Team F2:
Gesture Based Home Robot

Seungmin Ha, Rama Mannava, Jerry Yu

Current Home Assistant

Smart Homes	Home Robots
• Google Home, Amazon Alexa, Apple Homepod • Input limited by voice • Output limited by voice and turning on/off devices • No physical component	• Anki, Zenbo - limited to entertainment and education • Roomba, Alfawise, Landroid - home cleaning • Budgee - carries goods around by following the user, cannot be commanded

Use Cases

- Transporting goods around the house
  - to the user or anywhere

- How can we go beyond current solutions and…
  - make robotic control feel more intuitive
  - integrate robots seamlessly into our homes
  - naturally tell robots where to go

- Proposal: A robotic assistant that responds to gesture-based commands to easily navigate your home with a payload.
- ECE Areas: Software, Signals, Robotics

Assistant tasks

1. Tele-operated robot control via gestures (MVP)
2. Save specific locations and drive to them on command (MVP)
3. Drive to the user to deliver goods (snacks, phone charger)
4. Drive to a location the user points to

Requirements (Rationale)

01	**Gesture Identification**	• Intuitive control of robot • Easier to just point at something
02	**Mapping and Robot Localization**	• Tasks require position of robot and human in room
03	**Room Control**	• Tasks require 2D path planning and robot navigation of the home environment
04	**Communication**	• Robot, camera system, and AWS servers relay information about tasks and room state
05	**Overall Responsiveness**	• Robot should be fast to respond to commands

Requirements (Implementation)

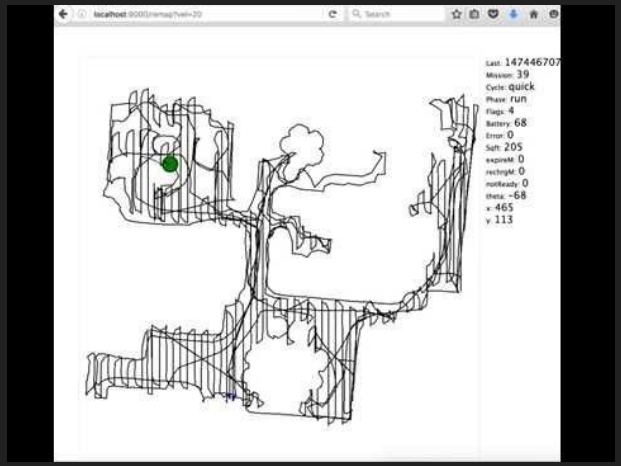
01	**Gesture Identification**	• Overhead room camera • Running CMU OpenPose on NVIDIA Jetson Xavier • >90% average precision on gesture detection (OpenPose)
02	**Mapping and Robot Localization**	• Init phase to sync 2D roomba map with 3D camera view. • Realtime readjustment with encoders and camera (OpenCV). • <1 ft drift for both robot and user location.
03	**Room Control**	• Using motor control via Roomba Create 2 Open Interface • Average cruising speed of at least 200 mm/s • Able to operate without charging for at least 1.5 hr • Maximum carrying capacity of 15 lbs
04	**Communication**	• Arduino to communicate with Roomba (at 115200 Baud) • RPC between camera <> AWS server and AWS server <> Arduino on robot ~500 ms
05	**Overall Responsiveness**	• Robot should start to perform the task <1.9s

Challenges

- Accurate mapping of the environment
- Discrepancy between the estimated current location and the actual location
- Syncing 2D map view and 3D camera view
- Computational cost of gesture analysis, running ML and CV system on embedded board (Xavier)
- Latency between camera, server, and robot communication
- Utilizing Roomba open APIs, can’t map and manually drive at the same time

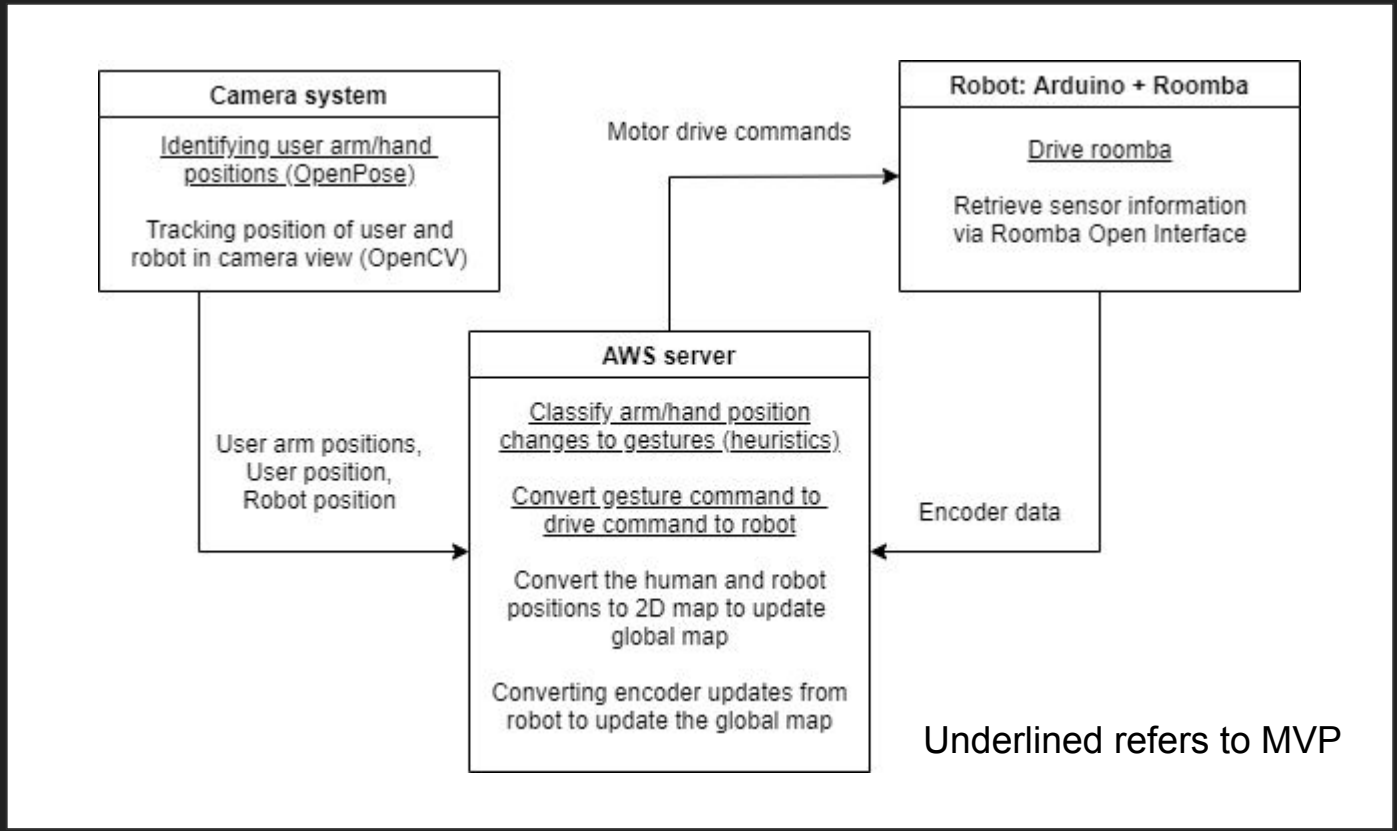
Solution Approach: Initialization Mapping

1. Run Roomba cleaning via Dorita980 SDK to obtain 2D room map
2. Track robot via camera to map camera view pixels (3D view) to 2D map



Ex: Roomba at position (56, -54) maps to pixel block (820, 210)

Solution Approach: Runtime



Testing, Verification, and Metrics

- F1 score of gesture recognition system, >90% F1 (OpenPose)
- Latency of robot response
  - OpenPose + Gesture recognition + server RPC latency + arduino to roomba < 1.9s
- Distance from estimated position to actual position < 1 ft, still can accomplish task with 1 ft drift
- Trip completion rate > 90%
- Measure power consumed by Arduino and Roomba
  - At least 1.5 hr without charging

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Date	1/13	1/20	1/27	2/3	2/10	2/17	2/24	3/2	3/9	3/16	3/23	3/30	4/6	4/13	4/20
Team Research															
Ideation															
Robot / Roomba Research															
Gesture / Openpose Research															
Mapping / Localization Research															
MVP Building															
Set up OpenPose with webcam															
Classify basic gestures (point, arm wave)															
Connect to Roomba using SDK															
Implement Tele-op driving of Roomba															
Implement setting waypoints (home)															
Set up OpenPose with Xavier and AWS															
Slack															
Mapping and Localization															
2D roomba view to 3D camera view sync															
Roomba live tracking with camera															
User live tracking with camera															
Slack															
Integration															
Implement robot driving to user															
Find direction from user point															
Implement robot driving to point															
Slack															
Course Logistics															
Abstract															
Proposal															
Design document															
Final testing															
Public demo and report															

Division of Labor

- Sean - Robot
  - Control, communication with Roomba
  - 2D path planning
- Rama - Gesture Identification
  - Running OpenPose on Xavier board
  - Classifying raw body data into gestures
- Jerry - Mapping and Localization
  - Position of human and robot
  - Live map updates based on camera and encoders
- Everyone - System integration, end-to-end testing