Real-Time Hand Tracking Virtual Reality Interface

UNION COLLEGE

Lu Zhang. Student

Matthew Anderson, Advisor

CS

Introduction:

The goal of this project is to achieve controlling a ROS robot by using real-time hand tracking. This is a continuous work based on the previous research projects. Due to the fact of the epidemic, we are not able to get access to the real robot. Instead of that, we run the robot in a simulator. The system needs an Oculus Quest VR headset and graphic-capable PC to run, and it is base on Unity Engine.

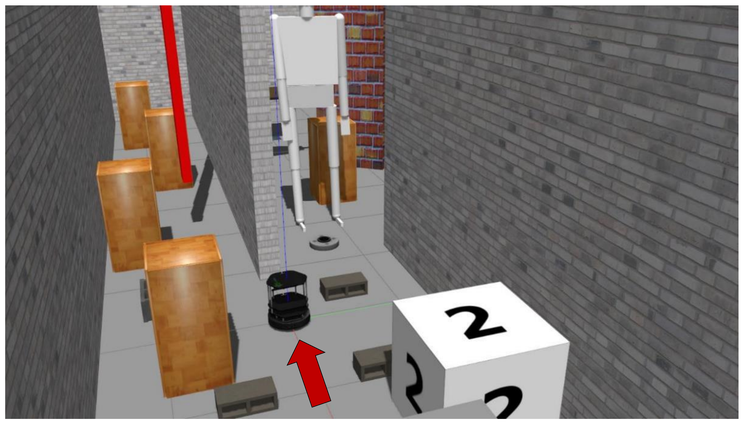

Figure 1. the simulated world that robot runs in.

Gesture Detection:

The hand tracking SDK from Oculus tracks 24 different bones on hands. For each bones, it tracks its Transform data, including Position, Rotation, and Scale. By utilizing bones' Position data, we are able to calculate the distance between each bone and wrist.

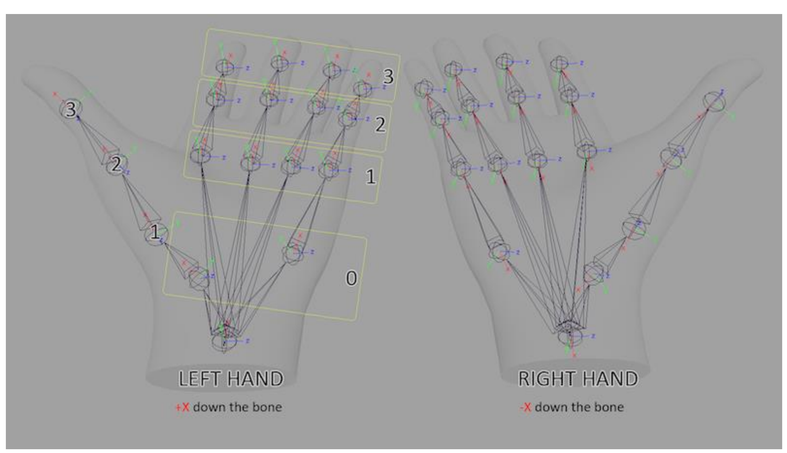

Figure 2. Oculus Documentation. Tracked hand joints.

We call it bone's local position, and then compare user's data and saved gesture data like shown Equation 1.

$$D = \sqrt[2]{\frac{\sum_0^N (H_{local\ position} - G_{local\ position})^2}{N}}$$

Equation 1. The equation used to measure the difference between saved gesture data and real-time gesture data.

$H_{local\ position}$ is the real-time local position of each bone, and $G_{local\ position}$ is the pre-recored one. In each frame, we calculate square root errors of the difference between each bone's local position. If the result is less than a threshold, then we can say if the user perform a specific gesture.

Control implementation:

Controlling robot requires users understanding robot's real time status. At mean time, robot also needs to identify and follow users' command. We use a Homunculus Model to solve this problem. User's state and robot's state are linked through a Virtual Reality Control Room(VRCR). This allows a human user has the same view point and correlation to the robot without more complicate system.

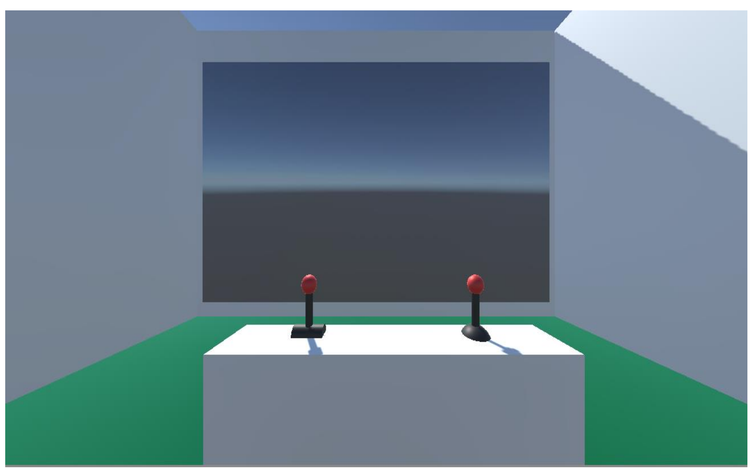

Figure 3. the Virtual Reality Control Room.

Completed Prototype:

Thanks to the previous research projects, we can generate visualized mesh inside Unity base on the data from RGB and depth cameras on the robot. Inside the VRCR, users can control robot's speed and direction through control stick. Users can make the control terminal hide and appear through performing hand gesture.

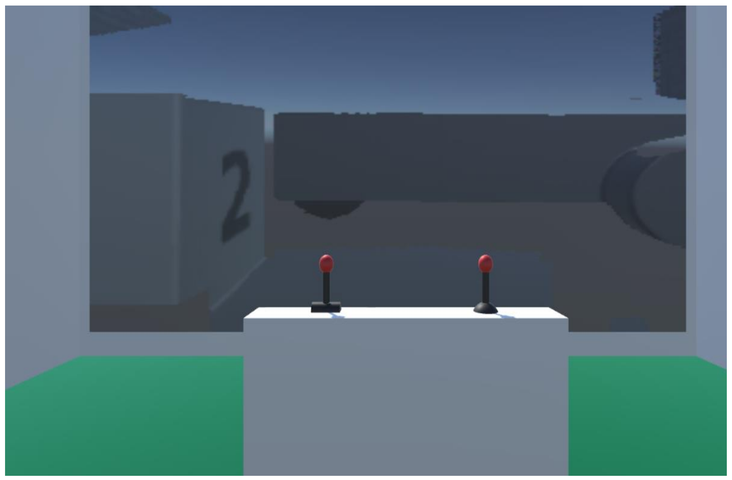

Figure 4. the main screen shows what robot is seeing in the simulator.

Further Study:

The real-time generated mesh is laggy and not so stable sometimes. It is better to have a pre-load map in Unity for better user experience and future development. Improve the view inside of the control room for better understanding the status of robot and movement control. We can also simplify the control schema to one joystick controlling speed land direction at the same time.

Acknowledgments:

Thank you to Union College, Unity, and Davenport Research Fellowship

References

[1] "Baxter's Homunculus: Virtual Reality Spaces for Teleoperation in Manufacturing", Jeffrey I. Lipton, Aidan J. Fay, and Daniela Rus, IEEE Robotics and Automation Letters, VOL. 3, NO. 1, January 2018.