VIRTUAL REALITY MEDIATED ROBOT TELEOPERATION AND GRASPING

Kun Chen ‘18, Prawesh Dahal ’18, Mariam Avagyan ‘18

Advisor: Prof. Kevin Huang ‘12

ABSTRACT

Teleoperation, or robotic operation at a distance, leverages the many physical benefits of robots while simultaneously bringing human high-level control and decision making into the loop. Poor viewing angles, occlusions, and delays are several factors that can inhibit the visual realism and usability of teleoperation. Furthermore, poor translation of human motion commands into resultant robotic action can impede immersion, particularly for architectures wherein the remote device and the local device are kinematically dissimilar. In this project, commodity **virtual reality (VR)** headsets and trackers along with high dexterity gloves are utilized to overcome these technical obstacles, and provide an enhanced, visually immersive, and realistic real-time operator experience. Using glove-like interfaces that provide additional ten degrees of freedom per hand, the User’s natural grasping motion can be obtained, and resultant motion commands can be executed by the remote device, in this case the **Rethink Robotics Sawyer 7DOF manipulator.** The method is general and extendable to platforms with up to two dexterous manipulators, and has implications in use of telerobotics in manufacturing, disaster response and surgery to name a few. Finally, the three subsystems of teleoperation were successfully implemented.

INTRODUCTION

Teleoperation, as a topic of research over the past 50 years, has proven to be an efficient method of tackling problems such as reducing mission cost, avoiding loss of life, and accessing hostile or difficult to reach environments [1,2]. The need for human engagement in teleoperation, despite advances in autonomous robotics, is still significant due to advantages like faster task execution speed and failure recovery [3,4]. With applications in unmanned underwater vehicles, telesurgeries, space robotics, and others, telerobotics calls for a clear remote perception interface and effective remote manipulation [5,6]. In our project, we aim to develop a real-time telerobotics system with a virtual reality interface. The user is able to interact with the virtual workstation using a HTC Vive Headset and Manus Data Gloves. User’s hand motion is mapped in real-time to the robot’s gripper, allowing the teleoperation of the robot.

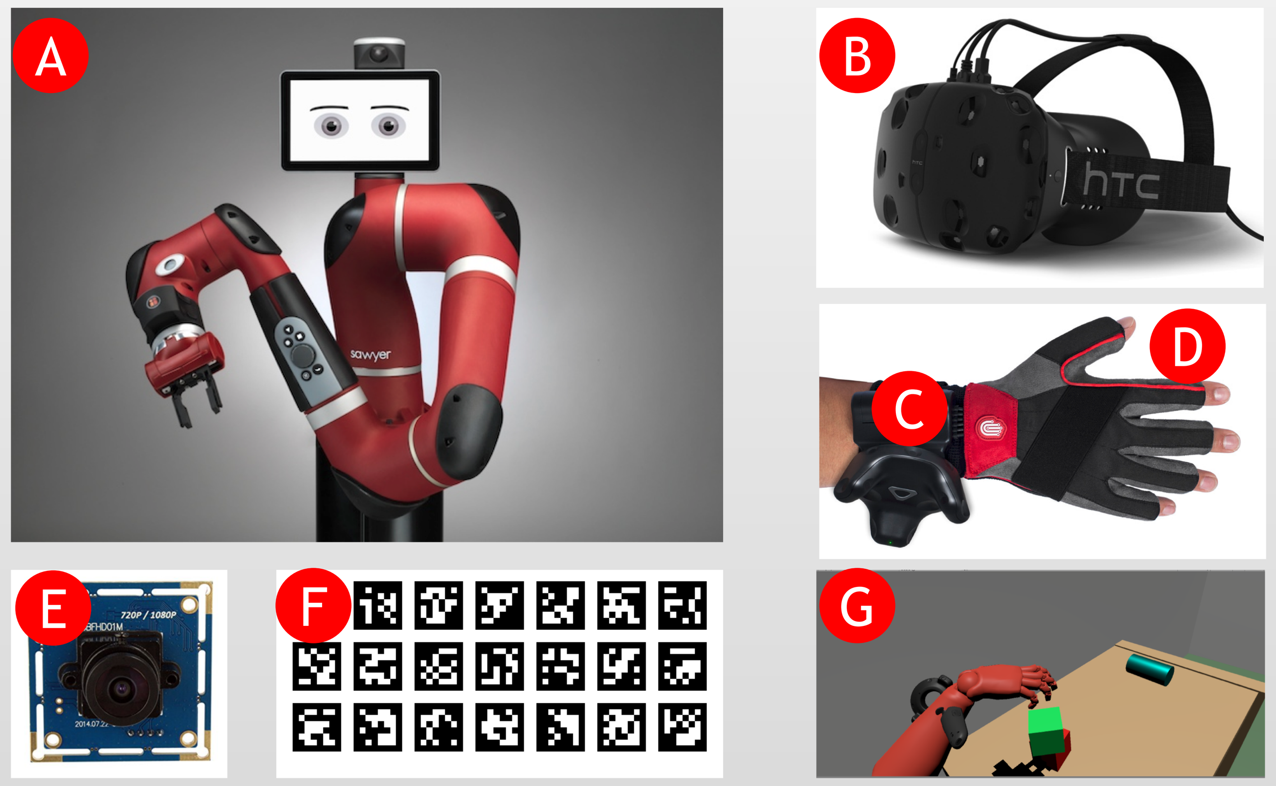


FIGURE 1. A) Sawyer Robot, B) HTC Vive Headset, C) Manus Data Glove, D) External Camera, E) AR Tags, F) Unity Game Engine

METHODS & PROCEDURE

First, the Unity Game Engine was used to create our virtual environment with a table, objects (small boxes), and hands. All created items were game objects, i.e. they followed basic rules of physics. The User can enter the virtual workspace and see the virtual objects by wearing the HTC Vive Headset. When the User wears Manus gloves and moves their hands, virtual hands in Unity move accordingly.

Information flow from ROS to Unity: In order to make it possible for the robot, Sawyer, to visualize the real environment and identify real objects on the table, two ELP USB cameras were used along with the right-hand camera in the Sawyer. Using ***AR_Track_Alvar* package**, AR tags with different IDs and sizes were generated and attached to these real objects. Once an object with an AR tag is placed in the real workspace on the table, the tag is detected individually by all cameras. To combine the useful information from all cameras effectively, the ***tf* broadcaster** (a ROS package) was used to calculate the relative position of this tag with respect to the robot. This was verified by plotting the *tf* tree. Once the accurate position and orientation of an object was obtained in ROS, the next task was to send this information from real to virtual workspace. For this, a websocket **TCP/IP connection** was created using the ***Rosbridge* package**. ROS publisher was used to send the position of real objects to Unity.

Information flow from Unity to ROS: At the same time, by installing *Rosbridge* on Unity, we were able to subscribe to the messages sent from ROS and update the VR environment in real time. Then, the position and orientation of the fingers (obtained through Manus gloves in Unity) were published to ROS. In ROS, these position data served as an input to the inverse kinematic (IK) solver, *IKService*, to calculate the necessary joint angles. Using these joint angles, Sawyer was able move to the desired location in real time. Finally, once the Sawyer moved to the desired location, it would need to grab the target object. To do so, the different levels of gripper contraction – open, hold, grab – were utilized, with respect to hand’s configuration, to accomplish the goal of grabbing an object.

User

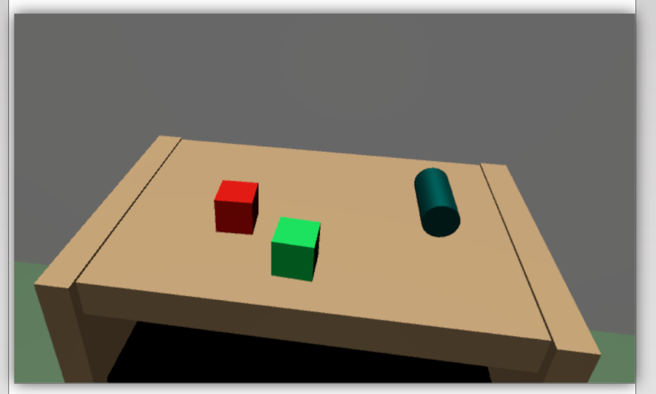
OPERATOR

Agent

STEP 1

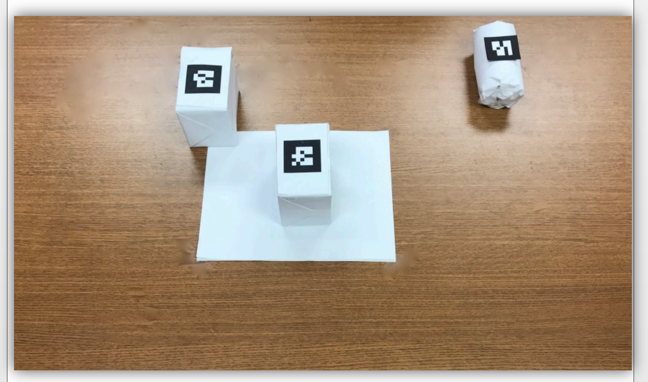
User wears the HTC Vive Headset and Manus Data Gloves. As User moves in virtual space, the Sawyer robot follows the User’s motions with small (a few ms) time delay.

Virtual Table



WORKSPACE

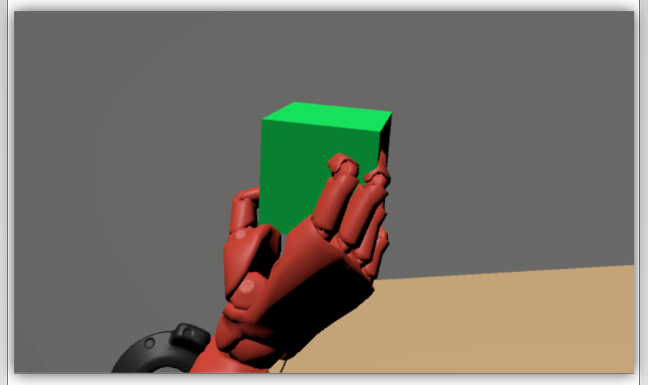
Real Table



STEP 2

Unity game objects are created in virtual reality. For each object in VR, an object and corresponding AR tag is presented in the real workspace.

Virtual Object



ACTION: GRASPING

Real Object

STEP 3

As the User picks up an object in VR, the Sawyer, despite being kinematically dissimilar, follows the User’s motion and picks up the corresponding real object.

SYSTEM COMPONENTS

HARDWARE:

- **Rethink Sawyer robot --** a high performance automation robot arm with 7 degrees of freedom (Fig. 1A)
- **HTC Vive Headset --** a virtual reality headset that allows the user to move in 3D space (Fig. 1B)
- **Vive Tracking Devices --** small attachable motion tracking accessories. Provide wrist position and orientation (Fig. 1C)
- **Manus Data Gloves --** virtual reality gloves. Ensure action VR, track hand movements in real-time (Fig. 1D)
- **Camera OV 2270 - two** USB Camera with 1080p HD (Fig. 1E)
- **Augmented Reality (AR) tags --** visual fiducial identification labels. Contain information about the object they are attached to (Fig. 1F)

SOFTWARE:

- **Steam VR Platform --** a platform that realizes VR space
- **Unity Game Engine --** development of 3D graphics (Fig. 1G)
- **Robot Operating System (ROS)** -- meta-operating system
- **Intera SDK** - a software development kit (SD) for the Sawyer
- **AR Alvar Kit** -- detects position and rotation of objects based
- **Rosbridge** - used to send information from ROS to Unity

RESULTS & FUTURE WORK

In our project, we successfully implemented the three subsystems of teleoperation. In the virtual Unity workspace, the User wearing the headset and gloves could realize the hands and the individual finger joints movements accurately. The position and orientation of objects in the virtual environment were continually subscribed to messages from ROS detecting AR tags. These objects were in the same location in VR as they were in the real workspace. As the User translated or rotated the Manus hand in VR towards a desired configuration, the gripper base of Sawyer subscribed to these JointState messages in the ROS frame, advancing towards the same direction. Due to the computation time of the inverse kinematics solver, we observed a delay of few milliseconds between the User and robot’s motion. However, having a virtual fixture in Unity allowed the User to follow teleoperation of the Sawyer.

REFERENCES

[1] Hokayem, P. F., and Spong, M. W., 2006. “Bilateral teleoperation: An historical survey”. *Automatica*, ***42***(12), pp. 2035– 2057.

[2] Chen, J. Y., Haas, E. C., and Barnes, M. J., 2007. “Human performance issues and user interface design for teleoperated robots”. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, ***37***(6), pp. 1231– 1245.

[3] Fong, T., and Thorpe, C., 2001. “Vehicle teleoperation interfaces”. *Autonomous robots*, ***11***(1), pp. 9–18.

[4] Brizzi, F., Peppoloni, L., Graziano, A., Di Stefano, E., Avizzano, C. A., and Ruffaldi, E., 2018. “Effects of augmented reality on the performance of teleoperated industrial assembly tasks in a robotic embodiment”. *IEEE Transactions on Human-Machine Systems*.

[5] Cho, K., Ko, K., Shim, H., and Jang, I., 2017. “Development of vr visualization system including deep learning architecture for improving teleoperability”. In Ubiquitous Robots and Ambient Intelligence (URAI), 2017 14th International Conference on, IEEE, pp. 462–464.

[6] El-Shawa, S., Kraemer, N., Sheikholeslami, S., Mead, R., and Croft, E. A., 2017. “Is this the real life? Is this just fantasy?: Human proxemic preferences for recognizing robot gestures in physical reality and virtual reality”. In Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on, IEEE, pp. 341–348.