

# **Surgical Simulator for Endoscopic Carpal Tunnel Surgery**

December 8th, 2010

*Team:*

Charles Donaldson - Team Leader

Whitney Johnson - Communicator

Sarah Switalski - BWIG

Mason Jellings - BSAC

*Clients:*

Dr. Benjamin Mandel

Dr. Robert Radwin, UW-Madison

*Advisor:*

Prof. Willis Tompkins, Ph.D.

## **Abstract**

Surgical simulations are becoming an integral part of surgeon's training due to decreased cost and increased repeatability. The objective is to create a prototype that interfaces a physical anatomical model with a virtual software program for endoscopic carpal tunnel surgery. Based on specifications from the client, the final prototype includes a silicone rubber hand model and a tracking device using infrared LEDs that are followed by a Wii remote. For the silicone hand, anatomical structures in the carpal tunnel region were modeled by suspending a rigid tube in a matrix consisting of softer silicone. A dense strip of silicone containing corrugations on its underside was placed at the roof of the tube, providing a structure representing the transverse carpal ligament. The LEDs are attached to a circuit board on the back of the endoscope, and the Wii remote views the LEDs from behind the working camera view. The Wii communicates with a Java program via Bluetooth. Testing was performed to determine the optimal distance between the Wii remote and the LEDs. Results indicated that no significant difference in accuracy exists between the tested distance parameters. Future work includes incorporating haptic feedback into the simulation, creating a more realistic surgical experience.

## **Table of Contents**

Background .....	5
<i>Motivation</i> .....	5
<i>Surgical Procedure</i> .....	5
Problem Statement.....	7
Client Specifications.....	7
Current Devices.....	8
Design Alternative – Hand Model .....	10
<i>3D Printing</i> .....	10
<i>Alginate/Ballistics Gel</i> .....	11
<i>Silicone</i> .....	11
Decision Matrix – Hand Model.....	12
Design Alternatives – Tracking Device.....	14
<i>Trackball</i> .....	14
<i>Accelerometer/Gyroscope</i> .....	15
<i>Wii Remote</i> .....	16
Decision Matrix – Tracking Device.....	17
Final Design.....	18
<i>Ergonomic Considerations</i> .....	20
<i>Fabrication</i> .....	21
Testing and Analysis.....	21

Future Work.....	24
Acknowledgements.....	25
References.....	25
Appendix A: Product Design Specifications.....	27
Appendix B: Java Code.....	29
Appendix C: Test Data.....	36

## **Background**

### *Motivation*

The surgical field is moving towards using only simulators for the teaching and learning of surgical procedures and moving away from the use of cadavers. Practicing surgical procedures on cadavers has been common practice in the surgical field for years. However, this practice comes with some downfalls. The cadavers can only be used once for a particular surgery and over time, the process becomes an expensive practice.

To improve on these problems, surgical simulations can take the place of cadavers. Simulations provide a solution to the problem because of the durability and reproducibility of the procedures. Instead of using several cadavers, a single simulator could be used to teach multiple surgeons how to do a surgical procedure.

Here in lies the motivation for a simulator for endoscopic carpal tunnel surgery. The surgery is relatively simple and only takes a short time to complete. Instead of wasting time and money on cadavers to practice, there is a need for a simulator to teach surgeons how to complete the procedure. This would allow for surgeons to learn how to do the surgery multiple times and also teach them the proper way to perform the surgery.

### *Surgical Procedure*

Endoscopic carpal tunnel surgery is a non-invasive procedure that reduces the symptoms of carpal tunnel syndrome. With this syndrome, the median nerve is pinched by the swelling of the surrounding tendons. This can stem from a variety of causes such as repetitive movements of the wrist and fingers, genetic predisposition, and sometimes injury or illness [1]. To alleviate the stress, an endoscope with a trigger activated blade is used to complete the surgical procedure.

Before the endoscope is inserted, an incision is made on the distal wrist crease of the hand. Surgical tools are inserted into the incision to widen the carpal tunnel for the endoscope blade. During this

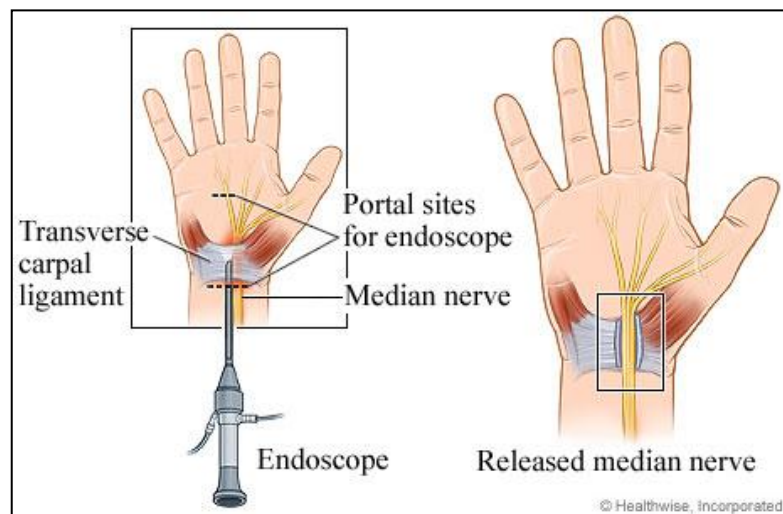


Figure 1: Placement of endoscope and cutting of transverse carpal ligament. (<http://www.health.com/health/library/mdp/0,,zm2464,00.html>)

time, the surgeon is able to feel the corrugations on the underside of the tendon, allows him or her to sense the location and borders of the transverse carpal ligament. The endoscope also comes equipped with an optical camera attached to the back and allows the surgeon to see inside of the wrist during surgery. The camera displays a picture on a screen, allowing the surgeon to visualize the surgery. Before insertion into the hand, the surgeon pulls the trigger and looks at the display screen to check and see if the optical camera is working correctly. Once all the pre-surgical tests are complete, the surgeon inserts the endoscope into the incision.

As the surgeon slowly pushes the blade further into the wrist, he or she is constantly checking the display screen and feeling with his other hand to make sure that the endoscope stays in place. The blade tip must stay in line with the ring finger of the hand during surgery in order to avoid causing damage to other structures in the carpal tunnel area. Once the end of the transverse carpal ligament is located, the blade is deployed by the surgeon by pulling the trigger on the endoscope. The surgeon then slowly starts to pull back on the endoscope to cut the ligament until the blade exits the incision. To check if the carpal ligament is completely cut, the endoscope is reinserted into the carpal tunnel. If the display shows a picture like that seen in Figure 2, the surgery is complete as the tendon is completely cut. To finish the procedure, the endoscope is removed, the incision is sewn up, and the patient is allowed to recover from the operation.



Figure 2. Endoscopic camera view of carpal tunnel after cutting of transverse carpal ligament. [3]

By cutting the transverse carpal ligament, the pressure caused by swelling of the surrounding tendons is removed from the median nerve [2]. This allows the nerve to return back to its normal function and eliminate the symptoms of carpal tunnel syndrome.

During the surgery, the surgeon uses both the feel and vision provided by the endoscope. As stated before, when the endoscope is inside of the wrist, the surgeon uses his free hand to feel where the endoscope is within the wrist. The surgeon also feels resistance from the tendon when the blade is cutting it. This tells the surgeon that the blade is actually cutting the tendon, and not the surround tissue. Also, the surgeon is using the optical camera to guide the blade and endoscope while inside the hand. The combination of physical feel and visualization during the operation allows the surgeon to be as accurate and safe as possible.

## **Problem Statement**

The methods to practice surgical procedures have begun to change in the medical field. Instead of using expensive cadavers to practice, many surgeons are starting to utilize simulators. The team's objective this semester was to design a simulator that will be used as a training device for endoscopic carpal tunnel surgery by interfacing a life-like physical hand model with a virtual 3-D representation of the wrist and hand. The simulator must replicate both the haptic feedback and visualization of the surgery in order to provide the most accurate teaching experience.

## **Client Specifications**

During carpal tunnel release surgery, surgeons greatly rely on direct sight and feel to accurately locate anatomical structures. As such, the hand model must resemble that of a patient, helping to create a realistic simulation experience. A skin-like material must cover the exterior of the model. As the simulation solely relies on a virtual environment to visualize the inside of the carpal tunnel, complete anatomical accuracy of the inner hand is not required. Being surrounded by bone and dense connective tissue, the carpal tunnel portion of the model should resemble a restrictive tube; this will provide the realistic resistance of neighboring tissue. Corrugations must be present on the top of the tunnel to reflect the texture on the underside of the transverse carpal ligament.

Measurements of anatomical structures (see Figure 3):

1. Transverse carpal ligament width – distal wrist crease to mid palm
2. Ligament corrugations – ~1 mm by 1 mm
3. Carpal tunnel diameter – 1-2 cm

The tracking system must interface with the virtual carpal tunnel created by the client. It must relay information relating the position of the instrument to a virtual position in the simulation environment. The system should detect movement with 6 degrees of freedom and have a precision of 2 mm. External devices, such as wires and receivers, should not interfere with the surgical procedure.

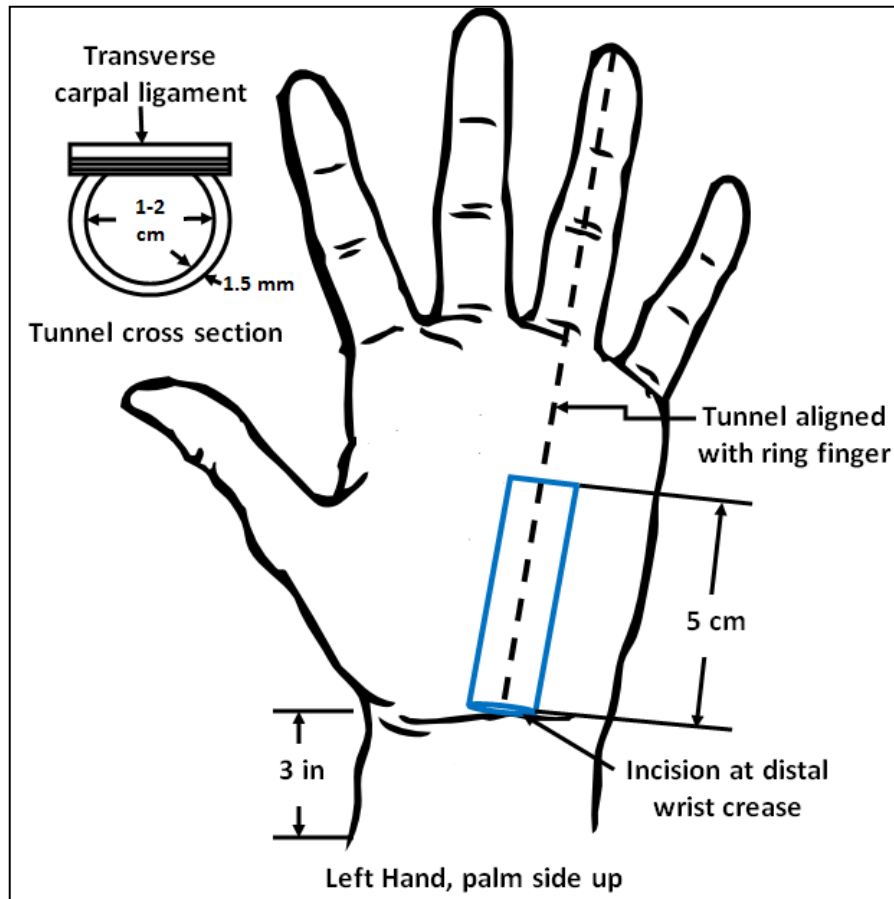


Figure 3. Schematic of carpal tunnel placement within hand model. The carpal tunnel is highlighted in blue and is aligned with the ring finger. The incision is made at the distal wrist crease, and the carpal tunnel is approximately 5cm in length, extending to mid-palm. The cross section view gives dimensions of the carpal tunnel tube.

## Current Devices

Surgical simulators are becoming a popular method to train surgeons due to decreased costs and increased repeatability. Many different systems are available with open source technology packages meant to interface existing simulation instruments with visual processing software specific to a surgery of interest. Tracking devices can be categorized into three general types: mechanical, optical, and electromagnetic. Simulation systems such as TrEndo [4] and SIMENDO offer haptic feedback and virtual visualization, but lack a physical anatomical model (see Figures 4 and 5 on next page).





Figure 4. (left) displays the TrEndo surgical simulation system. The handheld endoscopic instruments mechanically couple with a device that uses optics to detect movement ([www.ncbi.nlm.nih.gov/pmc/articles/PMC2211365/figure](http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2211365/figure)). Figure 5. (right) displays the set-up for TrEndo simulation system. The instruments allow for advanced haptic feedback throughout the surgery ([www.ncbi.nlm.nih.gov/pmc/articles/PMC2211365/figure](http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2211365/figure)).

Hybrid surgical simulators allow for interaction between virtual and physical models. The surgeon views virtual images on a monitor, but uses real instruments and has the option of studying a physical structure. Most hybrid systems, such as ProMIS (see Figures 6 and 7), simulate laparoscopic surgery. No device currently exists for the simulation of carpal tunnel release surgery. The hybrid concept of integrating physical with virtual will be utilized in the design of the prototype.



Figure 6 (left) shows the ProMIS surgical simulator utilizing virtual graphics to complete the surgery. Figure 7 (right) shows the real anatomical model included with the ProMIS system. (<http://www.haptica.com/>)

## **Design Alternatives – Hand Model**

The purpose of the physical hand model in the simulation is to provide the users with haptic feedback similar to what is felt during an actual endoscopic carpal tunnel release surgery. The inside of the physical hand model will not be observed because the camera view will be based on images predetermined from a previous surgery. As mentioned in the client specifications, the only elements needed are a restrictive carpal tunnel tube and the corrugations on the underside of the transverse carpal ligament to provide the correct haptic feedback. Therefore, the team chose various materials in the design alternatives that have similar resistance and mechanical properties as the ligaments and tissues in the carpal tunnel.

### *3D Printing*

There were three design alternatives when considering the physical hand model. The first design utilized 3D printing to create a model with the necessary anatomical accuracy. The team investigated the process and cost of producing a hand model with Proto Labs, Inc., which has a Protomold® injection molding service to make a prototype of a 3D CAD model, similar to Figure 8. The formats accepted for injection molding included IGES (.igs) files, STL (.stl) or stereo lithography files, and Solid Works (.sldprt) files, among others. These files would be created by importing a CT scan of the hand and wrist. The injection mold would be made out of a soft polymer to create a life-like feel when the model is used in the surgery simulation. However, because the prototype would be based off of a CT scan, all of the anatomical structures inside the hand would be included. As mentioned in the client specifications, the only internal anatomical structures needed were the carpal tunnel and the transverse carpal ligament. Using 3D printing provides more anatomical accuracy than needed for the scope of the simulation. Therefore, due to the high complexity of the injection mold, the price from Proto Labs, Inc. would be quoted at approximately \$1495 [5].

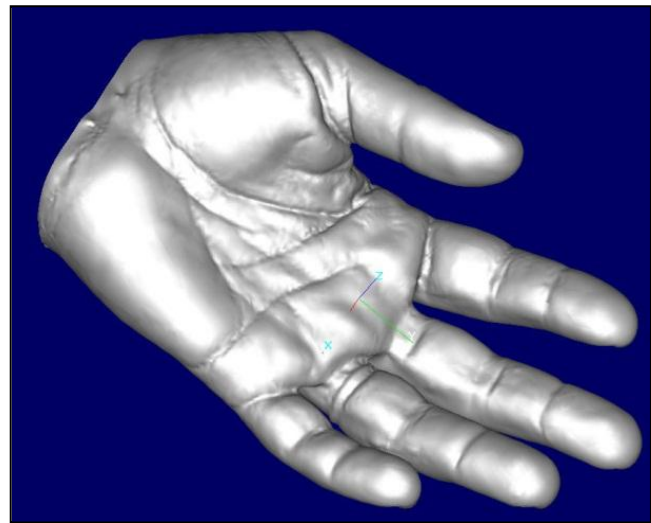


Figure 8. 3D CAD representation of the hand based off of a CT scan. (<http://www.rhinoreverse.icapp.ch/english/gallery.html>)

### *Alginate/Ballistics Gel*

The second design alternative involved making a physical hand model by pouring ballistics gel into an alginate cast mold. Ballistics gel was chosen because it is used as a standardized medium to test the terminal performance of firearms ammunition [source], and therefore has the density, viscosity, and mechanical properties similar to human tissue. First, an outer mold of a team member's hand would be created using a quickly-forming cast of alginate (Figure 9). The ballistics gel would be poured into this mold. As the gel is setting, a soft flexible tube would be embedded to represent the carpal tunnel. In addition, coarse fibers would be inserted around the carpal tunnel tube to serve as anatomical landmarks inside the wrist. This would provide a more realistic feel in the surgery simulation. The cost of creating this design would be minimal; the alginate kit costs approximately \$50, and the ballistics gel is \$10/lb [6]. However, because ballistics gel does not maintain its shape for longer than a few days, this hand model would not provide the durability required by the client.



Figure 9. Alginate mold of hand and wrist, which would be filled with ballistics gel [6].

### *Silicone*

The final design alternative utilizes a combination of silicones to create a hand model. Silicone rubbers are inert polymers that include silicon with carbon, hydrogen, and oxygen [7]. Coming in a variety of hardness values, it is possible to find varieties of silicone that mimic the resistance of human tissue. In addition, multiple types of silicone can be combined in the same mold to create variability in haptic feedback, and silicone has high durability and reusability. In

the hand model, a firmer silicone would be shaped as a tube to form the carpal tunnel and suspended in a softer gel (Figure 10). A silicone with a color and texture similar to human skin would cover the outer surface of the hand model to provide a realistic appearance. As an additional feature, the hand would have a modular palmar section, making repairs on the model easily accessible. The team would work with Greg Gion at Medical Art Prosthetics, LLC. to create the model, and the cost would be \$300.

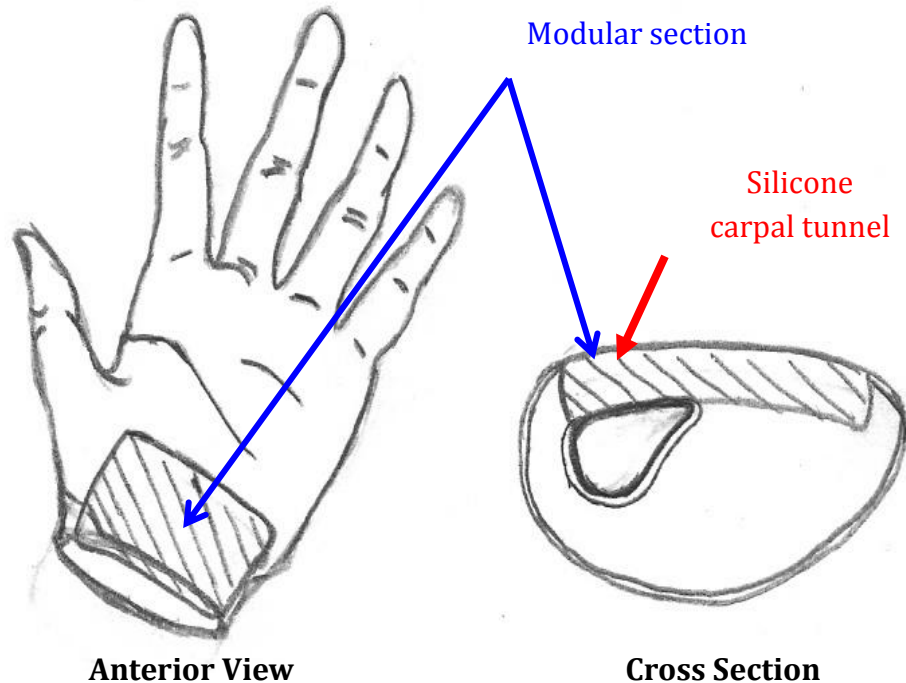


Figure 10. Sketch of silicone hand model, emphasizing the modular palm and carpal tunnel area.

### **Decision Matrix – Hand Model**

A decision matrix was used to analyze the three hand model design alternatives. The designs were evaluated based on five categories: Resistance/Haptics, Anatomical Accuracy, Durability, Aesthetics, and Cost. Resistance/Haptics was given the highest weight of 40/100 because this determines how effective the model is in providing a life-like feel during the simulation. Because the model needed to withstand multiple simulations while providing reproducible haptic feedback, Durability was weighted second-highest with 25/100. Anatomical

Accuracy, Aesthetics, and Cost were given values of 10/100, 15/100, and 10/100, respectively (Table 1)

**Table 1. Hand model decision matrix.**

Criteria		Design Alternatives		
Category	Weight	3D Printing	Alginate/ Ballistics Gel	Silicone
Resistance / Haptics	40	30	28	35
Anatomical Accuracy	10	10	6	8
Durability	25	23	5	23
Aesthetics	15	12	8	14
Cost	10	3	10	8
<b>Total</b>	<b>100</b>	<b>78</b>	<b>57</b>	<b>88</b>

The three design alternatives received fairly similar scores in the categories of Resistance/Haptics and Anatomical Accuracy. All material choices had texture, density, and viscosity similar to human tissue. However, silicone scored the highest in Resistance/Haptics because combining silicone grades allowed more variability in resistance than the other designs. 3D printing received the highest rating for anatomical accuracy because the prototype would be based off of a CT scan containing all of the anatomical structures.

Two categories that showed the widest range in scores were Durability and Cost. The 3D printing and silicone models were equal in durability, but the ballistics gel can only last a few days, which does not meet the client specifications. This resulted in a low Durability score of 3/25 for the ballistics gel. However, the ballistics gel requires minimal cost to produce, resulting in a high score in the Cost category. The 3D printing, due to the complexity of the model, is very expensive, and therefore received a low score of 3/10 for Cost.

Overall, the silicone hand model fulfilled the client's requirements, receiving close to the maximum score in each category. The final ratings for the 3D printing, ballistics gel, and silicone designs were 78, 57, and 88, respectively. Because the silicone model received the highest rating, this is the design the team decided to pursue.

### **Design Ideas for Tracking Device**

The purpose of the tracking device is to interface the physical hand model with the surgical simulation software already developed by the client. In order to provide a realistic experience similar to the endoscopic carpal tunnel procedure, the user will move an endoscope shell with the corresponding blade attachment inside the hand model. As mentioned in the client specifications, the hand model will not be observed with an endoscopic camera. Instead, a software program developed by the client will display images of the carpal tunnel that are typically viewed by the camera. The tracking device must monitor the position of the endoscope shell and communicate that information with the virtual software environment.

#### *Trackball*

There were three design alternatives when considering design alternatives for the tracking device. The first design incorporated a trackball, similar to Figure 11. A trackball consists of a ball held in a socket lined with sensors to detect the rotation of the ball around two axes (x and y). As opposed to a mechanical design seen in computer mice, where a slotted wheel interrupts beams of light in order to measure rotation, trackballs have direct optical tracking that follow dots on the ball. Therefore, smaller increments of movements can be traced with a trackball than with a computer mouse, which is currently used with the client's simulation software.

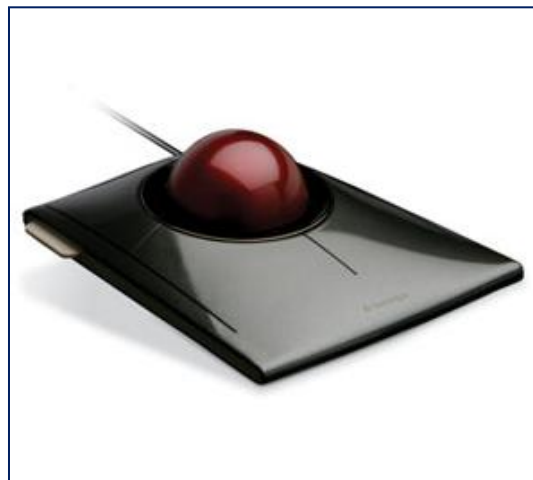


Figure 11. Typical trackball design.

In the trackball design, the ball would be positioned so that its top surface would be exposed through the bottom of the carpal tunnel tube. There would be a direct connection between the trackball and the endoscopic blade attachment (Figure 12) such that insertion and subsequent movement of the endoscope causes trackball rotation. This rotation will be communicated to the computer directly through a USB, and this will be perceived as

translational linear motion in the virtual environment. Because track balls are often used in place of a mouse, the interface between the trackball and the computer simulation system is simplified significantly. However, the trackball offers only two degrees of freedom, which limits further development of the software. Also, the interaction between the trackball and the endoscope depends on good frictional contact between the components. To guarantee such contact would require significant modification of the endoscope, the trackball, or both and would reduce the level of haptic feedback to an unacceptable level. Finally, because the trackball would expose a bottom section of the carpal tunnel, this may compromise the properties of the silicone material and not provide the correct haptic feedback to the user.

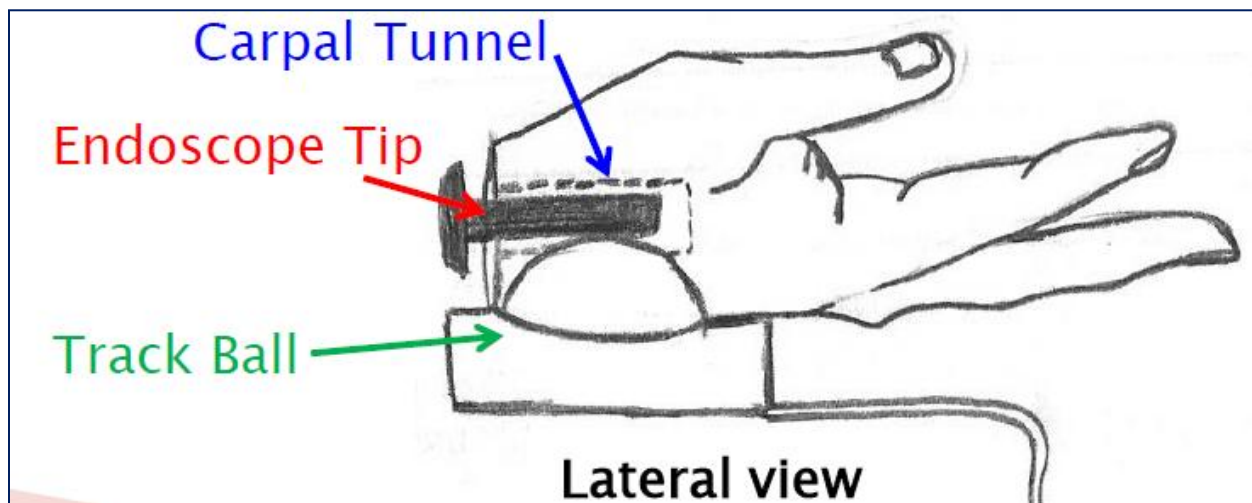


Figure 12. Lateral view of the hand model and track ball. There is a direct physical connection made between the endoscope tip and trackball, and the information is communicated to the software through a direct wire connection.

### *Accelerometer and Gyroscope*

The second tracking device alternative uses a combination of an accelerometer and a gyroscope to achieve a relative knowledge of the endoscope's position. Three-axis accelerometers have the ability to track their own linear acceleration in up to three directions. Gyroscopes operate in a similar fashion but provide information on their angular acceleration. When combined, these two devices allow for six-axis relative motion tracking. In this design, the

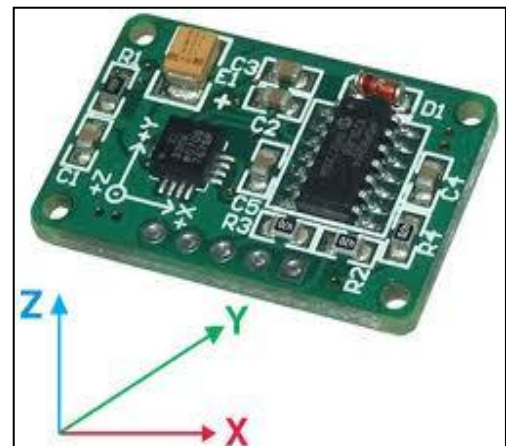


Figure 13. A typical accelerometer and gyroscope chip. <http://www.ebaheth.com/home/index2.php>

accelerometer and gyroscope would be combined on a chip, similar to Figure 13, and mounted onto the outside of the endoscope shell. The initial size of the chip was small, a typical chip being 4 x 4 x 0.9 mm [8], so it would not interfere with the surgical procedure. However, attaching accelerometers and gyroscopes to the endoscope in the correct configuration presented a challenge. Another challenge inherent to this design was in data acquisition. Accelerometers and gyroscopes produce voltage responses to acceleration. Such voltages represent analogue signals which require conversion to a digital form for processing and interfacing. Ideally, this conversion and processing step would occur within a microcontroller which would interface with the computer simulation system. However, a microcontroller would add additional wiring and bulk to the system which might affect the flow of the simulation. Finally, the accuracy and reliability of a tracking system dependent on relatively inexpensive accelerometers and gyroscopes might not satisfy the clients' requirements for precise motion tracking.

### *Wii Remote*

The third and final design alternative for the tracking device employs the built in infrared camera based tracking system of Nintendo's Wii Remote. The infrared camera in the Wii Remote detects infrared light around 940nm [9]. In addition to the camera, the remote includes a processor which resolves up to four bright spots on the camera's array into pixel coordinates. Therefore, tracking of up to four point sources of infrared light is possible. The processor transmits information through Bluetooth signals which allow relatively simple detection by most Bluetooth equipped computers [10]. However, extracting and processing the data contained in the Bluetooth signals would require more complex techniques. Motej [11] is a free Java library which provides a framework for interfacing with the Wii Remote. Infrared LEDs, which approximate point sources, would be attached to the endoscope shell outside of where the endoscope would enter the incision as to not interfere with the resistance and haptics of the carpal tunnel (Figure 14). Attaching these LEDs and applying the correct Java algorithm to the data collected through Motej should allow for at least four-axis tracking of the endoscope's position..

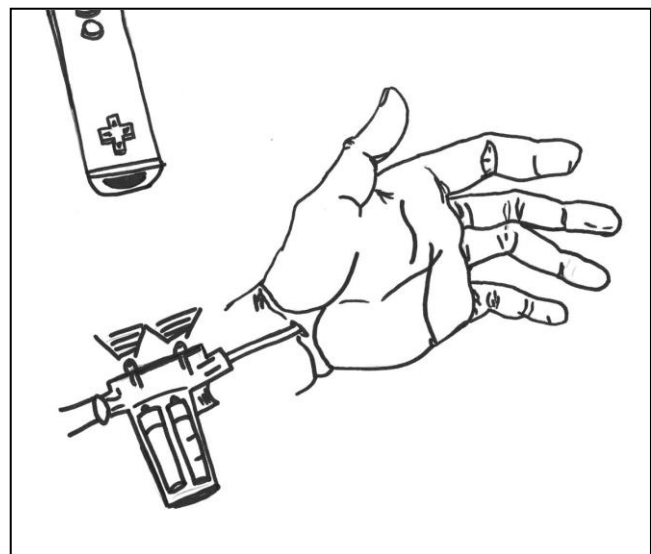


Figure 14: Sketch of the Wii Remote tracking system. The infrared LEDs will be placed on the top or back of the endoscope shell as to not interfere with the surgery, and the stationary remote will track their position in space.



## Decision Matrix – Tracking Device

A separate decision matrix was used to analyze the three tracking device design alternatives. The designs were evaluated based on five categories: Size, Signal Processing, Degrees of Freedom, Ergonomics, and Cost. The values out of 100 for the categories were 22, 25, 20, 28, and 5, respectively (Table 2).

Table 2. Tracking Device Decision Matrix

Criteria		Design Alternatives		
Category	Weight	Wii Remote	Accel/Gyro	Trackball
Size	22	20	12	14
Signal Processing	25	15	5	25
Degrees of Freedom	20	17	20	8
Ergonomics	28	25	14	20
Cost	5	3	3	5
<b>Total</b>	<b>100</b>	<b>80</b>	<b>54</b>	<b>72</b>

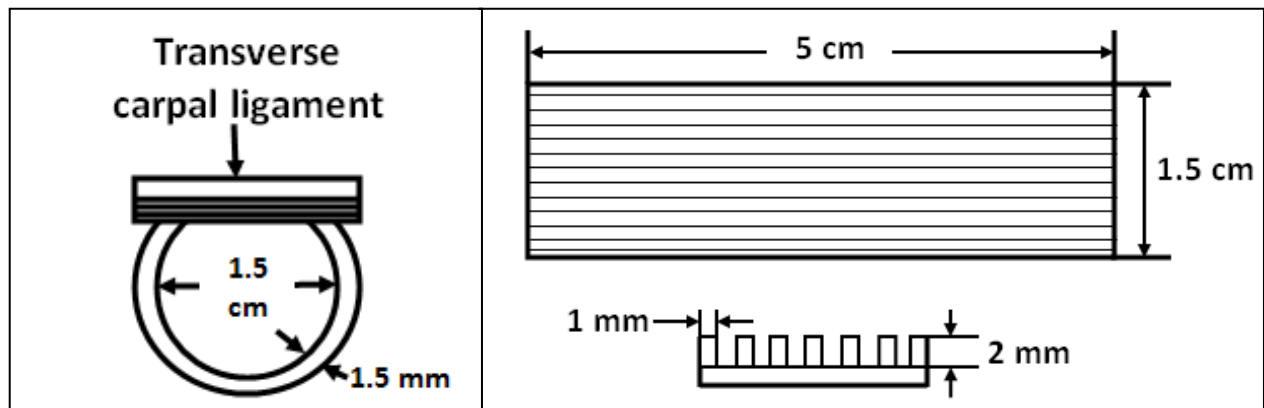
Ergonomics, in this context, refers to the system’s ability to track the position of the endoscope without disrupting the flow or feel of the simulation. This feature was the highest priority as indicated by its large weight (28/100) in the design matrix. The accelerometer and gyroscope assembly seemed to disrupt the simulation the most due to the wires and size of the endoscope attachment required, therefore, it received the lowest Ergonomics score (14/28). The Wii remote should be the most ergonomically sound because it is connected wirelessly via Bluetooth, which would not interfere with the feel of the simulation.

Signal processing was given the next largest weight (25/100) due to the time and hardware considerations it introduces. The trackball scored best in this category as it requires only a simple USB interface, while the accelerometer and gyroscope requires both a microcontroller component and a computer based software component. The next consideration

was Size (22/100), which was given the next third highest ranking because it is important that the size does not interrupt the procedure. The Wii Remote system received the best score (20/22) because the size was minimized due to the small components on the endoscope and the small and wireless nature of the Wii Remote. Degrees of Freedom, while significant, was weighted the fourth highest (20/100) due to the limited current state of the simulation software. The software current has one degree of freedom, but it would be optimal to expand upon this. The accelerometer scored the best in this category because it allows for all six degrees of freedom. The Wii remote scored second for Degrees of Freedom because it can offer up to four, while the trackball scored lowest because only two degrees of freedom are possible. Finally, cost received the lowest weight due to the flexible budget of the clients. Considering all these factors, the Wii Remote seemed to be the best choice for the tracking system and therefore was pursued.

### **Final Design**

The final prototype design interfaces a silicone hand model with a Wii remote tracking device. The team collaborated with Greg Gion of Medical Art Prosthetics, LLC. to create the hand model to the clients' specifications. The carpal tunnel tube was 5 cm in length, inner diameter at 1.5 cm, and a thickness at 1.5 mm (Figure 15). The tunnel was made of a more rigid silicone (hardness of A30) and was suspended in a surrounding tube of silicone gel. This provides the proper resistance when the endoscope is inside of the carpal tunnel. The transverse carpal ligament covers the entire top surface of the carpal tunnel tube so the corrugations can be felt as the blade attachment is pulled along the ligament in the simulation. The shape of the corrugations can be seen on the bottom of Figure 16. The rest of the hand (fingers and thumb), which is primarily for aesthetics, is filled with an expanding silicone foam, and the outer surface of the hand is cover in a skin-like layer of silicone.



Figures 15 (left) and 16 (right). On the left are the dimensions of the carpal tunnel tube cross section. On the right gives dimensions of the transverse carpal ligament. Above is a view from the top of the ligament, below is a cross section showing dimensions of the corrugations.

The final tracking device consists of a printed circuit board which powers three infrared LEDs and attaches to the back of the endoscope shell with a rubber cap (Figure 17), a Wii Remote and a Java program (see Appendix B) based on the Motej framework. The circuits on the board are powered by a 6 volt, 300 milliamp AC to DC adapter. This adapter supplies power to two circuits, one which consists of two series LEDs in series with a 330 ohm resistor and one which consists of a single LED in series with a 330 ohm resistor and controlled by a simple on-off switch. On the board, the LEDs are positioned in line with one another and separated by a distance of 2 inches (Figure 18 on page 20). A rubber stopper attached to the board allows for the coupling of the board with the endoscope shell. The Wii Remote is positioned behind the endoscope working area in full view of the LEDs. When the program is run and the LEDs are powered, the distance between the Wii Remote and the center LED is continuously calculated in the X, Y and Z dimensions based on trigonometry. With this data and data from future versions of the program such as rotational data, interface between the surgery simulation software and the tracking device will be possible.



Figure 17. The circuit board is attached to the back of the endoscope shell by a rubber cap. The Wii remote views the LEDs from behind the endoscope working view.

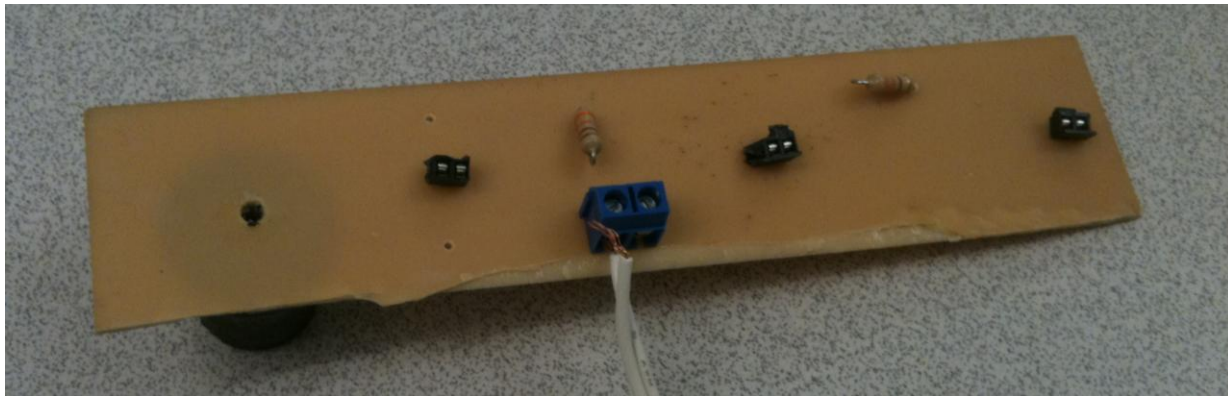


Figure 18. The circuit board for the LEDs, where in blue is the AC to DC adapter, in black are the locations of the LEDs, and in brown are the 330 ohm resistors. The wire from the AC to DC adapter leads to the power source.

In the final design, a signal needs to be sent to the simulator program to show blade and cut tendon when the trigger is pulled. To accomplish this, the final design includes a mechanical switch on the inside of the endoscope shell that functions as an on/off switch for the middle LED on the circuit board (Figure 19). When the trigger is not pulled, the switch is compressed by the trigger and remains in the “on” position, lighting the middle LED. With the developed Java program, the presence of 3 LED lights signals the Wii remote to track the position of the three lights. When the remote is following three LEDs, this will tell the simulator software to show pictures of the carpal without the blade. When the endoscope reaches the end of the carpal tunnel, the cutting sequence is ready to begin. When the trigger is pulled, the switch is no longer compressed and goes to the “off” position, turning off the third LED. With the middle light off, the tracking system will then only track the position of two LED lights. This will signal the simulator software to begin showing pictures of the blade and the cut transverse carpal ligament as the endoscope is pulled out of the tunnel.



Figure 19. Mechanical limit switch to serve as on/off function for third LED.

### *Ergonomic Considerations*

In order to replicate a surgical procedure as accurately as possible, the ergonomics in the simulation become an important factor in this project. For the simulation, the program must provide the user with a simple operating program while still providing the details necessary to learn the complex surgery properly. Therefore, the entire simulation must be intuitive so that new users are able to learn the surgical technique without extensive difficulty. The system must also be intuitive and provide user instructions so that the simulations will be operated with the same method for each trial. This way, the simulation will give reliable and repeatable results on a consistent basis to provide feedback about improvement. Multiple surgeons will potentially use this prototype for repeated simulations, and the program must remain constant to compare different surgeons on a fair scale.

## *Fabrication*

The printed circuit board was fabricated using a toner transfer technique. This technique involves designing the circuit board on a computer (Figure 20) and printing the design on glossy photo paper using a toner based printer. The resulting printout is then placed onto a copper clad plastic board and high heat, generally from a household iron, is applied for around 15 minutes with high pressure. Under these conditions, the toner melts off of the glossy paper and onto the copper clad board. To ensure the proper level of coating, the tracks of the design are then drawn over using a marker with high plastic content ink. Once the ink has dried, the board is cut to size with tin snips and submersed in a ferric chloride solution. This solution dissolves copper off of the copper clad board but does not dissolve plastic. Therefore, the areas of the board covered by the high plastic content toner and ink are not dissolved away. The board must remain in the regularly agitated solution long enough for all excess copper to dissolve but not long enough for the protected tracks to be dissolved from below the toner and ink. The dissolving process may take up to an hour but is expedited by heat and agitation. The completed product is then drilled to make room for the components' leads which are soldered into place to complete fabrication. A 3/16 of an inch hole was also drilled in the circuit board and rubber stopper. This allowed the switch wires to go from the inside of the endoscope to the circuit board on the outside.

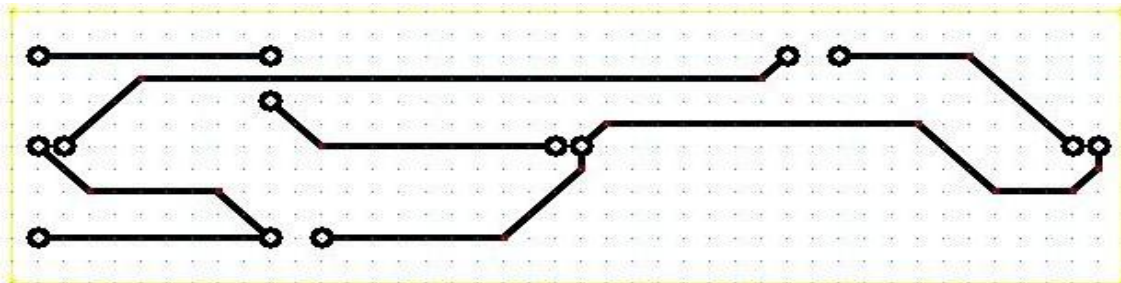


Figure 20. Diagram of circuit board. Ports for LEDs run along the center. LEDs on far right and left are in series with 330 ohm resistor, and the center LED is in series with another 330 ohm resistor on top left.

## **Testing and Analysis**

Testing for this prototype involved determining the optimal distance between the Wii remote and the infrared LEDs attached to the endoscope shell. By determining the distance between the Wii remote and infrared LEDs that result in the lowest percent error, the accuracy of the prototype is optimized.

The testing procedure is as follows:

1. The starting distances between the Wii remote and the infrared LEDs were chosen at 20, 30, 40, 50, and 60 cm.
2. At each starting distance, the position of the Wii remote remained constant.
3. The circuit board containing the LEDs was moved away from the Wii remote a total of 5cm in 1cm increments. 5cm was chosen because this is the approximate length of the carpal tunnel.
4. At each increment, distances were recorded by the processing software and compared to the set experimental distances, which were measured with two fixed metric rulers.
5. Three repetitions were executed for each starting distance data set.
6. The average deviation and percent errors were calculated for each data set.

An optimal distance between the Wiimote and the LEDs was expected to exist and become evident through trends in the testing data. Upon testing the system at initial distances ranging from 20 cm to 60 cm, a span that reflects possible simulator set-ups, no such trend was found. Average interval distances ranged from 0.97 cm to 1.02 cm, however, there was no consistent increase or decrease in the data that related to the specific starting position (see Table 3 and Figure 21). These results indicate that there is not a significant difference in the accuracy of the system over the tested range of initial positions.



Figure 21: Testing set-up. The Wii remote is aligned with 0 on the metric ruler, and the circuit board with LEDs was moved along in 1 cm intervals.

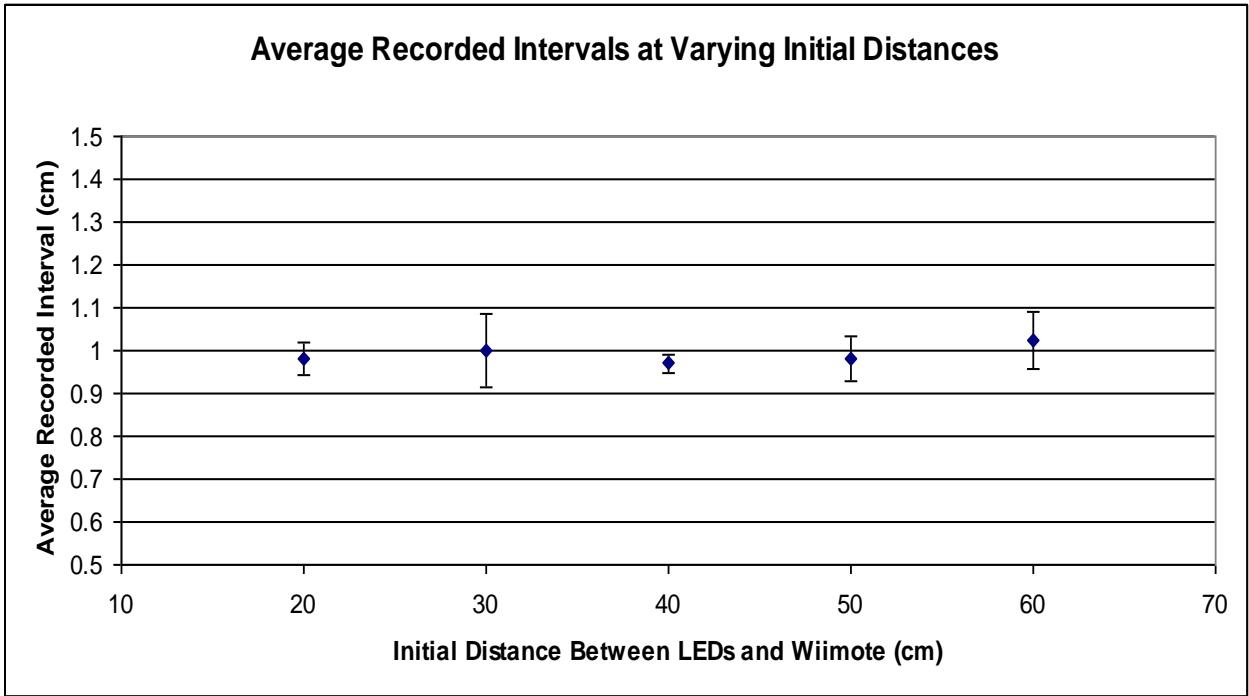


Figure 22. Graph of the average interval distances recorded by the Wii remote at initial starting distances of 20, 30, 40, 50, and 60 cm. All averages are centered on 1 cm because that was the experimental increment chosen. Error bars represent the standard deviations between the trials.

Table 3. Average increment values (cm), St. Dev., % Error for starting distances 20-60cm

	Starting Distance (cm)				
	20	30	40	50	60
Trial 1	0.940564	0.951648	0.992176	0.93632	0.97226
Trial 2	0.98424	0.95212	0.97174	1.03956	0.99682
Trial 3	1.01464	1.09954	0.94906	0.96964	1.10148
Average	0.979815	1.001103	0.970992	0.98184	1.02352
St. Dev.	0.037236	0.08525	0.021568	0.05269	0.068623
% Error	2.018533	0.110267	2.9008	1.816	2.352

To optimize this testing procedure, a better calibrated positioning system would reduce human testing error. One possibility would be using a digital caliper system to set and record distances instead of a ruler. Another method would be to utilize a linear motor moving at a constant velocity. In this procedure, the circuit board containing the infrared LEDs would be fixed to the motor, and the information recorded would be distance and time. From this, the velocity could be calculated and compared to the set velocity of the motor. Measuring velocity would allow for a greater sampling frequency, resulting in more accurate percent error and standard deviation measurements.

### **Future Work**

For the next semester, the project can be improved in several ways. The first step is to integrate the current java program into the surgical virtual environment made by the client. The actual simulator program uses a series of pictures that change with the forward and back arrows. The current java program for the tracking system tracks the distance in the z direction in centimeters. Coordinating the distance in the z direction with the different pictures in the simulator program is a task for the spring semester.

In addition to integrating the tracking program with the surgical software, other mechanical parts of the endoscope could be improved. During the actual surgery, the surgeon feels a slight resistance when cutting the ligament with the blade. Replicating this with a more sophisticated force feedback system is also a task for future semesters. A few preliminary ideas have been discussed.

The first idea involves placing two rollers at the beginning of the carpal tunnel. The rollers would be free to rotate in one direction, but they would provide resistance when rotating in the opposite direction. The disposable blade attachment slides between the rollers when it enters carpal tunnel, and the rollers are able to spin freely. Therefore, the user does not feel any resistance while the endoscope moves forward in the carpal tunnel. However, when the blade moves backwards and cuts the ligament, the rollers are not able to spin freely. They instead provide frictional resistance to the endoscope attachment, similar to what is felt by the surgeon when the ligament is cut. The resistance is only felt in one direction and would supply the surgeon with adequate force feedback during the simulation.

The second idea uses the trigger to deploy a piece of rubber to provide resistance. When the endoscope is used with the physical hand model, the actual blade is removed and would be replaced by another material for the simulation. No modifications to the current carpal tunnel would be required. Instead of the blade, a piece of rubber could be deployed when the trigger is pulled. The rubber would push up on the carpal tunnel similar to what the blade does during surgery and would provide resistance when pulling back. Depending on the size of the material, the amount of resistance required could be adjusted according to client and surgeon feedback.



The third idea uses an external clamp around the carpal tunnel to provide resistance. As the endoscope enters the carpal tunnel, the clamp does not apply any pressure. When the trigger is pulled to simulate the blade being deployed, the clamp would then tighten and constrict the diameter of carpal tunnel. Because of this, the surgeon would then feel an increased resistance when cutting the ligament. However, with this idea, the hand model would need to be modified to accommodate the external clamping mechanism.

Improvements on the hand model can also be made. The most important aspect of the model is the resistance and dimensions of the carpal tunnel. The current carpal tunnel will be tested by the clients and can be updated based on their feedback. The overall diameter of the carpal tunnel can simply be adjusted by Greg Gion and corrugations on the ligament can also be adjusted in the future to the clients' needs.

Once all the necessary adjustments are made and the entire simulation program is working properly, a series of tests need to be completed. The program needs to first be tested for accuracy and precision during the simulation. The endoscope and blade need to be accurate according to the clients' specifications in the PDS. Also, an evaluation will be conducted to collect feedback from the clients and multiple surgeons. This will drive the direction for improving the overall program and completion of the final design.

### **Acknowledgements**

The team would like to thank our advisor, Professor Willis Tompkins, Professor Tom Yen, Greg Gion from Medical Art Prosthetics, LLC., and our clients, Dr. Robert Radwin and Dr. Ben Mandel for their support of this project.

### **References**

- [1] Haque, Mustafa. "Carpal Tunnel Syndrome." *womenshealth.gov*. N.p., 1 Jun 2009. Web. 4 Dec 2010. <<http://www.womenshealth.gov/faq/carpal-tunnel-syndrome.cfm>>.
- [2] Zheng, Y., Li, Z., Chen, X., Lu, M., Choi, A., et al. (2006). Ultrasound palpation sensor for tissue thickness and elasticity measurement-assessment of the transverse carpal ligament. *Ultrasonics*, 44.
- [3] (2007). Carpal tunnel syndrome. National University Hospital. Retrieved 2 December 2010 from <http://www.handmicrosurgery.org/services/carpaltunnelsyndrome/>.
- [4] Chmarra, M. , Bakker, N. , Grimbergen, C. , & Dankelman, J. (2006). Trendero, a device for

- tracking minimally invasive surgical instruments in training setups. *Sensors and Actuators A-physical*, 126(2), 328-334
- [5] (2010) Proto Labs. Retrieved 10 October 2010 from <http://www.protolabs.com/>.
- [6] Beck, E. (Aug 20, 2007). Build plans: how to make a fake hand. Backyard FX. Message posted to <http://www.indymogul.com/backyardfx/post/1972/build-plans-how-to-make-a-fake-hand>.
- [7] (December 2010). Silicone. Retrieved 30 November 2010 from <http://en.wikipedia.org/wiki/Silicone>.
- [8] Johnson, R. (2010). MEMS gyro/accelerometer combo chip debuts. EE Times. Retrieved 2 December 2010 from <http://www.eetimes.com/electronics-news/4210545/MEMS-gyro-accelerometer-combo-chip-debuts>.
- [9] (May 15, 2008). Re: What is the Wiimote's Preferred Peak Wavelength?. Message posted to <http://www.wiimoteproject.com>.
- [10] Lee, J. (2008). Hacking the Nintendo Wii remote. *Pervasive Computing, IEEE*, 7(3), 39-45
- [11] Volker Fritzsch (April 7, 2009). Motej. Retrieved October 12, 2010, from <http://motej.sourceforge.net/>.

## Appendix A

### **Product Design Specifications for BME 400 Group 47: Surgical Simulator**

**December 4th, 2010**

Group Members: Charles Donaldson, Mason Jellings, Sarah Switalski, Whitney Johnson

#### Problem Statement:

The methods to practice surgical procedures have begun to change in the medical field. Instead of using expensive cadavers to practice, many surgeons are starting to practice on simulators. Our objective this semester is to design a simulator for endoscopic carpal tunnel surgery by using a life-like physical hand model to interface with a virtual 3-D representation of the wrist and hand.

#### 1. Design Requirements:

The device must meet all of the client requirements

- a. Performance requirements: Hand model should feel as close to a regular hand as possible. Signaling device should potentially be able to detect motion within all six degrees of freedom without detectable delay.
- b. Safety: Device should not cause any harm to the surgeon or other connected devices involved with the simulator
- c. Accuracy and Reliability: Signaling device should be able to detect position of the endoscope within 1 mm. Device should maintain accuracy of position throughout single and multiple simulations.
- d. Life in Service: Simulator should be able to withstand repeated uses by surgeons practicing the surgery for at least a year.

- e. **Operating Environment:** Device will be used in a hospital setting at room temperature and standard humidity.
- f. **Ergonomics:** The hand model should look and feel like a real hand. The hardware should be arranged in a way that does not get in the way of the surgery. The method of signaling should not change the physical components of the endoscope.
- g. **Size:** Hand model should be life size and the simulated incision should be anatomically similar to what is seen during surgery.
- h. **Materials:** Hand material feel, look and act like regular tissue. Material should also have similar mechanics involving structure that is similar to tissue of the hand.

## 2. Production Characteristics:

- a. **Quantity:** One reproducible working prototype is necessary.
- b. **Target Product Cost:** \$400

## 3. Miscellaneous:

- a. **Standards and Specifications:** No specific standards because prototype is only used in simulation, not actual surgery
- b. **Customer:** Client is currently working on creating the virtual environment. Client also specified the need for a signaling device that will track position of endoscope accurately and precisely. The feel of the hand model should also be similar to a normal hand so that practicing surgeons can get used to the feel of the surgery. The only anatomy specified by the client is the restrictions of the carpal tunnel.
- c. **Patient-related Concerns:** None
- d. **Competition:** A current device involving minimally invasive surgeries called TrEndo. It creates a physical connection between the tracking element and the surgical device.

## **Appendix B**

Java Code:

```
import motej.Mote;
import motej.request.ReportModeRequest;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

//import java.awt.*;
//import org.javablueetooth.stack.*;
//import org.javablueetooth.stack.hci.*;
//import javax.bluetooth.*;

public class Wiiread {

    private static Logger log = LoggerFactory.getLogger(Wiiread.class);

    private static Object loc = new Object();

    //private static BluetoothStack bluetooth;

    Wiiread(){
        //bluetooth = BluetoothStack.getBluetoothStack();
    }

    public static void main(String[] args) throws Exception {
        //HCIDriver.init(new UARTTransport("COM3"));
        //BluetoothStack.init(new BluetoothStackLocal());
        //Wiiread blue = new Wiiread();

        FindPixDist dist = new FindPixDist();
        buttonStop stop = new buttonStop(loc);
        RippedMoteFinder find = new RippedMoteFinder();

        Mote myMote = find.findMote();

        myMote.addIrCameraListener(dist);
        myMote.enableIrCamera();
        myMote.setReportMode(ReportModeRequest.DATA_REPORT_0x36);
        myMote.addCoreButtonListener(stop);

        try{
            synchronized(loc){
                loc.wait();
            }
        }
    }
}
```

```

        }catch(InterruptedException ex){
            log.error(ex.getMessage(), ex);
        }

myMote.disableIrCamera();
myMote.disconnect();

        dist.writeData("testingDists.txt");
        System.out.println("done");

```

---

```

import motej.event.CoreButtonEvent;
import motej.event.CoreButtonListener;

public class buttonStop implements CoreButtonListener{

    private Object ob;

    public buttonStop(Object o){
        ob = o;
    }
    public void buttonPressed(CoreButtonEvent evt){
        if(evt.isButtonAPressed()){
            stop();
        }
    }
    private void stop(){
        synchronized(ob){
            ob.notifyAll();
        }
    }

```

---

```

import motej.IrPoint;
import motej.event.IrCameraEvent;
import motej.event.IrCameraListener;

import java.util.*;
import java.io.*;

public class FindPixDist implements IrCameraListener{
    private List<distEvent> al;
    private double Fl;
    private double d_o;

    public FindPixDist(){
        al = new ArrayList<distEvent>();
        Fl = 1343.657435;
        d_o = 101d;
    }

    public void irImageChanged(IrCameraEvent evt){
        IrPoint p0 = null;
        IrPoint p1 = null;
        IrPoint p2 = null;

```

```

if(evt.getIrPoint(0).x != 1023){
    p0 = evt.getIrPoint(0);
}

if(evt.getIrPoint(1).x != 1023){
    if(p0 == null){
        p0 = evt.getIrPoint(1);
    }
    else{
        p1 = evt.getIrPoint(1);
    }
}

if(evt.getIrPoint(2).x != 1023){
    if(p0 == null){
        p0 = evt.getIrPoint(2);
    }
    else if(p1 == null){
        p1 = evt.getIrPoint(2);
    }
    else{
        p2 = evt.getIrPoint(2);
    }
}

if(evt.getIrPoint(3).x != 1023){
    if(p0 == null){
        p0 = evt.getIrPoint(3);
    }
    else if(p1 == null){
        p1 = evt.getIrPoint(3);
    }
    else if(p2 == null){
        p2 = evt.getIrPoint(3);
    }
}

if(p0 == null || p2 == null){
    return;
}

double d_i = p0.distance(p2);

//double T = Math.atan2(Math.sqrt(1d/(2d*Math.pow(Fl, 2d) + 2d*Math.pow(d_i, 2d)))*(Fl - d_i),
Math.sqrt(1d/(2d*Math.pow(Fl, 2d) + 2d*Math.pow(d_i, 2d)))*(d_i + Fl));

//double z = .5*Math.sqrt(1d/(2d*Math.pow(Fl, 2d) + 2d*Math.pow(d_i, 2d)))*(d_i*Fl +
Math.pow(d_i, 2d) + 2d*Math.pow(Fl, 2d))*d_o/d_i;
int ex0 = p0.x;
int ex1 = p2.x;
int wy0 = p0.y;

```

```

int wy1 = p2.y;

double z = (F1*d_o/d_i)/10d;
double x = z*((double)ex0 + (double)ex1)/(2d*F1)/10d;
double y = z*((double)wy0 + (double)wy1)/(2d*F1)/10d;

System.out.println("z = " + z);
System.out.println("x = " + x);
System.out.println("y = " + y);
//System.out.println("T = " + T);
//double doo = p0.distance(p1);

    System.out.println("di = " + d_i);
    System.out.println("x0 = " + ex0);
System.out.println("y0 = " + wy0);
    System.out.println("x1 = " + ex1);
System.out.println("y1 = " + wy1);

    makeList(z, ex0, wy0, ex1, wy1, x, y);
}

private void makeList(double d, int x0, int y0, int x1, int y1, double tee, double pix){
    distEvent event = new distEvent(d, x0, y0, x1, y1, tee, pix);
    al.add(event);
}

public double aveListDist(){
    double tot = 0d;
    double size = 0d;
    Iterator<distEvent> iterD = al.iterator();
    while(iterD.hasNext()){
        distEvent tmp = iterD.next();
        double tmpDist = tmp.getDist();
        tot = tot + tmpDist;
        size = size + 1d;
    }
    double ave = tot/size;
    return ave;
}

public void writeData(String filez){
    File theFile = new File(filez);
    try{
        PrintStream ps = new PrintStream(theFile);
        Iterator<distEvent> iterD = al.iterator();
        while(iterD.hasNext()){
            distEvent tmp = iterD.next();
            double tmpDist = tmp.getDist();
            int tmpX0 = tmp.getX0();
            int tmpY0 = tmp.getY0();
            int tmpX1 = tmp.getX1();

```



```

        int tmpY1 = tmp.getY1();
        double tmpT = tmp.getT();
        double tmpD_o = tmp.getD_o();
        ps.println(tmpDist + " " + tmpX0 + " " + tmpY0 + " " + tmpX1 + " " + tmpY1 +
" " + tmpT + " " + tmpD_o);
        ps.flush();
    }
    double theAve = aveListDist();
    ps.println("Average Distance: " + theAve);
    ps.close();
} catch(FileNotFoundException e){
    System.out.println("no file");
}

```

---

```

public class distEvent{

    private double dist;
    private int X0;
    private int X1;

    private int Y0;
    private int Y1;
    private double te;
    private double d_o;

    public distEvent(double d, int z, int o, int p, int q, double a, double da){
        dist = d;
        X0 = z;
        Y0 = o;
        X1 = p;
        Y1 = q;
        te = a;
        d_o = da;
    }
    public double getDist(){
        return dist;
    }
    public double getT(){
        return te;
    }
    public int getX0(){
        return X0;
    }
    public int getX1(){
        return X1;
    }
    public int getY0(){
        return Y0;
    }
    public int getY1(){
        return Y1;
    }
    public double getD_o(){

```

```

        return d_o;
    }
    public int[] getXs(){
        int[] Xs = new int[2];
        Xs[0] = X0;
        Xs[1] = X1;
        return Xs;
    }
    public int[] getYs(){
        int[] Ys = new int[2];
        Ys[0] = Y0;
        Ys[1] = Y1;
        return Ys;
    }
    public int[] get0(){
        int[] zer = new int[2];
        zer[0] = X0;
        zer[1] = Y0;
        return zer;
    }
    public int[] get1(){
        int[] one = new int[2];
        one[0] = X1;
        one[1] = Y1;
        return one;
    }
    public void setDist(double newD){
        dist = newD;
    }
    public void setX0(int newX0){
        X0 = newX0;
    }
    public void setX1(int newX1){
        X1 = newX1;
    }
    public void setY0(int newY0){
        Y0 = newY0;
    }
    public void setY1(int newY1){
        Y1 = newY1;
    }
    public void setT(int newT){
        te = newT;
    }
    public void setD_o(double newD_o){
        d_o = newD_o;
    }
}
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import motej.Mote;
import motej.MoteFinder;

```

---

```

import motej.MoteFinderListener;

public class RippedMoteFinder implements MoteFinderListener{

    private Logger log = LoggerFactory.getLogger(RippedMoteFinder.class);
    private MoteFinder finder;
    private Object lock = new Object();
    private Mote mote;

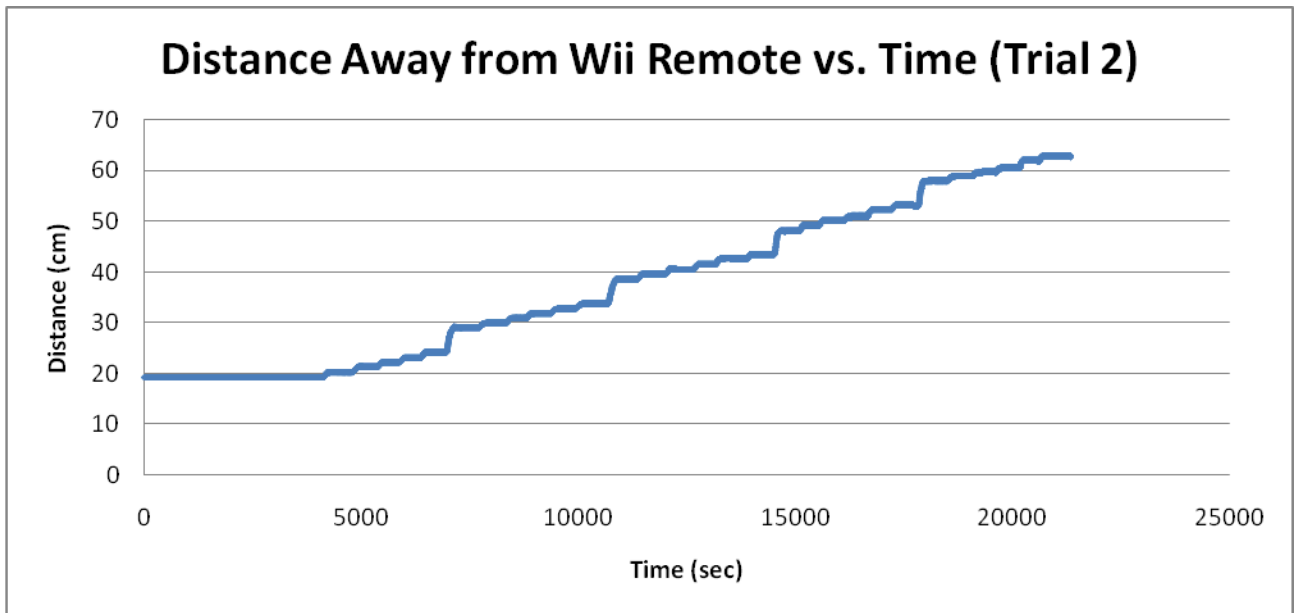
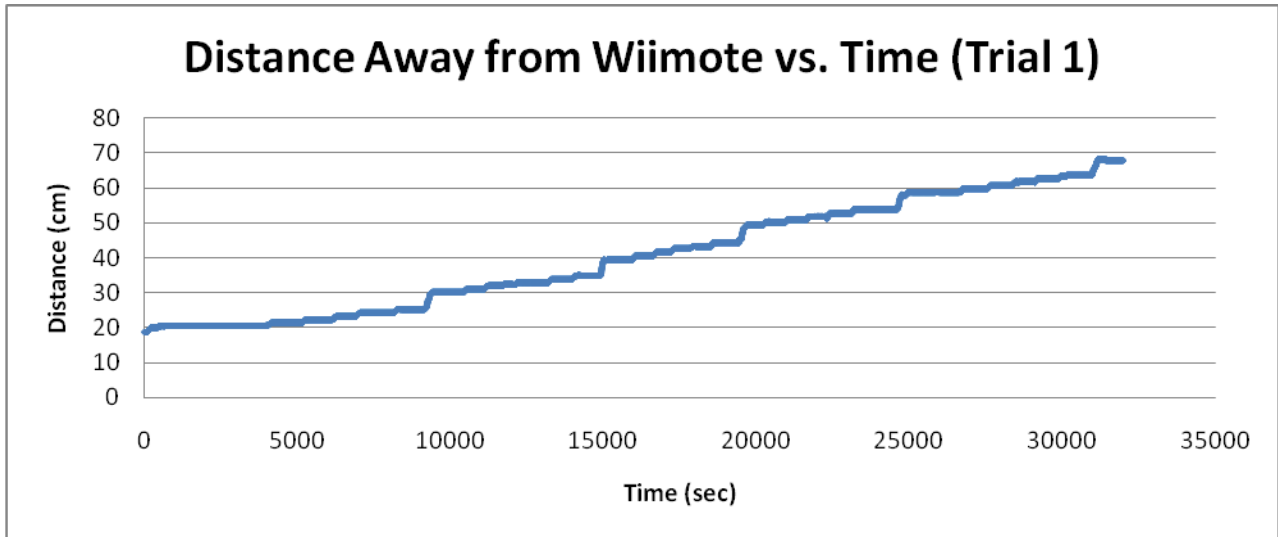
    public RippedMoteFinder() {
    }
    public void moteFound(Mote mote){
        log.info("RippedMoteFinder received notification of a found mote.");
        this.mote = mote;
        synchronized(lock){
            lock.notifyAll();
        }
    }
    public Mote findMote(){
        if(finder == null){
            finder = MoteFinder.getMoteFinder();
            finder.addMoteFinderListener(this);
        }
        finder.startDiscovery();
        try{
            synchronized(lock){
                lock.wait();
            }
        }catch(InterruptedException ex){
            log.error(ex.getMessage(), ex);
        }
        return mote;
    }
}

```

## Appendix C

### Test Data

The figures below show the distance interval data for initial distances of 20, 30, 40, 50, and 60 cm. The distance values are displayed in real time.



### Distance Away From Wii Remote vs. Time (Trial 3)

