Improving Remote Environment Visualization through 360 6DoF Multi-sensor Fusion for VR Telerobotics

Austin Sumigray*
Brown University
Providence, Rhode Island, USA

Eliot Laidlaw*
Brown University
Providence, Rhode Island, USA

James Tompkin
Brown University
Providence, Rhode Island, USA

Stefanie Tellex
Brown University
Providence, Rhode Island, USA

Figure 1: A scene inside the VR headset of a robot teleoperator, showing the 360° environment, point cloud from depth sensor, and rendered robot model. The operator controls the robot remotely by requesting arm and gripper poses using VR controllers.

ABSTRACT

Teleoperations requires both a robust set of controls and the right balance of sensory data to allow task completion without overwhelming the user. Previous work has mostly focused on using depth cameras, yet these fail to provide situational awareness. We have developed a teleoperation system that integrates 360° stereo RGB camera data with RGBD sensor data. We infer depth from the 360° camera data and use that to render a VR view, which allows for six degree of freedom head motion. We use a virtual gantry control mechanism, and provide a menu with which the user can choose which rendering schemes will render the robot's environment. We hypothesize that this approach will increase the speed and accuracy with which the user can teleoperate the robot.

*Both authors contributed equally to this research.

HRI '21 Companion, March 8–11, 2021, Boulder, CO, USA

ACM ISBN 978-1-4503-8290-8/21/03…$15.00
https://doi.org/10.1145/3434074.3447198

KEYWORDS

Virtual reality, telerobotics, robotics.

ACM Reference Format:
Austin Sumigray, Eliot Laidlaw, James Tompkin, and Stefanie Tellex. 2021. Improving Remote Environment Visualization through 360 6DoF Multi-sensor Fusion for VR Telerobotics. In *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction (HRI '21 Companion), March 8–11, 2021, Boulder, CO, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3434074.3447198

1 INTRODUCTION

In many environments that are too dangerous or difficult for humans to physically be in, robots are able to complete the tasks that humans cannot. Situations ranging from bomb defusal to surgeries to exploring another planet can all benefit from use of a robot. While autonomous control for robots has greatly improved over the last few decades, humans remain far better at many tasks. Robot teleoperation by a remote human allows for both risk-free interaction with dangerous environments, and the intelligence, experience, and dexterity that a human operator provides. However, to perform a task with speed and accuracy, an operator requires both an intuitive set of controls and the situational awareness to use them.

Control interfaces are often 2D [7], particularly for tasks like motion planning and item grasping. Monitors, keyboards, and mice

allow for control in 3D space via 2D projection. However, such methods are often difficult to use with precision because they do not accurately represent how humans interact with our 3D world[17]. Having multiple camera views or a point cloud display is possible, but mapping the 3D world to a 2D display, and 2D inputs to 3D robot actions is inherently difficult.

As a contrast to 2D interfaces, working with a robot physically in view can provide the situational awareness needed. However, control suffers due to the required mapping from using a joystick or other controller to robot actions like moving an arm. Using a physical gantry [4] with the same degrees of freedom of the robot or a 3D motion controller "virtual gantry" [18] device can aid in this issue; however, if the scene is not from the same perspective then the control can still be unintuitive. Putting these issues together leads naturally to controlling the robot from within a virtual reality environment, where many operator actions can map more naturally to robot actions. Performing such a task remotely raises the challenge of populating the 3D VR environment with a compelling representation of the robot's environment. This representation is limited by the available sensing capacity of the robot and network bandwidth required to transmit the sensory data.

Most work with VR systems in the past has been conducted either over a low latency and high bandwidth internal network, or by using low fidelity environments such as solely point-cloud-based methods and additional haptic feedback [9]. Low-fidelity environments limit the operator in the tasks that can be performed, and do not allow sufficient situational awareness to allow the operator to move the robot easily within the space.

We propose a system that uses data-efficient methods and multiple sensors to enable VR teleoperation over a common Internet connection. The operator is placed into a virtual environment constructed in Unity. Here, the state of the robot can be seen on a digital double, with its joint state information being updated in real time. The user can move the VR controller to a world position and command the arm's end effector to move to that same position. The virtual environment allows for the operator to both move around and see every angle of the robot, but also to move the hand to the exact position they place their own in. In addition, the environment shows the state of the world around the user using a combination of multiple sensing and scene reconstruction methods.

2 RELATED WORK

Many teleoperation systems use a monitor to display information about the robot's world to the user. Telepresence robots like the Beam from Suitable Technologies [2] often stream one or more video feeds from cameras mounted on the robot to the user's screen. These give some situational awareness, but with no depth perception. Other systems attempt to mitigate the depth perception issue by rendering a point cloud onto the operator's 2D screen [19]. The user must manually change the viewpoint if they wish to exploit the 3D benefits of a point cloud.

Static camera viewpoints can lead to difficult occlusions and parts of the scene that are outside the field of view. Valiton and Li [16] and Rakita et al. [13] begin to solve this issue by having hardware and software which provide moving viewpoints. While these works represent significant advances, they still fail to portray depth, making control more difficult. They also only show a small portion of the robot's environment to the user. We believe that a full 360° view is important for safe and efficient control of a moving robot. One natural technology to use for 3D operation that can also provide binocular stereo and 360° viewing is virtual reality.

VR systems are now inexpensive [1] and provide potential advantages over 2D systems [17]. A common approach is to mount two cameras as the robot's 'eyes' and stream video to each corresponding eye of the user's VR display [14] [5] [3]. This provides depth cues from stereo disparity, and can allow the user to view the robot's environment in a wide range of directions. However, latency between user and robot head movements can cause confusion or nausea [11]. It is possible to transform the video feed based on the current head pose [5], or use a 360° camera[10] to improve situational awareness. Thatte and Girod [15] demonstrated the importance of six-degree-of-freedom (6DoF) viewing in VR, where the user experiences both binocular disparity *and* motion parallax.

A common 6DoF alternative is a point cloud rendered via data from an RGB-D camera [20] [17] [8]. This provides real-time rendering of novel viewpoints that are both rotated and translated from the true camera pose. The downsides are that they often have many occlusions and only show a small field of view. Whitney et al. [17] also positioned the RGB-D camera separate from the robot, which limited the ability for robot movement. We use point clouds from RGB-D cameras mounted on a robot, but integrate them with an immersive view synthesis system for greater situational awareness.

VR based control systems have shown improvements in user comprehension and performance [17] [18]. Methods include homunculus-based methods which place the user in a virtual control room [6] as well as taking advantage of a robot being in a fixed location [20]. We aim to both allow for the advantages of a VR mimicry-based system while also providing an immersive enough environment for the user to control a movement-capable robot.

3 METHOD

We have designed a system that provides the user an immersive 6DoF VR teleoperation experience. We collect sensor data from multiple sources, transmit it over the network in real time, and use it to reconstruct the robot's environment in VR. The user then uses this environment reconstruction and our intuitive control interface to perform manipulation tasks or move the mobile robot.

Platform and sensing. Our environment is currently implemented on a Kinova MOVO robot and controlled using ROS. For 360° imaging, we use an Insta360 Pro 2. We use a Kinect V2 mounted on the head of the MOVO as well as a RealSense D435 RGB-D camera mounted on the robot's wrist (Figure 3). We also have a Velodyne Puck mounted on the robot underneath the Insta360 Pro 2 that we hope to integrate with our scene reconstruction system. For the VR side of the system, we use Unity with SteamVR, which has allowed us to use both the HTC Vive and Oculus Quest in development and use with the potential for additional controls.

Remote environment reconstruction. The first aspect of the environment is the robot's general surroundings in all directions. For this aspect, we have implemented the MatryODShka algorithm. This algorithm can convert a left-right stereo pair of 360° ODS

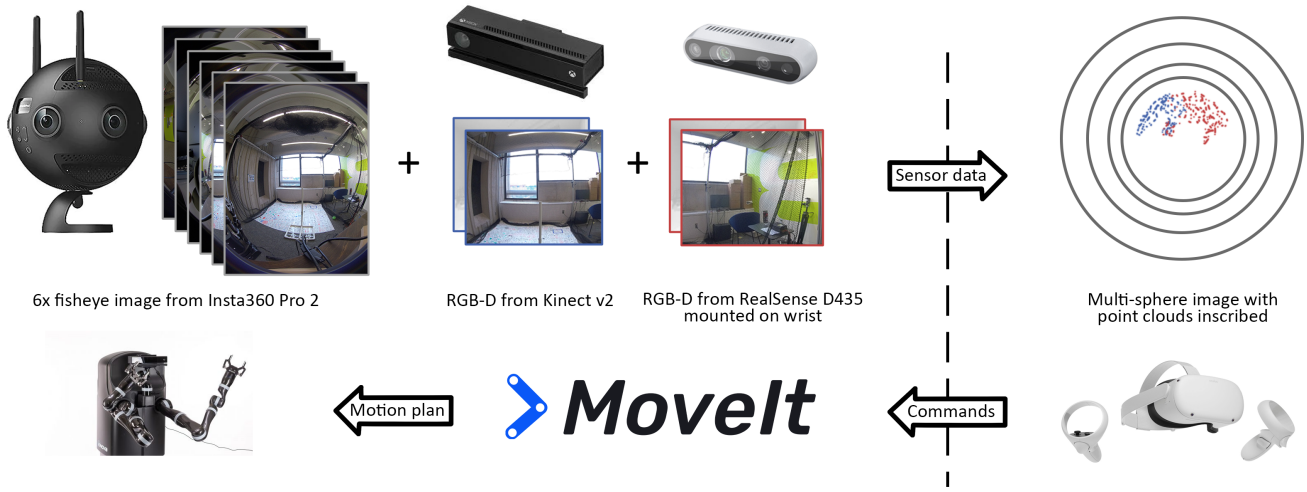

Figure 2: Our teleoperation system fuses sensory input from multiple sources into an immersive VR environment for robot teleoperation. Commands are sent back over the network and planned into robot motion.

Figure 3: The MOVO robot with ODS camera, Kinect, and point cloud wrist camera.

images into a series of RGBA concentric sphere layers (Figure 2, right), called a multi-sphere image (MSI). MSIs provide a sense of depth and motion parallax to the user, as the spheres are transparent except at the depths at which objects in the scene appear. For this, the algorithm constructs a pair of sphere sweep volumes at fixed depths, then converts these to into alpha (transparency) values per layer through a deep convolutional network. Given the MSI, the pose of the rendered view is updated in real time based on the VR headset's 6DoF pose within the sphere, which increases user comfort and reduces disorientation. As the spheres are computationally cheap to render, they give the operator 360° 6DoF viewing in real time independently of the CNN inference speed. We have also experimented with collapsing these spheres into one 'spherical' mesh that is deformed according to the inferred depth.

While the MatryODShka algorithm provides 360° situational awareness to the user, it struggles to reconstruct high resolution depth at close range. To mitigate this, there are also two point clouds in the scene. The head-mounted Kinect camera and wrist-mounted camera each provide RGB-D data which can be rendered either as a point cloud or sparsely connected mesh. The point clouds are positioned in the scene based on the robots current pose.

The final aspect of the virtual environment we consider is how to display the robot state itself. At all times, we render the URDF of the robot in its current state in the center of the scene.

Transmission. We compress all image data prior to transmission, for which we use ROS Bridge. We have also experimented with the Parsec SDK [12], which provides a system for low latency and high-resolution video transmission. In general, sending multiple high-resolution sensor feeds requires more bandwidth than a limited 2D view, and this can be challenging over a slower network such as a standard home network or any bandwidth-limited location such as a dangerous or uncontrolled environment.

Control interface. We have implemented end effector controls using the 6DoF VR controller, and in this we include both input and visual feedback. The user sees the 3D model of the robot end effector inside the 3D space at the position of the VR controller, and so it is possible to exactly position where the end effector of the robot should go. Using the robot's hand model in the environment removes ambiguities about the orientation of the robot end effector after motion planning—this is an issue we noticed with existing solutions. When a movement command is issued by the user, the desired joint state positions of the hands in reference to the main body of the robot are then passed to the MoveIt Motion Planning Framework. An attempt is made at constructing a plan, and the user is informed of the plan in progress through a visual update on the robot: the relevant arm of the robot turns red. This provides immediate feedback while not consuming additional display space.

Since the left arm of the robot features the RGB-D wrist camera and accompanying point cloud, it can be positioned to illuminate any finer detail needed to be used to, for example, grasp and move a smaller object, while the inferred MSI is able to provide needed context to the user. This is especially useful in areas with safety or damage considerations, as the user and move around the robot and examine its rear to ensure that while backing up the robot there are no collisions. The user can see the environment around the robot updated in real time with the information needed to operate safely.

The base of the robot can also be controlled by an operator. Using the inferred MSI for context to prevent collisions, the robot can be moved forwards, backwards, or turned at varying speeds using the trackpads or joysticks on the VR controllers. These capabilities can also be turned off and on through a VR menu option so as to avoid any accidental motions.

In addition to enabling and disabling base movement, we include other control options in the VR menu. The user can disable either or both of the arms. They can also switch between rendering schemes for both the RGB-D data and 360° image data. RGB-D data can be viewed as either a point cloud or mesh, as mentioned previously. The 360° camera's data can be viewed as an MSI, a depth-deformed mesh, a stereo 360 skybox, or a simple single-depth sphere.

4 EXPERIMENTS

This setup has been used successfully to move the robot's base and to perform grasps. Within a short period of time, new system users could control and move the robot within the lab from their own home, such as picking up a ball from a table and placing it into a recycling bin (Figure 4).

We have experimented with different scene reconstruction options. We found that using reconstructed depth via MSIs or a depth-deformed mesh to view the 360° camera data with 6DoF provides a less jarring-and-nausea-inducing experience than just using the stereo image pair directly or than only rendering at a single depth. These preliminary experiments show the promise of the methods.

Our planned work includes the ability to visualize any current plans of the robot and approve them before the plan is carried out, as well as the ability to use a newly-trained version of the MatryODShka network to process the individual lenses of the ODS camera. This will allow us to infer depth information without pre-stitching the six camera feeds, and so should improve both the speed of the pipeline of the network and the image quality (as data loss from stitching the frames first will not be introduced).

5 CONCLUSION

We believe a robust 3D environment with human-like control supporting movement options represents the next step in teleoperation. However, movement in three dimensions and especially remote control presents unique challenges in spatial awareness and operator control. Our research work is intended to discover and implement interfaces that allows an operator to better teleoperate a robot. Thus far, we have been able to introduced a new human operator to our visualization and control mechanisms and allow them to control the robot from their own home using a set of intuitive controls and a low-cost ($300) VR headset and controllers.

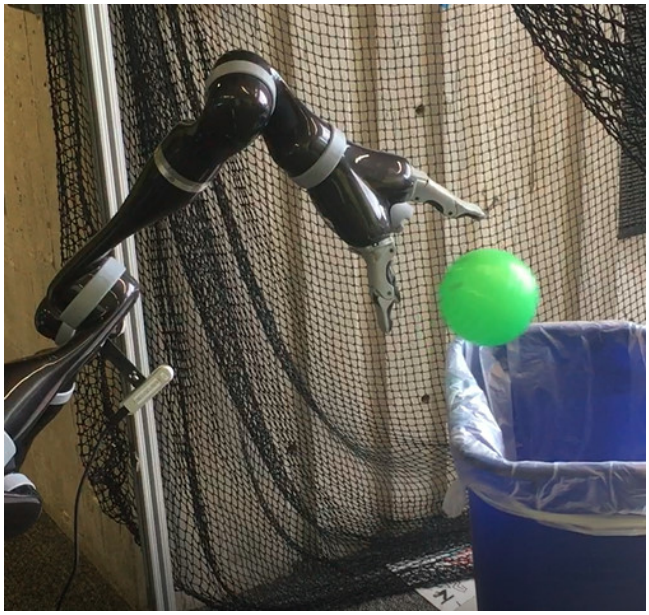

Figure 4: A new user was able to pick and place objects within minutes of using the system, here picking up and dropping a green ball into a bucket.

ACKNOWLEDGEMENTS

This research was supported by an Amazon Research Award, a Brown OVPR Seed Award, and a Brown SPRINT award.

REFERENCES

[1] Signe Brewster. 2020. The Best VR Headset. https://www.nytimes.com/wirecutter/reviews/best-standalone-vr-headset/
[2] Anne Eisenberg. 2014. The Rolling Robot Will Connect You Now. https://www.nytimes.com/2014/03/02/technology/the-rolling-robot-will-connect-you-now.html
[3] Mohamed Elobaid, Yue Hu, Jan Babic, and Daniele Pucci. 2018. Telexistence and Teleoperation for Walking Humanoid Robots. *CoRR* abs/1809.01578 (2018), 1106–1121. arXiv:1809.01578 http://arxiv.org/abs/1809.01578
[4] C. Freschi, V. Ferrari, F. Melfi, M. Ferrari, F. Mosca, and A. Cuschieri. 2013. Technical review of the da Vinci surgical telemanipulator. *The International Journal of Medical Robotics and Computer Assisted Surgery* (2013). https://doi.org/10.1002/rcs.1468 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/rcs.1468
[5] Matthias Hirschmanner. 2017. *Teleoperation of a humanoid robot using Oculus Rift and Leap Motion.* Master's thesis. Technische Universtität Wien.
[6] Jeffrey I Lipton, Aidan J Fay, and Daniela Rus. 2017. Baxter's Homunculus: Virtual Reality Spaces for Teleoperation in Manufacturing. arXiv:1703.01270 [cs.RO]
[7] Pat Marion, Maurice Fallon, Robin Deits, Andrés Valenzuela, Claudia Pérez D'Arpino, Greg Izatt, Lucas Manuelli, Matt Antone, Hongkai Dai, Twan Koolen, John Carter, Scott Kuindersma, and Russ Tedrake. 2017. Director: A User Interface Designed for Robot Operation with Shared Autonomy. *Journal of Field Robotics* 34, 2 (2017), 262–280. https://doi.org/10.1002/rob.21681 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21681
[8] Zoe McCarthy. 2016. Mind Meld - Zoe McCarthy. https://www.youtube.com/watch?v=kZlg0QvKkQQ
[9] Dejing Ni, AYC Nee, SK Ong, Huijun Li, Chengcheng Zhu, and Aiguo Song. 2018. Point cloud augmented virtual reality environment with haptic constraints for teleoperation. *Transactions of the Institute of Measurement and Control* 40, 15 (2018), 4091–4104. https://doi.org/10.1177/0142331217739953 arXiv:https://doi.org/10.1177/0142331217739953
[10] Yeonju Oh, Ramviyas Parasuraman, Tim Mcgraw, and Byung-Cheol Min. 2018. 360 VR Based Robot Teleoperation Interface for Virtual Tour.
[11] Jason Orlosky, Konstantinos Theofilis, Kiyoshi Kiyokawa, and Yukie Nagai. 2020. Effects of Throughput Delay on Perception of Robot Teleoperation and Head Control Precision in Remote Monitoring Tasks. *PRESENCE: Virtual and Augmented*

Reality 27, 2 (2020), 226–241. https://doi.org/10.1162/pres_a_00328

[12] parsec. [n.d.]. Parsec SDK. https://parsec.app/docs/sdk

[13] Daniel Rakita, Bilge Mutlu, and Michael Gleicher. 2018. An Autonomous Dynamic Camera Method for Effective Remote Teleoperation. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction* (Chicago, IL, USA) *(HRI '18)*. Association for Computing Machinery, New York, NY, USA, 325–333. https://doi.org/10.1145/3171221.3171279

[14] Susumu Tachi. 2019-09-11. Forty Years of Telexistence —From Concept to TELESAR VI (Invited Talk). In *ICAT-EGVE 2019 - International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments*, Yasuaki Kakehi and Atsushi Hiyama (Eds.). Eurographics Association. https://doi.org/10.2312/egve.20192023

[15] Jayant Thatte and Bernd Girod. 2018. Towards Perceptual Evaluation of Six Degrees of Freedom Virtual Reality Rendering from Stacked OmniStereo Representation. *Electronic Imaging* 2018, 5 (2018), 352–1–352–6. https://doi.org/doi:10.2352/ISSN.2470-1173.2018.05.PMII-352

[16] Alexandra Valiton and Zhi Li. 2020. Perception-Action Coupling in Usage of Telepresence Cameras. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 3846–3852. https://doi.org/10.1109/ICRA40945.2020.9197578

[17] David Whitney, Eric Rosen, Elizabeth Phillips, George Konidaris, and Stefanie Tellex. 2018. Comparing Robot Grasping Teleoperation across Desktop and Virtual Reality with ROS Reality. In *Robotics Research*. Springer, 335–350.

[18] David Whitney, Eric Rosen, Daniel Ullman, Elizabeth Phillips, and Stefanie Tellex. 2018. ROS Reality: A Virtual Reality Framework Using Consumer-Grade Hardware for ROS-Enabled Robots. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 1–9. https://doi.org/10.1109/IROS.2018.8593513

[19] X. Xu, B. Cizmeci, A. Al-Nuaimi, and E. Steinbach. 2014. Point Cloud-Based Model-Mediated Teleoperation With Dynamic and Perception-Based Model Updating. *IEEE Transactions on Instrumentation and Measurement* 63, 11 (2014), 2558–2569. https://doi.org/10.1109/TIM.2014.2323139

[20] Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. 2018. Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation. arXiv:1710.04615 [cs.LG]