

# Data Integration during Robotic Ultrasound-Guided Surgery

Members: Tiffany Yung, Vineeta Khatuja, Andrew Wang

Mentors: Theodoros Katsichtis, Colin Lea, Michael Choti MD MBA, Russell Taylor PhD

May 2013

## 1. Background

Laparoscopic procedures are most commonly performed in surgeries dealing with the abdomen, and computer-assisted laparoscopies in particular are increasingly used in contrast to the traditional procedure since the former allows for minimally invasive procedures in the abdominal cavity using a few small incisions, as opposed to one large incision in an open surgery. These laparoscopies can be augmented with ultrasonography which provides real-time imaging to update the preoperative data. Laparoscopic ultrasound can greatly improve the efficacy of many operations including increasing the accuracy of biopsies and ablations as well as scanning and detecting lesions on the liver, kidney and many other organs.

Real time ultrasonography is commonly used in modern open, laparoscopic and robotic surgery. Ultrasound data guide the surgeon's decision-making process intraoperatively when localizing a lesion and determining its position relative to other anatomical structures. A robotic platform offers the surgeon an array of advantages compared to conventional open and laparoscopic techniques. Stereoscopic 3D vision, seven degrees of motion, and physiological tremor attenuation allow the surgeon to be more precise and competent.

Currently, robotic surgeries are most often performed with the aid of the da Vinci robotic system. This system consists of a master and a slave side. The surgeon sits at the master-side console, which consists of a display, clutches, and pedals, with which the surgeon can view the operating field at the slave side and manipulate the tools in the da Vinci robot's slave-side arms.

The da Vinci console display shows the live, stereoscopic camera feed from the camera on the slave side. TilePro, a da Vinci system feature allows an interface to be projected onto the da Vinci console display in a separate window at the bottom, thus splitting the display into two halves. The top half displays the stereoscopic camera feed and the lower half displays the projected interface in the TilePro. The size of the projected interface can be adjusted using the da Vinci clutch pedals at the console, while the mouse in the projected interface can be controlled using the da Vinci tool manipulators. TilePro can be used to add more complicated interfaces capable of displaying multiple images, video feeds, etc. It is used in the laparoscopic surgeries to display real-time ultrasound images from the ultrasound machine in the operating room. TilePro can also be used to hold other information and saved images, for example, preoperative studies.

## 2. Problem

While the da Vinci platform improves surgeon precision and control, ultrasound images along with other patient imaging data are still displayed and processed in a different unit external to the da Vinci master console inside the operating room. Thus, the surgeon must view multiple monitors and requires assistance from ultrasound technicians to process the ultrasound data. This impairment of surgeon autonomy increases the cost and duration of operations while decreasing their efficiency and ergonomics. In addition, anything that distracts the surgeon from the procedure or necessitates him to re-direct his attention away from the immediate operating field increases the risk and danger to the patient.

Therefore, an interactive interface that could be incorporated into the da Vinci console display along with the live da Vinci camera feed itself would be of great importance and usefulness to robotic surgical procedures that require the surgeon to divide his attention between multiple sources of information. In particular, such an enhanced environment interface would greatly help with laparoscopic surgeries since the surgeon must take notice of both the da Vinci camera feed and the real-time ultrasound feed during such procedures, as well as review the patient's preoperative scans. An interface compiling and displaying all of this information in one location would greatly simplify the task for the surgeon.

A novel da Vinci interface must be developed which integrates and displays live, intraoperative ultrasound as well as additional preoperative data (CT, MRI, X-ray and DICOM) to the surgeon. Displaying all of the necessary data to the

surgeon through the da Vinci master console would increase the efficacy of robotic procedures improving patient outcome.

### 3. Method

The interface was designed to allow the maximum, but user-friendly, display of data within the available space in the console display while preserving the aspect ratio of the videos and images to prevent image distortion. The layout was kept simple to allow for easier incorporation of new features later on, as well as to avoid diverting the surgeon's attention with flashy widgets. A dark-toned color scheme was also chosen for this reason.

The interface was implemented so that events are triggered by sliding the mouse over the desired tabs and images, rather than using a mouse click, as mentioned later. This was implemented in order to avoid fatiguing the surgeon's further during the procedure.

The interface described below is contained in the bottom half of the da Vinci console display inside the TilePro; the top half is reserved for the da Vinci camera feed. On the right side of the interface is an image browser used to view saved intraoperative ultrasound and stereo images, as well as preoperative images. The left side holds a video viewer for the live ultrasound feed, in addition to two image viewers to allow the surgeon to view a saved ultrasound image with its corresponding stereo image taken at the same time. Alternatively, this left side of the interface can display an enlarged preoperative image. The former layout appears if the surgeon choose to view an intraoperatively-saved ultrasound; the latter takes its place if the surgeon instead chooses a preoperative image to view. All of these features (Figure 1) are described in detail under the Software Approach section.

The project was divided into two aspects: the software aspect focused on developing the interface, while the clinical aspect focused on planning and preparations for clinical testing of the interface, as well as on the nonclinical testing of the interface in the mock OR.



Figure 1: The view from the surgeon console . The da Vinci camera feed appears at the top half. The bottom half holds the developed interface in the TilePro with all integrated data. Image browser is displayed on the right, and the live ultrasound video viewer with the two above-described image viewers.

## 4. Software Approach

### 4.1. QT framework

The interface is built using the QT application framework, a cross platform application framework that is widely used for developing graphical user interfaces of application software. QT uses a layout-based approach to widget management, in which widgets are arranged in the optimal positions in windows based on simple layout rules, leading to a consistent and organized look and feel.

QT provides a signal and slot mechanism to communicate between objects. A signal, emitted when a particular event occurs, triggers a slot, which is a function that is called in response to a particular signal. QT widgets have many predefined signals, but widget subclasses can always be made in order to add custom signals to them. Pre-defined slots can be used, or new slots can be defined.

#### 4.2. Other software dependencies

The Computer-Integrated Surgical Systems and Technology (cisst) library was used to communicate between computer processing and hardware activity. Access to the da Vinci Intuitive application programming interface (API) was also obtained in order to allow access to and modify the da Vinci console display.

#### 4.3. Masters as Mice

The da Vinci tool manipulators at the master console were chosen as the mouse controller to be used in the interface since the surgeon would not have to leave the console or look away, as would be the case if the laptop mouse were to be used instead. Synthetic QT mouse events are generated by the movement and clicking of the Master Right Tool Manipulator (MRTM), the right-hand side tool manipulator at the master's side (Figure 2). These synthetic events are then posted to the main QT application and handled accordingly by the program, thus triggering events in the interface such as a change in tab or image selection.



Figure 2: da Vinci right-hand side tool manipulator at the master's console. Tool manipulators allow the surgeon to control the tools held by the da Vinci's arms at the slave side.

The interface was designed such that pressing and holding the left clutch pedal at the master console and then pinching both tool manipulators simultaneously triggers the Masters as Mice mode, thus activating the mouse within the interface. Masters as Mice is released by pinching both manipulators simultaneously again. This feature was implemented using a callback mechanism. Callback functions are called when da Vinci events like putting the head into the console, taking the head out, and clicking the right tool manipulator occur and are detected by the da Vinci system. The MRTM move callback retrieves the 3D coordinates of the MRTM tool tip position. These points are projected onto a 2D plane by discarding the z-coordinates to get a 2D point position. A QT signal is then emitted, which is connected to a QT slot which creates a new QMouseMove event and updates the cursor position.

The common press-and-release clicking model used by computers for mouse control is simulated using the pinch-and-release model via the right tool manipulator. Thus, clicking in the interface can be accomplished by pinching the right tool manipulator. The MRTM pinch callback emits a QT signal which is connected to a QT slot that creates a new QMouseClicked event and posts the slot to the application.

It was also essential to have an intuitive and easy scroll option. Therefore, the scrolling method was designed to be similar to the kinetic scrolling feature seen in mobile phones where tapping, dragging, and releasing is used to scroll. Similarly in the interface, the pinch-drag-release model is used, where pinching is achieved by clicking the da Vinci right tool manipulator. The scrollbar is implemented using the QScrollBar. The value field of the QScrollBar is responsible for which part of the scrollbar is currently visible. The 2D position of the mouse cursor in the interface is noted at the time of pinch and release, and the scroll value is incremented by taking the difference between the y-coordinates of the two positions.

#### **4.4. Image displays**

A button was implemented in the interface to allow the surgeon to save images; this button is triggered by the use of the clicking event as explained above. Saved images are then stored in the image browser placed on the right side of the interface, which contains two tabs to display to categories of images: intraoperatively-saved ultrasound images and their corresponding da Vinci stereo images, and preoperative patient scans.

Scrolling vertically through the image browser is implemented in the same method as mentioned above, while hovering the mouse cursor over the images in the image browser changes the current selected image in the image viewer. The browser is implemented using QTabWidget objects, one for each tab.

In order to allow the surgeon to view these images, image viewers were also implemented to the left of the browser. When the surgeon selects a preoperative image to view, an image viewer appears on the left side of the interface, taking up all of the space not used by the browser. When an intraoperative saved ultrasound image is selected instead, the left-side region of the interface is shared between the live ultrasound display at the top, and two image viewers showing the saved ultrasound and stereo images. These features were implemented using QWidget and QLabel objects, the latter of which is used to display the images and the former being a general widget container.

#### **4.5. Video displays**

A video viewer, which is used to display the live ultrasound feed mentioned above, was implemented and placed on the left of the interface using a QWidget as the medium with which to display it. Also as mentioned above, the video viewer is only visible while viewing the intraoperatively-saved ultrasound images. The cisst SVL library is used to capture the video and display it on the QWidget.

### **5. Clinical Approach**

The clinical aspect of the project focuses on testing the software for bugs and glitches, and preparing the software for testing in a clinical environment by surgeons on actual patients undergoing a surgical procedure. In order to prepare the interface to be deployed in a hospital OR for use on patients, the software first had to be tested in a nonclinical setting to check that the software would not glitch, crash, or perform unexpectedly during use. Thus, the clinical portion of this project was divided into two phases: preparation and deployment of the interface in an imitation environment, and clinical testing in the real hospital setting.

#### **5.1. Hardware setup**

In order to project the new interface onto the da Vinci console's display in the operating room, the laptop computer on which the interface was developed had to be connected to the da Vinci robot itself. An HDMI cable was used to connect the development laptop to the da Vinci so that the laptop could project its real-time desktop image to the da Vinci console.

Video grabbers were used to connect the laptop to both the da Vinci machine and the ultrasound machine present. This allowed the laptop to receive the corresponding live video feeds from these two sources and to incorporate them into their respective widgets in the interface, which in turn was projected from the laptop to the TilePro interface in the da Vinci console by the HDMI cable mentioned above.

#### **5.2. Nonclinical testing and preparations**

##### **5.2.1. Equipment**

Testing the interface required several hardware dependencies to be met, including access to a da Vinci robot, ultrasound machine, video grabbers, and the laptop used for the interface development. In order to perform the testing, access to mock OR on the Johns Hopkins Homewood campus was granted. The mock OR is equipped with a da Vinci console and ultrasound machine (Figure 3), and the team was able to use video grabbers (Figure 4) connected to the development laptop to share video feed between the da Vinci robot and the laptop. One video grabber was used to receive video feed from the da Vinci camera tool, while the other was used to receive real-time ultrasound images from the ultrasound

machine. An ultrasound drop-in probe was obtained from Dr. Choti to be used with the ultrasound machine in the mock OR.



Figure 3: Mock OR testing setup. In the forefront is the ultrasound machine used to grab live ultrasound feed from the probe and transmit it to the laptop and the TilePro interface. Left and in back is the master's console, and to the right is the slave side complete with the operating table and equipped da Vinci robot arms.

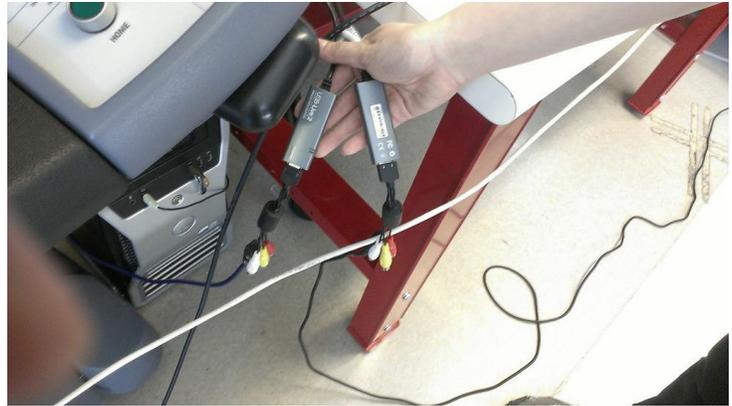


Figure 4: Two video grabbers. Both are connected to the laptop controlling TilePro. One each go to the da Vinci and the ultrasound machine.

With the access to the hardware acquired, the team sought to design a phantom that would mimic the physiological properties of the liver and contain fake lesions that would appear tumor-like when imaged by ultrasound. The formula used was made from powdered collagen obtained from pig skin. Water was heated to between 70 and 80 degrees Celsius, and the appropriate amount of collagen powder was added. The solution was stirred and poured into a plastic container to cool, whereupon the solution solidified into a solid block filling about half of the container. Three chunks of the gelatin were scooped out to make room for the pseudolesions; the recipe used for these fake lesions is described below. After the pseudolesion solution was poured and set into the concavities in the gelatin, another batch of gelatin solution for the "normal" tissue of the liver phantom was made and poured to the top of the container to cool and solidify. This completed the creation of the liver phantom.

Two formulas were found for mimicking lesions: one consisting of cellulose and water, and another consisting of agarose, glycerol, and water. Of the three lesions in the final phantom, one was made from the former solution and two from the latter (Figure 5).

The phantom was cast approximately three days before its use in nonclinical testing in the mock OR, so that it would had time to solidify, but would not have dried out by the day of the testing. The phantom was transported from the medical campus to the refrigerator in the Homewood campus mock OR.



Figure 5: Gelatin phantom. An white agarose/glycerol pseudolesion is visible at the center of the phantom.

### 5.2.2. Testing plan

Nonclinical testing was carried out in the mock OR by setting up the equipment and connecting the development laptop to the da Vinci to project the TilePro interface to the bottom half of the da Vinci console display. Video grabbers were also run from the laptop to both the da Vinci robot and the ultrasound machine in order to grab the stereo and ultrasound images from them to display in the interface.

The gelatin phantom was removed from its plastic container and placed on the surgical table at the slave side of the da Vinci machine (Figure 6). The ultrasound drop-in probe was placed in one of the da Vinci slave side's tool holders, connected to the ultrasound machine, and the gel was applied to the probe. The probe was placed directly over and in contact with the phantom.

Team members then took turns controlling the da Vinci. Everyone else observed the projection of the da Vinci console display on the television screen to look for possible design flaws, glitches, and improvements that could be made to color and arrangement of the widgets.

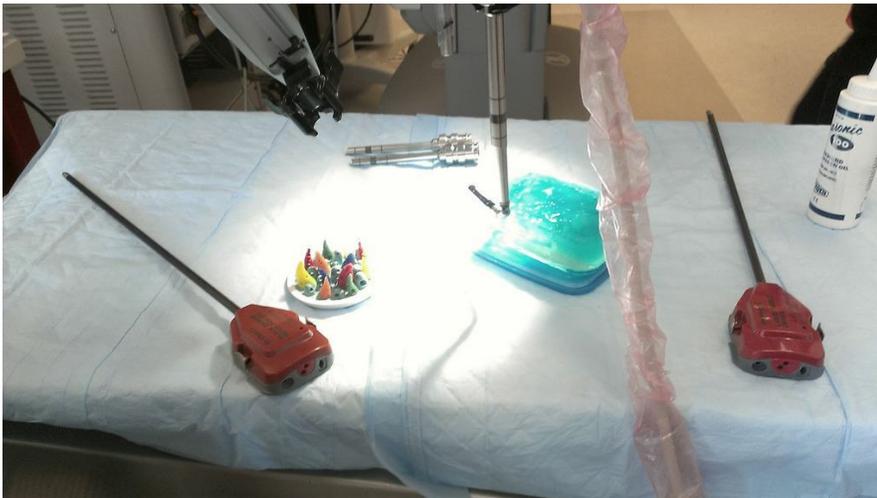


Figure 6: Mock OR operating table setup. The blue block is the gelatin phantom, and the reversed-L shaped object is the ultrasound probe.

### 5.3. Clinical testing

Since testing of the interface by surgeons at the hospital involved actual patients, an IRB proposal had to be submitted and approved before testing could commence. A draft written by Theo prior to the beginning of the CIS 2 project was used as the basis for the proposal. Although the actual IRB proposal cannot be released prior to its full approval, the general outline of the proposed procedure involves deploying the interface on the da Vinci machines at the hospital and asking surgeons to use the interface during a procedure involving a real patient. The surgeon would then fill out a questionnaire that would be scored using the Likert system, where each participant would rank different aspects of the interface from 1 to 5 with 1 being poor and 5 being satisfactory.

The team also required surgical collaborators at the hospital who were interested and willing to participate in the study and give feedback about the interface. This was accomplished by establishing contact with Dr. Choti, who asked amongst his colleagues to determine those that showed interest and enthusiasm for such an endeavor.

## 6. Results

### 6.1. The interface

The current baseline interface (Figure 1) developed from this semester-long project features a TilePro interface based in a black background to enhance the clarity of the stereo and ultrasound video feeds and to make them stand out from the background. The da Vinci console displays the video feed from the da Vinci slave-side camera at the top of the interface, which is not connected to the TilePro interface. Thus, if the interface were to crash, the da Vinci camera feed would remain, and patient safety would not be compromised. In the bottom half is the interface. The surgeon can transition from controlling the da Vinci end-effectors tools at the slave side to controlling the mouse in the TilePro interface by pressing and holding the clutch pedal of the master side.

The interface consists of an image browser on the right and the feed from the real-time ultrasound on the left. Below the live ultrasound, the surgeon can also view a saved ultrasound image and its associated da Vinci camera image. This left side of the interface can change depending on which tab in the image browser the surgeon has selected, as detailed below.

The image browser contains two tabs, one for viewing ultrasound images saved by the surgeon intraoperatively and another for viewing preoperative scans uploaded to the system. The surgeon can access each tab by sliding the mouse over the desired tab. To display an ultrasound image saved intraoperatively, the surgeon can slide the mouse over the desired image, which displays that ultrasound image, accompanied by the stereo image from the da Vinci taken at the same time, to the left of the image browser and below the live ultrasound display.

The surgeon can also view an enlarged preoperative image by sliding the mouse over the preoperative tab and then over the desired image, which displays the image in the space to the left of the browser. While viewing an intraoperatively saved ultrasound and its accompanying stereo image preserves the real-time feed from the ultrasound (Figure 1, bottom half), sliding over and viewing a preoperative image causes the enlarged preoperative image to take over the entire space to the left of the browser (Figure 7). The interface was implemented in this way in order to make the preoperative scans clearer, more detailed, and easier for the surgeon to view.



Figure 7: Same interface, but with a preoperative image selected, instead of an intraoperatively-saved ultrasound image as in Figure 1.

## 6.2. Nonclinical testing

### 6.2.1. The setup

Preparations for the nonclinical testing were successful. All of the necessary hardware, particularly the drop-in probe and gelatin liver phantom, had been obtained by the day of the testing. When viewed on the ultrasound, it was apparent that both of the formulas for mimicking lesions were successful; all three lesions showed up clearly on the real-time ultrasound image as the probe passed over them.

### 6.2.2. The interface

Nonclinical testing revealed that the da Vinci end-effectors continued to control the mouse in the TilePro even after the clutch pedal was released. Since this could cause problems such as inadvertently saving images while the surgeon manipulated the end-effectors during surgery, this glitch had to be remedied. Work on it is currently in progress, as of the writing of this report.

It was also observed that the interface should not change to a certain tab when the mouse rolled over the given tab; rather, the Masters as Mice implementation ought to wait for the surgeon to clutch the end-effectors while the mouse is over the tab, so that the surgeon does not accidentally slide over a tab and lose an ultrasound image he wanted to save. Work on this is also currently in progress.

### **6.3. Clinical testing**

The IRB proposal required for clinical testing on patients at the hospital has been approved by the IRB with minor administrative changes. As of the writing of this report, the requested changes have been made and re-submitted to the IRB. The team is currently waiting for the full approval of the proposal to come back from the IRB. No trials for the clinical testing was performed during the semester but will occur over the summer.

## **7. Significance**

The implementation of the TilePro interface allows for real-time intraoperative ultrasound feedback with dual-image saving capabilities and image recall from within the da Vinci console. Usage of this interface would significantly increase the amount of data immediately available to the surgeon during a procedure. In addition, this interface would also allow the surgeon to collect more data and images during the surgery that he could refer back to later, either intraoperatively or postoperatively. The dual-image saving feature would also allow the surgeon to save and view both the da Vinci camera image and the ultrasound image at the same point in time. Thus, there is no delay between taking the one image and taking the other; therefore, the surgeon can compare these to notice anatomical details in one image that he didn't notice in the other without worrying that the images might not be exact images of each other.

Also, by eliminating the need for the surgeon to leave the console during the operation, the developed interface improves the efficiency of laparoscopic surgeries while increasing surgeon autonomy. Improving surgeon autonomy allows for the surgeon to quickly gather any information needed to proceed with the operation. As the surgeon is the most qualified individual in the operating room to gather and analyze information, the surgeon would benefit greatly by minimizing the relegation of image collection and interpretation. Through the use of this interface, the surgeon would have all of the required images at his fingertips for immediate access, allowing him to cross-reference preoperative scans with what he sees on the da Vinci stereoscopic and additional ultrasound feeds.

Additionally, allowing the surgeon to remain at the console throughout the entire procedure improves patient safety and reduces the risk of the surgeon committing an error due to distractions. The surgeon need not divert his attention or gaze from the operative field, and the images and data are available to him for immediate viewing, allowing him to make quick decisions during the procedure.

Because the implementation of the interface is operated entirely on an external computing unit, any failure of the program or the laptop will not impede the normal operation of the da Vinci console. Thus, the surgeon always has the option to revert to the classic setup which requires the surgeon to leave the console to view ultrasound and preoperative data. In this way, the patient would also not be put at risk by the use of this interface. Should the interface fail, the surgeon would still be able to continue the surgery and would retain a view of the operative field. The interface simply allows the surgeon to hold much more information for immediate use without compromising the operation of the original da Vinci robot.

## **8. Management Summary**

### **8.1. Division of tasks**

As preparations for the project began, Vineeta worked on obtaining a compatible laptop and installing the necessary software components for interface development. Colin, Theo, and Dr. Taylor helped the team to obtain a laptop that would be compatible with all of the software needed. Anton Deguet and Balazs Vagvolgyi helped in resolving the software dependencies. Meanwhile, Tiffany compiled a budget proposal to obtain funds from Dr. Taylor to be used to pay for the cost of using the da Vinci robot in the mock OR for debugging and testing the interface.

Also, upon discussion with the surgeon when Vineeta and Theo visited the hospital to view a live procedure, it was decided that the deliverables list had to be re-prioritized, and the preoperative image viewer was moved to the minimum deliverables.

Once the project was underway, with the software installed and the computer ready for development, Vineeta focused on the software aspect of the project. She came up with the interface design and implemented it from scratch. Throughout the development process, she kept in contact with the team's mentors, as well as her two teammates, and presented the interface prototypes for feedback. As Vineeta coded the interface design, Andrew, Tiffany, Colin, Theo, and Dr. Choti all provided suggestions and feedback about coloring, arrangement of the widgets, and possible improvements in how the surgeon could better interpret and manipulate the interface. With help from snippets from Colin's code, Vineeta implemented the interface to allow the da Vinci right tool manipulator to be the mouse.

Andrew and Tiffany focused on making the preparations for nonclinical and clinical testing of the interface, as well as staying in contact with Dr. Choti through Colin and Theo. Through their mentors, Andrew and Tiffany developed the idea of splitting interface testing into nonclinical and clinical phases in order to minimize the amount of time, and thus the cost of, using the da Vinci robot to check for bugs and glitches in the interface. They also developed the specifics for casting a gelatin liver phantom with pseudolesions with which to test the baseline interface in the mock OR without the need to bring in surgeons to test a possibly faulty interface. In particular, Theo obtained the collagen gelatin needed for the "normal" tissue in the phantom and cellulose for the fake lesions; Andrew found a recipe to mimic tumors using agarose and glycerol, and also managed to obtain the materials to do so.

Working with Theo and Dr. Choti, Andrew and Tiffany also helped to edit the IRB proposal and procedure required to conduct testing of the interface on surgeons in a clinical setting operating on actual patients. Theo and Dr. Choti had previously drafted an IRB proposal, which was further developed and finally submitted to the IRB. For the nonclinical testing, Tiffany, Vineeta, Colin, and Theo were present in the mock OR to help with the setup, testing and completion of the procedure.

The team also worked together on deciding the contents of the presentations, while Tiffany and Andrew also organized and designed the presentation, in addition to writing the bulk of the report, allowing Vineeta to focus on supplying technical information, images, and updates from the software aspect of the project.

## **8.2. What was actually accomplished**

The original plan was to enhance and reorganize Colin's TilePro interface to make it more user-friendly, take out the incompletely-implemented 3D lesion mapping tool, and to rearrange and recolor the interface so as not to inadvertently draw the surgeon's attention away from the operative field with brightly-colored or oddly shaped or placed widgets. However, it was found that development would proceed much more quickly and easily if the interface were to be implemented from scratch using snippets from the original code as a guide.

The team had also planned to implement and add various tools to the interface, such as building and displaying 3D models of anatomical structures using preoperative CT scans and providing the surgeon with the ability to manipulate these structures to view them at different angles. However, this was not accomplished due to multiple setbacks with regards to software installation and compatibility, including being unable for a week to obtain administrative rights to install software on the development laptop. It was ultimately decided between the mentors and team members that the project and the interface as a whole would benefit more from having a properly developed, robust, and usable baseline interface to having an incompletely-implemented interface with advanced tool features.

Clinical testing had also been planned; however, due to the late submission and approval of the IRB proposal, this was not possible to achieve during the semester. As of the writing of this report, the IRB has approved the proposal with minor administrative changes, and the team is waiting for the proposal to come back completely approved. After this semester, the project must continue with the clinical testing of the interface in the clinical setting at the hospital. Andrew and Tiffany will be present over the summer and plan to oversee the deployment of the interface in these clinical trials.

## **8.3. Future work**

Future work on the interface can also involve implementing the advanced tools, such as 3D organ model building and an ultrasound measurement tool to measure distances within the ultrasound image, that could not be accomplished during this semester-long project. Additionally, in order to further improve surgeon autonomy, it may be beneficial to develop a modality which allows for the surgeon to control the ultrasound machine from the da Vinci console. Although the surgeon controls the movement of the ultrasound probe, he still depends on a technician to adjust other parameters on

the ultrasound machine, thus limiting his autonomy and ability to act quickly and immediately. Robotic positioning of the ultrasound probe may be superior to that of a technician in both accuracy and deviations from hand tremor.

#### **8.4. Lessons learned**

One of the most important lessons the team received from this project was on how frequently setbacks occur in a project and how to deal with them as they occur. Although seemingly simple, our greatest setback was the software installation. Due to the incompatibility of the team members' personal computers, the resulting search for a compatible computer that could be borrowed for development, and the inability to obtain administrative installation rights to the borrowed computer, our project timeline was set back by three weeks. However, the team learned to work around these roadblocks, remaining patient and focused on the clinical aspect of the project while sorting out the problems in the software aspect. Thus, the clinical aspect would be ready once the problems in the software aspect of the project were solved. In short, the team learned not to get hung up or discouraged by an obstacle in the project, instead using that time to prepare the project to run more smoothly once the obstacle was overcome.

#### **9. Acknowledgements**

Team 9 of the CIS 2 course would like to thank their mentors Colin Lea, Theodoros Katsichtis, Dr. Russell Taylor, and Dr. Michael Choti for their advice and guidance on the clinical and technical aspects of this project. We would also like to thank Anton Deguet and Balazs Vagvolgyi for their help in resolving the software dependencies.

#### **10. References**

- Bartosz F. Kaczmarek, S. S., Firas Petros, Quoc-Dien Trinh, Navneet Mander, Roger Chen, Mani Menon, Craig G. Rogers (2012). "Robotic ultrasound probe for tumor identification in robotic partial nephrectomy: Initial series and outcomes." International Journal of Urology.
- Caitlin M. Schneider, B. P. D. P., MD; Russell H. Taylor, PhD; Gregory W. Dachs II, MS; Christopher J. Hasser, PhD; Simon P. DiMaio, PhD; Michael A. Choti, MD, MBA, FACS Surgical Technique: Robot-assisted laparoscopic ultrasonography for hepatic surgery.
- Caitlin M. Schneider, G. W. D. I., Christopher J. Hasser, Michael A. Choti, Simon P. DiMaio, Russell H. Taylor Robot-Assisted Laparoscopic Ultrasound, Johns Hopkins University; Johns Hopkins Medicine; Intuitive Surgical, Inc.
- Craig G. Rogers, M. R. L., MD; Akshay Bhandari, MD; Louis Spencer Krane, MD; Daniel Eun, MD; Manish N. Patel, MD; Ronald Boris, MD; Alok Shrivastava, MD; Mani Menon, MD (2009). "Maximizing Console Surgeon Independence during Robot-Assisted Renal Surgery by Using the Fourth Arm and TilePro." Journal of Endourolgy 23(1): 115-121.
- Francesco Volonté, N. C. B., François Pugin, Joël Spaltenstein, Boris Schiltz, Minoa Jung, Monika Hagen, Osman Ratib, Philippe Morel (2012). "Augmented reality to the rescue of the minimally invasive surgeon. The usefulness of the interposition of stereoscopic images in the Da Vinci robotic console." The International Journal of Medical Robotics and Computer Assisted Surgery.
- Joshua Leven, D. B., Rajesh Kumar, Gary Zhang, Steve Blumenkranz, Xiangtian (Donald) Dai, Mike Awad, Gregory D. Hager, Mike Marohn, Mike Choti, Chris Hasser, Russell H. Taylor DaVinci Canvas: A Telerobotic Surgical System with Integrated, Robot-Assisted, Laparoscopic Ultrasound Capability, The Johns Hopkins University; Intuitive Surgical, Inc.

#### **11. Appendices**

##### **11.1. LapUS interface code**

The LapUS code written to generate the interface can be found on the project wiki page at <https://ciis.lcsr.jhu.edu/dokuwiki/doku.php?id=courses:446:2013:446-2013-09:446-2013-09>.

## 11.2. User's manual

### 11.2.1. Dependencies to compilation of the code

A number of dependencies must be resolved before the code can be compiled; these dependencies must be obtained in the proper order.

#### *The da Vinci API*

- Obtain the rights to install the da Vinci API by signing a Non-Disclosure Agreement.
- See <https://intranet.lcsr.jhu.edu/JHDFS> for instructions on how to mount a JHDFS.
- Go to /Software and browse to find isi\_api...exe and install it.

#### *QT*

- Download and install QT version 4.3 from the Internet.

#### *cisst*

- Download from <https://trac.lcsr.jhu.edu/cisst/wiki/DownloadAndInstallationFAQ>.
- Follow the instructions for Windows.
- Make sure the SVL\_HAS\_DIRECTSHOW option is checked. If not, set the path manually.
- Check the option to install saw.
- Check the cisstsawDaVinci option to build the saw component for the da Vinci API.
- Check the BUILD\_EXAMPLES option.
- After installing cisst, run the following examples to test it
  - svlCameraViewer - Connect the video grabber and run the example, choosing S-video as the video type if using the da Vinci feed. Everything is correct if the video plays.
  - cisstsawDaVinci - Connect the da Vinci signal to the computer using a LAN and attempt to move the tools around. Everything is correct if the value displayed on screen changes.
  - Visual Studio 2008 - the da Vinci API is only compatible with VS2008, else the code will not link.

### 11.2.2. Launching the code

#### *Compiling the code*

- Download the LapUS source code.
- Make a directory called "build" inside the LapUS directory.
- Run CMAKE, using the path of the LapUS directory as the source and the path of the newly-built directory as the build. Choose the "Compile for VS2008" option.
- Open the VS2008 file made from running CMAKE in VS2008.
- Choose the release mode and build.

#### *Preparing to run the code*

- Connect the two video grabbers and plug in the video signal cables.
- Connect the laptop to the da Vinci using an HDMI cable.

#### *Launching the code*

- Find the release folder generated in the build directory after building the project.
- Create a folder called "preop" and add the preoperative images here.
- Create a folder called "savedImages".
- Place the live.dat and US.dat files attached with the LapUS code in the release folder.
- Launch LapUS.exe.
- The interface is now ready to use on the da Vinci console!

### 11.2.3. Using the interface

*Displaying the interface in the da Vinci console* - Although the TilePro partition should appear in the bottom half of the console display by default, if this is not the case, pressing the camera pedal twice will display the TilePro partition.

*Activating Masters as Mice mode* - Once the interface has been launched and is projected onto the console display, the da Vinci tool manipulators must be used to enter the Masters as Mice control mode. This can be done by pressing and holding the left clutch pedal, inserting the thumb and index finger of each hand into each manipulator, and pinching them simultaneously. Masters as Mice will now be active, and moving the right tool manipulator moves the mouse in the interface. Pinch the right manipulator simulates clicking in the interface. If the clutch is released, the manipulator will no longer be able to control the mouse. To exit Masters as Mice, simply pinch both manipulators simultaneously again.

*Image browser navigation* - Navigation between the two tabs of the image browser, as shown in Figure 1, can be accomplished by sliding the mouse over one of the tabs to select that tab. To scroll vertically in the browser, pinch the right tool manipulator and slide it up and down. An image can be selected to be viewed by sliding over it, just like with the tabs.

*Saving and viewing images* - When a preoperative image is not being viewed, the live ultrasound feed can be seen to the left of the image browser. Next to it is a button labeled "Save" that, when clicked, will save an image of the current ultrasound, as well as the current da Vinci camera image. These images are saved under the "Ultrasound" tab of the image browser and can be viewed by navigating to that tab and to the desired image. Saved images appear in chronological order and are labeled numerically, where "1" indicates the first image saved. Preoperative images can be viewed in a similar manner by navigating to the "Preoperative" tab in the image browser. Note that the live ultrasound is not visible when viewing a preoperative image, since the image is enlarged to display greater detail.