IntelliSAR
October 18, 2019
IntelliSAR

Tianye (Arthur) Zhu

Yong Li

Derek Sun
Background and Motivation

- Safety and information of the environment are very important aspects of rescue missions
- Not fully understanding the environment and situation can lead to unnecessary risks and dangers

Examples:

- Cave rescue
  - Explorers trapped or lost
- Urban search and rescue
  - Victims trapped in Collapsed buildings
Goal

- Provide ability to remotely examine the situation and environment
- Reduce possible risks or dangers
- Improve efficiency of rescue teams in unknown environments
Method of Resolution

- A robot car that utilizes various sensors, machine learning, and computer vision to autonomously or remotely navigate around the surrounding environment and send data back to user.
Requirements Analysis

- Be able to be remotely controlled via Wi-Fi
- Be able to work in dim lighting conditions with night vision
- Be able to provide real time GPS location
- Gathered sensor data can be viewed remotely
- Can traverse uneven/sloped ground
- Be able to detect obstacles and navigate accordingly
- Be able to detect and classify objects
Requirements Analysis: Specifications

- Speed of up to 3 mile per hour
- Approximately 10 pounds
- Approximate size: 300 * 220 * 120 millimeters
- Approximately 3 hours of battery life
- Maximum grade: 30 degree
- Effective detection range of 4 meters
- Robust and durable enough to withstand minor collisions
Requirements Analysis: Inputs and Outputs

• Input
  • Camera data
  • Ultrasonic sensor
  • GPS tracker
  • Environmental sensors
  • User’s control signal

• Output
  • Live video feed with object detection
  • GPS data
  • Environmental data (temperature, moisture)
Design Alternatives

iRap Robot
- designed for SAR teams
- exploration, victim detection, 2D map generation
- high maneuverability
- remotely controlled

iRobot 510 PackBot
- designed for military personnel (high-threat battlefield scenarios)
- surveillance and reconnaissance, bomb disposal, vehicle inspection, etc.
- remotely controlled with few autonomous features

## Design Alternatives

<table>
<thead>
<tr>
<th></th>
<th>IntelliSAR</th>
<th>iRap Robot</th>
<th>iRobot 510 PackBot</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size</strong></td>
<td>Small</td>
<td>Medium-Large</td>
<td>Small-Medium</td>
</tr>
<tr>
<td><strong>Communication</strong></td>
<td>Wi-Fi</td>
<td>Wi-Fi/Radio</td>
<td>Radio</td>
</tr>
<tr>
<td><strong>Navigation Sensor</strong></td>
<td>Camera</td>
<td>LIDAR</td>
<td>Stereo Camera, LIDAR</td>
</tr>
<tr>
<td><strong>Visual Object Detection</strong></td>
<td>Common objects</td>
<td>Hazmat/QR code</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Target Audience</strong></td>
<td>Search and Rescue</td>
<td>Search and Rescue</td>
<td>Military</td>
</tr>
<tr>
<td><strong>Cost</strong></td>
<td>Low (&lt;$500)</td>
<td>High (~$30,000)</td>
<td>High ($100,000+)</td>
</tr>
</tbody>
</table>
Block Diagram

Robot (Raspberry PI 4B + Motor Driving Board + Chassis)
Peripherals -- Sensors, Camera, GPS

Robot (Raspberry Pi 4B + Motor Driving Board + Chassis)
Peripherals -- Sensors, Camera, GPS

- Requirements
  - Measure temperature
  - Measure geographic location
  - Capture video at dim light conditions
  - Navigation

- Implementations
  - Temperature sensor (BME280)
  - GPS (NEO-6M)
  - Infrared camera (5 megapixel, nightvision)
Robot

**Robot** *(Raspberry Pi 4B + Motor Driving Board + Chassis)*
Robot

- Requirements
  - House all sensors
  - Robust & stable
  - Certain degree of maneuverability
  - Peripherals scalability
  - IoT supportability

- Implementation
  - Chassis (214*280*114 mm)
  - 12V DC motors GA25Y370)
  - Raspberry Pi 4B
Raspberry Pi 4B

- Power: 5V DC (USB Type-C)
- Dimensions: 88 x 60 x 24mm
- Cores: 4 * 1.5 Ghz
- GPIO: 3.3V power rail 40

- Why Pi 4B
  - Performance comparable
  - Extensions
  - IoT Support
  - Economical

*Data Source
External PC

**Robot** (Raspberry Pi 4B + Motor Driving Board + Chassis)
External PC

- Requirements
  - Communicate with robot through Wi-Fi
  - Display sensor data
  - Display live video feed
  - Display GPS position
  - Transmit navigation instructions to robot (manual mode)
  - Object detection

- Implementations
  - Web GUI interface
  - Edge device publish data to Azure
  - External PC retrieve data from cloud
  - Render locally
  - Navigation signal send via cloud to IoT service on robot
Autonomous Navigation

Robot (Raspberry PI 4B + Motor Driving Board + Chassis)
Autonomous Navigation

- Requirements
  - Object detection
  - Obstacle avoidance
  - Control motors accordingly

- Implementation
  - OpenCV
  - Tensorflow
Autonomous Navigation

- OpenCV
  - open source computer vision library
  - used for image processing
  - object detection
  - You Only Look Once v3 (YOLOv3) - Joseph Redmon et al.

- Tensorflow
  - open source machine learning library
  - used to build neural network
  - neural network will help make navigation decisions
## Budget

<table>
<thead>
<tr>
<th>Component</th>
<th>Cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raspberry Pi 4B 4G</td>
<td>80</td>
</tr>
<tr>
<td>Infrared Camera (500w Pixel)</td>
<td>20</td>
</tr>
<tr>
<td>Chassis Platform</td>
<td>100</td>
</tr>
<tr>
<td>Motor * 6 (GA25Y370)</td>
<td>60</td>
</tr>
<tr>
<td>Sensors and GPS module</td>
<td>50</td>
</tr>
<tr>
<td>Li Battery 2200 7.4v mAh 25c</td>
<td>20</td>
</tr>
<tr>
<td>Battery Charger 7.4v</td>
<td>20</td>
</tr>
<tr>
<td>SD card 32GB</td>
<td>20</td>
</tr>
<tr>
<td>Azure IoT service</td>
<td>Free Tier</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>370</strong></td>
</tr>
</tbody>
</table>
Responsibilities

- Yong Li
  - Hardware selection, setup Pi
  - Azure related (Sensor data, GPS, video feed)
  - Sensor data transfer
- Arthur Zhu
  - Wi-Fi connectivity
  - Maneuverability
  - Autonomous navigation
  - Robot motor control
- Derek Sun
  - Object detection
  - Autonomous navigation
  - Application development
Roadblocks/Challenges

- Autonomous navigation
- Accurate object detection
- Component compatibility and system connectivity
- Robot maneuverability
Proposed MDR Deliverables

- Functional robot able to be remote controlled
- Azure setup for our system
- Train YOLOv3 model to be able to detect/classify certain objects

Responsibilities

- Yong Li
  - Robot functionality
  - Sensor connectivity, Azure connectivity
- Arthur Zhu
  - Networking, Motor control
- Derek Sun
  - Object detection
Proposed FPR and Demo Day Deliverables

FPR
▪ Live demonstration of IntelliSAR capabilities

Demo Day
▪ IntelliSAR on display
▪ Object detection demonstration
▪ Video that shows IntelliSAR in action
  ▪ Perspective of robot (w/ object detection)
  ▪ Data from sensors
Questions?