1-1-2013

Advanced Augmented Reality Telestration Techniques With Applications In Laparoscopic And Robotic Surgery

Stephen Dworzecki
Wayne State University,

Follow this and additional works at: http://digitalcommons.wayne.edu/oa_dissertations

Part of the Computer Engineering Commons, and the Surgery Commons

ADVANCED AUGMENTED REALITY TELESTRATION TECHNIQUES WITH APPLICATIONS IN LAPAROSCOPIC AND ROBOTIC SURGERY

by

STEPHEN TERRENCE DWORZECKI

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2013

MAJOR: COMPUTER ENGINEERING

Approved by:

_____
Advisor   Date

_____

_____

_____

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1 – BACKGROUND AND MOTIVATION

Motivations

The impetus behind this research originates from multiple sources, all of which relate to the needs and opportunities created by laparoscopic surgery. The usage of laparoscopy has increased greatly in the past decade or two. With its many advantages in certain procedures, demand to perform these procedures laparoscopically has outpaced the supply of expert surgeons in this field [1]. This is a lack of surgeons to not only perform the procedures, but to also teach additional surgeons to perform these tasks. This is further hindered by the fact that while laparoscopic surgery has advantages to the patient, it also has disadvantages for the surgeon. To perform the procedures, a completely different skill set in spatial reasoning and motor skills needs to be developed. The increased difficulty in training, a shortage of trainers, and a strong desire to increase the useful training of students before they ever operate on a living patient, has motivated the two main areas of this research.

Firstly, this research will address communication between a surgeon and the camera operator, most likely a novice surgeon. In the operating room, the surgeon is dependent on the camera operator to supply a steady, upright view of the operative area. To maintain the desired view, verbal communication between the surgeon and camera operator is required. This adds to the conversations already taking place between all of the staff involved with the procedure and adds a potential distraction for the surgeon. Rather than replace the human with a robot camera holder, denying a novice surgeon the learning experience that comes with involvement in the procedure, systems can be created that can allow for non-verbal

communication that provides clearer surgeon requests and intentions in a shorter amount of time. If the situation necessitated a robotic camera holding system, these non-verbal cues could potentially be adapted for robotic use.

Secondly, this research will address instruction or collaboration between an expert and novice surgeon. Telementoring and telestration systems that are in use today have numerous limitations in practice. However, with improvements to the telementoring system, an expert surgeon could direct the novice's attention in the scene more clearly. These improvements should allow the expert to be able to clearly direct novice motions, and most importantly, be able to instruct as effectively from afar as they could being in the same room as the novice.

Background

This research touches on topics from a few different fields, though the common thread among them all is the utilization of technology to help teach the skills necessary to perform laparoscopic surgery. A review of the basic information related to each field is presented below.

Laparoscopic Surgery

Laparoscopic surgery, also well known as minimally invasive surgery, is a fairly recent surgical field uniquely identified by the doctors access to and interaction with the inside of the patient. In an open surgery, a large incision is made to the exterior of the body to allow the surgeon direct access, physically and visually, to the organs to be worked on. In laparoscopic surgery, significantly smaller incisions are made on the body, only large enough to allow the entrance of the tools the surgeon will utilize, and a small incision to insert a camera to allow the surgeon to see inside of the body. The body cavity is inflated with CO_2 to increase the working

volume in the body, and ports called trocars are placed at each incision to allow tool access without further damaging the skin, and to keep the CO_2 in the body.

Laparoscopy as a technique has been around for quite some time. Mostly used as optical, lighted viewports for exploratory diagnosis and simple procedures, it took some technological advances to stimulate its usage in the surgical field. Tools needed to be developed that could slide into a small entry hole, but still allow the surgeon the freedom and control needed for each task. Tools such as a grasper and cutting tool are shown in Figure 1. The long shafts can pass through the entry hole and the handles can be opened and closed to operate the claws or blades on the end of the tool.
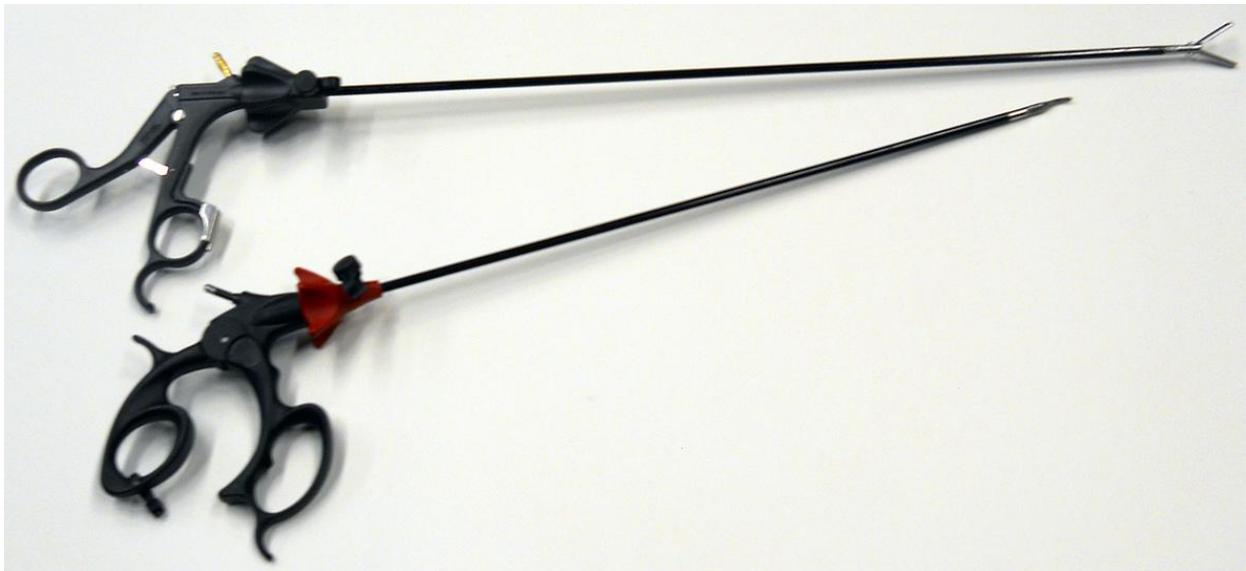
Figure 1: Laparoscopic tools: grasper and cutting tool

However, the main technological advance needed was the miniaturized integrated circuit camera. The small camera on the end of the optical scope placed in the body allows the view of the body to be displayed on a monitor at a magnified level. This afforded the surgeon a better view than they could attain with their naked eye in an open surgery, along with

removing the need for the surgeon to be hunched over the patient to attain the desired viewpoint. A scope that can have a camera attached to it is shown in Figure 2.

Figure 2: 10mm, zero degree scope

The small incisions and mostly closed body cavity in laparoscopic surgery hold a few important advantages over open surgery. The smaller incisions lead to less bleeding, and less pain for the patient. It also significantly shortens recovery time and reduces the necessary hospital stay. Keeping the body mostly closed also reduces the internal organ exposure to contaminants, reducing damage and possible infection. In addition to patient advantages, the surgeon gets a magnified view of the operative area with the camera and optics in use. However, even with these advantages, numerous disadvantages face the surgeon. Instead of an open view of the entire operative area, the view is restricted by the field of view of the camera, and is further restricted by the camera operator's expertise at knowing where the surgeon wants to see at any given moment. The working area is also restricted from being wide open to only being the intersection of where the camera can see and where the tools can reach, based on the insertion points of each device. Most importantly, since the surgeon is not directly looking at their hands interacting with tools that interact with the patient, a whole new set of spatial reasoning and motor skills need to be developed. The 2D camera view provided by the camera does not give the surgeon depth and positioning information that could be seen in an open surgery. In addition, tools operate on a pivot located at the trocar, which inverts the

tool directions between the surgeon's hands and the tool tips on camera. Mastering the movement of tools and their interaction with the patient on a 2D display requires additional practice and training not required in open surgery.

Augmented Reality

Augmented Reality (AR) is a computer related field that deals with the combination and interaction of real-world imaging and 3-dimensional (3D) computer graphics. A video feed combined with registered 3D AR data can be seen in Figure 3. Some of the earliest work in the field attempted to use computers to aid in the teleoperation of robots. One of the earliest papers overlaid a stick-figure representation of a robot on a low-resolution, approximately one frame-per-second (fps), video feed of the actual robot [2]. The limitations of network bandwidth would only allow the actual video to display robot movement with a one-second delay, but the AR overlay would be updated locally at a much higher frame rate to let the operator see an approximation of where the robot would be during the delay.
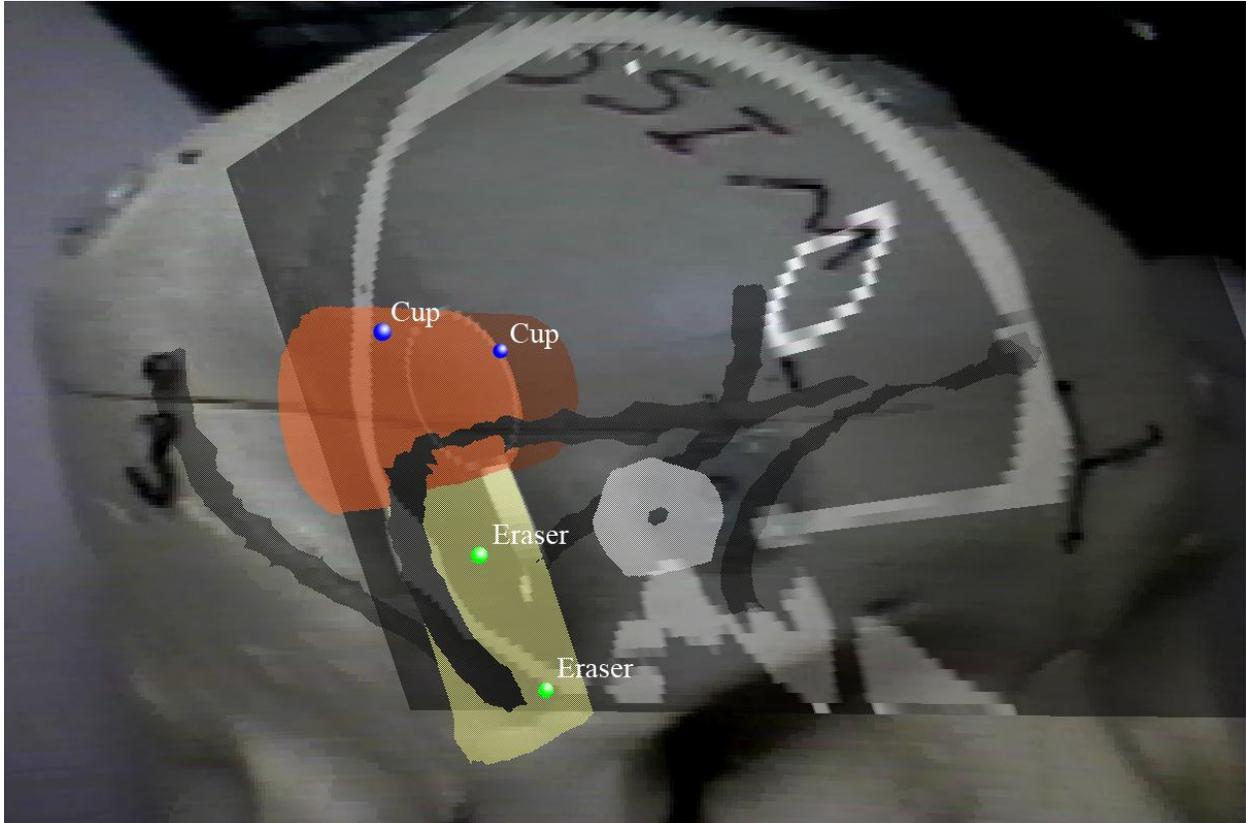
Figure 3: AR overlay of sagittal CT slice with AR markers for material scan sites inside skull phantom

With advances in technology and significantly more powerful personal computers, real-time augmented reality became a possibility in the early 1990's. The term augmented reality was coined and interest in the field accelerated after an important paper in 1993 [3]. A team at Columbia University designed a system to track a user wearing a see-through head-mounted display to provide assistance while they were performing maintenance on a laser printer. The user would look at the printer placed in a static position. Depending on the service requested, a 3D wireframe model was overlaid on the user's viewpoint to show how they should be interacting with the printer to complete their desired task.

Augmented reality does not need to be 3D data overlaid on a video feed. This research itself will handle augmented reality in 2D, 2.5D, and 3D. The naming depends on the user interaction and display readout of the implementation. 2D will refer to a two-dimensional

interaction from the user in addition to a two-dimensional display in the scene. 2.5D will refer to a two-dimensional interaction from the user that results in a three-dimensional display in the scene. 3D will refer to three-dimensional user interaction and display.

Today, AR is stepping out of academic research and is becoming a recognizable term for the masses. Sony Computer Entertainment has delivered multiple applications for their video game consoles that involved AR. Eye of Judgment for the Playstation 3 has users playing a card game under a camera that displays on the television. When cards interact with each other on the table, 3D representations of the monsters on each card materialize in the environment and battle each other. In Sony's EyePet, a small virtual pet exists in the environment your camera is viewing. The software analyzes the scene and the objects in it and the virtual pet can interact with the people and objects moving in the scene. Outside of games, the commonly used ARToolkit programming platform being ported to Adobe Flash has resulted in a multitude of AR related items [4]. BMW has printed advertisements for their vehicles that you can see 3D models of if you hold the magazine up to your webcam. Baseball card companies are starting to add card recognition so that you can see 3D players come out of your cards on your webcam. Finally, people have even started creating AR business cards that pull up different 3D objects when viewed on camera on that person's webpage.

Usage of AR will only continue to grow with the advent of the modern smartphone. The building blocks necessary for AR, a camera, a display, and significant processing power, are all now in a device small enough to fit in a pocket. With network connectivity, and position registration with the global positioning system and local tilt sensors, the smartphone market should be on the leading edge of AR applications and games in the near future.

Telementoring

Telementoring is the usage of telecommunication devices to support a mentoring relationship. It entails the teaching of other people, near or far, using some sort of communication (telephone, e-mail, Voice-Over-IP, instant messaging, etc.). This research will focus more toward telestration as a mentoring tool, though many of the other communication options have been used in surgical telementoring in the past.

A telestrator is a device that allows the user to draw on an image or video feed to provide augmentation for the viewing audience. Invented in the 1950's, it has found its only major usage in television sports broadcasting. John Madden popularized the device in football broadcasts by using it to show player and ball movements during instant replays. Modern systems contain many other features, such as AR cues (arrows, circles, curves), highlighting ability, video pause, zoom, etc. This past year, the newest Intuitive Surgical da Vinci S HD surgical robot includes a touchscreen on the unit that allows for telestration. Attendants in the operating room can draw on the screen with their fingers to point out information to the surgeon. These systems are still mostly utilized in television broadcasts and limited short-range applications because of the bandwidth requirements for sending these video feeds in addition to the system's reduced usefulness with communication latency. On top of that, most of these systems only register the drawings to the display screen rather than to the content on the screen. When the camera or the objects on the screen move, the drawings stay at the same place on the screen. If an important object was circled and the camera or object moved, that object would no longer appear within that circle.

Research Questions

The questions guiding the research plan laid out in this dissertation are as follows:

- How can augmented reality be utilized to assist in the usage of minimally invasive surgical tools?

- How can telestration systems be improved to make their usage in telementoring feasible and more useful?

Hypotheses

The research questions above have led to the following hypotheses:

- Advanced augmented reality techniques can assist in providing and executing commands between the surgeon and laparoscopic camera operator.

- Advanced augmented reality techniques can improve the utility of telementoring systems.

Specific Aims

This research is broken into two specific aims. Firstly, we want to create a head-mounted direction of focus indicator to provide non-verbal assistance for camera operation. A system was created to track where the surgeon is pointing and provides augmented reality cues to the camera operator explaining the camera desires of the surgeon. Secondly, we want to create a hardware / software environment for the tracking of a camera and an object, allowing for the display of registered pre-operative imaging that can be manipulated during the procedure.

Aim 1 is focused on 2D augmented reality techniques in the training or execution of laparoscopic camera navigation. This is covered in Chapter 2. Aim 2 is a hardware / software platform that will support the development of unique augmented reality features using pre-operative imaging and tool tracking for the operating room. This aim will expand into 2.5D and 3D augmented reality and is covered in Chapter 3. Chapter 4 summarizes the work described in this document and the contributions of this research before closing out with future possibilities building on the research that has been completed.

CHAPTER 2 – HEAD MOUNTED DIRECTION OF FOCUS INDICATOR TO PROVIDE NON-VERBAL ASSISTANCE FOR CAMERA OPERATION

Background and Significance

This research aim was designed to see if a non-verbal language, and a system to convey it, could be created to assist in the communication between the surgeon and camera operator. These needed to integrate seamlessly into the actions of the surgeon, providing movement cues from natural motions without leading to an increase in the mental and physical demands on the surgeon. A system was created to evaluate its effect on the basic camera movements in laparoscopic surgery. The objective was to build upon the work already done in the supporting fields, and to evaluate its performance compared to other options available.

Mental and Physical Complexities of Laparoscopy

When approaching from the training side of things, the skills necessary to perform laparoscopic surgery are not simple. The doctor is working with a layer of separation between the operative area and their hands and eyes. Everything that is seen is watched on a monitor in a different location using a camera that is not directly under their control. All of the tools are long and thin to pass through the trocar points in the body and these are the only things touching the internal tissues. Forces need to be gauged without direct contact with the tissues, and the tools have to be operated in reverse because they pivot on the trocar points. It is more taxing to the surgeon, physically and mentally than an open surgery [5]. With this increased

difficulty, one does not want to make things even more difficult. A key to this, and something this aim is focusing on, is helping to reduce potential problems with the camera operator.

The camera operator is most likely using a laparoscope with an angled tip. This causes the view from the camera to go off at an angle from the tip of the tool inside of the body. With this feature, the camera operator now has to perform translations and rotations to get the optimal offset view of the operative area. The camera must be focused on the desired area for the surgeon to perform their duties. If the picture is not being held steady, the surgeon will have trouble also. Multiple studies have also been done on the rotation of the viewpoint. One study found that the performance of the surgeon in cutting and tying decreased as the viewing angle moved away from the horizon of the standing doctor [6]. Another study found the same issue, that the speed of suturing slowed and the errors increased as the camera horizon increased to 90 degrees from the doctor's horizon [7]. It is clear that making sure the camera operator can look at the correct area, hold the camera steady, and maintain a convenient working horizon, will allow the surgeon to operate more effectively.

Training

With these unique skill requirements, additional training is needed for laparoscopic surgery beyond normal open surgical training. As expected, this training needs to be provided by experts in the field. However, at this time, demand for training outstrips the available experts that can provide the training. Some groups have investigated how to foster usage of laparoscopic surgery in rural areas with a different training routine [8]. Another paper has discussed how to handle shortages of all operating room personnel in smaller countries [9].

With these problems, researchers started moving toward training systems that could augment or replace the expert. Some groups focused on coming up with a list of basic skills to teach. One group produced a list of six skills, including moving small objects, placing clips, and suturing [10]. Other groups even outlined some basic tasks that included camera navigation [11].

With a basic set of skills outlined, companies around the world starting building simulators to test those tasks. Systems such as the METI SurgicalSIM, Haptica ProMIS, and Surgical Science LapSim started to see usage. Research groups began to evaluate these systems in different ways. One group focused on showing that training on simpler laparoscopic skills resulted in reduced mastery time on more complex skills such as suturing [12]. Another group showed that training on a videotrainer system resulted in a reduced time until proficiency for the Fundamentals of Laparoscopic Surgery test [13]. Groups then started to try translating skills from the trainers to the operating room. One group moved students from LapSim to a test animal and found the virtually trained students to be faster [14]. Another group even tried using LapSim with experienced surgeons as a warm-up for the doctors to prepare for the operation, and they found the warmed-up doctors to perform significantly better in a long list of metrics they watched [15]. With the usefulness of training on these simulators shown in all of these areas, one other group performed a longer-term study showing that some of the skills learned on the trainers are retained, even after a year of not using them [16].

Even with all of the advantage of the simulators and trainers, the systems themselves have many limitations. Many of them offer minimal flexibility in the design of the training routine. Systems like the ProMIS and METI offer little control over the metrics used to evaluate

the performance of the student at the task, and provide little more than time to completion and number of errors as their feedback in many instances. These metrics have been incorporated in the greater evaluation of the student, but some medical schools still have an expert on hand to watch important things that the simulators are not watching [17]. Other groups have written long articles lamenting the training problems that exist, along with the lack of standards on many areas of training [18]. This has caused some groups to attempt to build their own replacement trainers and systems rather than use the commercial products. One group effectively built a physical version of the virtual camera navigation training from METI. They found the system inexpensive and useful for training, but it does nothing that METI does not do [19].

Operating Room Communication

When moving outside of training and into the operating room, many other issues are present. With a group of surgical staff all working within a small area and worrying about different things, movements and discussions can cause distractions. A group at an academic hospital in Massachusetts did a study that found that problems related to communication, the flow of information, and competing tasks had a negative impact on the performance of the team and the safety of the patient [20]. Another group watched a number of surgeries and found that communication problems cause the most stoppages and errors [21]. A very recent study was performed that watched how often the surgeon took their attention off the task they were performing. They said that the surgeons were frequently distracted and that work needed to be done to allow the surgeon to keep their attention on the patient for a faster and safer operation [22]. With these studies, and many others, it is clear that any work to reduce

the volume of verbal communication in the room and keep the surgeon focused on their task could improve the speed and safety of the procedure.

Robot Assistants

A large amount of research has been focused on removing the human camera operator from the operating room and replacing them with a robot. The three robots most prevalent in studies are the Aesop, LapMan, and EndoAssist. Aesop is primarily a voice-controlled unit, LapMan is controlled by a joystick mounted on the laparoscopic tools, and EndoAssist utilizes infrared tracked head movements and a foot pedal. Many initial studies looked at the usage of robots compared to human operators. An early study found procedures with Aesop to have a steadier view, and a similar operating time as a human [23]. A group out of the UK used EndoAssist in a significant number of procedures, found it to have no major issues, and found that they had faster operations when they used it [24]. Another surgeon out of the UK also found the EndoAssist to be perfectly useful as a camera holder [25]. One other recent study involving the Aesop found that the Aesop went where the surgeon wanted more than the human operator, but the system was a lot slower in getting there [26].

With the various systems found to be at least an adequate replacement for human operators, other researcher set out to compare the systems. One group performed a study in a simulated environment between EndoAssist and Aesop using vertical, horizontal, diagonal, and zoom movements. They found EndoAssist to be faster at translations, especially the diagonals, which cannot be combined in Aesop commands, though both performed the same in zooming [27]. Any complicated movements were faster with EndoAssist. Another study compared EndoAssist and Aesop in a simulated environment. They found the time to perform a complex

movement to be significantly less with EndoAssist than with Aesop [28]. This group also had significant issues getting the Aesop voice control to recognize their commands consistently. Interestingly, another group compared the EndoAssist and Aesop in a clinical setting. In their real-world setting, they only found EndoAssist to be faster in 2 out of 13 parts of their procedure. They concluded that the performance of each system was equivalent [29].

A lot of work has been done in the area of robotic camera operators. However, the systems do have a list of drawbacks. The different options can be cost prohibitive to use. The robots are costly and require setup time for each case. Some of the systems are also slower than humans, increasing operating time. A major drawback is the loss of training time for the camera operator. Under normal circumstances, the camera operator can be learning laparoscopic operations from the expert surgeon while they are being performed. The robot is performing something a learner could be doing.

Pointer Study

One final item applies to the work being done in this aim. A group out of Canada performed an experiment to show that a pointer would be faster than verbal instructions [30]. They had 20 points of interest on a surgical model and had the students touch the points based on the verbal or pointer instructions. They only looked at time to completion, and the pointer was faster.

This aim intends to build upon the value of training in laparoscopic surgery to not only focus on the camera operator, but to look at skills required in that position more than the time it takes to complete a task. It also intends to help alleviate communication problems in the operating room between the camera operator and the surgeon. It could also result in more

efficient movement of the camera than a robot or verbally commanded human camera operator provides.

Preliminary Foundational Work

This section will cover an overview of everything that has led up to this aim and its completion.

Surgical Simulator

The initial goals of this aim were to enhance the METI SurgicalSim VR system used by the Detroit Medical Center in the training of laparoscopic surgeons. The system allows the trainee to utilize approximations of surgical tools in a magnetically tracked environment to perform simple laparoscopic tasks. The tools are placed through holes in an elevated flat surface. No physical graspers or cutters are present on the tool tips, but handles are present that can be actuated to open and close the virtual representations of the tools being used for the tasks.

We were specifically looking at the 0° endoscope training. One magnetically tracked tool represents the endoscope, and a single virtual target appears in the environment. The user must align and orient the camera to the target and hold it in that position. After the hold period, the target disappears and a new one appears. The user is graded on the time to find all of the targets.

The hospital has trouble using the simulator because it does not provide comparable results. The target positions are random, providing the user random difficulty. Because of this, they could not directly compare results between users or even between trials. Within the

constraints of the closed system, they need to at least know the positions of the targets to be able to formulate a difficulty index to normalize the results.

The problem is that the simulator was not giving any of that data, and any software methods we devised were not leading us toward the target positions. We decided that our best line of action would be to build a hardware device to tap into the data stream passing from the Polhemus magnetic sensor and the simulator. As shown in Figure 4, we built custom hardware to tie into the data cable between the METI circuitry and the Polhemus magnetic tracker. This allowed us to capture the data traveling across the cable without the system circuitry detecting that the cable was tapped into.
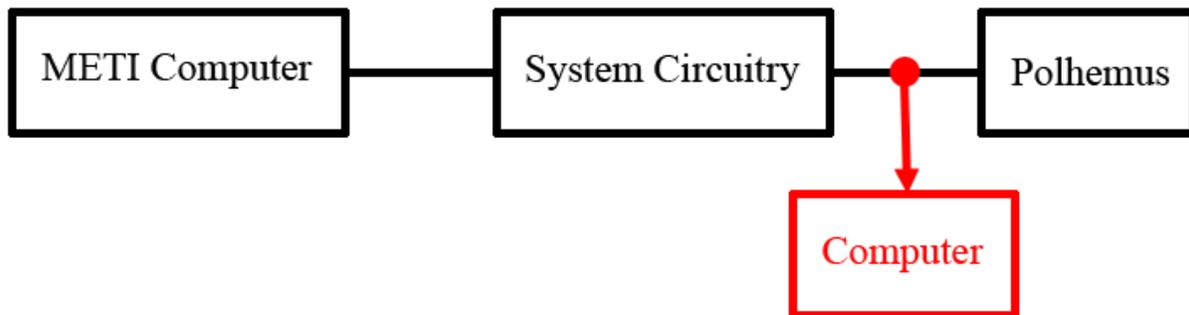
Figure 4: Block diagram of METI simulator and hardware tap

During a trial, the data stream would be saved to the computer so that it could be analyzed later. After decrypting the data using the message format used in the Polhemus documentation, we acquired the position and orientation of the magnetic sensor in the tool. With that data, we wrote an algorithm to determine the position and orientation offset between the magnetic sensor in the tool, and the tip of the tool. We placed the tip of the tool into a divot in the surface of an object. We then moved the tool around in 3D space while keeping the tip of the tool in the same position in the divot. An optimization routine was written to take all of those position and orientation points and try to find a transformation that

would result in the same position for every point. This gave us the position and orientation of the tool, but did not give us any time information. This was a problem because the Polhemus was being queried asynchronously, giving us no time base or time delta to work from. For other training exercises, we would have to build additional circuitry to possibly tap into the METI circuitry to get the scope angle and grasper states. That would also need to be synchronized with the position and orientation data. We would also need to automate the file capturing of the data to another computer.

This work stopped when a METI employee let us know that trial data is stored temporarily on the METI computer before being deleted. After spending considerable time decrypting the file format they temporarily stored the data in, we now had a data stream with position, orientation, and time. This allowed us to pull the data and look at it, just as shown in Figure 5.
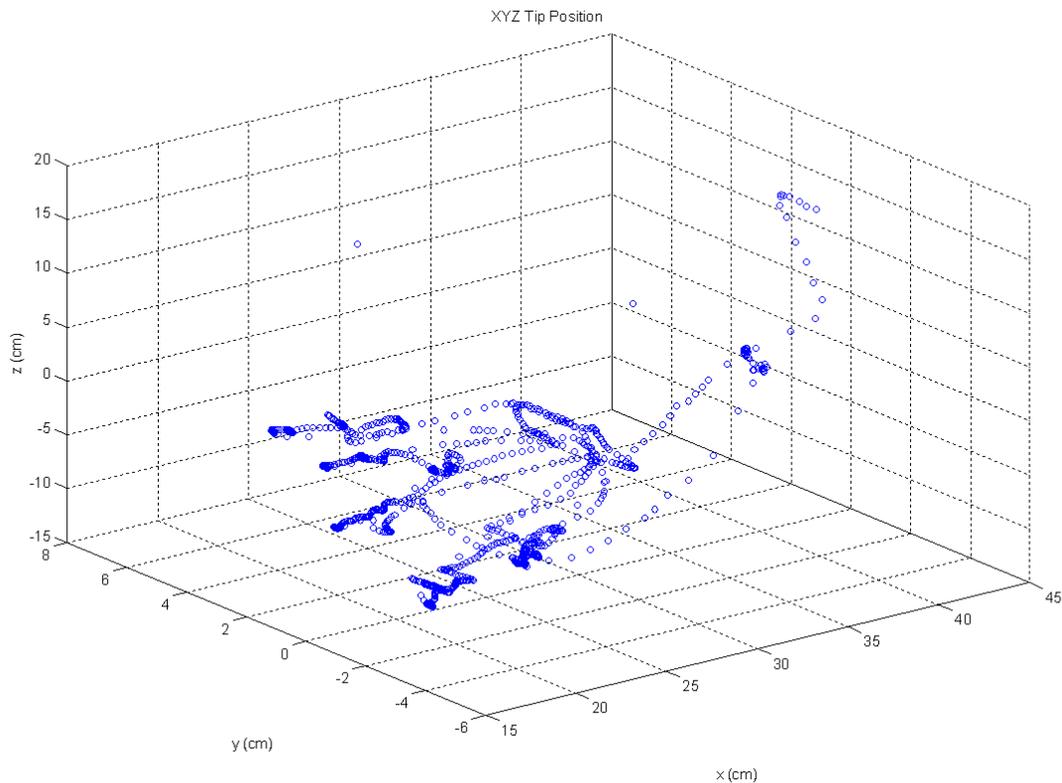
Figure 5: 3D Position data for camera tip movement in a single 0° endoscope trial

Now that we had position and orientation data, we could use it to find where the targets were located. A 2D graph of the data shown in Figure 6 displays the interesting behavior inherent with the endoscope trials. In a trial with six target acquisitions, the user must find the target, hold the camera over the target for a specified period of time and repeat as quickly as possible. Upon seeing this consistent behavior in the data, we wrote an algorithm to search the data set for periods of time, of a length matching the simulator settings, that the user held the endoscope very still. This would then be followed by sudden movement while the user went to find and acquire the next target. The holding period ended by a sudden movement would be our best approximation of the position of the target. This was automated in a computer program that returned the position, orientation, and time of the target acquisition, as shown in Figure 7. Now we had data to be able to determine how far away each

target was from each other and from the starting position to be able to determine some kind of

difficulty index.
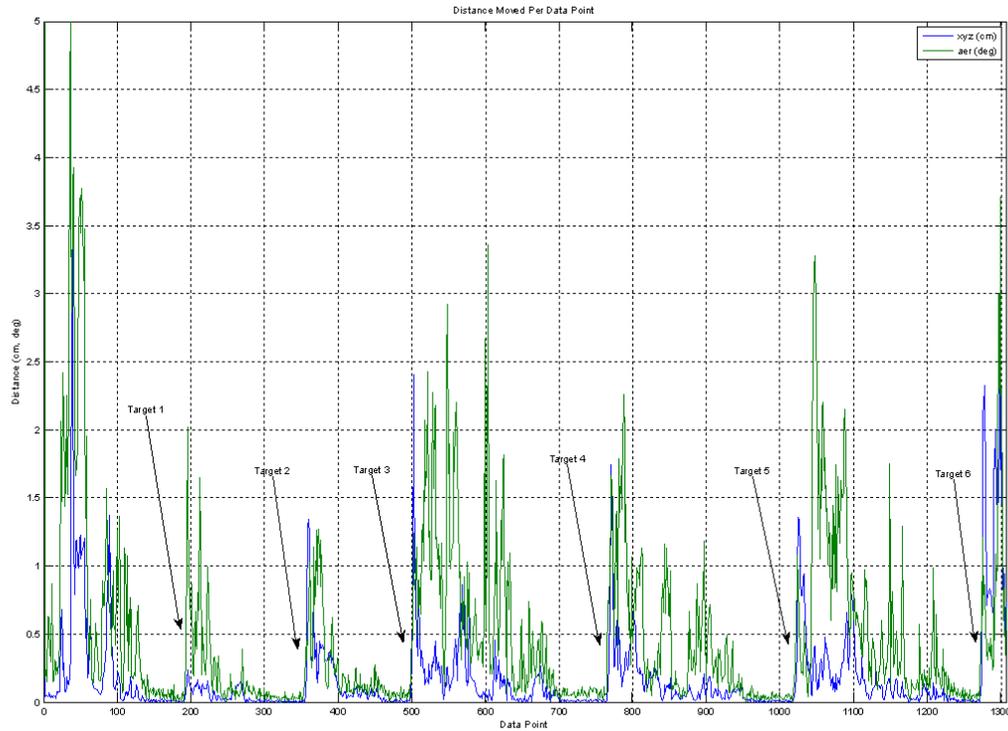
Figure 6: Distance moved per data point in a single 0° endoscope trial
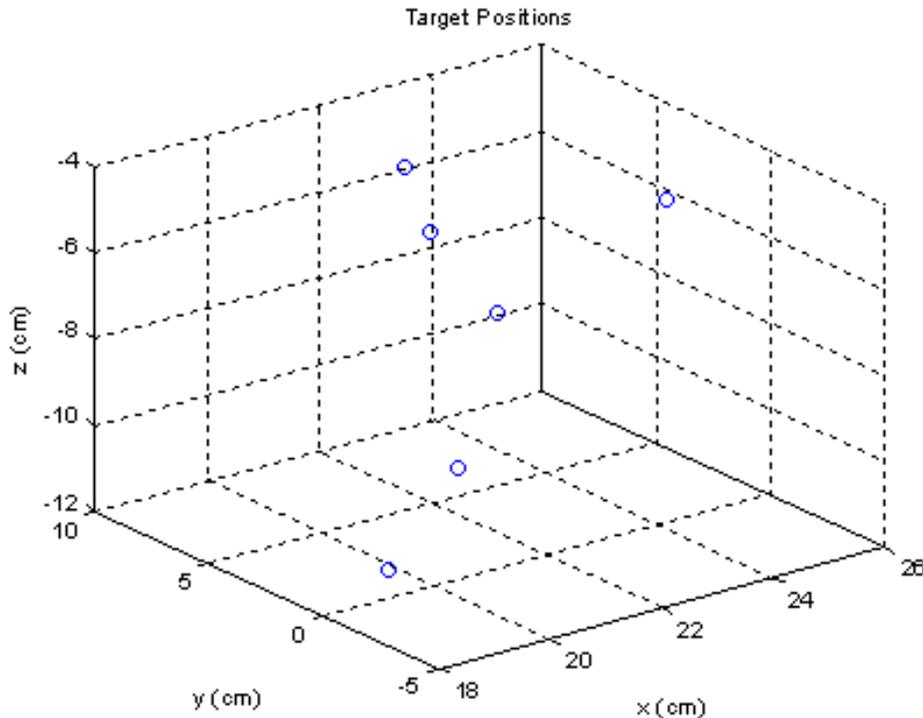
Figure 7: Target positions in a single 0° endoscope trial

In addition to this, we now had additional information for analysis. One search program we wrote determined the phases of the target acquisition. Now that we know the target positions, we can traverse the data with that prior knowledge. Figure 8 shows a graph of the distance from the endoscope to the target. This can be used to determine when the user was searching for the target, when they acquired it, and when they were holding on it. Search is the time from the initial movement spike from the appearance of a new target until the last point at which they were moving away from the target. This works under the assumption that as soon as they found the target they were only moving toward it after that point in time. The hold phase is the place where the target distance is near zero and is being held for the simulator specified time period. The acquisition phase is the time period between the other two, in which the user is moving toward the target until it is acquired. This algorithm provided useful data for most targets. It could break down the times to compare how long it was taking

users to find the targets, how long it was taking to move in toward the target, and how long it

took them to hold still long enough to finish that target acquisition.
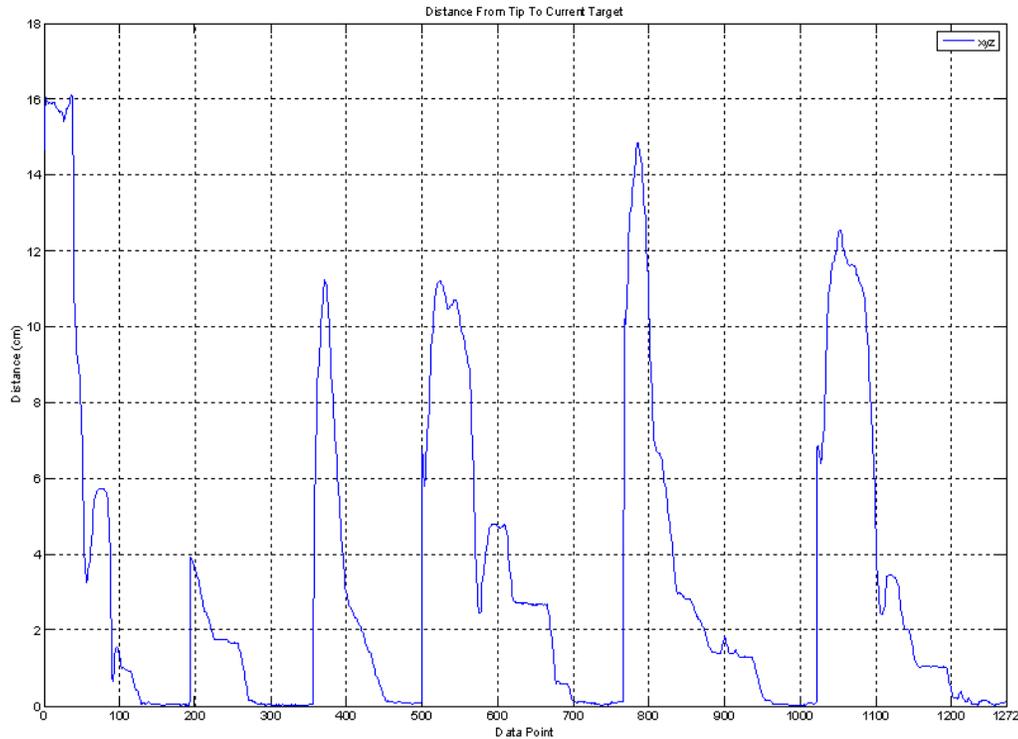
Figure 8: Distance to target for a single 0° endoscope trial

Having the position of the tool during each stage and the time duration of each stage

allowed us to come up with a way of normalizing the data to compare between trials. We could

normalize for the distances, but the randomness in the system brought up other difficulties.

Targets from different trials could be the same distance apart from a previous target, but one

could be within the viewing cone of the camera and the other might not. If a target appeared

within the view of the user, a search phase would be non-existent. However, it the target

appeared outside of the view or behind the view, the user would have to spend time searching

for the target.

After running data for the hospital, we decided that it would be worthwhile to avoid the limitations of the METI simulator and attempt to use something else. We immediately moved over to the other major simulator available in the surgical training lab, Haptica's ProMIS simulator. This device used a plastic human abdomen with cameras along the inside of a large open cavity in the body. Laparoscopic tools with specially dimensioned colored tape on them could be inserted into the cavity and the system was supposed to track their position and orientation optically using image processing calibrated to the tape. Operations with the simulator were also available in augmented reality and virtual reality depending on what you were doing. The major advantage of the system was that we were able to procure a license to use their basic developer tools for the system to create our own tasks for the user to do.

Unfortunately, that was the only advantage. After a long evaluation period, we decided not to go forward with the ProMIS system. The development tools were very limited within the simulator environment and would not allow us to control many things in the environment, nor allow us to acquire the data we wanted to look at. The simulator also had major problems tracking the tools being used. Even after careful calibration and control of the area lighting, significant amounts of positional jitter and complete loss of tracking would occur during use. The system also had frame rate problems that introduced jerky response in the feedback of what the user was doing. Due to the frame rate problems, we even evaluated the platform using the Zeus surgical robot instead of human hands holding the tools. We were hoping the slower moving robotic arms would be less affected than human hands moving the tools. Unfortunately, with the limited working volume inside of the body cavity that could be tracked

by the internal cameras, we were severely limited in the movements we could make with the Zeus. The ProMIS was deemed unusable and abandoned.

Headtracking System

After trying to work within the limitations of these other simulators, it was decided that it would be worthwhile to attempt to build our own simulator program for the endoscopic camera operation task. With this, we would be in control of all aspects of the trials. This then expanded to building a system that could not only be used to evaluate the skills used in laparoscopic camera control, but could also be used to potentially assist in the operation of the camera, or at least provide a new avenue for discussion between the surgeon and camera operator. Combined with difficulties we witnessed in the operating room related to breakdowns in communication between surgeons and camera operators when it came to where the camera should be placed, we thought that would be a good avenue to follow.

Since verbal communication is the normal interaction between surgeon and camera operator (human and some robotic), we felt we would be able to augment that with a non-verbal communication method. After looking at many options, it was determined that we should be tracking where the surgeon is looking at the video feed to be able to tell the camera operator where they should be centering the camera. We evaluated many eye and head tracking systems on the market, and all had major limitations. Due to the large area that the surgeons move around in, almost all of the eye tracking systems on the market would not be able to keep track of the doctor, let alone be able to determine where the doctor was looking. Eyeglasses and safety glasses in the operating room also caused major problems even if the doctor did not move around. These systems and other head tracking systems were optically

based and had problems with people wearing surgical masks on their faces, along with interactions with the operating room lighting.

Hardware

To combat all of these issues, it was decided to stick with head tracking and work with a 3D Guidance trakStar magnetic tracker. This system works within a magnetic field large enough for the surgeon to walk around normally in; it would not be affected by what the surgeon is wearing or the lighting in the room either. It would allow four separate items to be tracked, so we could track more than the surgeon's head. Finally, it would have an 80Hz data acquisition rate, significantly faster than the Polaris IR tracking system we use as a part of Aim 2. This would allow us to start looking at velocities and maybe even accelerations of the positions and orientations.

A hardware and software system needed to be developed that could track the surgeon's head movements and place a crosshair over an endoscopic video feed. Figure 9 shows the hardware system architecture that was devised.
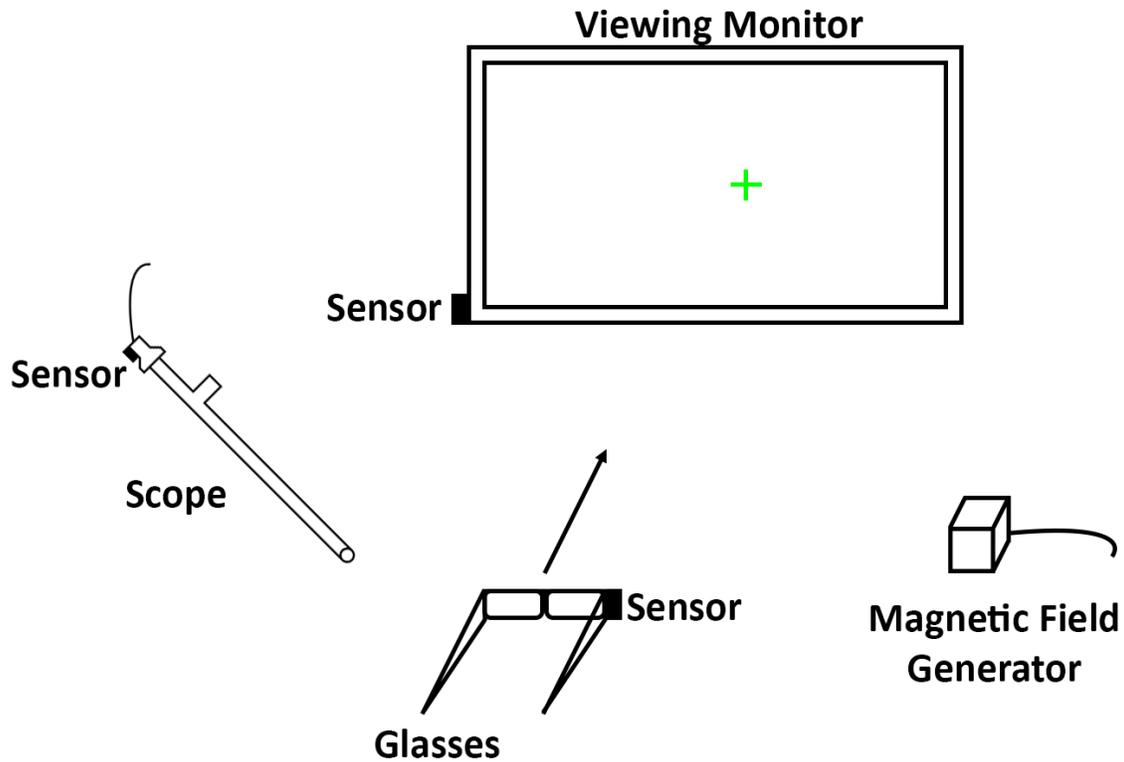
Figure 9: Headtracking system hardware diagram

The magnetic field transmitted would be centralized in the environment. A sensor would be attached to the surgeon's head, along with a sensor on the viewing monitor where the endoscopic video feed is played. With the position and orientation of the surgeons head and the position and orientation of the monitor, we could determine where the line projecting from the surgeon's head intersected the monitor screen and know the exact point of interest. With additional sensors, we could place a sensor on the endoscopic camera to provide additional data for evaluating the performance of the camera operator.

We used the Model 800 sensors with the system. They are 7.8mm x 7.8mm x 19.8mm in dimensions, small enough to be placed where we want them. The mid-range transmitter with our system has an effective range of 78cm with the sensors that we used. This was

enough range to handle our study, but for a practical operating room application, the wide-range transmitter would need to be utilized with its 2.1m range.

System Architecture

With the availability and the maturity of the libraries for the trakStar system within MATLAB, it was chosen to work in that environment. With the environment and hardware set, we formulated a software architecture plan that is shown in Figure 10.
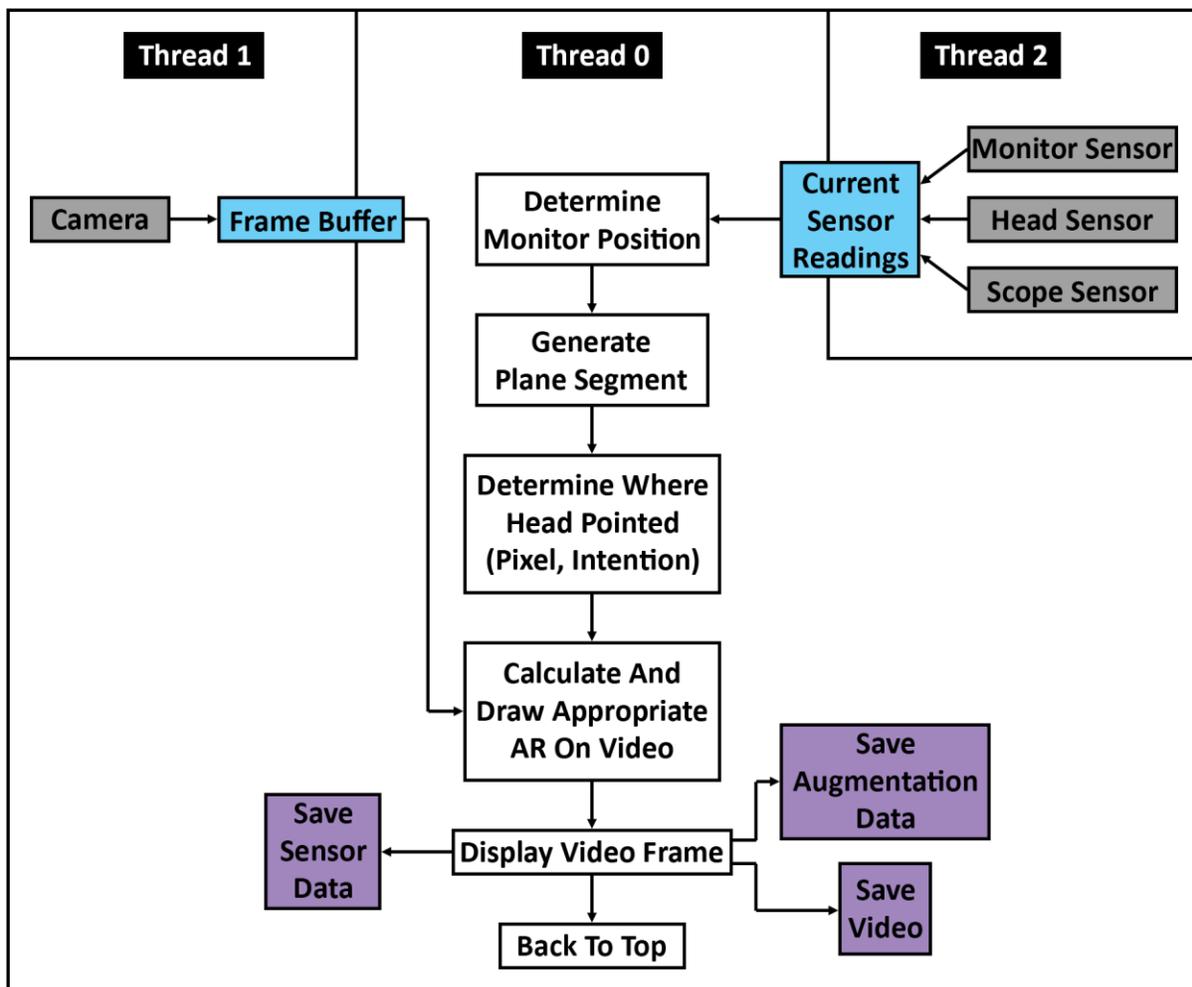
Figure 10: Headtracking system software architecture

With the camera providing frames at 30Hz and the tracking system providing data up to 80Hz, we decided to collect the data asynchronously and design the code in a multithreaded

fashion. Threads 1 and 2 are the worker threads of the system. They interface with the hardware and collect data from them as fast as the hardware can provide it. Thread 1 interfaces with the camera on the laparoscope and holds each frame in a frame buffer in memory. Thread 2 interfaces with the traskStar to get the sensor values for the monitor sensor, head sensor, and scope sensor. These are also stored in memory to be available for the main thread.

The main thread, Thread 0, handles most of the processing for the system. The system initially checks the sensor readings to find the position and orientation of the monitor that is displaying the video feed from the camera. Based on prior measurements, it generates a plane segment that represents the monitor in space. It also checks the frame buffer for dimensions to know the resolution of the video that will be played on the screen. The system will then read the sensor for the head of the surgeon and project a line segment out from that position and orientation to indicate where the doctor is pointing. It will also watch the roll of the sensor to see if the doctor is tilting their head, and the change in position of the sensor to determine whether the doctor is leaning forward or back.

With that information, the program calculates whether the line segment from where the doctor is pointing intersects with the plane segment representing the monitor. If the doctor is pointing at the screen, it will return the (x, y) position on the monitor and then segment the screen up into as many pixels as are in the video feed and determine the exact pixel in the video that the doctor is pointing at. Figure 11 shows a representation of that process. The white space is the trackable area of the system and the circle with the line pointing out of it is the head of the surgeon and where they are pointing. The blue box is the

position and orientation of the monitor in the scene. As can be seen, the doctor is pointing at

the screen and the program has determined that they are pointing directly at pixel 325x256 in

the 640x480 video feed used in this instance.
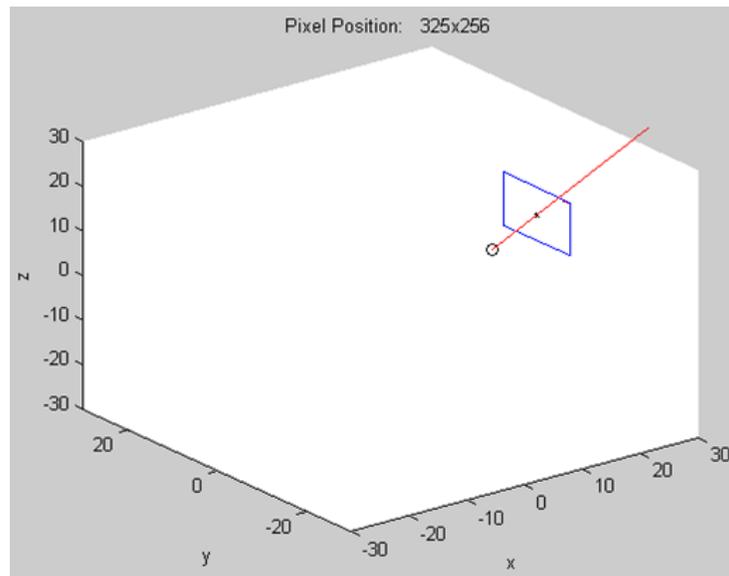
Figure 11: MATLAB plot of head pointing at monitor and reading back intersected video pixel

After knowing the pixel the doctor is pointing at and their intentions in leaning or tilting

their head, the appropriate augmented reality cue is drawn directly on the frame data and

outputted to the screen. The program then saves all of the sensor data for that frame, saves

the viewed pixel and intentions of the surgeon and saves the video feed. That process is

repeated until program termination.

The system of constantly checking the position and orientation of the monitor, the

surgeon's head, and the scope works well for an operating room application. Because the

surgeon is moving around and the equipment in the room sometimes moves around, being able

to always determine where everything is allows the system to continue working no matter what

moves. An initial reading was also taken of the surgeon's head to provide a comparison to

determine whether they were leaning forward or back, or tilting their head, for operating the

system of cues. We set up a foot pedal with the system to reset that value with their current position in case they needed to move around in the operating room. If a foot pedal is not desired, other input methods are available for them, or someone else, to operate.

Augmented Reality Cues

With the ability to display video and determine where the surgeon's head was pointed, we needed to create a language of intention between the surgeon and camera operator. As a basis for the system, we needed to first determine the degrees of freedom of the camera and the possible motions that could be described. A laparoscope inserted in a surface is shown in Figure 12.
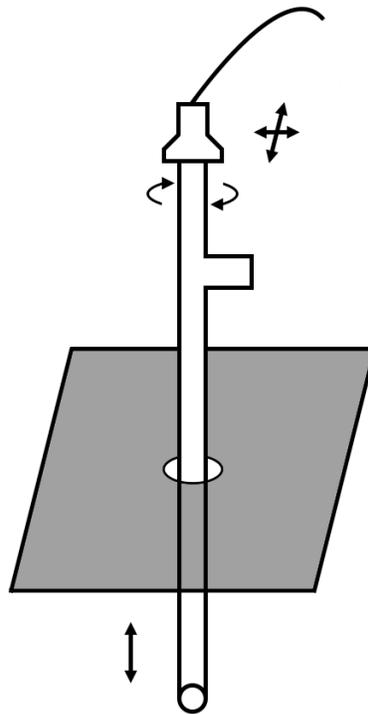
Figure 12: Laparoscope with degrees of freedom indicated

The camera is placed through a trocar in the body and has a limit imposed on its degrees of freedom. With that in mind, the camera operator has control of where the camera is

pointed (its positioning), how far in or out it is (its zoom), and its rotation around its central

axis. After a good deal of prototyping and evaluation, we came up with the symbols in Figure

13.

Zoom
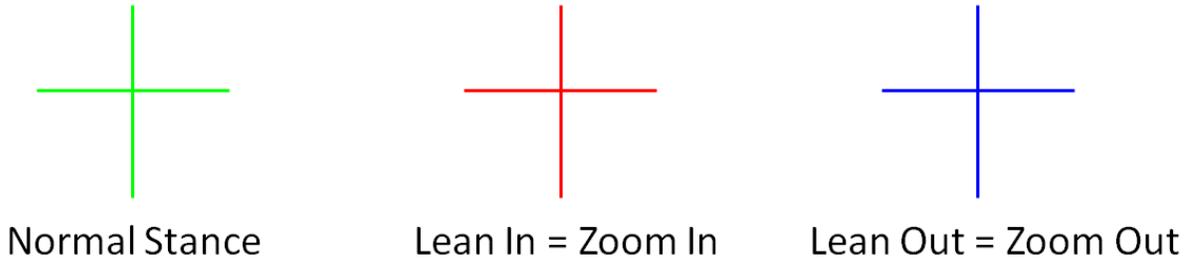
Rotate
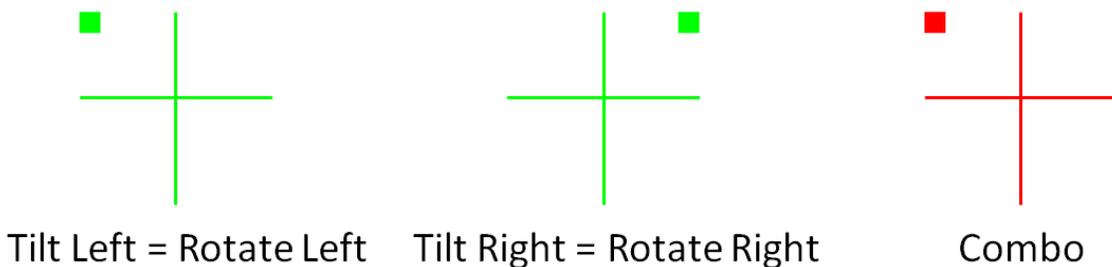
Figure 13: Example crosshair symbols for zoom and rotate operations

This set of symbols all involved a simple crosshair overlaid on the video feed where the

surgeon was pointing. The system would track the direction of interest and display the

crosshair in real-time on the video for the camera operator to know where the surgeon wants

the camera centered. For zoom levels, we needed to handle three states: no zoom required,

zoom in, and zoom out. That was conveniently covered by the three additive primary colors,

red, green, and blue. Green would indicate that no zooming was necessary, while red indicated

a desire for the camera to be zoomed in, and blue to be zoomed out. This was extended with a

dot indicator to communicate a desire to rotate the viewpoint. The clearest indicator we could

find was to place a dot in the upper left if there was a desire to rotate the top of the view to the left (counter-clockwise), or place a dot on the right to rotate the top to the right (clockwise). If desired, all three of the commands, position, zoom, and rotate, could be given simultaneously. All of these symbols were chosen to be as minimalist as possible to reduce the complexity of interpreting them and to make sure that they covered as little of the screen as possible to not block the surgeon's view.

As is also written in Figure 13, we needed to determine how the surgeon would input these commands. Since they are not using voice and they probably do not have any hands free, or want to use their feet, we needed to come up with gestures for the head that we were already tracking. Positioning is already taken care of just by where the doctor is pointing. Zoom level was most intuitive when we used a leaning system. Leaning the head toward the monitor indicated zoom in, and leaning out indicated zoom out. The amount of leaning required to trigger the change in state is user configurable, but a number of engineers and surgeons that used the system found a value of 8-10cm to be the most convenient. This would keep the system from being triggered just from normal movement but would not require the surgeon to be making uncomfortable movements to trigger the changes. The rotation changes were found to be most intuitive when the head was tilted. We found a good balance with a 10-15 degree tilt trigger that needed a tilt to the left for a rotate left and a tilt to the right for a rotate right. If the surgeon needed to move around at any time, the initial states for the zoom and rotating could be reset using an alternative input device.

Experimental Design

With a functional system, we then completed the experimental design for what we were going to evaluate with the system. Taking into account everything we have previously mentioned, we wanted to evaluate, within the operation of an endoscopic camera, whether a head tracking based system would be faster and more efficient than verbal commands for a user unfamiliar with the task. Even though they are of different utility in the operating room, we wanted to evaluate all of the motions of the scope. Translation and zooming are a common occurrence while operating a laparoscopic camera. Rotation of the scope is not. The camera operator would normally adjust the horizon once so that the surgeon has the easiest viewpoint in reference to the directions they need to move their tools. If something special requires a horizon adjustment, the camera operator will roll the camera, but it is not as common an occurrence as the other two motions. We wanted to set up a grid of targets to acquire that could be identified for position, zoom, and roll. The surgeon would know the grid layout, the target pattern, and the order of target acquisition. The camera operator just follows the surgeon's directions, whether they be crosshair or verbal. A series of targets would need to be acquired with the desired position, zoom, and rotation. At a minimum, we would be evaluating the time taken and how much movement was made with the camera, breaking it down into the individual movement skills (position, zoom, rotation).

For the experimentation, we decided to use the laparoscopic testbed used by the lab group for robotic surgery. It is a trapezoidal box with an open central cavity that had an opaque side with entry ports on it. Laparoscopic tools can be inserted into these ports and the user

cannot see into the cavity from that vantage point. It can be seen in Figure 14. If looking at the leftmost picture, the box has an internal width of 11.5", depth of 13", and height of 7.25".
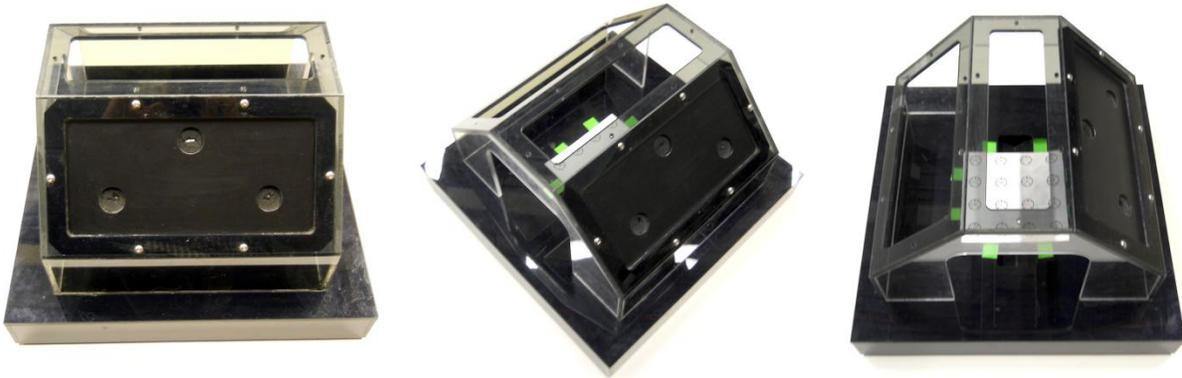
Figure 14: Scope testbed from three angles

We then needed to decide the scope to use for the experimentation. The specific scope used is shown in Figure 15.

Figure 15: 10mm, zero degree scope used in study

In all of the laparoscopic surgeries we have attended over the years, the surgeon always used an angled scope, most commonly a 30° one. With an angled scope, the camera is not looking directly out of the scope, it is looking out of the end of the scope at a certain angle. This adds some more freedom to look around inside of the body and get different angles at objects, but it increases the complexity of operating the scope. For our experiment, all of the users were going to be novices that have never navigated with a scope before. To eliminate one more thing that the user needed to learn that had no specific effect on what we were looking at in the experiment, we decided to stick with a 0° scope. Since we had a large enough opening

for the camera in the testbed, we chose a 10mm scope over smaller options because the picture should be clearer.

The experiment then needed specific targets to find in the environment. We ran many different types of targets until finally deciding on the target shown in Figure 16.
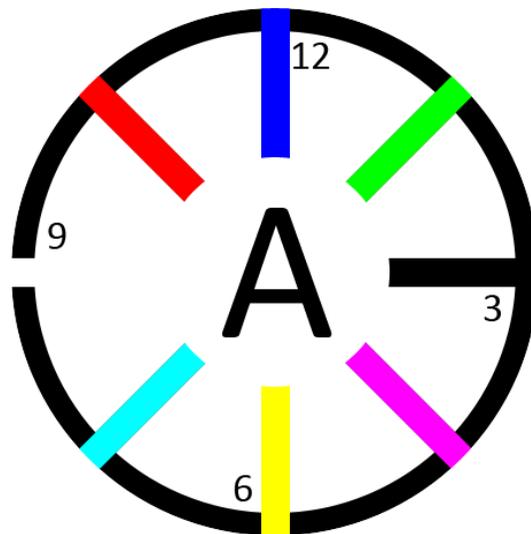
Figure 16: Target A in experiment

The target design was iterated to this based on a few requirements. Since the test subjects were expected to not know the environment and the individual running the trials was not a medical doctor, an easily identifiable symbol was needed. Multiple copies of the symbol needed to be present in the environment and they needed an easy and quickly identifiable way of being specifically found. They also needed an easy way to show how zoomed in the camera was, and exactly what rotation it was at. This was also limited by the 320x240 resolution and the color quality of the endoscopic camera.

With these limitations, we found that for our setup, a 4x4 grid of 0.625" diameter targets worked the best. They were arranged with a 1.375" spacing. These were lettered A through P. With this many targets, they could be printed large enough to make the letter easily

visible even when the camera was zoomed all the way out. The outer ring could be used as a reference for zoom level. In addition, the eight marks around the target could be used to decipher the rotation of the camera. Just as with the crosshair, the colors were chosen as the eight colors furthest apart from each other in the RGB spectrum to ease the identification of each color. For letters that could be interpreted from multiple angles, such as H and I, the color-coding, or the clock times on the target still indicate what your rotation is. All of these options ensured that the person running the experiment would have a reference for the position and orientation of the camera at all times. The video feed for the experiment looked like Figure 17.
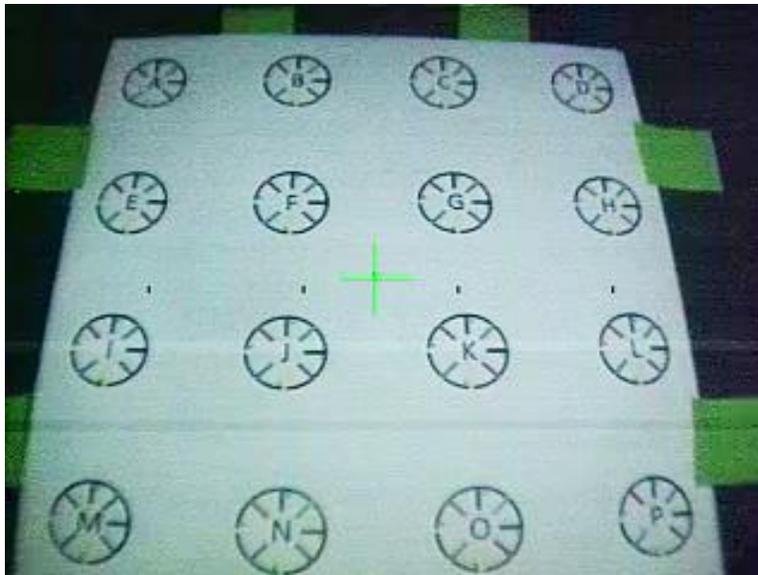
Figure 17: Video feed of experiment – origin position with no commands

We evaluated many different ways for the operator of the system to determine whether the user had reached the correct position, zoom, and rotation. The automated, computer-based methods were unfeasible for accuracy and for processing time in MATLAB. We ended up deciding that all trials would be administered by a single person. Small indicators, in the form of augmented reality, were added to the video feed to help that person. A center pixel was

labeled so that the experimenter could immediately identify when the center of the target, the letter, was directly in the center of the screen. In a horizontal axis from that center were two sets of hash marks. They demarcated three zoom zones. When the entirety of the target was within the interior hash marks, the target was zoomed out. When the target ring was outside of the inner hash marks, but inside of the outer hash marks, the target was partially zoomed. When the target ring was completely outside of the outside hash marks, the target was zoomed in. Finally, for rotation, the desired angle was denoted by color, and the experimenter commanded the user to rotate the target until that color bar was vertical in the 12 o'clock position.

For the experiment, a path distance of four targets was agreed upon. In our testing, four targets was long enough to give us adequate data to work from, but short enough to keep the subject interested in the operation. In all of the trials, the procedure for the user is to start at an origin point, in the center and zoomed out. A target is acquired in position, approved by the experimenter, is zoomed into, approved, is rotated, approved, and then moved back to the origin before moving to the next target. The experimenter is noting all of the approval times with a key press. After four target cycles, the trial is over. We ran many test trials to work out the visual tolerances for the experimenter on what an acceptable position, zoom, and rotate were. We worked on a plan of being able to hold the target in the center without moving around, being able to hold the ring of the target in the correct hash mark zone without moving into other zones, and being able to hold the correct rotational mark at 12 o'clock without moving the target around as acceptable.

Target paths were randomly assigned to trials using an algorithm that ensured that the optimal distance to travel, the zoom level changes, and the rotations needed were all equivalent over the four target trial. We wanted to ensure that every trial was equivalent in overall difficulty and directly comparable to each other. We had the experimenter time stamp approved steps of target acquisition so that we could break the trial down into its components. With timestamps and the data streams, we would be able to use the approval times to determine where the targets were in position and orientation to calculate out optimal distances of the targets from each other along with the time taken to do each individual operation of the camera manipulation.

We randomly assigned whether the user doing the trials would be doing all of them with the crosshair, or with verbal commands.   We based the verbal commands on the limitation that the user did not know the environment, and also on the command set of the Aesop robot used for camera manipulation in the operating room.   These verbal commands were limited to "move (left/right/up/down)", "zoom in/out", "rotate left/right", and "stop." We ran half of our users with crosshair, and half with verbal. We also brought in expert surgeons from the Detroit Medical Center to try the system out using both commands.

We ran through guidelines for how the experimenter could discuss the experimental tasks before the trials, and allowed the user to practice with the system until they indicated they were comfortable with the commands from the experimenter and comfortable with the operation of the laparoscopic camera.

Hypotheses

With segments evaluating different camera motions and the system looking at different points of interest, we have formulated multiple hypotheses for this experiment. The crosshair system should have the largest time advantage in the translational target acquisitions. That motion has the largest difference between the crosshair and the verbal. While zoom and roll have discrete instructions from the crosshair and verbal commands, the translation is closer to an analog system with the crosshair. The crosshair can immediately tell the camera operator where the surgeon wants to go while the verbal commands, like left and up, still leave the camera operator guessing as to the final destination until it is reached.

We expect the time results on the zoom and roll portions to be close to one another. As stated, both the crosshair and verbal systems have discrete commands for these motions. The operator can only move their head so fast comfortably and they can only speak at a limited speed. Adding those up, we expect that the crosshair system should be noticeably faster overall for completing the 4 target trials.

We expect the economy of movement results to mirror those of the time to completion. With the expected advantage in the translational aspect of the trials, we expect to see a slightly lower overall movement of the camera to complete the trials. The zoom displacements should be relatively close due to the expected lower difficulty of that task. The roll may be problematic. We expect the camera operators to be erratic in trying to keep the target centered during the rotation. Users may struggle to make the correct mental translations for operating the camera as the angle of the camera differs from their reference orientation.

Results

Upon the completion of our subject testing, we were left with hundreds of megabytes of data points, and gigabytes of trial videos. 9 subjects performed 5 trials of the crosshair system and 9 subjects performed 5 trials of the verbal system. 5 surgeons completed a 7 trial mixture of verbal and crosshair experiments. The normal subjects were all students at the university that were approached at random to see if they would like to participate. All students were between the ages of 18 – 35 with no surgical experience. The surgeons were individuals who work with our research group but were unfamiliar with the experiment. They range in age from their 30's to their 60's, and were present to provide an alternate perspective on the system.

All of the following results work from a breakdown of trials into 16 different stages. Each of the four targets has a stage where the target is being acquired from translational movement, a stage where the target is being zoomed in, a stage where the target is being rolled to the desired angle, and a stage where the camera is being returned to the origin point.

Time to Complete Tasks

We will start by looking at a box plot of the overall time to completion.
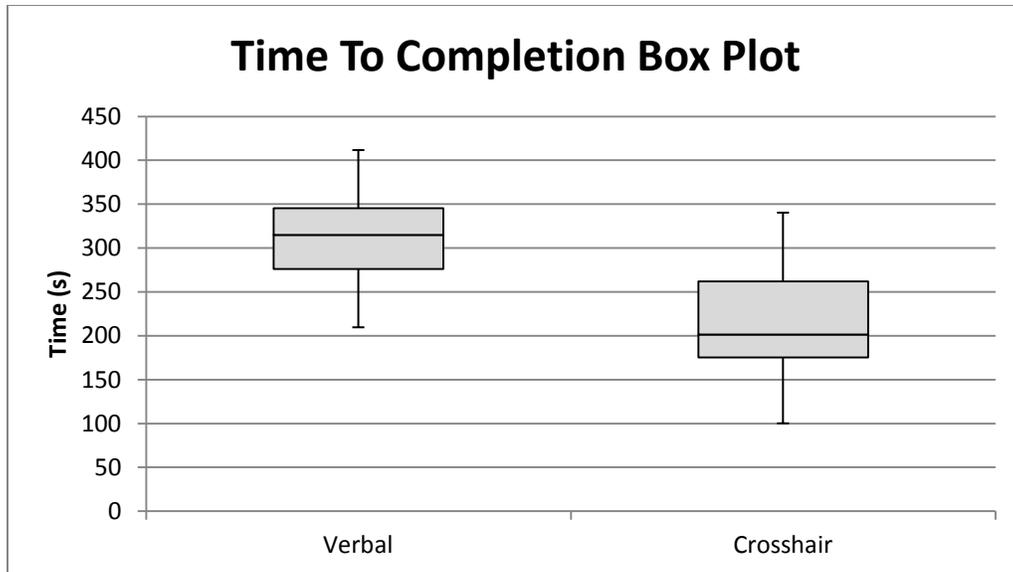
Figure 18: Box plot of time to completion (Whiskers are 95% confidence interval)

Figure 18 shows a box plot of the time to completion data for verbal and crosshair trials where the whiskers are the 95% confidence interval. Verbal trials had a median of 315s, a lower quartile of 276s and an upper quartile of 345.30s. The confidence interval ranges from 209.64s to 411.69s. The crosshair trials had a median of 201.45s, a lower quartile of 175.13s and an upper quartile of 261.81s. The confidence interval ranges from 100.18s to 340.36s.

We will move on to looking at all of the individual data points by first looking at the segments of the trials before moving to the overall results. All of the following graphs have the data from all trials collected and sorted from fastest time to slowest time. This starts with the total time to acquire all four targets.
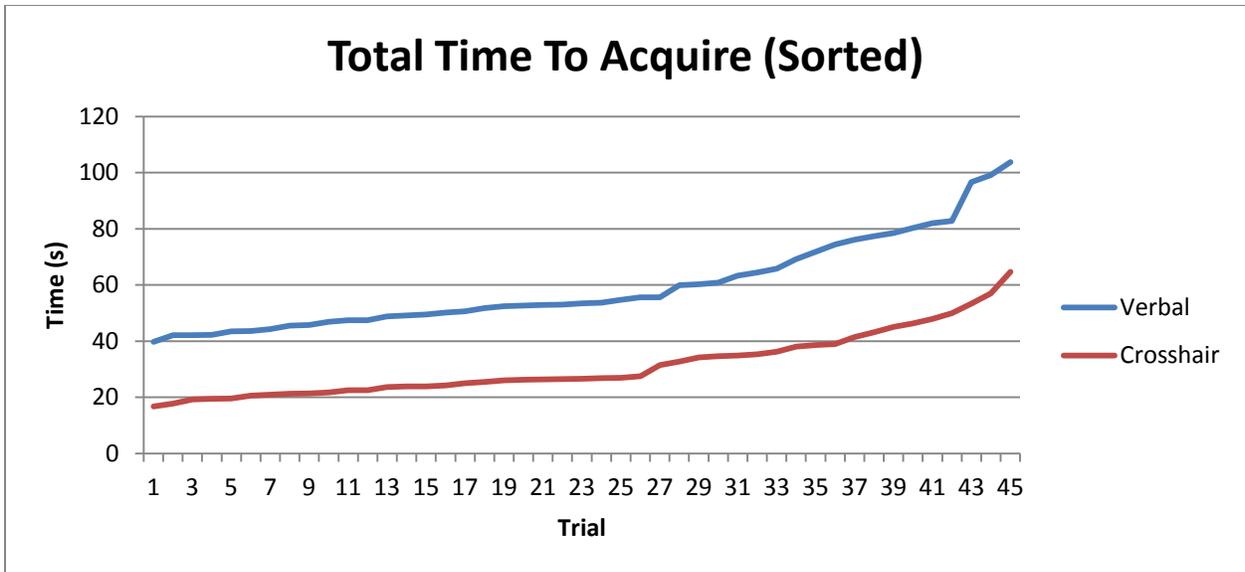
Figure 19: Graph of total time to acquire targets, sorted from lowest to highest

Figure 19 shows the total time to acquire all of the targets sorted for clarity. Verbal had a mean of 59.59s with a standard deviation of 16.25s. Crosshair has a 47.5% lower mean at 31.27s with a standard deviation of 11.40s. Student's t-test returned a p-value of $2.73*10^{-15}$.
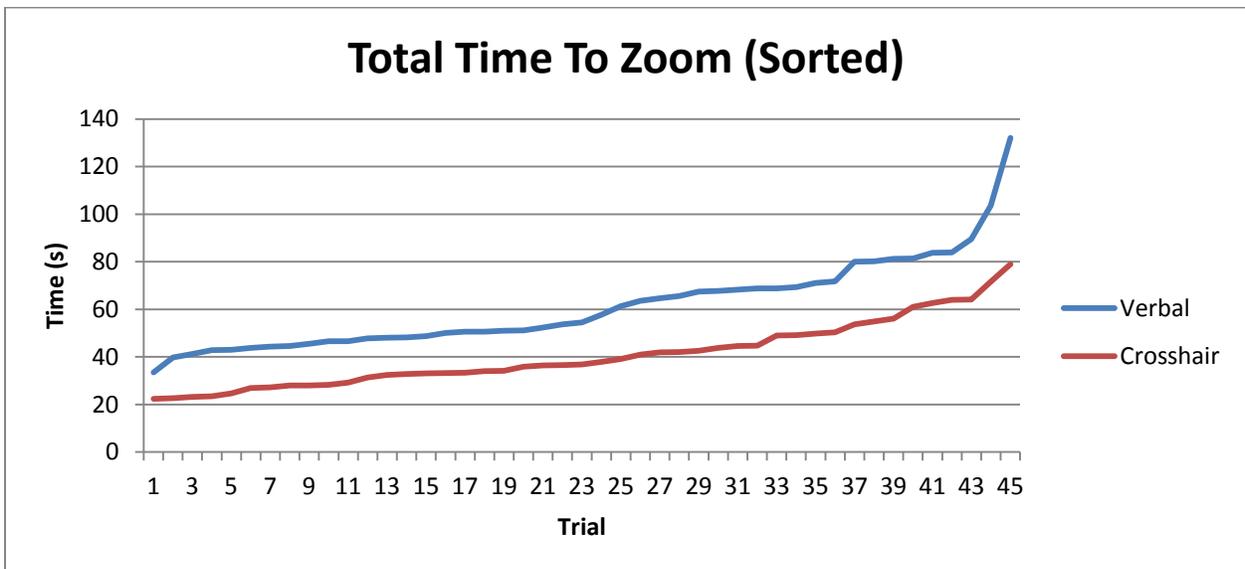
Figure 20: Graph of total time to zoom into targets, sorted from lowest to highest

Figure 20 shows the total time to zoom into all of the targets sorted for clarity. Verbal has a mean of 61.33s with a standard deviation of 19.04s. Crosshair has a 33.4% lower mean at 40.83s with a standard deviation of 13.84s. Student's t-test returned a p-value of $8.57*10^{-8}$.
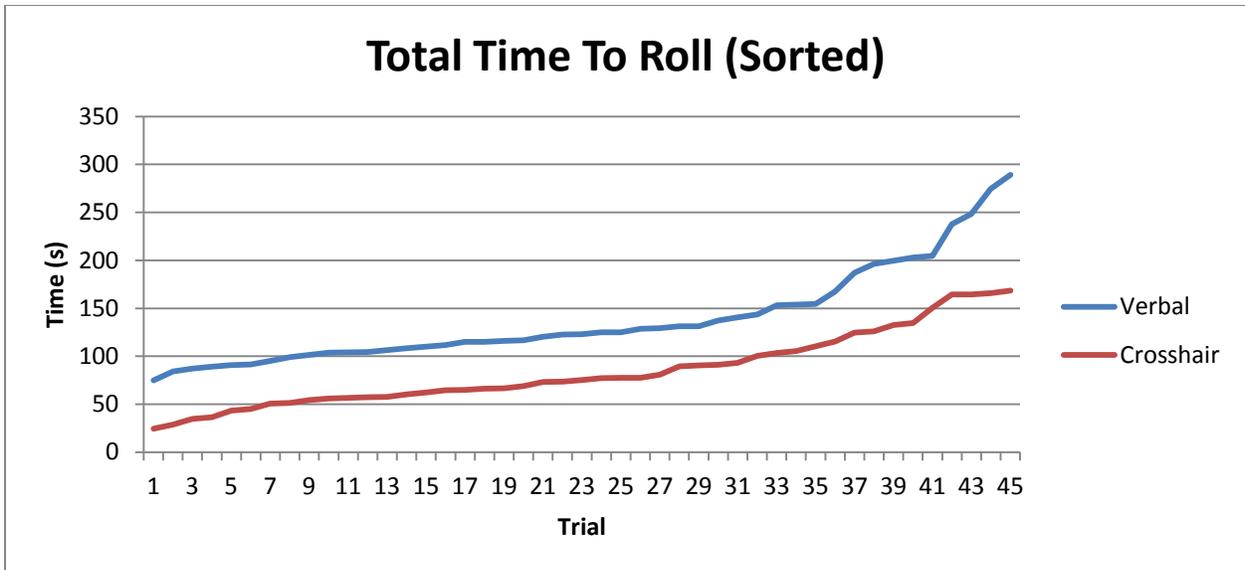
Figure 21: Graph of total time to roll targets, sorted from lowest to highest

Figure 21 shows the total time to roll to the desired angle for all of the targets sorted for clarity. Verbal has a mean of 138.92s with a standard deviation of 51.02s. Crosshair has a 39.0% lower mean at 84.79s with a standard deviation of 38.72s. Student's t-test returned a p-value of $1.8*10^{-7}$.
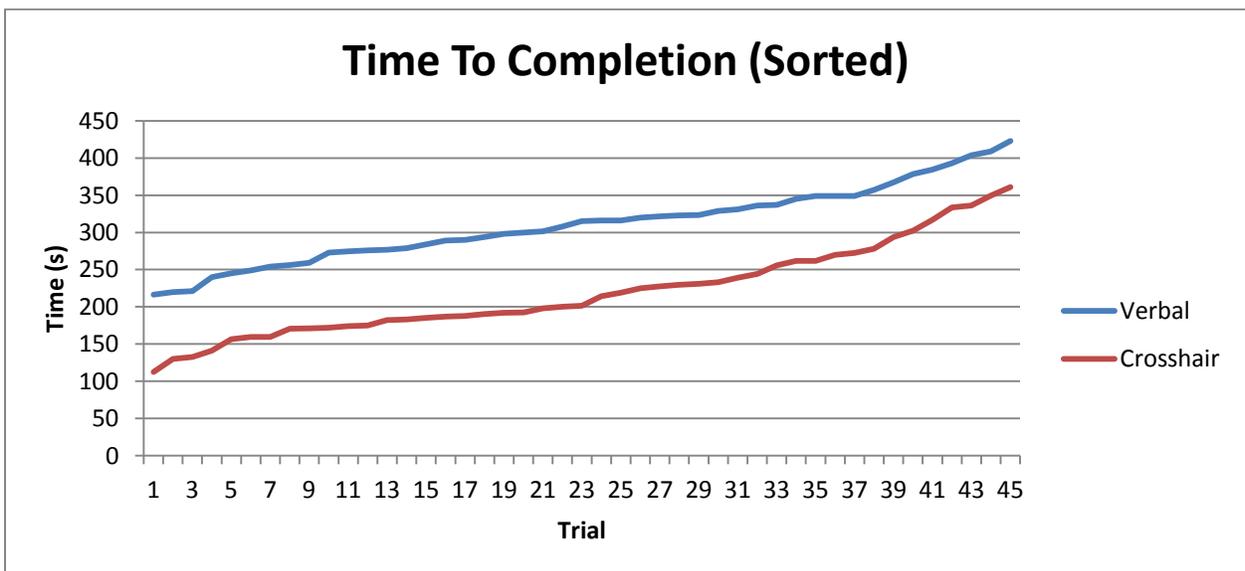
Figure 22: Graph of total time to completion, sorted from lowest to highest

Figure 22 shows the total time to completion for the trials sorted for clarity. Verbal has a mean of 310.66s with a standard deviation of 51.54s. Crosshair has a 29.1% lower mean at 220.27s with a standard deviation of 61.27s. Student's t-test returned a p-value of $3.41*10^{-11}$.

On all counts, the crosshair system appears to have a noticeable reduction in the time it takes to complete the task when compared to the verbal system. The 90-second difference on average for the total time to completion is actually more than we predicted. The acquisition, zoom, and roll data do follow what we initially hypothesized. The target acquisition step takes almost half the time with the augmented reality cues than it does with the verbal. It at least confirms that being able to point directly at your desired outcome gets the message across much more quickly than having to describe how to get there. The zoom and roll data is tighter than the acquisition, but the 30-40% drop in time required is still significant.

The other interesting observation from the data is that the profile of the two curves is very similar. On a whole, from the best users to the worst, there is a fairly consistent time advantage to the augmented reality system. We find it interesting that the best users of the verbal system get close to the augmented reality system in the zooming stage. The zooming in process is probably the simplest of the activities, so it would make sense that regardless of instruction, abilities should even out. On the reverse of that, it appears that the worst users are even worse with the verbal system. The acquire, zoom, and roll data show an increase in time among the worse users in the verbal system that outpaces the slope of the worst in the crosshair. That may imply a comfort level among the less skilled with having a constant direction on the screen. It may just be a couple poor trials from users that ran the verbal system.

Even with the large percent differences between the data sets, the standard deviation values are relatively large. None of the data sets are within one standard deviation of each other, but they are all within two. The box plot for the total time to completion clearly shows no overlap in the interquartile range, though the whiskers at a 95% confidence level cross each other. The student t-test results do back up the differences in the results between the verbal and crosshair systems. The p-values returned in each situation are significantly smaller than even a 99% confidence level would require.

Laparoscope Positional Displacement

This section covers the amount of movement that the tip of the laparoscopic camera made during the trials.
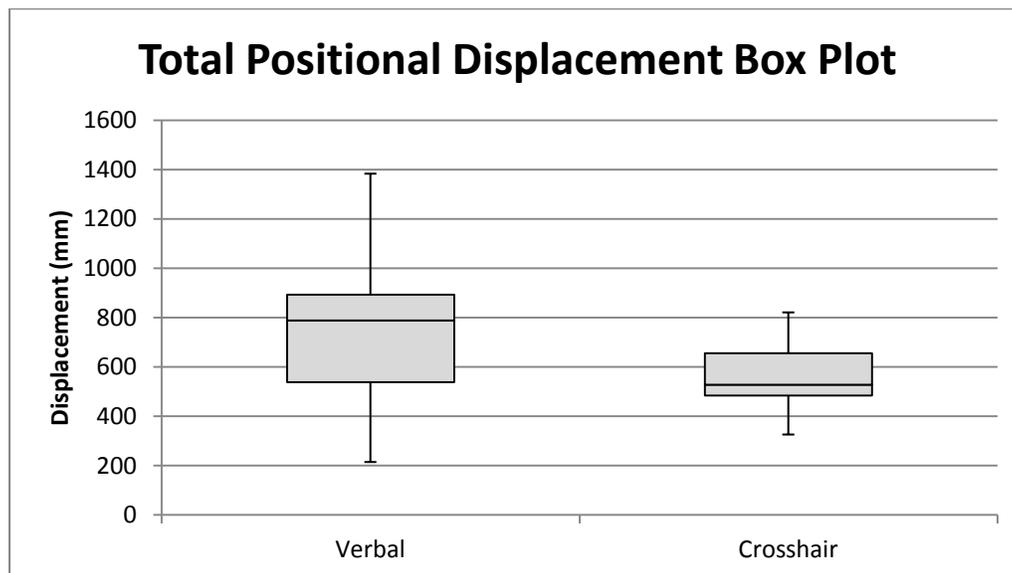
Figure 23: Box plot of total positional displacement (Whiskers are 95% confidence interval)

Figure 23 shows a box plot of the total positional displacement data for verbal and crosshair trials where the whiskers are the 95% confidence interval. Verbal trials had a median of 787.72mm, a lower quartile of 537.66mm and an upper quartile of 892.32mm. The confidence interval ranges from 214.62mm to 1383.46mm. The crosshair trials had a median of

527.47mm, a lower quartile of 483.67mm and an upper quartile of 655.00mm. The confidence

interval ranges from 326.04mm to 821.15mm.
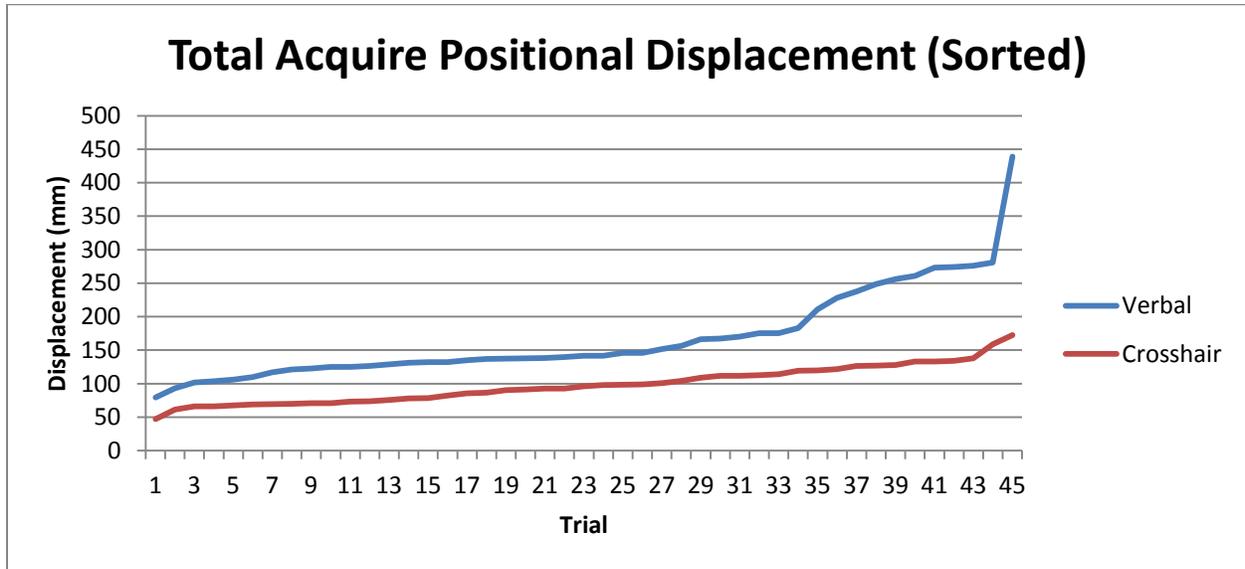
Figure 24: Graph of total displacement during acquisition stages, sorted from the lowest to the highest

Figure 24 shows the total positional displacement that took place during target

acquisition for each trial. Verbal has a mean of 168.64mm with a standard deviation of

68.94mm. Crosshair has a 41.7% lower mean at 98.37mm with a standard deviation of

27.65mm. Student's t-test returned a p-value of $9.3*10^{-9}$.
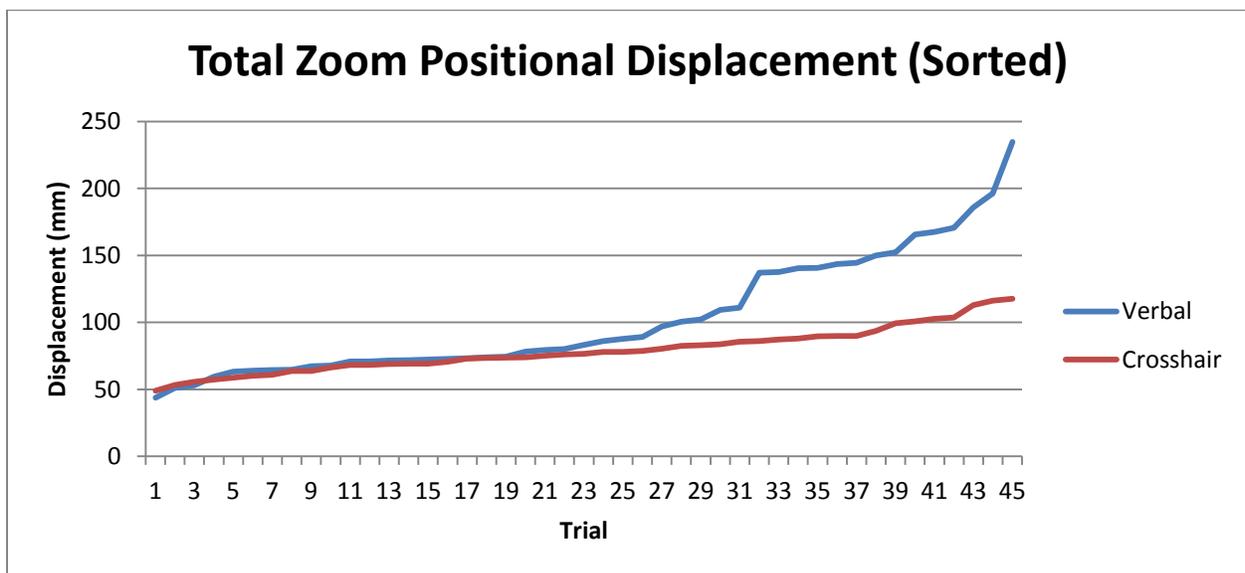
Figure 25: Graph of total displacement during zoom stages, sorted from lowest to highest

Figure 25 shows the total positional displacement that took place during target zooming for each trial. Verbal has a mean of 102.69mm with a standard deviation of 45.04mm. Crosshair has a 23.1% lower mean at 78.93mm with a standard deviation of 16.49mm. Student's t-test returned a p-value of 0.0013.
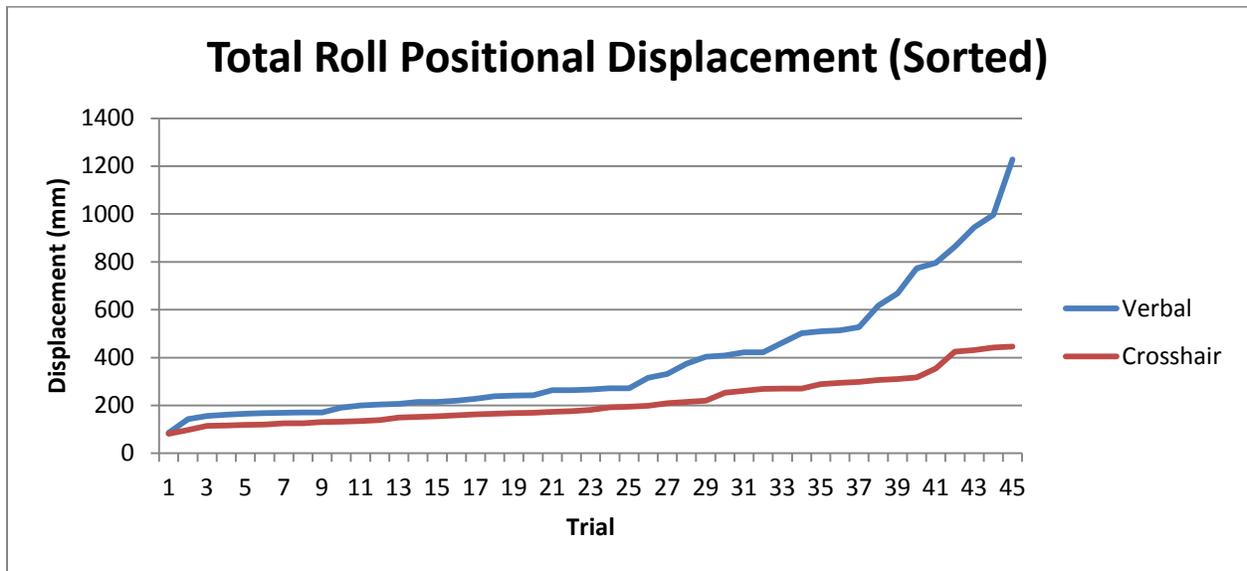
Figure 26: Graph of total displacement during roll stages, sorted from lowest to highest

Figure 26 shows the total positional displacement that took place during target rotation for each trial. Verbal has a mean of 382.24mm with a standard deviation of 262.22mm. Crosshair has a 43.6% lower mean at 215.76mm with a standard deviation of 97.23mm. Student's t-test returned a p-value of 0.00014.

Total Positional Displacement (Sorted)
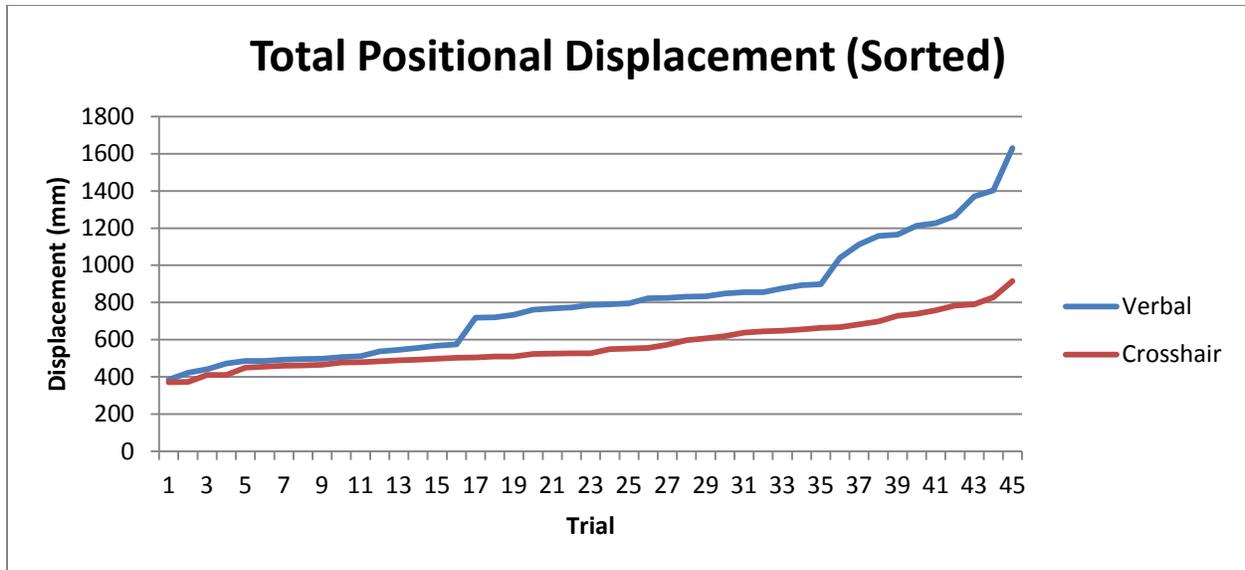
Figure 27: Graph of total displacement for the trials, sorted from lowest to highest

Figure 27 shows the total positional displacement that took place during each trial. Verbal has a mean of 799.04mm with a standard deviation of 298.17mm. Crosshair has a 28.2% lover mean at 573.59mm with a standard deviation of 126.30mm. Student's t-test returned a p-value of $1.07*10^{-5}$.

This reconfirms the data from the time durations. Again, the crosshair system has a lower average value, but it shows some more interesting characteristics. All of the curves show a relatively flat response from the crosshair group. Best to worst, they all have a relatively low movement value. The verbal group shows that the best operators can match the crosshair group, but the lower ranked verbal users drop off heavily and travel significantly father to achieve the same goal.

As before, the target acquisition steps show the largest difference between the two groups. The best users in each group were still 32mm away from each other. That is almost double the distance in the verbal. Some of that difference can probably be accounted to the fact that the augmented reality cues tell the user directly where they need to go. They can then

take a diagonal path to that location. The verbal users are only using their four directional commands. The zoom and roll are very close to each other for the top 50% of the results; the other half leads to a gap between them. The best user results start to approach the optimal distance between targets. It appears that individuals that have a good grasp of the motions approach the optimal results regardless of the system used. For those without a complete grasp of the camera operation, it is interesting how much the gap increases on the top end for the roll and zoom users.

Finally, the standard deviations are still large here. The data sets from the acquisition are the only ones outside of one standard deviation. The others are close to call, exactly as we expected. The box plots show some overlap in the interquartile range. The crosshair results are tightly grouped on the low end of the results while the verbal shows a larger spread. Even with the data showing closer results than with the total time results, the t-test p-values continue to show values smaller than what would be needed for 99% confidence.

Navigational Errors

In this segment, we are defining an error as a point at which the operator moves away from their goal. We analyzed the data and broke each of trials up into their 16 stages. Using the endpoint of each stage, we converted the sensor data from position and orientation to distance from the goal for that specific segment. Every time the distance moved from getting smaller to getting larger, we counted an error for the operator. The results follow.
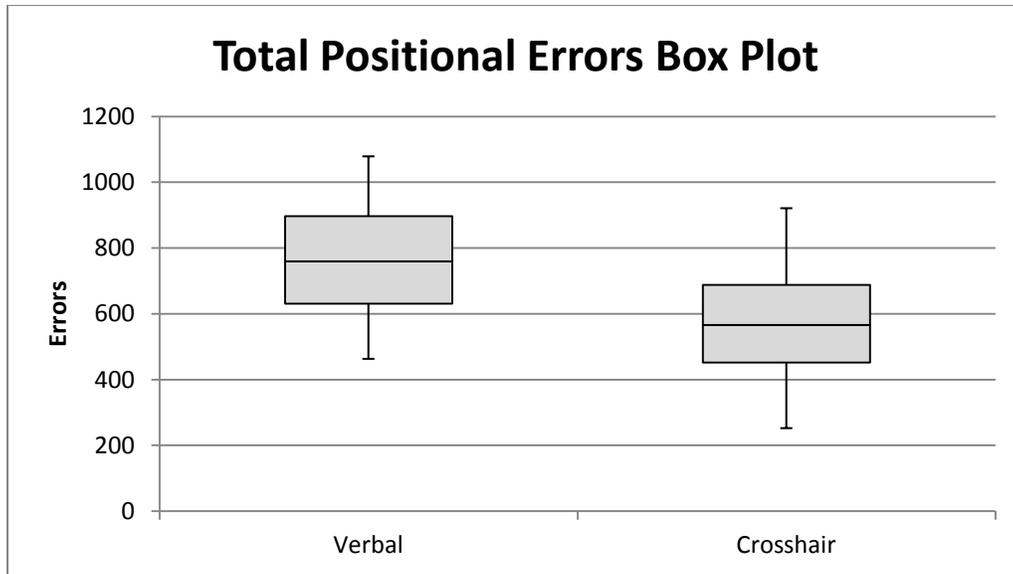
Figure 28: Box plot of total positional errors (Whiskers are 95% confidence interval)

Figure 28 shows a box plot of the total positional error data for verbal and crosshair trials where the whiskers are the 95% confidence interval. Verbal trials had a median of 760, a lower quartile of 631 and an upper quartile of 897. The confidence interval ranges from 463.25 to 1078.84. The crosshair trials had a median of 566, a lower quartile of 452 and an upper quartile of 688. The confidence interval ranges from 252.55 to 921.10.
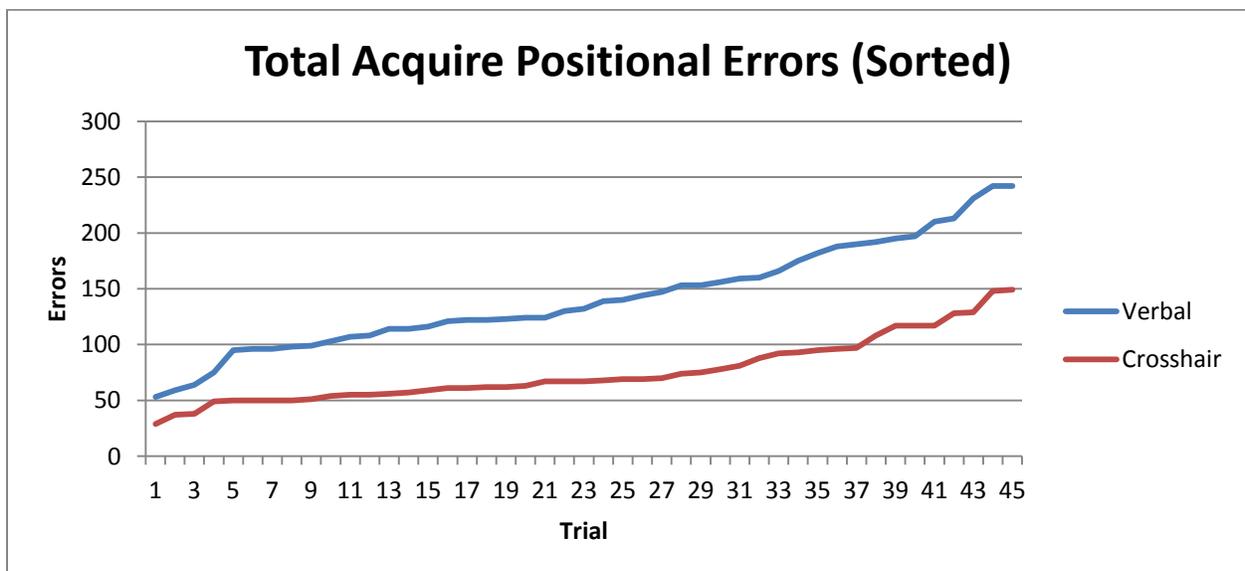
Figure 29: Graph of displacement errors for the acquisition stages for the trials, sorted from lowest to highest

Figure 29 shows the total position errors that occurred during acquisition stages in each trial. Verbal has a mean of 141.53 with a standard deviation of 47.42. Crosshair has a 46.5% lover mean at 75.73 with a standard deviation of 29.04. Student's t-test returned a p-value of $6.2*10^{-12}$.
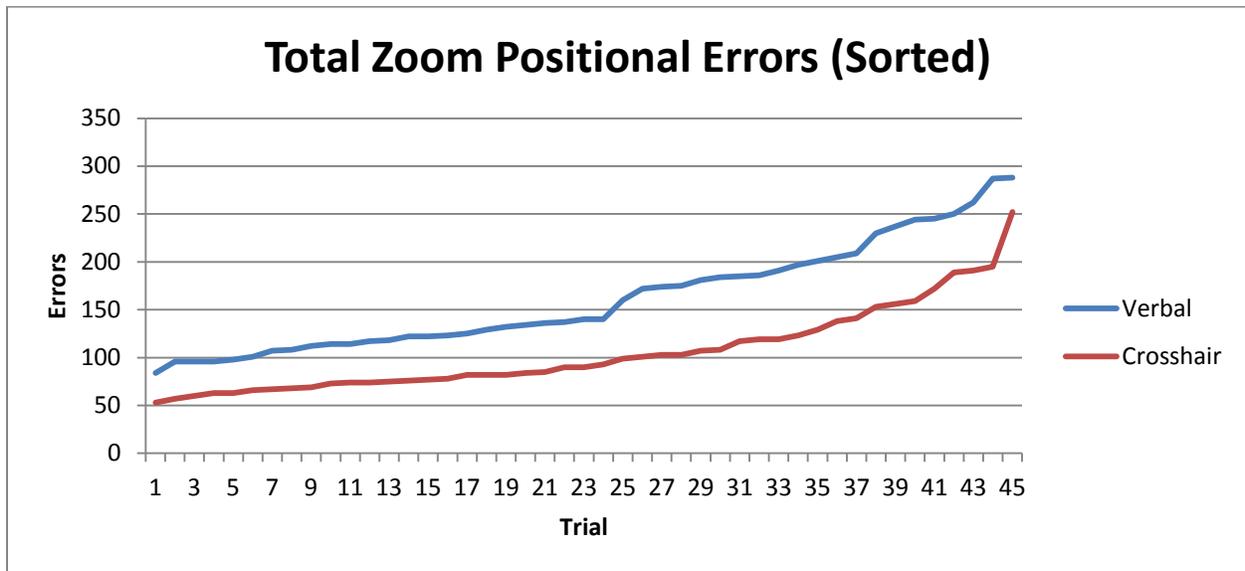

Figure 30: Graph of displacement errors for the zoom stages for the trials, sorted from lowest to highest

Figure 30 shows the total positional errors that occurred during zoom stages in each trial. Verbal has a mean of 161.42 with a standard deviation of 56.05. Crosshair has a 17.4% lower mean at 105.67 with a standard deviation of 44.15. Student's t-test returned a p-value of $1.08*10^{-6}$.
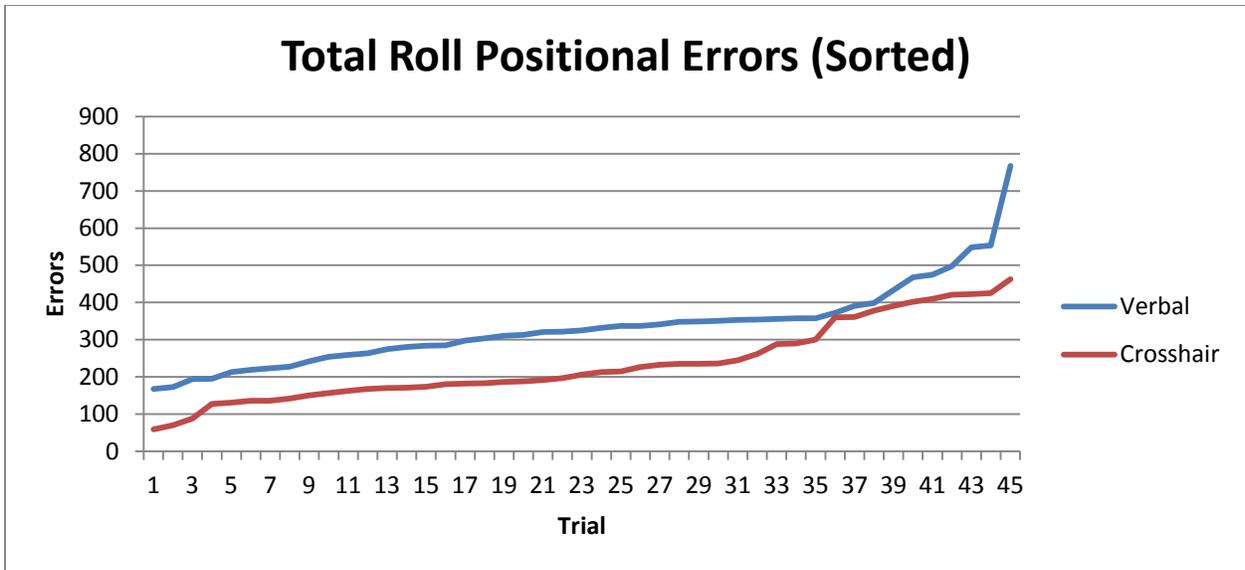
Figure 31: Graph of displacement errors for the roll stages for the trials, sorted from lowest to highest

Figure 31 shows the total positional errors that occurred during roll stage in each trial. Verbal has a mean of 334 with a standard deviation of 112.01. Crosshair has a 29.7% lower mean at 234.96 with a standard deviation of 105.20. Student's t-test returned a p-value of $4.03*10^{-5}$.
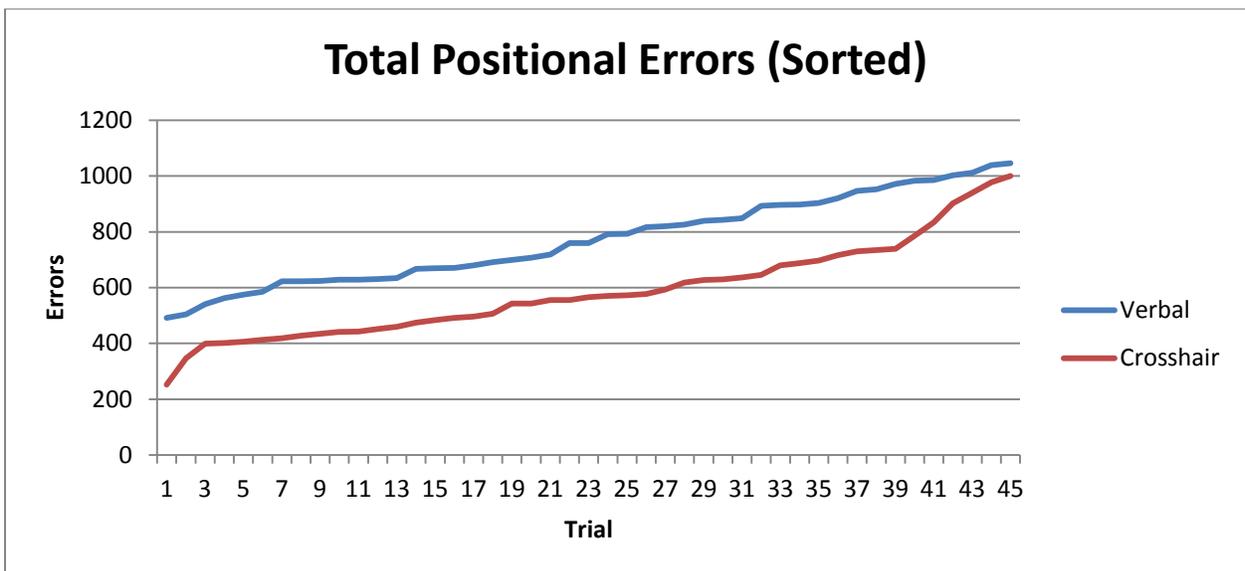
Figure 32: Graph of displacement errors for the trials, sorted from lowest to highest

Figure 32 shows the total positional errors that occurred during each trial. Verbal has a mean of 771.04 with a standard deviation of 157.04. Crosshair has a 23.9% lower mean at 586.82 with a standard deviation of 170.55. Student's t-test returned a p-value of $7.48*10^{-7}$.

The error curves for all of the waveforms are again very close to each other in profile. While the population seems to have a linear ramp in errors, the crosshair curve is still below the verbal curve in all instances. The acquisition stage has the largest difference, as expected. The changes in direction from using a zigzag pattern have to be inducing errors much more than moving directly toward the target. All data sets are outside of one standard deviation of the others except for the roll segment. Regardless, the crosshair continues to outperform the verbal on every segment. The box plot shows a slight overlap on the interquartile range. However, t-test comparisons on each of the data sets show p-values much smaller than necessary for 99% confidence.

Additional Observations

Some interesting results showed up in different areas of the data, but did not have enough samples or enough statistical impact to make any conclusions about them.
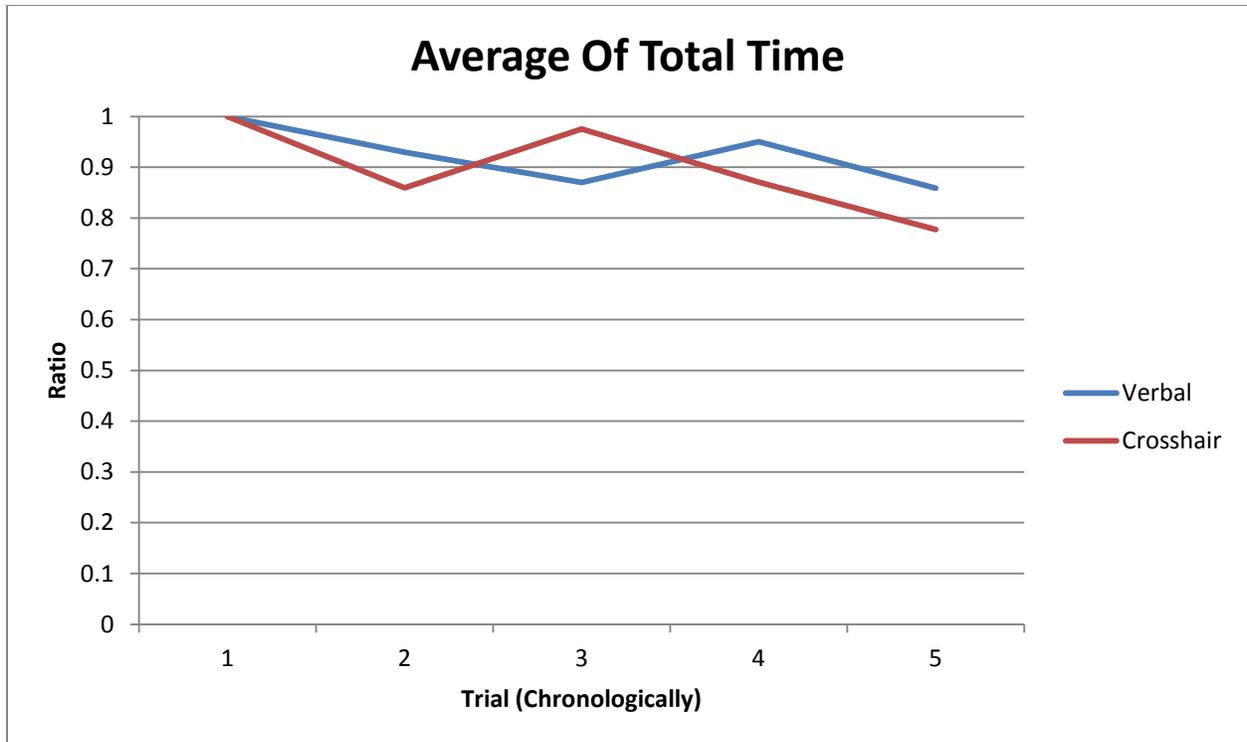
Figure 33: Graph showing time to completion learning curve for average of all user trials

Figure 33 shows the learning curve across the five trials that each user performed. The graph shows the average of all of the users in each group. The lines are the ratio of their completion times in reference to the time set in their first trial. The graph shows a general improvement in overall time between both groups. We expected to see a power law curve with a sudden drop in overall time in the beginning that was then followed by a decay in improvement as the user gained experience with the system. Since we allowed the user to practice with the system until they felt comfortable, we did not know in which part of the power curve our trials would rest. The average of all users resulted in the approximately linear decrease seen above. However, if a few of the users that had a very large variance in their results were removed from the data, the learning curve actually looks like the power law curve dropping from the first and second trials and flattening out from the third onward. Additional experimental data would be needed to make any conclusions on the learning curve.

Expert Time To Completion (Sorted)
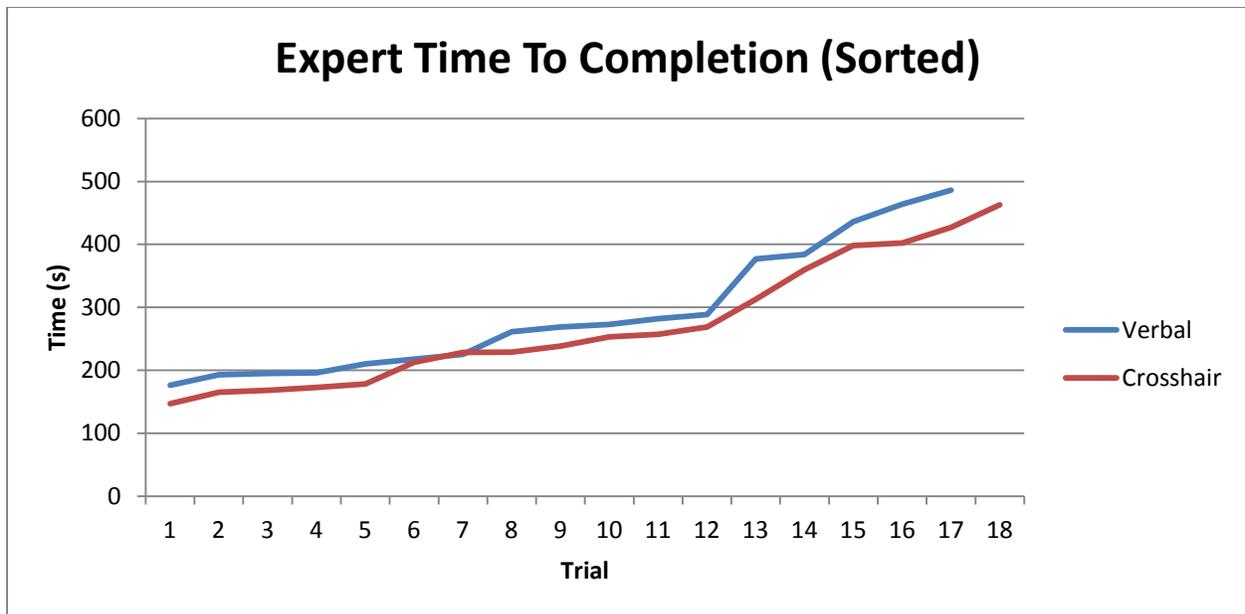
Figure 34: Graph of total time to completion for the medical doctors, sorted from lowest to highest

Figure 34 shows the time to completion for the medical doctors that performed the experiment, sorted from lowest time to highest time. This data set only had a limited number of subjects and trials, but did have some interesting observations. Unlike the student group, this group has significant experience using a laparoscopic camera. They are also used to receiving verbal commands in the operation of the camera. When looking at the time to completion and economy of movement, the medical doctor group was very close in performance using the verbal and the crosshair. Just as in the student group, the crosshair operation does show an advantage in each situation, but the differences are too small to be statistically significant with this small of a data set. One interesting observation was that the medical doctor group always had a slightly lower mean time and displacement using the verbal system than the student group, but it always had a higher mean time and displacement with the crosshair system than the student group. The novice group outperformed the doctors using the crosshair system in all areas. This may be a result of that group having experience using a

laparoscopic camera in a different way. We would need to perform further analysis to make any conclusions.

Conclusions

Every data set that was looked at showed the crosshair system at an advantage. Those advantages ranged from 17.4% to 47.5%. Even with the size of the sample population and the variability in testing results, the p-values returned in each data set comparison were very small. All of the hypotheses were confirmed. The overall time, movement, and errors were all lower with the crosshair system. This was aided by the crosshair system showing a clear advantage with the target acquisition process, and maintaining a slight lead with the zoom and roll portions. The worst trials are progressively worse on the verbal system. Finally, the users that had the hardest trouble with maintaining the position of the scope while rotating the camera had an excessively hard time. As predicted, that weeded out the individuals that could not do the mental transformations.

The preliminary work for this aim was presented at the iSURGITEC Conference 2009 [31]. It was awarded second prize after the proceedings.

CHAPTER 3 – PRE-OPERATIVE IMAGING FOR OPERATING ROOM AUGMENTED REALITY

Background and Significance

The significance of this aim applies to two different areas. It holds significance as a system that could assist in the surgical environment. It could also provide an advancement in telementoring utilizing registered annotations.

Pre-Operative Imaging

Pre-operative imaging, such as CT scans and MRI scans allow the doctors to have a snapshot to see inside of the patient. The information contained within the scans can provide the doctor with a clear picture of the bone and tissue structures inside of the body to diagnose or pinpoint any problem areas. When a patient is in the operating room, the doctor only has the images on the wall without a direct reference to the patient. The doctor is required to estimate, while reading the scans, where inside of the patient the structures in the scans exist. This estimation is complicated when the scans are actually mirror images of the body and need to be flipped. The ability to have pre-operative imaging projected on an image of the patient would immediately solve many of the problems related to reading the scans. When registered to the patient on the operating room table, the scan information is overlaid directly on the portion of the body that it pertains to. No estimation is necessary, and no images need to be flipped in the doctor's mind. The positions of everything are directly where they should be. Important structures can be easily highlighted before the operation and can also be overlaid on the patient to provide helpful information on where the doctor should and should not be going.

The system could reduce the time needed to look at the scans relative to the patient in the operating room, and it could potentially reduce any mistakes involved with reading the scans and mentally applying them to the patient.

Telementoring

Telementoring as a field has existed for multiple decades. An expert on a procedure that could not make it to the specific hospital that is in need of that procedure could provide assistance over the telephone. With video playback devices, this extended into offline teaching. Satellite communication led to possibilities of video conferencing, and computers and the internet have expanded the speed of communication and the methods of communication between people. Telementoring can be applied to any aspect of surgery, but it has some specific ties to laparoscopic surgery because the procedure already uses a camera. Advanced laparoscopic procedures are also specialized enough that there is a shortage of experts and teachers of those techniques. With telementoring, an expert is able to instantly appear anywhere on the planet to teach other doctors how to perform the procedures without having to physically travel to that place. The expert can also provide assistance to a non-expert surgeon to perform a procedure on a patient that is located nowhere near an expert of that procedure. With these advantages, numerous studies have been performed to evaluate the difference between telementoring and mentoring. One of the earliest studies found no difference in skill between learners who watched the procedure in the operating room, and learners who watched a telementored video of the procedure [32]. A more recent study found that while learning on a LapSim system, students that received on-site instruction performed exactly as well as students who received videoconferenced instruction [33]. Another study

verified that both mentoring and telementoring led to retained skills in the mentored subject when they were later working on their own [34].

All of the systems that have been built for telementoring purposes have limitations, however, almost universally, the evaluators of these systems found utility in the systems even with the limitations. One early system utilized an ACECAT II Telestrator, used for football broadcasts, and a VCR for instant replays of what was shown [32]. They transmitted video across the grounds of the hospital for the expert to be able to draw on.

This study was soon followed by a group that performed seven procedures with telementoring over a 3.5 mile distance on a 1.54Mbps network connection using a computer [35]. The processing power and network bandwidth allowed for a 30fps video stream at 176x144 resolution. Audio was sent, along with the ability of the telementor to control the AESOP robot holding the camera. They were happy with the system's performance and had no problems with their procedures.

A very similar system was developed by another research team to do three procedures with the expert in a completely different country, telementoring over the internet [36]. All of the same system capabilities were achieved with a 384Kbps internet connection utilizing higher compression on the signals. With the increased distance between mentor and student, the system performed with a 1sec delay. The study found the 1sec delay workable, as long as everything was done slowly.

Another group took the bandwidth limitations even further with a study using telementoring to assist a novice surgeon in Ecuador over a phone modem [37]. This group used Microsoft Netmeeting to videoconference at 12Kbps. They received video at 5fps at what they

called a terrible resolution. For anything that needed visual detail, still images could be sent uncompressed and edited in Microsoft Paint. With the available bandwidth, it could take 20 minutes to send a picture and receive an edited one back. For their purposes, and considering their conditions, they found the system usable.

The US Navy performed some trials evaluating the feasibility of having an expert surgeon provide assistance through telementoring to help a ship doctor perform an operation on a patient rather than risk sending them to shore [38]. The system they designed used voice over the telephone. Still images could be sent, along with video. Digital images could be sent back and forth, and edited in computer image programs. Chat and e-mail were also present. They were utilizing satellite data transfers at 9600bps to 21600bps, giving them very low-resolution video at 2-4fps, along with a 2-12 second delay depending on conditions. With these limitations, they still found the system useful, though the satellite connection was very unreliable, knocking out the data streams on occasion. That could potentially be dangerous to the patient if the expert could not reconnect quickly.

A group in Japan found a videoconferencing system on a 384Kbps network connection useful for telementoring [39]. Another group working between Italy and the United States was happy with their Chryon Telestrator system working on 832Kbps with less than 1sec delay running video, audio, an AESOP robot, and a PAKY robot [40]. Another group utilized an extremely similar system running on a 512Kbps connection between the United States and Brazil and were happy with the system for teaching surgical procedures [41].

After 2003, with faster computers and faster network connections, the research community finally stopped marveling over the fact that they could send video and audio across

the world, and started to worry about the quality of the experience. One of the first papers to address this was a group out of Canada that performed 19 cases with an audio / video conferencing system [42]. They experimented with network connections between 384Kbps with 300ms lag and 1.2Mbps with 150ms lag. They found the system to work on both ends of the spectrum, but stated that the bandwidth for the video and audio was critical to the quality of the telementoring. They found that they preferred if the bandwidth was never below 512Kbps. They also found both 300ms and 150ms of lag workable for the surgeon, but they both required an adjustment to your working speed and the 150ms delay was noticeably better than the 300ms delay. Low bandwidth and high latency negatively affected the ability to teach.

The same group also performed a study at the same time involving Zeus robot control with the system [43]. Telementoring audio and video transferred at 384Kbps to 1.2Mbps at 300ms latency and the robot ran on a local network with up to 15Mbps bandwidth. Not all of the 18 procedures were completed without problems, but the group was able to conclude that even though they could adjust to the 300ms of latency on the videoconference, the robot control absolutely needs a high-bandwidth, low latency connection to be usable. One member of the group did another study pertaining to telementoring and robotic control across 400km in Canada [44]. He looked at 22 cases with telementoring and Zeus robotic control. The system worked well for them, but he concluded that 175-200ms of lag was detrimental to the performance of the surgeon. With latency at that level or larger, the surgeon had to slow down everything he was doing to compensate.

Finally, multiple papers have come out concerning the research that NASA is still working on involving the challenges related to medical procedures on long space flights.

NASA's NEEMO project placed people in an undersea environment examined the difficulties involved with a selection of medical procedures and communication latencies [45]. They determined that astronauts, or people of sufficient intelligence, could perform interventional procedures with the help of a telementor. They found that high-bandwidth and low-latency connections were vital to the quality of the teaching experience, and they also found that they needed training for the astronauts to be able to keep the patient alive and safe in case the communication link fails and needs to be reestablished. Another paper extended and elaborated on this work [46]. They looked at what could and could not be done depending on how far away from Earth the astronauts were. They felt that robot control from Earth would stop being feasible with a 2s delay, telementoring would lose its usefulness at 50-70s, and after that, expertise would be needed on board with Earth functioning as a consultant.

All of this previous work comes to a few conclusions. Telementoring is useful, but bandwidth and latency are crucial to its usefulness. Basic telementoring systems have begun to appear in commercial products, such as the da Vinci robot, but most of the research has shifted over to teleoperation instead of telementoring. This research aim intends to utilize some new technology and some original concepts to increase the usefulness of telementoring further. In doing so, some problems in all of the current systems will be eliminated.

Universally in the above mentioned research, the resolution and frame rate of the video are severely limited. In our own experimentation using heads-up-displays in the operating room, we have found that if the video feed drops below NTSC level video resolution, too much detail is being lost in the laparoscopic camera feed to have an accurate picture of what the surgeon is working on. High-resolution picture quality must be maintained.

Most of the systems mentioned are also nothing more than a simple videoconferencing client. The ability to point out structures on the video feed and draw on it are also important to the usefulness of telementoring. The systems that do allow the user to draw on the scene all have a few flaws, one of which is displayed in Figure 35.

Figure 35: Moving Camera Problem – Camera moves yet drawing stays in the same place

In all of the current systems, if the mentor circles a point of interest, such as the eraser above, the drawing is not made in reference to the eraser; it is only made in reference to the video frame. When the camera moves, the drawing stays in the same position in the video frame, but the point of interest is no longer under the drawing. This causes major problems in current systems if there is high latency, because by the time the mentor draws on the video and it gets back to the student, the camera is probably not in the same position anymore and the mentor's drawing was never where they intended it to be.

All of these problems would be conveniently alleviated by registering the drawing to the environment instead of the video frame. One objective of the following system is to allow the drawings and annotations of the expert to be registered to the pre-operative imaging and be registered to the patient that exists in the environment. When the camera moves, the information that the expert was trying to convey should still be useful to the user.

Preliminary Foundational Work

The preliminary implementation of Task 1 has been assisted by the work that Dr. Pandya completed toward his Ph.D. [47]. In the course of his work, a system was set up to track the position and orientation of an object. A composite skull was covered with fiducial markers and their positions from each other were calculated. CT scans were taken of the skull with markers and the data was imported into Brigham and Women's Hospital and MIT's 3D Slicer software [48]. This allowed for the creation of 3D models based on various internal structures within the skull. A camera was also set up with fiducial markers on them and a Polaris infrared tracking device was used to report the locations of all of the markers in the scene. Additional software was written to determine the position of the camera in relation to the skull so that the 3D models of the internal structures could be displayed on top of the video feed as though the viewer were looking through the skull with x-ray vision [49].

Hardware

The base hardware setup is shown below in Figure 36. Everything is attached to one platform for the ease of demoing the system. A Microscribe G2X passive robotic arm holds a custom mount for a cylindrical NTSC camera. This arm, with a stated accuracy of 0.23mm, calculates the position and orientation of its end effector and returns it over a serial connection.   The system contains a checkerboard pattern for camera transformation calculation, and a skull phantom covered with fiducial markers for the object transformation calculation. The open skull is shown in Figure 37.
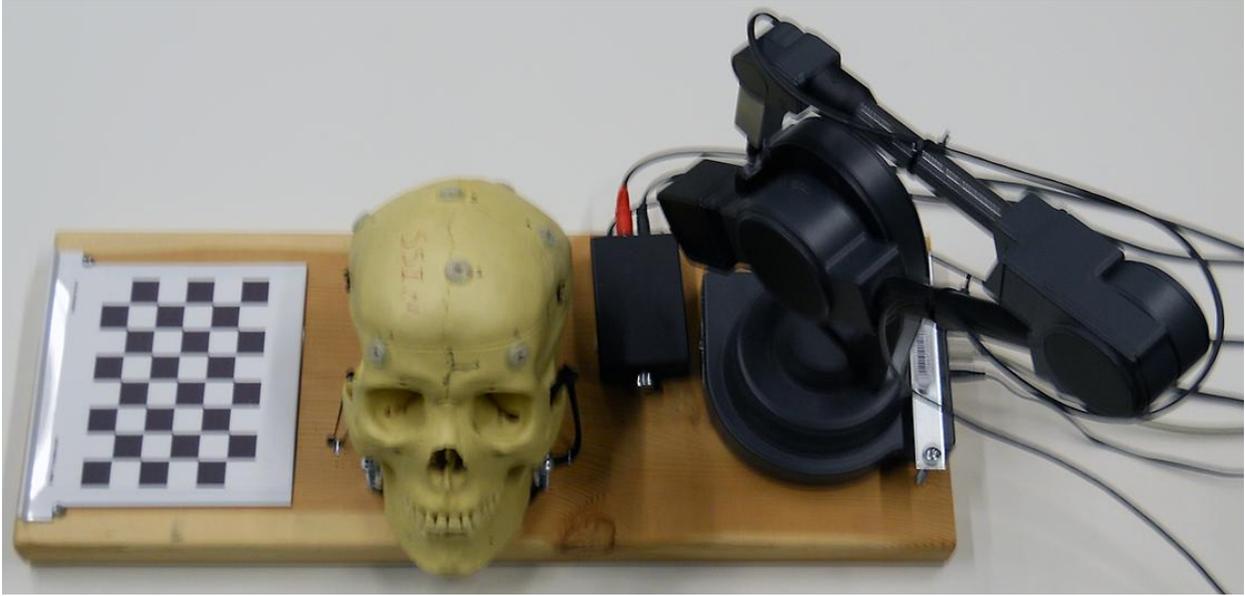
Figure 36: Hardware testbed for AR system
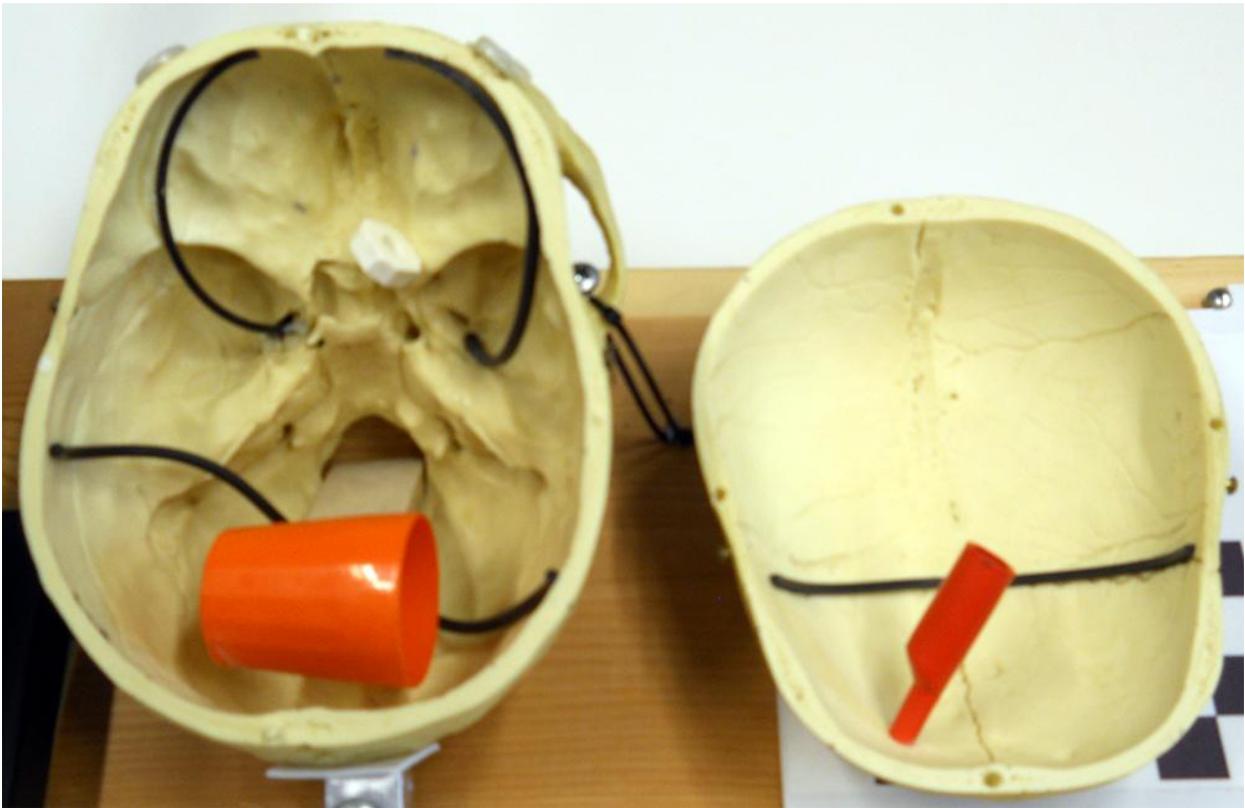
Figure 37: Inside skull phantom

Objects were affixed to the inside of the phantom to provide obvious features to identify and segment out of the CT scan data for the skull. This data could then be overlaid on the video feed as augmented reality.

System Architecture

The basic design for this system took into account the necessary requirements to implement augmented reality while addressing a couple factors specific to our needs. The architecture is shown below in Figure 38.
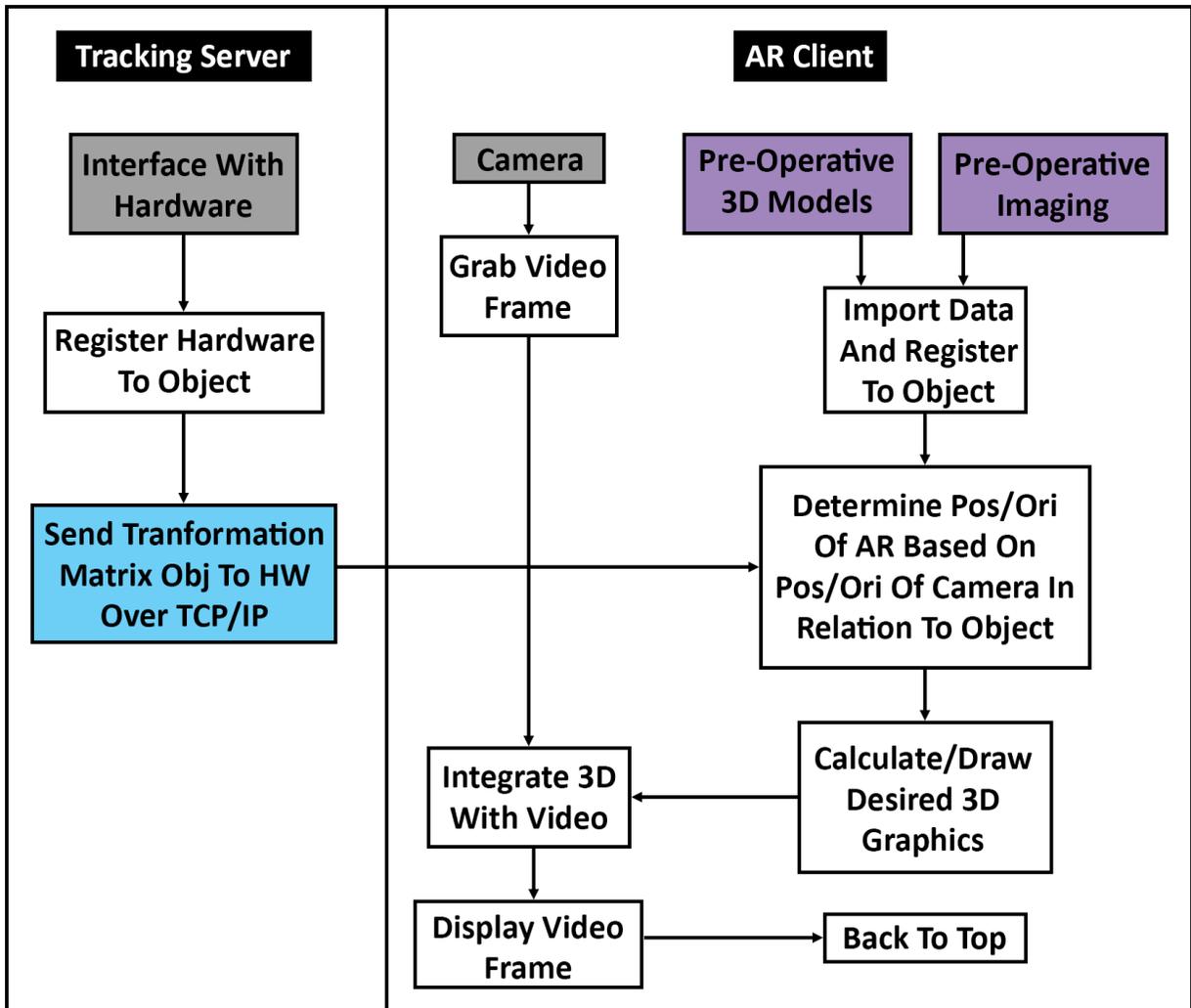
Figure 38: AR system architecture

The high-level design intention with the software was to separate the object and hardware tracking from the augmented video playback. This would allow hardware connected to a separate computer to be able to influence the scene. It would also allow multiple pieces of

hardware to be connected. Offloading the tracking of hardware to another program can provide a simple advantage of lowering the processing requirements on the AR displaying computer, but it can also allow an expert surgeon on the other side of the planet to connect to the system through the internet and augmented the scene themselves for instructional purposes.

Tracking Server

One basic requirement of augmented reality is that the movement of the camera needs to be tracked in order to know where to draw on the video feed. Our system, using pre-operative imaging, also needs to know where the object that was imaged is located so that it can track where the camera is in relation to the object.

To register the object to some known point in the system's world, we needed to use the fiducial markers on the object. These markers, located at various points on the surface of the skull show up clearly in the CT scans. After importing the scan data into 3D Slicer, the software could be used to determine the positions of each of those points in 3D space. A system of pair point matching was then used to come up with a transformation between the object and the robotic arm, as illustrated in Figure 39.
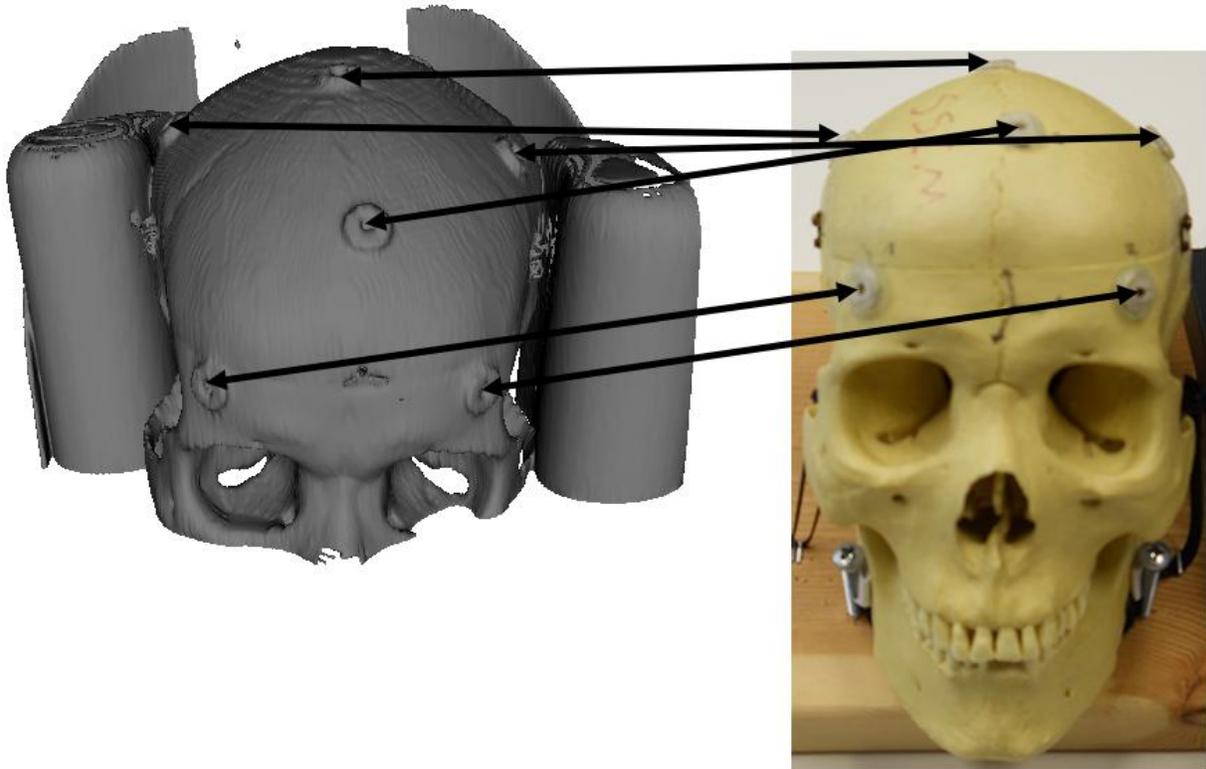
Figure 39: Matching fiducial markers between CT data and skull phantom

Since we know the locations of the fiducial markers on skull from the CT scans, we can use the robotic arm end effector to touch each of the fiducial markers on the actual skull in the environment in a predetermined order to come up with the current world location of those same markers. Each of the scan points matches with a current point when multiplied by the standard 4x4 homogeneous transformation matrix. A Levenberg-Marquardt algorithm based optimization routine was used to determine the transformation between the real world space and the CT scan space. A set of 3D translation values and Euler rotation angles were optimized by applying them to the pairs of points and iterating through the values to minimize the distance between each pair of points.

With that transformation set for the current skull position, the tracking server maintains a network connection with the AR client. It continuously polls the robot arm to determine its

end effector position and orientation and sends a transformation matrix for the object to the end effector over to the client when the client requests it. The bandwidth usage for the operation is only a 4x3 tranformation matrix of 32-bit floats, since we know the final row of the matrix is [0 0 0 1]. That totals to 48bytes plus overhead for each transaction, synchronized to the AR client camera. At 30 frames per second, that only adds to approximately 1.4KBps.

AR Client

The client software in the system is responsible for generating and displaying the final augmented video output for the user. It interfaces with the NTSC camera we used and displays that video data on the monitor. It connects with the tracking server and asks for continuous updates on the position of the robot end effector. However, the transformation between the object and the end effector is not enough to be able to draw virtual object in the scene that are registered to the real objects. The working parts in the scene and the transformations between them are shown in Figure 40.
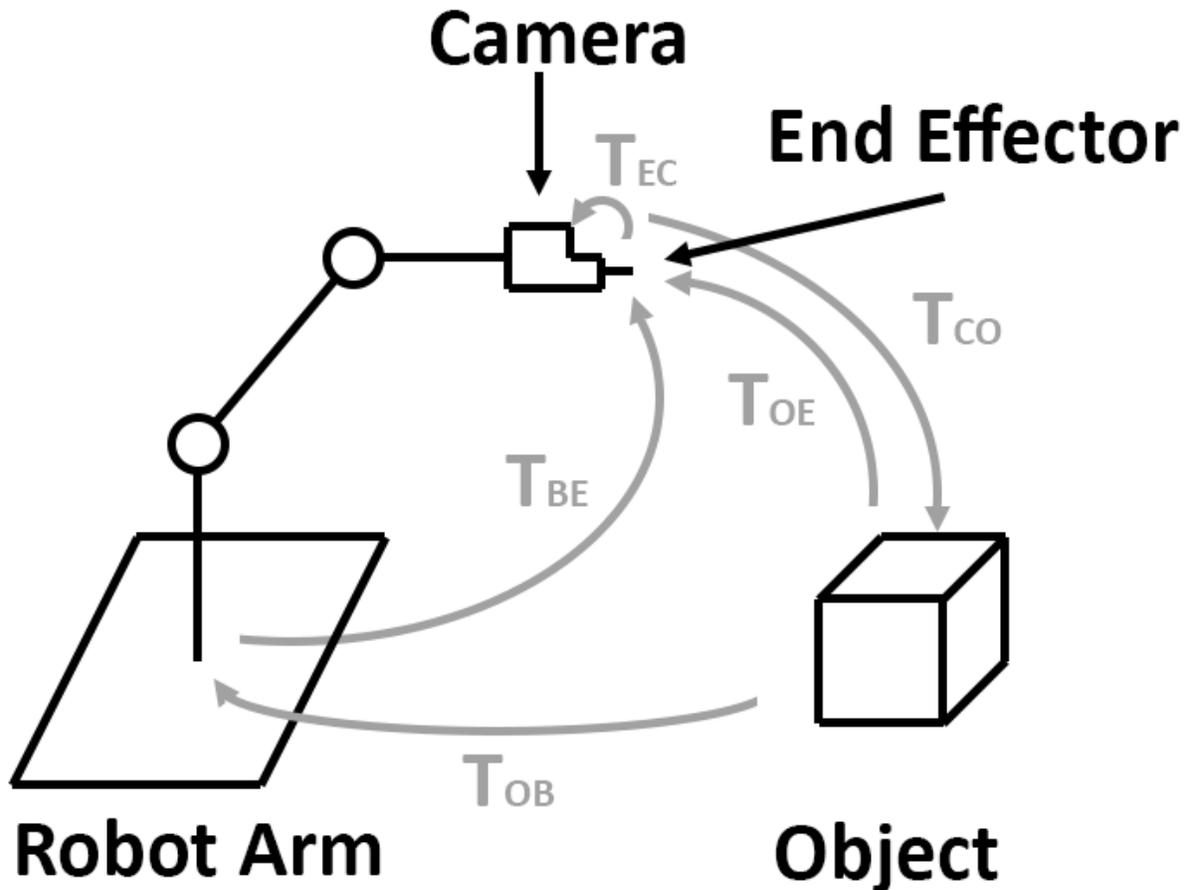
Figure 40: Transformations in AR system

The tracking server returns T_{OE}, the transform from the object to the end effector. To draw the augmentations registered to the object, we need to know the transform T_{CO} from the camera to the object. We can find T_{CO} by running the transformations in reverse to get $T_{CO} = T_{EC}^{-1} * T_{OE}^{-1}$, the inverse of the transform from end effector to camera times the inverse of the transform from object to end effector. We don't know T_{EC} though. T_{EC} would be the rotation and translation from the end effector to the CMOS sensor in the cylindrical camera. We can estimate the position and rotation of the camera in its mount, but cannot measure it exactly.

Instead of measuring T_{EC}, we can calculate it by running the inverse transform path where $T_{EC} = T_{BE}^{-1} * T_{OB}^{-1} * T_{CO}^{-1}$. The software from the robot arm returns T_{BE}, and if the end

effector is touching the object, we can find T_{OB}. To find T_{CO}, we need the camera calibration grid from Figure 36 and the Camera Calibration Toolbox for MATLAB [50]. Using the toolbox, we took 16 pictures of the grid and then pointed out the corners of the grid in addition to declaring the square size to the 18.5mm the pattern uses. The toolbox then calculated the intrinsic camera parameters governing the internal properties of the camera, including focal length and lens distortion. Using one additional picture, the toolbox could return the extrinsic camera properties for that picture, which included T_{CO}, the transformation matrix between the camera and the grid. The pixel error on that calculation was less than 0.3 in both dimensions, quite small for a 720x480 image. When that picture was taken, we recorded the value of T_{BE} from the Microscribe software. We used the grid one last time to calculate T_{BO} as shown in Figure 41.

Figure 41: Calculated axis to find robot base to object transformation
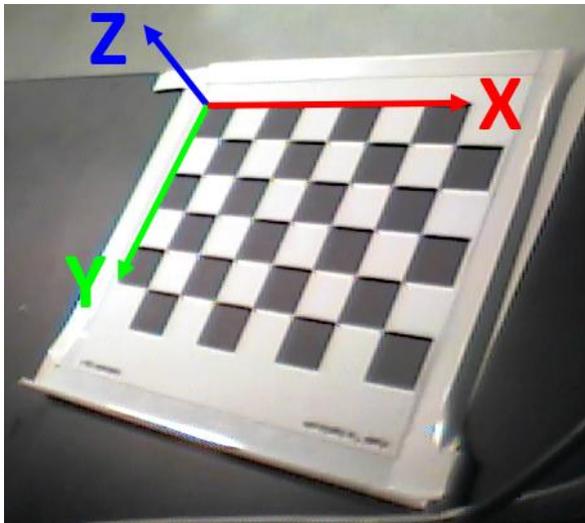
$$Z = X \times Y$$

$$T_{BO} = \begin{bmatrix} X_0 & Y_0 & Z_0 & O_X \\ X_1 & Y_1 & Z_1 & O_Y \\ X_2 & Y_2 & Z_2 & O_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

To find T_{BO}, we needed to generate our own reference axis on the object. Using the end effector on the robot arm, we recorded an origin point in the upper left corner of the grid and recorded position values for an X point down one axis of the grid and a Y point down the other axis of the grid. Vectors were generated by subtracting the origin from each point and then the

cross product of the X and Y vectors were taken to generate the Z vector. The transformation matrix was then formed by using the principle axes as the rotation matrix and using the origin point as the translation reference.

T_{EC} could then be calculated as $T_{EC} = T_{BE}^{-1} * T_{BO} * T_{CO}^{-1}$. Our results showed a very tiny rotation of the camera and a translation of (-2.43mm, -21.85mm, 55.27mm), very close to the values we measured by hand.

With the transform T_{EC}, we could now integrate the equation $T_{CO} = T_{EC}^{-1} * T_{OE}^{-1}$ into the software to perform the transformation between the camera and the object so that we could transform all of our OpenGL 3D drawings by the same matrix to register our augmentations with the environment.

The remainder of the base software package was an implementation of OpenGL 1.0 that could display the 3D Slicer segmented objects inside of the skull registered to the environment. It would render the 3D graphics and overlay them on the video feed before displaying each frame.

Additional Features

With that baseline programs, I spent some time modernizing the code to run more efficiently on newer graphics processors and added OpenGL extensions to allow translucent rendering of polygons, texture mapping, and a few other systems for non-power-of-two pixel dimensions and drawing-order-independent rendering of alpha channels. Alternate codepaths were added for some features depending on the detected capability of the graphics card. This included pixel shader support. I restructured the codebase to allow my features to be

enveloped in a class that could easily be inserted into the main program and run. This also allowed me to separate my code into its own testbed for development and testing.

All of the following features have been added with an eye on computational efficiency. After upgrading the codebase to utilize a few features in modern graphics processors, there is still overhead for real-time rendering with additional features. The major computation will involve angle independent slicing of the image volume. Generating texture maps for axial, coronal, and sagittal views require the same amount of processing time once the data is structured correctly in memory. Other features have more open-ended performance requirements. Throughout the process, my code has been run heavily within graphics processing unit profilers with an eye on maintaining real-time (30 frames per second) rendering of everything.

Pre-Operative Scan Viewing

The objective with this feature was to allow the user to have the pre-operative imaging data directly overlaid on the object of interest. Instead of having to look elsewhere at imaging and mentally translate the images to the patient, the images could be registered and placed directly on top of the patient. Easily translating through scan slices and changing the viewing orientation gives many more options than looking at the specific films that have been printed up.

As a first step to incorporate pre-operative scans into the AR environment, the DICOM images that the CT scanner returned of the skull phantom needed to be converted into something OpenGL could render. There were 76 DICOM images at 512x512 resolution. The headers of the files indicated they were 12-bit grayscale in a 16-bit storage format. For

maximum graphics card compatibility, the 12-bit values were converted to 16-bit values and stored as such. At this resolution and color depth, each axial texture consumes 512KB of video memory, with coronal and sagittal using 76-128KB depending on whether the graphics card supports non-power-of-two texture dimensions. The header also showed a pixel size of 0.44921875mm with a slice thickness of 2mm. With 76 slices at 512x512 resolution, that results in a scan volume of 152mm x 230mm x 230mm. After reading in the slices, the OpenGL world origin would be set at the skull phantom with that scan volume around it; the appropriate 0.44921875mm spacing would be used in the x and y, and 2mm spacing in the z.

A testbed software program was created to render the CT scan volume and all of the non-hardware dependent features. As the CT scans are only a single set of stacked images in axial orientation, the program also needed to generate a volume based on those images and be able to allow the user to pick the viewing orientation and generate the correct slice of the image through the volume of image information. The base program can generate axial, coronal, and sagittal views shifted through the volume as shown in Figure 42. It also allows for the rotation of the image volume to view from any angle.
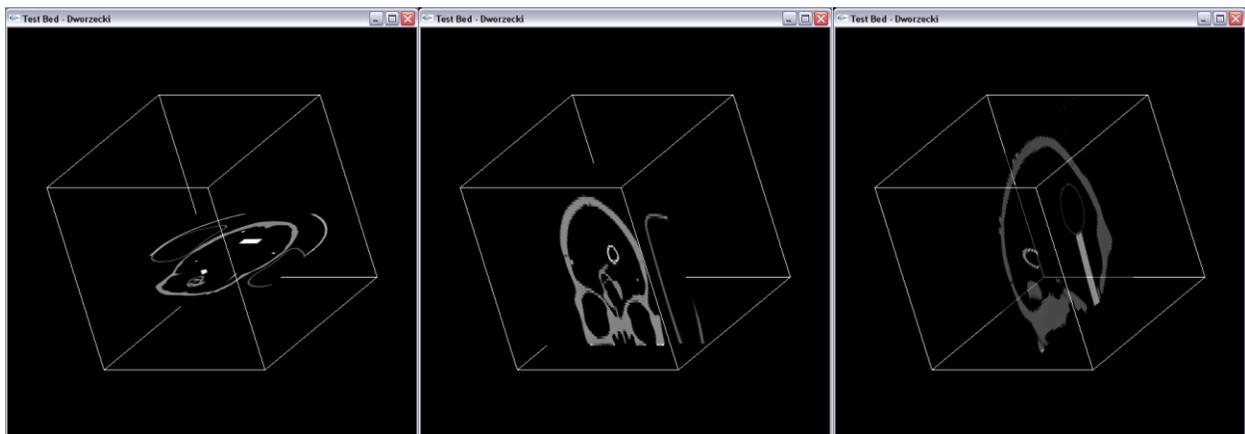
Figure 42: Software testbed rendering axial, coronal, and sagittal views of CT data.

With the axial data in a 3D data structure in memory, the desired slice is chosen and the (x, y) data is read from that desired z level before a texture is generated and bound to a polygon for display. Coronal viewing only requires the data structure to be read as (x, z) data from the desired y level before being generated. The sagittal view is (y, z) data at a desired x level.

The programming classes developed with the testbed were then integrated with the upgraded version of our augmented reality software. An interior coronal slice can be seen in Figure 43. The CT data needed to be registered to the 3D model data and then the positioning of the camera and skull in the scene. The user can then scroll through the volume of image data from axial, coronal, and sagittal orientations and choose the translucency of the slices as they look around the object with the camera.
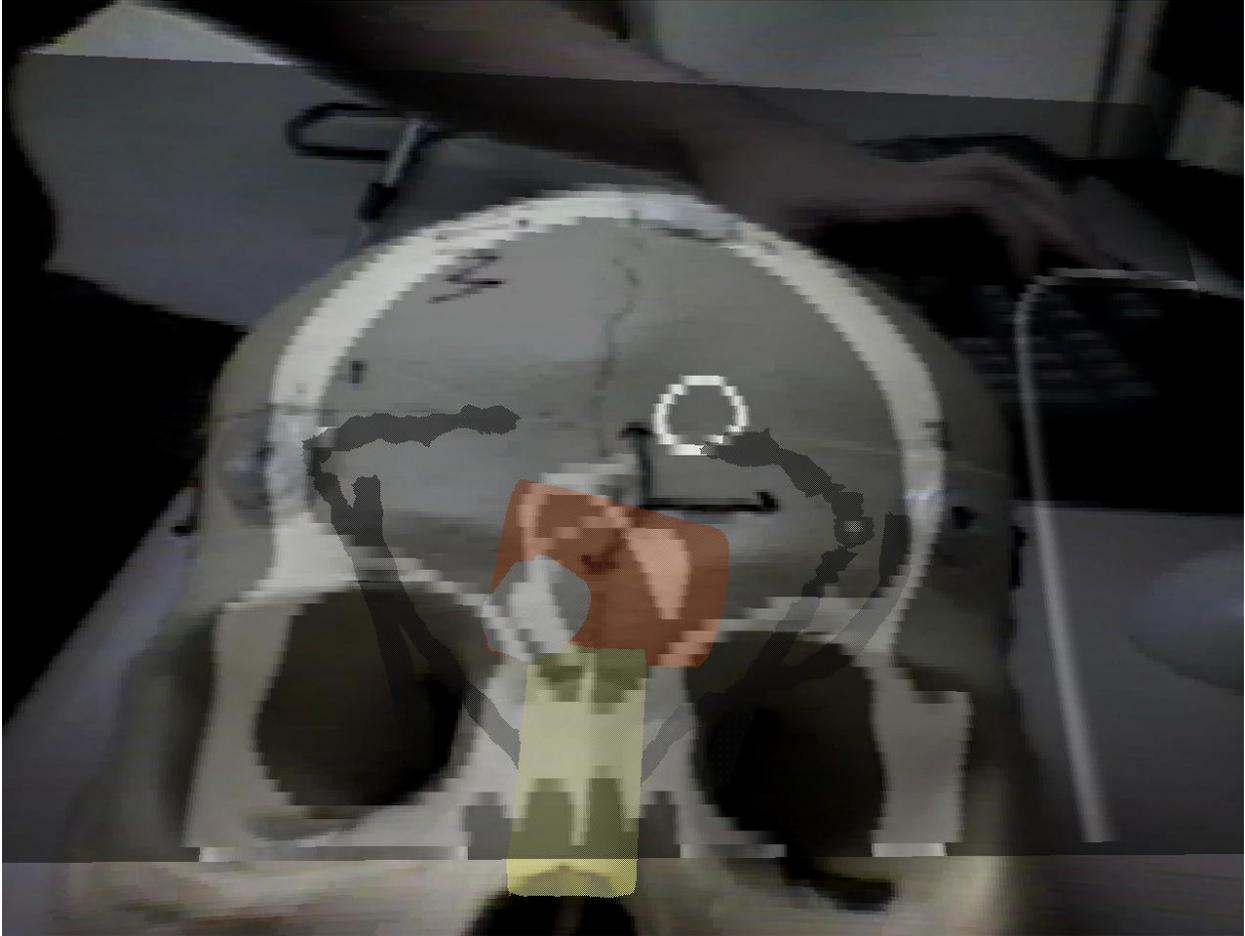
Figure 43: Camera view of skull front with translucent models on internal structures and coronal CT data overlaid

After initial calibration of the system, the model and imaging data maintain their correct

position and orientation with the skull independent of the viewing angle.

Slice Drawing

The next feature was the ability to provide annotation on the CT scan data. The

objective was to provide a 2.5D method to allow the user to pull up the currently viewed slide

and draw on it in 2D. When finished, the drawing would appear correctly on the CT data

registered in the 3D environment. Drawings could also be made permanent, no matter what

slice the user was looking at. This gives the option for an expert to be annotating directly in the

scene to provide assistance to the person performing the procedure. The process appears

below in Figure 44.
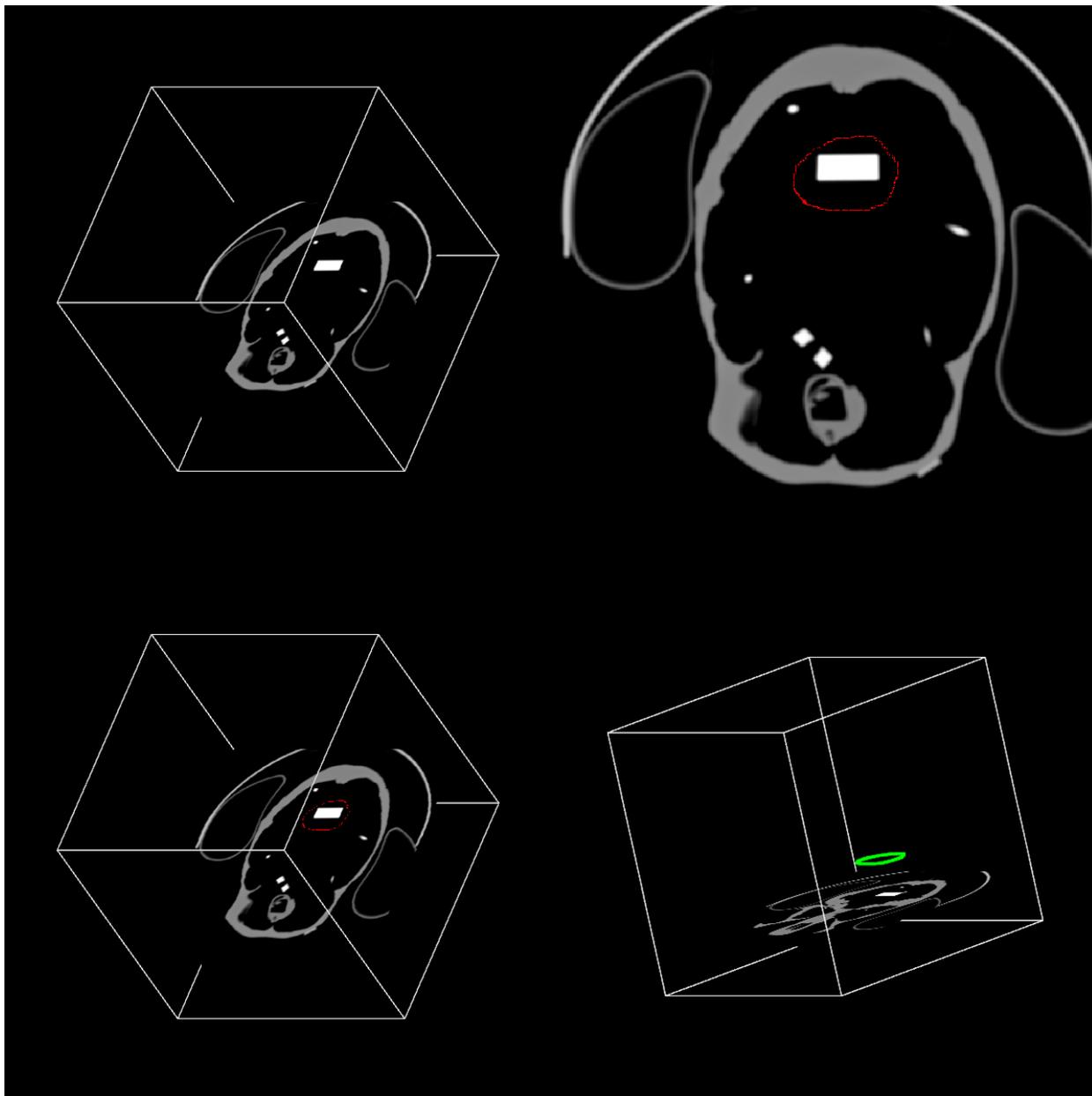
Figure 44: Process of viewing CT slice, drawing on it in 2D, viewing it in 3D, and making it permanent in the environment

In the upper left, an axial slice is being viewed. At the user's request, the slice is pulled

to the front and is expanded to full screen. The input device becomes active and the user can

draw directly on the slice. This is accomplished by watching where the input device is pointed

and converting the point at the screen resolution into an equivalent point at the resolution of the CT slice on the screen. Another data structure the same dimensions as the volume of CT slice data is set up to store where the user has drawn. Now, instead of the CT slice data being grabbed by itself, turned into a texture and bound to a polygon, multitexturing is being used. A texture is being generated from the CT slice and from the drawing, and they are blended together on the graphics card before displaying. The bottom left picture shows the 2D slice drawing appearing in 3D within the volume. Drawing textures are the same format as the CT data to ease texture blending. They have the same additional memory requirements of 512KB for axial textures, and 76-128KB for coronal and sagittal depending on non-power-of-two texture dimension support.

The final feature for this drawing routine is in the bottom right corner. The user can make the drawing permanent in the environment rather than just appearing on the slice while it is viewed. For permanence, the program builds a displaylist of a point cloud for all of the pixels that have been drawn by the user. When viewed like this, you can change your viewed slice or look at something else and still have the drawing on screen around the object in the environment you were annotating.

The 2D drawing on each slice can be done in axial, coronal, and sagittal orientations. They can be combined, with permanence, to create something 3-dimensional such as the drawing in Figure 45.
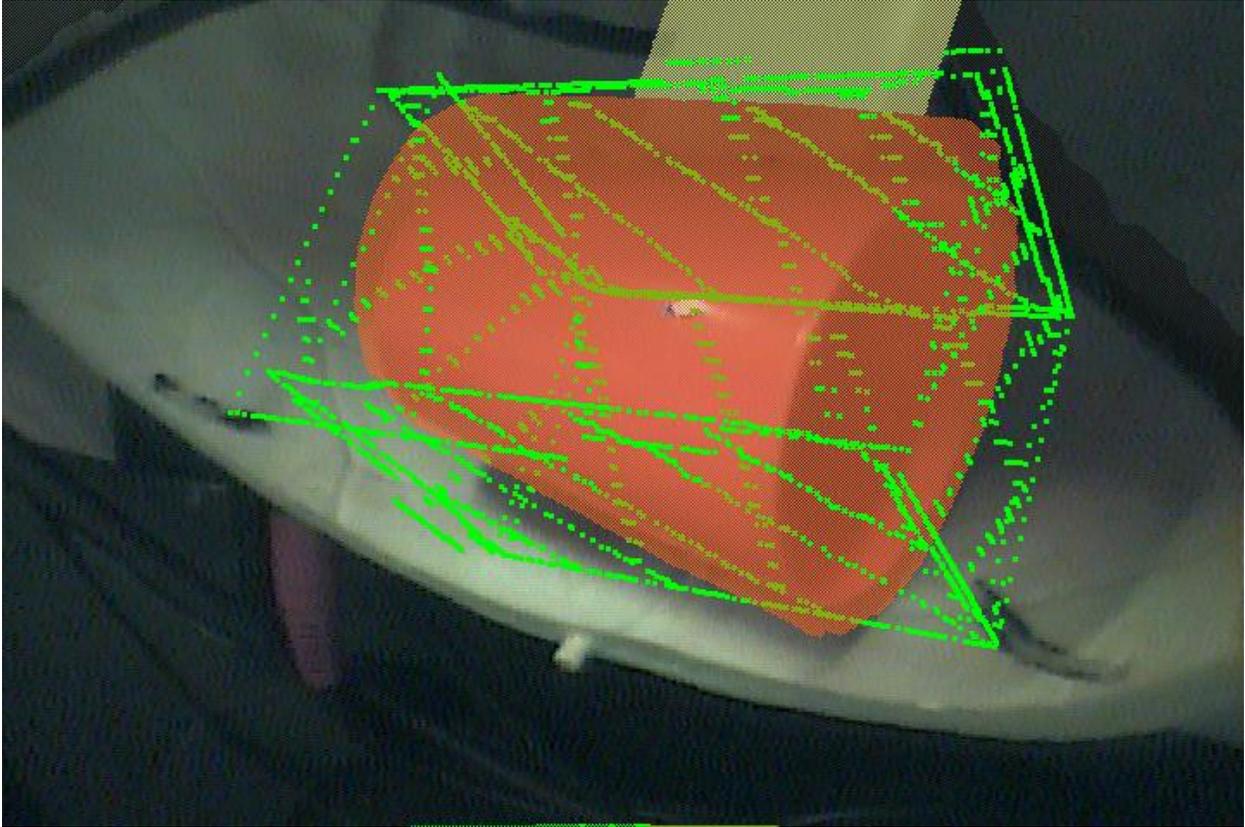
Figure 45: Hand-drawn striped box over cup

Viewpoint Perpendicular 3D Slice

The objective with this feature was to allow a real-time view into the CT volume. Instead of having to manually adjust the orthogonal slice options from before, the slice that is viewed could be continuously generated as a slice parallel to the video feed. This would allow the user to move the camera to be able to see the pre-operative imaging from a different perspective.

Of all of the features, this is the most computationally intensive. Two codepaths were written to accommodate the different computers this was running on. One codepath defaults to OpenGL 1.4 3D texturing with most of the calculations done on the CPU. The other uses basic pixel shading to accomplish the same results with most of the processing on the graphics card.

The initial steps to generate the 3D slice are shown in Figure 46.
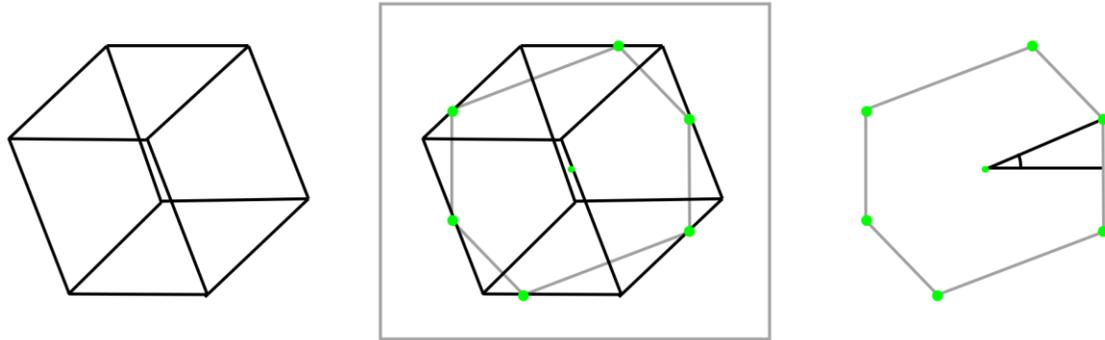
Figure 46: 3D slice generation – CT scan volume to convex polygon slice

In the figure above, we start with the 3D CT volume at an arbitrary angle. A plane segment, shown in grey, is calculated. To reduce the complexities of the calculation, the skull data and world origin have always stayed at (0, 0, 0) in the OpenGL environment. With the z-axis coming straight out of the camera through the viewport, and the plane being parallel to the viewport, the plane segment has a Z value of zero, only requiring intersection calculations to be done at a Z = 0 crossover.

To find the intersections with the extremes of the volume, the intersections with the line segments composing the outside cube need to be calculated. All of the corner points of the cube are generated and they are rotated by the same transformation matrix that rotates the entire scene for display. Point pairs are generated to represent the line segments that make up the cube. Once a line segment has been verified as not being parallel to the plane segment, the program just checks to see whether both end points are on different sides of the plane or one or both are on the plane. If they are, it builds the equation of the line and solves for the (x, y, z) value of that intersection. Duplicate entries then need to be deleted, because if a corner was on the plane, then the three segments would have been calculated. With that list of

intersection points, that can only range from 4-6, a polygon to be texture mapped has to be generated.

That is complicated by OpenGL's requirement that the vertex call order for an n-sided polygon needs to generate a convex, simple polygon. A line from any point to any other must be in the interior of the polygon, and no perimeter line segments can overlap. In order to meet this requirement, the program needed to sort the vertices. The chosen method was to project a line on a positive x-axis emerging from the calculated centroid of the point values, and calculate an angle from that axis to each of the points in the list. The points would then be sorted by how far they are from the x-axis in degrees. Issuing a draw command with the vertices in that order would satisfy OpenGL requirements.

The intersection points now need to be transformed back into their original coordinate system for intersection calculations with the volume. The methods are different for the CPU and graphics card, but they both follow Figure 47.
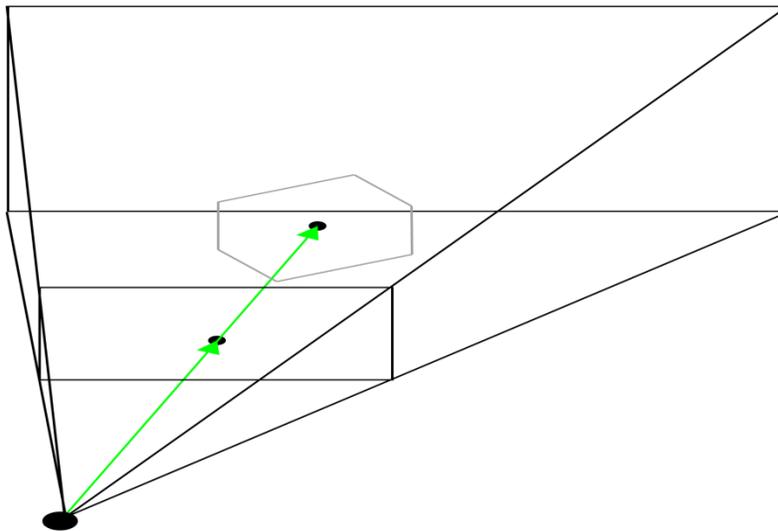
Figure 47: OpenGL world view of viewport and 3D slice

The figure shows the zoomed out world view of the OpenGL environment. The camera is at the bottom left looking through the viewport to see the partial pyramid that is the viewable rendering area. With the pixel shader code, there are effectively rays coming out from the camera that intersect with every pixel on the screen. The rays that end up intersecting the 3D slice polygon that is being rendered go through a calculation to find the exact point inside of the slice volume that would be at the intersection of the ray and the polygon. The color value at that point in space would be grabbed to display on the screen at that point. Doing these calculations without shaders is not extremely different. Once the texture coordinates have been altered to match the plane's slice through the volume, you are again calculating what voxel the displayable point on the polygon is occupying the same space as.

Any modern graphics card or CPU can handle the rendering of this feature smoothly. It is only on very old products with low memory that frame rate becomes an issue. Memory usage of the 3D texture volume takes 38-64MB of video memory. 76 slices of CT data takes up 38MB, but on most older graphics cards, all volume dimensions must be powers of two, so 128 slices, even though 52 are blank, take up 64MB of video memory. If the card has less than 128MB of memory, texture swapping across the system bus will be taking place every frame.

3D Drawing

The objective with this feature was to allow actual 3D drawing within the scene. If a motion needs to be described, or a path needs to be navigated in 3-dimensions, this would allow the expert to inject 3D information into the scene. An example of point drawing from the cup to the nut inside of the skull is shown in Figure 48.
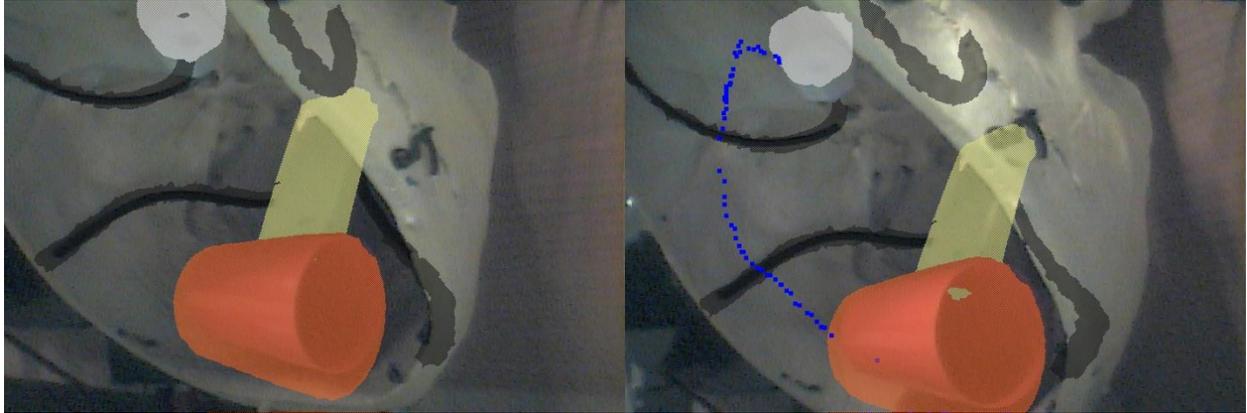
Figure 48: 3D point drawing from cup to nut

In this example, a path was drawn using the end effector of the robot arm. Where the end effector was pointed in the real scene was transformed into the OpenGL world coordinate system and displayed in the appropriate positions in the augmented scene. If points were not desired, the expert could have also drawn with solid lines between points that they chose to create a solid path. All drawn objects are stored in a displaylist of vertex data in addition to the OpenGL primitive requested for that sequence of points.

Danger Zone

The objective of this feature was to provide a method for the user to declare areas within the scene as areas to avoid and provide visual feedback to declare the area and warn of any tools encroaching on the area. An example danger zone is shown in Figure 49.
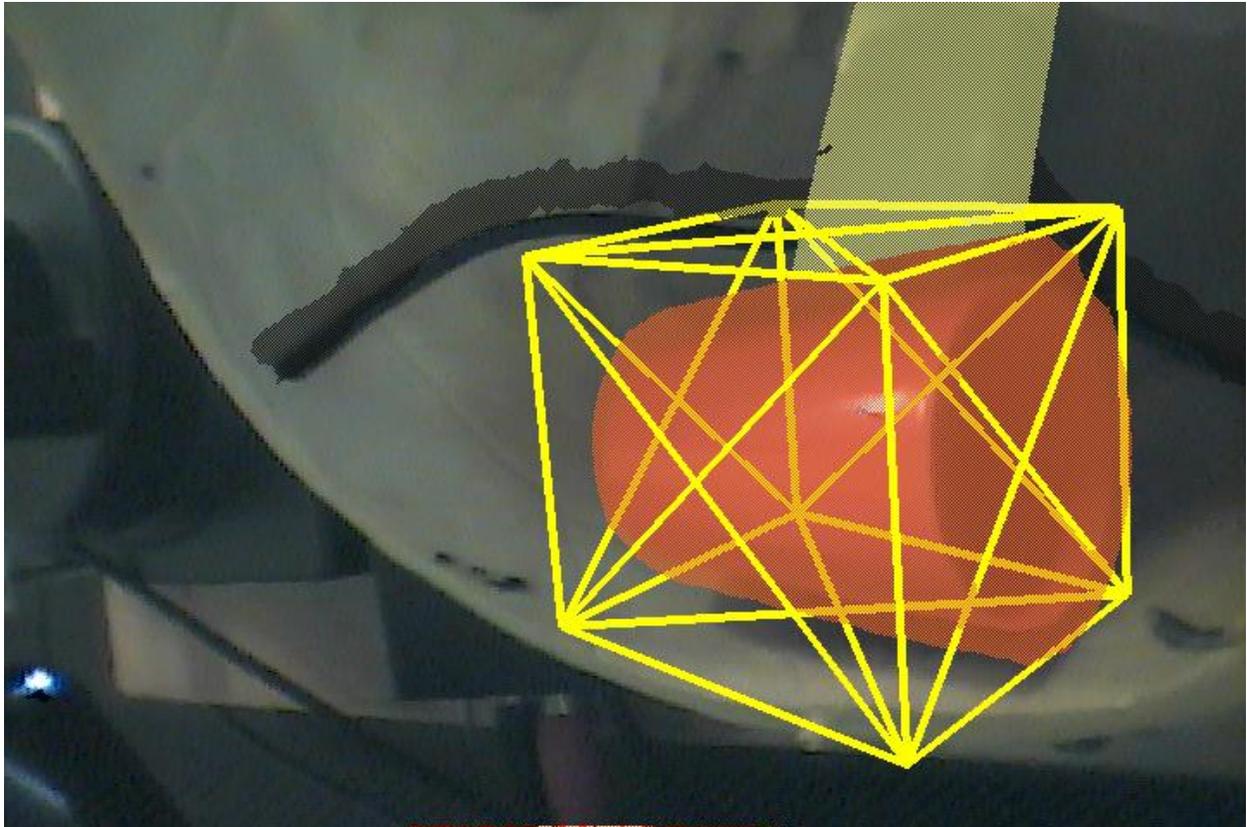
Figure 49: Declared danger zone around cup

In the above example, a danger zone was declared using the end effector of the robot arm. The user only needs to declare two opposing corners of the box by placing the tip of the end effector at the desired box corners in the real environment and capturing the input. Using a zone tolerance, 2cm in this example, when any piece of tracked hardware gets within the tolerance of the danger zone, the box starts flashing red and white to warn the user. If it were too dangerous, or inconvenient, to declare the danger zone by boxing it with tools, the zone could also be pre-operatively declared in the environment and appear in exactly the same way. It will still move and rotate with the viewpoint of the environment and will watch out for tracking items getting close to it.

Additional Hardware

The examples shown previously in this chapter have utilized the Microscribe robotic arm to track the position and orientation of the camera and interact with the environment through its end effector. With the decoupled state of the tracking server from the AR client software, we can utilize multiple tracked pieces of hardware in the environment. We have also utilized an infrared tracking system called the Polaris Accedo from Northern Digital Incorporated. Using a camera or a pointing tip attached to lexan adorned with three infrared tracking spheres, the system can build a transformation between the object and the device and return that to the AR client. This line-of-sight based setup just allows an additional individual to be able to augment the environment displayed in the AR client.

Discussion

The objective of this research aim was to come up with augmented reality features and an augmented reality system that could improve the utility of telementoring systems. This aim has presented a number of architectural design choices and system features to attempt to attain that goal.

The base system was designed around the concept that hardware tracking and augmentation viewing should be separated and networked together. This networking has allowed the system to support multiple tools interacting with the environment, and multiple displays to be generated for different users. Multiple tool tracking allows users at the scene to interact directly with the object of interest, and also allows anyone not on site to interact with some other reference to the scene. It also allows the offsite user to view the scene themselves on another AR client.

Integrating pre-operative imaging into the viewable scene offers a number of advantages to the user. They have the ability to have the imaging overlaid directly on the object of interest so that important structures can be identified just by looking at them rather than looking at the images and performing additional mental computations to align and transform the image to the position and orientation of the scene.

Providing axial, coronal, and sagittal slices of the CT volume was a logical choice. The scans we had were created axially, and generating the other two orthogonal viewing angles for the volume was not computationally expensive. It also provides the same three viewing angles that the doctors are usually presented within medical imaging software packages. The desired slice can be easily translated through. Control over the translucency of the slices was a necessary feature in this area. The scans could be presented completely opaque, but to allow the user to see everything that is happening behind the medical imaging, and align objects in the image, reducing the opacity allows them to see everything they want clearly.

Registration of the imaging to the scene provides the positive benefit of being able to then register annotations to the imaging and the scene. 2.5D drawing was introduced in the program. The expert can pull up a desired slice of CT data and the slice will be expanded to their view. They can then draw on the slice in 2D and reinsert that drawing back into the volume. After that is done, the drawing now exists in 3-dimensions and rotates and translates around, registered to the object. Instead of having to deal with a video registered annotation that moves with the camera, this annotation moves with the environment so that it is annotating the object that it was supposed to, no matter the viewing angle.

The viewport parallel 3D slice through the CT volume was an interesting problem. Implementation-wise, it required significant optimization to run on all of our demo machines. The objective was to maintain real-time rendering at all times. Even an ATI Radeon Mobility 9600 card with 64MB of memory was able to maintain greater than 30fps. This card could not handle non-power-of-two texturing, so the 3D volume of texture data completely filled the video memory. A sub-quarter-second pause occurs at the change to 3D slice mode, but everything runs smoothly after that. Any hardware newer than that video card ran the software without any hitches.

From a usability standpoint, the parallel 3D slice viewing had a few interesting notes. Initial versions of the software allowed the user to translate the 3D slice just as you can translate the axial, coronal, and sagittal slices. This, however, has limited utility. Since the slice is parallel to the viewport, the only reference you have to the distance to the slice is the size of the slice. The slice is not a set size because it depends on the orientation of the cube the parallel plane is intersecting with. Unless the camera is very close to the object, you cannot get a reference for how far away the slice is. All the user can see is that the slice that appears on screen is changing. It was decided after initial testing to restrict the 3D slice to sit at the center of the object at the world origin point. With a fixed reference, the user can at least gauge where they are looking.

A few features were deemed unworkable for the 3D slice viewing. Drawing on a 3D slice was not particularly useful. The program could generate a 2D slice to draw on that could be expanded to the size of the viewport for drawing without any problem. It could even mask off the areas that have no texture to draw on because the intersection slice is not necessarily

rectangular even though the viewport is. The problems came in a couple forms. Firstly, the addition of a drawing texture volume to blend with the CT volume texture to obtain the final image would require another 64MB of video memory on cards not supporting non-power-of-two texture dimensions. This was a significant performance decrease on cards with less than 128MB of video memory. The more important problem related to the alignment of the slice. If the expert draws on axial slice 13 and inserts it back into the volume, any time the user views axial slice 13, the complete drawing is present on that slice. If the viewport parallel 3D slice is pulled and drawn on, the only time the user will ever see that complete drawing is when the slice is aligned in nearly the exact same angle again. If the camera angle changes at all around the scene, the user will only see the part of the drawing plane that intersects with the currently viewed texture plane. This can be alleviated by making the drawing permanent so that it appears consistently in the scene regardless of the slice being viewed, but the other drawing methods can result in permanent annotation also.

The 3D drawing was a way to introduce 3D drawing mechanically in addition to the 3D annotated result. This requires a little more effort from the expert drawing in the scene, but the end result could be even more useful to the user. A path could be drawn in the scene that provides a roadmap for the user. Since the annotations move with the scene when the camera moves, the camera operator could follow the path directly while looking at the video feed. Paths and motions that exist in 3D require less mental manipulation to follow than interpreting a 2D projection of the same thing in the environment.

The danger zone feature is one that we have seen latent demand for in the operating room. Being able to set up effective "no fly" zones in the scene pre-operatively or during the

procedure has many advantage. If a brain surgeon was entering the brain to perform a procedure, regions of the brain that they do not want to damage because they will severely affect the patient can be declared danger zones. Blood vessels that they do not want to sever could also be declared danger zones. With those set up, they could be following a path that was laid out in 3D drawing, but if any of their tools ever get too close to any of the protected areas, they could receive a warning before doing any harm to the patient. This visual warning could easily be supplemented by an audible warning. In addition, if tied to a robotic surgical system, the feedback could potentially be given to the robot to say it should not allow the user to move the tools there.

The work from this aim has been included as a section in two larger publications. The first was CARS 2008 [51]. The second was SAGES 2008 [52].

CHAPTER 4 – CONCLUSIONS

Summary

Laparoscopic surgery provides many benefits over open surgery to the patient, but it is a difficult task to perform that requires a skill set that has an involved learning process. This research was an attempt to use a recent technology, augmented reality, to assist in the surgical operation and in the teaching of those skills.

The first aim of the research was designed to assist in the communication between the surgeon and the camera operator. A set of augmented reality cues were created to describe the possible actions that can be performed with a laparoscopic camera. Combined with a system that we designed, a surgeon can look at the camera feed on the video monitor and an indicator will appear on screen to tell the camera operator where the surgeon would like the camera pointed. A tilt of the head or a lean of the body can indicate that they would like the camera zoom level changed or the horizon adjusted with a rotation. All of this can be done without adding to the distracting chatter in the operating room.

An experimental study was performed based on this aim. We ran 23 subjects through a total of 125 trials where they were required to look at four different specially designed targets in the correct position, zoom level, and roll angle. Half of the subjects ran through the experiment receiving verbal cues, while the other half received our augmented reality cues. The study found that, in all actions performed with the scope, the augmented reality cue trials were completed faster and with better economy of movement than the verbal trials.

The second aim of the research was designed to implement pre-operative imaging in a way that could benefit doctors in the operating room, in addition to developing registered augmented reality features to improve the utility of telementoring.

We designed an augmented reality platform that can take pre-operative imaging and register it to a patient on the table. This results in medical scans and important biological structures being overlaid directly on top of the patient rather than requiring the doctor to look at a scan on a separate film and estimate the location of points on the scan when looking at the patient. Our system can display the orthogonal views of a medical scan volume and translate through the slices. It can generate a slice of imaging data that is perpendicular to the viewing angle, allowing the surgeon to more around the patient and get a constantly update viewpoint of the inside of the patient.

Drawing abilities were implemented that do not move around with the video feed like most telestration systems available. The annotations provided are registered to the environment and always appear where they were intended. An expert in the operating room, or an expert on the other side of the planet can tap into the system and annotate in 2D on the medical image slices and they are integrated into the slices and projected back into 3D. The expert can draw out paths in 3D in the 3D environment. The system can also be used to set up danger zones within the patient so that if any tools from the surgeon get close to an area they should not be going, warning signs can be given to the surgeon.

All of these features add up to a system that can make the usage of pre-operative imaging more convenient, and can also open up many possibilities for telementoring of surgical procedures by improving the capabilities of the system.

Future Work

Combination of Aims

The future is actually the fusion of both aims of this dissertation enabled by the advancement of technology. We appear to be on the cusp of wearable displays being produced in high enough volume that the economy of scale will bring the costs down to an affordable level for the general public. Multiple devices, such as Google Glass, are appearing in limited quantities to test out hardware and build up software support. These devices resemble eyeglasses, except that they have a translucent display in the glass. This allows for the projection of computer graphics on a screen that you can look through.

An end goal would be to have a tracking system similar to Aim 1 track where the surgeon is looking. With the sight path known, the augmentation for the patient could be projected into the glasses so that they overlay directly on the real world. There is no need to be looking at a video monitor to watch a feed when the world around you can be changed by your own glasses.

Registered Telementoring with 3D Cameras

We explored this area previously, but as with the last item, technological advances are expanding the possibilities. Telementoring is a field that has made advancements over the years with the improvements in available technology. Changes in video capture, video processing, and communication networks have all contributed to increased capabilities in these systems. Much work has been done in the field, and commercial products have started to appear for this market. A number of problems still remain. Most of the systems on the market

are nothing more than telestration systems. The drawing is only in 2D, and it is registered to the video frame rather than the objects in the environment. This problem is compounded by the high bandwidth requirements for most of these systems. The entire video frame with annotations is transmitted to every receiver. Our strategy was to expand upon our Aim 2, allowing the system to take imaging on site rather than pre-operatively and at the same time reducing the bandwidth and latency issues inherent with other telementoring systems.

This requires the mapping of the environment in real-time. This is a very complex process that has many years of research behind it. Groups have tried to identify points in the video stream and follow them across frames to determine their movement in relation to each other, and therefore their 3D structure [53]. Others have used stereo cameras of known geometry to identify like pixels in each image and calculate 3D structures [54]. Structured light has been used [55]. Even pulsed light sources of known location followed by shadow analysis has been utilized [56]. All of these methods are not perfect. Most have problems mapping featureless objects or environments, or ones with materials of different light absorption properties.

As of this last year, there is a mass-produced commercial camera on the market from PrimeSense that uses an IR-based structured-light output to generate a distance from the camera of each pixel within view. After years of evaluating expensive time-of-flight cameras, this inexpensive camera appears to do a very good job. With a known camera position and orientation, and the pixel distance data, a 3D mesh of the environment can be created. The image data and depth data can be sent to the mentor with a timestamp. The mentor can stop that stream to draw on it, and to save upload bandwidth, only the drawing, its location, and its

timestamp will be returned. This is unlike the aforementioned systems that send the entire edited frame back. No matter how much latency was involved, when the drawing gets back to the student, the machine will calculate the position of the drawing in the environment and it will be drawn in the correct location regardless of where the camera is currently looking.

Multiple cameras have been released this year, or will see release soon. One of these is a time-of-flight camera that is one of the first of its kind to have a reasonable price. Other cameras have been working at very close ranges for the first time. With the advancement of 3D cameras on the market, this research path might be an interesting one to follow.

REFERENCES

[1]   J. D. Richardson, "Surgeon Shortage in the US: Fact or Fiction, Causes, Consequences," in *Healthcare Transparency and Patient Advocacy*, Lexington, 2009.

[2]   M. Noyes, "Superposition Of Graphics On Low Bit Rate Video As An Aid In Teleoperation," Massachusetts Institute Of Technology, Cambridge, 1984.

[3]   S. Feiner, B. MacIntyre and D. Seligmann, "Knowledge-Based Augmented Reality," *Communications Of The ACM,* vol. 36, no. 7, pp. 53-62, July 1993.

[4]   H. Kato and M. Billinghurst, "Tracking and HMD Calibration for a video-based Augmented Reality Conferencing System," in *International Workshop on Augmented Reality*, San Francisco, 1999.

[5]   R. Berguer, "Ergonomics in Laparoscopic," in *The Sages Manual: Perioperative Care in Minimally Invasive Surgery*, New York, Springer, 2006, pp. 454-464.

[6]   A. G. Gallagher, M. Al-Akash, N. E. Seymour and R. M. Satava, "An Ergonomic Analysis of the Effects of Camera Rotation on Laparoscopic Performance," *Surgical Endoscopy,* vol. 23, no. 12, pp. 2684-2691, 2009.

[7]   J. Conrad, H. Shah, C. M. Divino, S. Schluender, B. Gurland, E. Shalsko and A. Szold, "The Role of Mental Rotation and Memory Scanning on the Performance of Laparoscopic Skills," *Surgical Endoscopy,* vol. 20, no. 3, pp. 504-510, 2006.

[8]   F. D. Reynolds, L. Goudas, R. S. Zuckerman, M. S. Gold and S. Heneghan, "A Rural, Community-Based Program Can Train Surgical Residents in Advanced Laparoscopy," *Journal of the American College of Surgeons,* vol. 197, no. 4, pp. 620-623, 2003.

[9]   P. Hourlay, "How to Maintain the Quality of Laparoscopic Surgery in the Era of Lack of Hands?," *Acta Chirurgica Belgica,* vol. 106, no. 1, pp. 22-26, 2006.

[10] J. Y. Chung and J. M. Sackier, "A method of objectively evaluating improvements in laparoscopic skills," *Surgical Endoscopy,* vol. 12, no. 9, pp. 1111-1116, 1998.

[11] C. R. Molinas, G. D. Win, O. Ritter, J. Keckstein, M. Miserez and R. Campo, "Feasibility and Construct Validity of a Novel Laparoscopic Skills Testing and Training Model," *Gynecological Surgery,* vol. 5, no. 4, pp. 281-290, 2008.

[12] D. Stefanidis, W. W. Hope, J. R. Korndorffer, S. Markley and D. J. Scott, "Initial Laparoscopic Basic Skills Training Shortens the Learning Curve of Laparoscopic Suturing and Is Cost-Effective," *Journal*

of the American College of Surgeons, vol. 210, no. 4, pp. 436-440, 2010.

[13] M. E. Rosenthal, A. O. Castellvi, M. T. Goova, L. A. Hollet, J. Dale and D. J. Scott, "Pretraining on Southwestern Stations Decreases Training Time and Cost for Proficiency-Based Fundamentals of Laparoscopic Surgery Training," *Journal of the American College of Surgeons,* vol. 9, no. 5, pp. 626-631, 2009.

[14] A. Hyltander, E. Liljegren, P. H. Rhodin and H. Lönroth, "The transfer of basic skills learned in a laparoscopic simulator to the operating room," *Surgical Endoscopy,* vol. 16, no. 9, pp. 1324-1328, 2002.

[15] D. Calatayud, S. Arora, R. Aggarwal, I. Kruglikova, S. Schulze, P. Funch-Jensen and T. Grantcharov, "Warm-up in a Virtual Reality Environment Improves Performance in the Operating Room," *Annals of Surgery,* vol. 251, no. 6, pp. 1181-1185, 2010.

[16] D. Stefanidis, J. R. Korndorffer Jr., R. Sierra, C. Touchard, B. Dunne and D. J. Scott, "Skill Retention Following Proficiency-Based Laparoscopic Simulator Training," *Surgery,* vol. 138, no. 2, pp. 165-170, 2005.

[17] S. M. B. I. Botden, I. H. J. T. de Hingh and J. J. Jakimowicz, "Meaningful assessment method for laparoscopic suturing training in augmented reality," *Surgical Endoscopy,* vol. 23, no. 10, pp. 2221-2228, 2009.

[18] A. Park, D. Witzke and M. Donnelly, "Ongoing Deficits in Resident Training for Minimally Invasive Surgery," *Journal Of Gastrointestinal Surgery,* vol. 6, no. 3, pp. 501-509, 2002.

[19] J. R. Korndorffer, D. J. Hayes, J. B. Dunne, R. Sierra, C. L. Touchard, R. J. Markert and D. J. Scott, "Development and transferability of a cost-effective laparoscopic camera navigation simulator," *Surgical Endoscopy,* vol. 19, no. 2, pp. 161-167, 2005.

[20] C. K. Christian, M. L. Gustafson, E. M. Roth, T. B. Sheridan, T. K. Gandhi, K. Dwyer, M. J. Zinner and M. M. Dierks, "A prospective study of patient safety in the operating room," *Surgery,* vol. 139, no. 2, pp. 159-173, 2006.

[21] D. A. Wiegmann, A. W. ElBardissi, J. A. Dearani, R. C. Daly and T. L. Sundt III, "Disruptions in surgical flow and their relationship to surgical errors: An exploratory investigation," *Surgery,* vol. 142, no. 5, pp. 658-665, 2007.

[22] E. Sutton, Y. Youssef, N. Meenaghan, C. Godinez, Y. Xiao, T. Lee, D. Dexter and A. Park, "Gaze disruptions experienced by the laparoscopic operating surgeon," *Surgical Endoscopy,* vol. 24, no. 6, pp. 1240-1244, 2010.

[23] L. R. Kavoussi, R. G. Moore, J. B. Adams and A. W. Partin, "Comparison of Robotic Versus Human Laparoscopic Camera Control," *The Journal of Urology,* vol. 154, no. 6, pp. 2134-2136, 1995.

[24] S. Aiono, J. M. Gilbert, B. Soin, P. A. Finlay and A. Gordan, "Controlled trial of the introduction of a robotic camera assistant (EndoAssist) for laparoscopic cholecystectomy," *Surgical Endoscopy,* vol. 6, no. 9, pp. 1267-1270, 2002.

[25] J. M. Gilbert, "The EndoAssist Robotic Camera Holder as an Aid to the Introduction of Laparoscopic Colorectal Surgery," *Annals of The Royal College of Surgeons of England,* vol. 91, no. 5, pp. 389-393, 2009.

[26] J. F. M. Rua, F. B. Jatene, J. R. Milanez de Campos, R. Monteiro, M. L. Tedde, M. N. Samano, W. M. Bernardo and J. C. Das-Neves-Pereira, "Robotic versus human camera holding in video-assisted thoracic sympathectomy: a single blind randomized trial of efficacy and safety," *Interactive CardioVascular and Thoracic Surgery,* vol. 8, pp. 195-199, 2009.

[27] P. Ballester, Y. Jain, K. R. Haylett and R. F. McCloy, "Comparison of task performance of robotic camera holders EndoAssist and Aesop," *International Congress Series,* vol. 1230, pp. 1100-1103, 2001.

[28] P. B. Nebot, Y. Jain, K. Heylett, R. Stone and R. McCloy, "Comparison of Task Performance of the Camera-Holder Robots EndoAssist and Aesop," *Surgical Laparoscopy, Endoscopy & Percutaneous Techniques,* vol. 13, no. 5, pp. 334-338, 2003.

[29] A. A. Wagner, I. M. Varkarakis, R. E. Link, W. Sullivan and L.-M. Su, "Comparison of surgical performance during laparoscopic radical prostatectomy of two robotic camera holders, EndoAssist and AESOP: A pilot study," *Urology,* vol. 68, no. 1, pp. 70-74, 2006.

[30] S. Jayaraman, I. Arpiasz, A. L. Trejos, H. Bassan, R. V. Patel and C. M. Schlachta, "Novel Hands-Free Pointer Improves Instruction Efficiency in Laparoscopic Surgery," *Surgical Innovation,* vol. 16, no. 1, pp. 73-77, 2009.

[31] S. Dworzecki, M. Dage, A. Cao, M. Weiner, D. Bouwman and A. Pandya, "Efficient Cues to Improve Laparoscopic Camera Driving," in *iSURGITEC*, Detroit, 2009.

[32] J. C. Rosser, M. Wood, J. H. Payne, T. M. Fullum, G. B. Lisehora, L. E. Rosser, P. J. Barcia and R. S. Savalgi, "Telementoring: A practical option in surgical training," *Surgical Endoscopy,* vol. 11, no. 8, pp. 852-855, August 1997.

[33] L. Panait, A. Rafiq, V. Tomulescu, C. Boanca, I. Popescu, A. Cabonell and R. C. Merrell, "Telementoring versus on-site mentoring in virtual reality-based," *Surgical Endoscopy,* vol. 20, no. 1, pp. 113-118, January 2006.

[34] C. M. Schlachta, K. L. Lefebvre, A. K. Sorsdahl and S. Jayaraman, "Mentoring and telementoring leads to effective incorporation of laparoscopic colon surgery," *Surgical Endoscopy,* vol. 24, no. 4, pp. 841-844, April 2010.

[35] P. G. Schulam, S. G. Docimo, W. Saleh, C. Breitenbach, R. G. Moore and L. Kavoussi, "Telesurgical mentoring: Initial clinical experience," *Surgical Endoscopy,* vol. 11, no. 10, pp. 1001-1005, October 1997.

[36] B. R. Lee, J. T. Bishoff, G. Janetschek, P. Bunyaratevej, W. Kamolpronwijit, J. A. Cadeddu, S. Ratchanon, S. O'Kelley and L. R. Kavoussi, "A novel method of surgical instruction: international telementoring," *World Journal Of Urology,* vol. 16, no. 6, pp. 367-370, 1998.

[37] J. C. Rosser Jr., R. L. Bell, B. Harnett, E. Rodas, M. Murayama and R. Merrell, "Use of Mobile Low-Bandwith Telemedical Techniques for Extreme Telemedicine Applications," *Journal of the American College of Surgeons,* vol. 189, no. 4, pp. 397-404, October 1999.

[38] M. Cubano, B. K. Poulose, M. A. Talamini, R. Stewart, L. E. Antosek, R. Lentz, R. Nibe and M. F. Kutka, "Long distance telementoring: A novel tool for laparoscopy aboard the USS Abraham Lincoln," *Surgical Endoscopy,* vol. 13, no. 7, pp. 673-678, July 1999.

[39] E. Taniguchi and S. Ohashi, "Construction of a Regional Telementoring Network for Endoscopic Surgery in Japan," *IEEE Transactions on Information TEchnology in Biomedicine,* vol. 4, no. 3, pp. 195-199, September 2000.

[40] P. Bove, D. Stoianovici, S. Micali, A. Patriciu, N. Grassi, T. W. Jarrett, G. Vespasiani and L. R. Javoussi, "Is Telesurgery a New Reality? Our Experience with Laparoscopic and Percutaneous Procedures.," *Journal of Endourology,* vol. 17, no. 3, pp. 137-142, April 2003.

[41] N. R. Netto Jr., A. I. Mitre, S. V. C. Lima, O. E. Fugita, M. L. Lima, D. Stoianovici, A. Patriciu and L. R. Kavoussi, "Telementoring Between Brazil and the United States: Initial Experience," *Journal of Endourology,* vol. 17, no. 4, pp. 217-220, May 2003.

[42] H. Sebajang, P. Trudeau, A. Dougall, S. Hegge, C. McKinley and M. Anvari, "Telementoring: An Important Enabling Tool for the Community Surgeon," *Surgical Innovation,* vol. 12, no. 4, pp. 327-331, December 2005.

[43] H. Sebajang, P. Trudeau, A. Dougall, S. Hegge, C. McKinley and M. Anvari, "The role of telementoring and telerobotic assistance in the provision of laparoscopic colorectal surgery in rural areas," *Surgical Endoscopy,* vol. 20, no. 9, pp. 389-393, September 2006.

[44] M. Anvari, "Remote telepresence surgery: the Canadian experience," *Surgical Endoscopy,* vol. 21, no. 4, pp. 537-541, April 2007.

[45] R. Thirsk, D. Williams and M. Anvari, "NEEMO7 undersea mission," *Acta Astronautica,* vol. 60, no. 4, pp. 512-517, February 2007.

[46] T. Haidegger and Z. Benyo, "Surgical robotic support for long duration space missions," *Acta Astronautica,* vol. 63, no. 7, pp. 996-1005, October 2008.

[47] A. Pandya, "Medical Augmented Reality System For Image Guided And Robotic Surgery: Development And Sugeon Factors Analysis," Detroit, 2004.

[48] A. Fedorov, R. Beichel, J. Kalpathy-Cramer, J. Finet, J.-C. Fillion-Robin, S. Pujol, C. Bauer, D. Jennings, F. Fennessy, M. Sonka, J. Buatti, S. R. Aylward, J. V. Miller, S. Pieper and R. Kikinis, "3D Slicer as an Image Computing Platform for the Quantitative Imaging Network," *Magnetic Resonance Imaging,* vol. 30, no. 9, pp. 1323-1341, 2012.

[49] A. Pandya, M.-R. Siadat and G. Auner, "Design, implementation and accuracy of a prototype for medical augmented reality," *Computer Aided Surgery,* vol. 10, no. 1, pp. 23-35, 2005.

[50] J.-Y. Bouguet, "Camera Calibration Toolbox for Matlab," 10 October 2013. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/. [Accessed 25 October 2013].

[51] B. King, L. Reisner, S. Dworzecki, A. Cao, H. Wills, G. Auner, M. Klein and A. Pandya, "Fusion of Tomographic Imaging with In Vivo Raman Cancer Diagnosis using Augmented Reality," in *Computer Assisted Radiology and Surgery*, Barcelona, 2008.

[52] H. Wills, B. King, L. Reisner, S. Dworzecki, R. Kast, A. Cao, A. Pandya, G. Auner, M. Klein, C. Steffes, M. Prasad and D. Weaver, "Robotically Positioned Raman Spectroscopy Probe for Cancer Resection," in *Society of American Gastrointestinal and Endoscopic Surgeons*, Philadelphia, 2008.

[53] C. Bregler, A. Hertzmann and H. Biermann, "Recovering Non-Rigid 3D Shape from Image Streams," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[54] F. Morgues, F. Devernay and E. Coste-Maniere, "3D reconstruction of the operating field for image overlay in 3D-endoscopic surgery.," in *IEEE and ACM International Symposium on Augmented Reality*, New York, 2001.

[55] S. Rusinkiewicz, O. Hall-Holt and M. Levoy, "Real-Time 3D Model Acquisition," in *SIGGRAPH*, New York, 2002.

[56] A. T. Saucedo, F. M. Santoyo and M. De la Torre-Ibarra, "Endoscopic pulsed digital holography for 3D measurements," *Optics Express,* vol. 14, no. 4, pp. 1468-1475, February 2006.

ABSTRACT

ADVANCED AUGMENTED REALITY TELESTRATION TECHNIQUES WITH APPLICATIONS IN LAPAROSCOPIC AND ROBOTIC SURGERY

by

STEPHEN TERRENCE DWORZECKI

December 2013

Advisor: Dr. Abhilash Pandya

Major: Computer Engineering

Degree: Doctor of Philosophy

The art of teaching laparoscopic or robotic surgery currently has a primary reliance on an expert surgeon tutoring a student during a live surgery. During these operations, surgeons are viewing the inside of the body through a manipulatable camera. Due to the viewpoint translation and narrow field of view, these techniques have a substantial learning curve in order to gain the mastery necessary to operate safely. In addition to moving and rotating the camera, the surgeon must also manipulate tools inserted into the body. These tools are only visible on camera, and pass through a pivot point on the body that, in non-robotic cases, reverses their directions of motion when compared to the surgeon's hands. These difficulties spurred on this dissertation. The main hypothesis of this research is that advanced augmented reality techniques can improve telementoring for use between expert surgeons and surgical students. In addition, it can provide a better method of communication between surgeon and camera operator.

This research has two specific aims:

(1) Create a head-mounted direction of focus indicator to provide non-verbal assistance for camera operation. A system was created to track where the surgeon is looking and provides augmented reality cues to the camera operator explaining the camera desires of the surgeon.

(2) Create a hardware / software environment for the tracking of a camera and an object, allowing for the display of registered pre-operative imaging that can be manipulated during the procedure.

A set of augmented reality cues describing the translation, zoom, and roll of a laparoscopic camera were developed for Aim 1. An experiment was run to determine whether using augmented reality cues or verbal cues was faster and more efficient at acquiring targets on camera at a specific location, zoom level, and roll angle. The study found that in all instances, the augmented reality cues resulted in faster completion of the task with better economy of movement than with the verbal cues.

A large number of environmentally registered augmented reality telestration and visualization features were added to a hardware / software platform for Aim 2. The implemented manipulation of pre-operative imaging and the ability to provide different types of registered annotation in the working environment has provided numerous examples of improved utility in telementoring systems.

The results of this work provide potential improvements to the utilization of pre-operative imaging in the operating room, to the effectiveness of telementoring as a surgical teaching tool, and to the effective communication between the surgeon and the camera operator in laparoscopic surgery.

AUTOBIOGRAPHICAL STATEMENT

With the completion of this dissertation, Stephen Dworzecki will obtain a Ph.D. in Computer Engineering from Wayne State University. Stephen already holds a Master of Science degree in Computer Engineering from Wayne State University in addition to a Bachelor of Science degree in Computer Engineering from Lawrence Technological University.