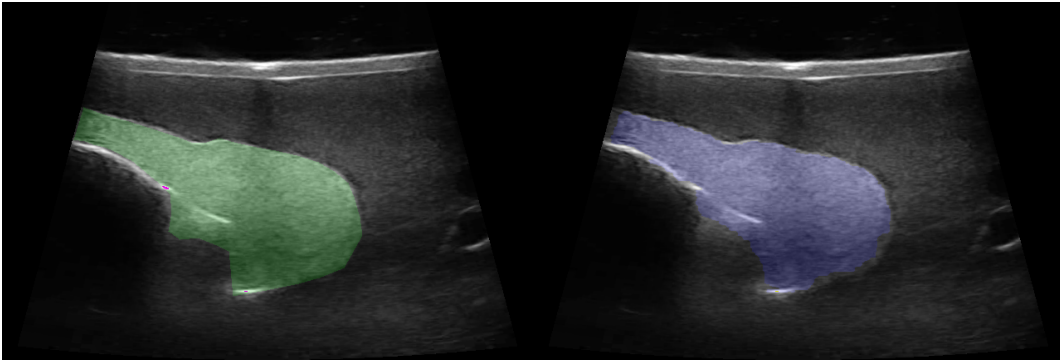


Figure 32: Comparison between ground truth segmentation (left) and automatic segmentation (right) late in the trajectory.

The average DICE score of the automatically segmented images was 0.904 with a standard deviation of 0.052. Figure 29 shows how the DICE score of the automatic segmentation improved over the trajectory and then reached a consistent level of accuracy. Figures 30, 31, 32 are provided as examples of the small differences seen between the automatic segmentation and the ground truth. These differences generally occur at the boundary of the thyroid or where the boundary of the thyroid is unclear.

7.6 Repeatability of the Volume Reconstruction Framework

To verify the repeatability of the volume reconstruction, an experiment was conducted that scanned the lower half of the right lobe of the thyroid in the phantom seven times. For these tests the full *RASTER* system was active throughout each trial. The data col-

lected from each trial was then sent through the non-real time segmentation and the volume reconstruction modules for analysis. For each test, the imaging planes were set to be one millimeter apart and the imaging depth was set to 5.2 centimeters.

Trial Number	Volume (mm^3)
1	4080
2	4240
3	4560
4	4360
5	4310
6	4550
7	4280

Table 2: The result of the volume estimation module over seven trials.

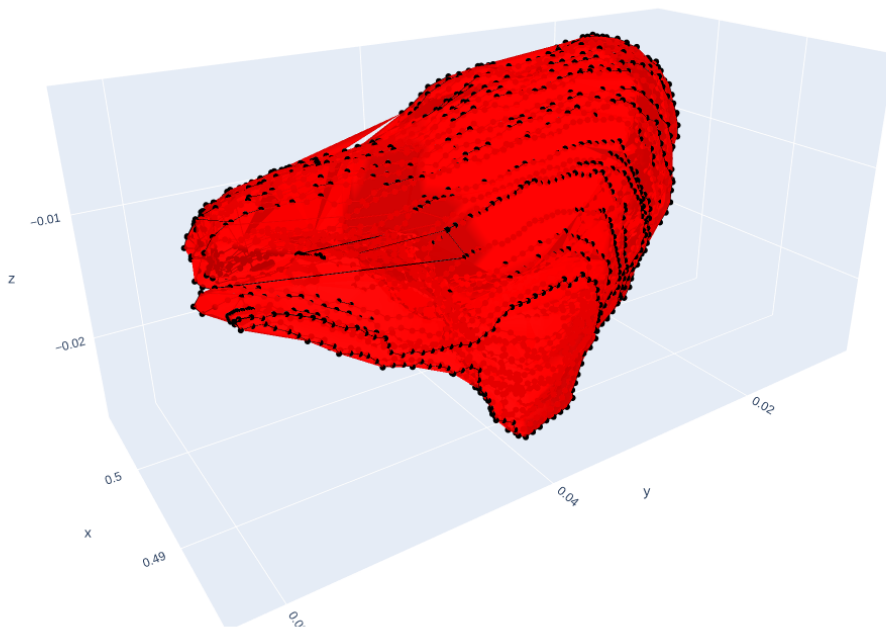


Figure 33: The visualization of a generated 3D volume.

Table 2 shows the results from these tests where the volume calculated by the module was on average 4340 cubic millimeters with a standard deviation of 170 cubic millimeters. Figure 33 shows an example of the visualization created from the data gathered in an exam.

8 Discussion

The successful completion of the experiments conducted on the thyroid ultrasound phantom demonstrate *RASTER*'s effectiveness. The force control algorithm applied a consistent force on the patient as the scanner moved over inconsistent and unknown patient anatomy. The trajectory generation procedure resulted in consistently spaced and parallel imaging planes that were not affected by patient anatomy or other control motions. The image balancing algorithm maintained proper contact with the patient throughout the trajectory despite no prior knowledge of the patient anatomy. The image centering algorithm ensured that the region of interest remained in the image despite unplanned movement in multiple axes and a planned trajectory that would have moved the region of interest out of view. *RASTER*'s non-machine learning image segmentation framework identified the region of interest selected by the operator with high accuracy throughout the trajectory with no further input from the operator. The volume generation module could repeatably estimate this region's volume and display it to the operator for further diagnosis. Combined these modules allowed *RASTER* to travel over unknown patient anatomy and create a smooth consistent segmentation for volume reconstruction. Additionally, through these experiments, *RASTER* was shown to perform in real-time with no prior knowledge of the anatomy or region of interest that was to be segmented.

While *RASTER* showed promising results overall, two limitations were found through the experimental process. First, while the use of a wireless scanner made the setup and overall operation of the system easier, the wireless nature of the scanner proved problematic. The scanner would overheat if used for extended periods of time and the battery of the scanner would need to be recharged almost as frequently increasing the number of times the scanner would have to be removed from the robot. Secondly, while the bright feature image balancing technique performed well on the ultrasound phantom, the system would not work directly on real patients as that bright feature that was approximated does not exist

in humans. However, the a sensing system such as the one proposed by Ma et al. [27] could be used to provide and even expand the capabilities of the system.

9 Conclusion

In this thesis a robot assistant for sonographic tissue examination and reconstruction (*RASTER*) was developed and validated. Experiments conducted with *RASTER* on a thyroid ultrasound phantom show that the system can, in real-time, accurately track regions of interest in an ultrasound image and connect those images with positional data to recreate 3D volumes.

This system’s objective was to help physicians and sonographers conduct thyroid ultrasound examinations in an outpatient setting with little disruption to the current procedure. The hardware requirements for *RASTER* are simple, requiring no permanent hardware in the exam room, fitting entirely on a mobile cart, utilizing mid-range computers, and integrating with existing ultrasound scanners already in use by physicians. The software is modular and lightweight, requiring standard computer power and built around an easy-to-use interface. The segmentation method is flexible and adaptable, requiring no foreknowledge of the region to be imaged before the exam starts.

The experiments demonstrated that each module of the *RASTER* system is critical to its overall performance. The force control, image balancing, and image centering modules all contributed to maintaining consistent patient contact and good ultrasound image quality throughout the examination while the trajectory generation module ensures adequate image spacing across unknown anatomy. The non-real time image segmentation framework accurately segmented the recorded ultrasound images. The volume reconstruction module repeatably estimated the volume of the thyroid and reconstructed a 3D shape using the robot’s pose information to display for the operator.

Future work on this project should be directed towards resolving the limitations mentioned above, improving the system’s hardware, and introducing additional control methods. This includes reducing the number of computing devices needed to operate the system to improve the portability and reduce the overall cost. Plus, giving the patient a direct method

of pausing the exam without fully emergency stopping the robot. Integrating a method of controlling the pitch angle of the scanner, possibly by analyzing the surface of the patient's skin around the scanner, and the yaw angle of the scanner, possibly by looking at patient features, should also be investigated to make a more complete control scheme.

An exploration of applications of *RASTER* beyond the thyroid should also be explored. As the experiments showed, *RASTER* does not need prior knowledge of the patient anatomy to maintain patient safety and conduct imaging trajectories. Furthermore, the non-machine learning segmentation model allows *RASTER* to segment any visually distinct feature in the ultrasound image without retraining. Combined with *RASTER*'s ability to work with any type of scanner, *RASTER* has the potential to examine other anatomical features on the body.

References

- [1] Cleveland Clinic. Thyroid: What It Is, Function, & Problems, June 2022. Last accessed 08 March 2024. my.clevelandclinic.org/health/body/23188-thyroid#anatomy.
- [2] Keyu Li, Yangxin Xu, and Max Q.-H. Meng. An Overview of Systems and Techniques for Autonomous Robotic Ultrasound Acquisitions. *IEEE Transactions on Medical Robotics and Bionics*, 3(2):510–524, May 2021. Conference Name: IEEE Transactions on Medical Robotics and Bionics.
- [3] Kim E. Barrett and William F. Ganong. *Ganong’s review of medical physiology*. McGraw-Hill Medical, New York, 23rd ed. edition, 2010. OCLC: 430823856.
- [4] Yasaman Pirahanchi, Muhammad Ali Tariq, and Ishwarlal Jialal. Physiology, Thyroid. In *StatPearls*. StatPearls Publishing, Treasure Island (FL), 2024.
- [5] Melanie Lyden, Tracy Wang, and Julie Sosa. Surgical anatomy of the thyroid gland. In TW Post, editor, *UpToDate*. UpToDate, June 2023.
- [6] Diana S. Dean and Hossein Gharib. Epidemiology of thyroid nodules. *Best Practice & Research Clinical Endocrinology & Metabolism*, 22(6):901–911, December 2008.
- [7] S. Guth, U. Theune, J. Aberle, A. Galach, and C. M. Bamberger. Very high prevalence of thyroid nodules detected by high frequency (13 MHz) ultrasound examination. *European Journal of Clinical Investigation*, 39(8):699–706, 2009. .eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1365-2362.2009.02162.x>.
- [8] Ravi Kant, Amanda Davis, and Vipin Verma. Thyroid Nodules: Advances in Evaluation and Management. *American Family Physician*, 102(5):298–304, September 2020.
- [9] Cosimo Durante, Giuseppe Costante, Giuseppe Lucisano, Rocco Bruno, Domenico Meringolo, Alessandra Paciaroni, Efisio Puxeddu, Massimo Torlontano, Salvatore Tu-

- mino, Marco Attard, Livia Lamartina, Antonio Nicolucci, and Sebastiano Filetti. The natural history of benign thyroid nodules. *JAMA*, 313(9):926–935, March 2015.
- [10] Louise Davies, Michelle Ouellette, Mark Hunter, and H. Gilbert Welch. The increasing incidence of small thyroid cancers: Where are the cases coming from? *The Laryngoscope*, 120(12):2446–2451, 2010. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/lary.21076>.
- [11] Sarah B. Fisher and Nancy D. Perrier. The incidental thyroid nodule. *CA: a cancer journal for clinicians*, 68(2):97–105, March 2018.
- [12] P W Wiest, M F Hartshorne, P D Inskip, L A Crooks, B S Vela, R J Telepak, M R Williamson, R Blumhardt, J M Bauman, and M Tekkel. Thyroid palpation versus high-resolution thyroid ultrasonography in the detection of nodules. *Journal of Ultrasound in Medicine*, 17(8):487–496, 1998. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.7863/jum.1998.17.8.487>.
- [13] Bryan R. Haugen, Erik K. Alexander, Keith C. Bible, Gerard M. Doherty, Susan J. Mandel, Yuri E. Nikiforov, Furio Pacini, Gregory W. Randolph, Anna M. Sawka, Martin Schlumberger, Kathryn G. Schuff, Steven I. Sherman, Julie Ann Sosa, David L. Steward, R. Michael Tuttle, and Leonard Wartofsky. 2015 American Thyroid Association Management Guidelines for Adult Patients with Thyroid Nodules and Differentiated Thyroid Cancer: The American Thyroid Association Guidelines Task Force on Thyroid Nodules and Differentiated Thyroid Cancer. *Thyroid*, 26(1):1–133, January 2016.
- [14] Domenico Salvatore, Terry F. Davies, Martin-Jean Schlumberger, Ian D. Hay, and P. Reed Larsen. *Thyroid Physiology and Diagnostic Evaluation of Patients With Thyroid Disorders*, page 355. Elsevier, 13th edition edition, 2016.

- [15] Malcolm C. Brown and Ralph Spencer. Thyroid Gland Volume Estimated by Use of Ultrasound In Addition to Scintigraphy. *Acta Radiologica: Oncology, Radiation, Physics, Biology*, 17(4):337–341, January 1978.
- [16] J. Brunn, U. Block, G. Ruf, I. Bos, W. P. Kunze, and P. C. Scriba. Volumetrie der Schilddrüsenlappen mittels Real-time-Sonographie * ¹. *DMW - Deutsche Medizinische Wochenschrift*, 106(41):1338–1340, March 2008.
- [17] Wael Shabana, Els Peeters, and Michel De Maeseneer. Measuring Thyroid Gland Volume: Should We Change the Correction Factor? *American Journal of Roentgenology*, 186(1):234–236, January 2006.
- [18] Nicholas J Peterman, Eunhae Yeo, Brad Kaptur, Emily J Smith, Anton Christensen, Edward Huang, and Mehmoodur Rasheed. Analysis of Rural Disparities in Ultrasound Access. *Cureus*, May 2022.
- [19] Wei Wang, Mathew Leonardi, and George Condous. The importance of obstetric and gynaecologic sonographer health and safety. *Australasian Journal of Ultrasound in Medicine*, 21(4):198–200, November 2018.
- [20] Murat Faik Erdogan, Alptekin GURSOY, and Gurbuz Erdogan. Natural course of benign thyroid nodules in a moderately iodine-deficient area. *Clinical Endocrinology*, 65(6):767–771, December 2006.
- [21] E. Papini, L. Petrucci, R. Guglielmi, C. Panunzi, R. Rinaldi, V. Bacci, A. Crescenzi, F. Nardi, R. Fabbrini, and C. M. Pacella. Long-Term Changes in Nodular Goiter: A 5-Year Prospective Randomized Trial of Levothyroxine Suppressive Therapy for Benign Cold Thyroid Nodules. *The Journal of Clinical Endocrinology & Metabolism*, 83(3):780–783, March 1998.

- [22] Felix von Haxthausen, Sven Böttger, Daniel Wulff, Jannis Hagenah, Verónica García-Vázquez, and Svenja Ipsen. Medical Robotics for Ultrasound Imaging: Current Systems and Future Trends. *Current Robotics Reports*, 2(1):55–71, March 2021.
- [23] Zhongliang Jiang, Septimiu E. Salcudean, and Nassir Navab. Robotic ultrasound imaging: State-of-the-art and future perspectives. *Medical Image Analysis*, 89:102878, October 2023.
- [24] Monica Georgescu, Arnaud Saccomandi, Bernard Baudron, and Philippe L. Arbeille. Remote Sonography in Routine Clinical Practice Between Two Isolated Medical Centers and the University Hospital Using a Robotic Arm: A 1-Year Study. *Telemedicine and e-Health*, 22(4):276–281, April 2016.
- [25] Wenli Jiang, Xia Zhao, Tian Gan, Ying Liu, Shuilian Luo, Meifang Zhu, Sikai Chen, Yu Jiang, and Meng Wu. Application of a Tele-Ultrasound Robot During COVID-19 Pandemic: A Feasibility Study. *Journal of Ultrasound in Medicine*, 42(3):595–601, February 2023.
- [26] Alexandre Krupa, David Folio, Cyril Novales, Pierre Vieyres, and Tao Li. Robotized Tele-Echography: An Assisting Visibility Tool to Support Expert Diagnostic. *IEEE Systems Journal*, 10(3):974–983, September 2016. Conference Name: IEEE Systems Journal.
- [27] Xihan Ma, Wen-Yi Kuo, Kehan Yang, Ashiqur Rahaman, and Haichong K. Zhang. A-SEE: Active-Sensing End-Effector Enabled Probe Self-Normal-Positioning for Robotic Ultrasound Imaging Applications. *IEEE Robotics and Automation Letters*, 7(4):12475–12482, October 2022.
- [28] Christoph Hennersperger, Bernhard Fuerst, Salvatore Virga, Oliver Zettinig, Benjamin Frisch, Thomas Neff, and Nassir Navab. Towards MRI-Based Autonomous Robotic

- US Acquisitions: A First Feasibility Study. *IEEE Transactions on Medical Imaging*, 36(2):538–548, February 2017.
- [29] Qinghua Huang, Jiulong Lan, and Xuelong Li. Robotic Arm Based Automatic Ultrasound Scanning for Three-Dimensional Imaging. *IEEE Transactions on Industrial Informatics*, 15(2):1173–1182, February 2019.
- [30] Jiyong Tan, Bing Li, Yuanwei Li, Bin Li, Xinxing Chen, Jiayi Wu, Baoming Luo, Yuquan Leng, Yiming Rong, and Chenglong Fu. A Flexible and Fully Autonomous Breast Ultrasound Scanning System. *IEEE Transactions on Automation Science and Engineering*, pages 1–14, 2022. Conference Name: IEEE Transactions on Automation Science and Engineering.
- [31] Jingyu Zhang, Yunjiang Wang, Tianjian Liu, Keji Yang, and Haoran Jin. A Flexible Ultrasound Scanning System for Minimally Invasive Spinal Surgery Navigation. *IEEE Transactions on Medical Robotics and Bionics*, 3(2):426–435, May 2021. Conference Name: IEEE Transactions on Medical Robotics and Bionics.
- [32] Juan Sandoval, Med Amine Laribi, Saïd Zeghloul, Marc Arsicault, and Jean-Michel Guilhem. Cobot with Prismatic Compliant Joint Intended for Doppler Sonography. *Robotics*, 9(1):14, March 2020. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.
- [33] Junying Chen, Haijun You, and Kai Li. A review of thyroid gland segmentation and thyroid nodule segmentation methods for medical ultrasound images. *Computer Methods and Programs in Biomedicine*, 185:105329, March 2020.
- [34] T.F. Chan and L.A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, February 2001.

- [35] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *Proceedings of IEEE International Conference on Computer Vision*, pages 694–699, Cambridge, MA, USA, 1995. IEEE Comput. Soc. Press.
- [36] S. Lankton and A. Tannenbaum. Localizing Region-Based Active Contours. *IEEE Transactions on Image Processing*, 17(11):2029–2039, November 2008.
- [37] Chunming Li, Chenyang Xu, Changfeng Gui, and Martin D Fox. Distance Regularized Level Set Evolution and Its Application to Image Segmentation. *IEEE Transactions on Image Processing*, 19(12):3243–3254, December 2010.
- [38] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. ”GrabCut”: interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, August 2004.
- [39] Nikhil S. Narayan, Pina Marziliano, Jeevendra Kanagalingam, and Christopher G. L. Hobbs. Speckle Patch Similarity for Echogenicity-Based Multiorgan Segmentation in Ultrasound Images of the Thyroid Gland. *IEEE Journal of Biomedical and Health Informatics*, 21(1):172–183, January 2017.
- [40] Prabal Poudel, Alfredo Illanes, Debdoot Sheet, and Michael Friebe. Evaluation of Commonly Used Algorithms for Thyroid Ultrasound Images Segmentation and Improvement Using Machine Learning Approaches. *Journal of Healthcare Engineering*, 2018:1–13, September 2018.
- [41] Jinlian Ma, Fa Wu, Tian’an Jiang, Qiyu Zhao, and Dexing Kong. Ultrasound image-based thyroid nodule automatic segmentation using convolutional neural networks. *International Journal of Computer Assisted Radiology and Surgery*, 12(11):1895–1910, November 2017.
- [42] Y.Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *Proceedings Eighth IEEE International Con-*

ference on Computer Vision. ICCV 2001, volume 1, pages 105–112, Vancouver, BC, Canada, 2001. IEEE Comput. Soc.