Project AutoMapper

Team BC - Aditya Ranade, Akshat Jain, Wenxin Xiao

Use Case

- First Responder Situation
- Two requirements for successful and efficient evacuation:
    - 2D floor plan of unknown environment
    - Position of trapped people
- **Deliverable** - Autonomous Robot that is able to navigate an unknown environment, create a 2-D floor plan, and mark the location of people present in that environment
- Software, Hardware, and Signals

Map taken from
http://www.ctralie.com/Teaching/DukeDusty2/

Requirements - Mapping and Localization

- Ability to create a 2-D Map (Floor Plan) of unknown environment and localize the robot within it.
- **Challenges:**
    - Uncertainty in Sensor and Actuator Data
    - Chicken and Egg Problem
- **Hardware Requirements:**
    - Wheeled robot platform with a mount for a LIDAR.
    - LIDAR sensor.
    - IMU Sensor to capture current pose.
    - Ability to rotate LIDAR sensor 360 degrees.

Requirements - Mapping and Localization cont'd

- **Software/Embedded Requirements:**
  - Ability to communicate data between laptop and sensors on robot over WiFi.
  - Ability to send control information to embedded system over WiFi..
  - Ability to capture point cloud data and process it to create a 2D Map.
  - Ability to estimate pose of robot from 2D map and vice versa.
- **Performance Metrics:**
  - Divergence between ground-truth and generated map.
 - Divergence between actual pose and estimated pose.

Requirements - Autonomous Navigation

- Ability to navigate to all unknown regions of the environment to create a complete map.
- **Challenges**
  - Obstacle Avoidance
  - Type of environment - Nearly impossible if we cannot find closed contour of the 2D Map.
- **Software Functionality**
  - Path Planning algorithm to venture into previously unexplored regions of the map.
- **Qualitative/Quantitative metrics**
 - Generate a map covering at least 90% of entire surface area.

Requirements - Human Detection

- Ability to detect and mark location of humans on the generated 2-D map.
- **Challenges**
    - Detect living objects from stationary objects; detect humans from other animals based on size and temperature.
    - Streaming recorded video to PC uses up lot of bandwidth and processing power. Some initial computation will need to be done on edge.
- **Hardware Requirements:**
    - Thermal Camera mounted on robot.

Solution Approach
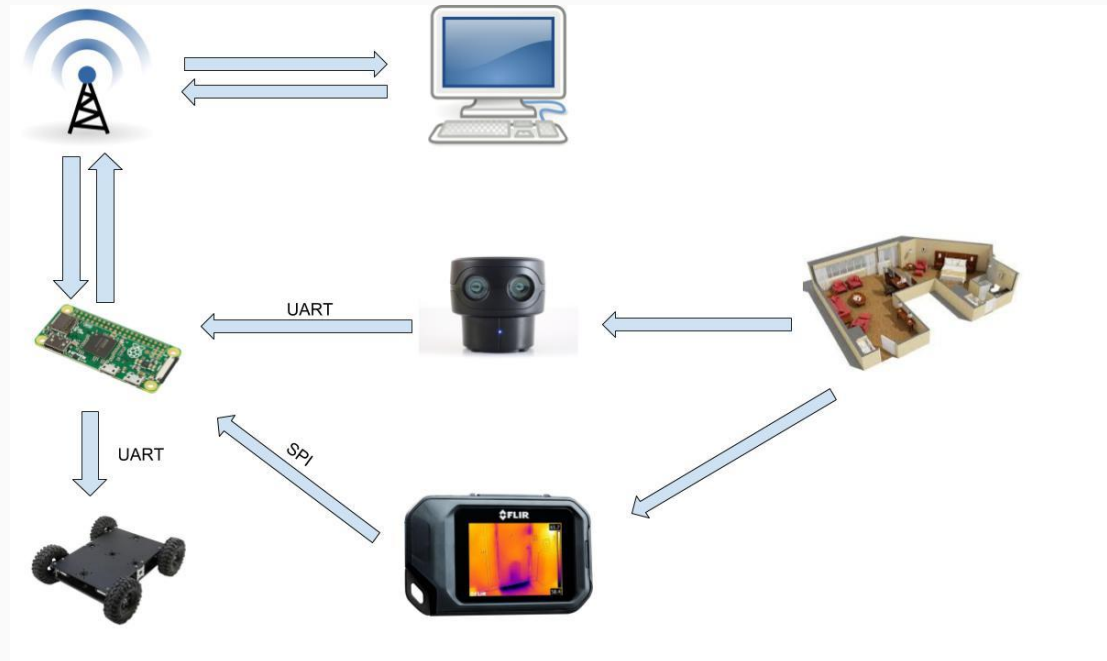
View of Subsystems
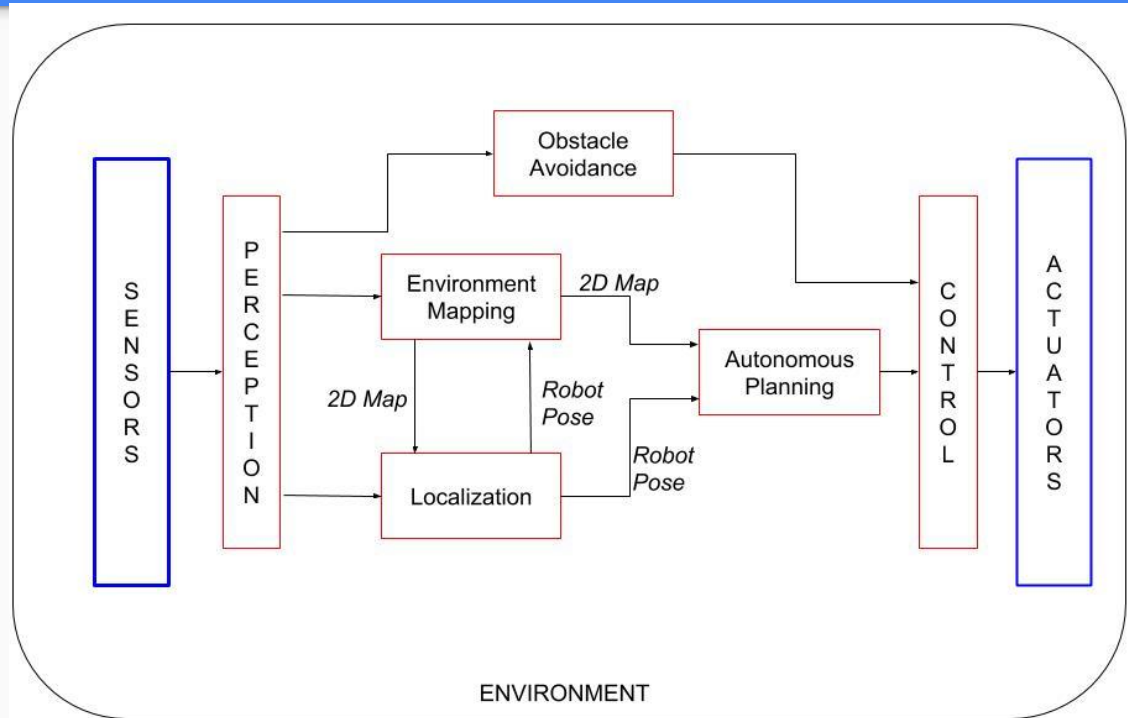
Solution Approach cont'd

- We will run the localization, mapping, obstacle detection, and autonomous navigation nodes on ROS running on a laptop.
- SLAM for localization and mapping nodes.
- Lidar Sensor - Point cloud data
- Thermal Camera - Detecting humans
- Communicate with laptop through the Pi on the robot.
- Third Party Libraries:
    - Move Base: Navigation stack of ROS

Testing, Verification and Metrics

- **Metrics**
    - Fidelity of Map (Divergence of generated map from ground truth)
    - Actual pose vs estimated pose over the entire path of robot
    - Coverage of Map Surface Area
    - Accuracy of location of humans
- **Simulation**
    - USARSim to simulate real-world sensor and actuator data. Has third party libraries to communicate with robot. Allows for virtual testing and verification of algorithms.
- **Real World Testing**
    - Need to test in real-world scenarios and make changes to algorithm.

Tasks and Division of Labor

S.No	Tasks	Duration	Team Members
1	Ordering all hardware equipment.	2 weeks	Wenxin, Aditya, Akshat
2	Tutorials on RobotOS, USARSim, openCV	1 week	Wenxin, Aditya, Akshat
3	Connecting all components and setting up communication interfaces.	1 week	Wenxin, Aditya, Akshat
4	Design Methodology	3 days	Wenxin, Aditya, Akshat
5	Manual SLAM Implementation - Mapping	1 week	Aditya
6	Manual SLAM Implementation - Localization	1 week	Akshat, Aditya
7	Simulated testing on USARSim, Bug fixing	1 week	Aditya
8	Autonomous Navigation in unknown regions	2 weeks	Wenxin
9	Simulated testing on USARSim, Bug Fixing	1 week	Wenxin
10	Combining autonomous navigation with SLAM	3-4 days	Wenxin, Aditya, Akshat
11	Obstacle avoidance	3-4 days	Akshat
12	Simulated testing/ Bug fixing	1 week	Wenxin, Aditya
13	Human Detection and testing	1 week	Akshat
14	Overlay Humans on map in right positions.	1 week	Akshat
15	Real world Testing and Bug Fixing	1 week	Wenxin, Aditya, Akshat

18-500 Capstone

Task	Hours	%
18-500 Capstone	0h	0%
Ordering All Hardware Equipment	0h	0%
Wheeled Robot	0	0%
Lidar Sensor	0	0%
Rasberry Pi	0	0%
Thermal Camera	0	0%
Connecting All Components	0h	0%
Set up communication interfaces	0	0%
Tutorials on RobotOS, USARSim, ope...	0	0%
Project Proposal	0h	0%
Proposal Presentation	0	0%
Design Methodology	0h	0%
Design Representation	0	0%
Manual SLAM Implementation	0h	0%
Mapping	0	0%
Localization	0	0%
Simulated Testing on USARSim, Bug ...	0	0%
Autonomous Navigation	0h	0%
Motion Planning	0	0%
Point Cloud Data	0	0%
Simulated Testing	0	0%
Combining SLAM With Navigation	0h	0%
Combining SLAM with Navigation	0	0%
Obstacle Avoidance	0	0%
Simulated Testing	0	0%
Human Detection	0h	0%
Human Detection	0	0%
Overlay Humans on Map	0	0%
Final	0h	0%
Real World Testing	0	0%
Slack	0	0%
Demo Preparation	0	0%

Assignments (Gantt bars):
- Wheeled Robot — Wenxin , aditya , akshat
- Lidar Sensor — Wenxin , aditya , akshat
- Rasberry Pi — Wenxin , aditya , akshat
- Thermal Camera — Wenxin , aditya , akshat
- Set up communication interfaces — Wenxin , aditya , akshat
- Tutorials on RobotOS, USARSim, ope... — Wenxin , aditya , akshat
- Proposal Presentation — Wenxin , aditya , akshat
- Design Representation — Wenxin , aditya , akshat
- Mapping — aditya
- Localization — aditya , akshat
- Simulated Testing on USARSim, Bug ... — aditya
- Motion Planning — Wenxin
- Point Cloud Data — Wenxin
- Simulated Testing — Wenxin
- Combining SLAM with Navigation — (akshat)
- Obstacle Avoidance — akshat
- Simulated Testing — akshat
- Human Detection — akshat
- Overlay Humans on Map — akshat
- Real World Testing — Wenxin , aditya , akshat
- Slack — Wenxin , aditya , akshat
- Demo Preparation — Wenxin , aditya , akshat