

Simulator for Endoscopic Carpal Tunnel Release Surgery

May 4th, 2011

Team:

Whitney Johnson - Communicator

Sarah Switalski - BWIG

Mason Jellings - BSAC

Clients:

Dr. Benjamin Mandel, MD

Integrated Plastics and Reconstructive Surgery Residency - University of Wisconsin

Dr. Robert Radwin

Departments of Biomedical and Industrial and Systems Engineering

Advisor:

Professor Thomas Yen

Departments of Biomedical and Industrial and Systems Engineering

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 visual software program for endoscopic carpal tunnel surgery. Based on specifications from the client, the final prototype includes a haptic element, which consists of a hand model and blade cap mechanism. There is also a visual element, which consists of a LED circuit tracked by the Wii infrared receiver and a positioning program that is synced with real endoscopic images provided by the client. Testing was performed to determine the optimal distance between the Wii remote and the LEDs and the horizontal distance at which the LEDs leave the view of the remote. Results indicated that no significant difference in accuracy exists between the tested distance parameters of 20-60 cm and that horizontal viewing range increases linearly with distance. These results influenced the dimensions of the final prototype. Future work includes gathering feedback on the blade cap mechanism from a panel of surgeons in order to further improve the realistic surgical experience.

Table of Contents

Background	4
<i>Motivation</i>	4
<i>Surgical Procedure</i>	4
Problem Statement.....	6
Client Specifications.....	6
Current Devices.....	7
Fall 2010 Progress.....	9
Spring 2011 Progress.....	10
Design Alternatives.....	10
<i>Blade Cap</i>	11
<i>Solenoid Activated</i>	11
<i>Motor Activated</i>	12
Decision Matrix	12
Final Design.....	13
<i>Blade Cap</i>	13
<i>Hand Model</i>	14
<i>Circuit Board</i>	14
<i>Positioning System</i>	14
<i>Platform</i>	15
Ergonomic Considerations.....	15
Fabrication.....	15
<i>Circuit Board</i>	15
<i>Blade Cap</i>	16
Testing and Analysis.....	16
<i>Testing Results</i>	18
Future Work.....	20
Acknowledgements.....	21
References.....	21
Appendix A: Product Design Specifications.....	22
Appendix B: Java Code.....	24

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 minimally-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

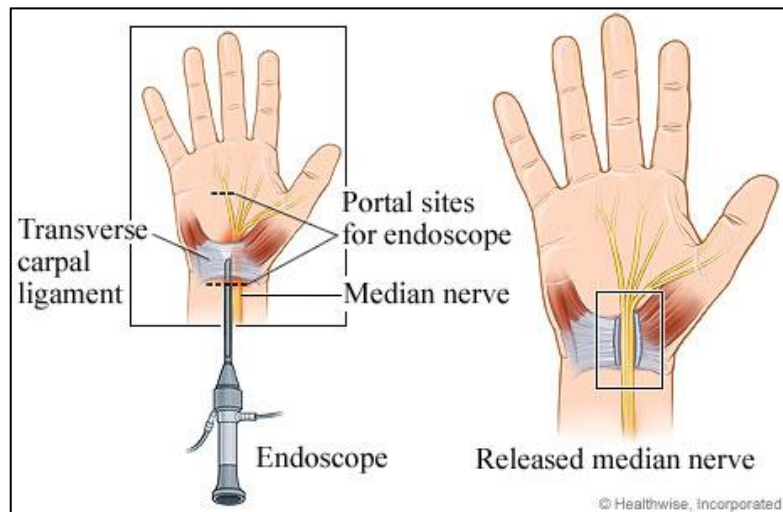


Figure 1: Placement of endoscope and cutting of transverse carpal ligament. (<http://www.health.com/health/library/mdp/0,,zm2464,00.html>)

time, the surgeon is able to feel the corrugations on the underside of the ligament, 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 ligament 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 ligament when the blade is cutting it. This tells the surgeon that the blade is actually cutting the ligament, 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

All surgical procedures require practice and repetition in order to achieve optimal outcomes. Currently, carpal tunnel syndrome release surgery is only performed on cadavers or live patients. Therefore, the cost of practicing this procedure is quite high. To reduce this cost and improve surgical performance, an ideal solution is an anatomically accurate simulator with realistic haptics. The team's objective was to design a simulator that will be used as a training device for endoscopic carpal tunnel surgery by interfacing a life-like hand model with a visual 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 length – 3.5 cm
2. Ligament corrugations – ~1 mm height by 1 mm width
3. Carpal tunnel diameter – 1 cm at proximal end with slight taper towards the distal end
4. Incision location – 5 mm proximal to distal wrist crease

The tracking system must interface with the array of carpal tunnel images captured 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 along the z-axis and rotation about the z-axis and have a precision of 1 mm. External devices, such as wires and receivers, should not interfere with the surgical procedure.

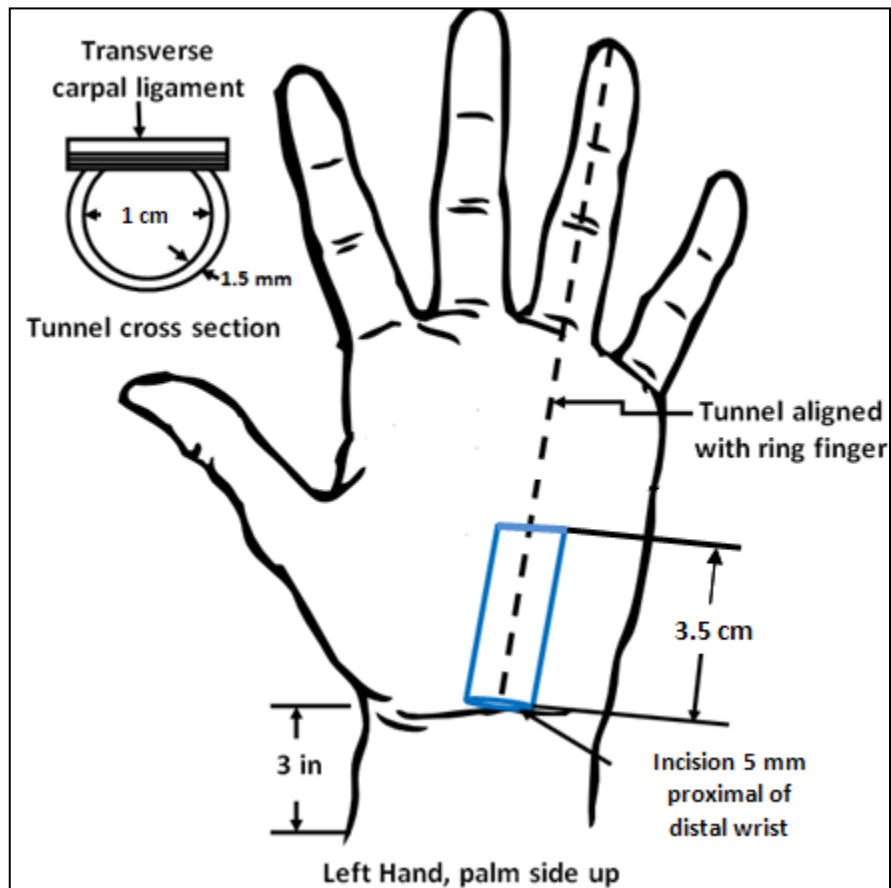


Figure 3. Schematic of carpal tunnel placement within hand model. The carpal tunnel is highlighted in blue and is aligned with the ring finger. The incision is made 5 mm proximal to the distal wrist crease, and the carpal tunnel is approximately 3.5cm in length, extending to mid-palm. The cross section 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). 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).

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 feedback with real-life images will be utilized in the design of the prototype.

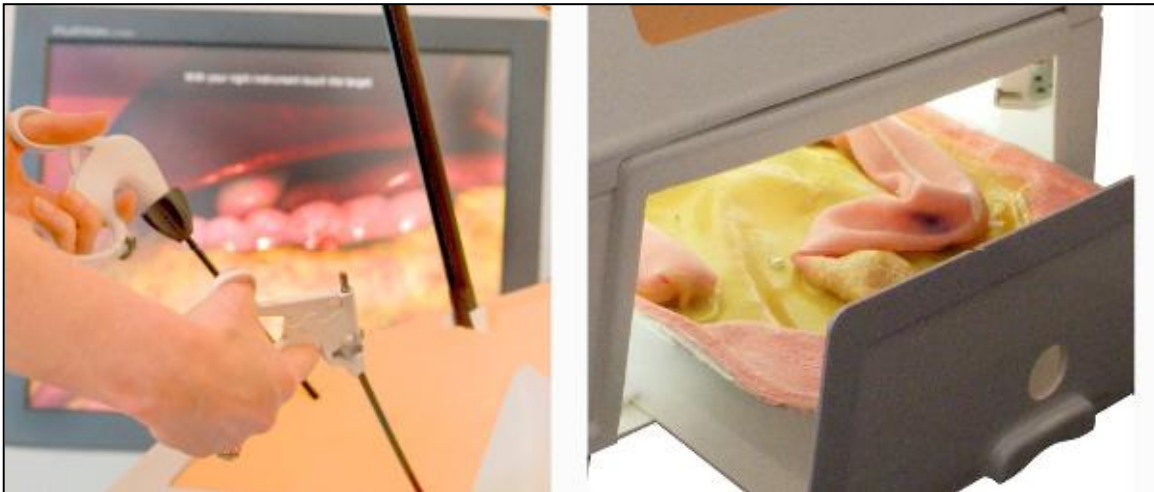
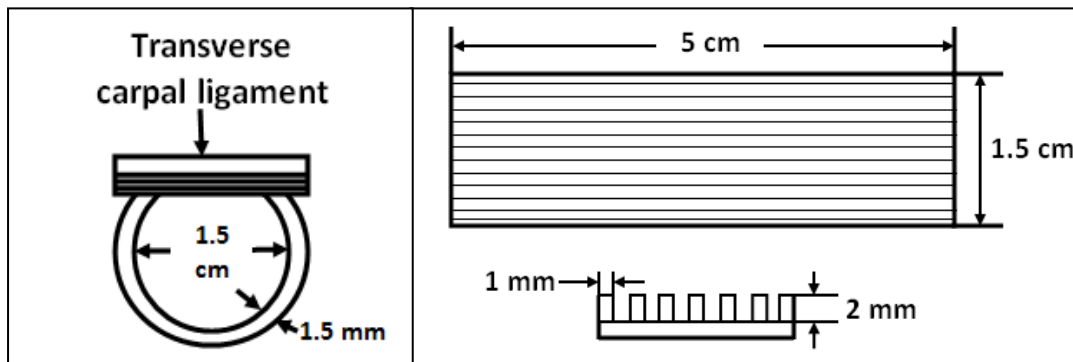


Figure 6 (left) shows the ProMIS surgical simulator utilizing virtual graphics to complete the surgery. Figure 7 (right) shows the real anatomical model included with the ProMIS system. (<http://www.haptica.com/>)

Fall 2010 Progress

Last semester's prototype design interfaced a silicone hand model with a receiver (Wii remote) to track the endoscope position in space. The carpal tunnel tube was 5 cm in length, 1.5 cm for the inner diameter, and a thickness at 1.5 mm (Figure 15). The tunnel was made of a silicone with an A30 hardness value, which was suspended in silicone gel. The transverse carpal ligament covered the entire top surface of the carpal tunnel tube so the corrugations could be felt as the blade attachment was pulled along the ligament in the simulation. The shape of the corrugations can be seen on the bottom of Figure 16. The remainder of the hand was filled with silicone gel, and the outer surface of the hand was covered in a skin-like layer of silicone.



Figures 7 (left) and 8 (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 consisted of an LED circuit board attached to the back of the endoscope shell with a rubber cap (Figure 9), a Wii Remote and a Java program (see Appendix B) based on the Motej framework. The circuits on the board were powered by a 6 volt, 300 milliamp AC to DC adapter. This adapter supplied power to two circuits, one which consisted of two LEDs in series with a 330 Ω resistor and one which consists of a single LED in series with a 330 Ω resistor and controlled by a simple on-off switch. On the board, the LEDs were positioned in line with one another and separated by a distance of 2 inches (Figure 10). A rubber stopper attached to the board allowed for the coupling of the board with the endoscope shell. When the program was run and the LEDs were powered, the distance between the Wii Remote and the center LED was continuously calculated in the X, Y and Z dimensions based on trigonometry.



Figure 9. 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.

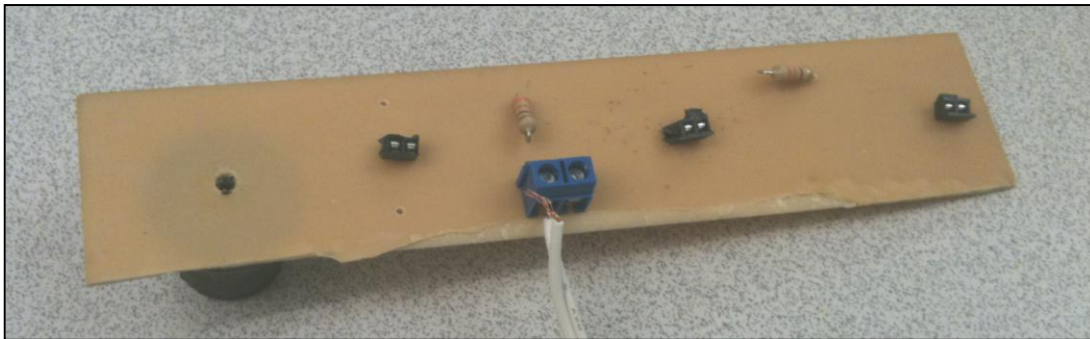


Figure 10. The circuit board for the LEDs, where in blue is the AC to DC adapter, in black are the locations of the LEDs, and in brown are the 330 ohm resistors. The wire from the AC to DC adapter

In last semester's design, a signal needed to be sent to the simulator program to show blade and cut ligament when the trigger was pulled. To accomplish this, the design included a mechanical switch on the inside of the endoscope shell that functioned as an on/off switch for the middle LED on the circuit board. When the trigger was not activated, the switch was compressed and remained in the "on" position, lighting the middle LED. With the developed Java program, the presence of 3 LED lights signaled the Wii remote to track the position of the three lights. This would tell the simulator software to show pictures of the carpal tunnel without the blade. When the trigger was pulled, the switch was no longer compressed and went to the "off" position, turning off the third LED. With the middle light off, the tracking system would then only track the position of two LEDs. This would signal the software to show pictures of the blade and the cut transverse carpal ligament as the endoscope is pulled out of the tunnel.

Spring 2011 Progress

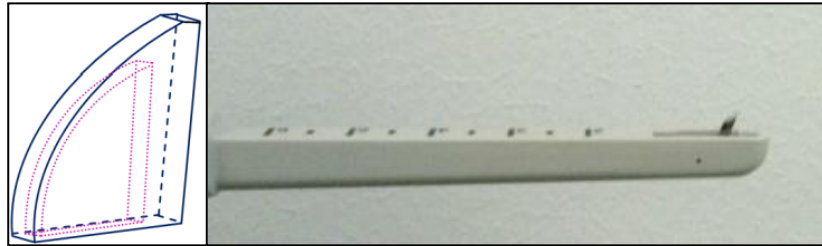
This semester's goal was to improve the prototype and progress the status of the simulation. Items that were addressed include the incorporation of digital pictures with the position tracking system, the addition of a trigger-activated force feedback mechanism, the improvement of physiological structures in the hand model, and streamlining of prototype aesthetics.

Design Alternatives – Force Feedback Mechanism

The purpose of incorporating a force feedback mechanism was to increase the resistance between the blade attachment and the silicone carpal tunnel when the blade was deployed in the simulation. The goal was to accurately replicate the amount of the force the surgeon feels when the blade is activated and the transverse carpal ligament is cut in the surgical procedure. The team devised three alternatives for a force feedback mechanism: a blade cap, a solenoid activated band, and a linear motor activated band.

Blade Cap

This design alternative consists of a hard polymer cap to cover the existing blade mechanism. The cap (Figure 11) would be fabricated with an inner pocket to slide over the current blade (Figure 12), and multiple caps would be made in a range of heights. The team then plans to test the array of caps with a panel of surgeons, and based on their feedback, the most realistic height of blade cap would be chosen. The advantages of this simple mechanical approach are that the caps will not interfere with the current trigger mechanism already in place, and they are easy to manufacture. The disadvantages are that the amount of force feedback is based on a subjective survey, and therefore, the amount of force provided is not easily quantifiable or adjustable.



Figures 11 (left) and 12 (right) show a sketch of the proposed blade cap design and the current trigger-activated blade mechanism, respectively.

Solenoid Activated Band

A second alternative for providing proper force feedback to the user of the simulation involves incorporating a semi-rigid band into the carpal tunnel structure which may be tightened or loosened to constrict or expand the space within the tunnel (Figure 13). One method of tightening such a band is the use of a solenoid. Solenoids consist of a coil of wire which produces an electromagnetic field, usually to move a metal core. Therefore, tightening of the band may be achieved through the motion of the solenoid core.

Use of a solenoid would allow for nearly instantaneous tightening of the band and thus a more accurate simulation. In addition, it would provide a pulsing capability for simulating the corrugations on the carpal ligament and might enable partial activation for graded force feedback.

However, incorporating a solenoid into the system would require a significant amount of extra programming and circuitry. Furthermore, simply tightening a band may not provide the correct feedback and will almost certainly not restrict the user's range of motion as much as actually cutting the carpal ligament.

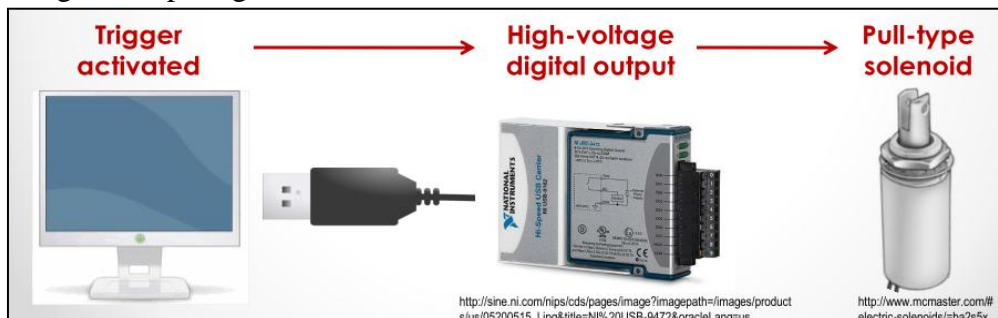


Figure 13. Schematic of the communication from trigger activation to solenoid

Motor Activated Band

The third alternative also uses the band concept to constrict the carpal tunnel structure. However, this alternative employs a motor to tighten the band (Figure 14). Although a motor would not tighten the band as rapidly as the solenoid, it might prove more reliable and would certainly allow for variation of the amount of force applied to the band. Similar to the solenoid alternative, however, the motor design would require extra circuitry and programming. In addition, the opportunity for simulation corrugations via pulsation disappears with the motor design and the integrity of the band becomes more of a factor.

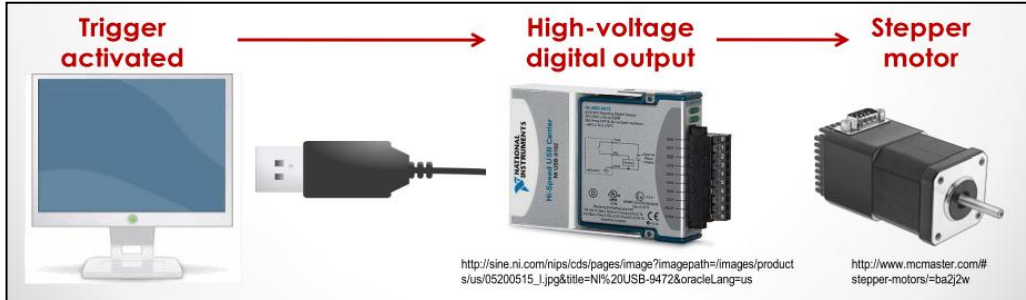


Figure 14. Schematic of the communication from trigger activation to motor activation.

Decision Matrix

The three design alternatives were rated based on the following criteria: Ease of Integration, Ease of Use, Durability, Capabilities, and Cost (Table 1). Because of its simplicity, the Blade Cap design was the easiest to integrate with the current surgical instruments and most intuitive to use. It also had a much lower potential cost than the solenoid and motor activated designs. The solenoid and motor designs would be more difficult to integrate into the current prototype because the location of the band around the carpal tunnel may compromise the structural integrity of the hand model. Even though these designs had a longer prospective life span than the blade caps, the blade cap design received the highest score of 80/100 in the matrix. This is the force feedback mechanism design that the team incorporated into the final prototype.

Table 1. Decision Matrix

Criteria	Weight	Blade Cap	Solenoid Activated	Motor Activated
Ease of Integration	30	25	18	18
Ease of Use	25	22	20	17
Durability/Life Span	30	14	17	16
Capabilities	15	10	13	8
Cost	10	9	5	6
Totals:	100	80	73	65

Final Design

The final prototype incorporates a blade cap in order to provide force feedback, an improved silicone-based hand model, an endoscope fitted with an LED circuit board, and a remote infrared receiver (Figure 15). The positioning data corresponds to a series of carpal tunnel images that provide visualization during the surgery. Finally, all the components are positioned on an acrylic platform, creating consistency throughout simulations.



Figure 15. Final prototype with the acrylic platform housing (from left to right) the hand model, the endoscope with blade attachment, the LED circuit, and the receiver.

Blade Cap

In order to increase the resistance inside the carpal tunnel when the transverse carpal ligament is “cut” in the simulation, blade caps were created to cover the current blade. Each cap was 0.040 inches in width and had a 0.010 chamfer on the curved edge to mirror the curve of the actual blade mechanism (Figure 16). Five caps were created with heights of 0.250, 0.275, 0.300, 0.325, and 0.350 inches, and each contained a pocket to fit around the current blade. The ledge in the tips of the plastic blade attachments was removed and widened to accommodate the width of the blade caps. This blade cap design did not interfere with the surgical mechanism already in place, allowing the user to obtain a realistic training experience during the simulation.

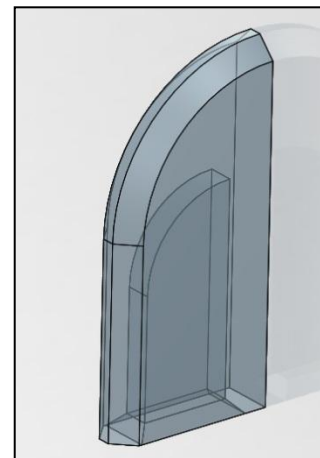


Figure 16. SolidWorks representation of the blade cap prototype.

Hand Model

After the first silicone hand model was finished, feedback from surgeons at the University of Wisconsin hospital prompted the following improvements in order to optimize physiological accuracy and haptics:

1. The transverse carpal tunnel ligament was reduced to a thickness of 5 mm.
2. The width of the ligament was shortened by approximately 1.5 cm.
3. The wrist incision was moved 5 mm in the proximal direction.
4. The incision was made slightly smaller. Circular holes were incorporated on the ends of the incision to minimize tearing due to endoscope insertion.
5. The entire tunnel was reduced to a diameter of approximately 1 cm, tapering slightly in the distal direction.

Circuit Board Endoscope Attachment

In order to increase the stability of the circuit board's fixation to the endoscope and improve the aesthetics and safety of the overall design, the entire LED assembly was redesigned. The first step was to select a suitable housing for the circuit board. This would reduce the risk of shock from exposed tracks and tidy the appearance of the endoscope attachment. The housing used in the final design consists of rigid plastic tubing and a plastic elbow joint of diameter approximately equal to the inner diameter of the tubing. To accommodate the housing, the circuit was miniaturized to a width approximately equal to the inner diameter of the tubing. The housing was fixed to the endoscope via a PVC collar with a slot for the endoscope and four tee nuts. Screws in the tee nuts fastened the housing to the collar and the collar to the endoscope.

Communication between Positioning System and Surgical Images

The Java program of the final design uses a subset of classes from the moteJ utility package. This package was written using extensive knowledge of the technology used by the Nintendo Wii and enables users to interface directly with a Nintendo Wii Remote. For the final design, the Wii Remote was used solely as an infrared detector. The information gained from the Wii Remote's infrared detector comes in the form of pixel coordinates on the Remote's photosensitive array. Using trigonometry (Figure 17) and the widely available Wii Remote specifications, these pixel coordinates are translated into real world coordinates of the center point between two sources of infrared light. Two IR sources also allow for detection of rotation. Once the real world z-coordinate and rotation are calculated, a look-up table of images is manipulated in sequence to produce the simulation

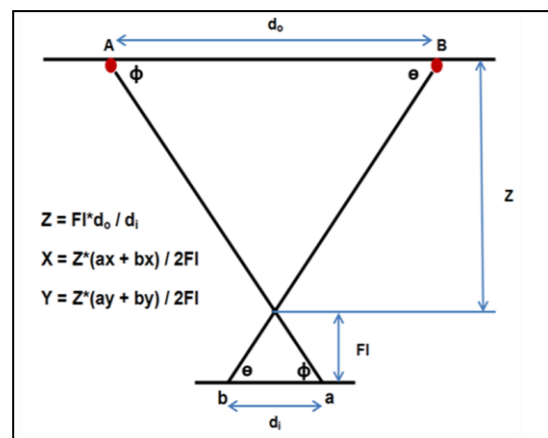


Figure 17. Trigonometric relationship and calculations for pixel coordinates.

video. A third, variable IR source is also employed to signal the program when the blade is to be deployed. When the source emits light, the blade remains down in the video. When it stops emitting, the program exchanges the “blade down” overlay with a “blade up” overlay. Furthermore, when the “blade up” overlay is visible and the z-coordinate decreases, images of the intact carpal ligament in the look-up table are replaced with images of the cut carpal ligament in order to simulate ligament cutting and facilitate checking for a successful cut.

Platform

A platform was created to house all the components (hand model, endoscope, and receiver) and increase the aesthetics of the simulator. The platform was fabricated out of 0.220 inches thick clear acrylic sheeting, and all pieces were secured together using an acrylic solvent. Three small acrylic pieces were arranged to mark the location of the hand model on the platform. In addition, a raised platform was built to mark the location of the receiver. These landmarks maintain a constant distance of approximately 50 cm between the infrared camera on the receiver and the incision in the wrist of the hand model. This distance was chosen because it fits within the z-axis accuracy range and provides adequate horizontal freedom of motion (see Testing). The purpose of choosing a constant distance is to easily calibrate and set an absolute zero point in the positioning software. Overall, the platform combines all elements of the prototype into a simple display that is visually pleasing to the user.

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 provide user instructions so that the protocol will be the same for each trial. This way, the simulation will give reliable and repeatable results on a consistent basis to provide feedback about improvement.

Fabrication

Circuit Board

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.

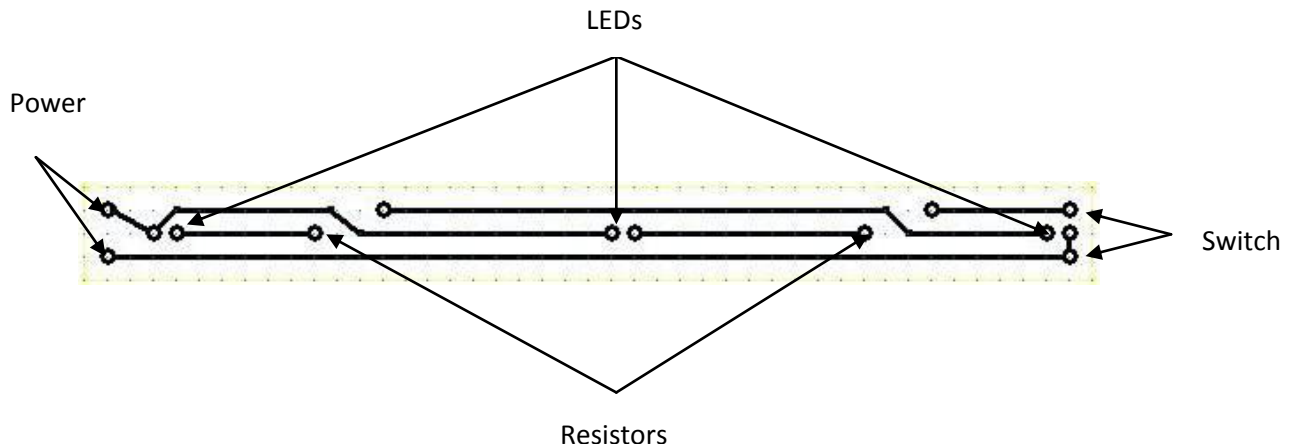


Figure 18. Schematic of the circuit board including locations of LEDs and power source.

Blade Caps

SolidWorks models of the five blade caps were submitted as stereolithography files to the SLA-Viper si2 printer. The caps are made by curing a transparent photopolymer resin (Accura 60) in 0.006 inch layers. This machine was chosen because it can create small models with high resolution and a fine surface finish. These caps were fitted on the blades and secured with Epoxy. Before attachment of the cap, the surgical steel blade was trimmed with a wire cutter, and the plastic housing surrounding the blade was removed to accommodate the width and length of the blade cap.

Testing and Analysis

The objective of testing this prototype involved determining the optimal range of LED detection by the receiver and using this information to choose the final dimensions of the platform. This was accomplished by testing how the accuracy of the positioning system changed with distance and determining the horizontal motion of the circuit board that remained in the range of the receiver. By determining the distance between the Wii remote and infrared LEDs that results in the lowest percent error, the accuracy of the prototype can be optimized. Also, recording the detectable horizontal motion of the LEDs at different initial differences will help

determine if the LED-Wii mote separation with the highest positioning accuracy allows for adequate movement throughout the simulation.

Procedure for testing positioning accuracy:

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 (see Figure 21). 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.

Procedure for determining detectable horizontal distance:

1. The starting distances between the Wii remote and the LEDs were 30, 40, 50, 60, and 70 cm.
2. The LED circuit board was placed at the above initial distances (see Figure 19).
3. The circuit was then moved perpendicular to the z-axis under the positioning program indicated it was out of range.
4. At each starting distance, the horizontal motion of the circuit board before moving out of range was recorded.

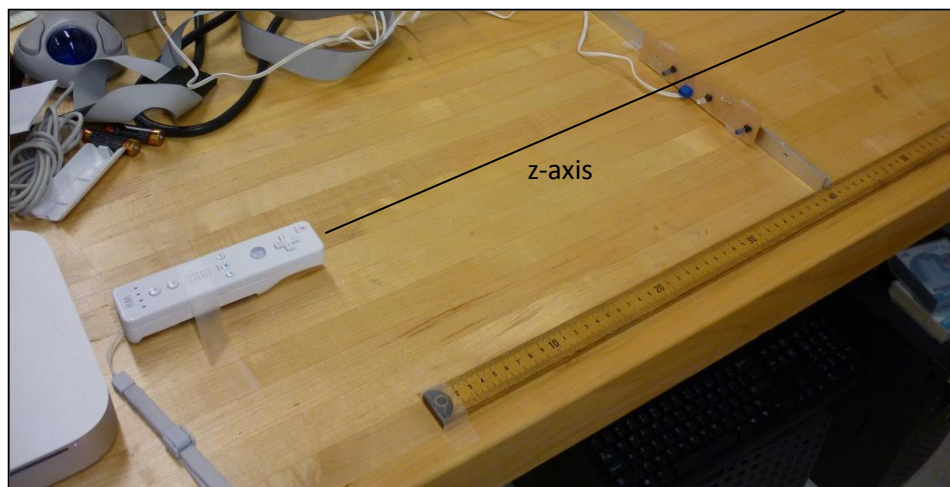


Figure 19: Testing set-up. The Wii remote is aligned with 0 on the metric ruler, and the circuit board with LEDs was moved along in 1 cm intervals.

Testing Results

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 2 and Figure 20). These results indicate that there is not a significant difference in the accuracy of the system over the tested range of initial positions.

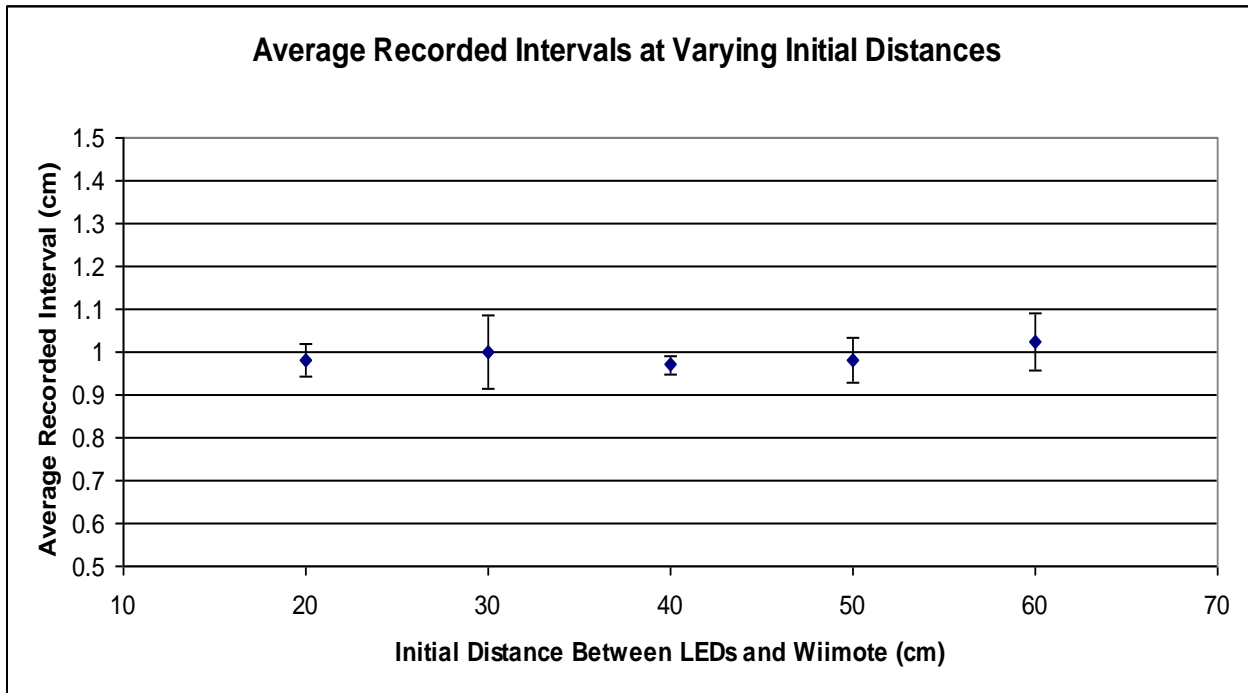


Figure 20. Graph of the average interval distances recorded by the Wii remote at initial starting distances of 20, 30, 40, 50, and 60 cm. All averages are centered on 1 cm because that was the experimental increment chosen. Error bars represent the standard deviations between the trials.

Table 2. 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

As expected, the horizontal range of motion detected by the receiver increased with increasing z-distance. A fairly linear trend was observed, increasing from $r = 6$ cm at a starting distance of 30 cm to $r = 20$ cm at a starting distance of 70 cm (see Table 3 and Figure 21). These results allow the final dimensions of the prototype to be balanced between keeping the platform small enough to avoid access bulk but large enough to accommodate the horizontal motion of the endoscope during the simulation.

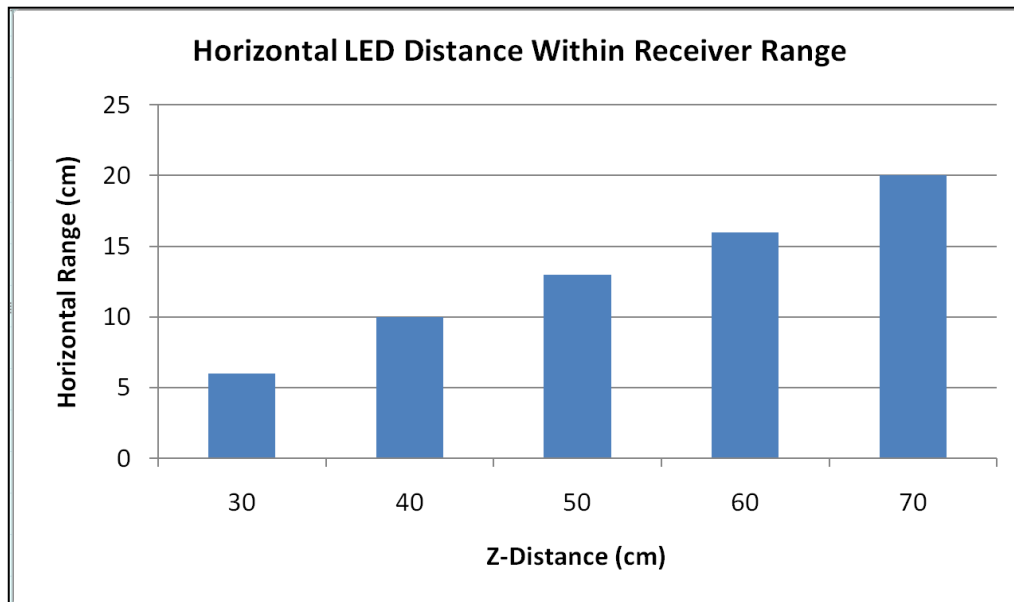


Figure 21. Graph showing the horizontal movement of the LED circuit board detectable by the receiver at the following starting z-axis distances: 30, 40, 50, 60, and 70 cm.

Table 3. Detectable horizontal movement of circuit board orthogonal to z-axis.

Starting Distance (cm)	30	40	50	60	70
Detectable Horizontal Range (cm)	6	10	13	16	20

To optimize both testing procedures, 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

Future work on the existing prototype includes lowering the height of the receiver platform in order to optimize the receivers connection with the LEDs based on the angle the endoscope is held when it is inserted into the carpal tunnel area. Also, although the mechanism connecting the endoscope to the circuit board is functional, a more ergonomic, streamlined design should be incorporated in order to increase the efficiency of simulation set-up and general aesthetics of the prototype.

In order to further optimize the haptics of the hand model and the general design of the simulation, a detailed survey should be developed and administered to a group of experienced surgeons. This survey would contain questions regarding the carpal tunnel width, the carpal tunnel depth, the transverse carpal ligament size, and the resistance supplied by the blade cap. More general questions would also be included to gain feedback on ease of use and overall aesthetics. Results from this survey would be used to optimize the complete prototype, ensuring that the simulation offers a realistic experience.

The aim of this prototype is to offer a more regulated, cost effective educational experience for developing surgical skills. With this in mind, future work includes integrating the carpal tunnel simulation into a greater learning tool that would contain step-by-step procedural instructions, pictures and descriptions of surgical instruments, and testing parameters to assess improvement. This package would then be a marketable, stand-alone teaching process that would incorporate all aspects of the surgery.

Acknowledgements

The team would like to thank our advisor, Professor Thomas 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] Lee, J. (2008). Hacking the Nintendo Wii remote. *Pervasive Computing, IEEE*, 7(3), 39-45
- [6] Volker Fritsch (April 7, 2009). Motej. Retrieved October 12, 2010, from <http://motej.sourceforge.net/>.

Appendix A

Product Design Specifications BME 402: Endoscopic Carpal Tunnel Release Surgical Simulator May 4th, 2011

Group Members: Mason Jellings, Sarah Switalski, Whitney Johnson

Problem Statement:

All surgical procedures require practice and repetition in order to achieve optimal outcomes. Currently, carpal tunnel syndrome release surgery is only performed on cadavers or live patients. Therefore, the cost of practicing this procedure is quite high. To reduce this cost and improve surgical performance, an ideal solution is an anatomically accurate simulator with realistic haptics. Our simulation device will not only reduce costs, but also improve patient safety and maximize hospital resources.

Although many surgical simulators are in use, no current devices incorporate haptics and realistic visuals simultaneously for a low cost. Hybrid surgical simulators allow for interaction between virtual and physical models where the surgeon views virtual images on a monitor but uses real instruments, but currently only exist for laparoscopic surgery. Until now, no device has existed for simulating endoscopic carpal tunnel surgery that employs the hybrid approach while providing an accurate anatomical model.

Our design is marketable to hospitals around the country as it reduces the need for cadaver utilization and increases the frequency at which surgeons are able to practice the procedure. Costs of simulation systems can range anywhere from \$75,000 for basic models, to over \$100,000 for advanced, high end models. Currently, the unit cost of our device is approximately \$900 with a possible selling price of \$10,000. Not only is this a considerable reduction in cost, but our design could also serve as a model for the development of subsequent simulators, opening tremendous commercial opportunity across surgical fields.

Our device consists of a silicone-based hand model, an endoscope fitted with a circuit board housing three LEDs, and a receiver which reads three-axis position data from the LEDs. This combines both a mechanical model and software images to create a realistic training tool. Our design has been tested for both accuracy and range of distance readings. Feedback from experienced surgeons was implemented to help optimize the prototype and produce realistic haptics throughout the simulation

1. Design Requirements:

The device must meet all of the client requirements

- a. Performance requirements: The force feedback provided by the hand model should replicate the forces experienced by the surgeon when performing the procedure. An increased resistance should be felt when the blade is deployed. Transition between endoscope images should be smooth.
- b. Safety: Device should not cause any harm to the surgeon or other connected devices involved with the simulation.

- c. Accuracy and Reliability: Signaling device should be able to detect position of the endoscope within 1 mm. The tracking system should maintain accuracy of position throughout multiple simulations. The force feedback mechanism should provide a consistent amount of force in each simulation.
- d. Life in Service: Simulator should be able to withstand repeated uses as a training tool for surgeons for a minimum of 5 years.
- e. Operating Environment: Device will be used at room temperature and standard humidity. The simulator will not be used in the operating room so does not need to be sterilized.
- f. Ergonomics: The haptic feedback from the hand model should replicate the forces felt while performing the actual surgery. The external circuit should be positioned so that does not interrupt the surgical technique. The method of device tracking should not alter the physical components of the endoscope or its blade attachment.
- g. Size: Hand model should be life size and the incision site should be 5 mm proximal of the distal wrist crease. The carpal tunnel should be 1 cm in diameter. The transverse carpal ligament should be 5 mm thick. The ligament corrugations are 1 mm in height and thickness with 1 mm of spacing between consecutive corrugations.
- h. Materials: Hand material looks like and has mechanical properties similar to carpal tunnel tissue. The force feedback mechanism should be compatible with the silicone material used in the hand model.

2. Production Characteristics:

- a. Quantity: One working prototype is necessary.
- b. Target Product Cost: \$1000

3. Miscellaneous:

- a. Standards and Specifications: No specific standards because prototype is only used in simulation as a training tool, not actual surgery.
- b. Customer: The client will use the device to train other surgeons on the endoscopic carpal tunnel release procedure. The tracking system will be incorporated with a virtual environment created by the client in Adobe Director.
- 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, however has not been applied to carpal tunnel surgery.

Appendix B

Java Code:

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

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

//import java.awt.*;
//import org.javabluetooth.stack.*;
//import org.javabluetooth.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(FI, 2d) + 2d*Math.pow(d_i, 2d)))*(FI - d_i),
Math.sqrt(1d/(2d*Math.pow(FI, 2d) + 2d*Math.pow(d_i, 2d)))*(d_i + FI));

//double z = .5*Math.sqrt(1d/(2d*Math.pow(FI, 2d) + 2d*Math.pow(d_i, 2d)))*(d_i*FI +
Math.pow(d_i, 2d) + 2d*Math.pow(FI, 2d))*d_o/d_i;
int ex0 = p0.x;
int ex1 = p2.x;
int wy0 = p0.y;
int wy1 = p2.y;

```

```

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

```