

Article

Remote Presence: Development and Usability Evaluation of a Head-Mounted Display for Camera Control on the da Vinci Surgical System

Tareq Dardona, Shahab Eslamian, Luke A. Reisner and Abhilash Pandya *

Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202, USA; tariq.dardona@gmail.com (T.D.); shahab.eslamian@wayne.edu (S.E.); lreisner@wayne.edu (L.A.R.)

* Correspondence: apandya@ece.eng.wayne.edu

Received: 26 February 2019; Accepted: 16 April 2019; Published: 19 April 2019



Abstract: This paper describes the development of a new method to control the camera arm of a surgical robot and create a better sense of remote presence for the surgeon. The current surgical systems are entirely controlled by the surgeon, using hand controllers and foot pedals to manipulate either the instrument or the camera arms. The surgeon must pause the operation to move the camera arm to obtain a desired view and then resume the operation. The camera and tools cannot be moved simultaneously, leading to interrupted and unnatural movements. These interruptions can lead to medical errors and extended operation times. In our system, the surgeon controls the camera arm by his natural head movements while being immersed in a 3D-stereo view of the scene with a head-mounted display (HMD). The novel approach enables the camera arm to be maneuvered based on sensors of the HMD. We implemented this method on a da Vinci Standard Surgical System using the HTC Vive headset along with the Unity engine and the Robot Operating System framework. This paper includes the result of a subjective six-participant usability study that compares the workload of the traditional clutched camera control method against the HMD-based control. Initial results indicate that the system is usable, stable, and has a lower physical and mental workload when using the HMD control method.

Keywords: robotic surgery; head-mounted display; laparoscopic surgery; robotic camera control; da Vinci Surgical System

1. Introduction/Motivation

Robotic surgery was introduced to overcome some limitations of traditional laparoscopic surgery and bring a new era with the advent of better 3D visualization, motion filtering/scaling, and flexible instruments [1]. The most advanced, leading robotic surgical system is the da Vinci Surgical System. It is designed to facilitate complex surgery using minimally invasive approaches. The da Vinci Surgical System is fully controlled by a surgeon using hand controllers and foot clutches. It consists of three main components: the surgeon console, patient-side cart, and an instrument tower (Figure 1a). The main purpose of da Vinci is to improve operative technique by allowing the surgeon to operate with enhanced vision, control, and precision. The high-resolution camera view provides immersive stereoscopic vision that improves depth perception and has been shown to improve surgery outcomes [2]. However, the operation of the da Vinci system can still be cumbersome.

The current da Vinci systems have cumbersome camera controls. Using foot pedals and hand controllers, the surgeon must alternate between manipulating either the camera arm or the instrument arms. Therefore, moving the camera arm requires an interruption in the flow of the surgery and may cause medical errors [3]. It may also cause surgeons to choose suboptimal/unnatural views, which

could also lead to medical errors. Having an immersive system that allows simultaneous control of both the tools and the camera without adding to the already heavy mental workload of surgery could improve surgical operation times and outcomes. A review of surgical camera control methods is given in [4].

To enable researchers to work directly on the da Vinci system, an open hardware and software system, the da Vinci Research Kit (DVRK), has been developed [5]. Using the DVRK, researchers have full access to read and control the robotic arms of the da Vinci standard system. For instance, researchers at Johns Hopkins University developed a new method to implement haptic feedback in teleoperated robot-assisted surgery to enhance the surgeon's sensation [6]. Yamamoto et al. also developed an approach to integrate graphical haptic feedback with robot control systems to improve safety and support the identification of tissue mechanical properties [7]. Eslamian et al. were able to implement an autonomous camera system developed using a da Vinci Standard Surgical System with the DVRK [8].

Nhayoung Hong designed a head-mounted master interface to control the camera arm of the DVRK using head motions [9]. This interface was implemented by adding 27 pressure sensors and a hall-effect sensor to the stereo viewer of the existing da Vinci system to detect the seven simple head movements of the user. The study confirmed that controlling the camera arm using head movements can shorten the surgical operation time and enable continuous surgical flow.

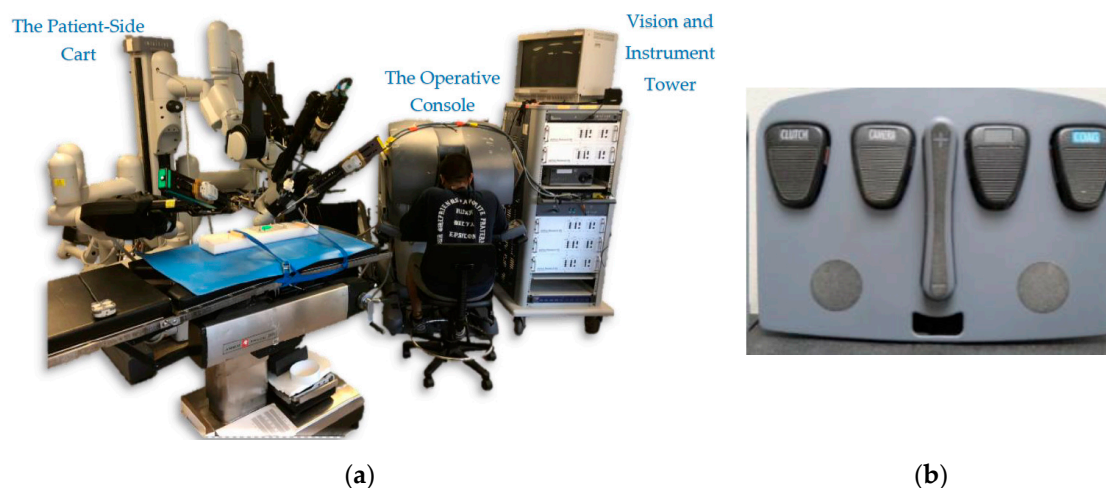


Figure 1. (a) The da Vinci Surgical System with its three main components. (b) The foot pedal tray of the da Vinci system. It is a part of the surgeon console and contains four clutches.

In this study, we aim to show how surgeon interruptions caused by the traditional clutch-based camera control mechanisms can be mitigated by using a head-mounted display to maneuver the robot's camera arm. The user in our system puts on a virtual reality headset, obtains a stereoscopic view, and controls the camera with simple head gestures as shown in Figure 2. This may also create an enhanced sense of presence. This new system has been implemented both in simulation and with the da Vinci hardware. This paper primarily addresses the implementation details of such a system and presents a usability analysis (and not a full user study). The usability study asked users to subjectively evaluate their physical and mental workload and posed a few questions regarding any issues they may have with the system. Some initial data on performance was collected and is presented here, but a full study with a statically significant number of subjects along with surgeon input is planned as the next step.

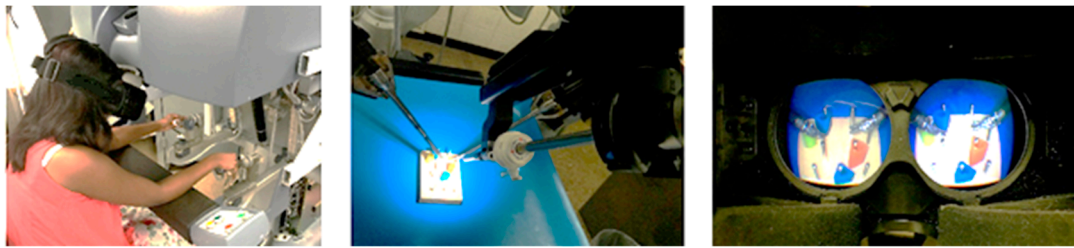


Figure 2. (Left) A da Vinci Surgical System user controlling the camera arm using the headset and hand controllers. (Middle) The worksite/surgical site with the surgical instruments and the camera. (Right) The stereo view from the head-mounted display. The worksite is projected to the display's screens to generate a stereoscopic view for the user.

2. Materials and Methods

2.1. Traditional Control of the da Vinci Surgical System

The surgeon console of the da Vinci system has two hand controllers called master tool manipulators (MTMs). They are used to manipulate both the instrument arms (called the patient-side manipulators, or PSMs) and the camera arm (called the endoscopic camera manipulator, or ECM). The da Vinci system can have up to three PSMs, and they are used to hold “EndoWrist” instruments, such as needle drivers, retractors, and energy-delivering instruments. On the other hand, the system has one ECM that is inserted with the PSMs inside the abdominal cavity to provide a 3D view of the worksite.

The surgeon uses the same hand controllers to control both the PSMs and ECM by using a foot clutching mechanism to change the control behavior of the MTMs (Figure 1b). Once the surgeon presses the camera clutch on the foot tray, the PSMs lock their pose, and movement of the MTMs begins to control the ECM. The orientation of the MTMs is also frozen to match the orientation of the PSMs. As soon as the clutch is released, the MTMs can again be used to control the PSMs. Thus, the surgeon is not able to control the PSMs and the ECM simultaneously, and must pause the operation to adjust the camera view. This can be cumbersome.

There is even more intricacy involved with clutching that can add to the overall complexity. The other three clutches of the foot pedal tray are used to trigger different events. For example, the far-left clutch pedal is used to reposition the MTMs by dissociating them from controlling the PSMs or ECM. Note here that when this clutch is engaged, the orientation of the MTMs remains locked to match the orientation of the PSMs. The remaining two pedals are used to enable the surgeon to perform different tasks such as swapping between diverse types of instruments. The long, two-part button in the middle is used to control the focus of the camera. In summary, the surgeon interface is a complex system that includes multiple clutch controls, and MTM movements are clutch-mapped to control both the ECM and the PSMs.

2.2. The Camera Arm and Headset Hardware

The camera arm, or ECM, is a four degree-of-freedom robot arm located on the patient-side cart (see Figure 3). It is used to manipulate a stereo camera inside the patient using the hand controllers (MTMs) at the surgeon console. The ECM is inserted into an incision to provide vision in the surgical cavity. The endoscopic camera system has two cameras to provide the operator at the surgeon console with a stereoscopic (3D) view of the patient. The resolution of the cameras varies with the model of the da Vinci, but ours provide an analog NTSC (National Television Standards Committee) signal that can be digitized at a resolution of 640×480 (per camera). The two cameras of the ECM are connected separately to two camera control units (Panasonic GP-US522). These units control the video parameter settings (such as color, contrast, white balance, shutter, etc.), and they provide us with S-Video output links of the cameras' views.

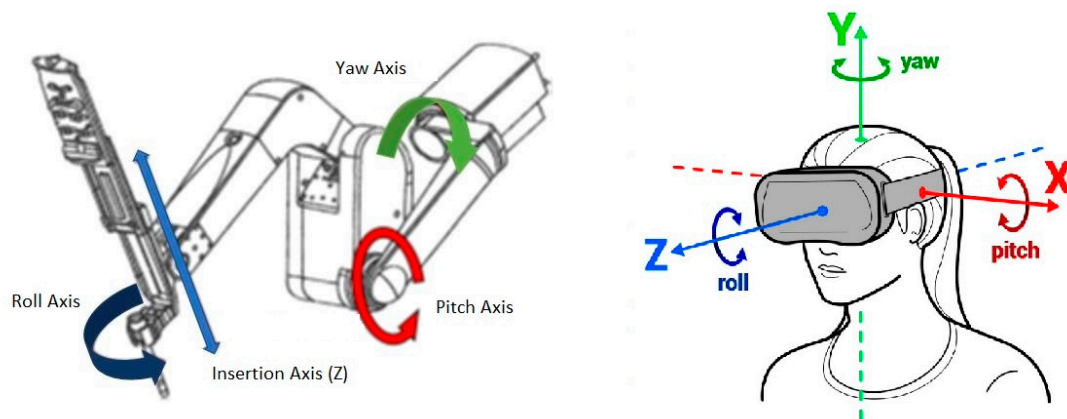


Figure 3. The three rotational joints and the insertion joint of the da Vinci camera arm (left) position the surgical camera based on the corresponding rotational axes and Z-position of the head-mounted display (right).

HTC Vive Head Tracking and Stereo View Interface

The HTC Vive is a virtual reality system developed by HTC and Valve Corporation that was first released in 2016. The Vive consists of three main parts: the head-mounted display (HMD), controllers, and base stations. The system is able to accurately track the poses of the headset and controllers in 3D space. In our study, we used the headset and two base stations, as explained below.

The Vive HMD has an OLED (organic light-emitting diode) display with a resolution of 2160×1200 (1080×1200 per eye) and a refresh rate of 90 Hz. A stereoscopic view of the worksite is created by providing the view of each of the ECM cameras to a different screen in the HMD: the left camera is rendered to the left screen and the right camera is rendered to the right screen. Thus, the human brain perceives the combined images in a way that creates a stereoscopic 3D view [10].

The Vive combines two methods to track the orientation (yaw, pitch, and roll) and position (X, Y, and Z) of the HMD: internal sensors (accelerometer and gyroscope) and external sensors (photosensors) that track lasers emitted from the base stations. The position- and orientation-tracking capabilities of the Vive were documented to have RMS errors below 0.02 cm and 0.02° , respectively, and it is also subjectively said to be fast and create a good sense of presence [11]. We used the OpenVR SDK within Unity to obtain the pose of the HMD from the sensor information. The HMD's pose (i.e., the pose of the user's head) was used to control the pose of the camera arm, as described below and shown in Figure 3.

To translate the pose of the HMD to the joint angles of the ECM, we use both the position and orientation of the HMD in a relative manner. Each of the ECM's rotational joint angles is centered at zero (its "home" position) and can be moved in either direction by going above or below zero. In addition, the insertion joint of the ECM is a prismatic joint that can slide in or out to basically alter the zoom level of the camera. The ECM movements are based on a remote center of motion. This means that the arm rotates around a keyhole (the insertion point). Hence, the yaw, pitch, and roll angle movement of the headset can be directly used to control the yaw, pitch, and roll joint angles of the ECM arm, respectively. The HMD's Z-position (with respect to a defined HMD reference frame) can be applied directly to the insertion joint. Consequently, the kinematic translation between the HMD and the ECM is relatively simple. The Euler angles of the HMD directly control the rotational joint angles of the ECM, and the relative Z position of the headset directly controls the insertion joint (Figure 3). This control scheme does not require the use of inverse kinematics and therefore avoids potential singularity and timing issues for smoother and direct movements.

2.3. Software/Hardware Integration

2.3.1. Robot Operating System

Robot Operating System (ROS) is an open-source software framework for developing robotic applications. ROS includes a message-passing service to facilitate the connection between different robot systems. Messages are passed from publishers to subscribers on channels called topics. For this application, there are topics for things like head movement sensors and robot actuators. The DVRK software and hardware systems use ROS to connect with and control the da Vinci robot. The proposed camera movement algorithm was first verified in a simulation environment using RViz, and a 3D ROS visualization package (see Figure 4). The simulated robot matches the real da Vinci robot in terms of geometric parameters, movements, and joint limits; note that initially the HMD and the ECM don't have to point towards the same direction. Once the algorithm was verified in simulation, the appropriate publisher and subscriber links were added to simultaneously and safely transfer the camera arm movements to the hardware. In addition, the actual stereo endoscopic video was linked to the HMD to provide the view of the patient/worksites.

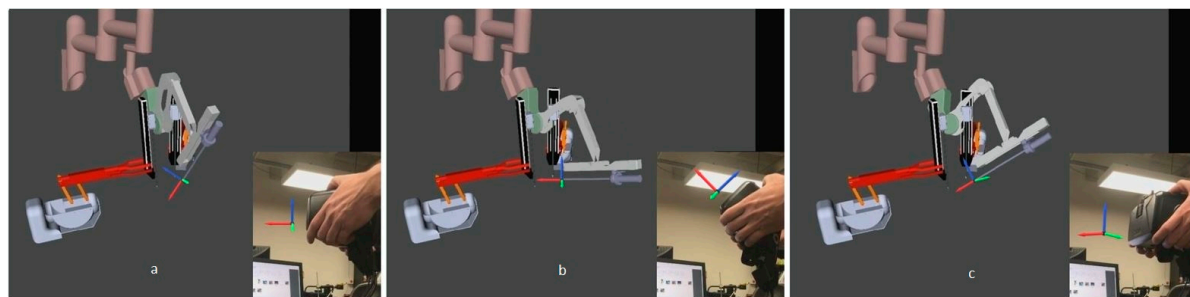


Figure 4. Demonstration of moving the simulated da Vinci robot (displayed in RViz) using the head-mounted display (HMD). (a) The initial poses of the endoscopic camera manipulator (ECM) and the headset. (b) The ECM's corresponding pose when moving the headset about the pitch axis. (c) The ECM's corresponding pose when moving the headset about the yaw axis.

2.3.2. D Rendering Using Unity

Unity is a popular development platform that is commonly used to create 2D and 3D video games. It is used in this paper to render the contents of the virtual environment to the HMD. Inside this environment there are two virtual cameras, with each one seeing one eye of the ECM cameras. Each camera is rendered to one screen of the HMD (Figure 5). In this manner, the user is able to see a 3D stereoscopic image of the da Vinci worksite. Unity is also used to read the pose (position and orientation) data of the headset. Because the Vive interface software is made for Windows, Unity software was enabled to serve the camera pose data to the Ubuntu client running ROS on another PC. More details on this implementation is provided in the next section.

2.4. The Cross-Operating-System Network Interface

Two different operating systems were used in our implementation. The HMD is only supported in Windows (due to driver restrictions), and the DVRK/ROS system only operates on Linux/Ubuntu (due to its ROS implementation). The Vive HMD was connected to a Windows PC and was programmed using the Unity environment. On the other hand, the ECM and the da Vinci control units were connected through the Ubuntu DVRK system running ROS. Due to ease of use and simplicity in the two environments, we are using two programming languages: C# to program in Unity on Windows and Python to program the ROS nodes on Ubuntu.

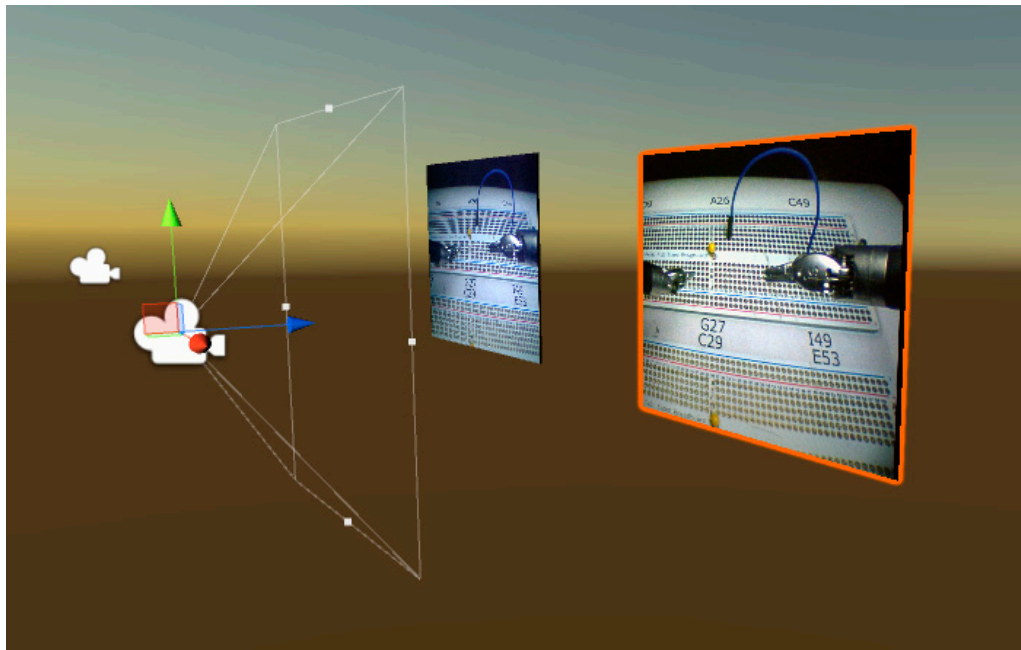


Figure 5. Unity 3D scene for stereoscopic display. The two camera symbols represent the human eyes, and the two flat objects (rectangles, the right one highlighted in orange) represent the headset screens. The images projected to the two flat objects are the images from the two ECM cameras. On the headset, each human eye sees the view of one camera, creating the stereoscopic image.

To connect the two sides together, we used socket communication between the two operating systems running on two different machines. The socket connection used the Transmission Control Protocol (TCP) to communicate between a server on Windows and a client on Ubuntu. The data (HMD pose) was sent from the server (Windows/Unity) to the client (Ubuntu/ROS).

The Ubuntu/ROS software executes a 3D simulation of the da Vinci system (including the camera arm). This can run independently of the hardware, and it enables simultaneous visualization and debugging. If needed, the simulator can run on a separate PC to minimize any performance impact.

Figure 6 shows the flow of data used to control the da Vinci ECM. As the surgeon moves the HMD with his/her head, software on the Windows PC uses Unity libraries to capture the position and orientation of the headset from its onboard sensors. In addition, the software also retrieves the camera images from both ECM cameras, processes them, and projects the camera views of the environment on the HMD screens. The pose of the HMD is then sent via a TCP connection to the Ubuntu machine (running the ROS nodes) to move the robot hardware. The HMD node subscribes to the HMD (HW) node that monitors the ECM's current position (and can also move the hardware). The HMD node then publishes the desired position of the ECM to the ECM (HW) node. The low-level interface software, which subscribes to the ECM node and is directly connected to the hardware, moves the ECM accordingly. Simultaneously, the HMD node also publishes the desired ECM position to the ECM (Sim) node so the camera arm of the simulated robot (in RViz) moves accordingly.

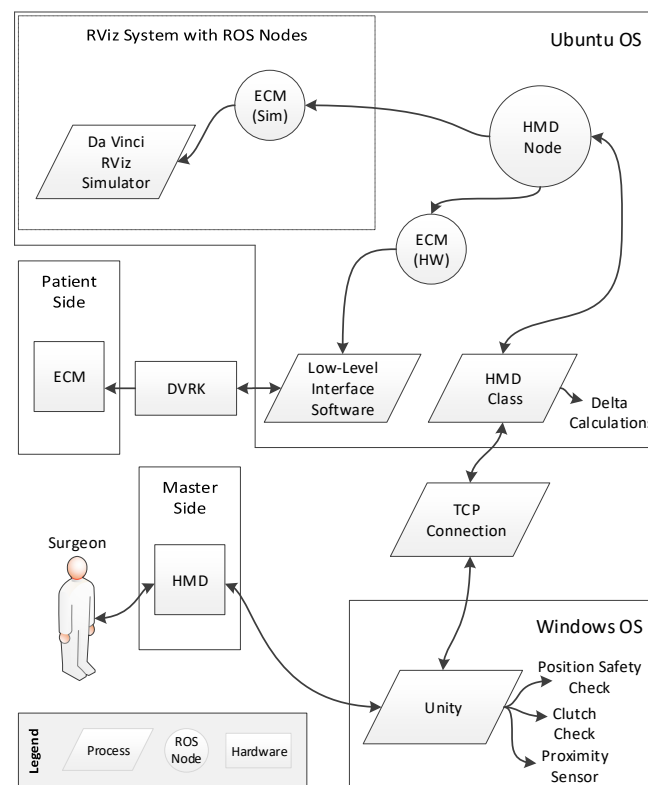


Figure 6. The network of Robot Operating System (ROS) nodes, hardware, and software used for the HMD system.

2.4.1. HMD-Based Control

Figure 7 shows a flowchart of the HMD system software. The overall task is to capture HMD pose information and publish the required joint angles to the ECM hardware in order to match the HMD view and render the ECM camera views to the headset. The first step is to establish a TCP connection for transfer of pose data from the Unity project on the Windows machine (server) to the ROS nodes on the Ubuntu machine (client). Once the connection between the server and the client is established, two conditions should be met for ECM hardware to be activated as follows:

- I. The user should be detected by the headset proximity sensor. The user is asked to sit at the surgeon console and put on the headset. The system only initiates if the user is detected.
- II. The user must be within 30 cm of the center position of the workspace. Once the user is in the desired position and ready to operate as shown in Figure 8, the position of the headset is re-centered to be at ($X = 0$, $Y = 0$, $Z = 0$). The computed joint angles are only sent to the hardware if the headset is within 30 cm of the center position in 3D space (position safety check). This is to avoid spurious movements such as when the headset is being taken off.

For additional safety, the session must be initiated by a person monitoring the system. Even if the system is ready to proceed, the default settings are set to the clutch-engaged setting to prevent the server from sending any data to the client/hardware. The human monitor of the system must activate the software to proceed. In a clinical system, this human monitor could be replaced by the surgeon engaging a foot pedal to activate the system. Additional redundant safety checks could be enabled, such as requiring an initiation step (e.g., closing and opening the grippers of the instrument arms).

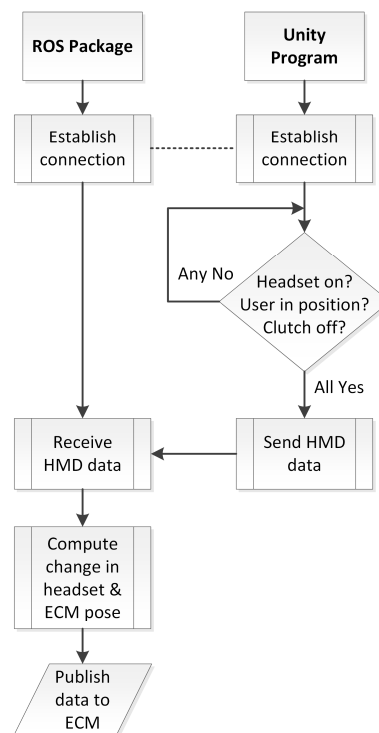


Figure 7. Operation of the HMD control system.

In addition, the user can reposition his/her head by pressing the assigned button on the foot pedal tray. This is a repositioning/reclutching operation to allow the user to re-center himself. This is like lifting a mouse on a mousepad to pause a cursor or repositioning the MTMs by pressing the clutch button. The user depresses the button to dissociate the headset from controlling the ECM, repositions himself and then releases the button to regain to control of the ECM. Once the user is done with the operation, he/she could take the headset off and the proximity sensor will detect this move and pause the system. The final step is to disengage the system using the Unity interface.



Figure 8. A user controlling da Vinci ECM using HMD. The cover of the surgeon console is removed to allow more room for the head movements.

2.4.2. Error Checking to Ensure Hardware Protection

When first donning the HMD for use, there is an initialization that aligns the HMD pose to the ECM pose. To avoid any sudden jumps during this initialization, a simple procedure in software was needed while the ECM adjusted to the position and orientation of the HMD. We created a function to map the ECM and HMD positions by first calculating the difference between the acquired ECM and the HMD positions. Then we added the offset (Delta) to the position values received from the HMD whenever we publish the HMD position to the ECM. Thus, any sudden movements related to initialization were prevented.

This also is a safety mechanism in case there are any sudden movements by the user. A computed delta value is applied whenever a sudden HMD movement (greater than 0.02 rad at a software timing loop of 0.01 s, or 2 rad/s) is performed by the user; when the new computed ECM angular motion is greater than a specified threshold (0.02 rad), a new delta is computed and applied. This value is less than the maximum allowable delta of the ECM (0.05 rad), which is specified in the FireWire controller package developed by Johns Hopkins to protect the hardware. This value is also large enough to accommodate typical head movements. Recalculating the value of delta using the new difference between the poses of the ECM and the HMD when the speed exceeds 2 rad/s prevents the ECM from responding with dangerous, hardware-damaging quick movements. This same mechanism is used for head repositioning when the user presses the foot pedal assigned for the HMD system and moves his/her head to re-center. After repositioning the head, releasing the food pedal creates an offset (bigger than 0.02 rad) which triggers the function to recalculate the delta value and map the ECM and HMD poses. The simplified pseudocode shown in Figure 9 illustrates how the measured head motion is used to control the ECM.

```

1  Obtain the initial poses of the ECM and the HMD
2  Calculate delta = initial_ECM_pose - initial_HMD_pose
3  While HMD node is active
4      IF the HMD speed < 2 rad/sec
5          Next_ECM_pose = current_HMD_pose + delta
6      ELSE
7          Recalculate delta = current_ECM_pose - current_HMD_pose
8          Next_ECM_pose = current_HMD_pose + delta
9      Publish Next_ECM_pose value to ECM node

```

Figure 9. Pseudocode that describes the use of head motion to control the ECM camera.

To enhance surgical dexterity and accuracy, the da Vinci System offers an adjustable motion scaling of 1:1, 2:1, and 3:1 between the MTMs and the PSMs. We implemented the same concept between the HMD and the ECM to have a motion scaling of 2:1. With this value, we tried to map/match the user's hand speed with the head motion. The motion scaling also prevented fast ECM movements, which cause shaking and instability in the ECM hardware. More studies can be performed to optimize the HMD–ECM motion scaling ratio.

2.5. Human Participant Usability Testing

To show the usability of this system on an actual task, an initial 6-subject study was conducted. Six subjects (ranging in age from 23 to 33) were recruited from the student population at Wayne State University in accordance with an approved IRB (Institutional Review Board) for this study. The aim of the study was simply to show that the system is usable and to get some initial objective and subjective feedback from the participants. We prepared a checklist of the essential information/details that the participants should be aware of before starting the study. The same checklist was reviewed by and explained to all the subjects. For instance, this checklist involved explaining the different parts of the system, understanding the usage of the foot pedal tray and the tool-repositioning technique,

and explaining the task. After the introduction, the participants performed the same training for both HMD control and clutched camera control method on a practice task pattern. In this way, we ensured that the subject was at the same level of experience in both methods.

To ensure that the movement method of the camera arm was the only item we tested, we normalized the study by using the same HMD for two conditions. In test 1, the HMD was free to be moved and its orientation controlled the ECM camera. In test 2, the HMD was fixed and the camera arm was moved with a standard clutch-based approach. In this setup, the participant wears the HMD and comfortably places his/her chin on a chin rest which is fixed to the arm rest of the surgeon console (Figure 10). Fixing all the parameters except the ECM movement control methods ensures that the results are not confounded by other parameters, such as screen resolution and comfort of the hardware.



Figure 10. A stationary HMD setup.

We invited the participants to perform certain tasks using both the HMD control method and the traditional clutch control method for a counterbalanced within subject design. We gathered some performance measures (joint angles, speed, and camera view) and survey results (NASA-Task Load Index (TLX)) for this initial study.

For novice users, learning to suture is very complex and takes a lot of training time. To make our testing task simpler, we have developed a system that incorporates movements similar to suture management and needle insertion, but can be done with less training. It involves simply grasping and inserting a needle attached to a wire into a marked point on a flat surface, as seen in Figure 11. The tasks start by asking the participant to move each end of a wire from one spot to another on an electronic breadboard following the blue and yellow arrows shown in Figure 11. The positions in which the wire is placed and where it should go are labeled above and below the breadboard; the rows are labeled from A–J and the columns from 1 to 60. We asked the participants to move both ends of the wire horizontally to the next spots on the upper breadboard (left to right) before moving the wire vertically to the lower breadboard and start moving horizontally again (right to left). The task involved a transfer of the wire tip from one hand to the other at each step. It also involved substantial camera movement including zooming to see the coordinates mapped numbers and letters. This task is like suturing in performance but it is simpler such that it can be performed by a novice user. Moreover, we placed a paper that has an exact image of the breadboards, with the same shape and dimensions, under the actual breadboards so that we could check the punched hole pattern for accuracy. This enabled us to analyze the user progress and detect errors during the test.

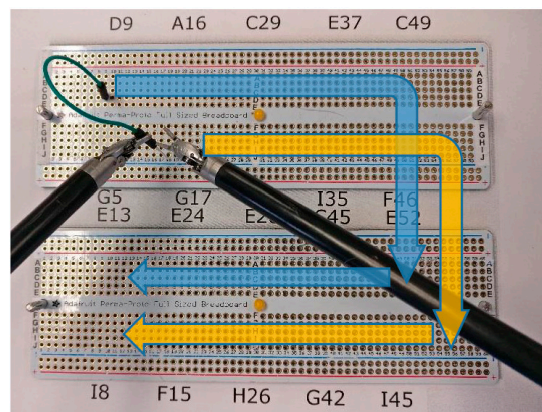


Figure 11. The test task includes a cable and two boards with labeled holes. The subject moves the cable around the boards by inserting the cable into specified holes using the robotic instruments.

The test consisted of eight tasks: one 5-min practice task and three 3-min actual tasks for each of the two camera control methods. The practice task had 20 instructions/steps while each trial had 12. To assure a fair comparison between the two methods, we created 1 pattern for the practice task and 3 different patterns for the actual tasks. In that case, the participant performed the same 3 patterns for each method, but with a counter-balanced and randomized design.

Once all the tasks were completed, we asked the participants to fill out a NASA Task Load Index (TLX) form to assess the workload of each camera control method. NASA-TLX assesses the workload of each method based on 6 criteria: mental demand, physical demand, temporal demand, performance, effort, and frustration. In addition to NASA-TLX forms, we asked the participants to answer two questions:

- (1) "Did you become dizzy or have any unpleasant physical reaction?"
- (2) "Did you feel your performance was affected by any of the following: movement lag, image quality, none, or other?"

The purpose of the two questions was to determine if the headset may have a negative effect on the user when used for a certain period. This preliminary testing took approximately 45–60 min per subject.

For consistency and ease of running our usability test, we also created a graphical user interface that consisted of three main functions: a recording function to log the pose of the camera arm, a function to select which camera control method the test will use, and a timer to keep track of the task time limit.

3. Results

The results of both the NASA-TLX survey and task progress are presented here.

3.1. NASA-TLX and Survey Results

The survey results shown in Figure 12 indicate that the HMD camera control method showed a better result in all the six criteria of the survey. Three of the participants claimed that their performance in both methods was affected by the image quality, while one participant said the movement lag of the HMD control method was affecting the performance. However, all of the participants preferred the HMD method over the traditional clutch control method. None of the participants claimed to feel dizzy or physically uncomfortable that may be a side effect of the headset. The participants also claimed that the logical movements of controlling the camera using the headset was helpful and easier to understand with less training time. The participants also indicated that the traditional clutch control is slow and contains a lot of clutching but the zooming felt very smooth. On the other hand, they indicated that the HMD control was faster and much easier when it comes to small/quick movements.

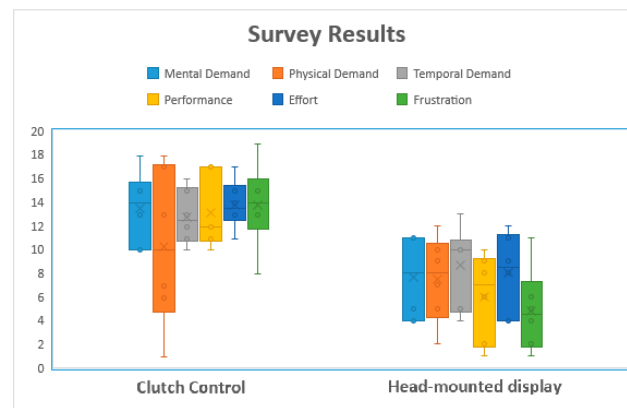


Figure 12. Survey results of the comparison between the traditional clutch camera control and HMD camera control methods.

3.2. Task Performance

In addition to the survey completed using NASA-TLX forms, we compared the two camera control methods based on the number of milestones completed for each test by the participants. The test consists of 8 tasks and each task has 12 instructions (coordinates where the needle of the wire should be pierced). We used the paper placed under the breadboards to count the number of completed instructions and graphed them as shown in Figure 13. It was found that the participants' performance was best when using the headset to control the camera arm with an average of 5 instructions completed for HMD to 3 instructions for the clutch camera control.

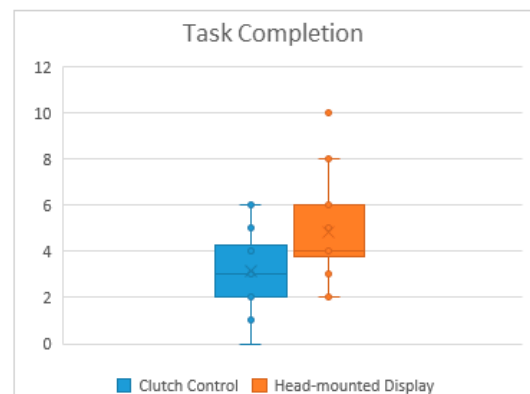


Figure 13. Task completion graph based on the number of tasks completed by the participants for each camera control method.

4. Discussion

After we fully implemented the HMD control system on the da Vinci robot, we performed a preliminary end-to-end study to test the usability of the system. Our main concern was with the potential side effects of the virtual reality (VR) headset on the human with respect to how it affects users in terms of motion sickness, eye strain, headache, and sometimes nausea. None of the participants claimed to have felt physically uncomfortable/dizzy during or after the study. To further investigate this matter, we are planning a much more extensive study with 20–25 subjects, including 3–5 experienced surgeons.

As stated in the results section, 3 of the participants claimed that their performance was affected by the image quality. This is due to the low resolution of the ECM, which provides an image resolution of 640×480 for each eye. The low-resolution image affects the view quality and also the depth perception of the stereoscopic image. To solve this issue, the ECM cameras need to be replaced with higher resolution ones that are more suitable for the VR headset. The newer da Vinci systems have higher

resolutions that may solve this issue. In addition, the quality of VR display panels is also increasing. For this study, both the clutched system and the HMD system used the same resolution, as we were just studying the control method and not the resolution issue.

We faced some challenges during the implementation of the HMD system. First, the DVRK requires the use of FireWire drivers that are only available on the Ubuntu operating system. In addition, much of the supporting software is typically used with ROS on a Linux operating system. On the other hand, the Vive system is only well-supported on Windows. To solve this issue, we established a TCP connection between the two operating systems/environments to send the HMD data to the ECM. A network test between two machines in our lab transferring 32 bytes of information on a round trip resulted in an average delay of 2 ms. Hence, we did not notice any connection delay that could cause significant lag in the ECM movement. Second, we faced a challenge when publishing the joint angles to the ECM. The PID (proportional-integral-derivative) controller of da Vinci is very sensitive and would crash the system when the velocities of the ECM movements were too fast; this is a safety mechanism to avoid any sudden movements that may harm the hardware. To solve this issue, we created a delta function that reads the ECM's current position and the HMD's position and maps the two positions to have the same initial values. The delta value keeps changing accordingly whenever the position sent to the ECM is more than 0.02 rad away from its current position. The same function is used to implement head repositioning, as explained in the operation section.

5. Conclusions

In this paper, we have demonstrated the development of a camera control method for robotic laparoscopic surgery. The hardware implementation of this method used a da Vinci Surgical System and HTC Vive head-mounted display. This method allows the surgeon to manipulate the camera arm of the da Vinci using head movements while having a 3D camera view in the HMD. To verify the usability and functionality of the developed system, we invited 6 subjects to participate in a study to compare the new HMD camera control system with the traditional clutched camera control system. Both the objective and subjective performance measurements of the human subject study were in favor of the HMD control method.

The 6-subject usability study is preliminary. Both a larger number of subjects and more clinically relevant tasks are needed to verify any statistically significant improvement in surgical performance. However, for this usability study, the HMD method showed a promising result to minimize the interruption caused by the clutched camera control method. Moreover, the HMD method seemed to improve the task progress of the tested subjects, which was attributed to the intuitive movement and better 3D representation of the worksite.

For a short video of the developed HMD system in action, please refer to [12]. For access to the HMD system software developed in this paper, please refer to [13].

Future Work

We will perform a more rigorous subject study with more surgically relevant tasks. We plan on using a statically significant number of subjects and involve surgeons in the study. In addition, we plan on further studying any deleterious effects of the HMD in terms of usability with longer duration tasks. Future work will also involve using augmented reality on the HMD. The system could be used to display patient imaging data or other annotations in 3D on top of the live video feed provided by the laparoscope.

6. Patents

A patent covering techniques related to a robotic system with autonomous camera control is held by some of the authors [14].

Author Contributions: T.D., wrote the software and manuscript did the data analysis and ran the usability testing; A.P. had the original concept, directed the project, assisted with the manuscript, assisted with data analysis, and helped with software design. S.E. assisted with software design and development, helped with usability testing, and reviewed the manuscript. L.A.R. helped write the manuscript, assisted with data analysis, and helped with software design.

Funding: The US Department of Veterans Affairs National Center for Patient Safety provided funding under grant “NCPS Robotic Operations Task Excursion Analysis” (VA701-15-Q-O179/2VHF).

Acknowledgments: We would like to thank the individuals who helped achieve this project, especially Benjamin Mick and the participants of the study who generously shared their time and effort. We would also like to acknowledge the Michigan Translational Research and Commercialization Program (M-TRAC) because some of the platform technologies used for this project were originally developed with our M-TRAC grant (“Development of an Automated Camera System”). We wish to thank the Henry Ford Health System and Intuitive Surgical for donating a da Vinci Standard Surgical System to the lab.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sodergren, M.; Darzi, A. Robotic cancer surgery. *Br. J. Surg.* **2013**, *100*, 3–4. [CrossRef]
2. Blavier, A.; Nyssen, A.S. Influence of 2D and 3D view on performance and time estimation in minimal invasive surgery. *Ergonomics* **2009**, *52*, 1342–1349. [CrossRef]
3. Omote, K.; Feussner, H.; Ungeheuer, A.; Arbter, K.; Wei, G.-Q.; Siewert, J.; Hirzinger, G. Self-guided robotic camera control for laparoscopic surgery compared with human camera control. *Am. J. Surg.* **1999**, *177*, 321–324. [CrossRef]
4. Pandya, A.; Reisner, L.A.; King, B.; Lucas, N.; Composto, A.; Klein, M.; Ellis, R.D. A Review of Camera Viewpoint Automation in Robotic and Laparoscopic Surgery. *Robotics* **2014**, *3*, 310–329. [CrossRef]
5. Chen, Z.; Deguet, A.; Taylor, R.; DiMaio, S.; Fischer, G.; Kazanzides, P. An Open-Source Hardware and Software Platform for Telesurgical Robotics Research. In Proceedings of the MICCAI Workshop on Systems and Architecture for Computer Assisted Interventions, Nagoya, Japan, 22–26 September 2013.
6. Okamura, A.; Okamura, A. Methods for haptic feedback in teleoperated robot-assisted surgery. *Ind. Robot. Int. J. Robot. Appl.* **2004**, *31*, 499–508. [CrossRef] [PubMed]
7. Yamamoto, T.; Abolhassani, N.; Jung, S.; Okamura, A.M.; Judkins, T.N. Augmented reality and haptic interfaces for robot-assisted surgery. *Int. J. Med. Robot. Comput. Assist. Surg.* **2012**, *8*, 45–56. [CrossRef]
8. Eslamian, S.; Reisner, L.A.; King, B.W.; Pandya, A.K. An Autonomous Camera System using the da Vinci Research Kit. Available online: https://pdfs.semanticscholar.org/98e7/49a767007f81bff0bb760663138be29386ea.pdf?_ga=2.180461543.1179920898.1555468191-422022476.1555468191 (accessed on 10 April 2017).
9. Young, H.N. A Study on the Development of Head-Mounted Master Interface for Laparoscopic Surgical Robot System. Ph.D. Thesis, Seoul National University Graduate School, Seoul, Korea, 2018.
10. Shibata, T.; Kim, J.; Hoffman, D.M.; Banks, M.S. The zone of comfort: Predicting visual discomfort with stereo displays. *J. Vis.* **2011**, *11*, 11. [CrossRef] [PubMed]
11. Niehorster, D.C.; Li, L.; Lappe, M. The accuracy and precision of position and orientation tracking in the HTC vive virtual reality system for scientific research. *i-Perception* **2017**, *8*, 2041669517708205. [CrossRef] [PubMed]
12. HMD System in Action. Available online: <https://youtu.be/nTwl-ybBHRA> (accessed on 18 April 2019).
13. Head Mounted Display. Available online: https://github.com/careslab/head_mounted_display (accessed on 18 April 2019).
14. Pandya, A.; Klein, M.D.; Mudunuri, A.V.; Cao, A.; Reisner, L.; King, B.; Ali, S. Intelligent Autonomous Camera Control for Robotics with Medical, Military, and Space Applications. U.S. Patent 9,439,556, 13 September 2016.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).