Augmented Virtuality for Model-Based Teleoperation

Balazs Vagvolgyi, Wenlong Niu, Zihan Chen, Paul Wilkening, and Peter Kazanzides

Abstract— **Ground-based teleoperation of robots in space is subject to time delays of several seconds or more. This leads to the use of model-based approaches, where the operator interacts with a model (simulation) of the remote environment and the remote robot attempts to reproduce the results of that interaction. But, it is also desirable for the operator to view (delayed) images from the remote scene. These images, however, are often from one or more monocular cameras mounted on the robot end-effector, which leads to several other problems: unintuitive teleoperation due to the eye-in-hand configuration, limited field of view, and lack of stereo visualization. We present an augmented virtuality interface for teleoperation, which can solve these problems by projecting the real camera images onto a registered 3D model of the environment and allowing the operator to select any desired viewpoint. This approach is suitable when there is at least a partial model of the environment, as is the case for satellite servicing. The proposed method begins with a video survey to register the 3D model to the physical environment, followed by a user interface that presents a stereo visualization of the model, augmented by projections of real camera images onto the model. We quantitatively and qualitatively compare the augmented virtuality images to real camera images taken from the same viewpoint and perform experiments to evaluate the efficacy of the augmented virtuality paradigm for teleoperation. The results suggest that this approach can improve operator situation awareness, potentially leading to better performance, especially when the camera views are unintuitive or limited.**

I. Introduction

There is considerable economic value in repairing or refueling satellites on orbit to extend their service life. One approach is to send a robotic servicing spacecraft to autonomously rendezvous and dock with the target satellite, and then teleoperate the robot from the ground. However, ground-based teleoperation of an on-orbit robot introduces significant time delays, often on the order of several seconds. We have previously developed a model-based approach [1], [2] to overcome these time delay issues and demonstrated it in single-user and multi-user experiments on a ground-based platform consisting of the master console of a da Vinci robot and a Whole Arm Manipulator (WAM, Barrett Technology Inc., Newton, MA) emulating the remote servicing robot [3], [4]. This prior work focused on the use of virtual fixtures on the master console [1], sensor-based control for the remote robot [2], and model updates [5], [6].

One common feature in all the above work is that the visualization on the master console consisted primarily of delayed stereo video, with augmented reality overlays in some cases. There are, however, significant practical issues with this approach. Our ground-based test platform used a stereo camera on the remote robot, which is ideal when using the stereo viewer of the da Vinci console, but is not consistent with common practices in space robotics that rely primarily on multiple monocular cameras. For example, the NASA Remote Refueling Mission (RRM) employs orthogonal tool-mounted cameras. One notable exception is the ROTEX mission from the German Aerospace Center (DLR), which used stereo cameras [7].

Another practical issue is that a tool-mounted camera is not intuitive for users because humans are accustomed to having separate control of their eyes and hands. With an "eye-in-hand" setup, however, motion of the tool also causes motion of the camera. Furthermore, when operating close to the satellite, a tool-mounted camera has a limited field of view and thus its operator can lose situational awareness. One solution is to use an alternate camera placement (e.g., one mounted to the base of the robot), but the view from other cameras may not be ideal for the current task or may be blocked by the robot arm.

This paper proposes to solve the above issues via the use of an Augmented Virtuality interface. In particular, we propose to perform an initial survey of the satellite by acquiring camera images from different positions of the robot arm and to register these images to the CAD model of the satellite. This enables a Virtual Reality interface, where the operator can request arbitrary stereo views by moving a virtual stereo camera with respect to the model. But, we do not wish to ignore the real world and therefore incorporate the real video as an overlay (projection) in the virtual world, which is considered Augmented Virtuality as defined in [8].

Our method relies on the ability to visualize camera images from different viewpoints, which we achieve by projecting (texture-mapping) the images onto the 3D model surface. A similar approach has been presented for endoscopic camera visualization from alternate viewpoints [9], [10].

II. Hardware System Overview

This section gives an overview of the hardware system setup, including robots, the satellite mockup and cameras.

A. Teleoperation

The teleoperation task is performed using a position-to-velocity based master to slave control scheme. In this setup, the user controls the slave robot via one 7 degree-of-freedom master tool manipulator (MTM) from the first-generation da Vinci master console [11] (see Fig. 1(a)). The master robot is controlled using cisst/SAW software [12] with a Robot Operating System (ROS) [13] extension API. Video

B. Vagvolgyi, W. Niu, Z. Chen, P. Wilkening, and P. Kazanzides are with the Department of Computer Science, Johns Hopkins University, Baltimore, MD, USA, {balazs, wniu3, zihan.chen, pwilken3, pkaz}@jhu.edu. W. Niu is also with the National Space Science Center, Chinese Academy of Sciences.

feedback is either streamed from the video captured by a camera mounted to the slave robot's end-effector, or from the proposed augmented virtuality view detailed in the following section. As a comparison, for NASA Robotic Refueling Mission (RRM) tasks the operator currently commands the end effector with a keyboard and joystick using delayed video feedback from multiple monocular cameras [14].

The slave robot at the remote site is a 6 degrees-of-freedom UR5 (Universal Robots, Odense, Denmark), as shown in Fig. 1(b). For the experiments reported here, a marker is mounted in a spring-loaded socket affixed to the end effector, which enables the robot to draw on the satellite surface. A tool camera is mounted on the end effector, as shown in Fig. 1(b), with the marker in the field of view. The control of the UR5 is implemented using cisst/SAW software, which establishes a connection to the UR5's Real Time client, reads the current robot state and sends robot motion commands at 125 Hz. Similarly, a ROS interface is provided via a cisst-to-ROS bridge. This robot replaced the Whole Arm Manipulator (WAM), used in our previous research [1]–[6] because it provides higher accuracy, repeatability and a flexible programming interface.

B. Mock Satellite

To better simulate the real operating environment and facilitate testing, a scaled down satellite mockup has been designed and manufactured. As shown in Fig. 1(b), it consists of three major parts: the satellite body with a refueling port, the solar panel, and an antenna. The body is wrapped with non-aerospace-rated materials that emulate the specularity and stiffness of the MLI used to insulate satellites. The mockup has a dimension of 0.61 m x 1.06 m x 0.93 m.

C. Cameras Setup

In our camera setup, we tried to mimic a realistic configuration of cameras that could be used in an actual servicing mission. We have installed two deck cameras, one pan-tilt-zoom (PTZ) (HuddleCamHD, Downingtown PA) and one rigidly mounted wide angle camera (PointGrey Flee3, FLIR Integrated Imaging Solutions Inc. BC, Canada), and installed a compact, lightweight camera (PointGrey BlackFly, FLIR Integrated Imaging Solutions Inc. BC, Canada) on the slave robot's end effector, oriented towards the tool tip. All cameras are equipped with color sensors with a resolution of approximately 2 megapixels each, capable of capturing video at 30 frames per second. The deck cameras are mounted on the left and right of the robot to provide an overview of the workspace, while the tool camera captures a high resolution image of the immediate surroundings of the tool tip.

III. Augmented Virtuality System

This section describes the augmented virtuality visualization system, including the software components that implement the feature. Similar methods were employed by Paul et al. [15] in a surgical environment, where computational stereo techiques were used to reconstruct the model on which the live image was projected.

A. Calibration and Registration in a Remote Environment

Robotic interactions with objects that have unknown characteristics requires accurate sensing and kinematic control. In a controlled laboratory environment, this task is facilitated by readily available calibration instruments and human operators. However, in an unmanned space flight mission, there is little help available to fix an instrument if it goes out of alignment. Therefore, we developed a visual calibration and registration program that can operate without laboratory assistance, relying instead on cameras, robot kinematic information, and visible natural landmarks. The resulting program, developed for this project, is called Vision Assistant and is the hub for all visual processing and geometry registration.

The Vision Assistant relies on ROS for communication with cameras and other modules of the system. Its features include, but are not limited to, capturing images from multiple cameras, the calibration of camera intrinsic parameters, hand-eye calibration, geometry registration, camera pose estimation, and the calibration of camera extrinsic parameters. All of these functions are based on the detection of either a traditional checkerboard pattern or natural landmarks of any known geometry.

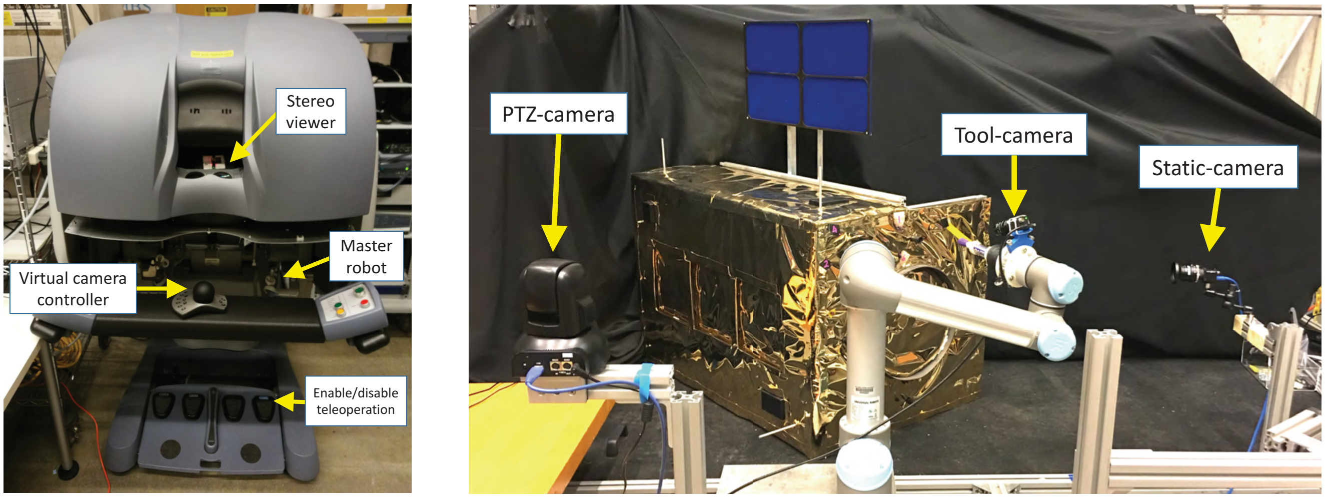
The software development process for the Vision Assistant is guided by the priorities of the NASA mission, but we plan to use this program in other applications as well. For the time being, the landmark detection on images requires an operator to pick the visible landmarks on the images. This is justified by the fact that real-time operation is not required in these missions, as they last several days and operators have hours to make decisions between every action. For real-time applications, we have a registration algorithm in development that will automatically detect features.

To gain an accurate registration of the target satellite with respect to the servicing robot, the operator must survey the satellite by moving the camera mounted on the robot arm to capture images from multiple view angles. These captured images are then used to perform a camera intrinsics calibration (optional if reliable ground-based values are available), hand-eye calibration, and a satellite registration. These actions can all be done within the Vision Assistant. This program publishes the results via ROS as they become available, which are then used by the visualization system to render accurate augmented virtuality views of the target satellite.

To achieve an accurate registration with our robot, we developed a multi-step optimization framework for calibration that combines some of the most widely used algorithms [16] [17] to produce a more reliable result. This calibration system was originally developed for the less-accurate WAM arm, and provided even higher registration accuracy with the UR5.

B. Visualization System

The visualization system comprises three main parts: 3D models, stereoscopic virtual cameras, and a texture projection capability. The entire 3D visualization pipeline is implemented within RViz, using both already existing RViz display plugins and custom-built ones.



(a) da Vinci Master Console and the 3D mouse.

(b) Universal Robot 5 and the three cameras mounted on the platform.

Fig. 1: A da Vinci Research Kit master console (left) is used to teleoperate a Universal Robot 5 (right) with the camera and the spring loaded pen holder mounted on the end-effector.

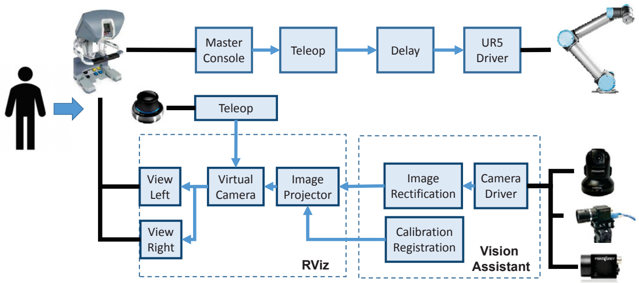


Fig. 2: Block diagram of system. Reported experiments were performed without introducing delay (beyond standard network delay).

The 3D model of the robot was provided as part of the ROS Universal Robot package, and our mock satellite model was converted from SolidWorks to the Collada 3D format and loaded into ROS using a URDF descriptor. This satellite model includes only the hard metal and plastic parts, and does not include the multi-layer insulation (MLI) blanket.

The stereoscopic virtual cameras were created using two of RViz's built-in camera display plugins, configured with a solid black background and orientations linked to the virtual camera pose specified in the ROS transformation tree. The left camera pose is directly linked to the virtual camera pose, while the right camera is offset by 5 cm in the X direction in order to correspond to the right eye view. The pose of these two cameras can be freely controlled through TF, the ROS transformation manager system. Our graphics system is set up so that the left camera view is routed through the VGA output of the graphics card to the left eyepiece of the da Vinci master console, while the right camera view is similarly routed to the right eyepiece, accomplishing the desired stereoscopic effect.

RViz does not provide dynamic texturing features by default, but custom RViz plugins may access lower level capabilities provided by the underlying Ogre 3D (Open source GRaphics Engine) package, which provides an object-oriented abstraction to low-level GPU hardware features. In our texture projection plugin, we took advantage of this functionality to add a second rendering step to the RViz rendering loop. This extra step renders the scene from the point of view of the projector to gain access to the texture coordinates of all visible faces in our 3D models, then assigns new values to the textures based on the image projection. As a result, a second textured material layer is added to the 3D model that mimics the appearance of an image being projected on the model. The method is completely dynamic, and both the projected image and the pose of the projector can be changed at any time. The drawback of this approach is that for every displayed frame the GPU needs to render the image twice; however, on modern graphics hardware it does not result in significant slowdown.

During teleoperation, the pose of the virtual cameras are controlled by a transformation published in the ROS TF system by our 3D joystick controller ROS node, which performs a remapping of the raw 6 DoF joystick inputs into the visual frame of the virtual cameras. As a result, the operator can freely 'fly' with the virtual camera to arbitrary positions in the 3D scene.

The texture projection plugin requires two inputs, one for the live image, and another for a ROS TF frame identifier. The TF frame provides the pose of the projector in the 3D scene, while the live images are mapped onto the 3D scene using the texture projection method. For accurate visualization, the pose of the projector needs to correspond exactly to the pose of the physical camera that captured the live images, and the projector's intrinsic parameters must also match the camera's intrinsics. The user interface of the texture projection plugin enables switching between cameras and poses at any time without restarting the system.

IV. Experiments

We performed two sets of experiments to evaluate our augmented virtuality teleoperation system. The first aimed to measure visualization accuracy by comparing photos to computer generated renderings, and the second aimed to validate the concept in a small-scale user study with our prototype teleoperation system. In the user study, we compare our visualization system to the existing teleoperation method in which the viewpoint is restricted to the physical cameras (usually, the tool camera). The teleoperation system in the user study did not include time delay.

Our system matches the performance of the existing method if we use a single virtual camera and limit its position to that of the tool camera, since in this case the virtual camera's view will match the original image pixel-by-pixel. Therefore, in these experiments we only need to prove that the accuracy of the camera view renderings and the freedom to change the virtual view to arbitrary poses enhance the user's ability to teleoperate. This means that visualization inaccuracies in virtual views provided to the user need to be weighed against the usefulness of having these camera views available.

A. Evaluation of Visualization Accuracy

There are multiple unknowns with regards to the appearance of the satellite to be serviced. One is its relative pose with respect to the optical sensors of the servicing spacecraft, which is defined by a single rigid 3D transformation. Another set of unknowns concern the MLI blanket covering the satellite body. The exact shape, thickness, color, coverage of the blanket, and the location of seams and patches on it are not recorded before launch, therefore there is no computer model to describe them. The MLI blanket was applied to the satellite following engineering guidelines, which lets us approximate the shape, thickness, and coverage, but we have no guarantee of accuracy for these parameters. For example, the color of the top Mylar layer is gold when new, but may change its hue after lengthy exposure to the extreme environmental conditions in space, including cosmic rays.

The last major unknown is the nature of the reflections visible on the Mylar. From the aspect of visual evaluation this poses the greatest difficulty, as the reflections dominate our field of view while they contribute little to our understanding of pose and context. In fact, the reflections from unknown and uneven surfaces, wrinkles, and patches distort the reflected environment and disorient the observer. To further complicate the matter, reflections observed at a given location on the MLI surface will have different appearances when viewed from different cameras, and even change in time as the satellite orbits Earth and the robotic arm moves. Barring other unknowns, the pose itself would be sufficient to estimate the appearance of the satellite from any of the cameras with high accuracy, given a 3D CAD model. However, the problems introduced by the MLI blanketing significantly complicate the evaluation.

For the evaluation, we took photos of our mock satellite from five distinct locations. We used the two deck cameras for two of the locations and used a hand-held mobile phone camera to take pictures from three other suitable locations. The two deck cameras had known intrinsics, and the mobile phone camera intrinsics were calibrated using the Camera Calibration Toolbox for Matlab. The camera extrinsics for each picture were determined by pose estimation based on the satellite's manually selected natural landmarks. The photos taken from these five locations provide the ground truth for our evaluation, against which we compare the augmented virtuality renderings of the 3D model from the same locations. The tool camera's live video feed was projected onto the satellite model for the renderings. For each of the five locations we moved the robot (and the tool-camera mounted on it) to multiple poses and recorded the renderings. This yielded multiple rendered images for each of the five photos, which we then evaluated. Altogether we had 30 rendered images for 5 ground truth photos.

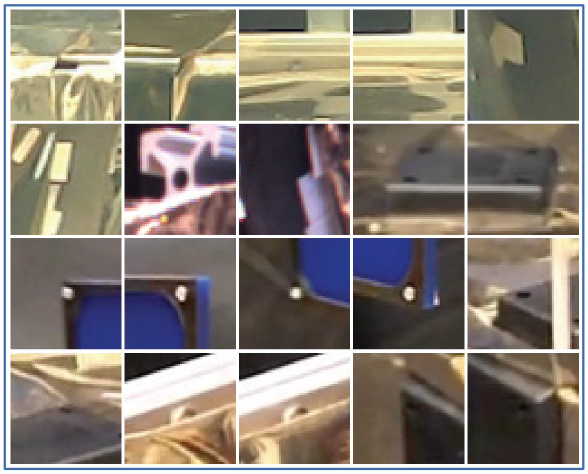


Fig. 3: Natural landmarks selected for quantitative visual evaluation.

We present a qualitative evaluation by comparing the real photos of the satellite from various angles with the corresponding rendered views. For a more quantifiable evaluation, we then compared the locations of natural landmarks on the satellite not covered by blanketing.

B. Visualization Accuracy: Quantitative Evaluation

Due to the difficulties of predicting the appearance of the MLI blanket, the most suitable approach for a quantitative evaluation was to use a set of natural landmarks on the satellite that are not covered by the blanket, and therefore clearly identifiable from any viewing angle. The landmarks were manually located on each image to maximize accuracy. The chosen landmarks (Fig. 3) are point-like, so that we can precisely pinpoint their 2D coordinates. We decided against using lines or other contours for evaluation, as it was too difficult to precisely define their locations on the images.

Overall, we manually identified 94 observations of 20 landmarks. The landmark locations were recorded on the ground truth images and the augmented virtuality renderings, and the 2D Euclidean distances between the ground truth and the observations were calculated. These distances represent

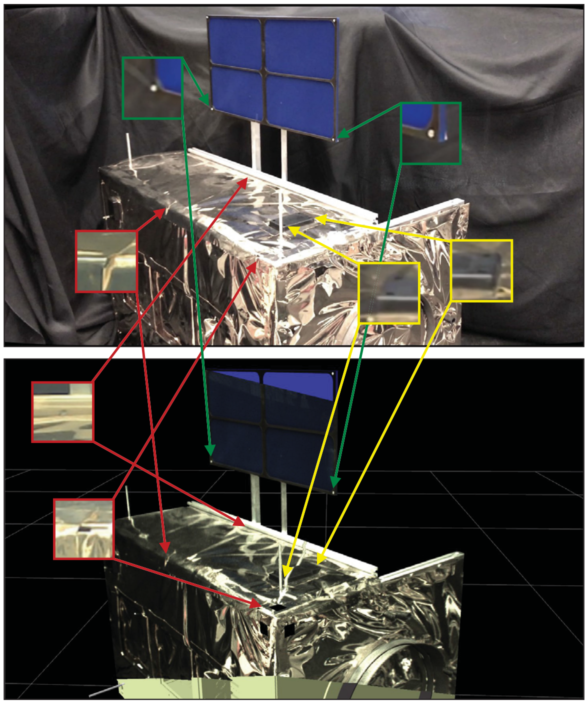


Fig. 4: Matching natural landmarks identified on both the photo and the rendered view.

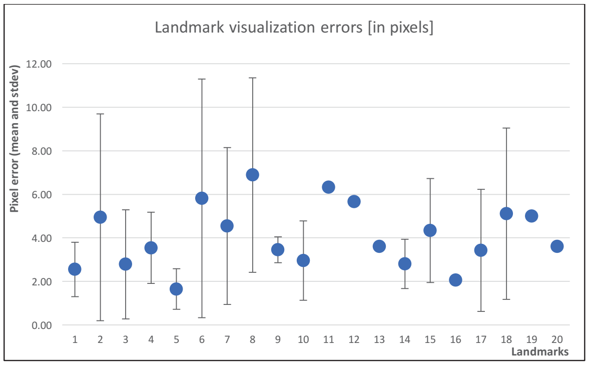


Fig. 5: Mean pixel errors and corresponding standard deviations for the 20 chosen natural landmarks.

visual rendering errors. The mean errors for each landmark and the corresponding standard deviations are shown in Fig. 5 and the distribution of all errors in the 2D image plane is visualized in Fig. 6. The mean error for all 94 observations was 3.95 pixels with a standard deviation of 3.22 pixels.

We used four different cameras to capture the images, and the image dimensions varied between 1600x1200 and 1900x1200. Additionally, we had to rescale all our images to 80% of their original sizes to make the pixel-to-pixel comparison possible.

C. Visualization Accuracy: Qualitative Evaluation

Visualization errors, flaws, and inaccuracies have taken multiple forms in our virtual camera views. We have grouped them into three distinct classes (Class I-III). Class I errors are caused by simply having inaccurate camera extrinsics or an inaccurate satellite pose registration, due to either inaccurate robot kinematics or a mismatch between the physical satellite model and the CAD model. These will result in a shift of the projected texture and may cause some texture features to appear a few pixels away from their real locations. Some examples of Class I errors are shown in Fig. 7.

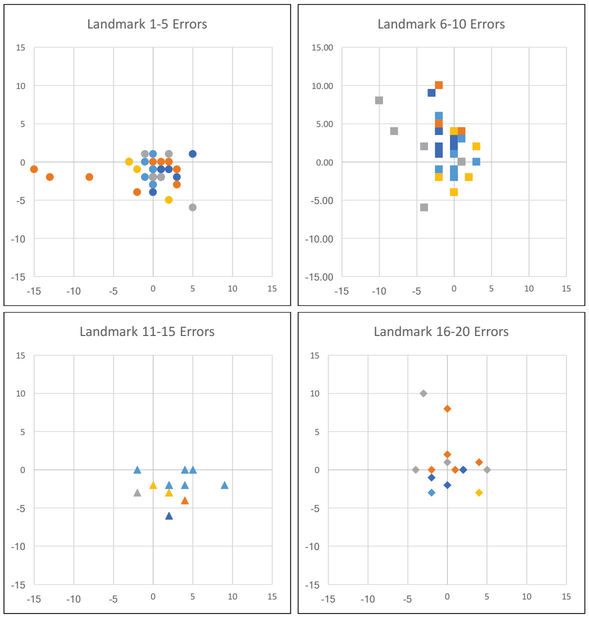


Fig. 6: Pixel error distributions for the 20 natural landmarks in the image plane.

Class II of the qualitative errors are caused by the unknown geometry and reflections of the MLI blanket. Since the exact thickness and shape of the blanket is uncertain, we cannot accurately model how much of the satellite it covers. This may result in some parts of the 3D model being inaccurately textured by the image of the blanket. On the top Mylar layer, the position of an object's reflection depends on the position and orientation of the blanket, and the reflected shape can become severely distorted by wrinkles in the MLI. Reflections appear rather natural when observed in the camera's original image, however they are potentially confusing and distracting when rendered from a virtual camera view. Human vision expects the reflection to be between the observer and the object, and so when we move our viewing angle, we expect the reflection to move as well. But in the case of the virtual camera this is not true, as the reflection will always stay between the object and the physical camera's location. Examples of Class II visualization errors are shown in Fig. 8.

Class III errors are visualization imperfections caused by 3D rendering flaws. The most significant issue in the current rendering pipeline is that our image projection method is not able to visualize shadows. When the image is projected onto the 3D model of the satellite from the location of the camera, the image is mapped not only on the front facing polygons of the model, but also the back facing ones and polygons that should be blocked from being projected onto by other objects. We are working on fixing this issue in the next iteration of the system. Fig. 9 shows a few examples of Class III errors.

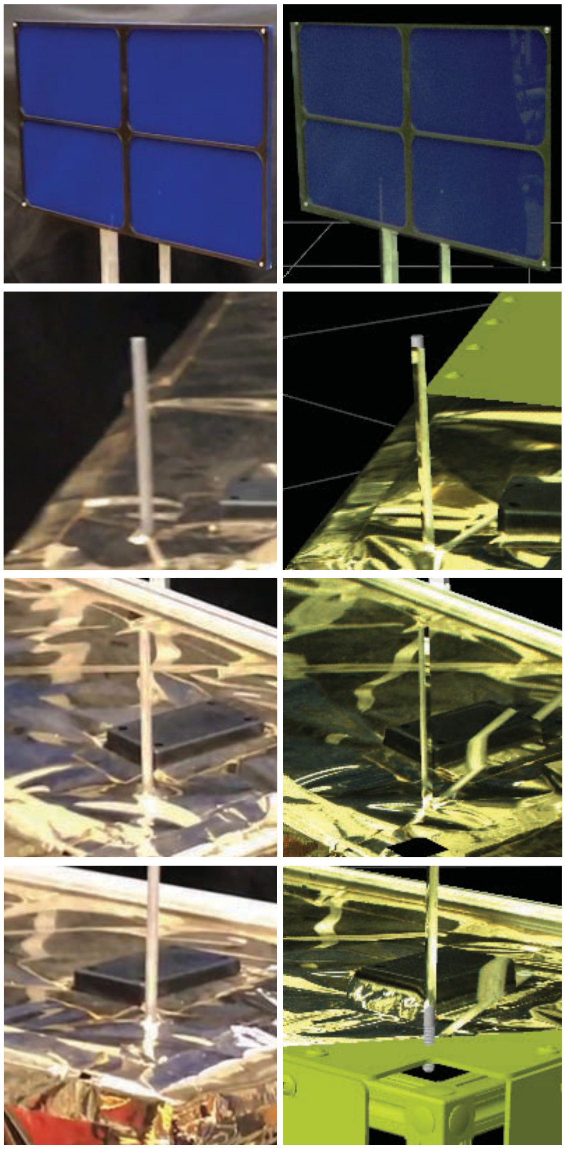


Fig. 7: Examples of Class I visualization errors: photos on left, 3D renderings on right. Notice the slight texture alignment errors at the corners of the solar panel and on the antennas.

D. Teleoperation Study

We compared the conventional teleoperation mode with our augmented virtuality teleoperation system in a small-scale user study. The users were all contributors in related projects, thereby they had a basic understanding of the concept of teleoperation. Most also had some experience with the da Vinci surgical system, and thus were familiar with the master console used in the experiment. In total, six participants completed all of the tasks outlined below.

Our goal was to simulate the task of MLI blanket cutting, which is one of the most difficult tasks during satellite servicing due to the unpredictable appearance and behavior of the blanket, by using the marker to draw on the MLI.

The experiment consisted of two teleoperation modes. The *conventional* teleoperation mode displayed the live feed from the monocular tool camera to both eyes in the da Vinci

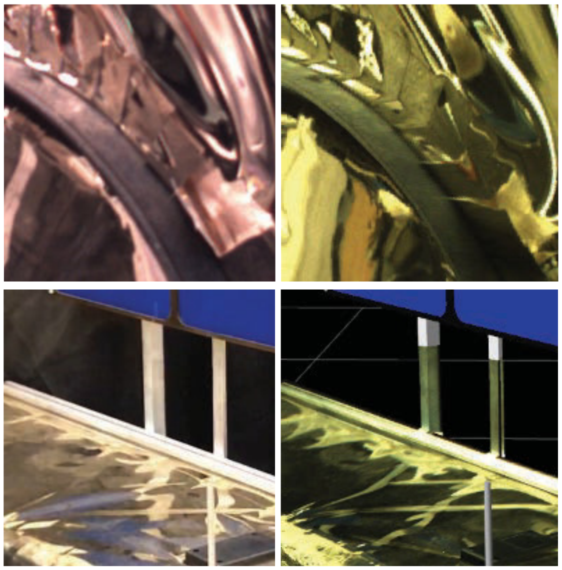


Fig. 8: Examples of Class II visualization errors. Photos on the left, 3D renderings on the right. Top row: the thickness of the MLI blanket did not match the satellite model; Bottom row: The reflections of the solar panel mount do not point toward the virtual camera.

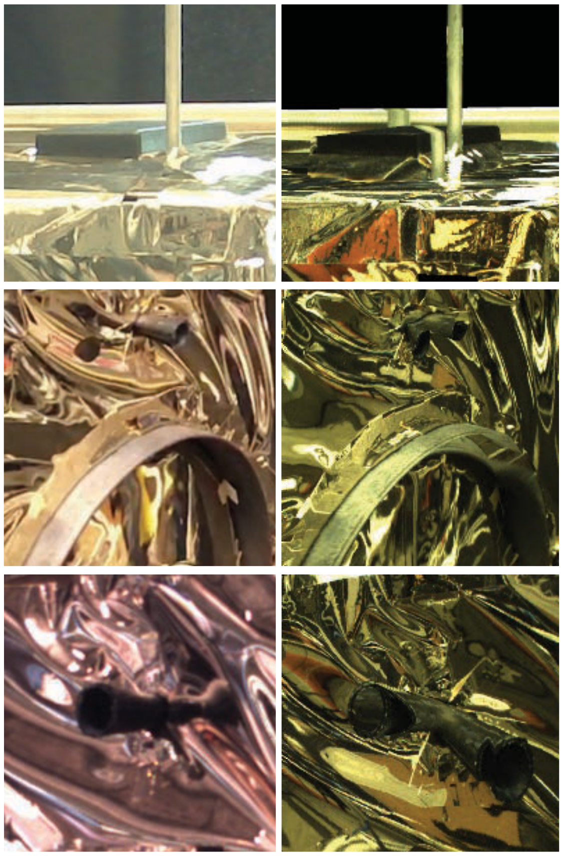


Fig. 9: Examples of Class III visualization errors. Photos on the left, 3D renderings on the right. The antenna images are projected behind the antennas, rather than casting a shadow; Similar effects noticeable near the thruster and marman ring.

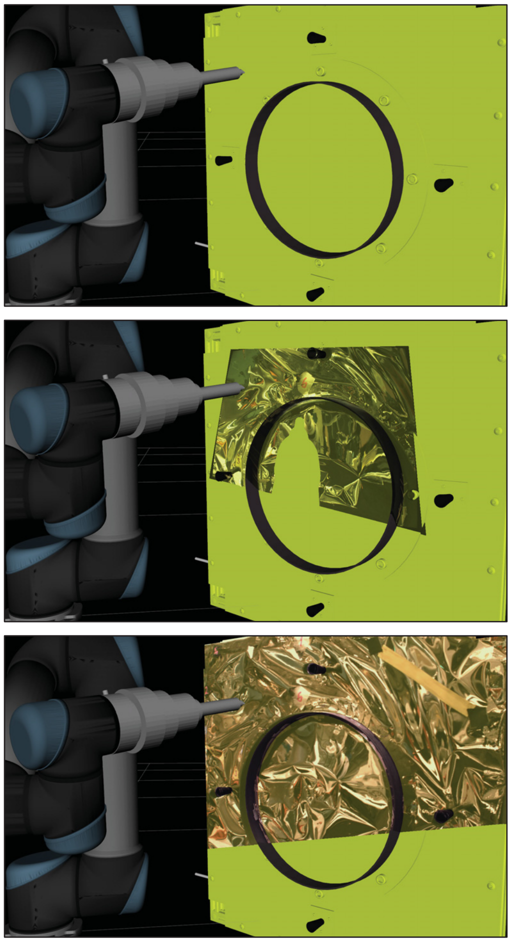


Fig. 10: Teleoperation study: Different camera projection modes. Top: no camera projection; center: tool camera with the pen-mask; bottom: wide angle deck camera.

master console and mapped the slave robot's motions to match the motions of the master with respect to the tool camera's orientation. The user had the option to stop the robot at any time and ask the instructor to switch from the tool camera view to the deck camera view, which provided a much wider angle perspective on the scene. Typically, users switched back to the tool camera view before resuming teleoperation.

In the *augmented virtuality* mode, the user was presented a stereoscopic rendering of the computer model of the scene and given a 6-DoF 3D mouse to control the virtual camera's pose by 'flying' it to any position in the virtual 3D space. The 3D mouse inputs were mapped into the virtual camera's frame, following the first-person control paradigm. In the future, the ability to reposition the camera can be implemented within the da Vinci master console, as currently done to reposition the endoscope during robotic surgery. The user was given the option to display any of the three live camera feeds (one at a time), but this time the videos were not presented in flat windows in the image plane, but were projected onto the satellite's 3D model, as explained in Section II. This augmented virtuality rendering with different camera projections is shown in Fig. 10 and the accompanying video.

The experiment started with each participant being explained the details of the study, including the handling of the master console, the properties of the visualization system, and the tasks to be performed. The users then practiced both modes of teleoperation by using each to draw a straight line within the confines of a yellow strip of tape on the mock satellite surface.

After the users became familiar with the two modes of teleoperation, the slave robot was set up in a neutral position and the users were asked to locate 3 markers on the surface of the satellite, indicated by nearby numbers, and draw a triangle between the 3 markers in a specified sequence. The markers were ordered such that the distance between each traversed pair was about 2 to 3 times larger than the coverage of the tool camera's field of view. For each line, the users were told to use all available camera views and the workspace of the arm to survey the indicated area, locate the next marker, and draw a straight line to it from the previous marker. Their execution times were recorded and a picture was taken of each final drawing. This drawing experiment was repeated as 4 distinct tasks (Tasks I-IV).

Task I was to perform a drawing using the conventional mode, and Task II was to perform a new drawing with a different set of markers in the augmented virtuality mode. For Task III and Task IV the instructor rotated the camera by 90 degrees to disorient the users. The point of this rotation was to emphasize that in space there are no dedicated directions, and so a proper evaluation of teleoperation methods must be performed using arbitrary orientations. Task III was to perform a drawing with this new orientation in the conventional mode with a third set of marker IDs, and Task IV was to repeat the drawing procedure with this new orientation in the augmented virtuality mode with a fourth set of marker IDs. The drawings were all digitized and are shown in Fig. 11.

The average execution times are shown in Table I. The times were comparable between the conventional and augmented virtuality modes, with the latter taking slightly longer. Although these measurements are noisy due to some technical difficulties encountered while running the software that caused slow-downs and sometimes required us to the restart the master-slave link, we plan to address these issues and conduct a wider scale human subjects study in the future.

TABLE I: Teleoperation study: task execution times, in minutes, per task and teleoperation mode.

Task	Mode	Mean (min)	Std. Dev. (min)
I	Conventional	7:37	2:16
II	Augmented Virtuality	8:32	5:10
III	Conventional	7:49	3:25
IV	Augmented Virtuality	8:11	5:20
I and III	Conventional	7:43	2:46
II and IV	Augmented Virtuality	8:22	5:00

These preliminary results confirm our initial intuition that a virtual camera provides better contextual understanding of the workspace and gives users a better sense of direction, allowing them to draw straighter lines to markers that are outside the tool camera's field of view with execution time comparable to conventional teleoperation.

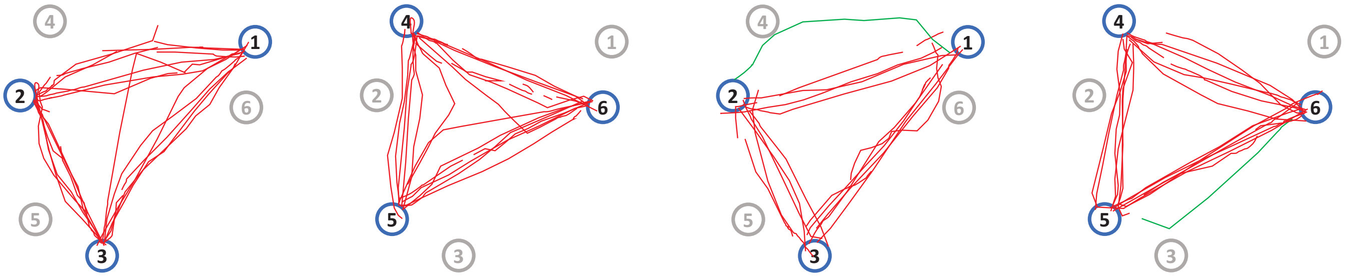


Fig. 11: Teleoperation study: marker locations and digitized paths drawn by study participants. From left to right: Task I (Conventional); Task III (Conventional); Task II (Augmented Virtuality); Task IV (Augmented Virtuality). Red lines represent deliberate paths, green lines represent paths that were drawn as a result of the user mishearing one of the marker IDs.

V. Conclusions and Future Work

We presented an augmented virtuality interface for telerobotic control in applications where a model of the environment is available. One such scenario is time-delayed teleoperation, where model-based control methods are often used to ameliorate the effects of time delay. The proposed teleoperation paradigm improves situational awareness by enabling the operator to select arbitrary stereo views of the model, which are augmented by projections of the live camera views. This is especially advantageous in cases where the real camera views are limited; for example, when a monocular camera is mounted on the robot end-effector. We evaluated the augmented virtuality visualization both quantitatively and qualitatively and demonstrated its use in a teleoperation scenario that is representative of a satellite servicing task; specifically, the remote robot was teleoperated to draw a specified pattern on the satellite insulation, as a surrogate for cutting the insulation in preparation for a refueling or servicing operation. The experiments, with no time delay, demonstrated that the proposed approach can improve operator performance. Task completion time was slightly longer, but we anticipate that this will be reduced as the implementation is optimized.

We are currently working on experiments that include several seconds of time delay, in which case the visualization includes a predictive display of the virtual robot in addition to the virtual robot based on the delayed telemetry. Furthermore, the quality of the 3D visualization will be improved by enabling multiple camera projections and eliminating rendering artifacts, such as the texture back-projections.

Acknowledgments

This work was supported by NASA NNG14CR58C and NNG15CR66C. Anton Deguet assisted with software development. Shuyang Chen and Jiajun Li designed and built the camera and tool attachments for the UR5 robot. The da Vinci Research Kit (dVRK) is supported by NSF NRI 1637789.

References

[1] T. Xia, S. Léonard, A. Deguet, L. Whitcomb, and P. Kazanzides, "Augmented reality environment with virtual fixtures for robotic telemanipulation in space," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Oct 2012, pp. 5059–5064.

[2] T. Xia, S. Léonard, I. Kandaswamy, A. Blank, L. Whitcomb, and P. Kazanzides, "Model-based telerobotic control with virtual fixtures for satellite servicing tasks," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2013, pp. 1479–1484.

[3] S. Vozar, S. Leonard, P. Kazanzides, and L. L. Whitcomb, "Experimental evaluation of force control for virtual-fixture-assisted teleoperation for on-orbit manipulation of satellite thermal blanket insulation," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2015, pp. 4424–4431.

[4] S. Vozar, Z. Chen, P. Kazanzides, and L. L. Whitcomb, "Preliminary study of virtual nonholonomic constraints for time-delayed teleoperation," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2015, pp. 4244–4250.

[5] X. Li and P. Kazanzides, "Parameter estimation and anomaly detection while cutting insulation during telerobotic satellite servicing," in *IEEE/RSJ Intl. Conf on Intelligent Robots and Systems (IROS)*, 2015, pp. 4562–4567.

[6] ——, "Task frame estimation during model-based teleoperation for satellite servicing," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2016, pp. 2834–2839.

[7] G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl, "Sensor-based space robotics-ROTEX and its telerobotic features," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 5, pp. 649–663, 1993.

[8] P. Milgram and F. Kishino, "A taxonomy of mixed reality visual displays," *IEICE Transactions on Information and Systems*, vol. 77, no. 12, pp. 1321–1329, 1994.

[9] D. Koppel, Y.-F. Wang, and H. Lee, "Image-based rendering and modeling in video-endoscopy," in *IEEE Intl. Symp. on Biomedical Imaging: Nano to Macro*, 2004, pp. 269–272.

[10] Y. Koreeda, S. Obata, Y. Nishio, S. Miura, Y. Kobayashi, K. Kawamura, R. Souzaki, S. Ieiri, M. Hashizume, and M. G. Fujie, "Development and testing of an endoscopic pseudo-viewpoint alternating system," *International Journal of Computer Assisted Radiology and Surgery*, vol. 10, no. 5, pp. 619–628, 2015.

[11] 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® surgical system," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2014, pp. 6434–6439.

[12] A. Deguet, R. Kumar, R. Taylor, and P. Kazanzides, "The *cisst* libraries for computer assisted intervention systems," in *MICCAI Workshop on Systems and Arch. for Computer Assisted Interventions*, Midas Journal: http://hdl.handle.net/10380/1465, Sep 2008.

[13] "ROS.org | powering the world's robots." [Online]. Available: http://www.ros.org/

[14] L. Jenner, "NASA tests new technologies for robotic refueling," Feb. 2014. [Online]. Available: http://www.nasa.gov/content/goddard/nasa-tests-new-technologies-for-robotic-refueling

[15] P. Paul, O. Fleig, and P. Jannin, "Augmented virtuality based on stereoscopic reconstruction in multimodal image-guided neurosurgery: Methods and performance evaluation," *IEEE Transactions on Medical Imaging*, vol. 24, no. 11, pp. 1500–1511, 2005.

[16] R. Y. Tsai and R. K. Lenz, "Real time versatile robotics hand/eye calibration using 3D machine vision," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 1988, pp. 554–561.

[17] A. Malti, "Hand–eye calibration with epipolar constraints: application to endoscopy," *Robotics and Autonomous Systems*, vol. 61, no. 2, pp. 161–169, 2013.