AN AUGMENTED REALITY APPLICATION FOR VISUALIZING ANATOMICAL DATA

A Thesis Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Mohammad Mainul Islam

May 2017

AN AUGMENTED REALITY APPLICATION FOR VISUALIZING ANATOMICAL DATA

Mohammad Mainul Islam

APPROVED:

Ioannis A. Kakadiaris, Ph.D., Chairman
Dept. of Computer Science

Zhigang Deng, Ph.D.
Dept. of Computer Science

Christophoros Nikou, Ph.D.
Dept. of Computer Science and Engineering
University of Ioannina

Dean, College of Natural Sciences and Mathematics

AN AUGMENTED REALITY APPLICATION FOR VISUALIZING ANATOMICAL DATA

An Abstract of a Thesis

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Mohammad Mainul Islam

May 2017

Abstract

Augmented Reality can be useful in medical science because the visualization of a patient's internal structure is important for planning an intervention. Traditional display technologies present images collected from CT, MRI or ultrasound on 2D screens. However, aligning those images to a patient's body helps physicians decide the target of the intervention. In this thesis, an iPad application is proposed, which overlays 3D models of a patient's organs acquired using MRI scan in real-time video.

The process is divided into two parts: 1) 3D registration and 2) tracking. The registration consists of acquiring the 3D structure of patient's torso using a depth sensor and overlaying pre-operative 3D models in real-time video. This is accomplished by a 3D point-to-point registration using the Iterative Closest Point algorithm. The tracking keeps the 3D models aligned on the patient's body by detecting the camera's pose. This is accomplished by using a technique called Simultaneous Localization and Mapping, which maps the environment and estimates the camera pose continuously. Registration accuracy is measured based on a study of eight different users of the application. The possibility of porting the application to a head-mounted display or glasses is also explored.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Augmented Reality

Augmented and virtual reality became favorable topics of research due to recent advancements in head-mounted display technologies. While virtual reality completely changes the user's perception replacing the user's world with an entirely new environment, augmented reality modifies the user's world and tries to incorporate virtual objects allowing users to feel as though the objects are part of the actual scene. Although augmented reality can be achieved in many ways such as targeting sound, smell, touch, and so on, visual augmentation is primarily used in applications. The application of augmented reality ranges from adding game characters into the world to overlaying information by detecting local landmarks, changing dress color in dressing room mirror, adding virtual furniture into a room, and so on.

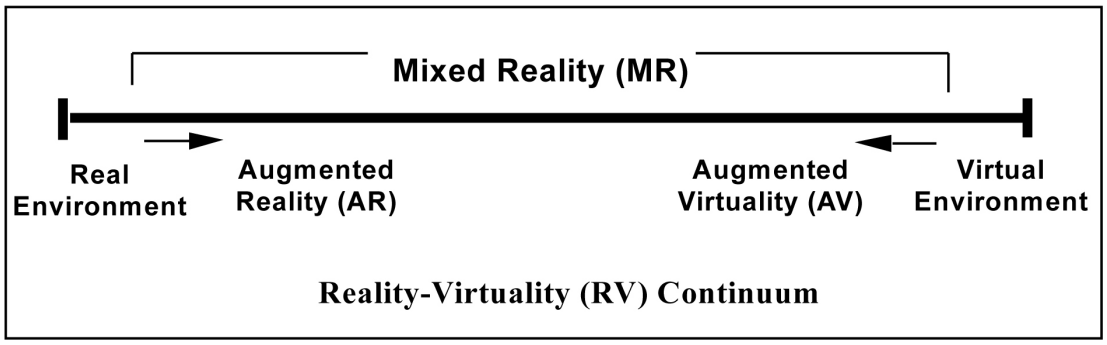

Figure 1.1: Milgram's reality-virtuality continuum [33].

As a formal definition, Augmented Reality (AR) is a technique that allows seamless overlaying of virtual objects over the physical world in real-time. The term Augmented Reality is defined by Milgram [33] as existing on a continuum from a real to a virtual environment where one end is the reality, and the other end is completely virtual.

In recent years, some manufacturers have started to develop head-mounted display (HMD) or glasses that mainly incorporate digital displays in optical glasses. The Microsoft HoloLens [7] is one such HMD technology. Additionally, see-through video technique is being used in AR that incorporates virtual objects into a real-time video. Mobile phones fit into this category, and recent advancements in mobile computing power make mobile phones a good platform for AR applications. Until now, the applications of AR were mainly limited to the entertainment and location services through mobile phones. Pokemon GO, a game from Niantic Inc. [12], has created a large market for AR. The future of AR is not only in the entertainment industry, but it is also being spread into real estate, fashion, manufacturing, and in medical science, the field of the focus of this thesis.

AR in medical science: One of the primary uses of AR in the field of medical science could be live, interactive imaging software for assisting physicians or teaching anatomy to medical students and children. With AR projections of bones, muscles, and other internal body parts, medical students could practice anatomy in the physical world. Another good example of AR in medical science could be the automatic retrieval of patient information and histories; physicians could just wear an HMD or glasses and see the patient. Based on face recognition, it overlays the patient's information from a database.

Compared to the almost two-decade-long development of AR technologies in large industry corporations such as the auto industry (Mercedes-Benz) and aircraft industry (Boeing), medical AR has been around for only a decade. Medical device manufacturers and researchers have started developing AR devices, although most of them are still at the research level. For example, Evena Medical Inc. [4] has developed AR glasses that use ultrasound to see veins beneath the skin. Although it has a limitation that it can detect veins only a few millimeters below the skin, it is very similar to the work in this thesis. Figure 1.2 depicts how these glasses are used to see veins through the skin.

Medsights Tech [9] developed another application that also uses ultrasound to reconstruct 3D images of tumors and superimpose them onto the patient's body in real-time video. Although the product has not yet been release, the technology is compatible with mobile phones, laptops, computers, and AR glasses. The common technique in both technologies mentioned above is that they both require the ultrasound for 3D reconstruction of internal organs.

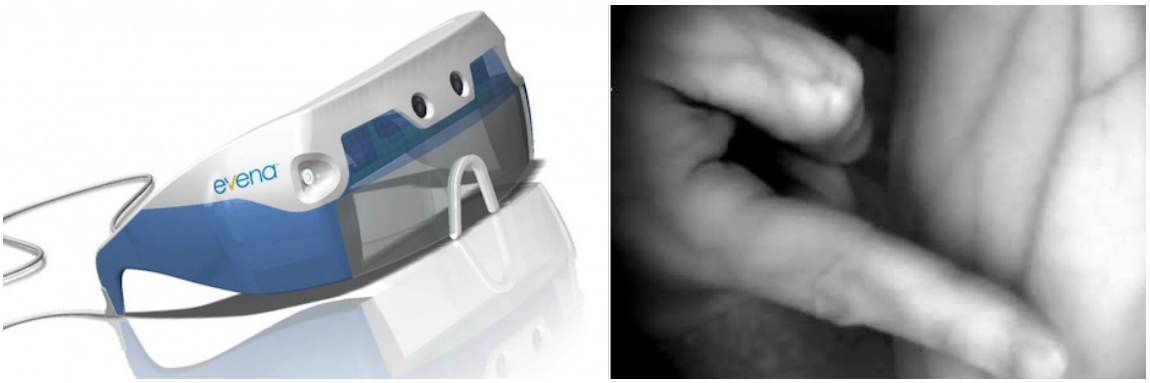

Figure 1.2: Eye-on glasses to see veins by Evena Medical Inc. [4]. (L) Depicted is the glasses and (R) Depicted is the visualization of veins.

The project of this thesis has developed a light AR system named **iRay** [30] that does not rely on ultrasound, but requires pre-operative images acquired by CT or MRI scans. In traditional diagnostics, physicians rely on 2D slices from CT or MRI scans. It is a very common procedure, and almost every patient has to go through that scan before any intervention. Therefore, there is almost no extra cost to generate 3D models of a patient's organs to use in this AR application. 3D reconstruction of the outer surface of the patient's body is necessary for accurate registration of virtual models in a real scene. The recent development of depth sensors made this easy and provided even better 3D registration in the real world.

1.2 Goal and Tasks

The main goal of this thesis is to: **Design and develop a prototype of an augmented reality application for superimposing pre-operative anatomical**

data on a patient's body in a real-time video. Following are the tasks that need to be accomplished in this thesis to support the main goal.

1. Generate and visualize pre-operative models of patient's organs.

   Task 1.a: Segment target organs from the medical scan and generate 3D models.

   Task 1.b: Visualize generated 3D models or render volume from the medical scan.

2. Design user interface and database systems.

   Task 2.a: Design user interface and database system in a mobile application.

   Task 2.b: Design a visual guidance system to reduce user subjective error.

3. Register pre-operative models in the 3D structure of patient's body.

   Task 3.a: Generate 3D model of patient's torso using a depth sensor.

   Task 3.b: Register pre-operative models on patient's body.

   Task 3.c: Track camera pose and render augmentation.

4. Measure registration and tracking error.

   Task 4.a: Measure 3D-to-3D registration error.

   Task 4.b: Measure tracking error using fiducial markers.

5. Perform a feasibility analysis on porting the iRay to different AR glasses/HMDs.

1.3 Related Work

Although no any medical equipment is available to visualize a patient's organs superimposed in real-time video, some research works were found that are closely related to this thesis. In 2007, Christoph Bichlmeier et al. used a stereoscopic video see-through head-mounted display to visualize pre-operative volumetric data of an anatomical region of interest [18]. Their primary focus was improving depth perception with alpha blending between the virtual model and skin at the edges. They used an infrared camera to track surgical instruments, and implanted markers before a CT scan to register virtual models. In another work [19], they used a display system instead of an HMD and a depth camera (Kinect) for tracking.

Su Li-Ming et al. published their work in 2009 on accurate real-time stereo-endoscopic visualization of a renal tumor during surgery [41]. Their work is very similar to this thesis, but they used manual registration of pre-operative data in stereoscopic video. The system further refines the registration using a modified 3D-to-3D iterative closest point algorithm, which only considers visible points for registration. Their tracking system used an automatic registration algorithm for which several points on the kidney surface surrounding the renal mass were selected as fixed reference points by a human operator.

In 2014, Haouchine Nazim et al. published a work [28], which is more advanced and can track the internal structure of liver based on a biomechanical model of the liver. Its specialty is that it can track the vascular network, tumors, and cut planes even after large deformation in liver structure. They also used a stereo-based tracking

algorithm, and their accuracy of an implanted artificial tumor is less than 6 *mm*.

A common feature used in most of the works mentioned above is that they used stereo-camera for tracking, whereas this project uses a depth camera for tracking. Depth cameras are very new, and tracking using depth is better than the stereo-camera-based tracking. Moreover, previously mentioned works focused on a particular organ; trying to visualize a small internal property of that organ such as a tumor. This project does not focus on a particular organ but visualizes multiple organs.

The most similar work to this project is SurgeryPad from the German Cancer Research Center [5], which also visualized internal structures of the human body in the iPad video. However, its technology is marker-based; it needs to implant needles with a cap on the body during intervention as well as the CT scan. Moreover, it does not have a tracking system. Therefore, the operator needs to hold the iPad steady, and there are frequent drifts. Additionally, one more drawback is that it sends iPad video to a workstation via Wi-Fi that processes and adds virtual objects and sends it back to the iPad for visualization. The project iRay does not rely on an external workstation; all process of registration and tracking are within the iPad without the help of any marker.

1.4 Accomplishments

Three-dimensional models of a patient's organs are generated from MRI data to use in this application. The main goal is to render these 3D models in a real-time video so that the models are superimposed onto the patient's body in the position

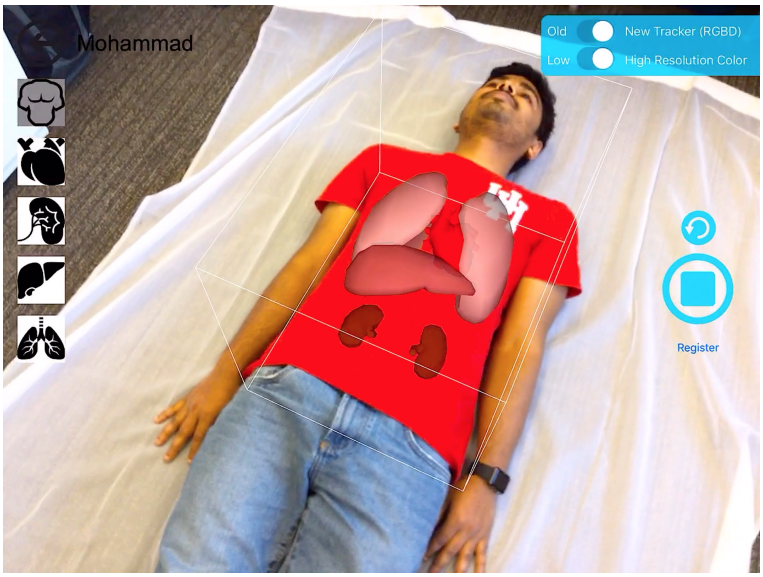

Figure 1.3: iRay application preview. Virtual anatomical models are overlaid on a patient's body in real-time video.

where the organs should be in the real world. Pre-operative 3D models and patient databases are stored on a remote server, but the local file directory in the application can be synchronized whenever necessary. The application requires a 3D sensor to reconstruct the outer surface of the patient's body. This reconstruction is required because the goal is to detect the real position of the patient's torso and appropriately register pre-operative 3D models onto the torso. Figure 1.3 depicts the visual output of the iRay application. Following is a description of accomplished tasks described in section 1.2.

Task 1.a: Torso and aorta are segmented from MRI scan of a male volunteer, and corresponding 3D models are created.

Task 1.b: Segmented 3D models are visualized in the mobile screen using OpenGL. A direct volume rendering based visualization is also explored.

Task 2.a An online database and repository system is developed for storing the list of patients and their data files. A fully functional iPad application is developed with a user interface to synchronize the local database with the online database and download pre-operative data.

Task 2.b: A visual guidance system is developed to assist users in selecting a scan area that achieves more accurate registration. A user study is performed on several users to validate the guidance system.

Task 3.b: Iterative Closest Points algorithm is used to register pre-operative 3D models onto the 3D scan of the patient's body. Scanning (Task 3.a) and tracking (Task 3.c) are accomplished with the help of a depth sensor, and its SDK.

Task 4: Both the initial registration error and tracking error are measured.

Task 5: A feasibility analysis is performed on three AR glasses/HMDs, and one (HoloLens) is chosen as a compatible device. Porting the iRay application to the HoloLens is also explored.

1.5 Publications

1. T. Xie, M. M. Islam, A. B. Lumsden, and I. A. Kakadiaris. Semi-automatic initial registration for the iRay system: A user study. In Proc. 4^{th} International Conference on Augmented Reality, Virtual Reality and Computer Graphics, Ugento, Lecce, Italy, June 12-15 2017. (In Press)

2. I. A. Kakadiaris, M. M. Islam, T. Xie, C. Nikou, and A. B. Lumsden. iRay: Mobile AR using structure sensor. In Proc. 15^{th} IEEE International Symposium on Mixed and Augmented Reality, Merida, Mexico, Sept. 19-23 2016.

1.6 Thesis Outline

The remainder of the thesis is organized as follows: Chapter 2 describes some technologies used in this thesis including depth sensors, medical imaging, and registration and tracking techniques. Chapter 3 includes implementation details of some tasks in this thesis. Experiments and results including limitations are described in Chapter 4. A feasibility analysis on porting the iRay application to AR glasses or HMDs is described in Chapter 5. Finally, Chapter 6 concludes the thesis describing some possible future works.

Chapter 2

Background

2.1 Medical Imaging Techniques

Different medical imaging techniques have been used by physicians to investigate the causes of illness for a long time. These images reflect the internal structure of a patient's specific organ or a portion of the body. The most common forms of medical images are X-rays, CT scans, and MRIs. X-rays (radiographs) are one of the most used medical imaging techniques, which sends electromagnetic waves (radiation) through the body, exposing a film to reflect the internal structure.

Computed Tomography (CT) is a more modern imaging technique that combines many X-rays to produce a more detailed, cross-sectional images of the body. While the patient lies on a table inside the CT scanner, an X-ray tube rotates around the patient, taking images from all directions. These images are combined to generate

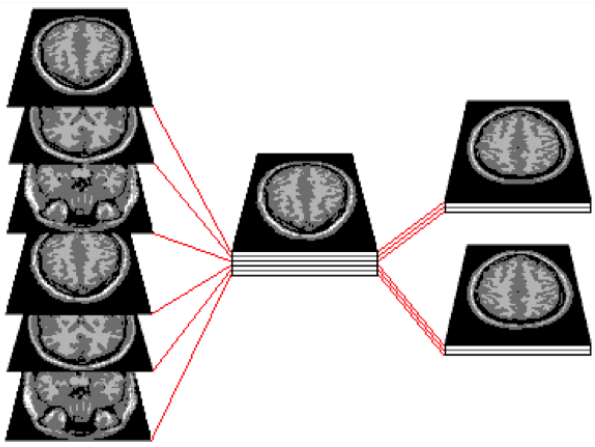

Figure 2.1: Sample DICOM slices [2]. An anatomical region of interest divided into slices is depicted.

cross-section slices of the patient's body. Magnetic resonance imaging (MRI), works without radiation while still producing similar cross-section images of the patient's body. The MRI machine uses magnetic fields and radio waves to resonate tissues. A computer records the rate at which various parts of the body vibrate and translates the data into two-dimensional images.

The generated cross-section images are called Digital Imaging and Communications in Medicine (DICOM), which is a global standard for medical images and related information. The DICOM standard, published in 1993, is one of the most widely used health-care messaging systems in the world. National Electrical Manufacturers Association (NEMA) [11] holds the copyright to this standard. DICOM consists of around few hundred 2D images called slices. Each slice contains around 512x512 grayscale values, that present the intensity of the tissue in the corresponding position. This scalar intensity value ranges from zero to a few thousand. Figure 2.1 depicts how a portion of the body is divided into 2D DICOM slices [2]. Converting

the slices to grayscale images is easy for computer software, and physicians frequently use 2D images for diagnosis. However, for real-time overlaying on a patient's body, 2D images are not sufficient, and 3D models of target organs are required.

2.2 Depth Sensors

Structure from Motion (SFM) is a technique for generating 3D structures from 2D images. SFM is a technique for creating 3D structures, which is similar to reconstruction using stereo vision in that, correspondence between images needs to be found for both. This correspondence involves computer vision algorithms for extracting and matching image features. However, recent advances in depth sensors have created a new way of generating 3D structures using depth instead of image feature correspondence. Since depth-sensor-based reconstruction is more accurate than SFM, many companies have started manufacturing such scanning devices. This type of scanner can be used in augmented reality because it can map the environment and track the camera pose more accurately. Many tech giants such as Google, Microsoft, and Intel have already developed such sensors or head-mounted displays. The following are few examples of depth sensors or spatial mapping technology already available for consumers.

Google Tango: Google has developed an augmented reality technology [6] for Android phones, which has features such as area learning (mapping surroundings), motion tracking, depth perception and overlaying virtual objects on the real world using multiple cameras.

Intel RealSense: Intel has manufactured a few infrared-based depth sensors [8] that are being used for 3D scanning and user interface interaction. They have released several versions including R200, SR300, and a cross-platform API for Windows, Mac, and Linux. Intel's SDK includes features such as 3D scanning, hand gesture recognition, and user interface interaction.

Microsoft Kinect: Microsoft has developed a depth-sensing camera named Kinect for its popular gaming console Xbox. Kinect also uses infrared for depth sensing and includes an RGB camera for facial tracking or other 2D image processing. Kinect is mainly produced for body and gesture tracking or voice recognition in Xbox, but it has been used in research extensively, especially for 3D reconstruction and mapping [38] of a rigid scene or even a non-rigid scene [37].

Structure Sensor: The operating system of this project is the iOS for iPad, and unfortunately, there is no device currently available from Apple for depth sensing or 3D reconstruction. Fortunately, another startup company has produced such a sensor for iOS devices only. It is easy to use and highly accurate in 3D mapping. This sensor is the Structure Sensor by Occipital [13], which is the main hardware used in this project along with an iPad. The device is a depth sensor-based 3D scanner for iPhone and iPad. Recently, Occipital has also released a mixed-reality SDK named Bridge and a kit to mount an iPhone for an AR or VR experience. Their SDK provides features such as texturized 3D model generation and export, raw depth and normal buffer access, OpenGL-based visualization of camera frames with a 3D scan overlay, and a very accurate Simultaneous Localization and Mapping (SLAM) algorithm for tracking. Figure 2.2 depicts a Structure Sensor.

Figure 2.2: Structure Sensor by Occipital [13] Inc. The device is 11.92 *cm* long.

2.3 3D Registration Technique

The registration technique used in this thesis is called an Iterative Closest Point (ICP) algorithm, which is a well-established algorithm in robotics for 2D and 3D point cloud registration. The goal is to register one point cloud to another fixed point cloud and find the best transformation with the least errors between the two point clouds. The fixed point cloud is called the reference, and the other is called the source. This transformation is rigid; the shapes of each point cloud are kept intact without any deformation. ICP is an iterative algorithm; in each iteration, a transformation between the reference and the last state of the source is estimated. This estimation is done in a manner that minimizes the overall distance between the two point clouds.

There are some variants of the ICP algorithm and its distance function. The most common variants of the algorithm are point-to-point and point-to-plane. In point-to-point, the distance or error between the two point clouds is the average distance between closest points pairwise. In point-to-plane, the average distance is calculated by the distance from the points in one cloud to the nearest surface on another cloud. Iteration continues until this error reaches a local minimum or is

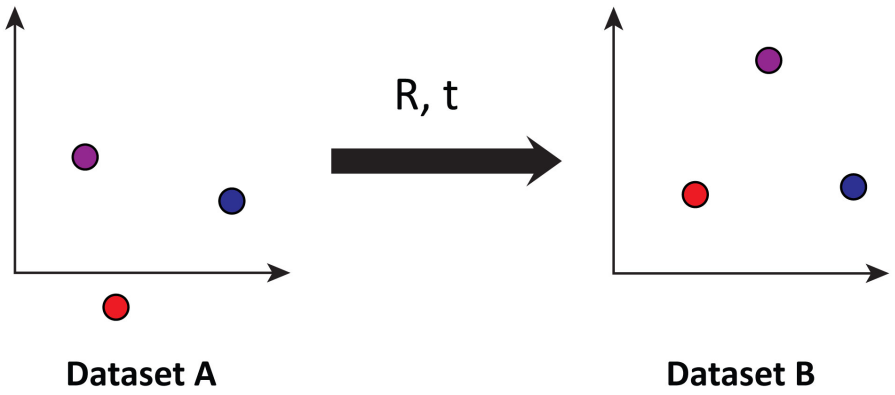

Figure 2.3: Source and reference point sets in ICP. Source dataset A needs to be registered to reference dataset B.

lower than a threshold. The initial pose of the two point clouds should be as similar as possible. Otherwise, ICP will converge to a local minimum for which registration is not perfect. A simple point-to-point ICP registration process is described below; however, a more robust ICP implementation [27] was used in this project.

For the two point clouds, A and B, we need to find the optimal transformation (rotation R and translation t), so that cloud A aligns to cloud B. Therefore, the transformation will be:

$$B = RA + t$$

To find the optimal rigid transformation, the centroid of two point clouds is first calculated, and both clouds are transformed to the origin. This step eliminates the translation estimation and focuses only on estimating the rotation. If P_A and P_B are the point sets in cloud A and B respectively, centroids of those point sets are:

$$C_A = \frac{1}{N_A} \sum_{i=1}^{N_A} {P_A}^i$$

$$C_B = \frac{1}{N_B} \sum_{i=1}^{N_B} {P_B}^i$$

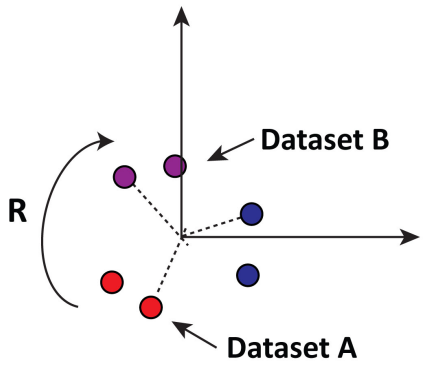

Figure 2.4: Datasets A and B transformed to the origin, by subtracting their centroid from their vertices.

Matrix M_A and M_B are generated from the two point sets after transforming to the origin. These two matrices are multiplied to generate a matrix H, which is passed to Singular Value Decomposition (SVD). From the components from SVD, rotation R is estimated.

$$M_A = \sum_{i=1}^{N_A} ({P_A}^i - C_A)$$

$$M_B = \sum_{i=1}^{N_B} ({P_B}^i - C_B)$$

$$H = M_A M_B$$

$$[U, S, V] = SVD(H)$$

$$R = VU^T$$

It is a single step in the iteration, and the algorithm continues until there is no change in error or the iteration reaches a maximum limit. The error is the average Euclidian distance between each point in cloud A and the corresponding nearest point in cloud B. The nearest point is selected using a k-d tree algorithm. In each iteration, rotation R is estimated, and the translation is then calculated using rotation and centroid.

$$t = -RC_A + C_B$$

2.4 Tracking Techniques

Tracking is the main feature of augmented reality. Whenever the camera or the person wearing the HMD moves, the virtual objects have to be repositioned accordingly so that they are kept in the same position in the real world. Even a small drift or displacement will cause the user to notice that virtual objects are not a part of the reality. Initial research in augmented reality was mainly on marker-based tracking in which a known pattern of texture is tracked continuously in video frames. However, using markers for medical applications requires the markers to be implanted in the patient's body during a CT/MRI scan as well as during the intervention.

An alternative technique for camera tracking is Simultaneous Localization and Mapping (SLAM), which has been a topic of research in robotics for a long time,

even though there have been no major changes compared to machine learning in the past decade. The recent need for SLAM in other fields has created a great demand for it. The most important applications of SLAM in computer vision are autonomous cars and augmented reality.

Researchers divide the SLAM technique into two main categories: the traditional feature-based technique and the comparatively newer direct method. Feature-based methods focus on tracking feature points such as Harris corners in multiple video frames. Direct methods use the entire image for tracking. Feature-based methods are faster, but direct methods are better for parallelism. Outliers can be efficiently removed from feature-based methods while direct methods are less flexible in removing them. Feature-based methods have no need for good initialization, but direct methods need some technique for initialization.

The history of SLAM is not very long, and even significant achievements are less than a decade old. Michael et al. proposed one of the first feature-based SLAM techniques, FastSLAM [34], in 2002. A pioneer researcher in this field, Andrew J. Davison, published his research on MonoSLAM [23] in 2003 using a single monocular camera while other researchers were still thinking about the solution of SLAM using a stereo camera. His approach is based on Structure from Motion (SFM) and involves the creation of a sparse but persistent map of natural landmarks within a probabilistic framework. In 2007, MonoSLAM was beaten by another approach called Parallel Tracking and Mapping (PTAM) [32], which uses separate threads for mapping and tracking. However, the constraints were that the scene has to be small and initial mapping was required using a stereo camera rig.

In 2011, after Microsoft released the Kinect camera for Xbox, Richard A. Newcombe, a pioneer researcher on Kinect authored several publications [36][39][29][38] on 3D reconstruction and tracking. These publications offered tremendous contributions to this field, and the use of an infrared-based depth camera for 3D reconstruction and mapping began. Recent research on SLAM is mainly on direct methods that rely on full image tracking instead of sparse feature tracking. A dense continuous-time tracking and mapping method using RGB-D cameras [31] was proposed; it uses direct image alignment for continuous trajectory representation.

Newcombe continued his research on 3D reconstruction and SLAM and proposed a technique [37] for real-time tracking and reconstruction of the non-rigid scene with moving objects and a deformable scene. Jakob Engel et al. published a few papers on large-scale direct SLAM [24][21] that are also involved in direct image registration instead of feature tracking. Last, but not least, notable research is on tracking ORB features [35] in real-time, which is similar to the previous technique, PTAM.

Chapter 3

Implementation

3.1 iRay Overview

iRay is the proposed model of an application starting from medical scan to real-time visualization and tracking on mobile devices. Initially, the mobile device was an iPad and later was extended with an augmented reality head-mounted display. Initially, the model was applied to the real-time video in an iPad that overlays virtual organs on video using OpenGL. A Structure Sensor [17] was used as the depth and SLAM information provider. A working prototype was developed based on the proposed model and was validated by measuring the accuracy.

Figure 3.1 depicts the model and highlights three main modules. The left part is offline pre-operations related to preparation of patient data including medical scan, generating 3D models of target organs, and preparing an entry in the online database.

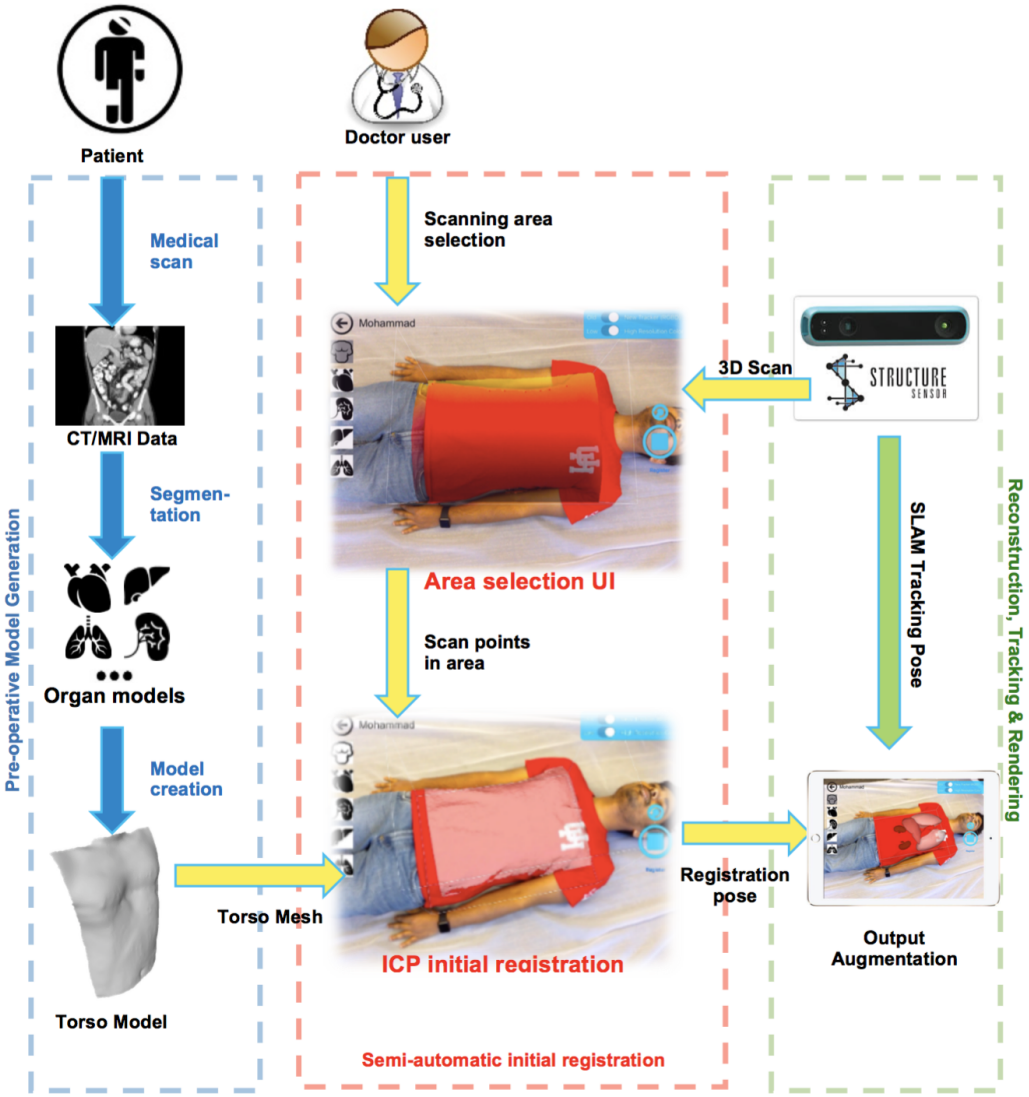

Figure 3.1: Overview of iRay. Depicted are three major modules: 1) Generating pre-operative models, 2) scanning and registration, and 3) tracking and augmentation.

3D anatomical models are created from medical data using freely available software. These models along with a torso model, which is mandatory for registration, are compressed into an archive and uploaded to an online repository. The mobile application has an interface to initiate database synchronization from the server. The user selects a patient from the list, and the corresponding 3D models are downloaded from the repository before starting registration and augmentation.

The middle portion in Figure 3.1 is the initial user-assisted registration wherein, the application loads the downloaded 3D models of the selected patient onto the real-time video. The models are aligned on the patient's body using the Iterative Closest Point algorithm. The registration is between only the pre-operative torso and the scan obtained by the depth sensor. The user has to select a scan area guided by a 3D cube and semi-transparent overlay of the pre-operative torso. This overlay provides the user with an important hint about the shape of the pre-operative torso so the user can select the area that matches best with that shape. Based on the user selection area, the 3D model of the world generated by the sensor is cropped. This cropping is necessary because registering a pre-operative torso with a scan of the entire environment is not accurate and is computationally expensive.

The right portion of Figure 3.1 includes 3D reconstruction, tracking, and rendering. The sensor accomplishes 3D reconstruction, which has to be done before starting the initial registration because the registration process uses reconstructed mesh. The tracking part is continuous and done by the sensor, which provides the camera pose to OpenGL, based on the SLAM algorithm of its SDK. OpenGL consistently applies that pose to all anatomical 3D models in the scene. OpenGL also

renders video frames provided by the device's RGB camera beneath all anatomical models.

3.2 Task 1.a: Segmentation

Segmentation is an important part of visualization from DICOM datasets. Much research has been done on 2D image segmentation, but segmentation from 3D volume is not widely explored, and the efficiency of 3D segmentation vastly depends on imaging quality and the nature of the target tissue. Application and methods of MRI segmentation, described in [22], are approaches that rely on machine learning and image processing algorithms. 3D segmentation of MRI images is explored in [20] [26], mainly on brain structure segmentation. Segmentation of torso and other organs will not be possible using a common technique, and different approaches should be considered for different parts of the human body.

The focus of this thesis is not on the segmentation algorithm, and it is assumed that any established algorithm or human operator will do the segmentation for this project. Therefore, target 3D anatomical data from DICOM slices were segmented manually using a freely available software called Slicer [16]. Figure 3.2 depicts an example of Slicer windows for volume segmentation.

The first step in segmentation is a threshold. The software has a slider where we can adjust the lowest and highest intensity thresholds to select the desired tissues in volume. The points that have an intensity value within this range are kept, and the corresponding region is highlighted in green. The target area is selected

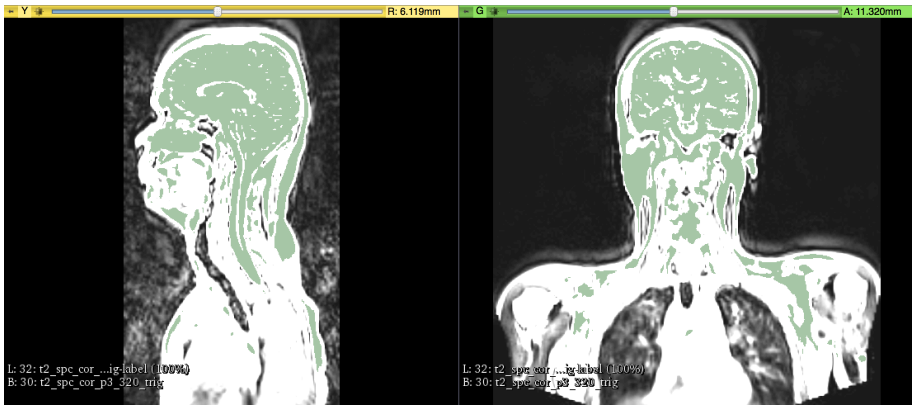

Figure 3.2: An example window using Slicer software while segmenting MRI data of a volunteer. Targeted points are separated with thresholds and are highlighted. (L) Depicted is a side view with 24 *cm* horizontal length and (R) Depicted is a front view with 50 *cm* horizontal length.

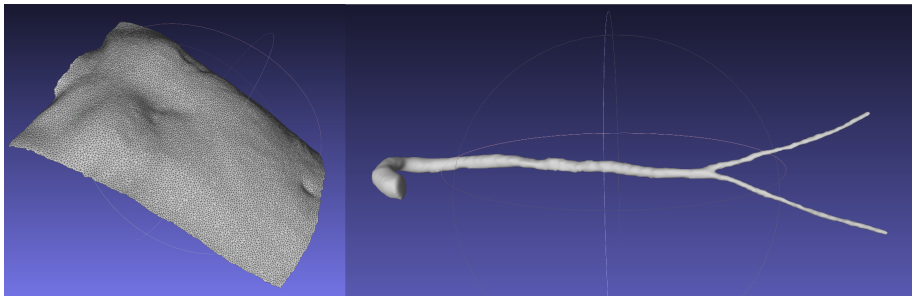

Figure 3.3: (L) Depicted is the segmented torso with 42 *cm* horizontal length and (R) Depicted is the aorta with 48 *cm* horizontal length. Both are obtained from a volunteer's MRI.

by using other tools within the application. These separated points are then used for generating a mesh of target organs. The mesh is resampled, and the torso is separated from other organs. Figure 3.3 depicts the segmented torso and aorta from a volunteer's DICOM dataset.

3.3 Task 1.b: Visualization

As a part of this thesis, a 3D volume rendering from DICOM dataset was implemented using OpenGL because it is available for mobile phones. All 2D slices were merged to build a 3D volume. Once we get a 3D volume, we can render 3D points directly, but rendering nearly a hundred million points is not a good choice, especially for mobile devices or computers without a GPU. Although direct point rendering looks like 3D, the frame rate becomes very low to use in a real-time video. Ray casting techniques that present 3D volume on a 2D screen are needed. Ray casting has several approaches: Maximum Intensity Projection (MIP), Local Maximum Intensity Projection (LMIP) [40], and alpha-composition. All three of approaches were implemented. Another variant of LMIP was also used, it is named Continuous Local Maximum Intensity Projection (CLMIP). A linear color transfer function was used, which maps color and opacity based on the ratio of the intensity value to the range of intensity values in volume.

The concept of ray casting is shooting rays through the volume in viewing direction and projecting the points that have sufficient intensity on the screen. In MIP, rays traverse the whole path and select the points that have the maximum intensity in their path. LMIP is similar to MIP, but rays stop when they get a scalar value greater than or equal to a threshold. One parameter is involved in this, but it significantly improves execution time because the rays do not need to traverse the whole path in the volume. Unfortunately, the output suffers in quality. Therefore, a modified version (CLMIP) is used in this project. The intuition behind the new

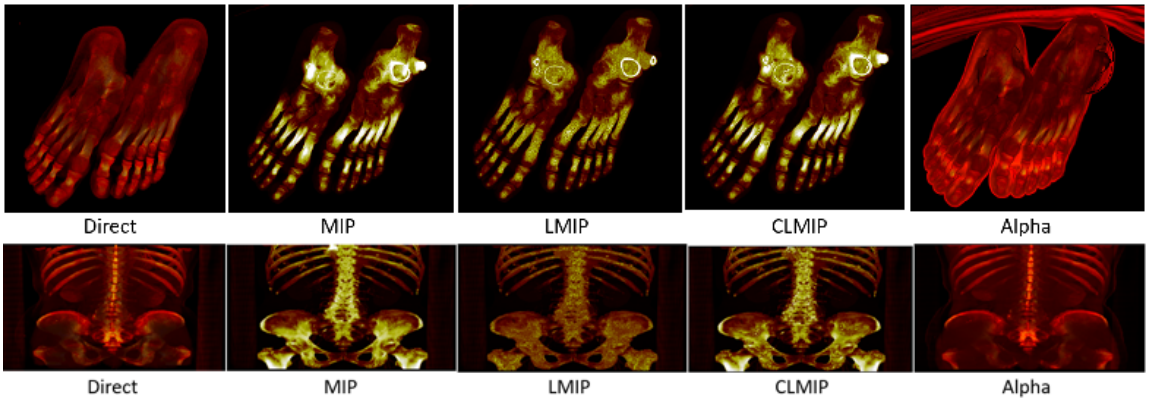

Figure 3.4: A comparison between the output of different volume rendering techniques implemented for two different DICOM datasets [14]. The horizontal length of each image is 256 pixels on the output screen.

technique is that after getting a threshold value, it is more likely that the rays will get similar or higher value in next points in their path. This is because similar tissues exist in a region, and taking the maximum scalar value in an area will present these tissues best.

The alpha composition, on the other hand, is entirely different from all of the above. Like MIP, rays traverse the whole path and consider every point to have a contribution on the screen. Therefore, it combines the color of the current output with the color and opacity of a new point in the path. The execution time is longer, which is almost similar to the direct rendering of all points. However, the quality of rendering is good and looks like 3D whereas all other ray casting techniques described above lose the depth cue and look like 2D. Figure 3.4 depicts a comparison between different ray casting techniques implemented for two different DICOM datasets.

This visualization implementation was not used in the final iRay project because it is not feasible to use on a mobile device due to higher computation requirements

and lower frame rate in interactivity. It takes around one second to cast rays through the whole volume, and we need parallel processing using a GPU. Therefore, an alternative approach was used where we segment target organs from the volume and generate 3D models using freely available software.

3.4 Task 2.a: Database and User Interface

Online and Local Database: The first implementation of iRay was accomplished on Apple iPad Air 2, and a Structure Sensor added the depth-sensing capability. The target operating system is the iOS, and native development using Apple's XCode was required. Therefore, iOS Frameworks, OpenGLES, and Structure SDK are required for the iRay implementation. The local database in the mobile device is based on SQLite with the corresponding program to access and update entries. The mobile application connects to the online database on demand, and adds and updates entries in the local SQLite database. This online database can be any SQL database, JSON, or even simple text file containing patient information as comma-separated values (CSV) in each line. Currently, such a simple CSV file is used in development. When the user selects a patient and then goes to the augmentation view, the application searches for corresponding data files in the local directory. If these files are not found, the application downloads them from the online repository and shows as a brighter color in the list.

User Interface: The interface of the iRay application is very simple and consists of only two pages. The first page is the list of patients, which includes features such

as synchronizing with the online database, downloading 3D models of a patient, and selecting a patient for the augmented reality view. Figure 3.5 depicts a list of patients in the application.

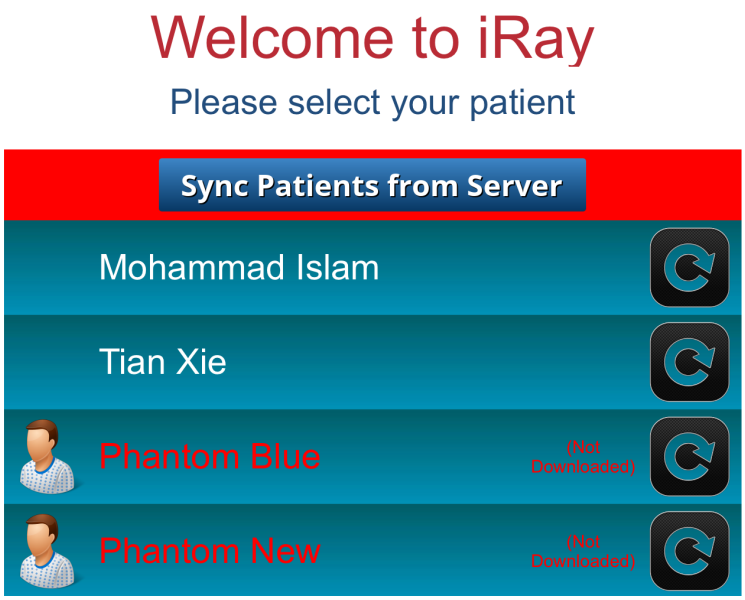

Figure 3.5: The first view of the iRay application. It includes features such as a list of patients, updating database, and downloading pre-operative data from server.

Once a patient is selected, the application switches to the next view, which is the main AR view in the application. Figure 3.6 depicts the AR view, which includes features such as buttons to show or hide organs, sliders to adjust the scanning box's depth, and Scan and Reset buttons. The most important component is the scanning box and pre-operative overlay which guides the user to select scan area. This guidance is paramount because registration accuracy depends on the portion of the body selected by the user for scanning. Input from the iPad camera is used as the background of the augmented reality view, with pre-operative models overlaid.

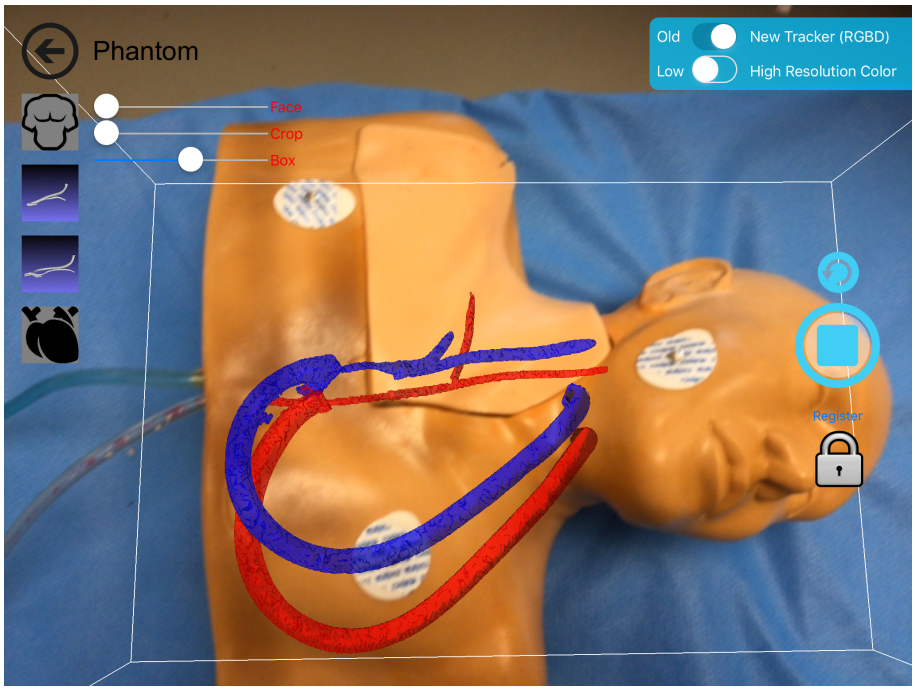

Figure 3.6: The augmented reality view in the iRay application. It renders anatomical models with iPad video in the background. The user interface includes scanning box, buttons, and sliders.

Because 3D models are incorporated into the 2D video, OpenGL for embedded systems (OpenGLES) was used as a visualization tool. The initial registration aligns pre-operative models onto the patient's body. In the tracking step, SLAM returns the camera pose in each video frame. This camera transformation is combined with model transformation to update model's pose in each frame.

3.5 Task 2.b: Improving Scanning Quality

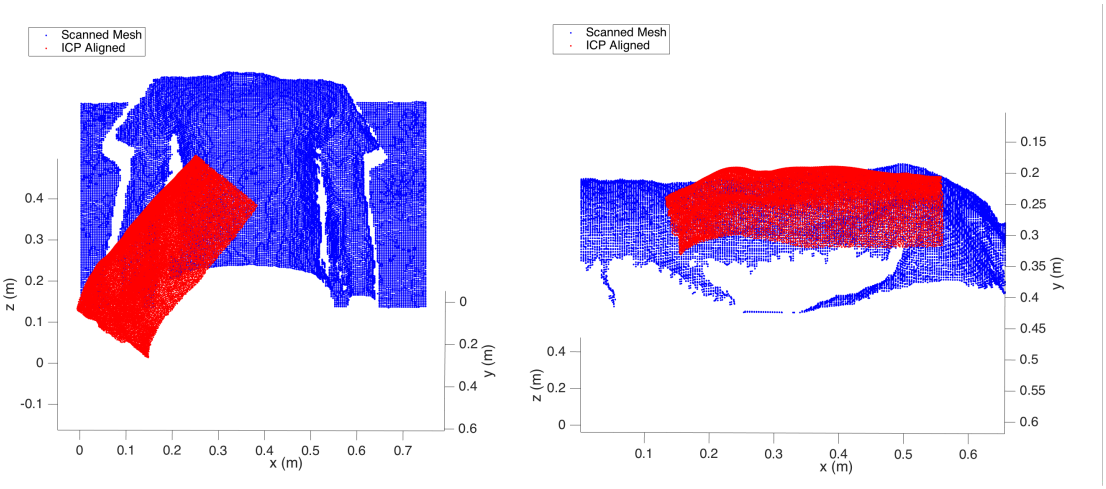

Figure 3.7: Depiction of two incorrect registrations of pre-operative torso. (L) Depicted is an intersected orientation and (R) Depicted is a flipped orientation.

The Iterative Closest Point (ICP) algorithm is highly dependent on the similarity between the two point sets, and the patient should have the same pose and wear clothes that do not deform the shape of his/her torso surface. The user can clearly identify a significant error that needs re-scanning. For examples of incorrect registration, figure 3.7 (L) depicts intersected orientation, and figure 3.7 (R) depicts flipped orientation.

Because the registration algorithm is significantly affected by the dissimilarity between the two inputs, registering the same data generated from the same object would be the best. However, the sensor generates a 3D map of the entire environment visible by its camera. Automatically searching or recognizing a torso surface in the entire 3D environment is certainly not feasible on mobile devices. Even though some

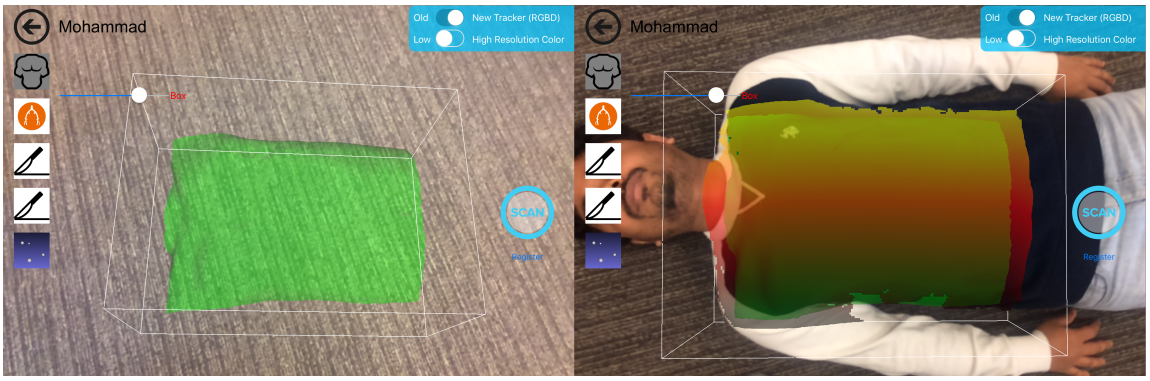

Figure 3.8: A visual guidance system to assist the user in selecting a scan area. (L) Depicted is the area selection box with the pre-operative torso overlay and (R) Depicted is a good selection of the torso area.

computer vision algorithms may do this, the accuracy of the recognition will be worse than human selection. Therefore, a visual guidance strategy was developed to help users understand which area should be scanned for a better registration result. The visual guidance includes an adjustable 3D box and semi-transparent overlay of the actual pre-operative torso. Users can adjust the box's depth to fit the overlay onto the real torso. Scanned 3D points within the box are cropped and supplied as the reference set for the ICP registration.

Figure 3.8 (L) depicts the area selection box with the pre-operative torso overlay, and figure 3.8 (R) depicts a good selection of the torso area. A user study [42] shows that this guidance system improves registration accuracy and overall operating time.

3.6 Task 3.a: 3D Reconstruction

Once we obtain the segmented parts of targeted organs from the patient's DICOM slices, we obtain one set of points that needs to be registered with the scan of the patient's body. The other set of points used in the registration process is a scan of the patient's torso, which is obtained using a depth sensor. A depth sensor is a type of camera that scans the surrounding environment and generates a 3D map based on the depth obtained using infrared or time of flight techniques. Since the depth sensor can scan only the outer surface of patient's body, only the torso points from pre-operative models are used for registration. However, once we registered the pre-operative torso in the 3D map, we can apply the same transformation to other organs because all organs and torso are obtained from the same volume and, therefore, are in the same relative pose.

Structure Sensor provides a raw depth buffer for the current video frame that can be used for reconstructing a 3D model of the world in front of the camera. The depth buffer is a grid of distances between the camera and the obstacle in light's direction. The sensor provides depth, normal, and color buffer for a 640x480 or 320x240 window. Thus, in each video frame, we obtain a 640x480 array of floating point values that represent the depth in millimeters for the corresponding screen point. A 3D point cloud can be generated from these depth values according to the following algorithm.

```
for each row do:
    for each column do:
        Point.x = depth * (C - Cx) / Fx;
        Point.y = depth * (Cy - R) / Fy;
        Point.z = depth;
    end for
end for
```

Here (Fx, Fy) and (Cx, Cy) are parameters intrinsic to the camera (focal length and center of the screen respectively) and (C, R) are column and row indexes in the depth buffer. The sensor also provides a feature for exporting processed 3D mesh with texture.

3.7 Task 3.b: Initial Registration

The iRay consists of two major components: initial 3D-to-3D registration and tracking using SLAM. The initial registration is the most important and the main contribution of this project. The main challenge is to overlay pre-operative models onto the patient's body in real-time. Previous sections have described how to obtain pre-operative torso points from the medical scan (with segmentation) and torso points during the intervention (using a depth sensor). Point clouds from these two models are obtained from their meshes and a 3D-to-3D registration algorithm is applied to estimate the transformation that needs to be applied on the pre-operative model.

The registration algorithm is ICP, which is described in the previous chapter. Pre-operative torso points are the source and scanned torso points are the reference set for ICP. For this kind of registration, it is very important for the two point clouds to be as similar as possible. Therefore, the iRay application includes a guidance system that instructs the user to select an area of the patient's body that is similar to the pre-operative torso shape. Since we select which patient is in intervention and load pre-operative models accordingly, the two 3D torso models should have the same shape. An ICP implementation [27] was used in this project, which returns rotation and translation to apply to pre-operative 3D models.

3.8 Task 3.c: Tracking

Structure SDK has a SLAM implementation, which is used in this project. That SLAM is not identical to any feature-based or direct method described in the previous chapter; rather, it uses depth acquired from the sensor to reconstruct a 3D map of the world and performs ICP registration between subsequent maps to estimate camera transformation. Following is a brief description of the steps used in Structure Sensor for both 3D reconstruction and tracking.

Step 1: Obtain the raw depth data from the infrared sensor. An infrared projector projects invisible lights onto the scene like lasers. These lights have some clusters with predefined patterns that are called structured light. The sensor reads the projected patterns on the obstacle and compares it with corresponding hard-coded patterns embedded at manufacturing. Based on the deformation of these

patterns, the sensor estimates the distance (depth in mm) between the sensor and the obstacle.

Step 2: Filter the raw depth data with a bilateral filter to remove erroneous measurements. A bilateral filter performs smoothing, which tries to reduce the difference between a point and its surrounding points so that better measurements around it compensate for error. It goes through each point and recalculates the point's value based on the values of the surrounding points. It performs this in a box window where nearer points around the target point have higher priority. It smooths error but keeps edges as well.

Sep 3: Each point in the depth map is converted to a 3D point. The depth map has only one value for each point (z in 3D), but we can estimate the x and y positions using intrinsic camera parameters as described in an earlier section of this chapter. Normal for each vertex is also calculated by applying the cross product between adjacent points.

Step 4: Whenever a point cloud is generated from the depth in a current video frame, it applies the ICP registration algorithm to determine a transformation between this point cloud and the point cloud previously generated (mapped from all previous frames). This transformation presents the inverse transformation of the camera. In this way, the camera pose is estimated in each video frame.

Step 5: The point cloud extracted from the current frame is transformed using the estimated camera pose and merged with the previously mapped cloud to refine it. It further enriches the mapping by adding new points acquired from a different

camera pose. If the estimated pose is too different from the previous frame's pose or the error in ICP is too high, the current depth frame is discarded.

This chapter described implementation details of tasks 1-3 of this thesis. Task 4, which includes measurement of registration and tracking error is described in the next chapter.

Chapter 4

Results and Discussion

4.1 Task 4.a: Registration Error

Measuring accuracy from real-time video frames is difficult even for expert eyes with knowledge of human anatomy. The overall performance of the application depends on the initial registration and tracking accuracy. However, other factors such as segmentation accuracy, 3D reconstruction accuracy, and errors caused by deformation may affect the performance. Therefore, we need to put some constraints on those factors and focus on registration and tracking error only. It is assumed that 1) a human operator or an algorithm will do the segmentation very accurately, 2) the patient's pose should be the same on CT/MRI and during the intervention, and 3) torso shape should not be largely deformed due to factors such as clothing, fullness of stomach.

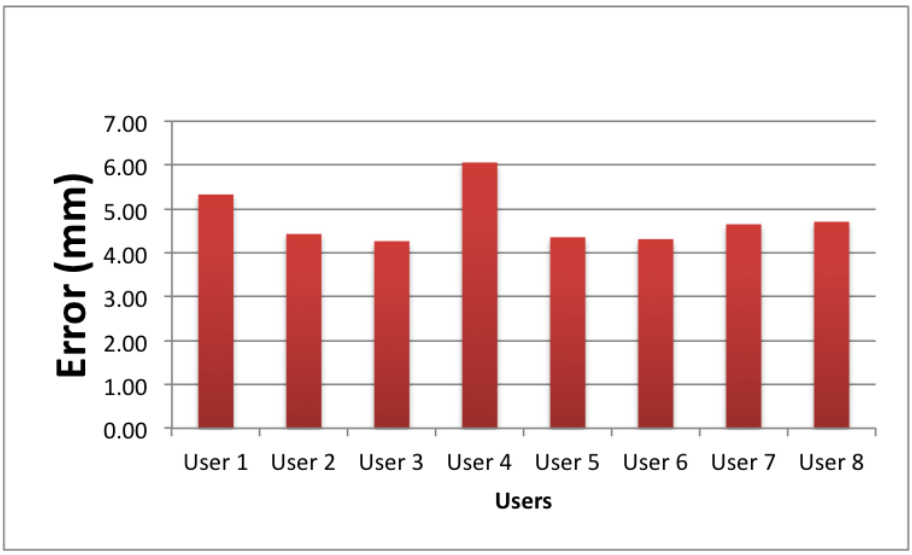

Figure 4.1: Average initial registration error of eight users. Total 40 trials performed for scanning torso and registration.

It is assumed that minimizing registration errors between the scanned and pre-operative torso will reduce Target Registration Error (TRE) as well. A user study was performed to measure the registration error. Eighty trials from eight different users were evaluated in this study for two slightly different versions of the iRay. The users were between 24-31 years of age with at least a college education. None of the users had any pre-knowledge about the iRay, nor did they have any related application experience. After five minutes of short training and one or two self-practices, every user was asked to select the target area five times for each version. The results for the latest version, which has the user guidance system, is slightly better in accuracy and operation time. The average error for eight users and 40 trials of the most recent version is **4.76** $\boldsymbol{mm}$. The bar chart in figure 4.1 depicts the average error for each user. However, this error can not represent Target Registration Error (TRE), which is the displacement of overlay of an internal organ from its actual position.

Figure 4.2: A phantom while measuring tracking error using four markers. The displacement between the actual position and the position of overlays are recorded in each video frame.

4.2 Task 4.b: Tracking Error

Not only does the initial registration error affect the overall performance of iRay, but the tracking error may also displace overlays from the positions in where they were initially placed. Marker-based registration is stable, but tracking the camera pose using SLAM always has some biases and drifts. Therefore, measuring the tracking error is also necessary for measuring the overall performance of the system. This type of measurement is known as Fiducial Registration Error (FRE), which is a standard measure of error between the actual position of fiducial markers in the patient's body and the overlay of them.

A fiducial marker-based tracking error measurement was accomplished in this project on a basic phantom. Four markers were placed on the torso of the Phantom, and it was scanned by the sensor instead of a CT/MRI to generate a pre-operative model. Scanning the same surface with the same device for generating pre-operative

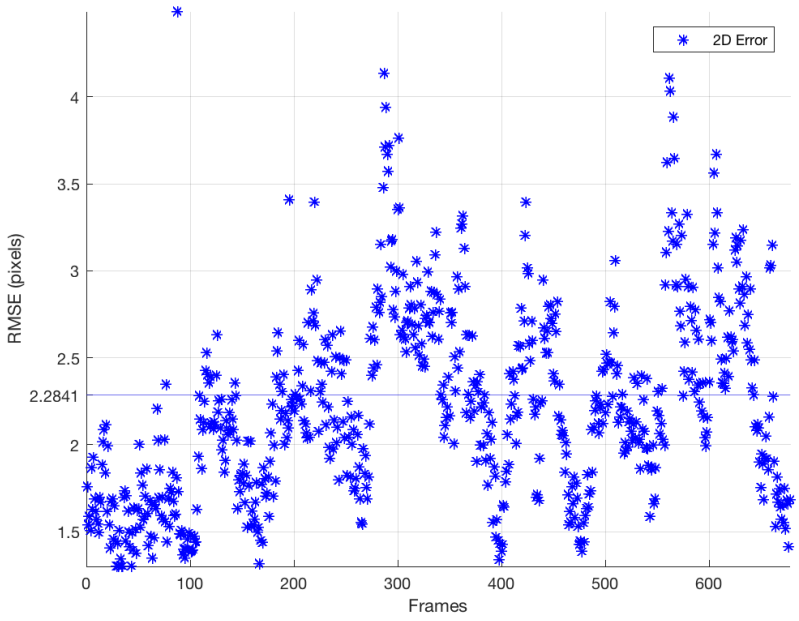

Figure 4.3: Depiction of 2D pixel errors in tracking obtained from video frames.

and scanned model will reduce registration errors, and we can estimate the tracking error more accurately. However, there is still a registration error, which biases the tracking error slightly, and it can not be removed. This FRE measurement presents overall quality of the system because it is the combination of both initial registration error and tracking error. However, minimizing FRE or point cloud registration error does not guarantee minimization of TRE, which is proven by Michael J. Fitzpatrick in his paper [25]. Figure 4.2 depicts the Phantom James while measuring tracking error.

Markers were detected continuously in video frames, and their screen position and corresponding 3D position from OpenGL were recorded for each frame. A few thousand samples were taken while rotating and moving the iPad randomly. The 2D

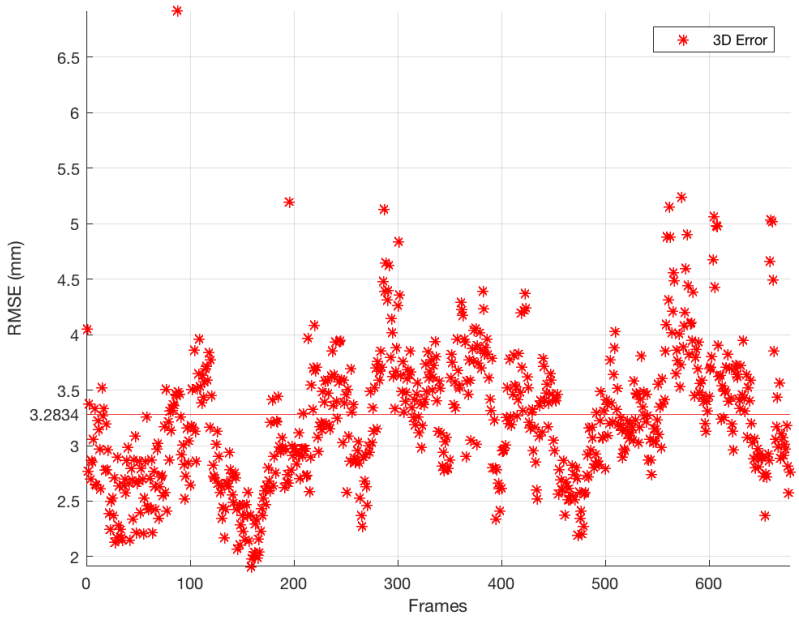

Figure 4.4: Depiction of 3D position errors in tracking obtained from video frames.

pixel error is the average distance between the screen position of actual markers and their overlays. The 3D error is the average distance between the world 3D position of markers (from depth) and OpenGL 3D coordinates of corresponding overlays. The final root-mean-square error (RMSE) was calculated from only the frames that could detect all four markers. Figure 4.3 and 4.4 depict plots of 2D screen and 3D position errors respectively. The average 2D screen error is **2.28 *pixels***, and the 3D position error is **3.28 *mm***. One thing to note is that this error also includes biases caused by initial registration error. Therefore, correct tracking error should be slightly lower than the estimation.

4.3 Discussion

Followings are some limitations of the iRay.

- The most significant limitation of iRay is the unstable registration accuracy of target organ; it does not guarantee accurate localization of a target organ even after registration accuracy of the torso is good. Also, initial registration is not guaranteed since ICP sometimes converges to an inappropriate local minimum, where the registration error is least, but alignment is not perfect.

- The application becomes semi-automatic for initial registration since the user has to select an area to scan for torso registration. This is usual for ICP-based registration where accuracy greatly depends on the similarity and initialization. The user-assisted scanning system obtains the similarity, and the initialization is obtained by defining the beginning pose of the scan.

- The segmentation of torso and other anatomical structures from the patient's medical data is fully manual using third-party software.

Chapter 5

Explore Holographic iRay

5.1 HMD Technologies and Glasses

In task 5 of this thesis, a feasibility analysis is performed on different augmented reality head-mounted display (HMD) and glasses. Although the augmented reality is a very new technology, several companies have been manufacturing such AR glasses or HMDs for a few years, and some have made noticeable progress already. A total immersive AR experience is not just on a computer or mobile screen, but includes seeing holograms in the real world through optical glasses or even with open eyes. This chapter presents the future possibility of iRay in wearable display technologies with a comparison between different HMDs and glasses. Some desired features includes optical see-through, depth sensor and position tracking. There are some techniques to track camera position from outside of the device, but we need a device with self-tracking capability. Following are some augmented reality technologies

currently available for consumers.

Microsoft HoloLens: HoloLens [7] by Microsoft is currently one of the best AR technologies. Although its field of view is narrow, it has some advanced technologies and enough pixel density for smooth blending of holograms with human vision. It has an excellent gesture recognition and voice command system to manipulate holograms in the scene. The most useful feature of HoloLens is that it has position tracking and 3D spatial mapping that enables holograms to understand the environment and track 6-DOF pose by itself wirelessly. Spatial mapping data is accessible and necessary for this project.

ODG: Osterhout Group Inc. [15] has been developing their ODG glasses for a few years. They have several versions available for consumers including R-7, R-8, and R-9 in pre-order. All versions have powerful processing units with an Android-based operating system. However, their glasses do not come with a depth sensor or spatial mapping feature.

Meta 2: Meta 2 [10] is probably one of the best AR HMD because of its wide field of view - 90 degrees. However, it requires a Windows-based computer to be connected to its headset. It has gesture recognition and position tracking features like HoloLens, but does not have a depth sensor. Therefore, it is not very suitable for iRay.

Atheer Air: Atheer Air [1] is the only AR technology we found, other than

HoloLens, that have a depth sensor. Their glasses are operated from an Android-based phone. The company focused more on software for hand gestures, head tracking, and voice command which is compatible with few other AR glasses. This product is even pricier than Microsoft HoloLens.

Epson Moverio: Epson has been developing their Moverio [3] glasses for a long time. Their glasses are very lightweight because of the separate processing unit. Available versions includes BT-100, BT-200, and BT-300 is in pre-order. Their glasses have a head movement tracker, but no depth sensors.

Other glasses such as Vuzix, and Ora 2 do not have a suitable configuration, or depth sensors included. Microsoft HoloLens still has the best features to date with the best tracking and spatial mapping. However, it does not provide access to raw depth data directly; rather it provides surface mesh which is not dense enough for accurate registration. Table 5.1 presents a comparison of the different AR devices available.

Device	**Processor**	**FOV**	**Resolution**	**OS**	**Depth**	**Price**
HoloLens	Microsoft HPU	30	720p	Windows	Yes	$3000
ODG R-9	Snapdragon 835	50	1080p	Android	No	$1800
Meta 2	Unknown	90	2560x1440	Windows	No	$949
Atheer Air	NVIDIA Tegra	50	Unknown	Android	Yes	$3950
Vuzix M300	Intel Atom	20	Unknown	Android	No	$1000
Epson BT-300	Intel Atom	23	720p	Android	No	$799

Table 5.1: A comparison of several augmented reality glasses and HMDs.

5.2 Exploration on HoloLens

Developing a holographic version of iRay is essential for the best AR experience. At first, Bridge kit by Occipital [13], which is a mixed reality kit using Structure Sensor, was chosen. This is because porting iRay to Bridge would be easier because it uses the same sensor and operating system. However, this kit is only for the iPhone, and mobile phone's screen is used as the display. Therefore, the product offers video see-through mixed reality technology; not optical see-through glasses. Moreover, its SDK renders the mapped environment in 3D with captured texture, which is not sharp enough for smooth visualization. The bridge has artifacts and causes dizziness within a few minutes of use. ODG R-7 was the second pair of glasses we tried, which has an Android-based operating system and a controller. The advantages are lightweight, sharp screen and larger field of view. However, it does not come with the spatial mapping feature. Also, it depends on marker-based tracking using third-party SDK such as Vuforia.

Compared to the devices mentioned above, HoloLens has both spatial mapping and tracking features. Once holograms are placed in the world, they are kept static in the surrounding environment. HoloLens automatically applies tracking to holograms so that developers can concentrate on the initial registration. For registration, the same strategy can be used such as the iPad version of iRay. Spatial mapping data is obtained from the SDK feature, and the points of the patient's torso mesh are

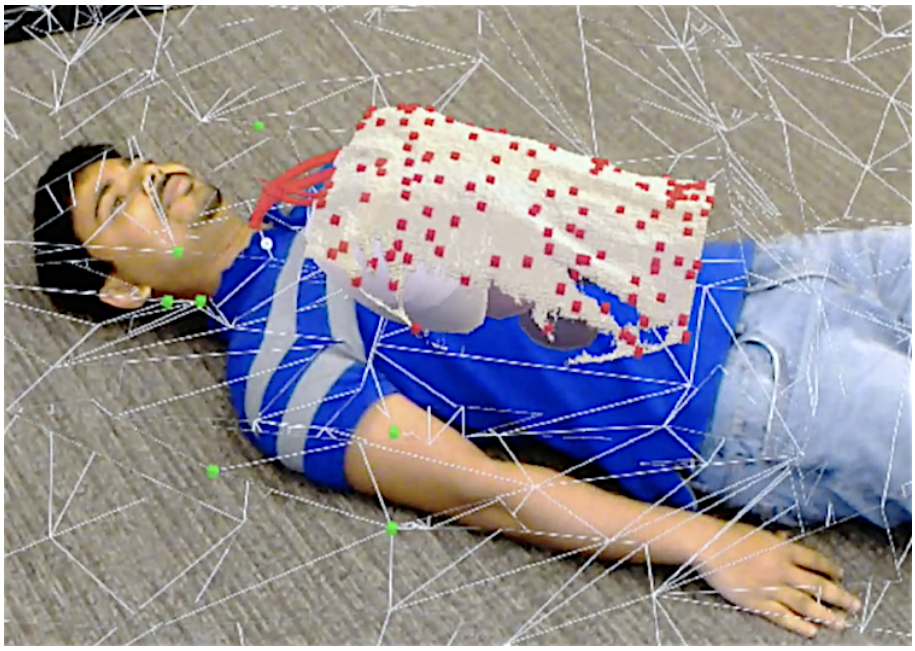

Figure 5.1: Exploration on HoloLens. White wireframe depicts spatial mapping. Small cubes depict some points obtained from target mesh and pre-operative torso.

separated from that. Separating these points is tricky because it needs cropping from the whole environment map.

HoloLens provides spatial mapping data as a collection of small meshes instead of a single large mesh. Based on the gaze (a pointer works as a cursor) direction, the possible surface that contains the patient's torso is detected. Points from that surface mesh are obtained and used as the reference set for ICP. Acquiring the points of the pre-operative torso is accomplished by reading vertices from that mesh. The same ICP library used in the iPad project is ported with DLL export and used to register these two sets of points.

Figure 5.1 depicts a screenshot of the current development version of the iRay in HoloLens. Small cubes in the figure present some pre-operative torso vertices, and

white wireframe depicts the spatial mapping mesh. ICP registration result is not reliable yet for some unknown issues which result in a large transformation.

Chapter 6

Future Work and Conclusion

6.1 Scope of Future Work

Based on limitations described in Chapter 4, iRay needs more refinement before being considered as a commercial application. Following are some future works in some specific tasks of the thesis:

Task 1.a: Separating anatomical data manually from a CT/MRI scan is a time-consuming task and not very accurate. Based on the target organs, an automatic algorithm for separating points should be considered.

Task 1.c: Rendering target 3D points directly from CT/MRI data can be an alternative to segmentation. In that case, mobile GPU can be utilized for parallel processing. This will reduce the segmentation error and visualize the exact structure of the patient's anatomy.

Task 3.b: We need more robustness in the registration algorithm so that it can accurately register in a large, unknown environment mapped by the 3D sensor. It will make the system fully automatic eliminating the necessity for user interaction to select a scan area. A machine learning approach can possibly be applied to detect a torso from a large point cloud.

Task 4.a: Measuring Target registration error (TRE) is not accomplished in this thesis. TRE is the difference between the estimated position of a particular organ and the real position of the organ inside a human body. Measuring TRE instead of ICP registration error will better present the overall performance. However, it will require an actual intervention.

Task 5: Analysis on different AR glasses/HMDs has been performed, and it was decided that it is possible to port iRay to a holographic platform with depth sensor (self-contained or combined). Exploring more AR devices and improving registration accuracy in HoloLens can be the next work for this project.

Others: SLAM can not track patient's movement, and occlusion may cause tracking to fail. When the camera is static, we can track object motion, but if both camera and object are moving, current SLAM technique can not track both. Therefore, we also need a solution for non-rigid scene tracking.

6.2 Conclusion

iRay is an application prototype targeted for medical use to help physicians plan before an intervention. It also can be useful in education teaching medical students anatomy with real data. The iRay itself is a combination of multiple existing technologies and algorithms. This thesis focused on solving a real-world problem with the help of available resources rather than developing or improving an algorithm. Improvement in those devices and technologies will increase the overall accuracy of the application. The proposed application model was successfully implemented and validated with sufficient accuracy measurements. The concept of iRay can be further extended with new devices. All tasks were accomplished, and following are some significant contributions of this work.

- Visualization of internal organs based on external surface registration.
- Spatial mapping and tracking using depth sensor to register anatomical data in the physical world.

Bibliography

[1] Atheer Inc. http://www.atheerair.com/smartglasses.

[2] DICOM Slices. http://www.cabiatl.com/mricro/mricro/mritut.html.

[3] Epson Moverio. https://epson.com/moverio-augmented-reality-smart-glasses.

[4] Evena Medical. https://evenamed.com/deepvu-wearable-ultrasound/.

[5] German Cancer Research Center. http://www.dkfz.de/en/index.html.

[6] Google Tango. https://get.google.com/tango.

[7] HoloLens. https://www.microsoft.com/microsoft-hololens/en-us/.

[8] Intel RealSense. http://click.intel.com/realsense.html.

[9] Medsights Tech. http://www.medsightstech.com.

[10] Meta. https://www.metavision.com.

[11] National Electrical Manufacturers Association. http://www.nema.org.

[12] Niantic Inc. https://nianticlabs.com.

[13] Occipital Inc. https://occipital.com.

[14] Osirix. http://www.osirix-viewer.com.

[15] Osterhout Group Inc. http://osterhoutgroup.com.

[16] Slicer. https://www.slicer.org.

[17] Structure Sensor. https://structure.io.

[18] C. Bichlmeier, F. Wimmer, S. M. Heining, and N. Navab. Contextual anatomic mimesis: Hybrid in-situ visualization method for improving multi-sensory depth perception in medical augmented reality. In *Proc. 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 129–138, Nara, Japan, Nov. 13-16 2007.

[19] T. Blum, V. Kleeberger, C. Bichlmeier, and N. Navab. Mirracle: An augmented reality magic mirror system for anatomy education. In *Proc. IEEE Virtual Reality Short Papers and Posters*, pages 115–116, Costa Mesa, CA, USA, Mar. 4-8 2012.

[20] M. Bomans, K.-H. Hohne, U. Tiede, and M. Riemer. 3-D segmentation of MR images of the head for 3-D display. *IEEE Transactions on Medical Imaging*, 9(2):177–183, 1990.

[21] D. Caruso, J. Engel, and D. Cremers. Large-scale direct SLAM for omnidirectional cameras. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 141–148, Hamburg, Germany, Sept. 28 - Oct. 2 2015.

[22] L. Clarke, R. Velthuizen, M. Camacho, J. Heine, M. Vaidyanathan, L. Hall, R. Thatcher, and M. Silbiger. MRI segmentation: Methods and applications. *Magnetic Resonance Imaging*, 13(3):343–368, 1995.

[23] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

[24] J. Engel, J. Stuckler, and D. Cremers. Large-scale direct SLAM with stereo cameras. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1935–1942, Hamburg, Germany, Sept. 28 - Oct. 2 2015.

[25] J. M. Fitzpatrick. Fiducial registration error and target registration error are uncorrelated. In *Proc. SPIE Medical Imaging 2009: Visualization, Image-Guided Procedures, and Modeling*, volume 7261, pages 726102.1–12, Lake Buena Vista, FL, USA, Feb. 7 2009.

[26] P. A. Freeborough, N. C. Fox, and R. I. Kitney. Interactive algorithms for the segmentation and quantitation of 3-D MRI brain scans. *Computer Methods and Programs in Biomedicine*, 53(1):15–25, 1997.

[27] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proc. IEEE Conference on Computer*

Vision and Pattern Recognition, pages 3354–3361, Providence, RI, USA, June 16-21 2012.

[28] N. Haouchine, J. Dequidt, I. Peterlik, E. Kerrien, M.-O. Berger, and S. Cotin. Towards an accurate tracking of liver tumors for augmented reality in robotic assisted surgery. In *Proc. IEEE International Conference on Robotics and Automation*, pages 4121–4126, Hong Kong, China, May 31 - June 7 2014.

[29] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera. In *Proc. 24th Annual ACM Symposium on User Interface Software and Technology*, pages 559–568, Santa Barbara, CA, USA, Oct. 16-19 2011.

[30] I. A. Kakadiaris, M. M. Islam, T. Xie, C. Nikou, and A. B. Lumsden. iRay: Mobile AR using structure sensor. In *Proc. 15th IEEE International Symposium on Mixed and Augmented Reality*, Merida, Mexico, Sept. 19-23 2016.

[31] C. Kerl, J. Stuckler, and D. Cremers. Dense continuous-time tracking and mapping with rolling shutter RGB-D cameras. In *Proc. IEEE International Conference on Computer Vision*, pages 2264–2272, Santiago, Chile, Dec. 13-16 2015.

[32] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 225 – 234, Nara, Japan, Nov. 13-16 2007.

[33] P. Milgram and F. Kishino. A taxonomy of mixed reality visual displays. *IEICE transactions on Information and Systems*, 77(12):1321–1329, 1994.

[34] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, et al. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proc. 8th National Conference on Artificial Intelligence*, pages 593–598, Edmonton, Alberta, Canada, July 28 - Aug. 1 2002.

[35] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[36] R. A. Newcombe and A. J. Davison. Live dense reconstruction with a single moving camera. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1498–1505, San Francisco, CA, USA, June 13-18 2010.

[37] R. A. Newcombe, D. Fox, and S. M. Seitz. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 343–352, Boston, USA, June 8-10 2015.

[38] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *Proc. 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, Basel, Switzerland, Oct. 26-29 2011.

[39] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. DTAM: Dense tracking and mapping in real-time. In *Proc. IEEE International Conference on Computer Vision*, pages 2320–2327, Barcelona, Spain, Nov. 6-13 2011.

[40] Y. Sato, N. Shiraga, S. Nakajima, S. Tamura, and R. Kikinis. Local maximum intensity projection (LMIP): A new rendering method for vascular visualization. *Journal of Computer Assisted Tomography*, 22(6):912–917, 1998.

[41] L.-M. Su, B. P. Vagvolgyi, R. Agarwal, C. E. Reiley, R. H. Taylor, and G. D. Hager. Augmented reality during robot-assisted laparoscopic partial nephrectomy: Toward real-time 3D-CT to stereoscopic video registration. *Urology*, 73(4):896–900, 2009.

[42] T. Xie, M. M. Islam, A. B. Lumsden, and I. A. Kakadiaris. Semi-automatic initial registration for the iRay system: A user study. In *Proc. 4th International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, Ugento, Lecce, Italy, June 12-15 2017. In Press.