Testing Robot Teleoperation using a Virtual Reality Interface with ROS Reality

Conference Paper · March 2018

Eric Rosen
Brown University
10 PUBLICATIONS **54** CITATIONS
SEE PROFILE

David Whitney
Brown University
10 PUBLICATIONS **77** CITATIONS
SEE PROFILE

Elizabeth K Phillips
Brown University
37 PUBLICATIONS **233** CITATIONS
SEE PROFILE

Stefanie Tellex
Massachusetts Institute of Technology
62 PUBLICATIONS **1,613** CITATIONS
SEE PROFILE

Testing Robot Teleoperation using a Virtual Reality Interface with ROS Reality

Eric Rosen
Brown University
Providence, Rhode Island
eric_rosen@brown.edu

David Whitney
Brown University
Providence, Rhode Island
david_whitney@brown.edu

Elizabeth Phillips
Brown University
Providence, Rhode Island
elizabeth_phillips1@brown.edu

Daniel Ullman
Brown University
Providence, Rhode Island
daniel_ullman@brown.edu

Stefanie Tellex
Brown University
Providence, Rhode Island
stefie10@cs.brown.edu

ABSTRACT

Virtual reality (VR) systems let users intuitively interact with 3D environments, and have been used extensively for robotic teleoperation tasks. Successful robot teleoperation requires operators to have sufficient contextual understanding of a robot's environment. While more immersive than their 2D counterparts, early VR systems were expensive and required specialized hardware. Fortunately, there has been a recent proliferation of consumer-grade VR systems at affordable price points. These systems are inexpensive, relatively portable, and easy to integrate into existing robotic frameworks. Our group has designed one such VR teleoperation package, ROS Reality. ROS Reality is an open-source, over-the-Internet teleoperation interface between a ROS-enabled robot and a consumer-grade VR headset. Due to the architecture of our system, it could be easily adapted to work with any ROS-enabled robot or Unity-compatible VR headset. We hope this system will be adopted by other labs to allow for easy integration of VR teleoperation of robots into experiments. To study the efficacy of our system, we completed a pilot study of 12 manipulation tasks on a Baxter robot, trying to complete each with both direct kinesthetic operation and ROS Reality. This study can serve as a baseline for future teleoperation interfaces and expose issues that need to be addressed in consumer-grade VR systems.

KEYWORDS

virtual reality, teleoperation, human-robot interaction, ROS Reality

ACM Reference Format:
Eric Rosen, David Whitney, Elizabeth Phillips, Daniel Ullman, and Stefanie Tellex. 2018. Testing Robot Teleoperation using a Virtual Reality Interface with ROS Reality. In *Proceedings of International Workshop on Virtual, Augmented and Mixed Reality for Human-Robot Interaction (VAM-HRI 2018).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

In this extended abstract, we detail our consumer-grade VR teleoperation interface, ROS Reality. We discuss how the package allows for a ROS-networked robot, like Baxter from Rethink Robotics, to bilaterally communicate over the Internet with an HTC Vive through the Unity game engine. We also present the results of a pilot study we conducted to test the efficacy of using ROS Reality to teleoperate a robot to perform 12 manipulation tasks.

2 RELATED WORK

Teleoperation enables robots to complete tasks they would otherwise be unable to complete autonomously, such as in the DARPA Robotics Challenge [3] or unmanned vehicle control [4]. Dangerous situations, such as bomb disposal [1], present ideal use cases for teleoperated robots, as robot use keeps humans out of harm's way.

2D interfaces for robot teleoperation, especially over the Internet, have been popular in recent years [5]. Monitor and keyboard setups have been used to control robots for a variety of classical tasks, like motion planing and item grasping [11]. Web browsers have proven especially useful in allowing anyone around the world with a computer to teleoperate a robot, broadening the user-base of operators [9]. However, 2D monitor interfaces do not reflect the natural way that humans observe and interact with the 3D world. We have previously shown that non-expert users are more efficient with a VR teleoperation interface, and that they prefer using one, compared to a keyboard and monitor [12].

Virtual reality interfaces and gantry systems offer intuitive means for directly mapping a user's motion actions and observations to those of the robot they are controlling [12]. For example, the da Vinci Robot System is an immersive haptic telesurgery system which has shown improvements for novice and experienced users in surgical performance [2]. While powerful, the da Vinci is very task-specific and stationary. Mallwitz et al. [8] developed a portable and easily-dressable exoskeleton that allowed a human user to naturally teleoperate a complex humanoid robot. This system is very intuitive, but again is limited to specific robots and is extremely expensive, heavily limiting the possible operator-base compared to web-based interfaces.

Recent advancements in graphics have made commercially available VR systems accessible to the gaming community. Systems like the HTC Vive, Oculus Rift, and Google Cardboard offer cheap and

---

VAM-HRI 2018, March 2018, Chicago, Illinois USA

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

portable VR hardware. As a result, lab researchers have recently begun exploring these VR systems for robot teleoperation. Zhang et al. [13] used an HTC Vive to teleoperate a PR2 and perform imitation learning. Lipton et al. [7] also used a commercially available VR system for performing teleoperation on a Baxter. Our previous work [12] on comparing VR to 2D systems also used an HTC Vive for the VR interface, enabled through ROS Reality. By having labs use the same VR systems, results and interfaces are easier to duplicate.

The proliferation of consumer-grade VR systems is very recent, so there has only been some research on the efficacy of teleoperation interfaces that use this technology (e.g., [13]). Although task completion depends heavily on interface type and the particular robot, we were interested in exploring what complex tasks could be completed on our open-source software using a common research robot.

3 ROS REALITY

This section first provides a brief synopsis of interacting with a robot in virtual reality, and then a technical description of ROS Reality [6]

3.1 VR as a Teleoperation Interface

The two most common virtual reality systems today are the Oculus Rift and the HTC Vive. Our group develops with the HTC Vive due to superior room-scale tracking, but the following description of how to use VR as a teleoperation interface applies to both systems.

There are multiple ways of displaying the robot's state to the user, and mapping the user's input to the robot. We bin these different methods into two main categories: egocentric or robocentric.

In egocentric models, the human is the center of the virtual world, and inhabits the same space as the robot. Lipton et al. [7]'s homunculus work and Zhang et al. [13] are examples of this egocentric mapping. Under these conditions, human users have reported feeling like they 'become the robot' or 'see out of the robot's eyes'.

In a robocentric model, the human and robot share a virtual space, but are not necessarily superimposed on one another. The model we used for evaluating ROS Reality [12], falls into this category. Under this model, the human walks around a virtual model of the robot, and controls its arms by virtually grabbing and dragging them. We therefore call this model a virtual gantry system.

3.2 System Overview

An HTC Vive is connected to a computer running the Unity game engine. Unity builds a local copy of our robot based on its URDF with a custom-made URDF parser. Unity connects to a ROS network over the Internet via a WebSocket connection. The pose and wrist cameras of the robot are sent via this WebSocket connection, as well as the color and depth image of a Kinect 2 mounted to the robot's head. The color and depth image are converted into a point cloud in Unity via a custom shader. When the user holds down a deadman's switch, the pose of the user's controllers are sent back to the robot, which uses an inverse kinematics solver to move the robot's end effectors to the specified poses.

3.3 ROS

ROS (Robot Operating System) is a set of tools and libraries to help program robot applications. ROS connects processes of programs, or nodes, that perform different functions. Nodes can communicate by streaming data over channels, or topics. Nodes can publish ROS Data Structures to different topics to broadcast information to different nodes, or subscribe to topics by registering handler functions to manage incoming publications to the topic. ROS provides a library, rospy, that allows programmers to create and use nodes in python. ROS Reality has a script that uses rospy to create a node that helps facilitate robot data to Unity.

3.4 HTC Vive

The HTC Vive is a consumer-grade virtual reality system. It has three tracked objects: one head-mounted display (HMD), and two wand controllers. Each device is tracked via a set of two infrared pulse laser emitters, known as lighthouses, allowing for tracking via time-of-flight calculations[1]. Each tracked object is positionally and rotationally tracked, with roughly 1-2mm of error. The wand controllers are fully wireless, and the HMD connects to a computer via a USB and HDMI cable. Each controller has a touch-pad, trigger, and two buttons for user input.

The HTC Vive supports several game and physics engines, but the initial (and in our opinion best supported) development platform is Unity[2]. ROS Reality contains C# scripts that open a WebSocket connection between Unity and ROS.

3.5 Unity

Unity is a game engine that is used for many popular 2D, 3D, and Virtual/Augmented/Mixed Reality applications. It has a built-in physics engine that can handle contact dynamics, as well as material simulation (such as water, sand, or cloth). Most scripts are written in C#, but Unity also provides a shader language for writing custom GPU shaders.

3.5.1 URDF Parser. Unified Robot Description Format (URDF) is an XML-based specification for representing robot models. URDFs include information about how parts of a robot are connected and interact with each other, as well as modeling data for rendering the robot in a simulator. URDFs provide a general way to represent virtual robots.

ROS Reality includes a C# program to parse URDF files and create gameobjects in a Unity scene from the URDF models. The URDF parser assembles the robot model together so that the human operator can interact with a virtual robot in a scene. Unity's high quality rendering allows generated virtual robots to look realistic and have no perceived lag in movement.

These virtual robots now have the ability to interact with other gameobjects in the scene. This allows the virtual robot to engage in complex tasks that are managed by Unity's powerful physics engine. This can be useful for practicing teleoperation interactions in simulated scenarios.

Once the URDF parser constructs a realistic model of the robot in Unity, ROS Reality provides the capability of connecting the virtual

---

[1] The Oculus Rift uses multiple cameras to track the HMD and controllers.

[2] Formally known as Unity3D.

robot to the real robot. This involves having the robot model's transform, or position and pose, mirror the real robot's transform, and vise versa.

3.5.2 Transform Listener. ROS provides a transform topic for a robot to publish information about its position and pose to. In a similar fashion, each gameobject in Unity has a transform that is tracking the position and pose of the object. However, the ROS coordinate system is different than Unity's. In order to calibrate virtual space to real robot space, ROS Reality provides a script, the *TFListener*, which can convert transform data from ROS to Unity, and vise versa. Using the TF Listener, the virtual robot model can mirror the state of a real robot, as well as have the real robot mirror the virtual one. This creates an interface for the human operator to perceive the real robot's state in virtual reality in an immersive fashion.

In order to make the robot interface interactable for the human operator, ROS Reality provides a Unity script that allows a VR operator to use the hand controllers in order to interact with the virtual robot model.

3.6 Robot

We use a Baxter from Rethink Robotics. Baxter is a robot designed for industrial automation applications, also serving as a useful research platform. Baxter has a fixed base and display screen head, with two 7 DoF arms and grippers with force sensing that enable Baxter to dexterously manipulate a variety of objects. We attached rubber grips that come in the Baxter toolkit in order to maximize the friction at the end effector.

4 VR TELEOPERATION STUDY

We considered desirable skills for a manipulator robot to have. Our goal was to answer two questions:

1. Is the robot physically capable of performing certain tasks?
2. If so, can a human teleoperating the robot in VR complete this task?

Because the physical capabilities of the robot depends on the hardware, our specific study used a research Baxter robot. Refer to section 3.6 for more information.

In order to answer these two questions, two authors of this paper acted as the expert teleoperators for performing the trials. For question one, we physically moved the robot arms in real life to complete the task. We used this methodology of Direct Manipulation in our previous VR study as a good measure for task feasibility [12]. For question two, we used our ROS Reality interface mentioned in section 3 to perform VR teleoperation to complete the tasks.

We came up with the set of tasks by choosing different common robot manipulation tasks that could be relevant to social life. For each task and interface, we performed a maximum of 5 attempts, and determined the task feasible if we were able to complete it at least once. We report our results in Table 1.

Table 1: Task Feasibility Evaluations

Task	Task Number	Direct?	VR?
Block Stacking	1	Yes	No
Unscrew Bottle	2	No	-
Uncap Marker	3	Yes	Yes
Hinge Board	4	No	-
Stir Pot	5	Yes	Yes
Push Spacebar	6	Yes	Yes
Move Checker Piece	7	Yes	Yes
Squeeze Purell	8	Yes	Yes
Insert Connect 4 Piece	9	Yes	Yes
Toss and Catch Ball	10	No	-
Use Fork	11	Yes	Yes
Unzip Zipper	12	No	-

4.1 Manipulation Tasks

1. *Block Stacking* - Stack ten 3x3cm wood blocks in a column.
2. *Unscrew Bottle* - Unscrew the cap to a bottle.
3. *Uncap Marker* - Remove cap from an Expo marker.
4. *Hinge Board* - Open all six latches on Melissa and Doug Latches Wooden Activity Board.
5. *Stir Pot* - Stir a wooden spoon in a metal pot.
6. *Push Spacebar* - Push the spacebar button on a keyboard.
7. *Move Checker Piece* - Pick and place a checker piece on a board.
8. *Squeeze Purell* - Squeeze out Purell from the bottle.
9. *Insert Connect 4 Piece* - Insert a Connect 4 piece into the slot.
10. *Toss and Catch Ball* - Toss a juggling ball up and catch it in the same hand.
11. *Use Fork* - Get a piece of food onto a plastic fork.
12. *Unzip Zipper* - Unzip a loose zipper.

5 DISCUSSION

We found tasks 3, 5, 6, 7, 8, 9, and 11 to be possible in Direct and VR interfaces. We believe the commonality between them is that the tasks were robust to failures and did not require extremely fine-grained control of the robot arm. We found task 1 to be possible by directly moving the robot, but not in VR; this is potentially due to the unforgiving precision required with the fragile tower. Tasks 2, 4, 10, and 12 were not able to be performed by directly controlling the robot, and thus were not attempted in VR; this is likely due to either the Baxter not being able to exert a powerful enough force or end effector deficiencies. This shines a light on what benefits and disadvantages VR has as a teleoperation interface, and may act as a guideline for what future researchers should be interested in when solving new robot problems, whether from a hardware perspective or a software one.

One of the major pain points when using ROS Reality is knowing what poses the robot can and cannot assume. The ROS node Baxter uses to solve inverse kinematics (IK) returns an error when the pose is unreachable. In the future, we would like to provide this information to the user in VR. If a goal coordinate is unreachable, the user's controller could vibrate, or turn red, to indicate this location is not reachable by the robot. Similar feedback systems have been shown to improve the skill acquisition in training for robotic assisted surgery [10] and may similarly apply here.

6 PROPOSED WORK

The project described in this paper serves as the foundation for a host of proposed work. Virtual reality is becoming increasingly available to everyday users, with hardware platforms steadily decreasing in cost. Robots with a high degree of functionality are simultaneously becoming less and less expensive, yet virtual reality systems in general remain more affordable than robotic systems. Most advanced robots still exist in lab settings, with skilled operators in control of these systems, making robots a relatively limited resource. As we try to train these robotic systems to complete increasingly demanding and autonomous tasks, we typically do so one by one, requiring dedicated time and effort with a robot to address each task. Learning from demonstration (LFD) is an approach often used to implement new tasks on robots, but is nonetheless demanding.

Collecting LFD data from human participants controlling physical robot platforms is time-consuming and resource-limited. One typical approach to addressing this problem has been to develop algorithms that require fewer and fewer demonstrations from humans; this is a challenging approach, and will inevitably require at least one, if not more, demonstrations by human users with physical robots. Even with this solution, at some point for most tasks, users will have to interact with robots to help train them. However, using VR as a mechanism to gather LFD data at scale is a promising alternative. Demonstrations could be provided to virtual robots by users accessed over the Internet in a crowdsourcing paradigm, completing tasks at scale, and thus addressing participant and resource limitations that currently plague extant LFD training methods. VR has the potential to offer a cost and time-effective solution to this challenge. The work reported here is a first step in identifying the manipulation tasks that could be demonstrable to robots using consumer-grade VR headsets in a crowdsourcing data collection paradigm.

REFERENCES

[1] Carlos Beltrán-González, Antonios Gasteratos, Angelos Amanatiadis, Dimitrios Chrysostomou, Roberto Guzman, András Tóth, Loránd Szollosi, András Juhász, and Péter Galambos. 2007. Methods and techniques for intelligent navigation and manipulation for bomb disposal and rescue operations. In *Safety, Security and Rescue Robotics, 2007. SSRR 2007. IEEE International Workshop on*. IEEE, 1–6.

[2] John C Byrn, Stefanie Schluender, Celia M Divino, John Conrad, Brooke Gurland, Edward Shlasko, and Amir Szold. 2007. Three-dimensional imaging improves surgical performance for both novice and experienced operators using the da Vinci Robot System. *The American Journal of Surgery* 193, 4 (2007), 519–522.

[3] Christopher M Dellin, Kyle Strabala, G Clark Haynes, David Stager, and Siddhartha S Srinivasa. 2016. Guided manipulation planning at the DARPA robotics challenge trials. In *Experimental Robotics*. Springer, 149–163.

[4] Sebastian Gnatzig, Frederic Chucholowski, Tito Tang, and Markus Lienkamp. 2013. A system design for teleoperated road vehicles. In *ICINCO (2)*. 231–238.

[5] Ken Goldberg, Michael Mascha, Steve Gentner, Nick Rothenberg, Carl Sutter, and Jeff Wiegley. 1995. Desktop teleoperation via the world wide web. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, Vol. 1. IEEE, 654–659.

[6] Brown University H2R Lab. 2017. ROS Reality. (2017). https://github.com/h2r/ros_reality

[7] Jeffrey I Lipton, Aidan J Fay, and Daniela Rus. 2018. Baxter's homunculus: Virtual reality spaces for teleoperation in manufacturing. *IEEE Robotics and Automation Letters* 3, 1 (2018), 179–186.

[8] Martin Mallwitz, Niels Will, Johannes Teiwes, and Elsa Andrea Kirchner. 2015. The CAPIO active upper body exoskeleton and its application for teleoperation. In *Proceedings of the 13th Symposium on Advanced Space Technologies in Robotics and Automation. ESA/Estec Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA-2015). ESA.*

[9] Gunter Niemeyer and Jean-Jacques E Slotine. 2002. Toward bilateral internet teleoperation. *Beyond Webcams: An Introduction to Online Robots* (2002), 193.

[10] OAJ Van der Meijden and MP Schijven. 2009. The value of haptic feedback in conventional and robot-assisted minimal invasive surgery and virtual reality training: A current review. *Surgical Endoscopy* 23, 6 (2009), 1180–1190.

[11] Jean Vertut. 2013. *Teleoperation and Robotics: Applications and Technology*. Vol. 3. Springer Science & Business Media.

[12] David Whitney, Eric Rosen, Elizabeth Phillips, George Konidaris, and Stefanie Tellex. 2017. Comparing robot grasping teleoperation across desktop and virtual reality with ROS Reality. (2017).

[13] Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Ken Goldberg, and Pieter Abbeel. 2017. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. *arXiv preprint arXiv:1710.04615* (2017).