Edge-native App for Indoor Navigation

Sarvesh Tandon, Qifei Dong (Mentor)
Carnegie Mellon University

Problem Statement

Key Issue

- Help people who are new to CS Building to navigate around

How do we do this?

- A mobile application that uses the camera and ORB-SLAM3 to tell location
- Give audio and text directions to destination
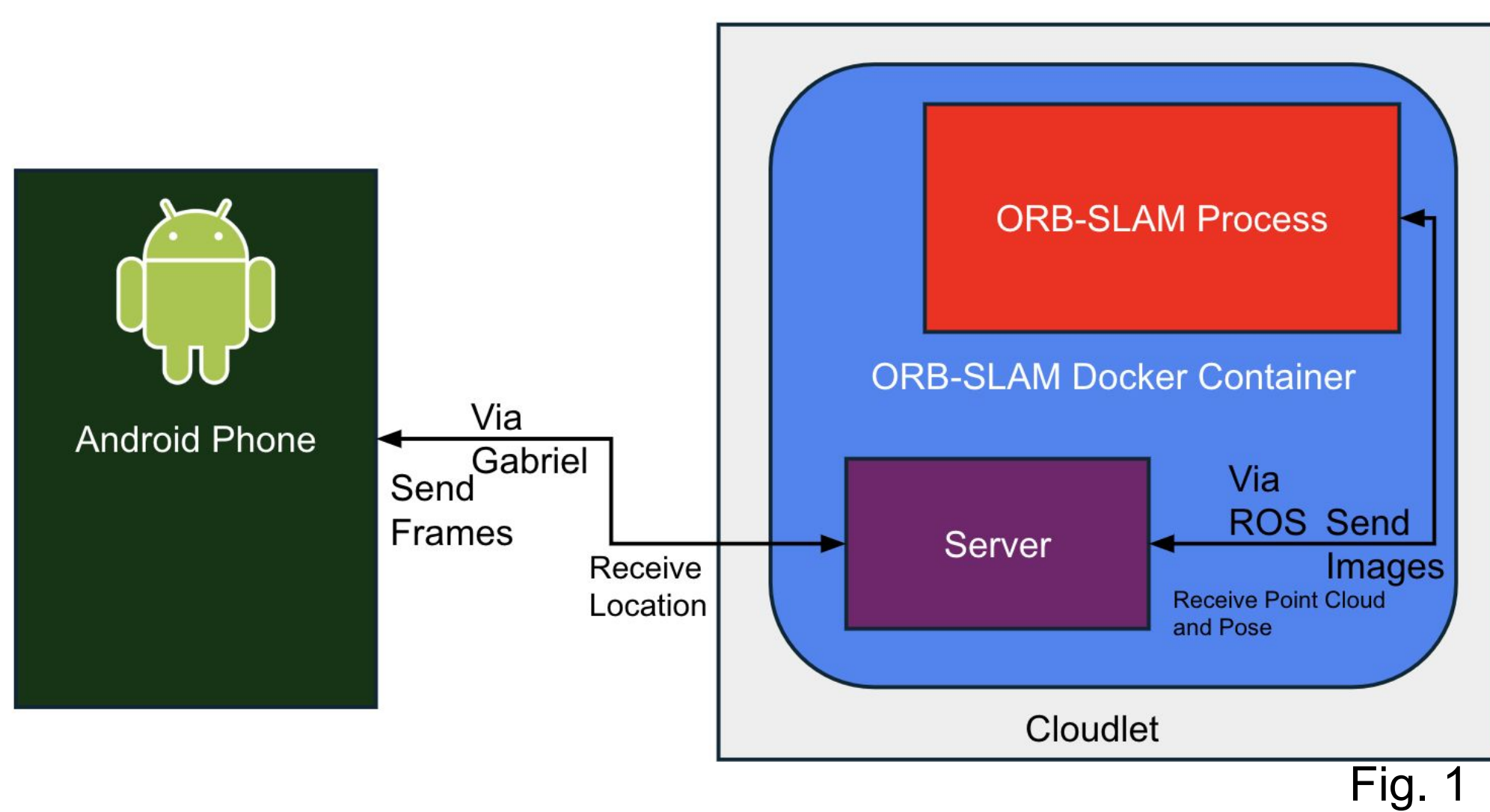- Map out 6th floor of SCS building

Design

Fig. 1

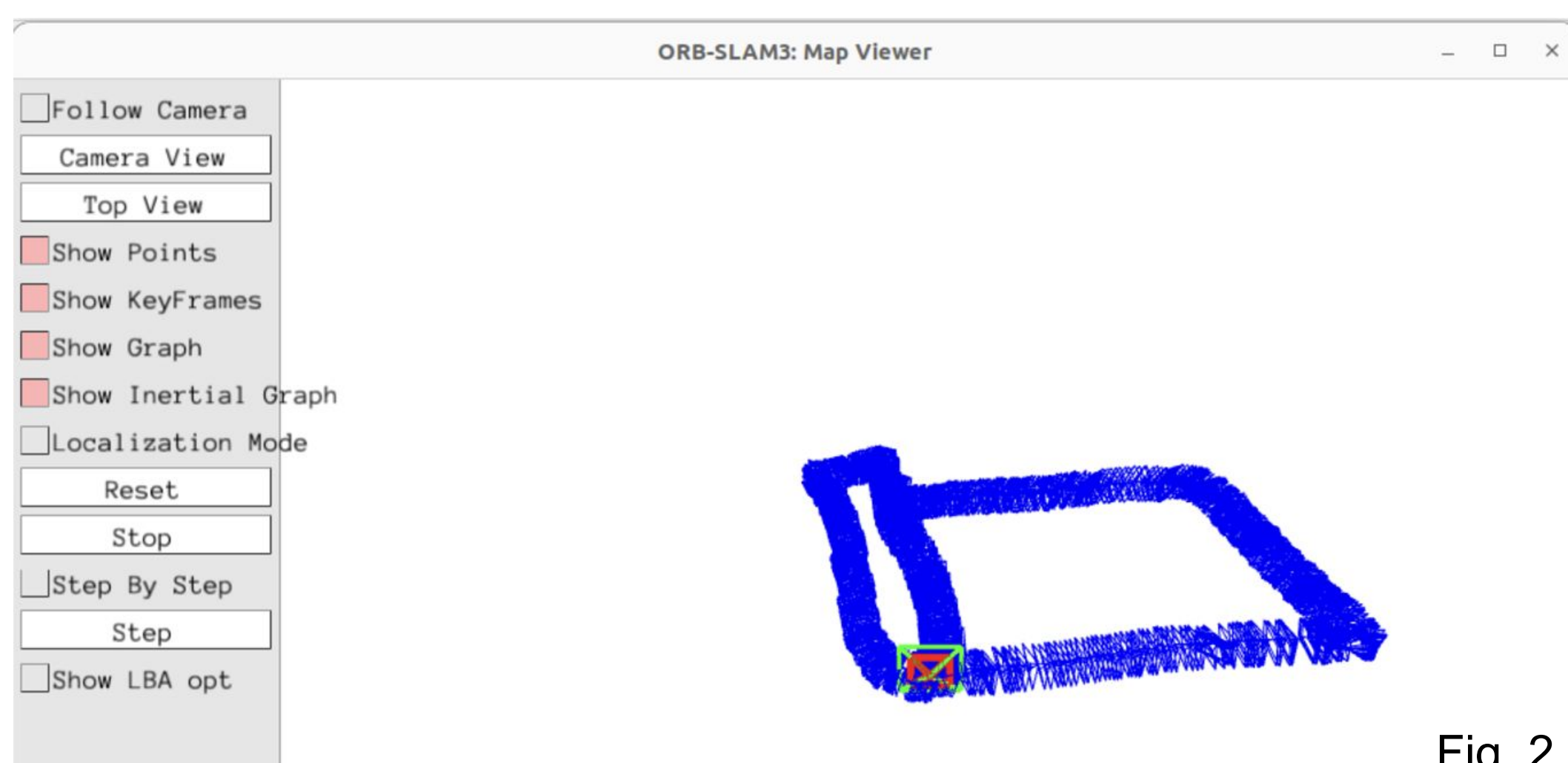## Built Map in ORB-SLAM3 for 6th floor

Fig. 2

Algorithm

- Created graph representation of the map.
- Each node represents an area zone. For example, there is an elevator zone, a bathroom zone, a kitchen zone, etc.
- Used Breadth-First-Search to find out the next zone the client should travel to.
- Use the current location and previous location to determine the direction in which the client is moving, also known as the movement vector.
- Use the vector to give directions relative to the clients location and movement direction.
- For example, if a client is coming from Elevator to room 6115, we calculate the movement vector to be south and give the direction "Go left" to reach SCS Dean. (Fig. 4)

Server and Android App Screenshots
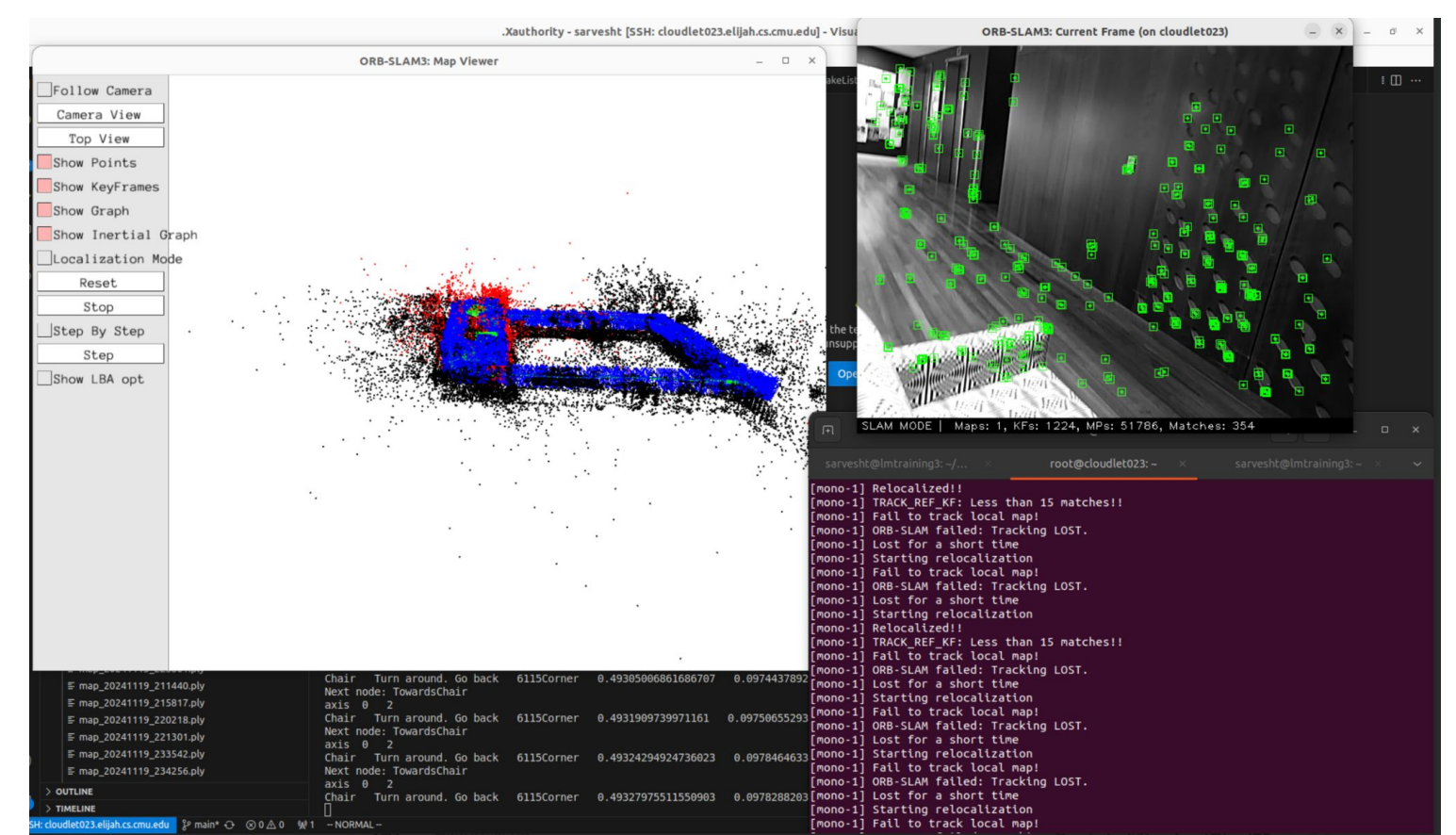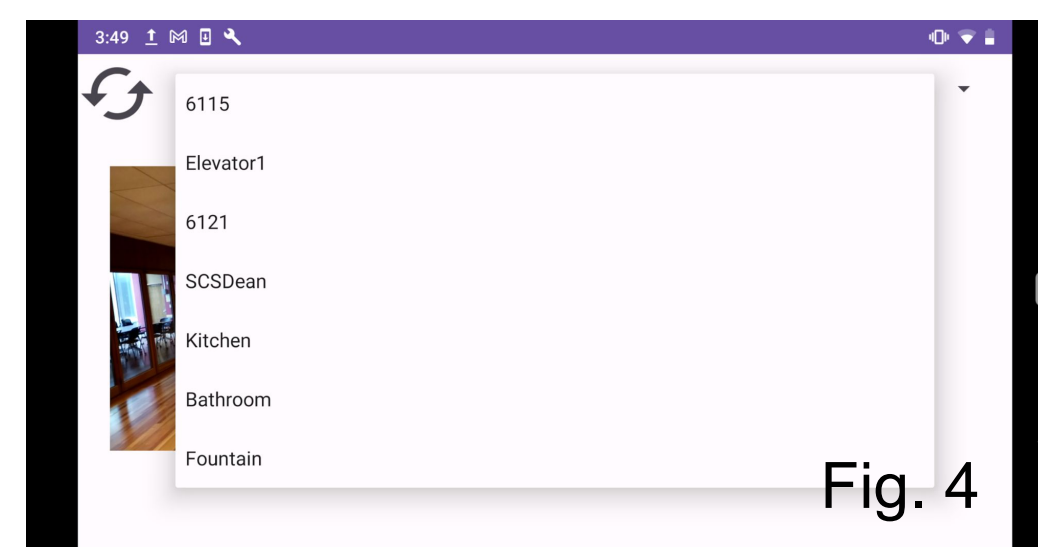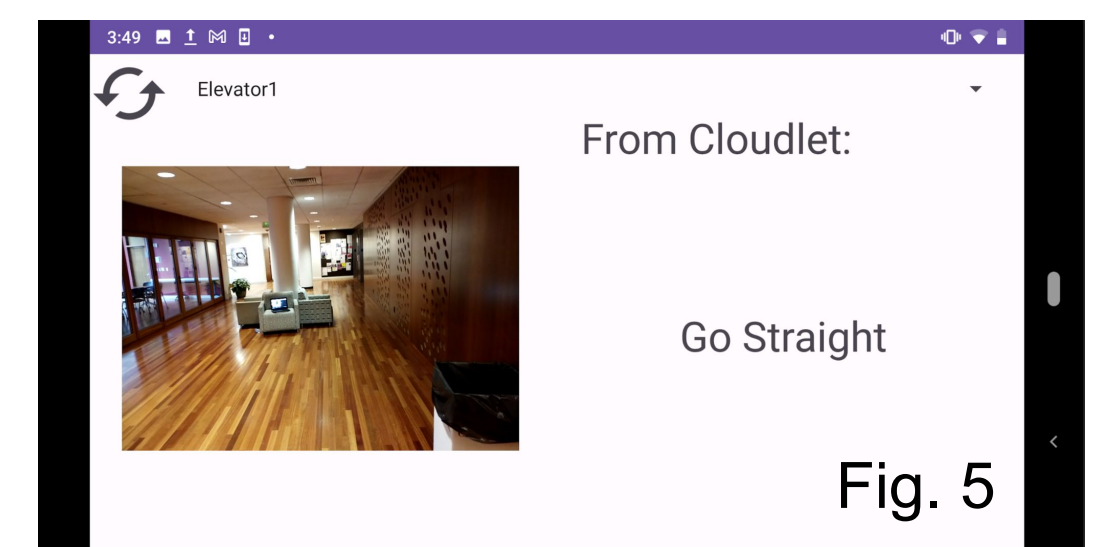

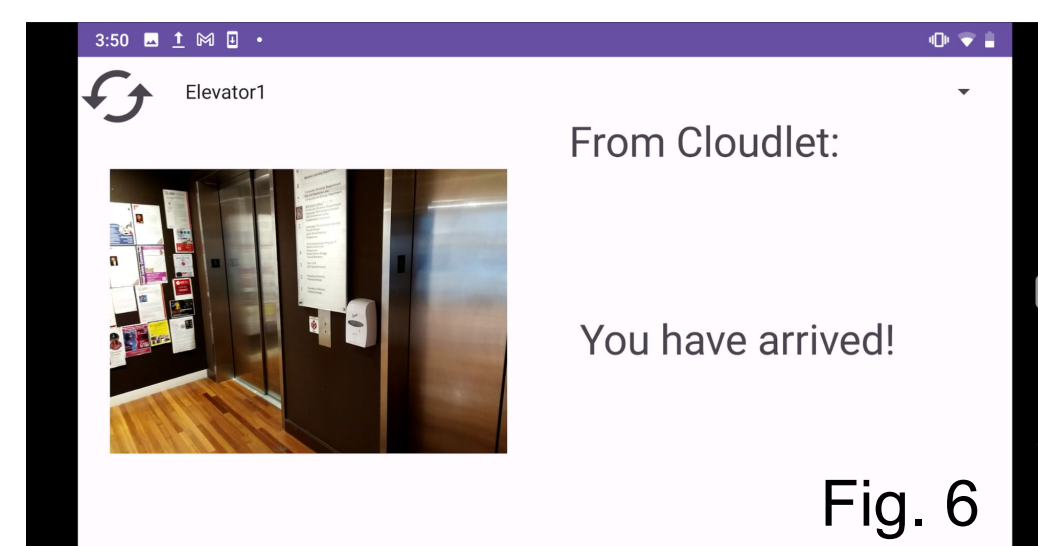ORB-SLAM3 Server View   Fig. 3

Fig. 4

Fig. 5

Fig. 6

Fig. 4 Pick a destination to navigate to
Fig. 5 The app prints the directions and speaks them out
Fig. 6 Follow the directions till you reach the destination
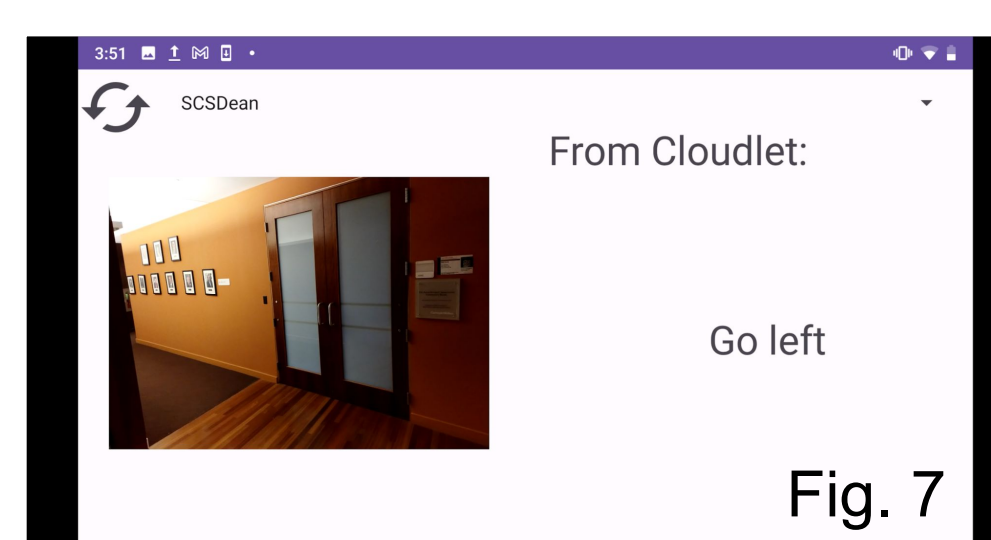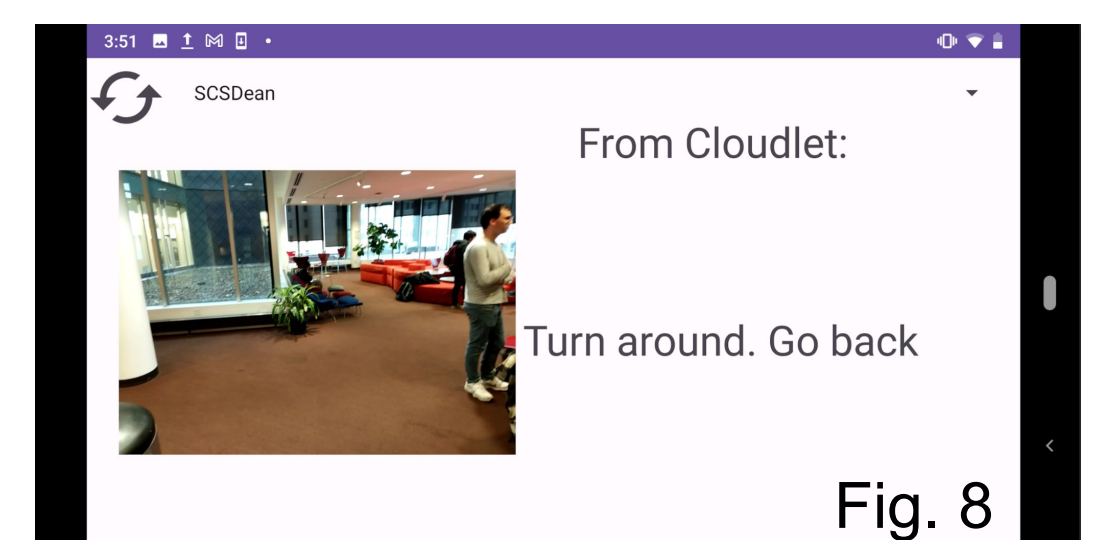
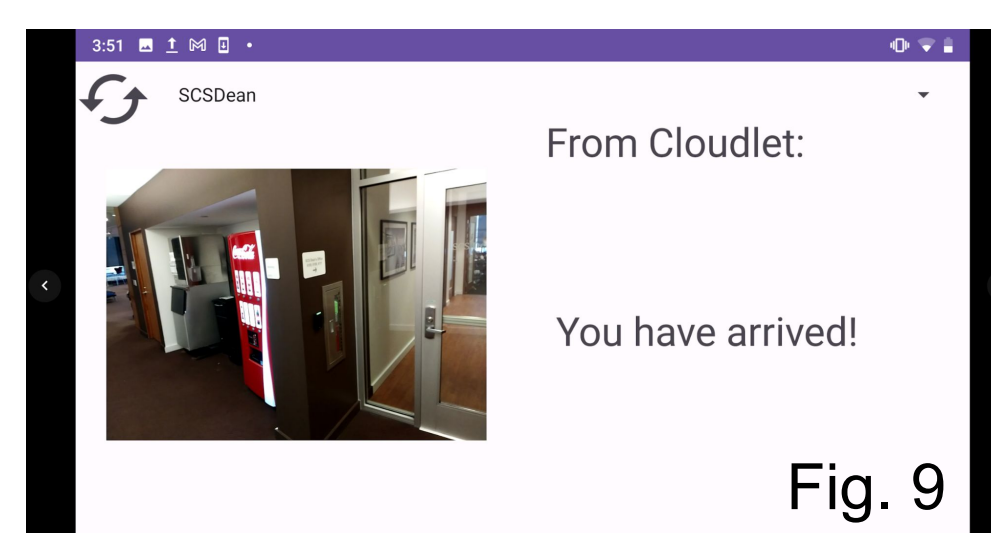## Self Correcting Directions

Fig. 7

Fig. 8

Fig. 9

Fig. 7 Change destination to a new location: SCSDean
Fig. 8 Client goes too far ahead and reach Lobby. Server updates path and gives new directions
Fig. 9 Follow the new directions till the you reach the destination

Leasons Learned

- Learned how to run a Gabriel Compute engine while also publishing and subscribing to ROS2 topics.
- ORB-SLAM3 requires a lot of CPU power to build maps and to accurately identify location. It works much better on cloudlets with faster CPUs.
- ORB-SLAM3 is very susceptible to drops in frame rates which can occur due to 'WiFi dead zones'.

Future Directions

- Modify real time direction algorithm to give directions based on where the camera is looking instead of being based on the direction of motion