INTEGRATION OF AVATAR TECHNOLOGY PROTOTYPES WITH ROBOT TELEOPERATION

Atreya Rawat[1], Grace Wang[2], Michael Ferdman[2]
Half Hollow Hills High School East[1], Princeton University[2], Stony Brook University[2]

INTRODUCTION

BACKGROUND

Avatar technology is part of an emerging field that examines how humans can participate in remote activities through an external source, without being physically present. This work investigates how the Turtlebot can serve as an avatar, allowing someone to explore a distant location and interact with the environment through hands attached to the robot. The practical applications of this work may be applied to remote classroom instruction, service work, or even dangerous rescue operations.

Figure 1: Turtlebot 2 with Kobuki base

Figure 2: Oculus VR Headset

OBJECTIVES

- Connect to the turtlebot from a remote computer and be able to control it from a webpage using a gamepad.
- Provide a visual representation of the users' hands on top of the live video feed, which is transmitted to the VR headset.

ROS (Robot Operating System)

- The Robot Operating System (ROS), an open-source meta operating system for robots, was used to teleoperate the turtlebot according to navigation inputs received from the user through the gamepad.

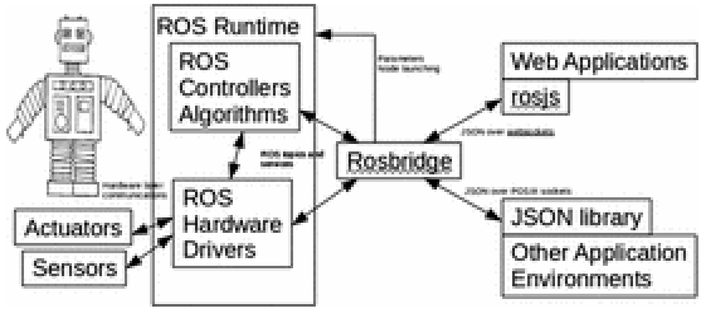

Figure 3:Diagram depicting transfer of information from the web to the robot.

METHODS

WEB DEVELOPMENT

- The HTML/JavaScript code was used to connect to Roscore via the Rosbridge, an API that provides the link that allows the web server to communicate with ROS to send messages to the robot. Commands are published to a teleop topic in the form of Twist geometry messages. Twist messages consist of two vector components, linear and angular, with each one having three fields: x, y, and z. The mobile base manager of the robot, which is subscribed to the same teleop topic, thus receives messages from the webpage and drives the robot accordingly.
- Websockets, a technology which allows for client communication with the server through a Websocket handshake, were used to connect the robot and headset.
- This work utilized the HTML5 Gamepad API to connect to the Xbox 360 Controller and query the state of the gamepad, including buttons pressed.

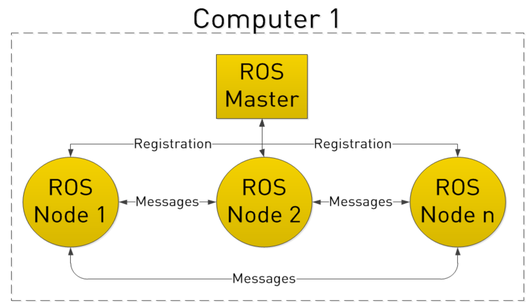

Figure 3: Diagram illustrating communication and transfer of messages between ROS nodes

VIRTUAL REALITY

- A-Frame is a platform for Virtual Reality development on the web that uses HTML and is compatible with numerous devices, such as the Oculus Rift.
- Hand tracking was achieved by using the LeapMotion JavaScript API and incorporating live data from the Leap Motion sensor in the A-Frame environment. The hand-tracking code provides the position and speed of the palm and five digits of each hand in frames.
- The user observes the scene from the robot's perspective using the VR headset, in this case, the Oculus Rift, which allows the user to have a 360 view of the robot's surroundings.

Figure 4:Demonstration of hand tracking using leap motion sensor

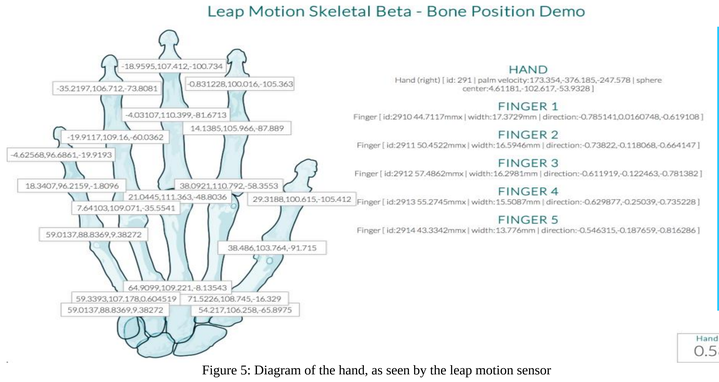

Figure 5: Diagram of the hand, as seen by the leap motion sensor

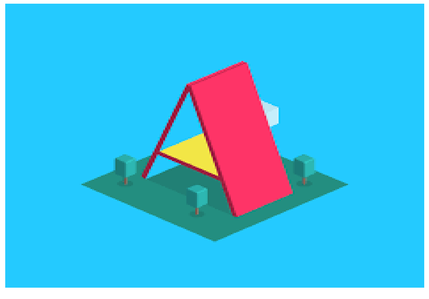

Figure 6: A-Frame Logo

CONCLUSIONS

- Integration of live hand-tracking and the Turtlebot's video feed in a VR headset was achieved, complete with movement control of the bot through an Xbox 360 controller connected to a remote computer.

REFERENCES

"Gestures." *Javascript | Leap Motion Developers*, developer-archive.leapmotion.com/documentation/v2/csharp/devguide/Leap_Gestures.html.
"Introduction – A-Frame." *A-Frame*, aframe.io/docs/0.8.0/introduction/.
"LeapJS." *Javascript | Leap Motion Developers*, developer-archive.leapmotion.com/javascript.
"Using the Gamepad API." *MDN Web Docs*, developer.mozilla.org/en-US/docs/Web/API/Gamepad_API/Using_the_Gamepad_API.
"Wiki." *Ros.org*, wiki.ros.org/ROS/Tutorials.
"Wiki." *Ros.org*, wiki.ros.org/DevelopersGuide.