ServoBot - Robotic Waiter Ecosystem with a Nodal Network of Order-Taking Modules

Aryan Agarwal, Kenson Nguyen, Marcell Veszpremi, Utkarsh Kumar

Our Team

Aryan Agarwal
4th Year
Electrical Engineering
IoT & Component Integration

Kenson Nguyen
4th Year
Electrical Engineering
IoT & Speech Processing

Marcell Veszpremi
4th Year
Electrical Engineering
Robotics

Utkarsh Kumar
4th Year
Electrical Engineering
Visual Processing

Context and Larger Purpose

- The COVID-19 pandemic led to a rise in staffing shortages
- We envision a future where diners can have a seamless experience from ordering food to getting it on their table directly from the kitchen
- This concept is demonstrated in our project - a semi-autonomous delivery robot alongside order-taking modules at each table
- This not only adds automation to the restaurant industry, but provides cheap and reliable labor

Other uses for this project

- This project can be expanded into several other sectors - we see a crucial use case for it in elderly hospitals or homes
- Voice recognition in each order taking modules can be provided to elderly/disabled persons who can use it to call for medication/food/etc.
- The delivery robot can be modified to bring medication or food to those with limited mobility
- This solves the problem of having to hire assistants to help you 24/7

Ecosystem Design

- Our team will solve the issues mentioned by coming up with a high-tech solution: **The ServoBot**
- ServoBot will be a nodal network consisting of three main components:
  - **The OrTak:** A expandable set of order-taking modules (RPis) placed at each table to take orders via speech and to pay bills
  - **The KitchenNode:** A screen module that will display all the orders and give the WOK-er the command to go
  - **The WOK-er:** A robot capable of delivering orders from the kitchen to a table

The OrTak – Order Taking Nodes

- Present on each of the tables in the restaurant
- Main processing done with Raspberry Pi Zero W
- Takes food orders from guests using a microphone
- Uses the Google Speech-to-Text API to process the order
- Sends orders to the Kitchen module for the kitchen staff
- For the demo, we will have the OrTak powered by the laptop and orders will be taken with a microphone on the Raspberry Pi
- Will also include bill payment and ID checking capabilities (future goal)

The KitchenNode

- Touch-enabled screen that shows all the orders that have been placed
- Includes details such as table number, item, quantity, and special requests
- The kitchen staff will be able to select items that are ready and send them out for delivery using a button
- Uses Kivy and KivyMD libraries for the GUI
- Receives order from OrTak via MQTT
- For the demo, we will be using a laptop to emulate the Kitchen

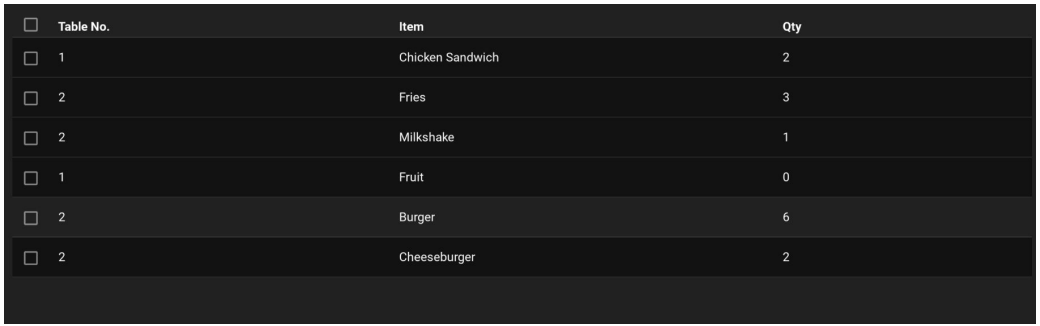
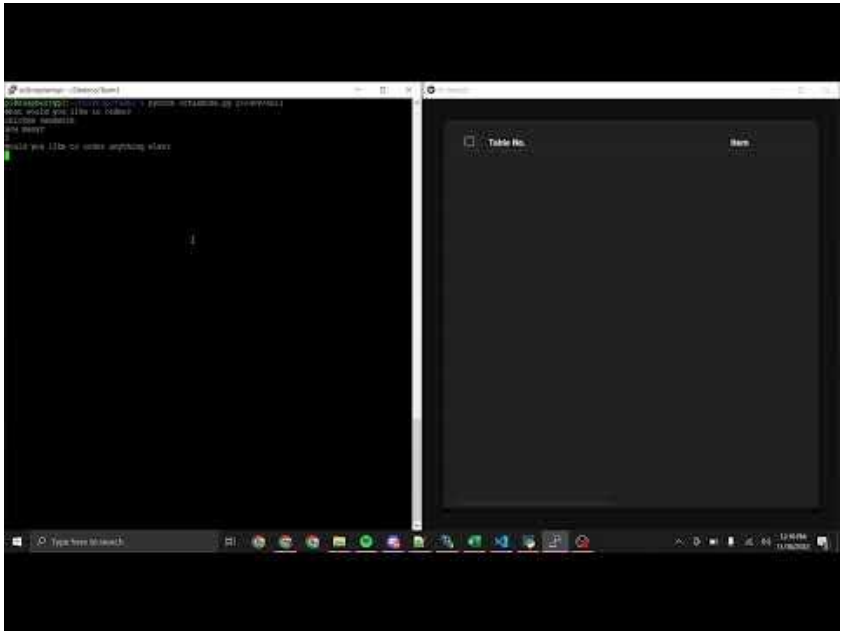


Table No.	Item	Qty
1	Chicken Sandwich	2
2	Fries	3
2	Milkshake	1
1	Fruit	0
2	Burger	6
2	Cheeseburger	2

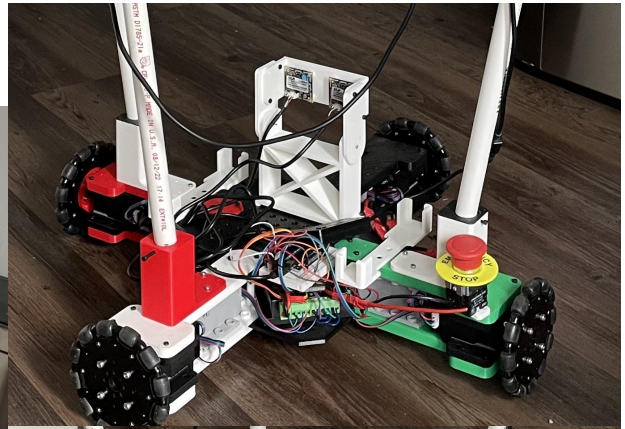
KitchenNode + OrTak Demo

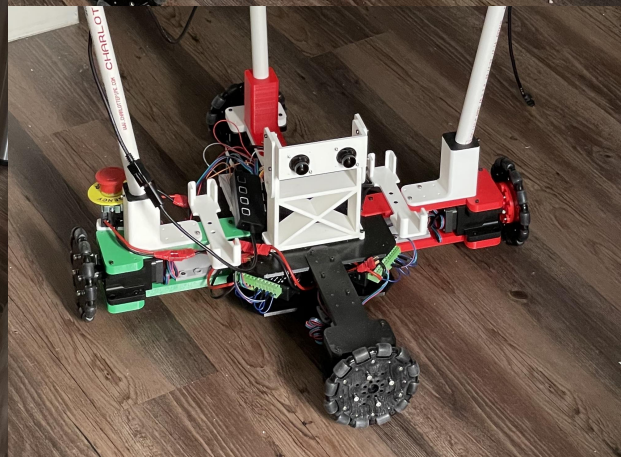


The WOK-er

- Four-wheeled, omnidirectional, and camera enabled semi-autonomous delivery robot
- Uses AprilTags to localize itself in restaurant
- Utilizes stereovision and path planning for optimal movement in restaurant
- Capable of gesture and speech recognition to take commands (future goal)
- Movement to tables dictated by the KitchenNode

The WOK-er Demo





The WOK-er Demo cont

System Integration Diagram

Or-Tak Module

Main Processing Unit

Raspberry Pi Zero - utilizing MQTT libraries and speech recognition APIs

KitchenNode

GUI

Utilizing KivyMD and Kivy libraries to develop a front-end interface for the kitchen staff to view all orders

WOK-er Module

Main Processing Unit

Raspberry Pi Zero since it runs on Python thus can utilize vision processing libraries

Motor Control Module

Arduino Nano controls *four omnidirectional* wheels which allow for movement in all directions

System Process Flow

Or-Tak Module

Receives the food/drinks order from the guests and parses through it and sends it to the Kitchen

KitchenNode

The order is then displayed on the screen where the kitchen staff is able to check off each item when prepared
It also sends a command to the WOK-er once all the food for a table is ready

WOK-er Module

It receives the table number and command to deliver the order and starts locating the april-tag associated with that table
It then starts moving towards that april-tag while avoiding the obstacles along the way
After delivering the food, it comes back to its original location

Speech Recognition

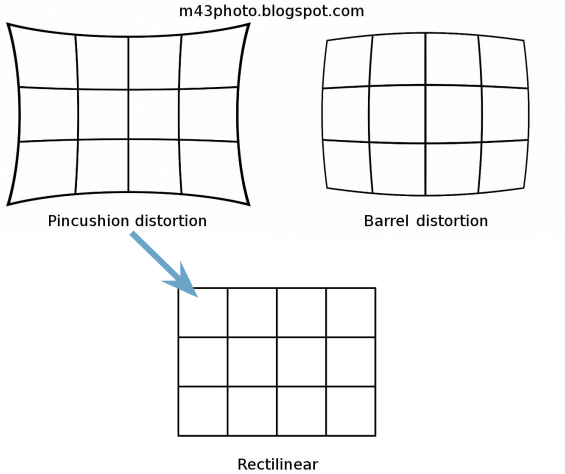
- Leveraging Google Speech-to-Text APi through SpeechRecognition library
- Collecting speech using inexpensive USB microphone
- Currently wait for each speech input and send results over MQTT
- Found that results are quite good, though must make sure internet connection is stable
- Transcription time a few seconds, acceptable for our use case
- Will dial-in microphone thresholds in future to account for noisy environments

AprilTags Localization

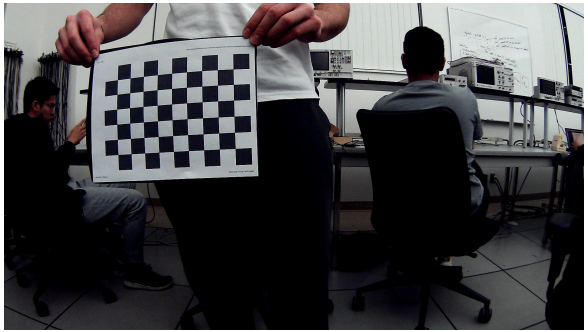
- AprilTags are a form of fiducial markers used for the robot to localize itself
- The advantages of AprilTags are:
  - They can be uniquely identified (unlike commonplace objects, such as tables)
  - They are designed to encode far smaller data payloads allowing robust detection
  - Can be detected from longer ranges than QR codes

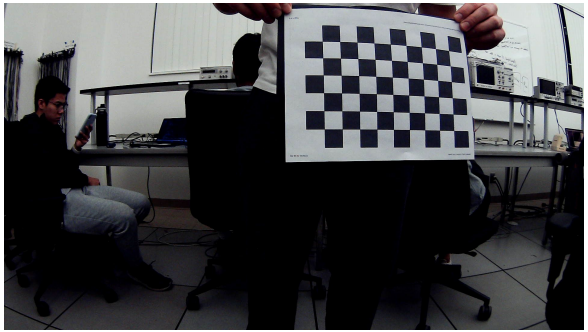
Camera Calibration

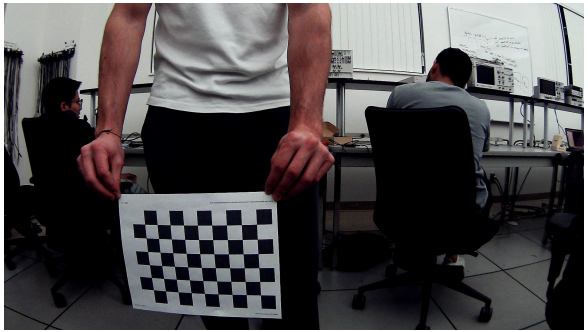
- Compensation for fisheye distortion.
- Obtained K matrix for camera specs:
- `K=[[1053.9492767154009, 0.0, 951.950093568802], [0.0, 1052.5528725501529, 465.04595064900246], [0.0, 0.0, 1.0]])`



Total 30-40 images captured for calib.

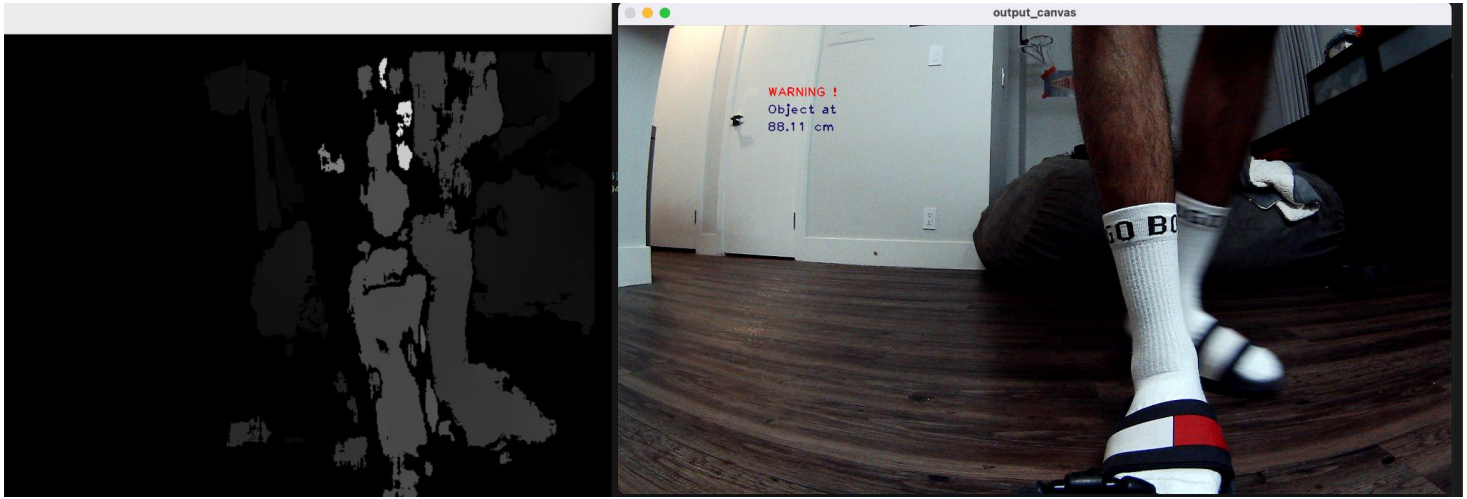




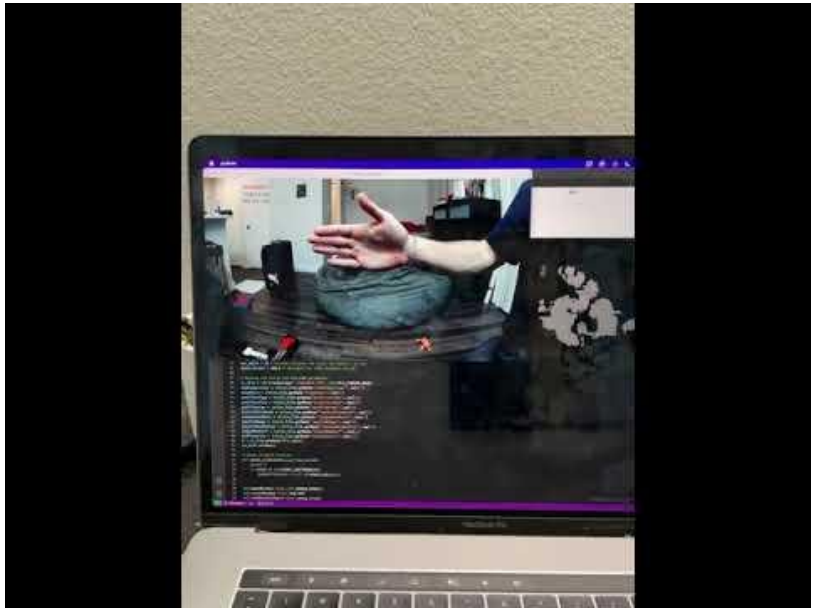


Stereo Vision

- Used to detect obstacles in way of robot. Planning on having algorithm to stop and go around.
- Best field of view for our specific application achieved by vertical placement.
- Tested for horizontal placement. Image result shown below.



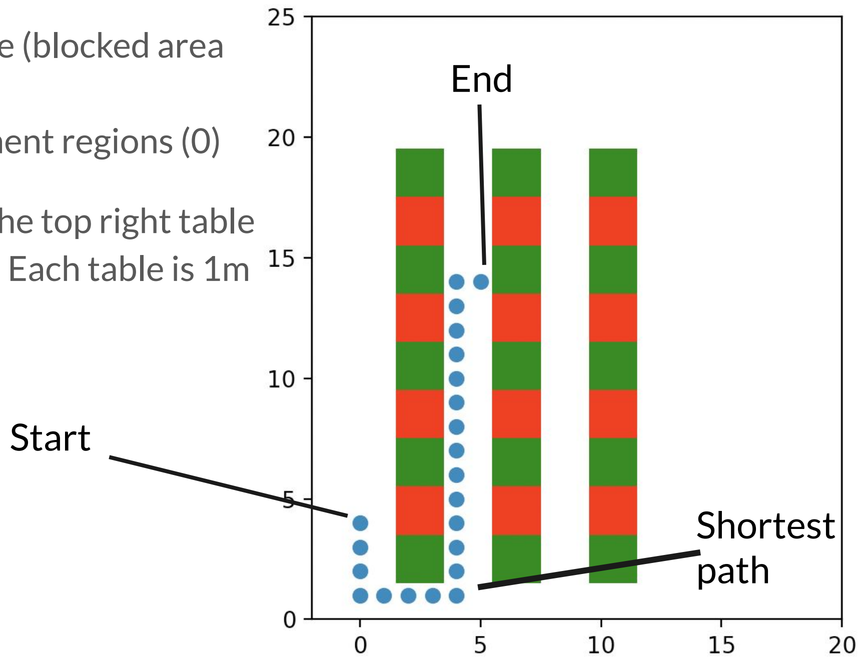
Stereovision Demo



Path planning algorithm using Binary Map

- Red: Tables (1)
- Green: In between table (blocked area for now) (2)
- White: Allowed movement regions (0)

Note: units are 0.5m. That is the top right table is at about x = 5.5m , y = 9.5m. Each table is 1m by 1m square.



Test Matrix:

```
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 2. 2. 0. 0. 2. 2. 0. 0. 2. 2. 0. 0.]
 [0. 0. 2. 2. 0. 0. 2. 2. 0. 0. 2. 2. 0. 0.]
 [0. 0. 1. 1. 0. 0. 1. 1. 0. 0. 1. 1. 0. 0.]
 [0. 0. 1. 1. 0. 0. 1. 1. 0. 0. 1. 1. 0. 0.]
 [0. 0. 2. 2. 0. 0. 2. 2. 0. 0. 2. 2. 0. 0.]
 [0. 0. 2. 2. 0. 0. 2. 2. 0. 0. 2. 2. 0. 0.]
 [0. 0. 1. 1. 0. 0. 1. 1. 0. 0. 1. 1. 0. 0.]
 [0. 0. 1. 1. 0. 0. 1. 1. 0. 0. 1. 1. 0. 0.]
 [0. 0. 2. 2. 0. 0. 2. 2. 0. 0. 2. 2. 0. 0.]
 [0. 0. 2. 2. 0. 0. 2. 2. 0. 0. 2. 2. 0. 0.]
 [0. 0. 1. 1. 0. 0. 1. 1. 0. 0. 1. 1. 0. 0.]
 [0. 0. 1. 1. 0. 0. 1. 1. 0. 0. 1. 1. 0. 0.]
 [0. 0. 2. 2. 0. 0. 2. 2. 0. 0. 2. 2. 0. 0.]
 [0. 0. 2. 2. 0. 0. 2. 2. 0. 0. 2. 2. 0. 0.]
 [0. 0. 1. 1. 0. 0. 1. 1. 0. 0. 1. 1. 0. 0.]
 [0. 0. 1. 1. 0. 0. 1. 1. 0. 0. 1. 1. 0. 0.]
 [0. 0. 2. 2. 0. 0. 2. 2. 0. 0. 2. 2. 0. 0.]
 [0. 0. 2. 2. 0. 0. 2. 2. 0. 0. 2. 2. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
```

Future Testing

- Robot localization with multiple AprilTags needed to be tested
  - Will be tested in known size room and comparing measured distance with AprilTag vs real-world distance
  - Backup: multiple cameras in the room
- Obstacle avoidance with StereoVision
  - Will be tested with variety of objects to see if robot will go around
- Text-to-speech for order taking to eliminate need for screen
  - Tentative plan to use pyttsx3 library and cheap USB speaker
- Dial in speech-to-text microphone thresholds; currently using defaults

Work Distribution

Aryan: (IoT & Component Integration)

- Programmed KitchenNode GUI to receive and send out orders
- Worked on the OrTak's speech processing and communication protocol with Kenson

Kenson: (IoT & Speech Processing)

- Programmed MQTT communication for between KitchenNode, OrTak, and WOKer
- Worked on OrTak speech recognition with Aryan
- Programmed logic for KitchenNode and integrated with GUI

Utkarsh: (Visual Processing)

- Worked on implementing stereovision and AprilTag code on the WOK-er
- Assisting Marcell in developing a path planning algorithm and collecting extensive test data

Marcell: (Robotics)

- Physical assembly and control of the WOK-er
- Developing WOKer path planning algorithm
- Assisted Utkarsh with stereovision and AprilTag optimization

Proposed Timeline

Utkarsh	Marcell	Kenson	Aryan
Week 1-3: Finish optimization of April Tag and stereovision (must work with RPi) Week 4-7: Path planning development alongside Marcell	Week 1-3: Addition of suspension system to improve robot traction. Design Ortak enclosure Week 4-7: Further path planning work.	Week 1-3: Finish integrating speaker and text-to-speech into OrTak Week 4-7: Work with Aryan to get WOK-er and KitchenNode integrated with MQTT	Week 1-3: Make the GUI more robust and add the special request section Week 4-7: Working with Kenson to get the WOK-er and KitchenNode integrated

Target Goals

For the WOK-er:

- Will be able to localize itself correctly and move to the table number it is sent without any manual assistance
- Will have sufficient traction so that it can move around without slippage and at a sufficient speed

For the OrTak:

- Will be able to accurately receive orders in a noisy environment and send orders to the KitchenNode

For the KitchenNode:

- Will be able to show all relevant order information to kitchen staff and easily manage order queue to send orders to the WOKer
- Create Firebase to store order information

Thank you for listening!

We welcome any questions!