MR Immersive Telemanipulation

Arda Duzceker
ETH Zürich
Switzerland
ardad@student.ethz.ch

Jonas Hein
ETH Zürich
Switzerland
heinj@student.ethz.ch

Qais El Okaili
ETH Zürich
Switzerland
qaise@student.ethz.ch

Sandro Boccuzzo
ETH Zürich
Switzerland
sboccuzzo@student.ethz.ch

Abstract

During the last decades robots have become more and more omnipresent not only in research labs but all around us. The ubiquitous presence of robots in our daily life brings the focus towards finding a simple, easily applicable and seamlessly integrated interaction methods with such devices. In this report, we discuss how we can use the concepts of telemanipulation to support seamless human-machine interactions. We focus on the general purpose of using a Microsoft HoloLens® 2[1] *as an untethered mixed reality device to connect and interact with a state of the art collaborative robot such as YuMi® from ABB Robotics*[2].

1. Introduction

There are environments such as nuclear reactors, where it is just not possible to work in human safe conditions. Furthermore there are situations where it would be much more convenient to operate on a much smaller scale, like operating a large construction robot remotely using the scaled version of it. Last but not least, we can imagine a scenario where a doctor specialist performs a surgery at anytime from a remote location. These are some examples of why telemanipulation has caught wide interest in research and industry during the last years. With this project we aim to explore and enhance existing methods to control a state of the art industrial robot using a mixed reality device. We implemented different methods of interaction, namely "External Holographic Mode" and "Internal Immersive Mode". The former enables a setup where the user observes the hologram of YuMi in front of him/her and can interact with the robot by grabbing and dragging the arms of the hologram in real time. Also, we provide a sub-mode in which the user can construct a task, preview how the physical robot would behave for that particular task and make the robot execute it if desired. The latter mode allows a more immersive control scheme where the user becomes the robot itself. We achieve this by tracking the user's hands and mapping them to control signals for the robot's end-effectors.

(a) Microsoft HoloLens 2

(b) ABB YuMi IRB 14000

Figure 1: Our Devices

With our methodology, we achieve an intuitive and natural-feeling way of interaction with our robot. Our main contribution is simply the proof of the concepts for various ways of interaction and a successful prototype working on Hololens 2 controlling the ABB YuMi® IRB 14000 robot. During implementation, we utilized and extended a control interface developed at ETH Computational Robotics Lab[3] used also in their previous work such as [2].

The remainder of this paper is organized as follows.

[1] https://www.microsoft.com/en-us/hololens
[2] https://new.abb.com/products/robotics/de/industrieroboter/yumi
[3] http://crl.ethz.ch

Section 2 covers related work on which we based our work on. In Section 3 we present our approach as a team. We discuss our results in Section 4 and wrap up our report with possible future work.

2. Related Work

One of the main goals of immersive telemanipulation is to facilitate human-robotic interaction. In our work we focus on approaches for two-armed robots. Telemanipulation has already been addressed by many researchers. Rodehutskors *et al.* in [3] propose to combine immersive 3D visualization and tracking of operator head and hand motions to an intuitive interface for bimanual teleoperation. With the da Vinci telerobotic surgical system Ballantyne *et al.* [1] focused on using telemanipulation for a system capable to treat hemorrhages of soldiers with rapid surgical treatment even outside a medical facility. Moreover, in [2] Duenser *et al.* addressed the challenges of robotic manipulation of elastically deforming objects.

3. Methodology

As we mentioned earlier, our project is based on two devices, namely a Microsoft HoloLens 2 and an ABB YuMi robot. Note that, hereon when we say the robot, it also refers to the simulation middleware which is developed by the Computational Robotics Lab at ETH Zurich and modified by us to fit our needs. The simulation middleware can be easily connected to the physical robot and works with it without requiring any modifications. And, HoloLens always refers to our mixed reality device.

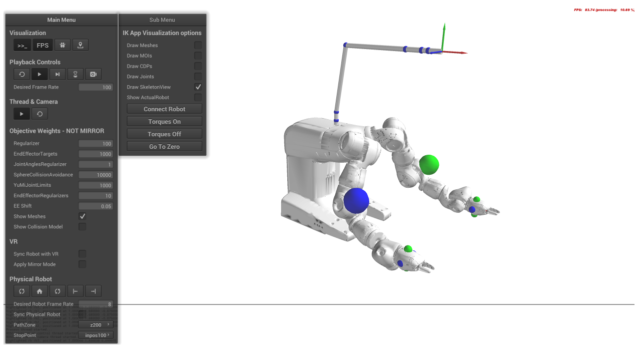

Figure 2: A screenshot from the simulation middleware.

The goal of the project is to enable the HoloLens user to remotely and intuitively control the robot while not requiring to see the physical robot. For this goal, we visualize the robot's 3D model as a hologram to the user wearing the HoloLens (Figure 3). The hologram's state is updated in real-time (with negligible network latency),

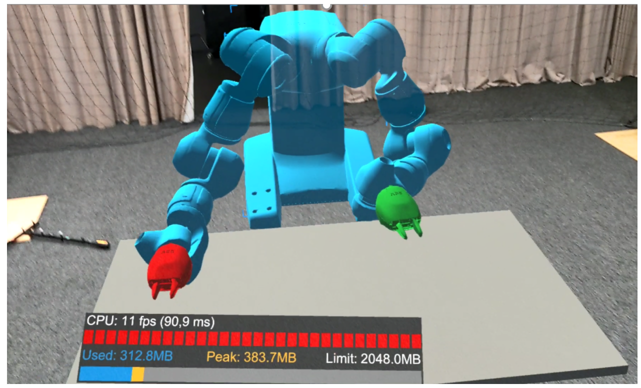

Figure 3: The robot hologram as the user sees. It is captured with Mixed Reality Capture of HoloLens.

so that the user knows exactly what the physical robot is doing in real-time. Furthermore, we make the robot hologram interactive in several different ways which are then mapped to the control messages to be executed on the simulation and the physical robot.

There are two main methods of interaction. We call the first method External Holographic Mode in which the user and the robot hologram are two separate entities in the mixed reality world. The second method is called Internal Immersive Mode where we aim to provide a more immersive experience to the user. We take advantage of having a two armed robot and let the user's arms act as the robot's arms in this mode. Also, in both of these modes, we include the camera stream received from the physical robot in the mixed reality scene enabling the user to observe the real-world environment of the physical robot. In immersive mode, similar to the arms, we map the user's head movement's to the camera arm so that the physical robot's "head" becomes the user's head and it moves accordingly.

3.1. Devices and Software

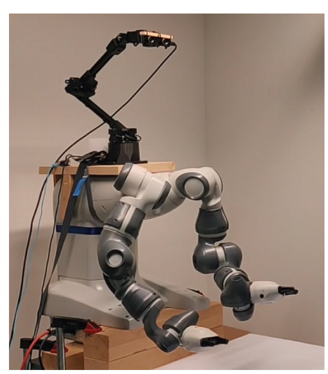

Figure 4: Our physical robot setup

ABB YuMi IRB 14000 is dual-armed collaborative robot. It is essentially designed for a new era of automation, for example in small parts assembly, where people and robots work side-by-side on the same tasks.[4] In each arm, it has an end-effector with two grippers. These grippers can move in one dimension, therefore they can be open, close or any configuration in between. Camera arm is a separate, simpler robot which can be mounted on top of YuMi. It can move in up-down and forward-back directions. The end joint where the camera is mounted can rotate as well.

On the robots' side, we base our work on the existing simulation written in C++ and provided by the Computational Robotics Lab (CRL). The simulation software includes a large library with a rudimentary, but high-level API. This API provides us with a inverse-kinematics solver, which calculates joint poses given a target pose, as well as methods to send these joint poses to the physical robots. Therefore, we can easily move the YuMi's arms by sending target poses or close/open its grippers. Similarly, through middleware when a target orientation is provided for the camera arm, the camera can look in that desired direction.

We extend the simulation with a communication interface and integrate control messages in order to receive commands from the HoloLens and send robot states to the HoloLens. We choose to use ROS framework for communication. There are two driving factors for this choice. First, it makes early development stage easier by handling low-level socket implementations. Second, there exists many standardized messages specially tailored for robot control in the ROS distribution which again increases the development pace by eliminating the necessity of message parsing. Note that, although ROS is a very sophisticated and comprehensive system, we only use it for communication purposes. On the simulation middleware, we directly include roscpp and we use the ROS-sharp[5] library inside the HoloLens. ROS-sharp library sends and receives messages in JSON format. For the conversion to/from JSON, we use rosbridge_suite which provides a JSON API to ROS functionality for non-ROS programs[6].

On the HoloLens side, we designed and implemented our mixed reality application entirely in Unity. We utilize Mixed Reality Toolkit (MRTKv2) provided as an open-source project by Microsoft. The toolkit provides many high-level functions for creating a user-interface. Furthermore, with hand-tracking enabled in HoloLens 2, we can query the tracked hand joints which then can be used to control the robot in the internal immersive mode.

3.2. External Holograpic Mode

In the External Holographic Mode, the user can manipulate the robot externally, e.g. by standing in front of it and simply grabbing and dragging the end-effectors to the desired poses. While the user has grabbed an end-effector, the changes in the users' hand position and orientation are copied onto the end-effector. Also, the user can control the grippers by doing an open/close gesture with the thumb and index finger in front of the corresponding gripper.

Additionally, we implemented a sub-mode of the External Holographic Mode, which we call the Task Mode. In this mode, the robot will not execute the manipulation immediately, but the user can rather draw a three-dimensional trajectory and only later let the robot execute it. The drawn trajectory is visualized as a 3D line along with sampled keyframes. The user can manipulate the trajectory by dragging the keyframes, e.g. in order to correct or optimize it. In order to give the user feedback on how the robot will move along the trajectory, an animated and transparent copy of the robot hologram visualizes the predicted movements. This feedback is important, since the robot cannot reach all target poses due to physical constraints. In such cases, the animated hologram will show any discrepancy between the given target trajectory and the calculated, executable trajectory.

3.3. Internal Immersive Mode

In the Internal Immersive Mode, we assume that the user becomes the robot. Thus, the users' hand movements are continuously copied onto the robot. In order to do this, we rely on the HoloLens hand tracking and send the detected poses as target poses to the robot.

To make switching to this mode as intuitive as possible, the user can enter this mode by simply walking into the robot hologram and looking in the same direction. As a safety precaution, the user has to approximately position his/her thumbs at the robots end-effectors positions in order to ensure that the robot will not do any large and rapid movements upon entering the internal mode. As an alternative, the user can also enter this mode by pressing the corresponding button in the hand menu, which will then move the robot hologram to his current position. However, the user still has to position his thumbs correctly in order to start the mode.

When the user is in this mode of operation, the tracked hand pose and joints must be mapped to the robot's end-effectors. Due to the vastly different shapes

[4] new.abb.com/products/robotics/industrial-robots/irb-14000-yumi
[5] https://github.com/dwhit/ros-sharp
[6] http://wiki.ros.org/rosbridge_suite

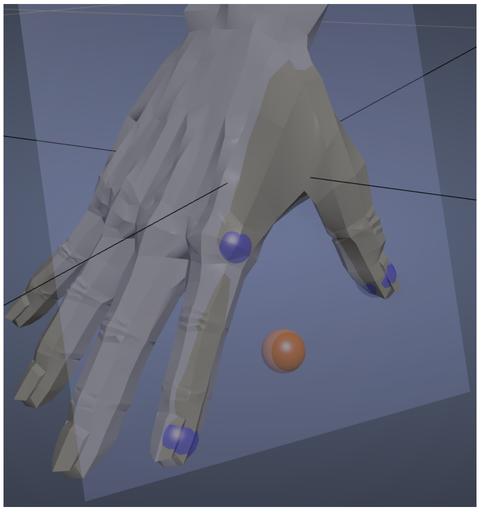

Figure 5: Visualization of the gripper alignment calculation. We use the position of the thumb tip, index finger tip and index finger knuckle (indicated by blue spheres) as detected by the HoloLens. The center of the thumb tip and index finger tip is indicated by the orange sphere.

and physical capabilities of a human hand and the dissimilarity with YuMi's end-effector, such mapping is not trivial. After some testing we decided to focus on aligning the thumb tip and index finger tip with the two fingers of the ABB gripper, since this alignment is important for the user to be able to pick up objects with two fingers.

The target pose of the end-effector is calculated as follows: We average the thumb tip and index finger tip position, which will be the target position for the center of the gripper tips. Then, we constrain the target end-effector orientation such that

1. its left-right axis is parallel to the baseline defined by the thumb tip and index finger tip (i.e. both gripper tips are positioned on this baseline)

2. its forward-backward axis is parallel to the vector between the index finger knuckle and its projection onto the baseline (i.e. the grippers are roughly pointing in the same direction as the fingers).

Additionally, the grippers are opened and closed such that their distance matches the distance between the users' finger tips. Finally, the users' head movements are tracked using the HoloLens's internal tracking algorithm. These movements are copied onto a stereo RGB camera, which is mounted on a 6 DoF robotic arm on top of the YuMi robot. The camera images are streamed to the HoloLens and visualized in the respective eye displays. As a result, the user has a depth sensation of the objects in front of the robot. This sense of depth is required for the user to be able to intuitively grab objects.

3.4. Voice Commands

To make the usage of our app more comfortable, we integrated several voice commands to enable the user to navigate between different operating modes when his/her hands are tied to the control signals in internal immersive mode. Namely, "exit internal mode" voice command exits that mode and places the holographic robot in front of the user and "switch full-screen" changes what user observes when in internal immersive mode of operation. Simply, the voice commands are complimentary to the buttons in the hand menu for the situations where user can not use the hand menu.

3.5. Camera Stream

In the External Holographic mode a video screen is attached in the world view of the user wearing the HoloLens 2. This video screen is made of a rectangular Quad mesh where each frame is rendered as a texture on top of it. The screen can be interacted with as a Hologram, it can be scaled and placed anywhere the user desires.

In the Internal Holographic mode we use the same Quad mesh but re-project it closer to the viewer's range of view. Moreover, the screen is fixed in the vertical axis while following the user's head motion in the horizontal axis.

The head motion is tracked with the HoloLens 2 and used to control a robotic arm that is mounted on top of the robot. On the end of the robotic arm a ZED Mini stereo camera is attached for the video stream.

4. Results and Discussion

We've published a demo video[7] which demonstrates the discussed concepts in action. We will also be publishing our code[8].

All in all, we presented a variety of methods for an intuitive and natural-feeling remote control of a dual-armed, collaborative robot. These preliminary concepts that we prototyped proves that it is possible to create much more immersive controllers utilizing mixed reality.

Improvements are still possible in addressing the stability and accuracy of mapping the movements of a human hand to the robot's end-effector. Furthermore, a user study would help collecting substantial information for more evidence on the benefits of our proposed

[7]https://youtu.be/Vg0ykQulvPg
[8]https://git.htoo.de/jhein/mr-immersive-telemanipulation

methods in regard to a intuitive and natural-feeling interaction with collaborative robots or even help us come up with better control methods. Finally, one exciting direction is to extend our interaction and communication framework towards other collaborative robots that differ in size, number of joints, speed and mobility.

References

[1] G. H. Ballantyne and F. Moll. The da vinci telerobotic surgical system: the virtual operative field and telepresence surgery. *Surgical Clinics*, 83(6):1293–1304, 2003.

[2] S. Duenser, J. M. Bern, R. Poranne, and S. Coros. Interactive robotic manipulation of elastic objects. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3476–3481. IEEE, 2018.

[3] T. Rodehutskors, M. Schwarz, and S. Behnke. Intuitive bimanual telemanipulation under communication restrictions by immersive 3d visualization and motion tracking. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 276–283, 2015.