UC-209

Remote Presence Robot

Andrew Goeden, Aaron Newson, Anna Song, Hajar Zemzem, Mohammed Rehaan, Pamir Ahmad, Tam Dang

Abstract

These robots can simulate a person's presence attending a meeting or help people with disabilities to interact with their friends or family if the world goes back to online interactions only. Our motorized robot has a screen, microphone, webcam, and speaker to perform video chat with the client. The robot can move about the environment by remote user input with a controller. The video call with the robot will have customization functionalities such as filters and live backgrounds, and the robot will have object tracking capabilities to center itself on the person in frame.

Introduction

Online interactions strip away the real human connection and make communication artificial. Many people are forced into these interactions which inhibits them from the advantages of communicating in person.

This remote presence robot aims to help the people who are forced to communicate online or the people with disabilities by adding a personalized feel to the user and help them to have meaningful interactions with their friends, family or in a classroom or work environment.

Research Question(s)

- How do we send video, audio, and controller input data from one computer to the other?
- What size should the robot be in order to effectively move about the environment and simulate a person's presence?
- What building materials are we using to make the robot?

Components and Methods

- **Raspberry Pi 4B -** Controls the robot and connects to the client servers.
- **Docker –** Packaging the client servers.
- **NodeJS & WebRTC –** Powering the client servers to enable remote connections, controls, and video call functionalities.
- **OpenCV –** Analyze and post-process images to detect faces/objects.
- **SolidWorks –** Designing the whole robot so it can be assembled swiftly.

Results

Design:

The robot frame is designed in Solidworks and made of 80/20 Extruded Aluminum. The gearbox is also custom designed, and 3D printed out of PETG.

The motors are brushless motors that provide us with a good mix of torque and speed as well as sensors that we can hook up to monitor if the motors stall, their speed, and their temperature.

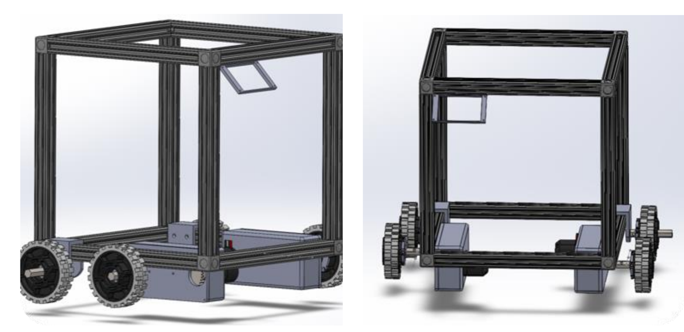

Figure 1: CAD design of all components

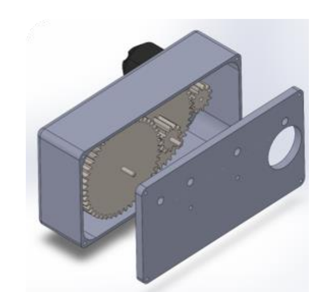

Figure 2: Gearbox CAD

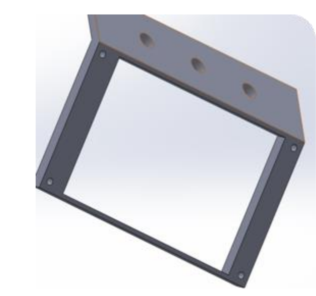

Figure 3: Monitor Cover CAD

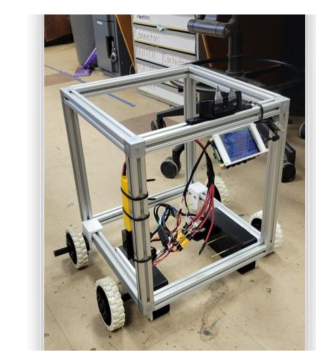

Figure 4: Final Robot Assembled

Client Servers:

Used Node and Express server along with Socket.io to establish client server communication. Using common socket.io events such as connection, join, or disconnect we can signal a connection between the client and server. For real-time communication we used WebRTC, combined with socket.io we were able to setup persistent communication between a client and a server.
WebRTC also depends on STUN and TURN servers, so we setup a TURN server and used a Google STUN server. The TURN server exists only to fall back on in case the client and server are not able to communicate through the STUN server.

- The TURN server was setup using a free service called Viagenie
- The user interface was built using ReactJS
- The icons and assets were obtained from free Figma UI kits.

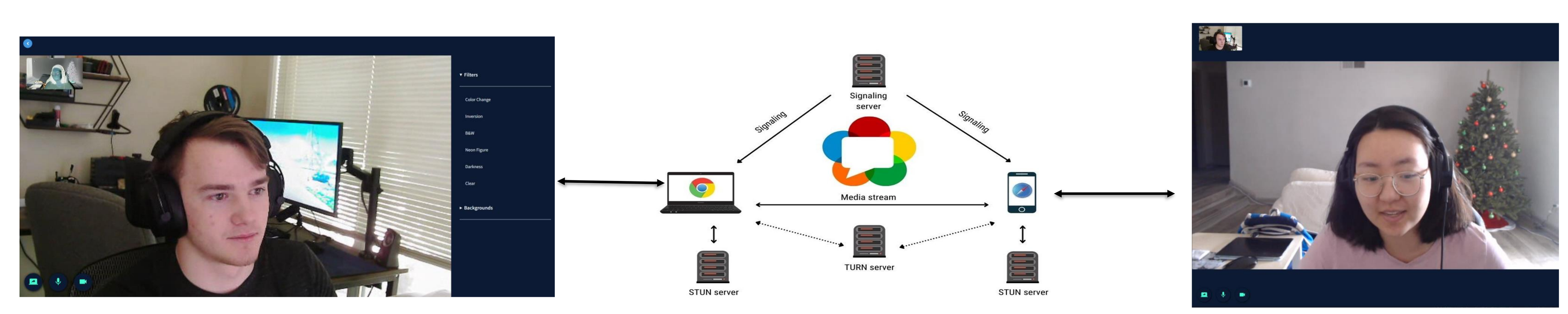

Figure 5: Demonstration of Client servers and the connection model

Robot Controls:

To control the robot remotely, we send input values varying from –1 to 1 inclusively for speed and turn from the client application to the web socket client in the robot application on localhost, then the information would be processed on the Python server. With the input commands from the client, we calculate the corresponding speed and steering values, send them to the Raspberry Pi to rotate the motor controllers attached to wheels, so our differential driven robot can move around the environment with the speed we want to apply on each wheel.

Object Tracking:

Making use of pre-trained weights and OpenCV, people detected in frame would be boxed with their confidence levels and their centers would be compared to the frame's center. Taking the difference between the frame and the object's center we would trigger the motors to turn either left or right towards the frame center until within a certain boundary.

Figure 6: Person Detection 1

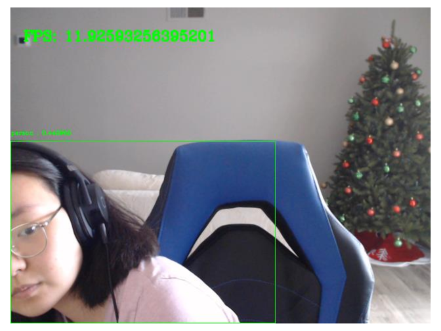

Figure 7: Person Detection 2

How to Improve

- More compact design built for versatility and varied environments
- Omni-Wheel design to reduce wheel scrub as well as 4-wheel drive
- Compartments for storage and convenience
- Safer and more aesthetic design with hidden wires and gearbox
- Object tracking for obstacles as safety measure

Acknowledgments

Electric Vehicle Team – providing a workshop space, tools, and equipment

Contact Information

LinkedIn URLs:

https://www.linkedin.com/in

Pamir Ahmad - /pamir-ahmad-84a0b21a3

Tam Dang - /tam-dang

Andrew Goeden - /agoeden12

Aaron Newson - /aaron-newson

Mohammed Rehaan - /mohammed-rehaan-14ba37164

Anna Song - /asong-8

Hajar Zemzem - /hajarzemzem

KENNESAW STATE UNIVERSITY
COLLEGE OF COMPUTING AND SOFTWARE ENGINEERING