
Prototyping a Hybrid Cooperative and Tele-robotic Surgical System for Retinal Microsurgery

Release 0.6

Marcin Balicki, Tian Xia, Min Yang Jung, Anton Deguet, Balazs Vagvolgyi,
Peter Kazanzides, Russell Taylor

June 15, 2011

Johns Hopkins University, 3400 N. Charles St. Baltimore, MD 21218 USA

Abstract

This paper presents the design of a tele-robotic microsurgical platform designed for development of cooperative and tele-operative control schemes, sensor based smart instruments, user interfaces and new surgical techniques with eye surgery as the driving application. The system is built using the distributed component-based *cissst* libraries and the Surgical Assistant Workstation framework. It includes a cooperatively controlled EyeRobot2, a da Vinci Master manipulator, and a remote stereo visualization system. We use constrained optimization based virtual fixture control to provide Virtual Remote-Center-of-Motion (vRCM) and haptic feedback. Such system can be used in a hybrid setup, combining local cooperative control with remote tele-operation, where an experienced surgeon can provide hand-over-hand tutoring to a novice user. In another scheme, the system can provide haptic feedback based on virtual fixtures constructed from real-time force and proximity sensor information.

Contents

1	Introduction	1
2	System Architecture	3
3	Control Schemes and Applications	7
4	The Conclusions and Discussion	8

1 Introduction

Vitreo-retinal surgery is the most technically demanding ophthalmologic discipline. It addresses common sight-threatening conditions including retinal detachment, complications from diabetes, macular pucker,

macular hole and removal of retina-associated scar tissue [1]. In current practice, retinal surgery is performed under an operating stereo-microscope with free-hand 20-25 gage instrumentation. In most cases, three incisions in the sclera (sclerotomy) are required: one for infusion to control the intra-ocular pressure, one for a fiber-optic “light pipe” and one for a surgical instrument (see Figure 1). The surgeons often operate in bimanual fashion with a “light pipe” in one hand and a forceps, laser, vitreous-cutting probe, fragmenter, aspiration, or another type of tool in the other hand. A typical vitro-retinal surgical task is peeling epiretinal membranes from the surface of the delicate retina. Another task which is considered to difficult but very desirable is cannulating 100um diameter retinal vessels for targeted drug delivery. Surgeons strive to overcome many human and technological limitations that include physiological hand tremor, poor visualization of surgical targets, and lack of tactile feedback in tool-to-tissue interactions.

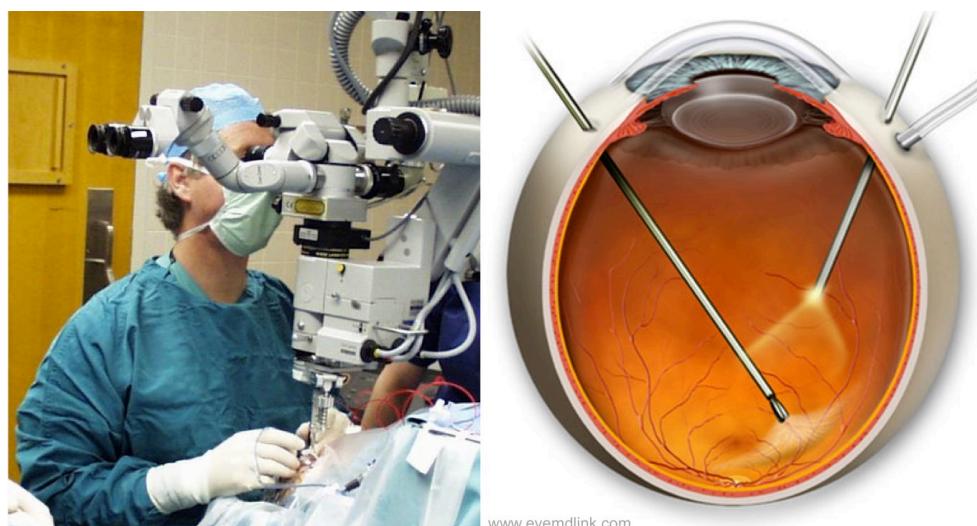


Figure 1 Left: Surgeon operating. Right: Forceps and light pipe Instrument inside the eye.

We address these challenges in a systems-based approach, creating the Eye Surgical Assistant Workstation (*eyeSAW*) platform for development and testing of micro-surgical robots, intra-operative sensors, human-machine interaction, robot control methods and new surgical procedures with vitro-retinal surgery as the driving application.

Robotic manipulators are inherent parts of the system and can provide the needed stability and precision. Although there are numerous robotic eye surgery concepts, we used two for this work. One type is the cooperative control robot, such as the EyeRobot2 (ER2), where the surgeon and robot share the control of the surgical instrument [2]. The main advantages are that the operator interaction with the surgical instruments is familiar and direct but much steadier than freehand operation and that the surgeon can remove the tool from the eye at any moment, without delay. This is very important in cases where the patient is locally anaesthetised and awake, and can move unexpectedly.

Another type is a tele-operation system where the surgeon controls the robotic manipulator from a remote master console [3], the best known example is the da Vinci Surgical System® (Intuitive Surgical, Inc.), a commercially available and clinically approved tele-robotic system for Minimally Invasive Surgical procedures (MIS). This system has similar advantages of minimizing hand tremor, but can provide an even finer degree of tool control by employing a motion scaling scheme. There are a few disadvantages,

including difficulty in performing safe gross motion outside of the eye due to lack of visualization, significant reliance on correct definition of the location of the remote-center-of-motion mechanism that prevents excessive motion of the eye by constraining tool motion to intersect the sclerotomy incision location, and the increased slave design complexity to comply with stringent safety requirements.

We believe that incorporating these two paradigms into a single hybrid tele-robotic and cooperative system will combine the advantages and supplement the weaknesses found in the respective standalone systems. In this paper, we present a prototype of such tele-robotic surgical system, including overall architecture, EyeRobot2 and daVinci Master manipulators, and visualization, and various user interaction modes that incorporate tele-operation, cooperative control, and real time sensing information (see Fig. 2).

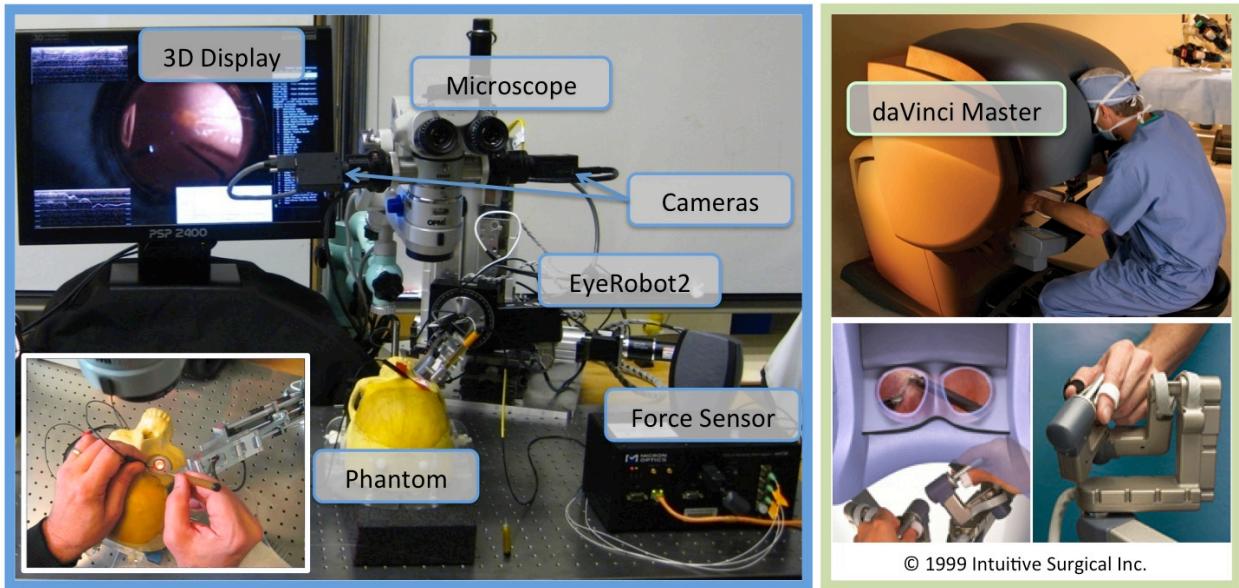


Figure 2 Telerobotic System: slave side display with EyeRobot2; The daVinci Master Console.

2 System Architecture

The *eyeSAW* relies heavily on the concept of component based system design by which new surgical system configuration and functionality can be rapidly prototyped with minimal or no modifications to existing devices and applications. The *cisstMultiTask (MTS)* library provides this underlying component-based software framework [9]. It supports two different architectures, a multi-threaded architecture that has optimal performance for robot control applications and a multi-process or distributed architecture that is scalable and extensible. The same programming model [4] allows these two architectures to be seamlessly and flexibly combined together to build a system with minimal source code changes at the user level. The connection and coordination of components in a system is managed by the Global Component Manager (GCM), which is unique to the system and defines the system boundary.

The basic building block is a software component which has a list of standard commands contained in provided and required interfaces. The component (A) that needs specific functionality has a *required interface* that is connected to another component (B)'s *provided interface*, which provides that functionality (see Figure 3). Upon connection, a component A can initiate a command, which executes B's function or receives an event command from B. One advantage is that B can be replaced by another component/device (C) that supplies a matching provided interface required by A. When the components are in the same process, the command execution is comparable to a standard function call. In the case when the components are distributed over a network, a typical round trip execution of a command over a multi-hop network with a payload of 150 bytes is 300 μ s, which, for most applications, is an acceptable execution time.

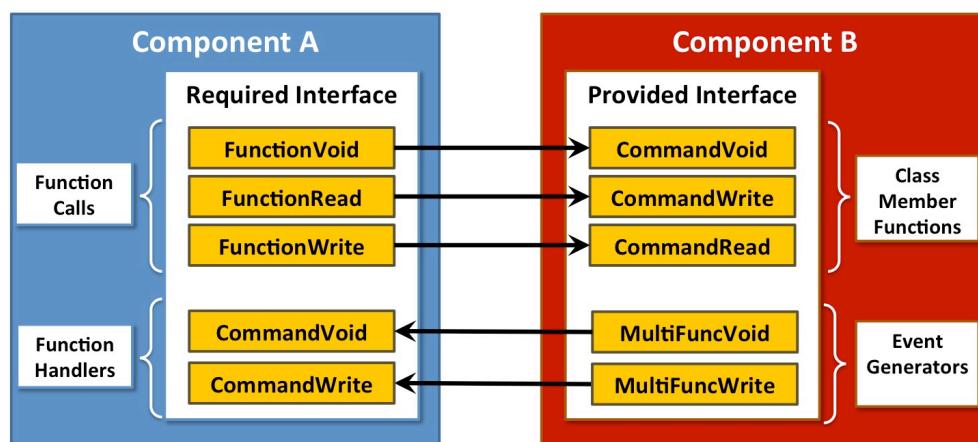


Figure 3 Component Interface Connections.

Using the above framework we have created a master/slave tele-robotic surgical system that uses existing devices and their component software interfaces (see Figure 4). A new component was added, serving as a bridge between the two robotic manipulators and executing tele-operation algorithms. To facilitate tele-visualization we have built a custom stereo video encoding and display subsystem.

da Vinci Master Console (DMC)

Through a collaboration agreement with Intuitive Surgical Inc., we have acquired the mechanical subsystem, which includes two Patient Side Manipulators (PSM) and a daVinci Master Console (DMC) unit, comprising of two Master Tele-Manipulators (MTM) and stereoscopic display system. Each MTM is a cable driven 7-DoF serial robot with a redundant wrist mechanism. Our custom system, using custom electronics and software, allows for low-level joint level control, enabling haptic force feedback research, which is not available on the commercial system.

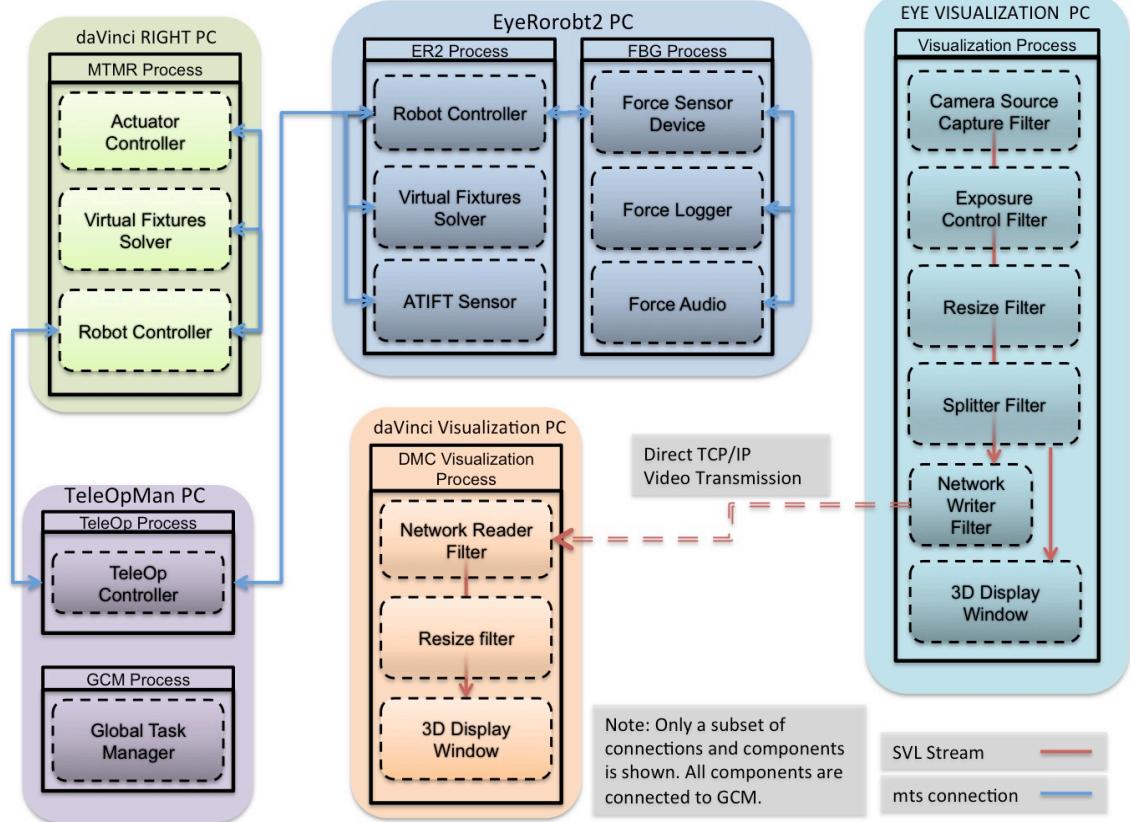


Figure 4 eyeSAW hardware/software view schematic with connections.

We want to use intra-operative sensing, from OCT or force sensors, to generate virtual fixture (VF) motion constraints that help the surgeon to perform a task. Virtual fixtures are task-dependent motion constraints for a robotic manipulator performing a task by limiting its movement into a restricted workspace and/or influencing its motion along a desired path and are classified as Forbidden Region (FRVF) or Guidance (GVF) virtual fixtures, respectively. VF primitives such as stay above a plane, move along a line, and rotate about a point can be combined using different operators to provide assistance for complex surgical tasks. We implement VFs using the constrained optimization formulation below, where $C(X(q+\Delta q), X_d)$ is the objective function associated with the difference between the actual state variable X and the desired state variable X_d . The state, $X = X(q+\Delta q)$ is a function of joint variables q and joint incremental motion Δq . The solution vector Δq_c must satisfy motion constraints in the form of one or more inequalities $A(X(q + \Delta q)) \leq b$. In addition, we make use of constraints that involve Δq_U and Δq_L , the upper and lower rate limit for Δq . For each control step, the optimization controller computes a desired incremental motion Δq_c , based on constraints and objectives generated from various real-time input sources [5].

$$\begin{aligned} \Delta q_c &= \min C(X(q+\Delta q), X_d) \Delta q \\ \text{s.t. } A(X(q + \Delta q)) &\leq b, \Delta q_L \leq \Delta q \leq \Delta q_U \end{aligned}$$

We also implement bilateral feedback. When the slave encounters a force in the environment, (e.g. obstacle), it lags behind the command position and the user haptically feels this resistance, as the bilateral feedback objective acts to minimize the daVinci master and EyeRobot2 tracking error, by opposing the user's input motion on the daVinci master.

EyeRobot2 (ER2)

The Eye Robot2 is a cooperatively-controlled robot assistant designed for retinal microsurgery. In cooperative control the surgeon and the robot both hold and control the surgical instrument simultaneously. The ER2 can assist during high-risk procedures, by incorporating virtual fixtures to help protect the patient, and by eliminating physiological tremor in the surgeon's hand during surgery. It is a 5-DOF serial manipulator with XYZ stages for the base and two rotary stages that create a mechanical Remote Center of Motion (RCM). The user "drives" the robot by manipulating the tool which is attached to a 6-DOF force sensor. Like the MTM above, the control software is based on the *cisst* library, complies with the *SAW* framework, and uses the Constrained Control Optimization library to calculate incremental joint motions. Its high level control loop runs at 250Hz and commands the motion controller running at 1kHz (Galil DMC1886 PCI). We have also designed a line of robot-compatible smart surgical instruments that have embedded micro-sensors. These provide direct real-time feedback to the robot which uses this information, if needed, to affect the surgeon's movements to prevent undesirable collisions with the retina, limit forces applied to the delicate tissues, or intuitively guide the surgeon to a particular target in the eye.

Visualization

Currently, video is the sole feedback modality for the surgeons operating inside the eye. In creating a tele-robotic setup we required a video stereo-microscope system with low latency, high frame rate, high resolution, high dynamic range, and Ethernet based communication for long range telecasts. Therefore, we have developed a high performance visualization sub-system built using the *cisstStereoVision* library (SVL). SVL provides a wide array of highly optimized *filters*, such as capture device interfaces, image processing, network transmission functionality, overlays and video formatting for stereo-specific displays. The library is highly multi-threaded, adopting the GPU streams concept, where each filter in the pipeline has access to a pool of threads to process the latest video frame. Configuring these filters in a pipeline is simple, allowing for rapid prototyping of various display architectures, e.g. tele-operation. Fig. 4 shows the two display applications linked by a TCP/IP connection.

The slave system's visualization comprises a standard ophthalmological surgery stereo-microscope (Zeiss OPMI-MD) outfitted with two IEEE-1394B capture cameras (Grasshopper by Point Grey Research) each capturing 1024x768 pixels resolution images at 45 FPS. The left and right video streams are interlaced and rendered on a passive 3D display (PSP2400 by Panorama Technologies) which separates left/right lines using passive polarized glasses. The capture and display latency is a few frames, on a modern multi-core Linux OS workstation. The stereo video stream is also split, encoded with lossy jpeg compression, and sent using *NetworkWriter Filter* over TCP/IP to the master's display subsystem. Intraframe encoding was chosen to minimize latency. In our prototype system we used a switched gigabit Ethernet network to transfer the dual XGA progressive video stream with a fairly low, approximately 6% (16x) average compression ratio in order to keep the quality loss introduced by the compression close to the source

image SNR. The resulting video bandwidth was around 56Mbps. Encoding latency is under 20ms thanks to the custom multi-threaded codec implementation.

The master's visualization application running on an older multi-core windows XP machine, uses a very similar SVL pipeline but the video source is provided by the *NetworkReaderFilter*. This video stream is formatted for DMC's two display CRTs, at 1024x768 pixels per video channel. The display console itself is very ergonomic allowing the optimal alignment of visual and motor axes and offering excellent stereopsis. The final display frame rate is about 25FPS, with barely noticeable delay of approximately 4-5 frames.

Tele-Operation Manager (TOM)

The TOM is a central component responsible for configuring, monitoring the state, and high-level control of the master and slave components. It also performs position offset computation and clutching. A typical TOM control loop queries the robotic manipulators for their latest Cartesian pose via standard MTS interfaces (e.g., *GetFrame(mtsFrm4x4)* command), computes the next desired pose and executes a command that sets the goal Cartesian pose on the robotic manipulators (e.g. *SetGoalFrame(mtsFrm4x4)*). The robots run their own high frequency control loops that servo the robot to the desired pose. The TOM also provides a Qt based GUI that allows the operator to change the tele-operation method, and the workspace motion scaling between the daVinci Master and the Eye Robot.

3 Control Schemes and Applications

The flexibility and inherent functionality allowed us to quickly prototype a variety of control schemes that extend the functionality of the system beyond the sum of its parts. The two robots and TOM, utilize the robot control constrained optimization framework to calculate incremental joint motion based on system state and given task constraints, e.g. force sensor information. Here are a number of control schemes we have implemented.

Classic **unilateral tele-operation (UTO)**, currently used by the daVinci system, does not include force feedback and the operator is required to close the control loop via the visualization system. This method is sufficient for simple tasks but it lacks fidelity for more complex maneuvers or when visualization is poor and the operating environment does not allow the slave to follow the master, resulting in tracking error.

To diminish the effects of inherent human hand tremor and enable more precise maneuvers found in micro-surgery, **motion scaling (MS)** was implemented. MS in a tele-operation system involves scaling the Cartesian position of the master by a factor relative to the Cartesian position of the slave.

In **bilateral tele-operation (BTO)** control the behavior of the slave is fed back into the master controller, and vice versa. If the slave motion is impeded by the environment, the motion of the master manipulator will also be impeded, pragmatically creating a sense of force-feedback. It is important to note that the EyeRobot2 has only 5DOF of control, so even though the tele-operation algorithm is commanding 6DOF Cartesian position and orientation, the desired rotation about the tool axis is omitted by the ER2 control optimizer.

The BTO control scheme is compatible with **virtual fixtures (VF)** [6] such as the virtual RCM (vRCM) where the tool motion is constrained so the tool shaft always intersects a single point in the robot workspace, e.g. the trocar inside the sclera [7]. The vRCM is implemented on the slave side and inherently reflected back to the master controller via the BTO logic. Since vRCM constrains the motion of the tool to 3 DOF there are a few ways to operate the master side. One way is to implement standard BTO on the master side, and another is to only consider the Cartesian translation of the master (operator is free to rotate the master joystick without affecting the slave) to drive the Cartesian position of the tool tip within the eye.

We have also created **dynamic virtual fixtures** on the slave manipulator by incorporating real time information from sensors embedded in surgical instruments [8]. One example is micro-force sensing, where interaction forces measured at the tool tip are virtually scaled up through cooperative control. Since the BTO involves position-position exchange the operator on the master side will experience this scaled force reflection.

The intuitive “hand-over-hand” control interface of the EyeRobot2 is very effective in general setup tasks such as gross positioning of the tool near the surgical site, aligning it for insertion through the sclerotomy trocar, or adjusting the position of the vRCM relative to the eye. These tasks are especially difficult in case of pure tele-operation due to lack of macro-field visualization from the microscope. In our system, a technician can quickly position the robot to the desired site, and the surgeon can “clutch-in” with the master and, at any time, take over control with BTO. The BTO can provide very fine positioning for a specific task, especially when used with motion scaling. Furthermore, we have developed a controller located on the slave side that considers both the master’s desired motion through BTO, as well as the desired motion of the ER2 operator. This **hybrid tele-operation and cooperative control** (HTCC) enables both operators to contribute to the control of the slave manipulator, while complying with any motion constraints, such as vRCM. Depending on the task, it may be more advantageous to have one operator use hands-on control of the slave to prepare for a maneuver and seamlessly “transfer” the control to an operator on the master side to perform the delicate maneuver with finer motion-scaled manipulation. Further, the level of contribution from the two inputs does not have to be equal. By adjusting the relative gains, one of the operators can be the dominant operator. Such control scheme could become a valuable educational tool, where the trainer directly guides the trainee through common surgical tasks.

4 The Conclusions and Discussion

The process of developing this tele-operation system has shown that component based software architecture is very convenient for multi-device system prototyping and development. The resulting system is functional and can easily be adopted to accept new devices or to fit other applications, such as micro-vascular surgery, or neuro-surgery. For example, the daVinci master can be replaced with the Phantom Omni (SensAble Technologies Inc) to control EyeRobot2’s tool tip while inside the eye with haptic feedback of the 3DOF. Similarly, in the next iteration of the system we will add another EyeRobot and the other master manipulator to construct a bi-manual tele-operation system.

The actual process of development with *cisstMultiTask* was very smooth. Occasionally, additional component interfaces needed to be built, or existing ones extended. The debugging and testing of new

modes of operation was efficient due to the ability to disconnect and reconnect specific components/processes without restarting every process in the system.

Despite the slight latency, the tele-visualization system has sufficient frame rate for development purposes. Since our video capture application can handle 1600x1200 px resolution, the system is compatible with newer daVinci master consoles containing higher resolution displays, with little or no compromise in performance. When the master and slave are separated by great distances, the performance will be limited by the large network bandwidth requirement for the JPEG-based intraframe video encoding. For these cases we are considering changing the video encoder to achieve a higher compression ratio and adopting UDP for lower latency. Long distance tele-operation will also require the transmission of audio.

Since EyeRobot2 was designed for pure cooperative control it does not have an actuated tool rotation axis or the ability to open/close instruments on its own. As is, it can be used with axisymmetric instruments such as a surgical pick or illumination guide; for use with more sophisticated tools it will be upgraded with a new tool actuation mechanism that will allow for tele-operation of all available degrees of freedom of the surgical instrument.

The operator of the master console indirectly experiences the dynamic virtual fixtures located remotely on the slave manipulator through standard bilateral tele-operation feedback. In order to provide a higher fidelity feedback for the master console operator, the master can create virtual fixtures locally using real time sensor information collected by the slave system.

The introduction of robots into the surgical flow generally adds to the operation time, yet this disadvantage can be outweighed by the benefits that they provide. From a practical point of view, **hybrid tele-operation and cooperative control** can overcome a problem when the conventional tele-robotic concept is applied to eye surgery: patient motion. This is especially problematic since the patient is awake and locally anesthetized for most operations. For example, the operator of the EyeRobot2 can assess patient state and anticipate patient motion, e.g., a sneeze.

The ultimate goal of the eyeSAW project is to enable more surgeons to perform currently impossible treatments and improve the safety and success rates of existing vitreoretinal procedures. With tools provided by the cisst/SAW infrastructure we can quickly progress towards that goal.

Acknowledgements

This work was partially funded by the National Science Foundation (NSF) under Engineering Research Center grant EEC9731748, National Institutes of Health (NIH) BRP 1 R01 EB 007969-01 A1, ARCS Foundation, and by the Johns Hopkins University internal funds.

Reference

- [1] Ida Y. et al. *Microsurgical robotic system for vitreoretinal surgery*. Int J CARS 2011
- [2] Uneri A. et al. *New steady-hand EyeRobot with micro-force sensing for vitreoretinal surgery*. In: Proceedings of IEEE BioRob, pp 814–819, 2010.
- [3] Das H. et al. *Evaluation of a telerobotic system to assist surgeons in microsurgery*. Computer Aided Surgery 4(1):15–25. 1999

- [4] Jung MY et al. “A Component-based Architecture for Flexible Integration of Robotic Systems,” In IEEE/RSJ IROS, pp. 6107-6112, 2010
- [5] A. Kapoor, M. Li, and R. H. Taylor. *Constrained Control for Surgical Assistant Robots*. In Proceedings of IEEE ICRA, pp. 231-236, 2006.
- [6] Abbott JJ. et al. Stable Forbidden-Region Virtual Fixtures for Bilateral Telemanipulation. *Journal of Dynamic Systems, Measurement, and Control*, 2006.
- [7] Mitchell B et al. *Development and application of a new steady-hand manipulator for retinal surgery*. In Proceedings of IEEE ICRA, pp 623–629, 2007.
- [8] Balicki MA. et al. Micro-force sensing in robot assisted membrane peeling for vitreoretinal surgery. In: Proceedings of MICCAI 2010.
- [9] cisst Library <http://trac.lcsr.jhu.edu/cisst>
- [10] Jung MY et al. “A Surgical Assistant Workstation (SAW) Application for a Teleoperated Surgical Robot System,” in MICCAI Workshop on Systems and Architecture for Computer Assisted Interventions, Midas Journal. 2009