

Team F2

Cookiebot - A Gesture Based Home Robot

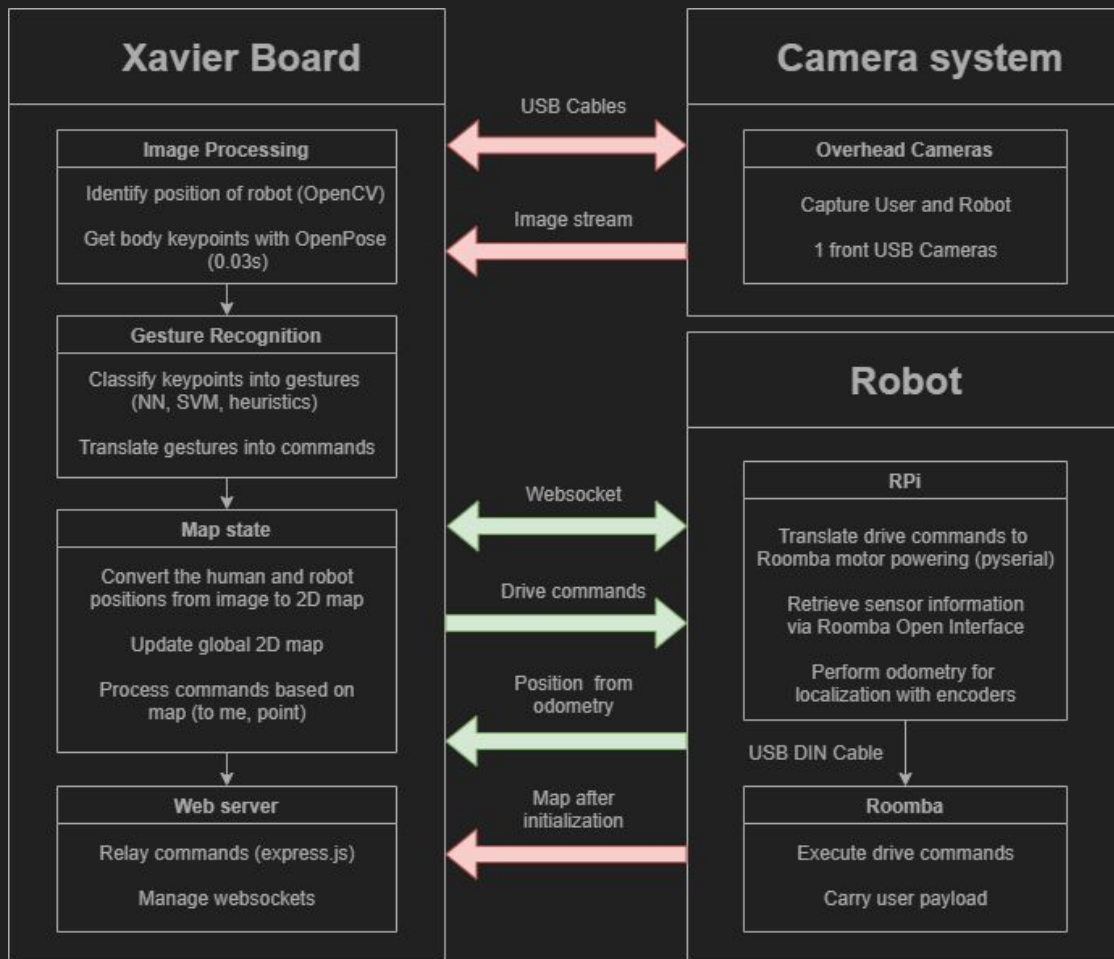
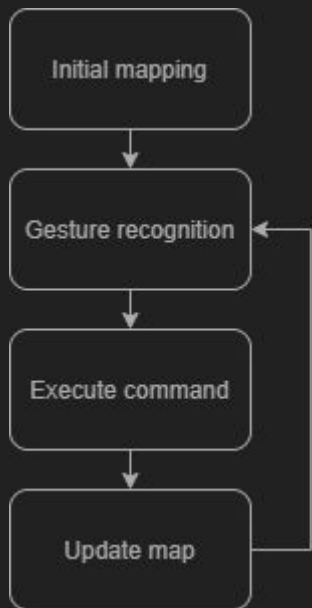
Seungmin Ha, Rama Mannava, Jerry Yu



Application Areas

- Home robot with intuitive gesture control
- Our target tasks:
 - Tele-operated robot control via gestures
 - Drive home to dock on command
 - Drive to a location the user points to
 - Drive to the user to deliver goods (cookies)
- Scope
 - One person - Limit complexity given time constraints
 - 8 gestures - 3 for tele-op, 1 gesture for stop, 1 gesture for go home, 1 gesture for going to user, 1 gesture for point ready, 1 gesture for going to pointed location
 - Will not avoid obstacles outside of those mapped during initialization

Block Diagram



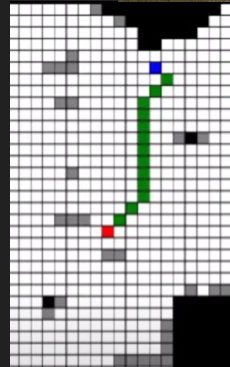
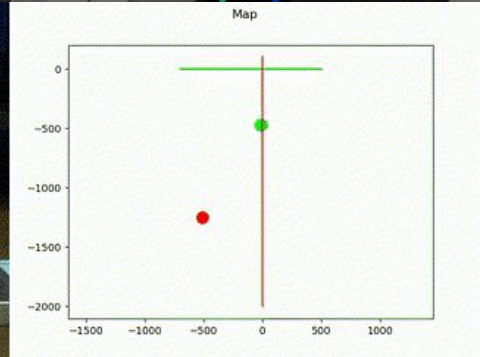
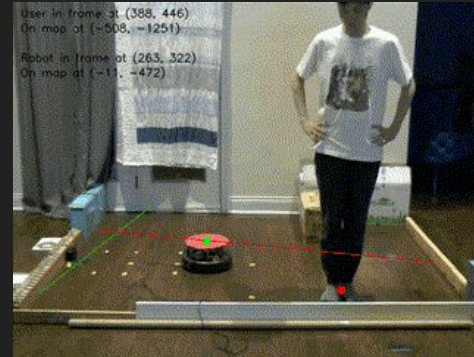
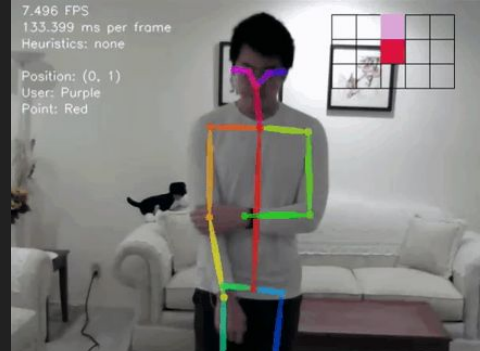
System Approach

- CMU OpenPose (CNN) for pose keypoints
- Embedded GPU: NVIDIA Xavier Board
 - Run OpenPose at 27 FPS
 - Train and run models on GPU
 - Run expensive OpenCV operations on CPU
- Roomba
 - Communicate via pyserial and Roomba Open Interface
 - Access motors and sensors
- Custom data scripts
 - Collect and label data
 - Run system on streaming video for remote dev
 - Calculate metrics
- Visualizations



Complete Solution

- Gesture Recognition
 - Classify user gestures from OpenPose keypoints with heuristics, SVMs (teleop), NNs (point)
- 3D to 2D Mapping
 - Track where user and robot are with OpenPose and OpenCV
- Robot
 - Explore and map room
 - RasPi handles I/O and odometry
 - Drive to location with A*
- Demo
 - Visualizations
 - All 4 tasks in action



Metrics and Validation: Gesture Recognition

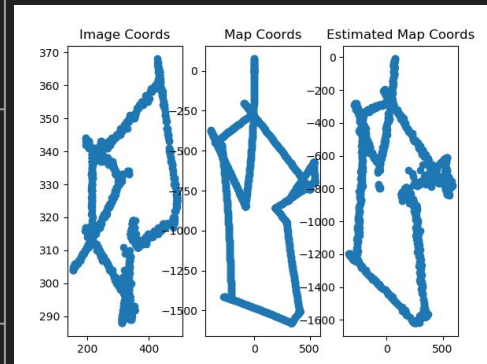
Tests assume user is in frame and face the camera

Tests are under different lighting conditions, are invariant based on position in room

Component	Metric	Method	Result
Teleop Classification (SVM)	> 90% accuracy (reasonable to use)	Tested on 76 gestures	Test accuracy: 90.8%
Point Classification (NN)	> 90% accuracy	Tested on 130 gestures	Test accuracy: 83.08%
Other Classification (Heuristics)	> 90% accuracy	Tested on 162 gestures	Test accuracy: 95.67%
All Gestures (Teleop, Point, Other)	> 90% accuracy < 10% misclassification (FP) < 5% unrecognized (FN)	Tested on 368 gestures	Accuracy: 90.2% Misclassification: 8.8% Unrecognized: 0.9%

Metrics and Validation: 3D to 2D Mapping

Component	Metric	Method	Result
User Position Accuracy	< 0.3 m drift on average (reasonable reaching distance)	Measure distance between the estimated position and the actual position over 3072 frames	0.04 m
Robot Position Accuracy	< 0.3 m drift on average (reasonable reaching distance)	Measure distance between the estimated position and the actual position over 3072 frames	0.12 m
3D to 2D Mapping Generation	< 60 sec per mapping (reasonable initialization phase overhead)	Time to generate mapping from 3072 frames of position data	4.8 sec



Metrics and Validation: Robot

Component	Metric	Method	Result
Robot Movement Accuracy	< 0.3 m drift on average (reasonable reaching distance)	Measure distance between the goal position and the actual position after 5 runs of movement	0.048 m
Robot Movement Speed	0.05 m/s to 0.2 m/s on average, safety and efficiency	Measure the time between the command and the movement completion of 10 gestures (~5m of travel each)	0.11 m/s
Robot Battery Life	> 2 hrs operation without recharging	Make the robot perform drive function every ten minutes, without recharging	Passed
Room Coverage	> 90% map accuracy	Count inaccurately represented 5 cm square cells from a 2D map	95.6%

Metrics and Validation: Performance

Component	Metric	Method	Result
Gesture Recognition Time (Xavier)	< 500 ms	Time from frame capture to recognition for 129 gestures	Average time: 56 ms Longest time: 261 ms
3D-2D Mapping Processing Time (Xavier)	< 100 ms	Time to calculate map positions from frame for 3072 frames	Average time: 16 ms Longest time: 166 ms
Robot Path Planning (Xavier)	< 1000 ms	Measure time for the robot to calculate a path to the goal after 6 runs	Average time: 230 ms Longest time: 780 ms
Robot RPi - Server Transmission (Xavier and RPi)	< 100 ms	Measure for 100 sends and receives	Average time: 3.5 ms Longest time: 13.5 ms
Robot Execution Time (RPi)	< 15ms	Measure 100 cycles from receiving command to execution	Average time: 13.7 ms Longest time: 16.2 ms
Total Time	< 1.9s (Google Home)	max(gesture, mapping) + path+ transmission + execution	Average time: 0.30 sec Longest time: 1.07 sec

Design Tradeoffs

- AWS GPU vs Xavier
 - Big GPU with network bandwidth vs GPU on local network
- Data Collection Time vs Accuracy
 - Point
 - Trained and tested on 1 hr of data
 - Trade accuracy for time to collect and label data
- Speed vs Accuracy
 - 3D to 2D Mapping
 - Frame dropping increases speed and decreases accuracy; ultimately fast enough
 - Trade accuracy for speed
 - Robot
 - Limiting robot speed improves odometry accuracy and user safety; lower efficiency
 - Trade speed for accuracy

Project Management

Week	1	2	3	4	5	6	7	8
Date	1/13	1/20	1/27	2/3	2/10	2/17	2/24	3/2
Team Research	[Blue bar]							
Research: Ideation, Robot, Gesture, Mapping	[Blue bar]							
MVP Building				[Red bar]	[Red bar]	[Yellow bar]	[Yellow bar]	[Yellow bar]
Install OpenPose locally			[Red bar]					
Install OpenPose on AWS			[Yellow bar]					
Setup Xavier board				[Red bar]				
Install OpenPose on Xavier board				[Red bar]				
Gesture recognition / classification on video					[Yellow bar]			
Gesture recognition with camera					[Yellow bar]	[Yellow bar]	[Yellow bar]	
Gesture data collection + model training						[Yellow bar]	[Yellow bar]	
Build hardware for overhead system					[Green bar]			
Connect to Roomba with connector and python					[Red bar]	[Green bar]		
Setup Raspi					[Red bar]			
Setup Xavier / raspi to connect to CMU network via ethernet					[Red bar]			
Setup server					[Red bar]			
Setup multiple cameras						[Red bar]	[Red bar]	
Finish Tele-op driving (roomba wired)						[Blue bar]	[Blue bar]	
Finish way points (home)						[Blue bar]		
Roomba headless operation w/ raspi					[Red bar]	[Green bar]		
Finish stop command						[Blue bar]		

Week	10	11	12	13	14	15
Date	3/16	3/23	3/30	4/6	4/13	4/20
Setup						
Plan out remote work, experiments	[Yellow bar]	[Yellow bar]				
Waiting for components		[Yellow bar]				[Yellow bar]
Make old code work (with videos)		[Yellow bar]				
Gather new data for teleop		[Yellow bar]	[Yellow bar]			
Setup environment		[Yellow bar]				
Pointing						
Gather new data for point			[Yellow bar]	[Yellow bar]		
Find location of user point			[Yellow bar]			
Visualization						
Build visualization of gestures		[Yellow bar]		[Yellow bar]		
Build visualization of point				[Yellow bar]	[Yellow bar]	
Course Logistics						
Demo video / final report					[Yellow bar]	
Setup						
Plan out remote work, experiments	[Green bar]	[Green bar]				
Waiting for components		[Green bar]				
Mapping						
Algorithm to explore room and generate 2d map			[Green bar]	[Green bar]		[Green bar]
Mapping - testing, tracking of robot and user			[Green bar]			
Improve go to home with mapping				[Green bar]	[Green bar]	
Mapping init phase			[Green bar]			
Mapping tracking during runtime				[Green bar]	[Green bar]	
Drive to user				[Green bar]	[Green bar]	
Pointing						
Implement robot driving to point				[Green bar]		
Course Logistics						
Demo video & final report					[Green bar]	
Setup						
Wait on Xavier delivery		[Red bar]				
Set up Xavier to allow remote access and run webserver		[Red bar]				
2D to 3D Mapping						
Identify Roomba and user in camera view			[Red bar]			[Red bar]
"Real time" updates to Roomba location on 2D map			[Red bar]	[Red bar]		
Create mapping from camera view to 2D map				[Red bar]	[Red bar]	
Integrate with webserver				[Red bar]	[Red bar]	
Visualization						
Create 2D map visualization			[Red bar]			
Course Logistics						
Demo video / final report					[Red bar]	