An Autonomous Camera System using the da Vinci Research Kit*

Shahab Eslamian, Luke A. Reisner, Brady W. King, and Abhilash K. Pandya, *Member, IEEE*

Abstract— **Manual control of the camera arm in telerobotic surgical systems requires the surgeon to repeatedly interrupt the flow of the surgery. This may lead to increased workload and potential errors. This paper provides the implementation details of an autonomous camera system developed using the da Vinci Research Kit with a da Vinci Standard Surgical System. We show that an on-demand autonomous camera system can be integrated with a modern surgical robot. None of the previous autonomous camera research has used a da Vinci robot as a test platform. We show that our hardware implementation closely matches a software simulation of the complete system. Usability testing suggests that this work has the potential to become a useful tool for minimally invasive surgery.**

I. INTRODUCTION

Teleoperating a medical robot using remote video views is challenging due to a limited field of view and issues with achieving proper views of the remote site. After observing several complex robotic surgery cases, it was noted that the camera view is manipulated as much as 100 times during one hour of surgery. When the surgeon is adjusting the camera, he has to interrupt the flow of the surgery, adjust the camera, and then resume. At times, non-optimal (not properly zoomed or positioned) views are selected to avoid continuous interruption of the flow of the surgery. In addition, it was noted that there were frequent times when a tool being actively used was placed outside the field of view. Moreover, sometimes both tools were not visible.

The problems of surgical interruption during camera movement, non-optimal views, and tools outside the field of view may cause errors and increase surgical times. The premise of this paper is that these issues could be mitigated by an on-demand autonomous camera system that makes these systems safer and easier to use. Here we explain and demonstrate such a system on a da Vinci Standard Surgical System (Intuitive Surgical, Inc., Sunnyvale, CA). The developed test platform and baseline autonomous camera algorithm are the first steps towards intelligent camera control in a clinically relevant surgical robot.

A. Related Work

A review paper on surgical camera automation [1] provides a summary of the current approaches. None of the current work uses a da Vinci robot as a test platform, which is currently the only clinically viable system on the market.

Previously, we implemented an autonomous tool-following camera algorithm in a simulation of the da Vinci Standard Surgical System [2]. In this paper, we describe how we translated the baseline camera control algorithm from the simulation to work with the actual hardware of a real da Vinci system.

II. METHODS

A. Hardware and Software Environment

Central to this research is the da Vinci Research Kit (DVRK). It is a hardware/software platform that helps researchers implement their ideas using a da Vinci Standard Surgical System. Besides the robot, the main components of the DVRK are hardware control boxes (containing FPGA boards and amplifiers) and open source software that enables computerized control of the arms. The DVRK software uses the open source Robot Operating System (ROS) framework [3] as well as the CISST and SAW libraries developed by Johns Hopkins University [4].

During system use, the user manipulates hand controllers in order to move the instrument arms (PSM1 and PSM2) and camera arm (ECM) of the da Vinci. The DVRK captures the joint values of these hand controllers. The system then sends this information to low-level interface software that computes and sets the joint values for the PSMs. Our autonomous camera (autocamera) algorithm uses these joint values to compute and set the desired joint values for the ECM.

B. Registration and Calibration

The camera placement algorithm relies on accurately knowing the relative poses of the tools and camera. The current DVRK hardware cannot provide this information because there are untracked "set-up" joints that allow the bases of the camera and tool arms to be moved. Therefore, a procedure based on mathematical optimization was used to co-register the robot arms.

First, the tips of the robot arms were touched together at numerous arbitrary locations throughout the robot's working volume. Next, an objective function was written that used an accurate kinematic model of each arm and the recorded joint values of each arm to calculate the average distance between the tips of each arm. Finally, using the Nelder–Mead method [5], we computed pose transformations between each of the robot arm bases by minimizing the calculated tip distances.

An accurate stereo camera calibration was performed to determine the rectification transform (to align the images from both cameras), the perspective transform (providing the viewing frustum), and other calibration/distortion parameters (to correctly relate 3D data to the 2D camera images). We performed camera calibration for the stereo cameras using the ROS *camera_calibration* package [3]. The identified camera calibration parameters included the focal length, field of

---

* Research supported by Children's Hospital of Michigan and the Children's Hospital of Michigan Foundation.

S. Eslamian, L. A. Reisner, and A. K. Pandya are with the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202, USA.

B. W. King is with the Department of Pediatric Surgery, Children's Hospital of Michigan, Detroit, MI 48201, USA.

Correspondence to A. K. Pandya (e-mail: apandya@ece.eng.wayne.edu).

view, distortion parameters, and the rectification matrix. The parameters were used to correct the actual camera images, and they were also utilized in the simulation to accurately model/simulate how the actual camera captures images.

C. *Autonomous Camera Software Algorithms*

Using the locations of the end-effectors in both 3D space and the 2D camera view, the ECM is continuously positioned to provide a suitable view of the operating environment. A midpoint-tracking algorithm finds specific joint values that enable the ECM arm to point towards the centroid of the tools. A zoom algorithm uses the locations of the tools in the 2D view to set the distance of the ECM from the tools (zoom level). The zoom level is maintained while the instruments are in an (adjustable) dead band. The camera zooms in or out if the tools move into the interior or exterior (respectively) of the dead band. The full details of the mathematics behind these algorithms are outlined in our simulation paper [2].

III. RESULTS

We measured distances from the ECM's tip to PSM1's tip and from the ECM's tip to PSM2's tip for both the simulation and the actual hardware. The absolute differences between corresponding real and simulation measurements were computed to determine how well they matched. For 10 ECM to PSM1 measurements, the mean of the absolute differences was 1.54 mm with 95% confidence interval of 0.40 to 2.68 mm. For 10 ECM to PSM2 measurements, the mean of the absolute differences was 1.91 mm with 95% confidence interval of 1.02 to 2.80 mm. These results demonstrate that our hardware implementation was very close to our ideal simulation.

Fig. 1 shows a comparison between the simulation and the actual da Vinci hardware with our autocamera algorithm. Images (a) and (c) compare a real and a simulated camera view from the ECM. One can note that the orientation of the tools and the sizes of the objects in the scene are very similar. In addition, the lower images, (b) and (d), provide overviews of the real and simulated environments from roughly the same viewpoint. Note the close similarities in the arm joint values and the orientation of the camera arm (ECM) relative to the two tool arms (PSMs). This indicates that the whole pipeline of camera calibration, robot co-registration, autocamera computations, and hardware manipulation is working correctly.

In a basic usability test, we were able to successfully complete a peg transfer task (common in surgical training) with the da Vinci's camera arm being controlled by our autocamera algorithm. The camera automatically followed the tools as intended, reducing the user's workload.

IV. CONCLUSION

This work shows that it is possible to have an autocamera system implemented on a da Vinci surgical platform. Our preliminary testing suggests that the work has the potential to become a useful tool for minimally invasive surgery. In order to help achieve this goal, some future work is needed.

First, we plan to perform one or more user subject studies that will compare the currently used manual camera control mechanism with our autonomous camera system. This will enable us to better understand the utility of the system and guide our future research efforts.

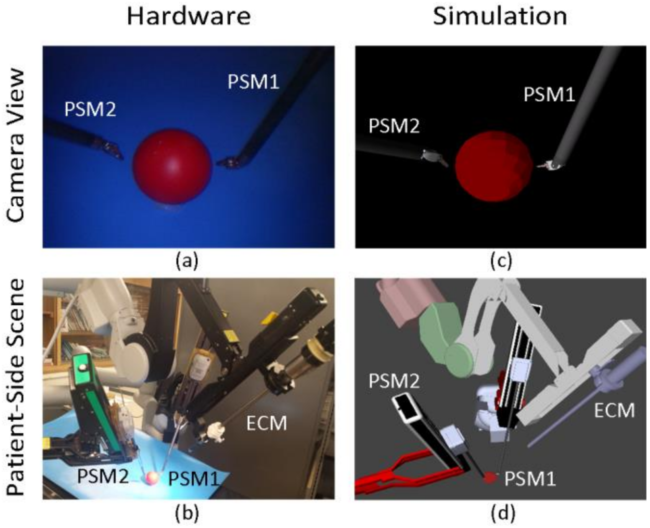

Figure 1. Comparison between real hardware images, (a) and (b), and the corresponding simulation images, (c) and (d), for a particular scenario.

Second, we recognize that the algorithm implemented here represents one of the simplest forms of autonomous camera movement. Therefore, we plan to develop forms of camera control with more intelligence. For example, the integration of task analysis and task-specific behaviors should allow the system to perform better for different surgical procedures. Imaging processing and other sensing techniques could be added to support the tracking of objects other than the robot arms (bodily structures, clips, needles, etc.). We anticipate that the adoption of more advanced techniques, guided by testing, will lead to better performance in clinical use.

ACKNOWLEDGMENT

For their assistance, we would like to thank Anthony Composto and Darin Ellis at Wayne State University; Anton Deguet, Peter Kazanzides, Zihan Chen, and Russell Taylor at Johns Hopkins University; and Greg Fischer at Worcester Polytechnic Institute.

REFERENCES

[1] A. Pandya, L. A. Reisner, B. King, N. Lucas, A. Composto, M. Klein, *et al.*, "A Review of Camera Viewpoint Automation in Robotic and Laparoscopic Surgery," *Robotics*, vol. 3, pp. 310-329, 2014.

[2] S. Eslamian, L. A. Reisner, B. W. King, and A. K. Pandya, "Towards the Implementation of an Autonomous Camera Algorithm on the da Vinci Platform," presented at the Medicine Meets Virtual Reality 22, Los Angeles, CA, USA, 2016.

[3] Open Source Robotics Foundation. (2017, September 14). *ROS Documentation*. Available: http://wiki.ros.org/

[4] Z. Chen, A. Deguet, R. Taylor, S. DiMaio, G. Fischer, and P. Kazanzides, "An open-source hardware and software platform for telesurgical robotics research," in *SACAI Workshop at MICCAI 2013*, Nagoya, Japan, 2013.

[5] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder–Mead simplex method in low dimensions," *SIAM J. on Optimization*, vol. 9, pp. 112-147, 1998.