SDM5002 Report: VR-enhanced Remote Manipulation with Robot Arm

JING Yonglin
School of System Design and Intelligent Manufacturing
Southern University of Science and Technology
ShenZhen, China
12132669@mail.sustech.edu.cn

***Abstract*—In SDM5002 (2022 spring), my group proposed a virtual reality enhanced remote manipulation system with robot arm. This paper aims to provide the following contents: First, a general introduction on project background. Second, the design of the system. Last, details worth discussing based on my share of work load. Hopefully this paper can serve as a useful information reference for similar works in the future.**

***Index Terms*—virtual reality, robot arm, remote manipulation**

I. Introduction

A. Need for Teleoperation

With the advance of robotics and automation in recent years, more and more real-world workloads are making a turn towards teleoperation and automation. In some special scenarios, the demand for off-site teleoperation is fairly high. For example, safety-critical scenarios such as explosive ordnance disposal, mine exploration and earthquake rescue. In these scenarios, an off-site teleoperation solution can ensure the safety of the operators. Another scenario is that the professionals is not nearby when a time-sensitive event happens. For example, a person unfortunately gets a cardiac arrest in a shop but no one nearby knows how to use the AED (Automated External Defibrillator). If off-site teleoperation is possible, we can have professionals sitting miles away from the site and manipulate the AED properly just in time, which may save the person's life.

B. Problems in Previous Solutions

We can already see some efforts being made in this field, such as robot platform designed for explosive ordnance disposal. Since things on site are still manipulated manually, the way the operator perception and interaction with on-site equipments are of great importance in this system. A common and easy solution is to let the operator percept the site with a traditional 2D display and interact with joysticks. This solution, though easy to build, can have several fatal drawbacks:

- 2D display eliminates 3D information, which can cause trouble when performing position-critical manipulation such as aligning tubes. Having multiple on-site cameras may ease this problem but is still tedious to use.
- Normal joysticks only provides 2D input, but common general-purpose robot arm have 3, 6 or even higher end effector DoF (Degree of Freedom). Control a high DoF device with a low input device can be counter-intuitive and tedious.

C. Changes Brought by VR Devices

VR industry has undergone rapid development recently. Normally a standard VR device set includes a headset with 3D rendering capability and two joysticks, and all 3 devices are able to track and report its 6D pose information. With these capabilities, we can see that VR device can bring some major changes to the previously mentioned problems.

- 3D display can enable depth perception for the operator, thus providing a more intuitive way to percept the 3D structure of the operation site.
- With 6D pose reported from the joystick, we can now have a more intuitive way to control a high DoF device such as a 6D robot arm.

D. Project Goal & Setup

This project aims to provide a demonstration on the enhancement brought by VR devices in a teleoperation system. In the project, to simulate a general-purpose operation platform, the on-site platform is equipped with a 6-DoF robot arm as the manipulator and a RealSense D435 stereo camera as the sensor. And on the remote control platform, we use META QUEST 2 (or known as "OCULUS QUEST 2" previously) as the VR solution. The project mainly focus on two things:

- Users will be able to percept the 3D structure of the site rendered by VR headset in realtime. 3D data is acquired by the on-site stereo camera.
- Users will be able to control the 6-DoF robot arm intuitively via the movement of the joystick held by the user's hand.

II. System Design

Fig. 1 gives an overview on the overall system design. The following part will give explanation to the hardware and data pipeline respectively.

A. Hardware

For the remote platform, we use a powerful gaming laptop for computation since rendering in VR can be a computational expensive job. META QUEST 2 is used with OCULUS link

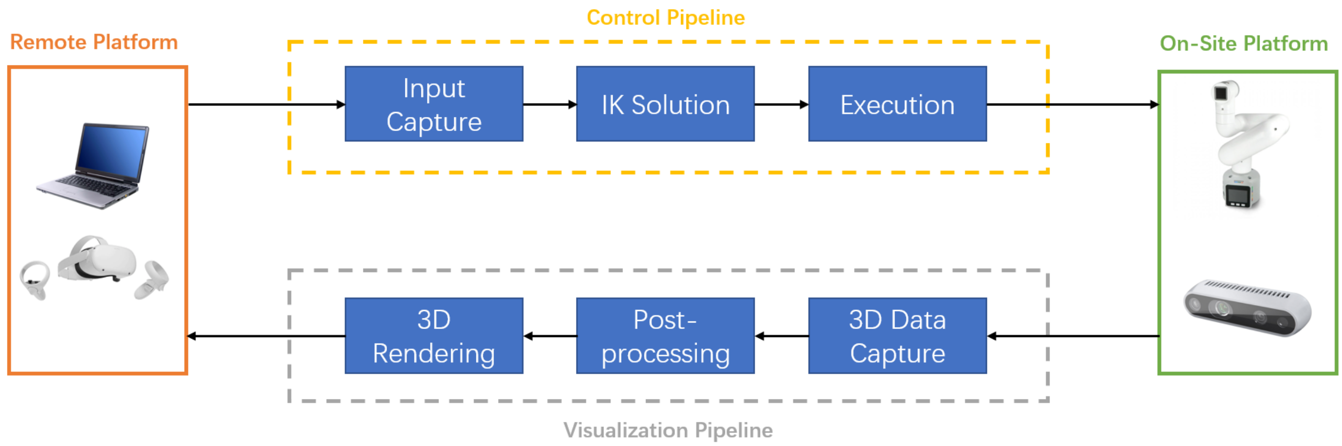

Fig. 1. System design overview

enabled (streaming data by wire) to ensure maximum display quality.

For the on-site platform, we use myCobot 280 M5 from Elephant Robotics as the 6 DoF robot arm and Intel RealSense D435 as the stereo camera. MyCobot 280 M5 is a fairly cheap choice for a desktop collaborative robot arm (3999 RMB). And Intel RealSense D435 is just a common solution for 3D data acquisition.

In our project, the on-site platform is connected to the remote platform with cables. Therefore, the laptop can directly send command to the robot arm and acquire 3D data from stereo camera. In scenarios where wire connection is unavailable, another computing unit can be added to the on-site platform, relaying the control and visualization data via a wireless solution. Our implementation is designed with this issue in mind. See the next section for details on the implementation.

B. Data Pipeline

For a teleoperation system, there are two vital element to concern: passing user control command to the site and transferring on-site data back. In our system, two data pipeline are designed to give solutions to these issues.

The first one is the control data pipeline. In this pipeline, user controls are first captured by computer, including 6D joystick movement, trigger press and button press. Then a new end effector position is computed based on the control command. After that, inverse kinematics is leverage to obtain a solution in joint space (target joint angles). At the end, the joint angles are sent to the robot arm for execution.

The other pipeline is visualization pipeline. At the beginning of this pipeline, 3D data, including depth map and RGB image are captured by the stereo camera on site. Then the post-processing stage is responsible for preparing the data for rendering, including aligning RGB and depth frames, converting to 3D mesh and other operations for better rendering quality. And finally, the prepared data is transferred to the VR headset for rendering.

III. Implementation

Fig. 2 gives an illustration on the implementation diagram. The whole system can be split into two major blocks. One is Unity engine, and the other is a robot arm control client written in Python. Between them is the Open Sound Control protocol for data exchange.

A. Arm Control Client

There are mainly two thing done in the arm control client:

- Continuously report the angles of all arm joints to the Unity so that the simulated arm in Unity can synchronize with the real one.
- Take in user command sent from Unity, give IK solution and perform the execution.

Two threads are used for the two tasks. The joint status thread query arm status with a frequency of 500 Hz and report to Unity via a UDP-based OSC (Open Sound Control) client. The other thread runs a UDP-based OSC sever listening to new commands from Unity. With a new command incoming, a new end effector position and its corresponding IK solution will be computed. Then the actual control command will be sent to the arm via the API provided by the manufacturer.

B. Unity

We choose to use Unity engine because of its convenience regarding VR-related application development. In this project, Unity engine is used for the following tasks:

- Acquire user input on META QUEST 2 via OpenXR standard. Then bridge those input to Unity's input system for application development.
- With the Unity wrapper provided by Intel RealSense SDK, acquire and render the 3D data stream from the D435 stereo camera.
- With Unity Roborics package, load URDF model of the arm, provide physics simulation and control interface for the simulated arm in Unity.
- Render all virtual content on the VR helmet display.

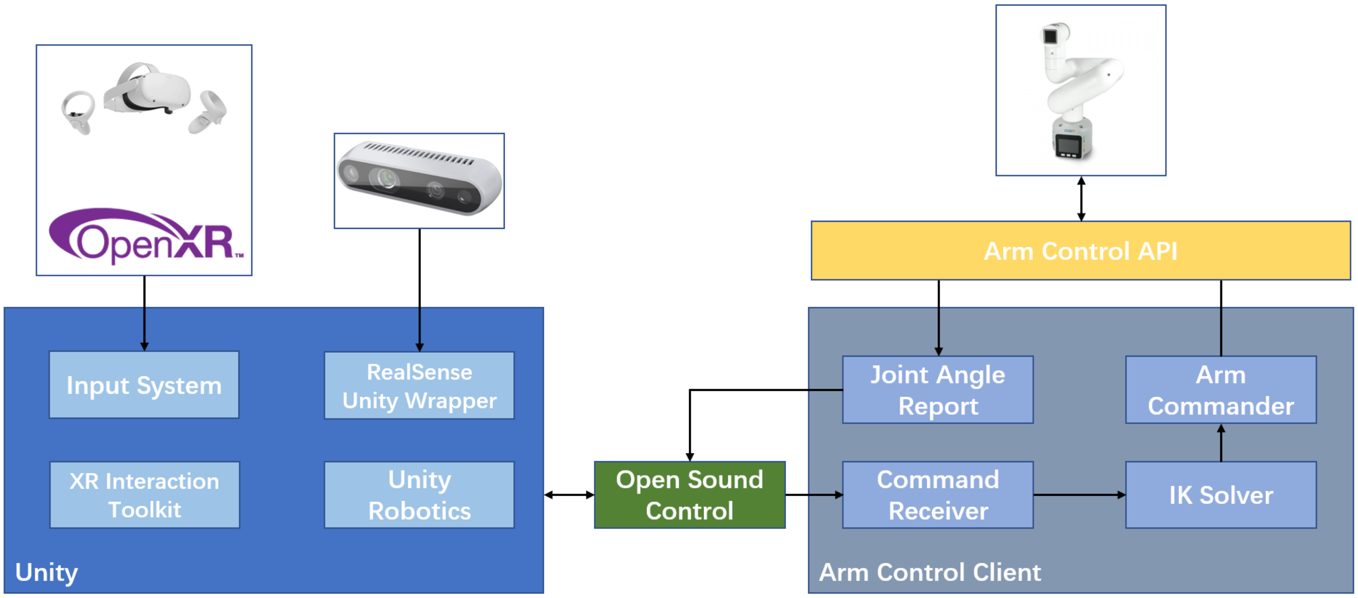

Fig. 2. Implementation overview

The XR Interaction Toolkit mentioned in the graph is just a Unity package provides many convenient basic functionalities for XR (VR + AR) development.

IV. Future Work

For the future work, regarding the hardware, a better robot arm can be used for replacement of the myCobot 280 M5, which is a choice with limited budget. And better stereo camera or other 3D sensors can be utilized. For the software, currently the project use the out-of-box 3D reconstruction algorithm from Intel RealSense SDK. In the future, a more elegant 3D reconstruction algorithm may be developed for better visualization experience. Also, the current software is not tested with a wireless solution. With different network environment, the networking solution may need changing accordingly.

V. Acknowledgment

I want to thank Prof. Hong for teaching this course and providing the chance for this project. I am personally interested in the topic of this project and the course gave me the chance to actual implement it. I also thank all my teammates for all the collaboration and work they do.