Immersive Remote Grasping: Realtime Gripper Control by a Heterogenous Robot Control System

Dennis Krupke,* Lasse Einig, Eike Langbehn, Jianwei Zhang and Frank Steinicke
Groups HCI and TAMS, Department of Informatics University of Hamburg, Germany

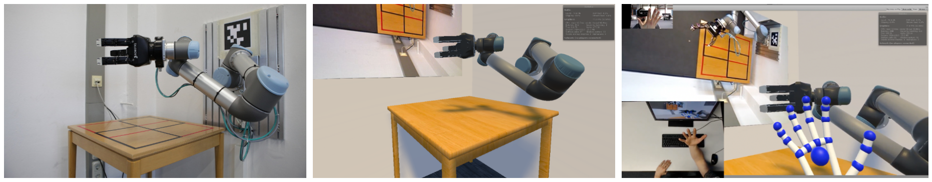

Figure 1: *The real laboratory environment (left) and the virtual corresponding scene (right). The posture of the virtual robot is synchronized by messages with the current posture of the real robot, which are sent via the ROSbride to Unity. A virtual screen at the left wall shows a webcam stream from a ceiling camera above the robot for safety reasons. An operator was recorded during the use of the teleoperation system (right). Head movements are shown by a webcam image in the upper left corner, as well as hand movements. A top-view camera recording in the lower left corner of the figure shows the hand postures with more detail. In the virtual scene a hand skeleton (LMC Orion SDK) mirrors the state of the tracked real hands.*

Abstract

Current developments in the field of user interface (UI) technologies as well as robotic systems provide enormous potential to reshape the future of human-robot interaction (HRI) and collaboration. However, the design of reliable, intuitive and comfortable user interfaces is a challenging task. In this paper, we focus on one important aspect of such interfaces, i.e., teleoperation. We explain how to setup a heterogeneous, extendible and immersive system for controlling a distant robotic system via the network. Therefore, we exploit current technologies from the area of virtual reality (VR) and the Unity3D game engine in order to provide natural user interfaces for teleoperation. Regarding robot control, we use the well-known robot operating system (ROS) and apply its freely available modular components. The contribution of this work lies in the implementation of a flexible immersive grasping control system using a network layer (ROSbridge) between Unity3D and ROS for arbitary robotic hardware.

Keywords: human-robot interaction, tele-operation, virtual reality

Concepts: •**Human-centered computing** → **Virtual reality;** *User interface programming;*

1 Introduction and Background

Using a head-mounted display (HMD), one can experience 3D visualization of computer-generated virtual environments or camera-based, streamed real-world scenes in an immersive way. The degree of immersion in such a setup is very high since visual information from the real surrounding of the operator is presented by the HMD. In particular in combination with position and orientation tracking of the operator's head, HMDs further increase the immersive sensation by providing motion parallax cues. Motion parallax cues have been shown to be advantageous for operators of an unmanned guided vehicle (UGV) [Kruckel et al. 2015]. Trials on using an HMD and a UGV in an augmented reality (AR) setup for teleoperation were successful, but were restricted by limited network bandwidth and high latency. As a matter of fact, high latency in the control loop decreases the controllability and the quality of the experience in such a system. In order to reduce the workload at single instances of a shared autonomy system, distributed network-based components or cloud systems have been used. Lorenzo Peppoloni et al. [Peppoloni et al. 2015] integrated an HMD into a ROS/Unity-based system to present a stereoscopic scene from an egocentric view of a robotic manipulator to the operator. The control of the miniaturized manipulator is performed by utilizing a Leap Motion controller (LMC), mounted at the HMD, which recognizes hand gestures and postures as control input. Another approach for high level grasping control which utilizes the LMC is presented by Jin et al. [Jin et al. 2015]. The integration of the LMC and an HMD for interaction with 3D objects is already done successfully. In combination with full head-tracking of the user, it allows manipulation and inspection of CAD models in 3D [Beattie et al. 2015].

2 System Description

The general system architecture follows figure 2. The cooperation between human and robot is guaranteed by the integration of Unity3D at the human-side and ROS at the robot-side of the system. The communication between both sides is provided by the ROSbridge, which uses WebSockets for the protocol and JSON strings as a container for information. The user interface is based on a closed-loop action-perception cycle, which is embedded in a VR scene presented to the user with an HMD. By utilizing positional and rotational head tracking users have the freedom to explore the scene in a more natural way, than in a classical 2D view on flat screens. Manipulation tasks benefit from the possibility to change the point of view in order to avoid occlusion of regions of interest by parts of the scene. Humans at the remote side of the system can take over control of the robotic hand by placing the controlling hand into a semi-transparent sphere in VR. For reasons of comfort and reliability, once the controlling hand is placed in the sphere it follows slow motions of the user's hand. Meanwhile placing the hand inside

*e-mail:krupke@informatik.uni-hamburg.de

VRST '16, November 02-04, 2016, Garching bei München, Germany
ISBN: 978-1-4503-4491-3/16/11
DOI: http://dx.doi.org/10.1145/2993369.2996345

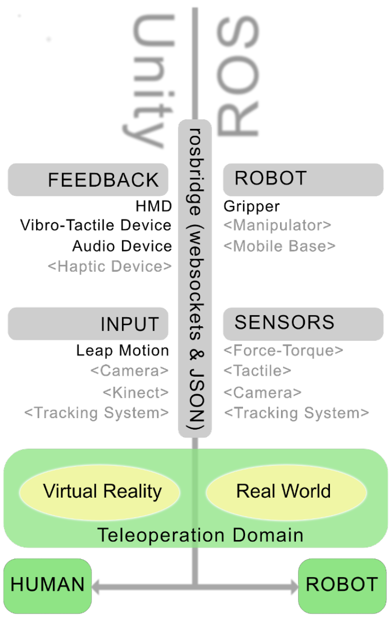

Figure 2: *A general system architecture for integrating ROS and Unity3D. The architecture is extendable and robotic components are easily exchangable.*

the control-sphere the posture of the real human hand is analyzed and transduced to high-level control signals for the robotic hand, which can easily be applied to different gripper hardware. As input device for the hand tracking an LMC is utilized, which is mounted using a modified version of the VR mount of the manufacturer of the LMC directly at the front side of the Oculus Rift.

3 Implementation of Control and Feedback

The ROS side implementation of the ROSbridge is maintained by the community. For the integration into Unity3D, a ROSbridge script is implemented in C#. ROS messages are implemented in C# as a ROS message layer. Additionally the Newtonsoft JSON library for .NET is utilized to generate the data strings from the message objects. Using the JSON library, a custom serializer for nested objects has been implemented. A Robotiq 3-finger adaptive gripper is controlled by *ROS gripper actions*, e.g. closing or opening the gripper to a certain percentage, which are available for a wide range of different grippers. The real robot issues its current state by publishing ROS messages on the `/jointstate` topic. At the user interface side visual feedback is provided when entering or leaving the control sphere and by mirroring the current state of the robot using the virtual robot model and a video stream of the laboratory.

4 Results and Discussion

The user interface has been evaluated in a first test. Using the metaphor of the *control-sphere* seems to be promising. Additionally, the ability to arrange the position of the control-sphere according to own preferences is reported as comfortable by participants of a pre-study for testing the user interface. The latency of the system in closed-loop control has been measured by implementing a *ping* program. By pressing a specific key on the keyboard, a ROS message is generated and sent to the ROS side. A response is generated by the addressed ROS node and the message with its original header is sent back to the Unity implementation. The time of arrival is compared with the point of time of the corresponding key press. A roundtrip latency of ≈ 40 ms can be expected for messages, which is low enough for slow hardware like typical robotic grippers. The experiment was conducted within a 100 mbit network within one building between the first and the third floor. Since there are two pairs of serialization and deserialization nodes we can expect that there is room for improvement of the specific code and by using more powerful computers than we used in the experiment. Realtime rendering of the program, and thus processing of the control and feedback function, was running with 75-85 fps, which is large above the minimal requirements for immersive experiences with the Oculus Rft DK2 without increased risk of simulation sickness in VR.

5 Conclusion and Future Work

We presented a flexible concept for immersive and natural teleoperation systems using both, state-of-the-art robot control software (ROS) and high-quality user interface software (Unity3D). A prototype has been implemented and a first test of the user interface for controlling a robotic hand has been performed. The metaphor of the *control-sphere* for a novel 3D user interface mechanism has been introduced. Instead of simple direct operating, the future applications of the system will focus on cooperation between humans and robots and on placing the robot between two humans, one at the robot side and one at the remote side in collaborative task execution. By utilizing different levels of autonomy in future, the ratio between flexibility and easy controllability is adjustable [Burdea 1999]. Target prediction in a human-computer interaction task will improve latency and reliability of the system [Lank et al. 2007].

Acknowledgments

This research was partially funded by the German Research Foundation (DFG) and the National Science Foundation of China (NSFC) in project Crossmodal Learning, TRR-169.

References

BEATTIE, N., HORAN, B., AND MCKENZIE, S. 2015. Taking the leap with the oculus hmd and cad - plucking at thin air? *Procedia Technology 20*, 149 – 154. Proceedings of The 1st International Design Technology Conference, DESTECH2015, Geelong.

BURDEA, G. 1999. Invited review: the synergy between virtual reality and robotics. *Robotics and Automation, IEEE Transactions on 15*, 3 (Jun), 400–410.

JIN, H., ZHANG, L., ROCKEL, S., ZHANG, J., HU, Y., AND ZHANG, J. 2015. A novel optical tracking based tele-control system for tabletop object manipulation tasks. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ*, 636–642.

KRUCKEL, K., NOLDEN, F., FERREIN, A., AND SCHOLL, I. 2015. Intuitive visual teleoperation for ugvs using free-look augmented reality displays. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, 4412–4417.

LANK, E., CHENG, Y.-C. N., AND RUIZ, J. 2007. Endpoint prediction using motion kinematics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, New York, NY, USA, CHI '07, 637–646.

PEPPOLONI, L., BRIZZI, F., AVIZZANO, C., AND RUFFALDI, E. 2015. Immersive ros-integrated framework for robot teleoperation. In *3D User Interfaces (3DUI), 2015 IEEE Symposium on*, 177–178.