Augmented Reality Predictive Displays to Help Mitigate the Effects of Delayed Telesurgery

Florian Richter¹, Yifei Zhang², Yuheng Zhi³, Ryan K. Orosco⁴ Member, IEEE, and Michael C. Yip¹ Member, IEEE

Abstract-Surgical robots offer the exciting potential for remote telesurgery, but advances are needed to make this technology efficient and accurate to ensure patient safety. Achieving these goals is hindered by the deleterious effects of latency between the remote operator and the bedside robot. Predictive displays have found success in overcoming these effects by giving the operator immediate visual feedback. However, previously developed predictive displays can not be directly applied to telesurgery due to the unique challenges in tracking the 3D geometry of the surgical environment. In this paper, we present the first predictive display for teleoperated surgical robots. The predicted display is stereoscopic, utilizes Augmented Reality (AR) to show the predicted motions alongside the complex tissue found in-situ within surgical environments, and overcomes the challenges in accurately tracking slave-tools in real-time. We call this a Stereoscopic AR Predictive Display (SARPD). To test the SARPD's performance, we conducted a user study with ten participants on the da Vinci[®] Surgical System. The results showed with statistical significance that using SARPD decreased time to complete task while having no effect on error rates when operating under delay.

I. INTRODUCTION

Teleoperation brings the advantage of remote control and manipulation to distant locations or harsh or constrained environments. The system allows operators to send commands from a remote console, traditionally called a *master* device, to a robot, traditionally called a *slave* device, and offers synchronization of movements. This allows the remote user to operate as if on-site, making teleoperational systems an ideal and often only solution to a wide range of applications such as underwater exploration, space robotics, mobile robots, and telesurgery [1], [2].

Perhaps the most well known and still widely used teleoperation system to date is the da Vinci[®] Surgical System, which is deployed in thousands of hospitals. However, the master and slave systems are operated in the same room, providing some advantages of tremor reduction and stereoscopic vision [3] but none of the remote teleoperation functionality. The main technical challenge in realizing remote telesurgery



Fig. 1: Stereoscopic AR Predictive Display (SARPD) showing the predicted slave-tools in real-time (highlighted with the dotted red lines), overlaid on the camera feedback under 1sec of round trip delay. SARPD can help users coordinate motor tasks preemptively to save time.

(and similarly, all remote teleoperation) is the latency from the communication distance between the master and slave. Latency can reach beyond a second when linking sites between continents [4], and may have higher ranges when using satellite or space communications. This delay causes overshoot and oscillations in the commanded positions, and are observable and statistically significant in as little as 50msec [5] [6] of round trip communication delay. Anvari et al. reported on 21 remote laparoscopic surgeries where a distance of over 400km had delay of 135-140msec [7]. Furthermore, 300msec has been stated as the largest delay where surgeons feel safe [8], [9], and when using satellite communication for operations between London and Toronto, a delay of 560msec was observed [10].

Teleoperational systems under delay is not specific to telesurgery and its history is too vast to be covered in this paper. Please refer to Hokayem and Spong's historical survey for a more extensive background on prior work [2]. Nonetheless, early work by Ferrel and Sheridan suggested supervisory control to mitigate the negative effects of delay [11]. This gives the robotic system the capability of making limited decisions on its own while being supervised by the operator; however, the master loses explicit control of the motions of the slave, and thus supervisory controlled can only be implemented practically at this time in structured environments.

The largest area of research in delayed teleoperation has

¹Florian Richter and Michael C. Yip are with the Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093 USA. {frichter, yip}@ucsd.edu

²Yifei Zhang is with the Department of Computer Science and Engineering, University of California San Diego, La Jolla, CA 92093 USA. yiz243@eng.ucsd.edu

³Yuheng Zhi is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China zyh1996@sjtu.edu.cn

⁴Ryan K. Orosco is with the Department of Surgery - Division of Head and Neck Surgery, University of California San Diego, La Jolla, CA 92093 USA. rorosco@ucsd.edu

been for haptic feedback. When haptic feedback is given to the operator, it has been experimentally measured and theoretically shown that delay causes instability [12], [13]. While there are some techniques to dampen the unstable overshoot and oscillatory behavior such as wave variables [14], these techniques have been shown to increase task completion time when under delay [15], [16], and teleoperating without haptic feedback can be often the better alternative.

Predictive displays are virtual reality renderings, generally designed for space operations, that show a prediction of the events to follow in a short amount of time. It can be used to overcome the negative effects of delay by giving the operator immediate feedback from a predicted environment. Furthermore, it does not suffer stability issues that arise with delayed haptic feedback. Early predictive displays included manipulation of the Engineering Test Satellite 7 from ground control where the round trip delay can be up to 7sec [17], [18] and Augmented Reality (AR) rendering where the prediction is overlaid on raw image data [19]. These strategies can be applied to telesurgery, but require overcoming the unique challenges in calculating and tracking the 3D environment for a full environment prediction, which includes non-rigid material such as tissue. Furthermore, prior work in the surgical robotics community highlights the need for active tracking rather than only relying on kinematic calibrations to localize the slave due to the millimeter scale of a surgical operation and the often utilized cable driven actuation [20], [21], [22], [23], [24].

In this paper we propose the first predictive display designed for teleoperated surgical robots. Most crucially, we calculate the predicted behavior of the robot arms and display this prediction in real-time to the operator to anticipate the delayed slave feedback video. The novel contributions of the paper are:

 a real-time strategy for kinematically estimated AR registration, rendering, and lens-distorted image overlays for stereoscopic left and right streams of laparoscopic stereo cameras,

- an Extended Kalman Filtering (EKF) strategy to address the challenges of visual-mismatch between the prediction and the actual movements that arise from imperfect kinematic calibrations with on-the-fly corrections, and
- an adaptive transparency filter that prevents confusion arising from overlapping virtual and real visuals of the robot arms.

We present a complete system, and show through a user study that the result is an efficient AR rendering architecture for streaming stereoscopic displays. We call this a Stereoscopic AR Predictive Display (SARPD). We show that over 30fps for the stereoscopic AR rendering and 24Hz for the slave-tool tracking from the EKF can be achieved while running simultaneously on a commodity GPU and using the robot operating system (ROS). A user study is carried out to demonstrate its ability to improve the speed of procedures without affecting error rates. Beyond telesurgery, this solution can be deployed to any teleoperated robot with visual feedback once calibrated. Furthermore, it does not require stereo cameras or displays since the real-time slavetool tracking is done on a single monocular camera data stream.

II. METHODS

A block diagram in Fig. 2 shows the architecture for SARPD, and the variables shown will be used throughout this paper. In surgical systems, such as the da Vinci \mathbb{R} Surgical System, translational motions from the operator are scaled down to improve accuracy. This constant scaling relationship to set the slave's pose, *s*, from the operators input pose, *m*, in a teleoperation system under delay is described as follows:

$$p_{s_m}[n] = scale(p_m[n] - p_m[n-1]) + p_s[n-1] \quad (1)$$

q

$$s_m[n] = q_m[n] \tag{2}$$

$$s[n] = s_m[n - n_d/2]$$
 (3)



Fig. 2: Architecture for SARPD with a round trip delay of d where m, s_m , and s are all poses, j are the joint angles, T_b^c are the hand-eye transforms between the base of the slave arm and camera, and I are stereo images. For the sake of simplicity, the entire system is assumed to have a sampling rate of f_s , and let $n_d = f_s d$

where p_x and q_x are the translational component and quaternion of pose x. Equations (1) and (2) give the target pose for the slave, s_m , through the constant scaling for the position and mirroring the operators rotational input. Note that both s_m and s are poses in the corresponding slave arm base frame. The rotation is mirrored because master consoles such as the a Vinci[®] Surgical System use wrist orientation as an input. Equation (3) simply highlights the delay channel.

The predictive display proposed here has two major components: slave-tool tracking by correcting the hand-eye transform in real-time to calculate an accurate prediction and stereoscopic AR rendering to display the prediction. Both are running asynchronously, and ROS is used to pass the data from the slave-tool tracking to the stereoscopic AR rendering.

A. EKF to Correct Hand-Eye

Ye et al. previously developed a tacking algorithm for the slave-tools and showed it to be accurate and robust to surgical environments [22]. It successfully tracked the slavetools in real-time by estimating the error of the initial handeye calibrations between the base of both slave arms and the left camera in real-time by using:

- 1) virtual slave-tool rendering to generate part-based templates online,
- template matching between virtual slave-tool parts and image data,
- 3) geometric context to extract the best location estimates of the slave-tool parts, and
- 4) EKF to track the error of the hand-eye transform using the previously found 2D estimates for the update.

The hand-eye correction from [22] is implemented on the master side, so the predicted AR slave-tools use the correction corresponding to the image data they will be displayed with. This means the joints angles, j[n], from encoder readings and image data, I[n], are passed through the communication channel before calculating the correction. This differs from [22] which is not concerned with delayed teleoperation. Each slave-tool part is given its own thread in step 1 and 2 to improve the real-time performance. Fig. 3 is an example photo showing the detected features and rendered slave-tools that use the hand-eye correction.

To initialize the EKF, the hand-eye transform from calibration completes steps 1 through 3, and instead of step 4, a Perspective-n-Point (PnP) solver on the detected features is applied [22]. To improve robustness, this is repeated when the distance or absolute difference in roll, pitch, or yaw between the corrected hand-eye and initial hand-eye from calibration is greater than a set threshold. This and the EKF updates to the corrected transform, \hat{T}_b^c , cause large steps in the hand-eye transform relative to the scale of surgical environments. These large steps will make the predicted AR slave-tool appear jumpy and therefore untrustworthy to operators. To smooth this, the following filter is applied to get the outputted hand-eye transform, T_b^c :



Fig. 3: Implementation of hand-eye correction inspired by [22] with adaptation for round trip delay of 1sec and parallelization to improve performance. Colored points represent visual features used in the EKF tracker that are related to tracking of kinematic robot links.

$$q_b^c[n] = \frac{\sin((1-a)\Omega[n])}{\sin(\Omega[n])}q_b^c[n-1] + \frac{\sin(a\Omega[n])}{\sin(\Omega[n])}\hat{q}_b^c[n] \quad (4)$$

$$\Omega[n] = \cos^{-1} \left(q_b^c[n-1] \cdot \hat{q}_b^c[n] \right) \tag{5}$$

$$p_b^c[n] = (1-a)p_b^c[n-1] + a\hat{p}_b^c[n]$$
(6)

where q_b^c and p_b^c are the quaternion and translation representations of the outputted, smoothed hand-eye transform, T_b^c . Likewise, \hat{q}_b^c and \hat{p}_b^c are the corrected transforms from the EKF's output, \hat{T}_b^c . This is simply a first order infinite impulse response filter with parameter *a* and spherical linear interpolation to average the rotations.

B. Stereoscopic AR Rendering

SARPD is rendered using The Visualization Toolkit (VTK) and OpenGL. To render the predicted slave-tools, their 3D CAD models are loaded as VTK Actors and uses the most recent filtered hand-eye transform to be in the left camera frame. The predicted joint angles for the slave-tools, $\hat{j}[n]$, are calculated through inverse kinematics from target pose $s_m[n]$, as shown in Fig. 2. For the right display rendering, the slave-tools are additionally transformed using the baseline from stereo camera calibration to be in the right camera frame.

Distortion must be applied to the rendered slave-tools since the master console shows the distorted images to the operator. Camera calibration procedures in MATLAB and OpenCV find the coefficients for equations that un-distort barrel and tangential camera distortions. Solving for the inverse of these equations gives multiple solutions. Therefore, bilinear interpolation on the inverse of the discretized undistort mapping is used to find the distortion map for both the left and right cameras.

Similar to how mappings are applied in OpenCV, the distortion map for each camera is split into two separate mappings, columns and rows. The four mappings are uploaded as textures onto the GPU once at the beginning and then applied to the slave-tool renderings using fragment shaders.

Two virtual cameras, left and right, render the stereoscopic display. Each virtual camera display overlays the predicted



Fig. 4: Example photos of AR prediction and dynamic transparency under 1sec of round trip delay. From left to right, a ring hand-off task is shown.

and distorted slave-tool renderings on top of the corresponding image frame which are uploaded to the GPU as textures. Both the left and right rendering pipeline are done in parallel and the VTK slave-tool actors and textures are shared resources on the GPU. Through this implementation optimization and utilizing fragment shaders to apply the distortion mapping, the rendering pipeline can be run on consumer grade GPUs. Fig. 1 shows an example of the predicted slave-tools and image data rendering.

Even with the real-time hand-eye correction, there are other kinematic inaccuracies that are unaccounted for. Joint angle inaccuracies on the wrist of the slave-tool will cause inconsistency between the AR rendering and the image. This is noticeable when the slave-tools are making no motion. Even small, sub-millimeter inaccuracies were observed to cause confusion to users during initial studies. To overcome this, the opacity, α , of the rendered slave-tools is dynamically set with the following equation:

$$l = \left| \left| p_{s_m}[n] - p_s[n - n_d/2] \right| \right|$$
(7)

$$\alpha = \min(\alpha_{max}, r(\max(l_{thresh}, l) - l_{thresh}))$$
(8)

The opacity will increase proportionally with the distance l, which is the distance between the new target pose that the prediction shows and the pose in the delayed image. The distance threshold, l_{thresh} , is the minimum value l must have to not be fully transparent, and α_{max} simply sets the maximum opacity. The dynamic opacity allows for the operator to use the image data undisturbed when making precise, slow motions where the kinematic inaccuracies would be apparent and naturally use the AR prediction when making larger motions. Fig. 4 shows an example of the dynamic transparency during a ring hand-off.

III. EXPERIMENTAL SETUP

To initialize the hand-eye transform for the EKF, we provide a calibration method that only requires a rigidly mounted checkerboard on the slave-tools (only during the calibration phase). This is the only additional hardware required to implement the proposed predictive display, and calibration is only required once before a procedure. Performance and latency tests were also conducted to show the efficiency of SARPD.

To measure the effectiveness SARPD, a user study with ten participants was conducted on the da Vinci (\mathbb{R}) Surgical System. The participants were seven novices and three surgeons who use the da Vinci for their practice. Errors and time to complete task are used as metrics, and statistical testing was done to draw conclusions of SARPD's performance. A recent study found that when using satellite communication for telesurgery between London and Toronto, a round trip delay of 560.7 ± 16.5 mecs was measured [10]. So a round trip delay of 1sec was used for the delayed environment in the user study to ensure SARPD performs in realistically high latency remote operations.

SARPD is rendered to the master console at 30fps. The operator manipulates two master arms in the console to get pose m. There are two 7-DOF slave arms which go to pose s. Modifications were made to the daVinci Research Toolkit (dVRK [25]) to support the architecture proposed in Fig. 2 for the da Vinci Surgical System. The computer used to run the modified dVRK and SARPD has an Intel® CoreTM i9-7940X Processor and NVIDIA's GeForce GTX 1060.

A. Initial Hand-Eye Calibration

To find the initial hand-eye transform, a checkerboard is rigidly attached to the end-effector of the slave-tool. The arm is moved around and photos are taken of the checkerboard and the corresponding joint angles are recorded. The *i*-th corner on the checkerboard in the checkerboard frame, $p_e(i)$, with side length *s*, is projected onto the image plane with the following equation:

$$p_c(i,j) = \frac{1}{d} K T_b^c T_{-e}^b(j) T_e^{-e} p_e(i)$$
(9)

 $\frac{1}{d}K$ projects a point onto the image plane using the camera matrix. $T_{-e}^{b}(j)$ is the pose of the checkerboard which is calculated by using forward kinematics from the recorded joint angles at image j, and T_{e}^{-e} is the constant offset error of the calculated $T_{-e}^{b}(j)$ which accounts for mounting errors of the checkerboard. To solve for the hand-eye transform in a robust manner, the following optimization problem is solved when there are n corners on the checkerboard and m recorded images and joint angles:

$$\underset{T_{b}^{c}, T_{c}^{-e}, s}{\operatorname{argmin}} \frac{1}{m} \sum_{j=1}^{m} \sum_{i=1}^{n} \left| \left| p_{c}(i, j) - z(i, j) \right| \right|^{2}$$
(10)

where z(i, j) is the pixel position of the checkerboard corner on the recorded image. Optimizing the constant offset of the checkerboard and side length, *s*, will help account for alignment errors and printing inaccuracies. T_e^{-e} is initially set to the identity matrix.

To initialize the hand-eye transform, a PnP solver is used on each recorded image to find the pose of the camera in the



Fig. 6: Calibration tool on the instruments to find the transform from the slave arms base to the camera frame. This calibration is carried out once, as needed to attain an initial hand-eye transform for the EKF. The circles highlight the re-projected checkerboard corners.

checkerboard frame. This is then further transformed with the corresponding $T_{-e}^{b}(j)$ to get image j's individual estimate of the hand-eye transform. The initial value for the hand-eye is simply set to the arithmetic average of the positional component and roll, pitch, and yaw of all of the images individual estimates. The optimization is solved by using MATLAB's fmincon function, and Fig 6. shows an example result of the re-projected corners on a recorded image.

B. Task for User Study

A peg transfer task, similar to a task in the Fundamentals of Laparoscopic Surgery, is the sample task used in the user study due to the complex motions involved with it. A photo of the environment from the endoscope is provided in Fig. 7. To complete the task, the operator must pick up the ring from the front right or left peg with the corresponding arm, pass the ring to the other arm, place the ring on to opposite front peg, and then repeat with the back pegs and ring.

Time to complete task and weighted error are used to evaluate performance of completing the task. To find the weighted error, all errors that occurred during a trial are counted and weighted according to Table 1. Example errors are shown in Fig. 5. The weightings were chosen such that the severity of the errors would be reflected properly when evaluating the performance of completing the peg transfer. Each participant was also shown a video of a complete task with no errors and examples of the errors before starting.

C. Procedure for User Study

After introducing the participant to the study and showing the video, the participant goes through the following procedure:



Fig. 7: Environment setup used for the peg transfer task. Transfer of rings are carried out between instruments and between rows of pegs to characterize single and dual arm control in pickup, hand-off, and place-down task.

TABLE I:					
/eights	associated	with	type	of	error

W

Error	Weight
Touch peg	1
Touch ground	2
Stretch ring during handoff for a second or less	2
Drop ring	3
Stretch ring on peg for a second or less	4
Stretch ring for an additional second	4
Stretch/move peg	10
Knock down peg	20

- 1) Practice: complete the task twice under no delay and with no SARPD
- 2) Record: complete the task once under no delay and with no SARPD
- 3) Repeat step 1 and 2 under delay and with no SARPD
- 4) Repeat step 1 and 2 under delay and with SARPD

Two practice trials are always completed before every recording, so that the participant can overcome the learning curve for the new environment. To further ensure this, participants are also offered additional practice before recording. The order was chosen so that participants only encounter one new environmental effect at a time. Through initial experimentation we set the following values: scale = 0.2(the default value for dVRK) for equation (1), a = 0.8 for equation (4) and (6), and $l_{thresh} = 5.3$ mm, $\alpha_{max} = 0.8$, and r = 100 for equation (8).

Note that SARPD without using the active tracking from the EKF is not included in the experiment. This was elected due to the he highly inaccurate prediction from the AR rendering when relying on the kinematic calibration.



Fig. 5: Example errors participants made. From left to right: touch peg, stretch ring on peg, drop ring, and stretch ring during handoff.

IV. RESULTS

The original work for the slave-tool tracking [22] was measured to run at 26Hz and 13Hz for the part-based template generation and overall tracking respectively. Through the parallelization efforts, the slave-tool tracking now runs at 50Hz and 24Hz for the part-based template generation and overall tracking respectively. During this measurement, the stereoscopic AR rendering ran asynchronously at 36fps.

We also measured the latency of the entire image pipeline, which is the time from when an image is captured to when it is displayed to the operator under no delay. When using the SARPD, the latency was measured to be 100 ± 20 ms. To compare, we ran the same experiment with ROS's *image_view* to display the raw image data and measured a latency of 110 ± 20 ms. ROS's *image_view* is the recommended way to view images in dVRK, so our solution also improves on the latency of the system.

TABLE II: Average and standard deviation from the user study results

	Weighted Error	Completion Time (sec)
No SARPD, d=0sec	2.8 ± 3.4	29.6 ± 8.8
No SARPD, d=1sec	4.9 ± 4.7	60.6 ± 14.8
SARPD, d=1sec	3.7 ± 3.2	49.2 ± 6.3

The results of the user study comparing delayed teleoperation with and without SARPD is shown in Fig. 8 and the statistics are given in Table 3. To identify a statistically significant (p < 0.05) change in weighted error or time to complete task under the different conditions, two sided paired t-test's were done. No statistically significant difference was measured in weighted errors when comparing SARPD under delay with not using SARPD under both no delay (p = 0.55) and delay (p = 0.50). Repeating for time to complete task, statistical significance was measured when comparing SARPD under delay with not using SARPD under both no delay (p = 1.7e - 4) and delay (p = 0.02). To conclude, we measured that using SARPD on average decreases the time to complete task by 19% when under delay while having no statistically significant change in errors.

V. DISCUSSIONS AND CONCLUSION

SARPD is the first predictive display developed for teleoperated surgical robots and has been shown to be an efficient system. On a consumer grade GPU, the tool-tracking is computed in real-time and the AR rendering pipeline is able to support two 1080p displays, each running above 30fps. By utilizing AR to show the prediction, no assumptions need to be made about the environment so SARPD can be applied to any calibrated teleoperated system under delay. SAPRD also on average decreases time to complete task while not affecting the number of errors. This is an expected result since the AR rendering should not assist when small, precise motions are made which is when the errors in our user study occur.

For future work, more studies need to be done on different surgical tasks with rigorous accuracy measurements and subjective data to understand cognitive load when operating with and without SARPD under various levels of delay. In particular, tasks with more environmental interaction. Furthermore, additions to the tracking algorithm must be done to incorporate endoscopic motions which is part of normal operating procedure. The rendering pipeline also needs to account for obstructions of the the slave-tools from environmental obstacles.

VI. ACKNOWLEDGEMENTS

We thank all the participants from the user study for their time, all the members of the Advanced Robotics and Controls Lab at University of California San Diego for the intellectual discussions and technical help, and the UCSD Galvanizing Engineering in Medicine (GEM) program and the US Army AMEDD Advanced Medical Technology Initiative (AAMTI) for the funding.



Fig. 8: Weighted error and completion time results from the user study. No statistically significant change in weighted errors is observed with SARPD, but a statistical decrease in completion time is observed when teleoperating under delay.

REFERENCES

- [1] M. Yip and N. Das, "Robot autonomy for surgery," *arXiv preprint arXiv:1707.03080*, 2017.
- [2] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035 – 2057, 2006.
- [3] A. R. Lanfranco, A. E. Castellanos, J. P. Desai, and W. C. Meyers, "Robotic surgery: A current perspective," *Annals of Surgery*, vol. 239(1), pp. 14–21, Jan 2004.
- [4] M. J. H. Lum, D. C. W. Friedman, H. H. I. King, R. Donlin, G. Sankaranarayanan, T. J. Broderick, M. Sinanan, J. Rosen, and B. Hannaford, "Teleoperation of a Surgical Robot via Airborne Wireless Radio and Transatlantic Internet Links," in 6th International Conference on Field and Service Robotics - FSR 2007, vol. 42 of Springer Tracts in Advanced Robotics, Springer, July 2007.
- [5] A. Kapoor, R. Kumar, and R. Taylor, "Simple biomanipulation tasks with 'steady hand' cooperative manipulator," *Lect Notes Comput Sc.*, pp. 141–148, 2003.
- [6] A. Schweikard, R. Tombropoulos, L. Kavraki, J. Adler, and J. Latombe, "Treatment planning for a radiosurgical system with general kinematics," *IEEE Intl. Conf. on Robotics and Automation*, pp. 1720–1727, 1994.
- [7] M. Anvari, C. McKinley, and H. Stein, "Establishment of the world's first telerobotic remote surgical service," *Annals of Surgery*, vol. 241(3), pp. 460–464, Mar 2005.
- [8] J. Marescaux, J. Leroy, M. Gagner, F. Rubino, D. Mutter, M. Vix, S. E. Butner, and M. K. Smith, "Transatlantic robot-assisted telesurgery," *Nature*, pp. 379–380, Sep 2001.
- [9] S. Xu, M. Perez, K. Yang, C. Perrenot, J. Felblinger, and J. Hubert, "Determination of the latency effects on surgical performance and the acceptable latency levels in telesurgery using the dv-trainer (Rsimulator," *Surg Endosc.*, pp. 2569–76, 2014.
- [10] C. Y. Nguan, R. Morady, C. Wang, D. Harrison, D. Browning, R. Rayman, and P. P. Luke, "Robotic pyeloplasty using internet protocol and satellite network-based telesurgery," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 4, pp. 10–14, Mar 2008.
- [11] W. R. Ferrell and T. B. Sheridan, "Supervisory control of remote manipulation," *IEEE Spectrum*, vol. 4, pp. 81–88, Oct 1967.
- [12] W. R. Ferrell, "Delayed force feedback," *Human Factors*, vol. 8, no. 5, pp. 449–455, 1966. PMID: 5966936.
- [13] R. J. Anderson and M. W. Spong, "Bilateral control of teleoperators with time delay," *IEEE Transactions on Automatic Control*, vol. 34, pp. 494–501, May 1989.
- [14] G. Niemeyer and J. E. Slotine, "Using wave variables for system anal-

ysis and robot control," in *Proceedings of International Conference on Robotics and Automation*, vol. 2, pp. 1619–1625 vol.2, April 1997.

- [15] M. C. Yip, M. Tavakoli, and R. D. Howe, "Performance analysis of a manipulation task in time-delayed teleoperation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5270– 5275, Oct 2010.
- [16] M. C. Yip, M. Tavakoli, and R. D. Howe, "Performance analysis of a haptic telemanipulation task under time delay," *Advanced Robotics*, vol. 25, no. 5, pp. 651–673, 2011.
- [17] W. Yoon, T. Goshozono, H. Kawabe, M. Kinami, Y. Tsumaki, M. Uchiyama, M. Oda, and T. Doi, "Model-based space robot teleoperation of ets-vii manipulator," *IEEE Transactions on Robotics and Automation*, vol. 20, pp. 602–612, June 2004.
- [18] T. Imaida, Y. Yokokohji, T. Doi, M. Oda, and T. Yoshikawa, "Ground-space bilateral teleoperation of ets-vii robot arm by direct bilateral coupling under 7-s time delay condition," *IEEE Transactions on Robotics and Automation*, vol. 20, pp. 499–511, June 2004.
- [19] A. K. Bejczy, W. S. Kim, and S. C. Venema, "The phantom robot: predictive displays for teleoperation with time delay," in *Proceedings.*, *IEEE International Conference on Robotics and Automation*, pp. 546– 551, IEEE, 1990.
- [20] D. Seita, S. Krishnan, R. Fox, S. McKinley, J. Canny, and K. Goldberg, "Fast and reliable autonomous surgical debridement with cable-driven robots using a two-phase calibration procedure," in 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 6651– 6658, IEEE, 2018.
- [21] L. Qian, A. Deguet, and P. Kazanzides, "Arssist: augmented reality on a head-mounted display for the first assistant in robotic surgery," *Healthcare Technology Letters*, vol. 5, no. 5, pp. 194–200, 2018.
- Healthcare Technology Letters, vol. 5, no. 5, pp. 194–200, 2018.
 [22] M. Ye, L. Zhang, S. Giannarou, and G.-Z. Yang, "Real-time 3d tracking of articulated tools for robotic surgery," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Cham), pp. 386–394, Springer International Publishing, 2016.
- [23] R. Hao, O. Özgüner, and M. C. Çavuşoğlu, "Vision-based surgical tool pose estimation for the da vinci® robotic surgical system," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1298–1305, IEEE, 2018.
- [24] A. Reiter, P. K. Allen, and T. Zhao, "Feature classification for tracking articulated surgical tools," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 592–600, Springer, 2012.
- [25] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da vinci (Rsurgical system," *IEEE Intl. Conf. on Robotics and Automation*, pp. 6434–6439, 2014.