Team F2
Cookiebot - A Gesture Based Home Robot

Seungmin Ha, Rama Mannava, Jerry Yu

Application Areas

- Home robot with intuitive control
- Transportation of goods around the home

- Cookiebot tasks:
  - Tele-operated robot control via gestures
  - Drive home to dock on command
  - Drive to the user to deliver goods (snacks, phone charger)
  - Drive to a location the user points to

Requirements and Scope

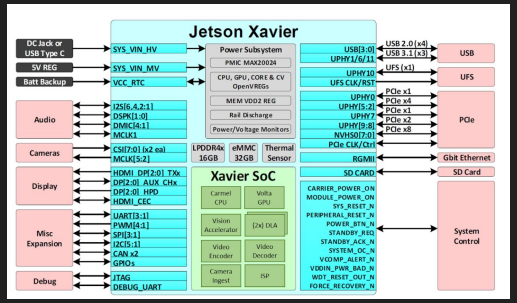
- Requirements
  - Accomplish all tasks with 90% end to end success rate
  - < 1 feet average drift between actual robot location and mapping location
  - Tasks should start to be performed < 1.9s on average

- Scope
  - One person - Limit complexity given time constraints
  - 7 gestures - 3 for tele-op, 1 gesture for stop, 1 gesture for go home, 1 gesture for going to user, 1 gesture for going to pointed location
  - Will not avoid obstacles outside of those mapped during initialization

Solution Approach

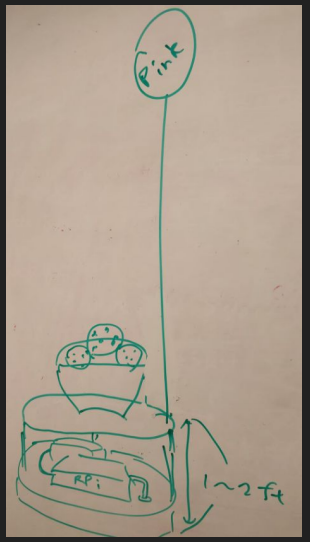
Overhead Cameras

- Mounted on top of room
- Front, back, side, overhead
- Capture user and robot
- Multiple to reduce blind spots



NVIDIA Xavier board
(8 CPU, 500 CUDA core GPU)

- OpenPose to process gestures
- Track position of user
- Track position of robot
- Web server for communication

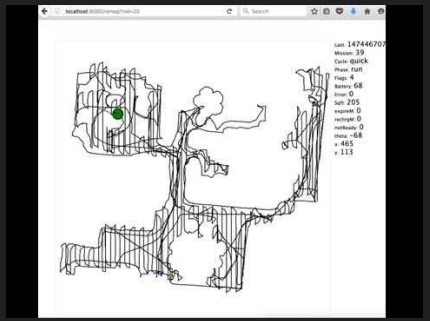


RPi mounted to Roomba

- Run commands
- Get encoder and bumper sensor data
- Carries user payload with basket

Solution Approach: Mapping

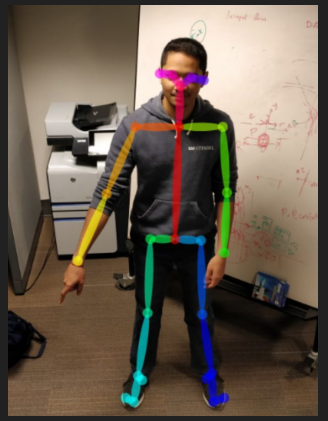
- Initialization
    1. Explore room while reading encoder data to obtain 2D room map
    2. Track robot via camera to map camera view pixels (3D view) to 2D map
- Runtime
    1. Use 3D to 2D mapping to localize user and robot
    2. Use 2D mapping for navigation to user, navigation to point
    3. Robot encoder updates are used to update the map

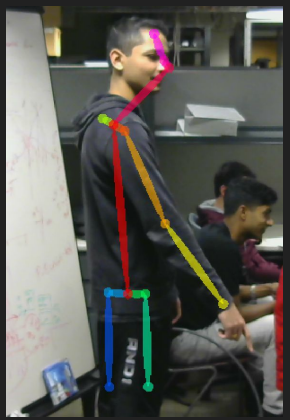


Ex: Roomba at position (56, -54) maps to pixel block (820, 210)

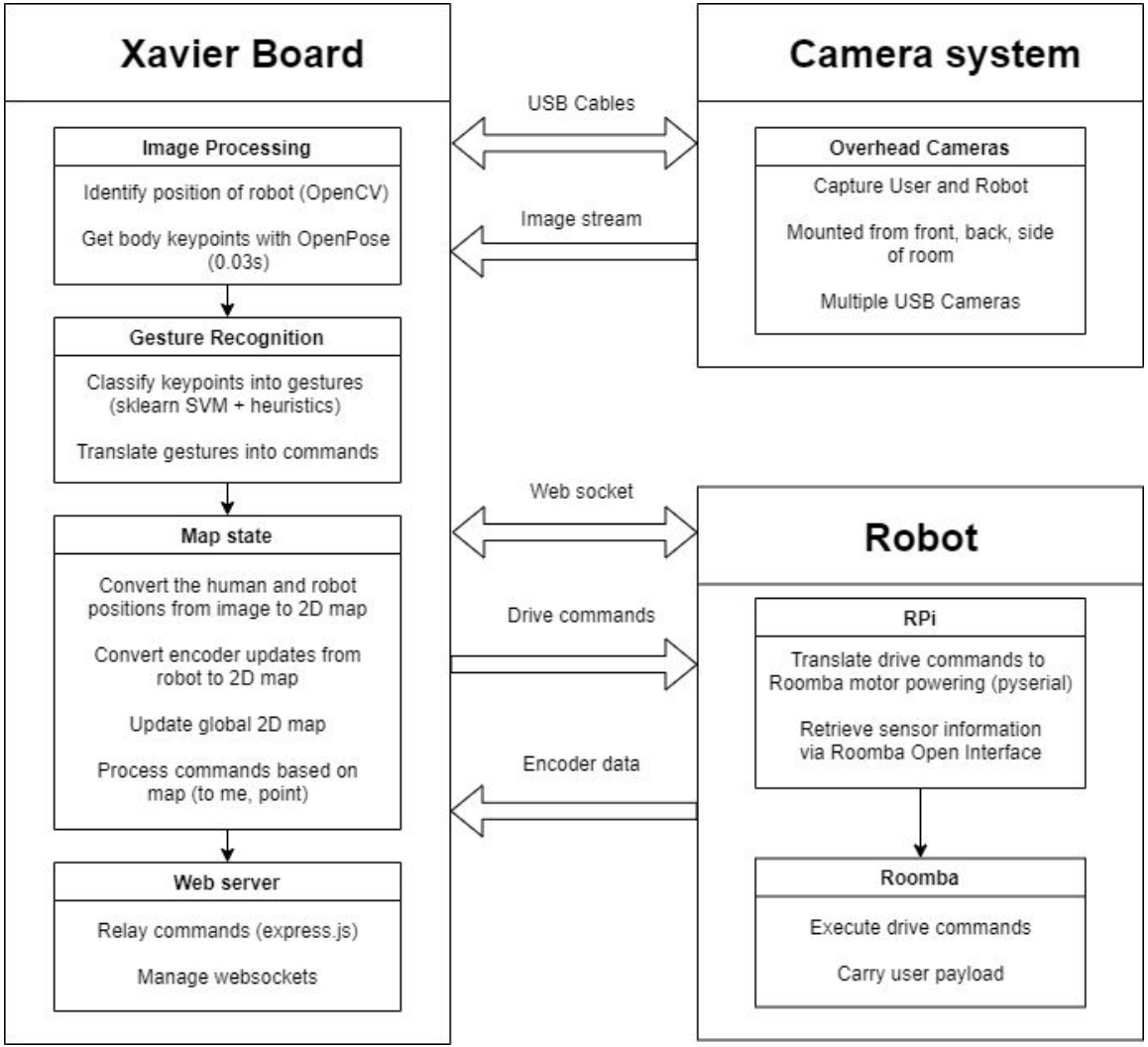
Solution Approach: Drive to Point

- Finding the point on the map the user points to:
  - Method 1: Using multiple images from different perspectives to draw a line from the user's arm to the ground.
    - Using trig with angles from core to arm, arm to shoulder, and feet position to determine ground.
  - Method 2: Using neural network classifier to predict the position in the room using keypoints as input
    - Collect training data of keypoints and proper bin
    - Treat every 1ft x 1ft square in the room as a bin in a grid
    - Regression model to determine x and y coordinate in the grid
  - Further testing required to determine method (leaning towards method 2)





System Architecture



Implementation Plan: Decisions

Component	Also considered	Reasoning
Keypoint recognition Algorithm: OpenPose (CNN)	Alphapose, Mask R-CNN, DeepCut	OpenPose had lowest latency (0.03s/frame on Xavier).
Hardware for image processing and web server: NVIDIA Xavier Board (500 CUDA)	Run on local CPU, AWS (5k CUDA)	Lowest latency solution. Not enough compute on local CPU. Need to send images to AWS
Camera setup: Multiple USB cameras from 3 sides + overhead	Overhead 3D camera, Fisheye camera	Provide backup in case OpenPose fails. Covers blind spots. USB cameras are low cost, easy to work with output image.
Web server Express: Node JS with websockets	Python Flask with endpoints	Node is asynchronous, handles requests concurrently. Sockets allow for low latency.
Robot: Roomba 671 with RPi	Building own robot	Roomba provides a drive base with bumpers, motor encoders. RPi provides wifi, python for serial communication

Implementation Plan: Buy, Use, Write

- Image processing
  - Buy USB Cameras and cables to attach to board
  - Use CMU OpenPose to detect keypoints and user in image
  - Write robot recognition algorithm with OpenCV
- Gesture recognition
  - Write keypoint to gesture classifier with sklearn and collected data.
- Mapping
  - Write Roomba exploring algorithm for 2D map
  - Write 2D to 3D mapping and map updating methods
- Server
  - Write web server using Node.js and Express.
- Robot
  - Buy Roomba, Assemble basket and RPi on top
  - Use Roomba drive base and Roomba Open Interface to send commands
  - Write serial communication from RPi using pyserial
  - Write method to translate gesture commands into Roomba commands

Metrics and Validation (Software)

Component	Metric	Method
Gesture Recognition (classification of keypoints to gestures)	> 80% per frame, Acceptable for high FPS, multiple cameras	Perform gestures in all parts of room. Compare identified gestures with known gestures
Mapping and Localization	< 1ft on average, Reasonable rift	Compare camera prediction of location on 2D map with actual position across 10 movement gestures
User Gesture - RPi Transmission	< 1.9s on average, Google Home	Measure the time from capturing a frame to sending the appropriate command to the Roomba via the RPi for 10 gestures

Risk: What if the system fails to identify a gesture?
→ Use other cameras and subsequent frames to aid identification

Metrics and Validation (Hardware)

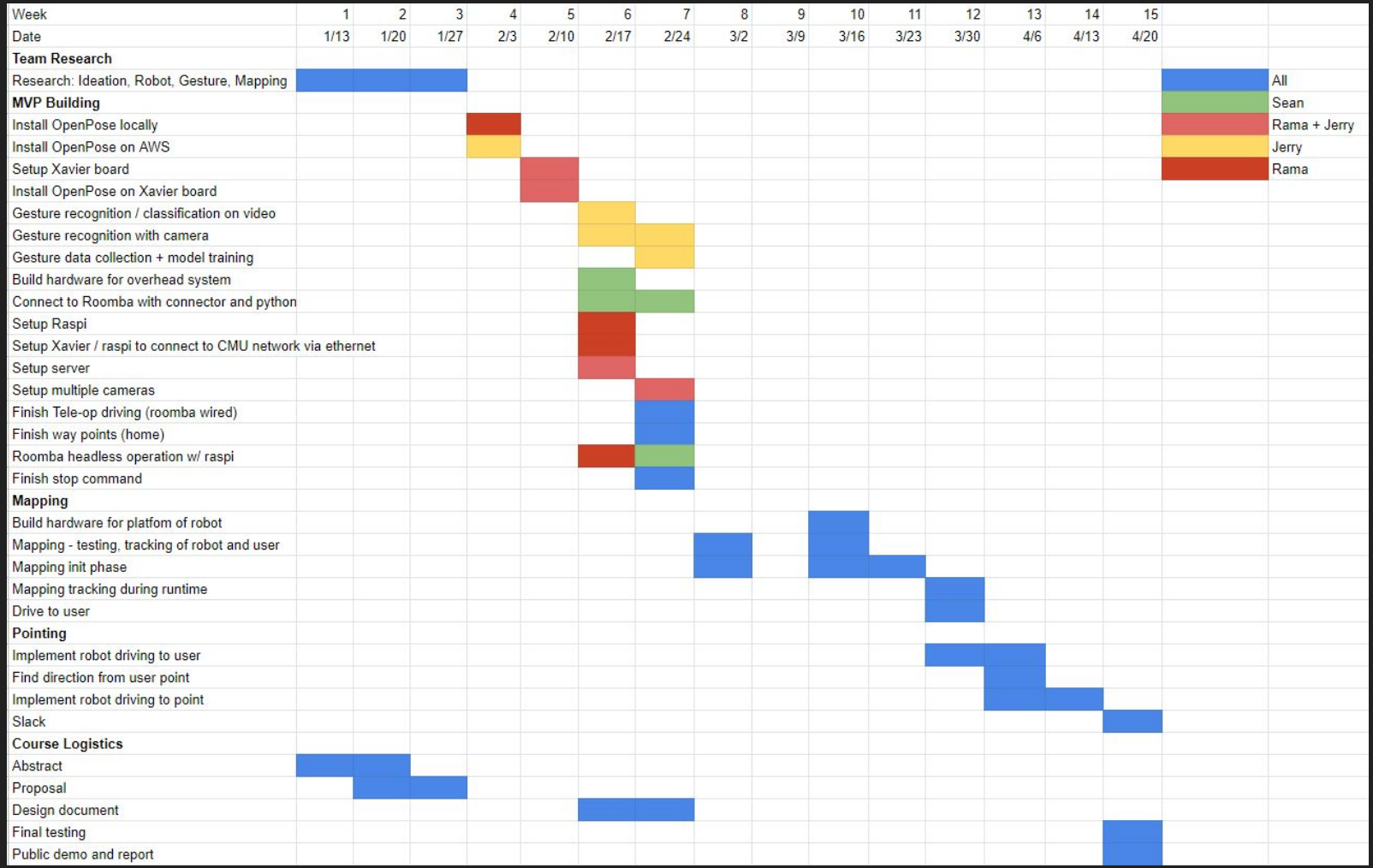
Component	Metric	Method
Server - RPi transmission	< 100ms on average	Measure time it takes to send 5 sensor data (RPi -> Server) and receive 1 path information (Server -> RPi)
Robot movement accuracy	< 1 ft on average, reasonable drift	Measure distance between the goal position and the actual position after 5 runs of movement
Robot movement speed	< 50s on average, safety and efficiency	Measure the time between gesture input and the movement completion (~10m) for 10 gestures

Risk: What if the robot's estimation of the current position is off by a lot?
→ Use camera as a means of backup localization

Project Management

Each member: ~20 hr / week

Constantly updated, break down tasks after each phase and testing



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															
Research: Ideation, Robot, Gesture, Mapping															
MVP Building															
Install OpenPose locally															
Install OpenPose on AWS															
Setup Xavier board															
Install OpenPose on Xavier board															
Gesture recognition / classification on video															
Gesture recognition with camera															
Gesture data collection + model training															
Build hardware for overhead system															
Connect to Roomba with connector and python															
Setup Raspi															
Setup Xavier / raspi to connect to CMU network via ethernet															
Setup server															
Setup multiple cameras															
Finish Tele-op driving (roomba wired)															
Finish way points (home)															
Roomba headless operation w/ raspi															
Finish stop command															
Mapping															
Build hardware for platfom of robot															
Mapping - testing, tracking of robot and user															
Mapping init phase															
Mapping tracking during runtime															
Drive to user															
Pointing															
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															