Portable Surgery Master Station for Mobile Robotic Telesurgery

Ganesh Sankaranarayanan* Hawkeye I King* Seong-Young Ko† Mitchell J.H. Lum* Diana C.W. Friedman* Jacob Rosen* and Blake Hannaford*

*BioRobotics Laboratory
Department of Electrical Engineering
University of Washington
Seattle, Washington 98195
E-mail:{ganeshs,hawkeye1,mitchlum,dwarden,rosen,blake}@u.washington.edu
†Human-Robot Interaction Research Center
Korea Advanced Institute of Science and Technology
Seoul, South Korea
E-mail:kosy@robot.kaist.ac.kr

***Abstract*— We describe a system that provides a low-cost, portable control station for experimentation in mobile robotic telesurgery. The software and hardware implementation of our system are described in detail. The device mapping between the Haptic Interface Devices (HID) and the surgical robot that enable the surgeon to effectively teleoperate the surgical robot are explained along with our communication protocols for telesurgery. We have also provided our initial results from extensive field testing of our system in different hardware and software configurations and challenging locations. We focus on working under sub-optimal network conditions for field operation in remote environments, and the importance of interoperability and distribution among networked surgical technologies.**

I. Introduction

Minimally invasive surgical procedures have revolutionized the area of surgery and patient care. With the advent of surgical robots, the increased dexterity, tremor control, and ergonomic interface have helped in applying MIS procedures to wider and more complex surgical cases. In robot-assisted minimally invasive surgery, the operating surgeon uses a robotic interface device (the master) to control a surgical manipulator robot (the slave) to do operating procedures. The commercially available da Vinci surgical robot [1] is an example of this kind of surgical system. However, robot-assisted procedures are currently limited to teleoperation with the surgeon located in the same operating room.

Telesurgery allows expert surgeons to perform life-saving procedures without having to travel to remote corners of the globe. The capability of remotely teleoperating a surgical robot offers many possibilities for providing critical care, particularly considering that life-saving procedures that are commonly done in the developed world are often out of reach for a large segment of people. Surgeons would also be able to provide vital medical care to injured soldiers while avoiding dangerous exposure in an active battle field.

Telesurgery brings many new challenges compared to conventional robotic surgery. First, to be able to teleoperate over long distances, the telesurgery system should be stable for the large latencies experienced over global-scale operation. Moreover, the system must also be stable in the presence of the jitter and packet loss characteristics of usual Internet traffic. Second, in case of communication blackout, a backup system should be in place to complete any ongoing surgical procedures. Also, in the absence of force feedback, the surgeon teleoperating the robot relies on stereo video feedback. Therefore, the system must be capable of providing high-quality video with minimum possible latencies. Unfortunately providing reliable high-speed video feedback via the Internet —where bandwidth is often limited— is still a great challenge. Finally, it is beneficial to provide haptic feedback to the surgeons, since it helps minimize damage to the tissues and organs during long-distance procedures. Fig. 1 shows the state of the art master station used to operate the da Vinci surgical robot. This FDA-approved, commercially available robot is not capable of performing telesurgery.

There has been active research in the past few years focused on designing and implementing telesurgery systems. In [2] an 8-DOF surgical robot called the *Black Falcon* was built, controlling the robot through the commercially available Sensable PHANToM haptic device. In [3] a 6-DOF laparoscopic telesurgery workstation was implemented with another commercially available haptic device —the 4-DOF Immersion Systems Impulse Engine 3000, with additional 2-DOF actuators— controling the surgical robot. Telesurgery over long distances has been demonstrated as well [4], [5] by successfully teleoperating a Zeus surgical robot (Computer Motion, Inc.) to perform a procedure on 68-year-old female patient from New York to Strasbourg, France, using a dedicated ATM. More recently, there has been work in telesurgical systems at the University of Tokyo [6] and remote surgical procedures from Japan to Thailand [7] performing a laparoscopic cholecystectomy on a porcine model. In addition, at the BioRobotics Laboratory in the University of Washington,

ROBOCOMM 2007, 14th – 16th Oct 2007, Athens, Greece.

ISBN 978-963-9799-08-0
DOI 10.4108/ICST.ROBOCOMM2007.2217

Fig. 1. Da Vinci master console. (Image courtey Intuitive Surgical, Inc ©2007)

a surgical robot named RAVEN, specifically suited to mobile telesurgical operations, was built [8], [9].

A. Goal

The objective of this paper is to present the design of a portable low-cost surgical master station for teleoperated surgical robots. Design goals for the device included:

- Low cost
- Off-the-shelf hardware
- Interoperability with multiple surgical robots
- Use of Internet protocols for communication flexibility
- Support of data collection in experimental surgical robotics.

While not designed or intended for actual human surgery, the present system has proved very useful for a variety of experiments which could have important implications in that domain.

II. System Description

A functional block diagram of the portable surgical master station is shown in Fig 2. Hardware elements include a Dell Inspiron laptop computer with a 2GHz Intel Core Duo processor and 1GB RAM, two Omni haptic devices (Sensable Technologies, Cambridge MA), and a USB foot pedal. In addition, video decoding and display were used in several configurations, described below.

Surgeon Site Software (SSS) consists of two pieces of software: the surgeon's graphical user interface (SGUI) and a Haptic Device Client (HDC) that communicates with the two haptic devices through their API, and with the remote surgical robot using UDP/IP. These two parts are separate applications running on one machine and communicating via TCP packets. Additionally, SGUI events are posted to a central log server via HTTP, keeping track of surgeon training time, proficiency and robot longevity.

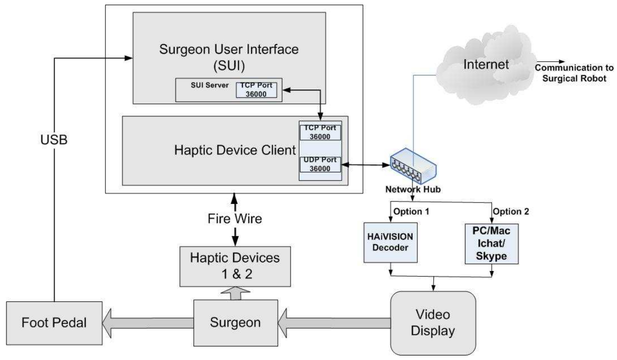

Fig. 2. Surgeon site software block diagram

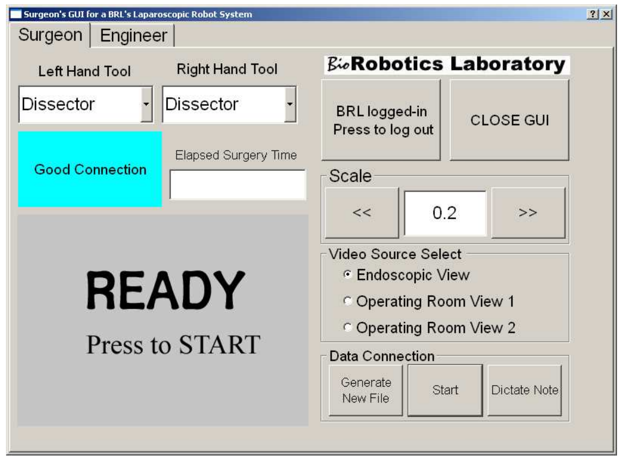

Fig. 3. SGUI window

A. SGUI

The SGUI allows the surgeon to execute high-level commands during medical procedures. The SGUI was written using the GPL release of Qt 4.1.2 from Trolltech, Inc. A snapshot of the SGUI window is shown in Fig. 3. The SGUI has a secured interface in which each surgeon needs to enter a password to gain access for operating the surgical robot. Once authenticated, they can start the surgery session by pressing an on-screen button with the mouse or touch screen. Movement of the surgical robot can be performed in either of the two windows selected through the SGUI tabs on top (*Surgeon* and *Engineer*).

The *Surgeon* window provides options and information relevant to the surgeon. For example, a variable scale factor setting controls the reduction of the surgeon movement to between 5% and 100%. In addition, a status display shows when all components of the SSS are ready. Another display indicates whether the SSS is idle, ready to operate or in active control of the robot while a real-time clock shows the elapsed time of the current operation. There is also an option to log out and switch surgeons.

In the *Engineer* window, technical parameters of the master station are set. The remote IP of the surgical robot (the patient site) can be selected from a drop-down box. Also, The SSS

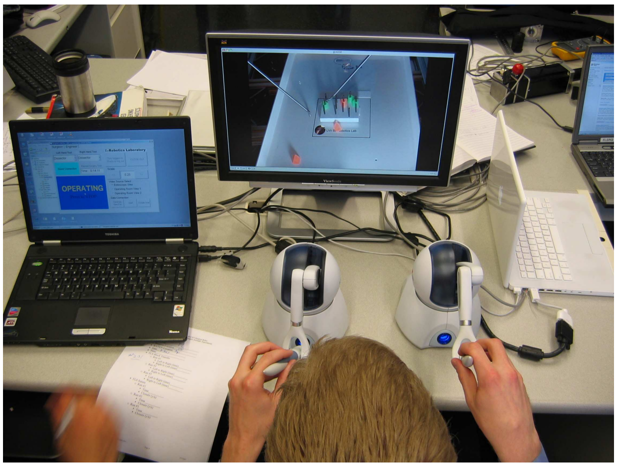

Fig. 4. System setup. SGUI (Fig. 3) appears on the laptop screen, surgical video transmission on the LCD monitor behind two Omni haptic devices. Setup is shown in-use during the NEEMO-12 experiment.

supports variable transmission rates for optimal performance in given network bandwidth conditions, and while the default rate is 1 packet/msec (1000Hz), it can be reduced by integer multiples (1000/n Hz, *n* integer). This option is customizable through the *Engineer* window.

The SSS uses a TCP/IP client-server model for communication between the SGUI and the HDC. The SGUI maintains a TCP server on port 36000, and the HDC connects as a client. Both the programs generally run on the same laptop but could be divided onto two if more computing power were needed.

B. HDC

The HDC connects with the Haptic Interface Devices (HIDs) that control the robot. It also transmits commands to the robot using UDP on port 36000. UDP is used for communication because its low overhead makes it a fast and lightweight protocol, suitable for high packet transmission rates. These high packet rates along with the incremental transmission scheme used by this system makes it robust in case of occasional packet loss, while a checksum algorithm protects against corrupt packets.

Once user options are selected, the HDC begins communication with the robot. Every millisecond (or longer if the packet rate has been reduced), the software reads the Cartesian position of each haptic device, checks the change in position since the last interval, and sends position increments to the robot. The HDC can run either arm, or both arms simultaneously: this option is selected at runtime, when the program is launched. The commercially available PHANToM Omni haptic devices are normally utilized for input, and PHANToM 6-DOF haptic devices have also been used.

C. Haptic Device Mapping

The motion of the HID must be related to the motion of the robot in a way that is easy for surgeons to use. To this end, the kinematic mapping between the HID and the robot is entirely in Cartesian space, and the joint kinematics of the HID and the surgical robot are handled separately by their respective, lower-level software. Thus, motion commands are position increments, rather than absolute position, which allows indexing or clutching of the HID stylus (see *indexing* below).

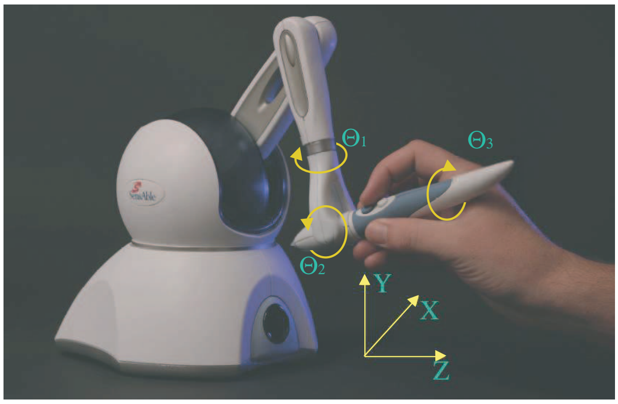

Fig. 5. Reference frame and orientation axes used in communication with surgical robot. (Image courtesy SensAble Technologies.)

Position in cartesian space is scaled and mapped from the HID to the surgical robot. The HIDs should sit on a table with the stylus rest positions toward the user and the video display set on the table behind the devices. Fig. 4 shows the surgeon console setup with a view of the robotic manipulators in the monitor. With this arrangement up/down, left/right, and in/out motion of the HIDs should cause corresponding movement of the end effectors in the viewing field. Rearranging the camera, HIDs, or user relation to the video monitor will adversely affect usability and performance of the system [10]. The reference frame used by the communications protocol for transmitting position increments is a right-handed frame with the X-axis pointing right, Y pointing up, and Z pointing out (Fig. 5). At the patient site, the slave robot makes whatever transformation is necessary for its own kinematics. The transmitted position and orientation increments are defined as follows:

$$X_n = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (1)$$

$$\Psi_n = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \quad (2)$$

where n is the current sample number, x, y, z are the current positions of the Omni end-point in the frame of the Omni, and $\theta_{1,2,3}$ are the three gimbal angles of the Omni. The increments of motion sent to the remote robot are given by

$$\Delta X_n = \gamma(x_n - x_{n-1}) \quad (3)$$

$$\Delta \Psi_n = \Psi_n - \Psi_{n-1} \quad (4)$$

The units of ΔX_n and $\Delta \Psi_n$ are microns and micro-radians respectively.

Further transformations to the end effector frame are performed on the robot side. In our current implementation, the tools fitted to the RAVEN do not have a sixth degree of freedom. In this setup, θ_2 (the elevation angle of the pen) is mapped to tool wrist, and θ_3 (pen roll) is mapped to tool shaft roll. The RAVEN controllers, software, and transmission are nevertheless designed to handle 6-DOF.

Orientation is a one-to-one mapping —without a scaling factor— from roll and pitch of the HID stylus to roll and wrist rotation of the surgical robot tool. The HID typically has three intersecting axes of rotation near the stylus tip, so when the slave robot has only two tool orientation degrees, as with the RAVEN robot, the third degree of orientation is simply ignored. This simple orientation mapping, and given that commands are sent as increments, means that a reference frame is not required (see *future work* for further discussion of this topic). Finally, two buttons on the HID stylus were used for controlling the grasping action of the surgical tool, with button 1 opening and button 2 closing it.

1) Indexing: After continuous manipulation, the situation in which the hands of the surgeon have moved to an extreme of the Omni workspace or to an uncomfortable position frequently arises. In these cases, the surgeon must be able to modify an offset in position and orientation between the hand-held haptic devices and the slave robot hands in order to move his/her hands to a more comfortable or manipulable position without changing the surgical robot (position indexing). To this end, when the foot pedal is disengaged, the system continues to send packets to the robot, but the commanded position increments are held to zero, ignoring motion of the HIDs. Therefore, each time the foot pedal is depressed, the current position of the end effector of the haptic device in the Cartesian space is recorded as the reference for the next position increment computation (eqn 3). As a result, the surgeon can continue to comfortably operate the device and avoid kinematic constraints of the HID.

2) Communication Protocol: Robot kinematics commands are sent from the surgical master to the slave as raw binary data over UDP. The binary datagram contains 12 data fields defined in this C/C++ structure:

```
typedef struct {
unsigned int sequence;
int c_timestamp;
int s_timestamp;
int delx[2];
int dely[2];
int delz[2];
int delyaw[2];
int delpitch[2];
int delroll[2];
int buttonstate[2];
int footpedal;
int checksum;
}masterToRobot_data;
```

The first three fields are a sequence number, a packet origination timestamp and a second timestamp field that may be used to measure network latency by setting the second timestamp and returning the packet to the sender [11]. The next three fields are position (delta or *del*), increments of X,Y and Z for the two arms, and fields seven through nine are joint angle increments for orientation control. Field 10 commands grasp on or off and could also activate other features, such as electrocautery, or ultrasound. The eleventh field is the status of the footpedal and the last field is the checksum, the integer sum of all other fields except the timestamp. Corrupt or out of sequence packets are ignored, since each position increment is small enough that losing one here or there will have no noticeable consequences, just as in video streaming.

The SGUI and the HDC communicate via a single bidirectional TCP. The data packets (which contain C/C++ structs in binary form) are sent at a far slower rate —event driven rather than clock driven— from SGUI to HDC, and at 10 Hz from HDC to SGUI, 100 times slower than communication to the robot. Also, given that each packet might contain a uniquely important command, a reliable connection oriented protocol is required and thus TCP is used instead of UDP.

The following C/C++ struct is sent from the SGUI to the HDC:

```
typedef struct{
unsigned int tick;
int flag01;
int UDPaddr;
int scale;
int checksum;
}SGUItoHDC_data;
```

The information sent from the SGUI to the HDC is a packet number, a set of flags including pedal and grasp state, the IP address of the surgical robot host, and the surgeon's motion scale factor. The checksum algorithm is similar to the one described above.

The following C/C++ struct is sent from the HDC to the SGUI:

```
typedef struct{
unsigned int tick;
int delx[2];
int dely[2];
int delz[2];
int runlevel;
int checksum;
}HDCtoSGUI_data;
```

This packet contains a packet number, the position of the Omnis (for information and debugging) and a description of the current state of the system [12]. Again, the checksum is similar to the one described above.

III. Video Feedback

The video feedback from the surgical robot operating either locally or at a remote location is completely decoupled from the rest of the SSS. This allows experimentation with any video transmission means available at both ends of the link. In local use, analog, digital or HD video can simply be displayed on a local monitor. When the system is used for remote surgery over an Internet link, we can use any of a rapidly expanding variety of video transmission systems. In selecting such systems we are looking for the following attributes:

- Video picture quality.
- Low encoding and decoding total latency.
- Robustness to network characteristics including lack of quality-of-service guarantees.
- Low cost and availability of the codecs and applications.

In our recent long-distance experiments we have explored four alternative video transmission systems. It will be important future work to systematically evaluate their relative performance, but at this point we can offer anecdotal experiences. The systems we have evaluated are:

1) Hardware-based high-performance video codec at both ends of the network link.
2) Hardware codec to encode the video signal and the VLC player media application, running on a standard laptop to display the video.
3) Skype video chat. (`www.skype.com`)
4) `Apple iChat.(www.apple.com/ichat)`

IV. Field Experiments

We have used this system for over 100 hours of experimental operation including about five hours of animal surgery, one hour of telesurgery between surgeons at Imperial College London and our lab, and two field deployments of RAVEN (seven hours and five hours). The remainder of the time has been local dry-lab testing in our laboratory in Seattle. In these experiments, substantial effort was required to prepare the hardware and make it work in the field. We are currently analyzing the data we were able to obtain but time and resources were not available in any of the three experiments to derive statistically rigorous performance benchmarks. Therefore, the experiences below are reported with the goal of documenting initial experiences under field conditions.

A. High Altitude Platforms for Mobile Robotic Telesurgery (HAPs/MRT)

The HAPs/MRT project was a collaboration with the University of Cinncinati, and HaiVision Systems Inc (Montral, Quebec) and AeroVironment Inc (Simi Valley, CA). The goal of this experiment was to explore the field deployment of mobile surgical robots and surgical master consoles [13]. For three consecutive days in June 2006 the portable surgeon console was set up under a tent in a semi-desert pastureland in Southern California. The RAVEN surgical robot was set up about 100 meters away in another tent. Portable generators powered both systems. The last-mile Internet connection was provided by a 3MB/sec wireless link supported by the AeroVironment PUMA unmanned aerial vehicle. Usable bandwidth for the prototype data link was limited to around 1MB/s with an average latency of 15ms. Field setup of the surgical robot was straightforward, however operation over limited bandwidth was challenging.

The restricted bandwidth pushed the envelope of two important teleoperation factors: video quality and sending rate to the robot. Only 200KB/s of bandwidth remained after video transmission so the sending rate of control packets was reduced from 1000Hz to 100Hz. There was no noticeable difference in performance or stability of the robot controls.

In this experiment, identification and adaptation to network conditions were identified as important capabilities for field operation of a networked surgical console.

One HaiVision Hai500 codec was used on each end of the Internet data link. An analog monitor was used to display surgical video. NTSC video from a Sony handicam was compressed to 800KB/s MPEG-2 and showed a good deal of pixelation and motion artifacts. In these conditions, video quality was sufficient for manipulation tasks and tying a suture. Latency was not carefully measured but was not noticeable to surgeon users.

B. London to Seattle

The first test of the master-slave system operating over standard Internet occurred in July 2006 in a transatlantic teleoperation between Imperial College, London and the University of Washington, Seattle [13]. The SSS was set up at Imperial College on pre-existing hardware. Two PHANToM "Premium" 6-DOF haptic devices were used. The RAVEN surgical robot was set up in the BioRobotics Laboratory at the University of Washington. Latency of the control packets was about 140ms. Two surgeons at Imperial College successfully conducted simulated surgical tasks.

This experiment compared surgical performance using video conferencing products from Skype and iChat. These products required Windows and Macintosh computers respectively. In both cases we selected the highest bandwidth setting in the setup menus of the application.

End to end video latency with both codecs was about one second. Video quality was noticeably lower than the hardware codec configurations and about the same for both Skype and iChat. Although the video quality was not excellent, it was sufficient with both codecs to expand the picture to full screen. A noticeable difference between the two was that sound appeared to be delayed in iChat in order to preserve synchronization with the video and this was useful in picking up and interpreting ambient sounds associated with RAVEN operation. For most of the experiments, iChat was used because of the better audio/video synchronization.

C. Local Animal Surgery

Ultimately, validation of a medical procedure or tool requires in-vivo testing. In March 2007 an experiment at the University of Washington Center for Videoendoscopic Surgery (CVES) tested the suitability of the RAVEN robot for MIS procedures, examined the current capabilities of the portable surgeon's console, and demonstrated the potential and limitations of the system. Animal surgery on a mature pig was

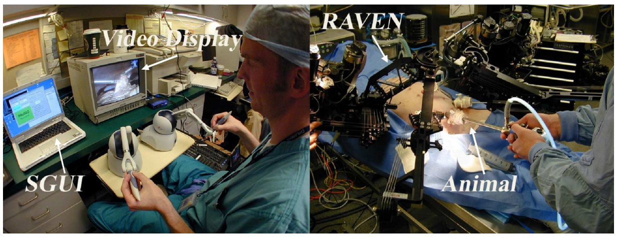

Fig. 6. Animal surgery showing the surgeon console on the left and the RAVEN operating on a mature pig on the right.

performed under a protocol approved by the University of Washington Institutional Review Board.

The surgeon's console with PHANToM Omni controls was installed in an office next door to the operating room and connected to the robot control computer via ethernet LAN (Fig. 6). Network latency was negligible (< 1 ms) and 100MB/s LAN bandwidth capacity was more than sufficient.

Experimental tasks were common laparoscopic procedures: running the bowel and tying sutures. Three surgeons participated in the animal surgery.

Results of this experiment brought to light engineering changes needed by the RAVEN. The surgical workstation's performance was deemed acceptable by the surgeons.

In this experiment, analog video was displayed to the surgeon directly using an S-Video cable and a standard CRT monitor, so no networked video codec was required or tested in this procedure.

D. NEEMO XII

The most recent test of the surgery system deployed the RAVEN surgical robot in the Aquarius undersea habitat, 19 meters below sea level and 5.6 km off the coast of Key Largo, Florida. In this experiment, conducted as part of the NASA Extreme Environment Mission Operations (NEEMO-XII) mission in May 2007 (http://www.nasa.gov/mission_pages/NEEMO/), the last 10 miles of the connection to the remote site were provided by a microwave link from shore to buoy and a short cable down to the habitat.

The RAVEN was controlled from three separate locations. Master consoles were set up in Seattle, at the NURC (National Undersea Research Center) shore base in Key Largo FL, and at the Cincinnati, OH, Museum Center. Operators at these locations tested other, non-surgical performances of the robot including simulated manipulation of moon rocks in a sterile environment. All locations used PHANToM Omnis as the input devices. Surgeons performed experimental benchmark tests drawn from the Fundamentals of Laparoscopic Surgery (FLS) test protocol [14], [15].

For surgery tasks in Seattle, two different video systems were used (HaiVision to VLC and iChat V.2.1.3 on Apple Macintosh). The HaiVision 1060 is a hardware video codec providing MPEG-4 AVC H.264 video compression and decompression. The HaiVision 1060 encoded the video stream, and a laptop PC running VLC media player (`http://www.videolan.org/vlc/`) displayed the video at the surgeon site. Picture quality was excellent at full laptop screen resolution. However, latency between Seattle and Florida was quite noticeable to users, on the order of one second. Internet round trip latency for the command packets was measured and is shown in Table I, was only about 76 ms so the majority of this time was due to video compression and decompression. The HaiVision video parameters for the NEEMO-12 experiment is shown in Table II.

TABLE I
NETWORK PARAMETERS FOR NEEMO-12 EXPERIMENT TO SEATTLE

Location	Roundtrip Mean (ms)	Standard Deviation (ms)
NURC	75.28	0.95
Aquarius	76.57	1.22

TABLE II
VIDEO PARAMETERS FOR NEEMO-12 EXPERIMENT TO SEATTLE

Parameters	Values
Encapsulation	H.264
Video Input	S-Video
GOP (group of pictures)	30
Framing Mode	IP
Interlacing	MBAFF
Resolution	Full D1 (720x480)
Video Bitrate	1-1.5MBs

V. ONGOING AND FUTURE WORK

The notion of a portable telerobotic surgical console is new and evolving. There is much work to do in making the system usable and effective for surgery with different surgical robots under varying conditions.

In addition, we consider that particularly:

- Communication security
- System reliability
- Redundant communication channels
- High quality video
- Stereo video

are among the features necessary for actual medical use that have not been currently pursued and will need to be addressed in any system contemplated for human use.

A. Interoperable Telesurgery Protocol (ITP)

An application layer protocol for teleoperative control of surgical robots is being developed in collaboration with SRI International, Palo Alto, CA. The ITP specifies not only the packet structure, byte order and port numbers for communication between master and slave, but also mutual reference frames, units and indexing procedures for true interoperability of telerobotic surgery systems. Using this common communication architecture, one master console will control different types of surgical robots in dispersed locations and one robot may be controlled by heterogeneous surgical masters. We are currently jointly specifying this protocol which is a basic extension of the one described in this paper. We expect to implement the protocol this year.

B. SGUI Improvements

Some additional features to improve the SGUI are:

- Ability to select the desired surgical tools for left/right arm from a list. This would require additional capacity of automated tool exchange on the robot side.
- Capacity to switch video sources easily, changing from endoscope view to an overhead view of the patient, surgical robot and operating room.
- Addition of haptic feedback. The current system has the capacity for haptic feedback, but this is not currently implemented.
- Changing teleoperative control between open-loop and feedback modes.
- Implementation of a better orientation mapping scheme instead of mapping orientation joint angle increments. One issue to be resolved is reduction of degrees of freedom between the Omni and a five Cartesian DOF robot.

C. End User Evaluation

A standard test of laparoscopic performance is the FLS assessment suite. We will continue to use this standard set of tasks to objectively evaluate this and other surgical technology [15].

VI. Conclusion

Robotic telesurgery is a new field with great potential to overhaul the way surgical care is delivered at home, in the field, and to remote or extreme environments. In this paper we have described a portable, flexible surgical master for control of such a system. Hardware and software considerations have been addressed, and our particular hardware and software solutions have been depicted.

We described field tests of the portable surgical master in extreme and challenging locations and the valuable insights gained from those tests. It was found that a low cost surgery master console built from off the shelf hardware running on common software and networking can facilitate collaboration and aid medical science. Furthermore, it was noted that dealing with sub-optimal network conditions is key to the success of field operation to/from a remote environment.

An exciting aspect of this project is the development of interoperability with different surgical robots through a standard protocol. Allowing different surgical technologies to work together opens the door to new collaborations between doctors. This will provide an easier adoption of telemedicine technology as well as improve education through telementoring and surgical care by allowing remote consultation and assistance for complex procedures. Additionally, collaboration will boost technological innovation by linking surgical robotic development efforts on a common platform. This, in turn, will facilitate teleoperation experiments, such as the one between London and Seattle described above, and will spur independent development of master and slave systems.

We predict that distribution and interoperability among networked surgical technologies will continue to increase collaboration among doctors and engineers alike. The result will ameliorate the methods and practice of telemedicine, and improve the delivery of surgical care to extreme and remote environments.

Acknowledgements

This work was supported by grants from the Peer Reviewed Medical Research Program US Army MRMC (DAMD17-02-1-0202), by the US Army Telemedicine and Advanced Technologies Research Center (TATRC), and by the University of Cincinnatti. We would like to thank Lance Boulet of HaiVision for assistance with their video codec, and our collaborators at AeroVironment. Special thanks to Tim Broderick M.D., Mika Sinanan, M.D. Ph.D., Thomas Lendvay, M.D., and Andrew Wright M.D., for invaluable advice and leadership. We also acknowledge fruitful collaboration with Thomas Low, Ph.D. of SRI International.

References

[1] G. Guthart and K. Salisbury, "The inituitive telesurgical system: Overview and application," in *Proceedings of the 2000 IEEE ICRA*, 2000, pp. 618–621.

[2] A. Madhani, G. Niemeyer, and J. Salisbury, J.K., "The black falcon: A teleoperated surgical instrument for minimally invasive surgery," in *Proceedings of the Intelligent Robots and Systems*, vol. 2, 1998, pp. 936–944.

[3] M. Cavusoglu, F. Tendick, M. Cohn, and S. Sastry, "A laparoscopic telesurgical workstation," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 4, pp. 728–739, August 1999.

[4] J. Marescaux, "Transatlantic robot-assisted telesurgery," *Nature*, vol. 413, Sept. 27.

[5] J. Marescaux, J. Leroy, F. Rubino, M. Smith, M. Vix, M. Simone, and D. Mutter, "Transcontinental robot-assisted remote telesurgery: Feasibility and potential applications advances in surgical technique," *Annals of Surgery*, vol. 235, no. 4, April 2002.

[6] M. Mitsuishi, J. Arata, K. Tanaka, M. M. amd T. Yoshidome, S. Iwata, M. Hashizume, and S. Warisawa, "Development of a remote minimally-invasive surgical system with operational environment transmission capability," in *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, vol. 2, 2003, pp. 2663–2670.

[7] J. Arata, H. Takahashi, P. Pitakwatchara, S. Warisawa, K. Konishi, K. Tanoue, S. Ieiri, S. Shimizu, N. Nakashima, K. Okamura, Y. S. Kim, S. M. Kim, J.-S. Hahm, M. Hashizume, and M. Mitsuishi, "A remote surgery experiment between japan-korea using the minimally invasive surgical system," in *Robotics and Automation, 2003. Proceedings. ICRA '06. IEEE International Conference on*, 2006, pp. 257–262.

[8] M. Lum, J. Rosen, M. Sinanan, and B. Hannaford, "Kinematic optimization of a spherical mechanism for a minimally invasive surgical robot." in *Proc. IEEE International Conference on Robotics and Automation*, New Orleans, LA, May 2004.

[9] M. Lum, D. Trimble, J. Rosen, K. Fodero, H. King, G. Sankaranarayanan, J. Dosher, R. Leushke, B. Martin-Anderson, M. Sinanan, and B. Hannaford, "Multidisciplinary approach for developing a new minimally invasive surgical robot system," in *Proceedings of the 2006 BioRob Conference*, Pisa, Italy, Feb. 2006.

[10] L. Stark, W. Kim, F. Tendick, B. Hannaford, S. Ellis, and et al., "Telerobotics; display, control, and communication problems," *IEEE Journal of Robotics and Automation*, vol. RA-3, pp. 67–74, 1987.

[11] G. Sankaranarayanan, L. Potter, and B. Hannaford, "Measurement and emulation of time varying packet delay with applications to networked haptic virtual environments," in *the proceedings of ROBOCOMM, the first International Conference on Robot Communication and Coordination*, Athens, Greece, 2007.

[12] K. Fodero, H. King, M. J. Lum, C. Bland, J. Rosen, M. Sinanan, and B. Hannaford, "Control system architecture for a minimally invasive surgical robot." in *Proceedings of Medicine Meets Virtual Reality*, Long Beach, CA, 2006, pp. 156–158.

[13] M. Lum, D. Friedman, H. King, R. Donlin, G. Sankaranarayanan, T. J. Broderick, M. Sinanan, and J. R. Hannaford, "Telesurgery utilizing a portable surgical robot and an unmanned aerial vehicle via wireless and wired communication." in *proceedings of Field and Service Robotics (FSR 2007)*, France, 2007.

[14] J. Peters, G. Fried, L. Swanstrom, N. Soper, L. Sillin, B. Schirmer, K. Hoffman, and the SAGES FLS Committee, "Development and validation of a comprehensive program of education and assessment of the basic fundamentals of laparoscopic surgery," *Surgery*, vol. 135, 2003.

[15] M. Lum, D. Friedman, G. Sankaranarayanan, H. King, T. Broderick, A. Wright, M. Sinanan, T. Lendway, J. Rosen, and B. Hannaford, "Objective assessment of tele surgical robot systems: Telerobotic fls." in *review for the Medicine Meets Virtual Reality(MMVR16) 2008 conference.*