Telemanipulation and Telestration for Microsurgery

600.446 Computer Integrated Surgery II

Final Project Report

Team Members: Orhan Ozguner, Robert Eisinger

Mentor: Marcin Balicki

Abstract

In this project, we developed a number of additional features for Microsurgical Work Station by enhancing an already-existing, error-prone telemanipulation framework for the Steady-Hand Eye Robot. Microsurgeries such as Vitreoretinal surgery require an immense amount of precision, concentration, and skill. During a surgery, problems such as exhaustion and fatigue are certainly not uncommon. This project addresses these issues by proposing a tele-system of cooperative control, allowing multiple surgeons to work together in a single surgery. This system includes the use of the Omni Haptic Device for telemanipulation, telestration, bilateral telemanipulation, and bimanual cooperation.

1. Introduction

Microsurgery is a difficult form of surgery for many reasons. Surgeons often operate near the threshold of human dexterity. Consider the common vitreoretinal surgery *Membrane Peeling* to heal a Macular Pucker. In this procedure, a surgeon must peel an especially thin membrane while applying constant force. The surgeon must ascertain that the membrane is not ripped or damaged during the process. Hence, the surgeon peels very slowly and carefully. It is a difficult task requiring great concentration and an immense amount of precision. A commonality among microsurgeons performing a distinct set of procedures is the awareness that exhaustion takes a toll [5]. There is little room for errors or mistakes when surgically being confined to a space that might be only span a few millimeters.

One way of tackling this problem is to develop a cooperative control system such as the Steady Hand Eye Robot, a five-degree-of-freedom fine control robot with a positional accuracy on the order of 10s of microns [1]. A cooperative control system allows the surgeon's superior intelligence and experience to be used with greater precision. It integrates safety and information into the procedure without taking away control from the human surgeon [2]. Specifically, the Steady Hand Eye Robot allows for hand over hand control. The surgeon *and* the robot hold the tool tip, a needle end effector used in vitreoretinal surgery. This adheres to the guidelines for a cooperative control system; the surgeon can control the needle as he/she normally would, but in addition, the robot can introduce even more precision. The Steady-Hand Eye Robot in particular has tremor-reducing capabilities allowing the surgeon to better position the needle with higher accuracy [2].

While this is useful for improved accuracy, it does not necessarily cut back on the surgeon's exhaustion, the aforementioned problem. To this degree, we propose a telemanipulation system, an additional cooperative control system. It takes the current system (tremor reduction, fine control, etc.) and enhances it even more through the use of telemanipulation. Telemanipulation is the concept of using a *master* device that controls the *slave* device. It is the idea of using another device (here, the Phantom Omni Haptic Device) to control the Eye Robot rather than controlling the Eye Robot directly. Prior to this work, the Steady Hand Eye Robot already had a very rough sketch of a telemanipulation system implemented [3]. It provided us with the connections necessary to have multiple devices communicating with each other. However it had errors that will be further discussed below.

The telemanipulation system we developed – which includes telemanipulation, telestration, bilateral telemanipulation, and bimanual cooperation – has many implications. Firstly, two surgeons are able to work together during a surgery. One surgeon controls the slave, whereas the other surgeon, at any point in the surgery, can take over and use the master to control the slave. This particular paradigm is useful for educating surgeons. The trainee, for instance, could sit at the slave while the instructor sits at the master. During a procedure or a mock procedure, the instructor can clutch in and take over whenever necessary. Secondly, two experienced surgeons can essentially perform a surgery together. When the primary surgeon is exhausted, the secondary surgeon can take over. Thirdly, a telestration system allows for communication between the two surgeons without the primary surgeon moving his attention from the surgery. The surgeon controlling the Omni is able to graphically annotate over a 3D display that receives its video feed from the microscopes used in surgery. Fourthly, bilateral telemanipulation – whereby both the master and the slave control the robot at the same time – allows the surgeons to both actually perform the surgery simultaneously. Fifthly, through the use of an external device to control the Eye Robot, we are able to implement a scaling feature that allows the Eye Robot to move at even greater precision. Lastly, a bimanual cooperation set up – using two Eye Robots at once – gives surgeons a way of performing the operation using both hands. Further,

The advantages obtained by using the Omni Haptic device as the master in this telemanipulation system are particularly beneficial. The Steady Hand Eye Robot does not currently have any haptic feedback. By controlling the robot with a haptic device, the surgeon can better understand what he/she is doing by relying on not only vision, but by relying on both vision and tactile feedback. And in a telemanipulation system, there are a number of further features that may be implemented to truly utilize haptic feedback to its fullest capabilities, such as using telestration to define a virtual fixture that constrains the Eye Robot.

2. Technical Approach

Here, we describe the telemanipulation system we developed and review the algorithms necessary for modulating the different tasks of the different components involved in the system.

2.1 System Setup

Figure 1 depicts the general system set up in our project. The force sensor box listens for forces applied to the Steady-Hand Eye Robot, which drives the robot. Separately, there is a microscope with a left and a right video feed, allowing the surgeon to view the eye in high detail. The microscope video feeds relay into a 3D display that the surgeon can optionally view rather than looking directly into the microscope. The master in this system set up is the Phantom Omni Device, which is connected via network.

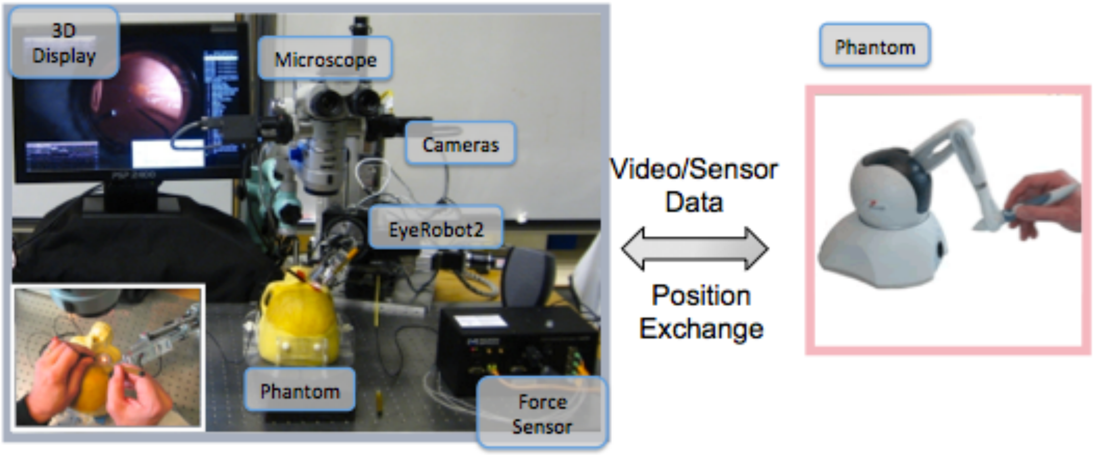


Figure 1. Our system consisted of a 3D Display, a Microscope, the Steady Hand Eye Robot, a Phantom, a Force Sensor, and a Phantom Omni Haptic Device.

This system setup was utilized for all portions of our project with the exception of bimanual cooperation. Bimanual cooperation makes use of two Eye Robots, two computers, and two phantom omni devices. It is essentially what is shown in in Figure 1, except two of each of the robotic components.

2.2 Telemanipulation

The telemanipulation system relies communication between the Eye Robot and the Omni device, as well as listening for clutch actions. These components are each connected to a Global Component Manager, as seen in Figure 2, which is responsible for sending a receiving messages between all the respective components. The telecomponent application is responsible for producing the desired actions and sending messages to the Eye Robot after receiving messages from the Omni device.

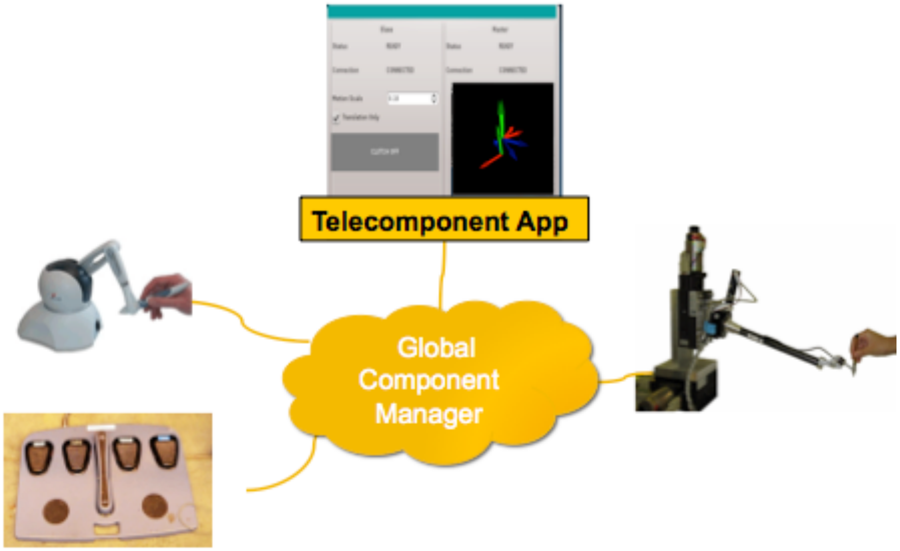


Figure 2. The various components involved in the telemanipulation system include the Omni, the Eye Robot, the clutch controller, and the Telecomponent Application. These were all connected to each other via a Global Component Manager.

These connections existed prior to our project work. However, the system was not working properly.

The first problem was that the robot and the Omni master systems have different frame orientations. In our case, we find the master difference and apply the same difference to the slave side by scaling according to the slave. Since the frame orientations are different, we had unexpected movement during our testing. In order to solve this problem, we find master frame orientations and robot frame orientations, and rotate either one to another.

The second problem was on the translation only mode, where we were expecting the robot to move only in the x, y, z directions; however, it started using its rotations to reach the desired goal. We found out that the robot control algorithm tends to use easier access to a desired position. For that reason, it was tilting and bending after some point. To cancel this problem, we implemented Constraint Cartesian Motion Control Algorithm [4]. In order to implement the method we took the following steps.

1) **<u>Specifying Task Frame Objective Function and Constraint</u>**

Given a tolerance, ϵ_g, to reach the desired location such that:

$$\left\| \begin{bmatrix} \Delta^g \mathbf{x}[1] \\ \Delta^g \mathbf{x}[2] \end{bmatrix} - \begin{bmatrix} \Delta^g \mathbf{x}_d[1] \\ \Delta^g \mathbf{x}_d[2] \end{bmatrix} \right\| \leq \epsilon_g \tag{1}$$

Where $\Delta^g x_d$ and $\Delta^g x$ are the goal and actual gaze displacements.

(1) can be replaced as a family of linear equations of the form:

$$\begin{aligned} [\cos(\theta_k), \sin(\theta_k), 0, 0, 0, 0]^T \cdot (\Delta^g \mathbf{x} - \Delta^g \mathbf{x}_d) \leq \epsilon_g, \\ k = 1, \cdots, n \end{aligned} \tag{2}$$

and also this linear equations can be rewritten in the form:

$$\mathbf{H}_g \Delta^g \mathbf{x} \geq \mathbf{h}_g \tag{3}$$

A weighting diagonal matrix is formed such that the coefficient matrix (H) correspond to each actuator will limit the actuator motion proportional to these coefficients. Therefore, we will have another function to optimize such that:

$$\|\mathbf{W}_g(\Delta^g \mathbf{x} - \Delta^g \mathbf{x}_d)\| \tag{4}$$

where W_g is a diagonal matrix that specifies relative importance of each actuator.

This procedure is applied to both end effector constraints (H_e) and joint limit constraints (H_j). From this, we get two more diagonal weighting matrices, W_e, W_j that represent minimization of the end effector motion and minimization of the joint motion in the form of objective function matrix.

2) Putting All Together:

When all defined weighting matrices are combines in a single equation, the objective function and the constraint equations can be represented as follows:

$$\left\| \begin{bmatrix} \mathbf{W}_g & & \\ & \mathbf{W}_e & \\ & & \mathbf{W}_j \end{bmatrix} \left(\begin{bmatrix} {}^g\mathbf{J}(\mathbf{q}) \\ {}^e\mathbf{J}(\mathbf{q}) \\ \mathbf{I} \end{bmatrix} \Delta \mathbf{q} - \begin{bmatrix} \Delta^g \mathbf{x}_d \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \right) \right\| \tag{5}$$

also we can form this as follows:

$$\begin{bmatrix} \mathbf{H}_g & & \\ & \mathbf{H}_e & \\ & & \mathbf{H}_j \end{bmatrix} \begin{bmatrix} {}^g\mathbf{J}(\mathbf{q}) \\ {}^e\mathbf{J}(\mathbf{q}) \\ \mathbf{I} \end{bmatrix} \Delta \mathbf{q} \geq \begin{bmatrix} \mathbf{h}_g \\ \mathbf{h}_e \\ \mathbf{h}_j \end{bmatrix}. \tag{6}$$

which can be represented as the final form of

$$\textbf{minimize } \|\mathbf{A}\ \Delta \mathbf{q} - \mathbf{b}\|, \textbf{ subject to } \mathbf{C}\ \Delta \mathbf{q} \geq \mathbf{d} \tag{7}$$

This final form of the equation can be solved using least square method.

3) Assignment of Optimization Weights:

For a specific purpose of an optimization process, assigning weighing coefficients for each actuator is very important. In our case, we need to define rotational weights higher than translational weights to escape using rotations. We needed to adjust weights according to our system.

After all of these adjustments, translation only mode started working the way we expected. It was a matter of trial and error to fine tune the weights and the parameters.

2.3 Telestration

The software for telestration is located in the TeleStereoCapture application. This application handles the input from the microscope and sends it to the 3D display. It is here that we intercepted this input, added the ability to graphically overlay, and implemented the graphics engine.

The telestration system is very much modulated. Figure 3 provides a schematic for this system. All the widget buttons are connected to their position in a widget master, which is essentially a sidebar. The sidebar pops in and out when the Omni, which controls a mouse pointer on the 3D display, becomes close to the far right side of the screen. The widget master is responsible for listening for events, such as the user pressing a button on the Omni indicating that he/she would like to draw a straight line, for example.

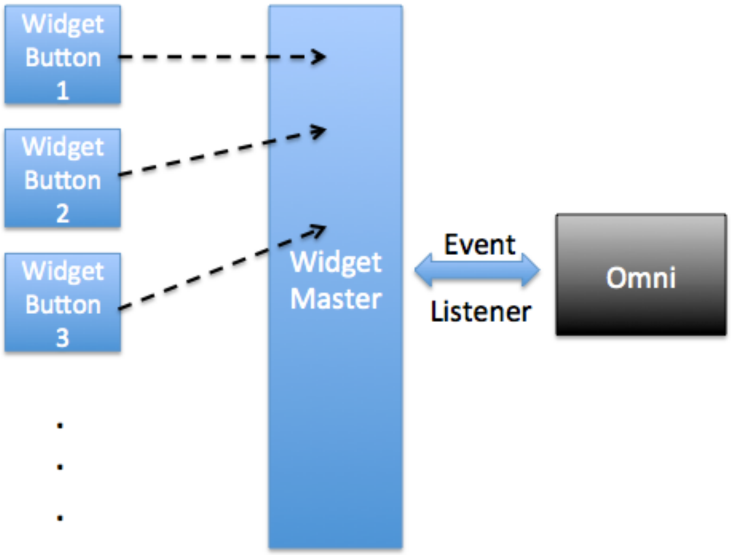
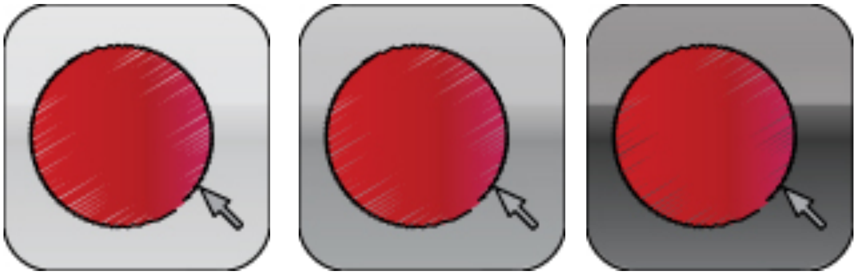


Figure 3. Telestration consisting of a Widget Master containing Widget Buttons and listening for events from the Omni.

Our current system allows for straight lines, curly lines, squares, circles, and erasing. To erase, the user selects the eraser, and then drags the mouse pointer (using the Omni) over a graphics object, such as a line. Using a cross product algorithm to determine the distance to a line, the system highlights a graphics object that it believes the user is attempting to erase. Once that object is highlighted, the user simply taps the button on the Omni and the graphics object is erased. While these seemed to be the most important graphical components, the system is highly abstract, and it is very feasible to add additional graphic utilities to the sidebar, further enhancing the usability of the telestration system.

The GUI for this portion of the project consisted of images. See below for an example (circle drawing).



Each graphics component had three images, one for when the cursor is not over the image, one for when the cursor is over the image, and one for when the cursor has clicked the image.

2.3 Bilateral Telemanipulation

On our project, we have another mode of telemanipulation called Bilateral telemanipulation. In this mode, we want to give more freedom to the slave side. That means, during the master/slave control for the Steady-Hand Eye Robot, the slave will be responsive to given direct input from its sensors. For the position controller, we adopted M.Mahvash and A. Okamura's position tracking controller method [6]. During the implementation, we made a new component called Teleopcomponent, and collected all necessary frames to this face. After that we made the necessary calculations; in our case, this consisted of setting it to the mid-point of the master and slave frame difference. Than desired position frames are set to both master and slave sides. The master side, the Phantom Omni, only responded with force feedback and keeps its end effector to the desired position. In order to implement this mode to the slave side, we used Cartesian Constraint Motion Control Algorithm [4] by adding force on the robot handle to the desired motion.

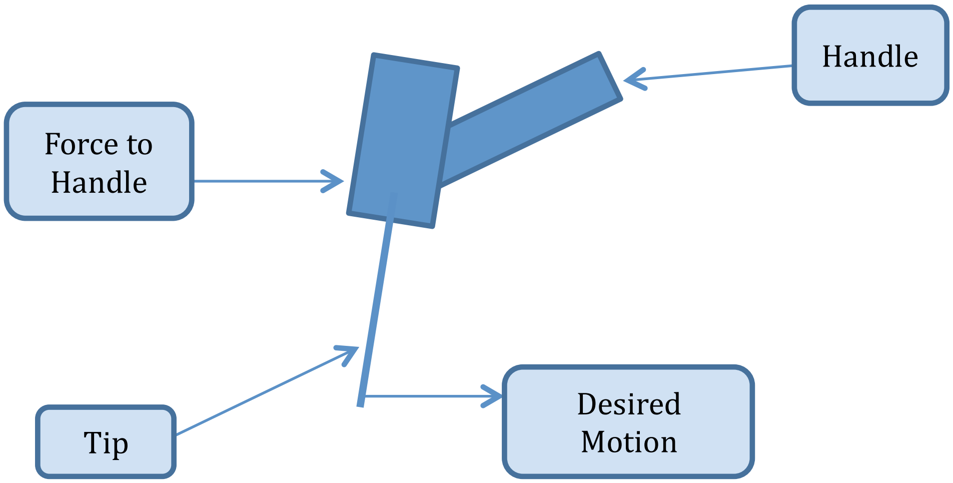


Figure 4. Problem definition for Bilateral telemanipulation.

In order to add handle force, we converted the desired motion to force and transformed it to handle by using an adjoint matrix. Since both desired motion and handle force are at the same frame and unit, we simply add them and use it in the constraint algorithm.

2.4 Bimanual Cooperation

Bimanual cooperation is simply an extension of what we developed in the telemanipulation system discussed above. It is simply a matter of duplicating the system, and combining. In this manner, we are able to control two Eye Robots using two Omni devices. However, we only use one 3D display, so it must be designated which Omni device will be used as the telestrator device. This is a minor problem and requires no novel solution. Rather, it is simply a matter of connecting the Teleo Stereo Capture application to the correct Omni.

3. Results

To see an image displaying our telemanipulation and telestration systems, please see https://jshare.johnshopkins.edu/reising1/public_html/.

The results of our telemanipulation system were tested multiple times and, from feedback from our mentor, has been deemed working. Our telestration system is also working successfully, as seen in the images below.

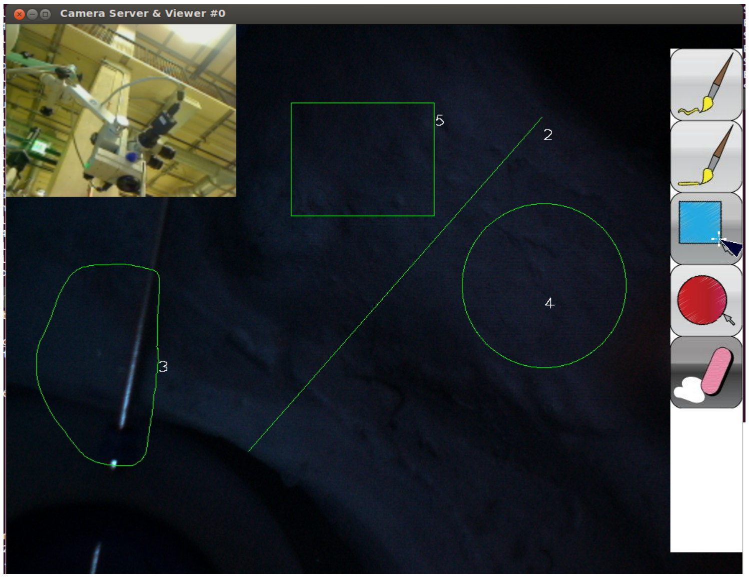


Figure 5. Screenshot of the telestration system.

Developing a brand new workstation on a brand new computer, and performing a telemanipulation/telestration over the Hopkins network validated our entire system.

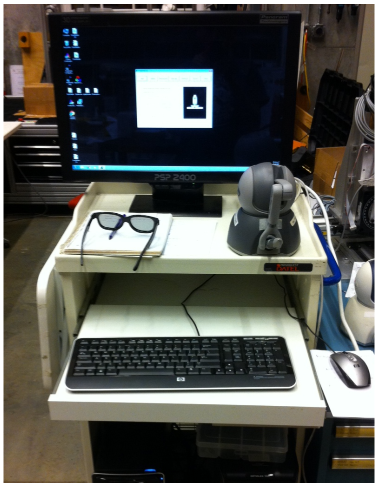
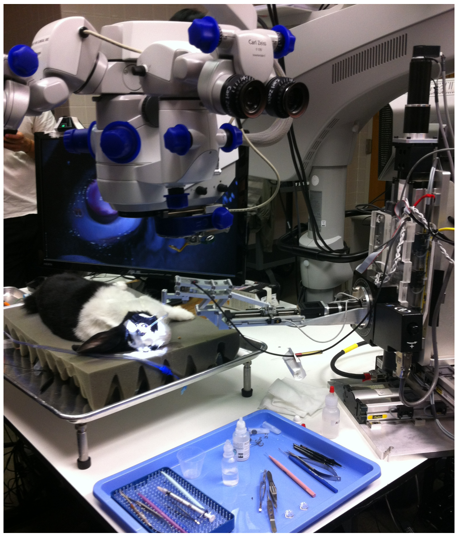
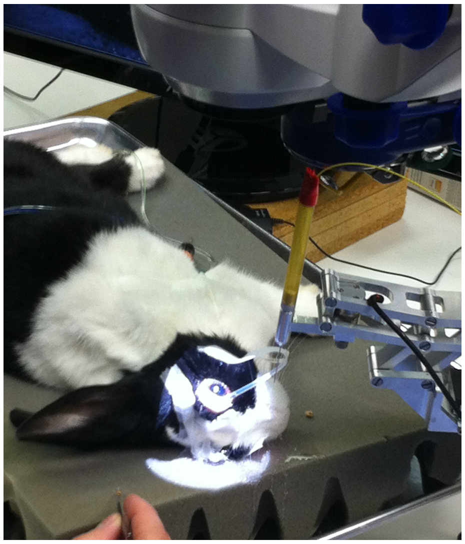


Figure 6. Workstation setup.

This was a tedious task that required installing many libraries, etc onto a new machine. After the installation and set up was complete, we simulated a mock telemanipulation, controlled the Eye Robot, drew some graphics, and confirmed that our system fully worked over wireless internet.

Furthermore, we tested our telestration experiment during a bunny expeirment. The pictures below display the experimental set up, as well as the different components of the Eye Robot. The experiment gave us some ideas for further work, and was in general, a success.





4. Discussion

In our results, we showed that we are able to telemanipulate the Steady Hand Eye Robot successfully over a network. This is a proof-of-concept more than anything else. It is a full tele-workstation that is completely controlled offsite.

The telemanipulation system, as we have demonstrated, is fully doable. The telemanipulation of the robot is an open project, per se. We have showed one method of controlling the robot, via position exchange and constrained control motion algorithm. However, there are many other algorithms which may be applied. Future work can explore these options.

In particular, bilateral feedback requires great attention. While we demonstrated a rough framework for a bilateral cooperation system, there are many questions which are still unanswered. What happens if one surgeon moves left, and the other moves right, each with difference forces? Which surgeon should have more control than the other? There are a number of questions which should be pursued by future studies, and there is a great wealth of room for work in this field.

As a whole, the system does truly work as a cooperative system. In fact, the system requires at least two people to fully operate. This is exactly what our goal was. We successfully developed a system that is compatible with multiple surgeons. It could very well be used for training, education, communication, etc, and surely has the potential to expand into a larger workstation for even many more surgeons to cooperate all at once on much larger surgeries.

5. Conclusion

We have successfully implemented a full-featured tele-workstation that includes telemanipulation, telestration, bilateral telemanipulation, and bimanual cooperation. The system is working well, but future work is required. The system should be optimized for different machines, as well as be expanded to include further enhancements to the graphical user interfaces, especially. In a system that is controlled by more than one user simultaneously, it is especially important for it to be user friendly. This is an area of further work.

In addition, there are many enhancements to the tele-workstation which can be added in after further dependencies are resolved. For instance, retinal tool tracking is required to use the Omni to define virtual fixtures via telestration. Once the tool tracking is working properly, we will be able to implement the tracking and use telestration to constrain the motion of the Steady Hand Eye Robot.

The tele-workstation that we implemented is a proof-of-concept framework for a much larger scale telemanipulation and telestration design flow. The idea that a surgery can take place away from the actual patient is remarkably powerful. Many possibilities open up with this proof-of-concept implementation. We propose many new ideas, such as three surgeons working cooperatively on a much more large-scale surgery such as heart surgery. The basic framework exists; next, it can be expanded and enhanced.

References

1- Uneri et. al., "New Steady-Hand Eye Robot with Micro-Force Sensing for Vitreoretinal Surgery," *IEEE RAS & EMBS*, 2010.

2- Kumar et. Al., "Performance of Robotic Augmentation in Microsurgery-Scale Motions," *MICCAI* , 1999.

3- Uneri A., Sevinc G., "Teleoperation of the Eye Robot," *Computer Integrated Surgery-II Final Report,* 2009

4- J. Funda, R. Taylor, B. Eldridge, S. Gomory, and K. Gruben, "Constrained Cartesian motion control for teleoperated surgical robots," *IEEE Translations on Robotics and Automation,* vol. 12, pp. 453-466, 1996.

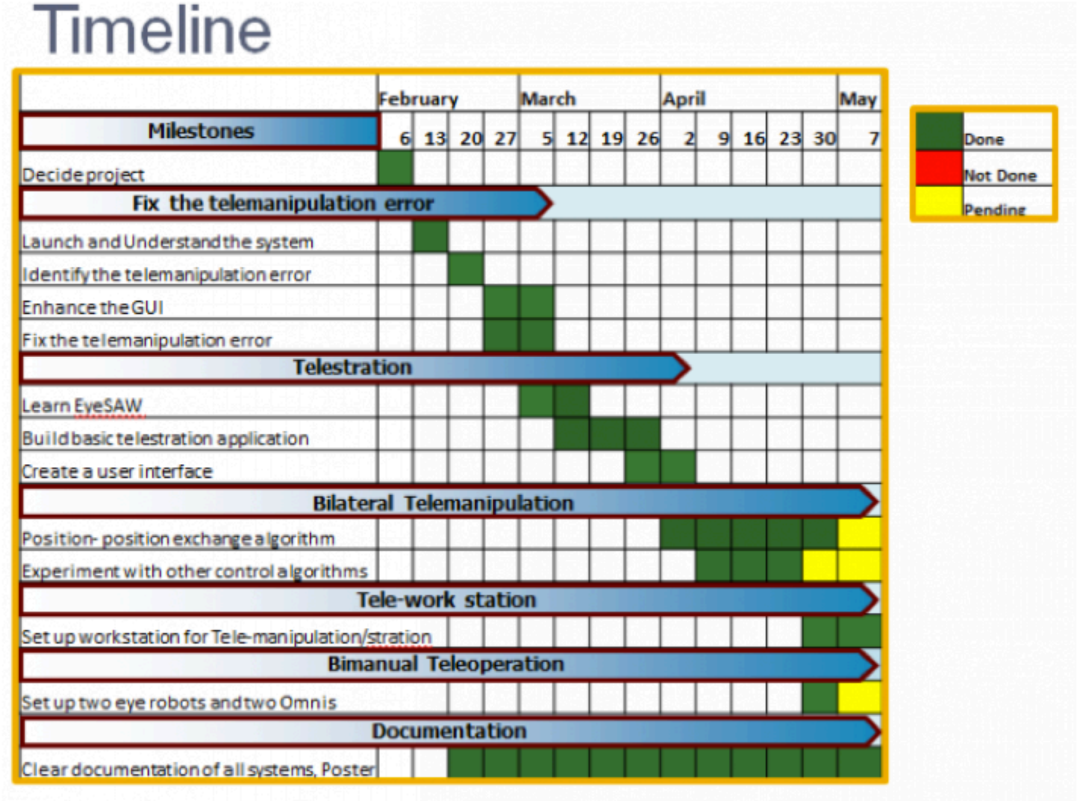
5- S. Ku and S. E. Salcudean, "Design and control of a teleoperated microgripper for microsurgery," vol. 1, pp. 889-894, 1996.

6- M. Mahvash and A. Okamura, "Enhancing transparency of a position exchange teleoperator," *WorldHaptics Conference*, pp. 470–475, 2007.

Acknowledgements

We would like to thank our mentor Marcin for his valuable help and support throughout the project. His input and guidance was immeasurable. We would also like to thank Professor Taylor for his suggestions along the way; for guiding us to use constrained control algorithm, and for giving us feedback on our various components of our system.

Management and Milestones Summary



Timeline

	February				March				April					May
Milestones	6	13	20	27	5	12	19	26	2	9	16	23	30	7
Decide project	🟩													
Fix the telemanipulation error														
Launch and Understand the system		🟩												
Identify the telemanipulation error			🟩											
Enhance the GUI				🟩	🟩									
Fix the telemanipulation error				🟩	🟩									
Telestration														
Learn EyeSAW					🟩	🟩								
Build basic telestration application						🟩	🟩	🟩						
Create a user interface								🟩	🟩					
Bilateral Telemanipulation														
Position- position exchange algorithm									🟩	🟩	🟩	🟩	🟩	🟨
Experiment with other control algorithms										🟩	🟩	🟩	🟨	🟨
Tele-work station														
Set up workstation for Tele-manipulation/stration													🟩	🟩
Bimanual Teleoperation														
Set up two eye robots and two Omnis													🟩	🟨
Documentation														
Clear documentation of all systems, Poster			🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩

🟩	Done
🟥	Not Done
🟨	Pending

Figure 7. Project Timeline

Figure 7 above displays our final milestone completion timeline. We decided our project by February 6th. The first major milestone we wanted to reach was to fix the telemanipulation error. In order to do this, we needed to understand the system (determine how to turn on the system, and learn how to target the spots of the software we would be interested in) as well as identify the error, enhance the GUI, and then fix the error.

We worked together on this portion of the project, under Marcin's close guidance. By February 13th, we were able to launch the system. This consisted of learning how to turn on the Robot such that it was usable. In turn, this required the launching of the Global Component Manager, the Omni Configuration, the Omni Master, and the Teleop Manager, as well as the 3D Display. Neither of these components was straightforward to get set up properly. Next, we needed to begin looking at the code for the Eye Robot in order to identify the telemanipulation error. We were able to pinpoint the error and attribute it to an optimization algorithm the Robot was using to move. By February 20th, the source of the telemanipulation error was located inside of a specific block of code. Our next task was to fix the telemanipulation error, and enhance the GUI. This consisted of adding features to the GUI to allow for translation-only, RCM, and bilateral feedback modes. To fix the telemanipulation error, Orhan implemented a constrained control motion algorithm. These tasks were complete by March 5th.

Our next major milestone was Telestration. At this point in the project, we wanted to speed up a bit, and thought it would be useful for one of us to stick with Telestration while the other moves on to Bilateral Feedback. At this portion of the project, Robert worked on Telestration while Orhan worked on Bilateral Feedback, which was the next major milestone.

In order to successfully telestrate, it was necessary to learn EyeSAW (the graphics system for the Eye Robot), build a basic telestration application, and then build a GUI for telestration. The first task, to learn EyeSAW, was completed by March 12th. Robert learned the different components of EyeSAW, such as the TeleStereoViewer application, as well as the TeleStereoCapture application. Next, Robert worked to build a basic telestration application, which was integrated into the TeleStereoCapture application. This application provided the framework for the connections between the 3D Display, the Phantom Omni, and the telestration algorithm for graphical overlay. This was completed by March 26th. By April 2nd, the full telestration system was running (the beta version), including the graphical user interface, which consisted of image buttons for line drawing, curly drawing, circles, squares, and erasing. This required coding for user events, as well as Omni events, and responding appropriately. Lastly, it required the user of data storage to maintain a list of graphical components.

Meanwhile, Orhan was working on bilateral telemanipulation. This proved to be a difficult task that no one had attempted before. It was difficult mostly due to the fact that both sides of our telemanipulation system (Eye Robot on one side, Omni on the other) were not the same device. This caused differences in movement algorithms, as well as frames of reference. There were two major algorithms that Orhan needed to implement during this portion of the project. The first was position to position exchange, and the second was the constrained control motion algorithm. Orhan also experimented with a few other ideas for algorithms. While progress was made in both parts of bilateral telemanipulation, the telemanipulation still needs to be refined. Currently, the system works for the Eye Robot moving in a steady motion However, for non-constant motion between the Omni and the Eye Robot, the system needs some fine tuning. These tasks remain as Pending, while the other portion of bilateral telemanipulation were complete by April 30th.

It was also necessary for both of us to work together to develop a Teleworkstation to use as a way of validating our system. We started with a brand new computer running linux. We needed to install Windows to begin with. Next, we installed many different libraries, as well as CISST and other graphics packages. In essence, we needed to start from scratch and make an Eye-Robot and Phantom Omni compatible machine. And the current machine that the Eye Robot was attached to was the only one prior to this project. It was a difficult task and required the close guideance of Marcin to help facilitate various errors that would arise. We had the system up and running by May 7th.

Throughout the whole process, we both were careful to document our code and ensure that software we added was well understood. In addition, we helped develop some text files that detailed how to turn on the system and ensure that things are working properly.

In general, we both worked together (other than the bilateral telemanipulation, which Orhan mostly worked on, and Telestration, which Robert mostly worked on) and kept each other updated every day. We reached our milestones on time, and were able to reach our minimum (telemanipulation and telestration) and expected deliverables (bilateral telemanipulation). Unfortunately, we were unable to achieve our maximum deliverable (bimanual cooperation, da Vinci telemaniulation), due to outstanding dependencies. Bimanual cooperation relies on two Steady Hand Eye Robots, and the second Eye Robot is still being built. For this reason, we were unable to try out bimanual cooperation. Also, we were unable to port our code over to be used with the da Vinci console due to lack of time. These tasks remain as future work.