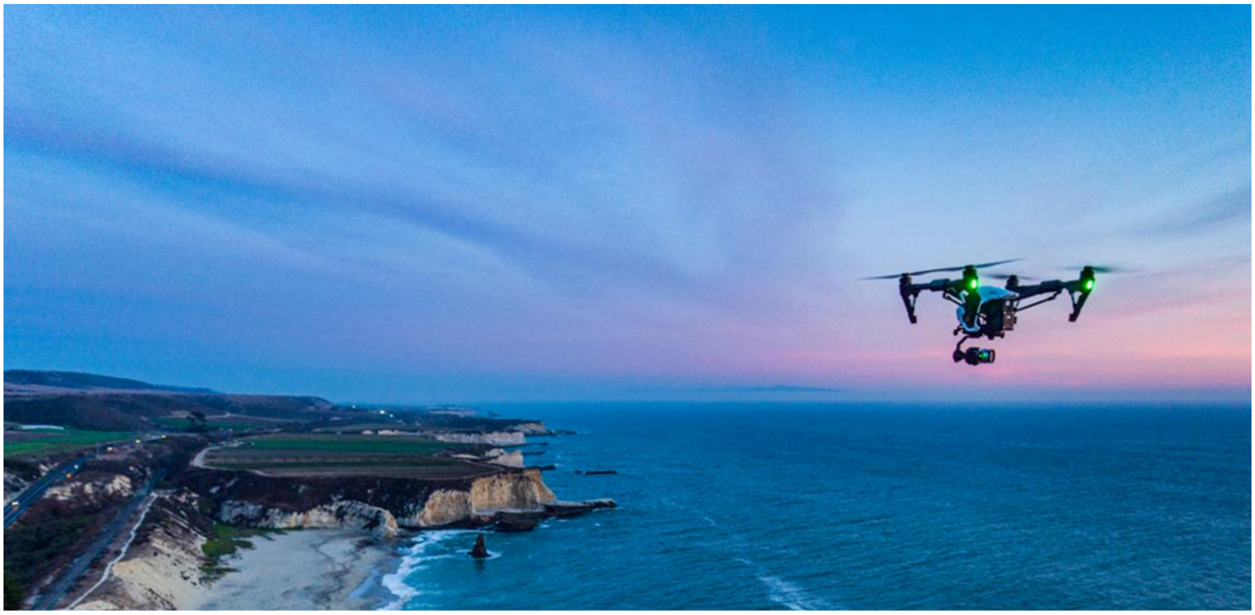

Autonomous Aerial Assistance for Search and Rescue

Final Report

May 11, 2017

Team F Rescue Rangers:
Karthik Ramachandran
Sumit Saxena
Juncheng Zhang
Xiaoyang Liu

Sponsor:
Near Earth Autonomy
Advisor: Paul Bartlett

Mentors:
Dr. John Dolan
Dr. Dimitrios Apostolopoulos

Abstract

In this project, we developed an autonomous aerial system to provide assistance in search and rescue missions in wildernesses. We tackle a simplified version of the problem where the human signatures are unoccluded.

This report summarizes the work done by our team, 'Team F: Rescue Rangers', on the project as a part of our Master of Science in Robotic Systems Development curriculum during the academic year 2016-17 at Carnegie Mellon University.

In this project, we have developed a system which requires minimal inputs from the user to conduct a search and rescue mission. Given waypoints to cover a given search area, the system can conduct a quick search over the area collecting RGB and IR imagery. We have automated the data processing to the extent that with the press of a single button, all the data is processed and we get the likely rescue locations as GPS locations. Rather than just trying to look for humans, our system also looks for other signatures that could possibly indicate human activity and thus might be useful. We also use sound to detect human voice activity to provide an additional layer of information which might be extremely useful for certain cases. Moreover, an efficient autonomous package drop subsystem makes it possible to deliver an item like a first-aid kit or satellite phone, urgently needed by the person to be rescued, much earlier than when the rescue team is able to reach the location.

We have fulfilled key functional and nonfunctional requirements of the project as per the performance metrics agreed upon after discussions with the associated faculty at Carnegie Mellon University and our sponsors, Near Earth Autonomy. With our system, we are able to cover a 50m x 50m area, process all the data and report likely rescue locations, and conduct a package drop, all within 25 minutes with minimal human supervision.

In this report, we describe in detail the whole system, the constituent subsystems, how we built them and how they perform. We also discuss the systems engineering and project management aspects of the project. In the end, we describe what we have learnt and what we think should be the important areas to focus on to make the system even better.

Table of Contents

1. Project description

1.1 Motivation

A typical search and rescue mission has very stringent requirements on time and the operating environment. This makes direct human involvement in the operation difficult and expensive, and has led to the use of automated vehicles to conduct the first wave of search. In such hazardous operations, where little information is available about the environment, aerial vehicles have a unique advantage of being able to quickly cover ground and gain an overview of the situation.

However, most of the existing approaches to SAR(Search and Rescue) using aerial vehicles currently rely heavily on teleoperated drones with minimal autonomy, which increase the risk for the rescue team and the cost of SAR operations. Apart from the huge cost, current approaches also impose strict piloting requirements on the operator, which limit the pervasiveness with which such technologies can be deployed. In addition, the capabilities of a teleoperated mission are extremely limited to certain categories of local terrain that always allow a link between the vehicle and the operator. All these issues in addition to the fact that there are roughly 11 SAR incidents each day at an average cost of $895 per operation[1], underscore the need for building systems that are as autonomous as possible.

1.2 Objectives

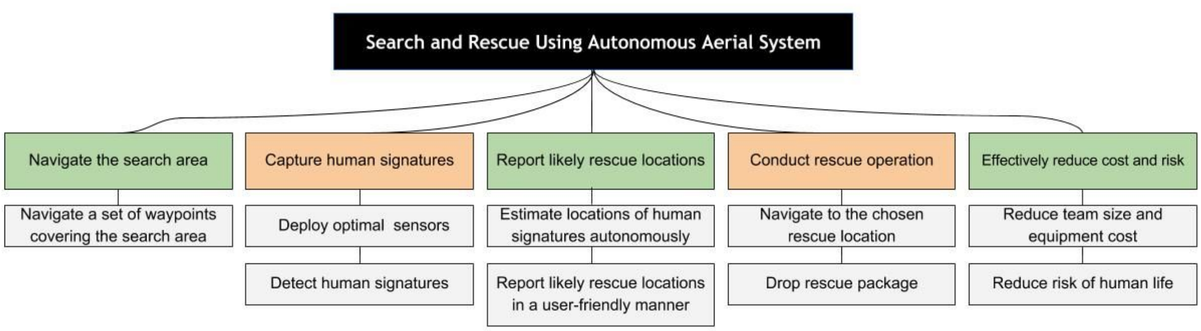

Fig 1.1 Objective Tree

As part of our quest to solve this challenging problem, we propose an autonomous aerial system for search and rescue, in order to effectively reduce rescue team size, equipment cost, as well as risk to human life. As is clear from Fig 1.1, this system should be able to autonomously navigate the search area collecting multi-modal sensory data, analyze the data to detect human signatures, report the likely rescue locations, and conduct a rescue package drop operation at the chosen rescue location efficiently and reliably.

2. Use case

Search and Rescue scenarios have potentially many use-cases that are challenging to fulfill. Since our final goal is to provide autonomous solutions to SAR, we have explored one such realistic use-case below from the Yosemite Search and Rescue (YOSAR)[7] that we believe can be fulfilled by a system like ours. An example scenario is depicted in Figure 2.1. Jamie is the Team coordinator for YOSAR and his team is responsible for conducting SAR activities in the Tuolumne Meadows region in the Yosemite Valley. While his team has repeatedly been touted as one of the most well-oiled SAR teams, he realizes the cost associated with maintaining this edge. They not only require to employ highly trained individuals with strong alpine skills (with an hourly rate of $23-24), but also have a huge budget for maintaining expensive helicopters and other equipments to be able to achieve high success rates in their missions. Looking for alternate solutions, he stumbles on a video showcasing the capabilities of the "Rescue Rangers" system in terms of being able to search for human beings in relatively un-occluded environments and decides to give it a shot. He orders the system online and the package arrives within the day and he spends a couple of hours assembling the system and familiarizing with the software for operating the system. Happy with his new gizmo, he wraps up the day unaware of the situation that awaits him the next day.

Figure 2.1 Illustration of a realistic Rescue Rangers SAR mission (distances not to scale)
[Image sourced from:[7]]

Early in the morning, he receives an emergency SAR SOS from the Yosemite Emergency Communications Center about two hikers gone missing during a routine hike, one of them reportedly injured. Jamie immediately sends an alarm to gather the team and prepare for rescue and while doing so, he wonders if this is the right opportunity to put his new gizmo to test. He fires up the drone and using the software, creates a waypoint mission for the drone to fly based on last known positions of the hikers and the drone takes off. In the meantime his team has assembled, and he briefs them on the mission and preparation is well underway for the mission to take off in 20 mins. Just as the mission gets started, the drone comes back to the base and Jamie, curious to evaluate it, looks at the processed data from the mission and is awestruck by what he sees. The system gives him precise information on the location of the hikers. He immediately communicates the location to his team and figuring that the team still might take some time to reach the location, he attaches a first aid package to the drone and launches the drone again, this time with a precise rescue location. The drone drops the package for the hikers and returns in no time. Shortly after that, the team whose mission has been reduced to a mere rescue mission with no search involved, bring the two hikers back to the base and express their surprise to Jamie, in seeing the victims with a first aid kit even before the team could find them. ! MISSION ACCOMPLISHED !

The use case we demonstrate for the current project accomplishes the key requirements of a system that can be employed for the above mentioned scenario. The key requirements that were the focus for this project are as follows:

- Eliminating the laborious task of analyzing sensor data manually.
- Build a learning system to automatically detect human signatures and suggest likely rescue locations

Figure 2.2 illustrates the use case that was demonstrated by the current system, and as can be seen from the figure, all the key requirements towards building an autonomous SAR system are satisfied .

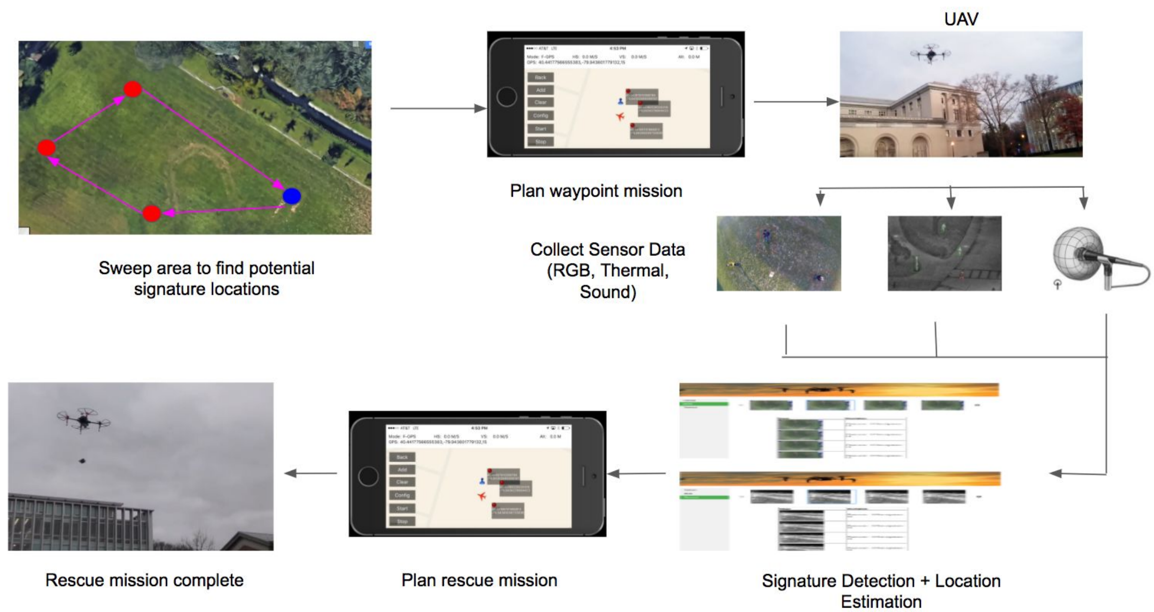

Figure 2.2 Rescue Rangers SAR usecase aimed for the project.

3. System-level requirements

3.1. Mandatory requirements

Mandatory requirements were arrived at after exhaustive research on the needs of search and rescue missions, numerous discussions with the sponsors and carefully considering what is achievable in the given timeframe. They were further modified based on feedback received on the Conceptual Design Review document.

Table 3.1 Mandatory Functional and Performance Requirements

Functional Requirements The system shall:	**Performance Requirements** The system will:
M.F.1. Autonomously sweep through a designated area looking for human signatures.	**M.P.1.** Attain up to 80% coverage of an un-occluded local search area with dimensions 50m X 50m
M.F.2. Collect perceptual data while navigating	**M.P.2.** Collect perceptual data limited to 3 types - IR radiation, visual imagery and sound
M.F.3. Process the data to identify human signatures	**M.P.3.** Identify at least 5 out of 7 of the locations with human signatures
M.F.4. Estimate and report locations of the human signatures identified	**M.P.4.** Estimate locations of human signatures with +-8m tolerance
M.F.5. Navigate to the chosen rescue location carrying the rescue package	**M.P.5.** Securely carry a rescue package weighing 100g
M.F.6. Drop the rescue package	**M.P.6.** Drop the package at the chosen rescue location with a tolerance of +-8m
M.F.7. Complete the mission within a stipulated time	**M.P.7.** Complete one iteration of search and rescue in < 25 minutes

Table 3.2 Mandatory Non Functional Requirements

Mandatory Non-Functional Requirements The system will:
M.N.1. Reduce the search team size required to <=2
M.N.2. Reduce risk to human lives
M.N.3. Reduce equipment cost required

3.2. Desired requirements

Table 3.3 Desired Non Functional Requirements

Non-Functional Requirements The system shall:
D.N.1. Have an interactive GUI to make it operable by an untrained human being ● Show detected signatures on the UI. ● Show options to pick particular rescue locations from the UI.

4. Functional architecture

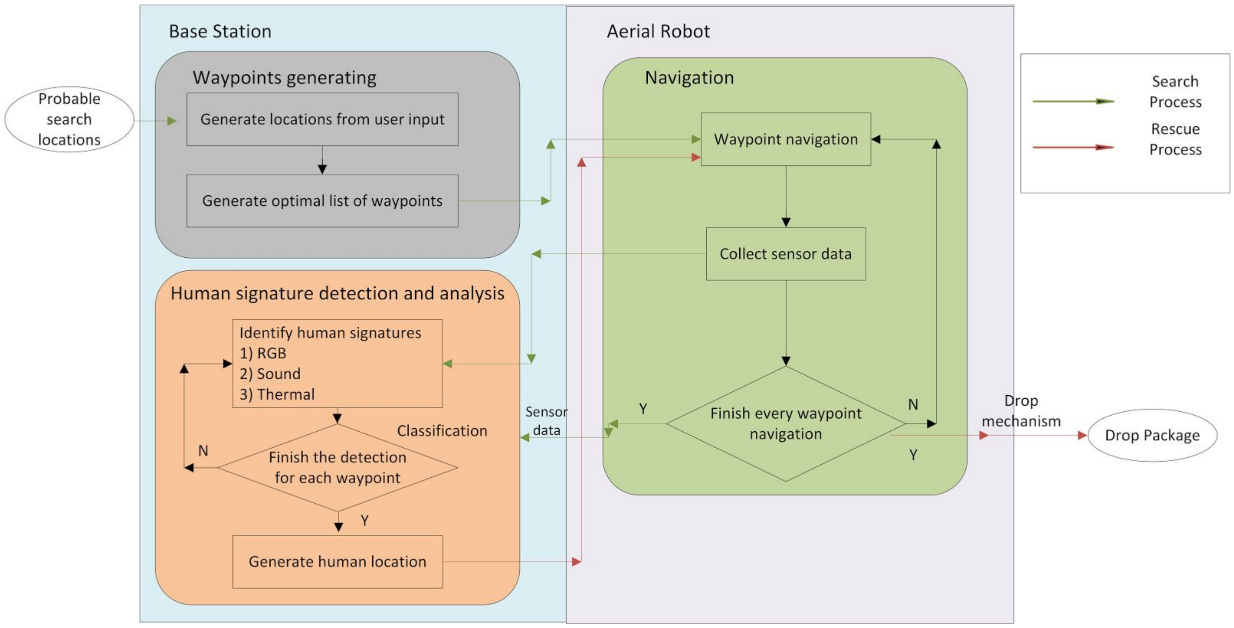

Figure 4.1 Functional Architecture

The architecture in Figure 4.1 is described below as a sequence of functions:

1. A mission begins with the user providing a list of geographic zones where the system should focus the search on. This information is then translated to GPS coordinates by the system and an optimal navigation path is generated as a list of ordered waypoints.
2. The aerial system navigates to the list of waypoints by flying with a back and forth pattern. After following this pattern, the drone will be able to capture reliable sensor data at each waypoint.
3. Once the waypoints are navigated and sensor data is collected, the drone returns to the ground station and initiates a data transfer.
4. Once the data is available, the ground station runs sophisticated algorithms to identify human signatures from the data and their precise locations.
5. The aerial system then navigates to the rescue location and drops a rescue packet as accurately as possible.

5. System-level trade studies

5.1 UAV Platform

Table 5.1 Trade study on UAV platform

	Weight(%)	DJI Matrice 100	DJI Matrice 600	3DR Solo
Cost	10	5.0	1.0	8.0
Flight Time	15	9.0	10.0	6.0
Flight Controller Capability	25	10.0	10.0	8.0
Payload Carrying Capacity	15	8.0	10.0	6.0
SDK Provided	20	10.0	9.0	7.0
Flight Simulator	15	7.0	7.0	8.0
Total	100	8.6	8.45	7.2

A proper UAV platform should not only provide stable flight performance, but also be easily programmable using provided APIs. Because of that, the two most important factors in choosing UAV platform for search and rescue operations are filght controller capability and SDK provided. In addition, the system is required to employ multiple sensors for extracting sufficient human signatures, which makes payload carrying capacity also a crucial part.

In order to have a structured search, the UAV will need to be able to run on battery for an extended period of time to completes the whole operation, which will require a battery to have enough basic flight time. Furthermore, cost is a also factor to be considered, since most of the flight platforms are expensive and we only have limited budget.

Considering all these factors, we finally select DJI Matrice 100 because of its superiority to other platforms in terms of its stable flight performance, extended flight time, strong payload carrying capacity, as well as its various available APIs provided by DJI SDK. Although Matrice 600 can provide similar performance with even better carrying capacity and battery time, it is too expensive to use for our project.

5.2 Sensors

Sensing is an important part in autonomous aerial system for search and rescue. Since we want the system to detect and identify locations of human beings, the system needs to extract potential human signatures based on multiple sensor data. In regard with selection of sensors, our system shall have image cameras for human signature detection and a sound sensor for human voice detection.

5.2.1 Image Camera

Table 5.2 Trade study on Image Camera

	Weight(%)	RGB+Thermal camera	RGB Camera alone	Thermal camera alone
Cost	10	3.0	5.0	6.0
Easy to Mount	10	6.0	9.0	8.0
Detection Accuracy	25	9.0	7.0	6.0
Information	20	10.0	8.0	9.0
Robustness to Environment	20	9.0	6.0	9.0
Availability from Sponsor	15	10.0	10.0	10.0
Total	100	8.45	7.45	8.0

To decide the best combination of image cameras, the main criteria is that whether the camera system can capture human signatures accurately. Except for that, the information detected in an image is also very important, because the more information we get through cameras, the more likely system is capable to extract useful human signatures.

Another factor which should not be neglected is that image quality may sometimes be influenced due to illumination or insufficient daylight. This makes the robustness to environment necessary to be considered. Other considerations include the cost of cameras, whether the camera is easy to mount, and availability from sponsor. Finally, the result turns out that the combination of rgb camera and thermal camera can provide high detection accuracy with great robustness to environment for our system.

Specifically, we would use the FLIR DUO R camera, which includes both a RGB sensor to provide high resolution visible images and an uncooled thermal sensor with affordable price. More information of this camera will be provided in the system description.

5.2.2 Sound Sensor

The sound sensor in our aerial system aims to detect human voice by analyzing the decibel and frequency of the external sound. Based on our requirement, we narrowed down on the following requirements

- High fidelity cardioid sound pattern for elimination of prop noise
- Functionality to record and store high quality sound in digital format
- Lightweight (mandatory less than 300 gms)
- Cost effective (mandatory less than 500$)

For this trade study, three microphones were considered namely: Tascam, Bose and Sennheiser. As can be seen from the trade studies, though Tascam has a slightly lesser fidelity, it trumps Bose and Sennheiser in cost and weight parameters and hence was picked ahead of the other two.

Table 5.3 Trade study of Microphones

	Tascam	Bose	Sennheiser
Sound pattern fidelity	7.0	9.0	8.0
Digital recording quality	9.0	9.0	9.0
Weight	9.0	6.0	7.0
Cost	9.0	5.0	6.0
Total	8.5	7.25	7.5

5.3 Human detection algorithm

Table 5.4 Trade study on Human detection algorithm

	Weight(%)	HOG+SVM	YOLO	Faster RCNN
Detection Accuracy	25	7.0	9.0	10.0
False Positive Rate	10	6.0	8.0	9.0
Speed	30	10.0	8.0	6.0
Required Training Dataset	20	9.0	7.0	6.0
Easiness of Implementation	15	9.0	8.0	7.0
Total	100	8.5	8.05	7.45

The human detection algorithm is an essential part of the whole system, as the successful detection of humans is the first step to accurately estimate the GPS locations of them. The detection accuracy and the speed are two important factors in deciding the appropriate algorithm to use, because missing any human during the operation or taking too much time to rescue are not allowable in our use case. Also, since it is difficult to find the dataset of aerial images including human beings and we probably have to create our own dataset for the training, the required size of the training set and easiness of implementation should also be considered.

As a result, we decided to use HOG+SVM for human detection because it doesn't require large training dataset and is very easy for implementation and training. Moreover, as we don't need to detect human beings in every frame, the performance of HOG+SVM is satisfactory in terms of the fact that it seldom misses humans in the whole process. The high false positive rate of the algorithm could be improved by fusing the results from RGB images and thermal images, as well as using the hard negative mining for the training. On the contrary, although the deep learning approaches like YOLO and faster RCNN can provide even better detection accuracy and false positive rate, these two algorithms are not realistic in our case due to their extremely long processing time without GPU and high requirement of the large training set.

6. Cyber-physical architecture

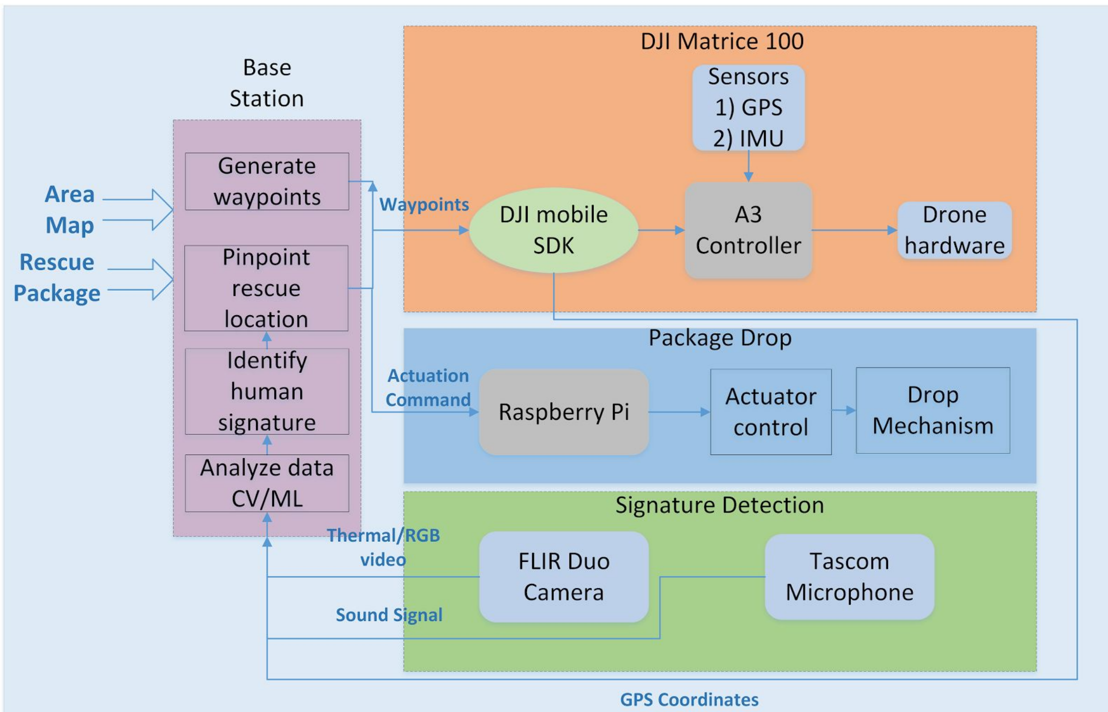

Figure 6.1 Cyber-physical Architecture

6.1 Autonomous Flight Subsystem

The autonomous flight system is based on DJI Matrice 100 platform. The GPS and IMU sensor embedded in Matrice 100 will primarily be used for navigation. The drone and navigation is achieved by interfacing with the DJI mobile SDK for generating waypoint and rescue missions.

6.2 Rescue Package Drop Subsystem

The Rescue Package Drop Payload System consists of the following components:

- **Package Drop Mechanism:** Custom designed and fabricated for the drone.
- **Location Estimation algorithm:** To compute locations for the detected signatures
- **Onboard computer:** Currently we use Raspberry Pi as the onboard computer. It is responsible for controlling the actuation of package drop mechanism. Also, the onboard computer serves as a server for communication between the ground station and the drone.

6.3 Sensing Subsystem

The sensors on the sensor payload consist of RGB and thermal camera as well as a sound sensor. The rationale behind using multiple types of sensors is so that the system can recognize different human signatures, and thus increase the possibility of finding humans.

After doing the trade study, we decide to choose the types of sensors as follows:

- Sound sensor: Tascom Microphone
- Thermal and RGB sensor: FLIR Duo R camera

6.4 Signature Detection and Analysis System

The Signature Detection and Analysis system resides in the base station and is responsible for analyzing all the sensor data and detecting human signatures.

Once the signatures are available, the software will generate a ranked list of candidates which will be presented to the operator to pick the best candidate. Once the candidate is available, the payload map can be used to lookup the coordinates of the location which will then be used by the drone for the rescue mission. Finally, those coordinates will be transferred to DJI onboard SDK for the next rescuing flight.

7. System Description and Evaluation

The system can be described in terms of the following subsystems

1. Autonomous Flight Subsystem,
2. Signature Detection Subsystem,
3. Rescue Package Drop Subsystem,
4. Backend processing console (Interface for user)

7.1 Autonomous Flight Subsystem

7.1.1 Descriptions

The autonomous flight subsystem comprises of the autonomous navigation and the waypoint generation system. Autonomous Navigation deals with being able to navigate the drone based on predefined set of waypoints provided as GPS coordinates. The DJI matrice 100 (Figure 7.1.1) was used as the drone and the DJI mobile SDK was used to control the drone. The DJI mobile app framework was extended and modified to add additional functionality to track and accept input GPS locations through a map. The app also provides functionality to set mission parameters like altitude and speed. A couple of screenshots from the app are shown below in Figure 7.1.2 and Figure 7.1.3.

Figure 7.1.1 DJI Matrice 100

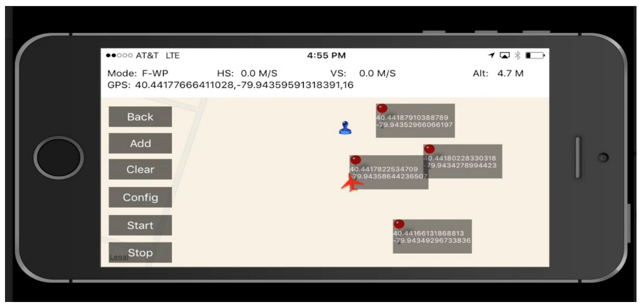

Figure 7.1.2. Screen for entering waypoints

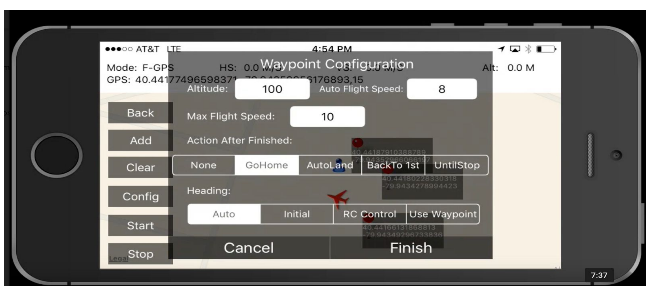

Figure 7.1.3. Screen for entering mission parameters.

A sweep approach was employed to cover as much area as possible within the designated borders. The sweep was conducted by providing the boundary waypoints to the drone and building a waypoint mission using these boundaries. The waypoint generation was implemented as an independent system and various tests were done by projecting the generated waypoints on a map and ensuring their accuracy. The system is implemented in such a way that the various parameters that govern the actual path of flight are all configurable. This ensures that the system can be launched in a specific configuration based on the requirements of the mission.

A server was written to track the pose of the drone which was used in conjunction with the signatures to estimate the location of the signature in the RGB and thermal images.

7.1.2 Validation and testing

The autonomous navigation component was initially tested using the DJI simulator and then subsequent tests were done through outdoor flights. A sample simulator test screenshot is shown in Figure 7.1.4.

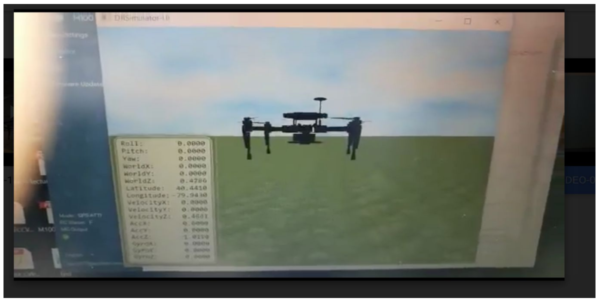

Figure 7.1.4. DJI Simulator test screenshot

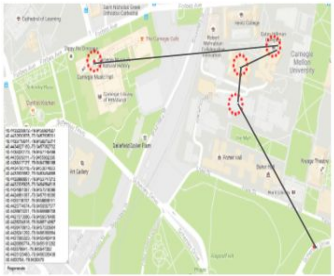

Figure 7.1.5. Waypoints on a map

For validating the GPS accuracy of the flight system, GPS from the two different sources were compared against each other. The GPS locations from the Drone's inbuilt GPS was compared and tallied with the GPS coordinates as reported by the smart phone. In addition, to understand any random errors within individual GPS sources, experiments were done to record GPS coordinates reported by each device for a given location and the standard deviations of the readings were noted. The two sources were found to be fairly accurate in terms of exhibiting any random errors. The one thing that was not tested extensively was the possibility of any static bias in the two sources. Though this is addressed to some extent by comparing and tallying the sources with each other, a more comprehensive test will help understand and address any static bias the systems might have. The waypoint generation system was tested by plotting the actual waypoints on a map and ensuring that the path generated by the system indeed goes through each of the required waypoints as shown in Figure 7.1.5.

7.1.3 Conclusions

While developing against the DJI SDK, we found it to be extremely flexible and rich in terms of the API to control the drone. This made it easy to rapidly iterate on the iOS App and add additional functionality within a short period of time. The UX that was developed was intuitive and easy to use in terms of being able to enter GPS coordinates.

7.2 Sensing Subsystem

7.2.1 Description

The sensing subsystem consists of a FLIR DUO R camera and a Tascam microphone with the recorder, in order to capture different human signatures. The images of the sensors are shown as below:

Figure 7.2.1 FLIR DUO R camera

Figure 7.2.2 Tascam microphone with recorder

The FLIR DUO R camera(shown in Figure 7.2.1) is a dual thermal camera designed for drone applications, which can record both RGB videos and thermal videos. The main specifications of the camera are:

- Thermal Imager：Uncooled VOx Microbolometer
- Thermal Sensor Resolution: 160 x 120
- Spectral Band: 7.5 – 13.5 μm
- Thermal Frame Rates: 7.5 Hz (NTSC); 8.3 Hz (PAL)
- Visible Camera Resolution: 1920 x 1080
- Visible Camera Frame Rates: 30Hz

The Tascam microphone(shown in Figure 7.2.2) is a cardioid microphone to capture the sound information within a certain direction. We use a suspension cable to keep the microphone away from the propeller noise, and the microphone is able to extract human voice using the melody analysis algorithm. The specifications of the microphone are:

- 3.8 ounces
- 6.3 x 2.8 x 2 inches

7.2.2 Conclusions

For the sensing subsystem, its **strong points** include:

- The total weight is small so that the Matrice 100 can easily carry the whole system
- The FLIR DUO Camera can record RGB video and thermal video at the same time with proper synchronization.

However, the sensing subsystem also has some **weaknesses**:

- The resolution of the thermal images is low
- The FLIR DUO Camera cannot record the video and be plugged into the computer at the same time, so the on-board processing is impossible.

7.3 Signature Detection and Analysis

7.3.1 Description

With the objective of search and rescue in wilderness in mind, we decided to go after the following two types of human signatures:

1. Humans themselves
2. Signatures related to human activity:
   (a) Bright objects: include bright clothing, tents, or mattresses generally used while hiking,
   (b) Hot objects: hot stove, fire, hot water, etc which might indicate human activity

Human detection:

Detecting humans in images is a challenging task owing to their variable appearances and wide range of poses. Our motivation behind developing an algorithm to detect the presence of human beings is that it can be used in various scenarios. More specifically, it can be applied in autonomous search and rescue operations through aerial platforms, which can effectively reduce the equipment cost and risks of injuries of humans.

In this project, we firstly implemented **Edge detection** in images for capturing potential human candidates (ROIs). Then we utilized **HOG** to extract features and classify whether there are human beings inside the ROIs based on linear support vector machine**(SVM)**[3][4]. We apply this model to both the RGB images and Thermal images. Plus, we've achieve fusing the two results and get a better result in human detection[5][6].

Detecting other signatures related to human activity:

(a) We implemented bright object detection by converting the images to HSV space and thresholding them based on saturation and value to obtain bright features, which was followed by morphological operations to get bright objects.
(b) For detecting hot objects, we used adaptive thresholding on thermal images.

In Figure 7.3.1, you can see the overview of signature detection and analysis subsystem. The modeling and analysis processes will be shown in details in the following part.

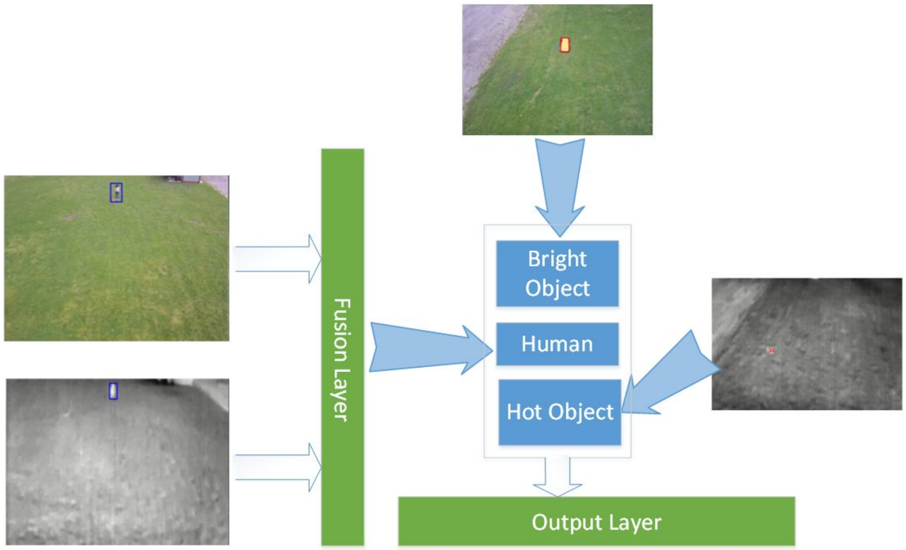

Figure 7.3.1 Overview of signature detection and analysis subsystem

The overall performance of the subsystem is shown in Figure 7.3.2. As you can see, the bounding boxes in frames are the final reported signatures, and blue boxes represent for those who are classified as humans while red ones below are bright feature (left) and hot object (right).

One of the key challenges in capturing sound signatures from a drone is to eliminate the background propeller noise. Standard techniques of filtering based on frequency does not work out of the box due to the high overlap in human voice and propeller frequencies. Various voice activity detection techniques were evaluated and finally a melody extraction approach was adopted. Melody extraction algorithms are generally used to separate melodies from background scores in recorded music, followed by a frequency filtering step. They use pitch salience tracked over the entire score to discriminate between melodies and background. Initial experiments with this technique showed surprisingly good results for the problem of detecting voice activity captured from a drone, and it was able to successfully filter out the background drone noise from the recording.

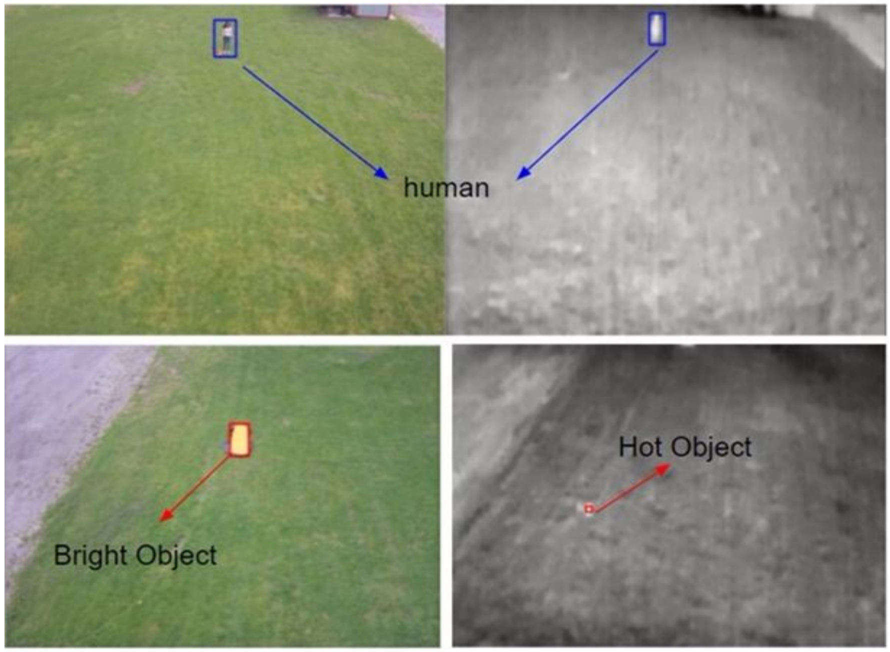

Figure 7.3.2 Overall performance of signature detection subsystem

7.3.2 Modeling and Analysis

Modeling of Edge Detection

Edge detection

- Use Sobel method[2] for edge detection (Figure 7.3.3 col 1)
- Use Dilate and Erode operations to fill the inner areas of edges and find connected components which exceed a minimum number of pixels (Figure 7.3.3 col 2)
- Rule out several improbable candidates based on the shape of connected pixels. (Figure 7.3.3 col 3)

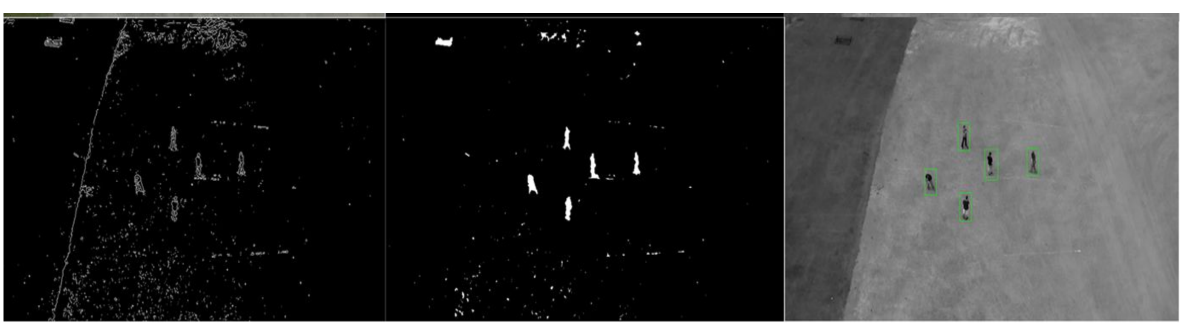

Figure 7.3.3 Description of Edge detection

Modeling of Multi Signature Detection

We want to detect some other signatures (mattresses, hot kettle, tents, etc.) except for humans. In this way, we will be able to know places where human are more likely to appear.

Therefore, this time I add the algorithm which can detect bright objects in RGB image and high intensity object in thermal image into the integrated object detection system. So, after combining the thresholding algorithm:

- In RGB images:
  We are now able to detect other objects except for humans
- In thermal images: We can find out the object with high intensity which can be used to eliminate false positives in thermal detection algorithms. To illustrate, we only output the bounding boxes which also belong to high intensity objects in thermal images.

Modeling of RGB+Thermal Human Detection Fusion Layer

- Combine ROIs from both algorithms
- Classify all ROIs by both RGB and thermal classifiers
- Choose those bounding boxes classified as humans by both algorithms
- Integrate intensity threshold algorithm into the thermal system

In other words, we used OR for choosing ROIs and AND for determining human bounding boxes. In this way, we get a bigger chance of considering potential human candidates and smaller chance of getting false positives.

Modeling of Output Layer

<u>Description:</u>

In the output layer, we need to report the human location after the fusion layer, as well as to report the multi signature locations. First, we try to eliminate false positives by checking whether the result after the fusion layer is within two consecutive frames. Also, in order to do our end to end test, we need the integrated signature detection system to generate an output file in a format that can be fed to the next GPS estimation system. After discussing the conversion of the two system, we decide to design the output to contain the name of output image, the timestamp which corresponds to each output image, the pixel location of detected signature, and the type of signature

<u>Testing Results:</u>

The result in table 7.3.1 shows that this method helps to reduce false positives effectively. But we still have some false positives in the beginning and the end of the video. Also, we can see the output file in Figure 7.3.4, which can be fed into the rescue location estimation algorithm.

Table 7.3.1. The false positives before and after the improvement

	False positives (video clip 1)	False positives (video clip 2)
Before eliminating false positives	62	70
After eliminating false positives	4	5

```
RGB/00145.bmp 11:24:02.133 1524 63 B
RGB/00146.bmp 11:24:02.167 1545 84 B
RGB/00147.bmp 11:24:02.200 1578 122 B
RGB/00148.bmp 11:24:02.233 1611 160 B
RGB/00149.bmp 11:24:02.267 1632 198 B
RGB/00150.bmp 11:24:02.300 1642 238 B
RGB/00151.bmp 11:24:02.333 1645 267 B
RGB/00152.bmp 11:24:02.367 1589 356 H
```

Figure 7.3.4 Output file example

Analysis of HoG + SVM for classification

HoG+SVM are very efficient classifying pedestrians[5]. However, we cannot confirm the feasibility of using this method before the analysis on aerial samples by using this algorithm. Therefore, we collect 299 pictures containing humans and 372 pictures without humans, and use them as positives and negatives to train the HoG+SVM classifier. It is very important to make all the pictures in the training set have the same size as the ROIs we will be used in the test set. Then, we used ROIs which are captured by applying two algorithm mentioned above as our test set. Before doing the final test, we labeled the ROIs with humans as positives and those without humans as negatives for the reason that it's easier to calculate the accuracy by comparing the test labels and predicted labels.

<u>Data Information:</u>

- Training set: 299 positive images, 372 negative images
- Test set: 111 positive images, 108 negative images

Table 7.3.2 Confusion Matrix of SVM+HOG classification

	Negative(Predicted)	Positive(Predicted)
Negative (Actual)	98	10
Positive (Actual)	26	85

<u>Testing Results:</u>

According to the Confusion Matrix shown in Table 7.3.2, we can get that

- The Predicted Positive Value is: 85/(26+85) = 76.6%
- The Predicted Negative Value is: 98/(98+10) = 90.7%

Conclusion:

The results demonstrate the feasibility of using HOG and SVM which can efficiently classify ROIs into the right classes.

Modeling and Analysis of Sound Signature detection

As mentioned before, sound signature detection was implemented as a frequency filter on top of a melody extraction technique that uses pitch salience to distinguish between human voice and the background score. Initial experiments were conducted using a simple phone microphone placed under the drone with voice activity from a few feet below the drone. Results for some of those experiments are shown in here, which is a link to a video with activations from the sound detection subsystem. The length of the bar indicates the strength of the detection at the given timestamp.

Subsequent experiments were conducted using a Tascam microphone suspended from the drone recording voice activity at a distance 6-8 feet below the drone to model realistic search and rescue scenarios. Once the sound samples are collected, the location of the signature is estimated using pose information from the flight of the drone to determine possible candidate locations. Since in this system, the voice activity was used more as an auxiliary signature to break the ties in cases where other signatures are inconclusive, the focus was more on capturing presence of voice activity rather than the precise direction of the source. Figure 7.3.5 and 7.3.6 show one such example of detected sound activity and potential signature locations identified after merging the timestamps with flight data. The orange vertical lines indicate the melodies isolated from the captured sound after applying melody extraction and the blue contour indicates the average background noise.

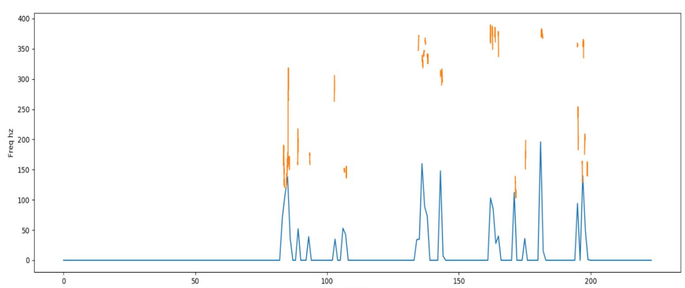

Figure 7.3.5. Isolation of melody from background noise

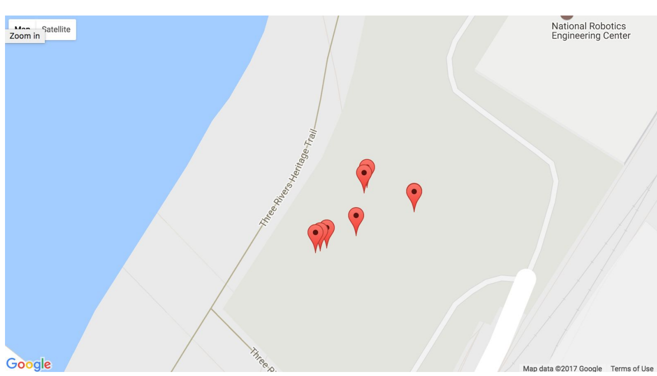

Figure 7.3.6. Estimated location of sound signatures

Testing Results for Sound Signature Detection:

The sound signature detection system was evaluated on multiple sound samples collected from various flights and was consistently able to detect voice activity. The one drawback was that it also detected some false positives in each of the flights. The results are summarized in the table 7.3.3 below. The data is sampled from 10 flights each with varying number of positive cases (from 1 to 4). It is difficult to represent output of sound signature as a confusion matrix since there are no supervised negative cases. Hence the column negative(actual), negative(predicted) is not applicable. negative(actual) positive(predicted) indicates the number of samples incorrectly predicted as positive. As can be seen the sound signature detects a lot of false positives.

Table 7.3.3 Confusion Matrix for Sound signature detection

	Negative(Predicted)	**Positive(Predicted)**
Negative (Actual)	NA	14
Positive (Actual)	0	28

7.3.3 SVE Performance Evaluation

Table 7.3.4 SVE Requirements for Human detection subsystem

SVE Step	**Procedure**	**Verification Criteria**	**SVE Performance**	**SVE Encore Performance**
5	Run integrated human detection software to report likely locations with human signatures.	The system should report at-least 5/7 locations with human signatures.	**Successful (6/7 detected)** The system was able to detect one out of two humans and all the other signatures	**Successful (7/7 detected)** The system was able to detect all the signatures

In conclusion, the human signature detection and analysis subsystem meets the FVE requirement, which is shown in Table 7.3.4.

7.3.4 Conclusions

The **strengths** of human signature detection and analysis subsystem are as follows:

- The idea of using combined signature detection to find human is very creative and efficient, since we can infer the human locations based on different detection result. It is a very realistic way when doing search and rescue mission.
- Since we use pre-processing algorithms to find ROIs that contain humans, it is very efficient in finding potential human candidates in an image.

- Being trained on aerial images, our human detection algorithm works pretty well.
- Our algorithm is able to process long videos very fast and thus offers a significant advantage over deep learning based approaches.
- Sound detection complements other detection algorithms and has an added advantage of being able to detect human presence even in occluded environments.

The **weaknesses** of human signature detection and analysis subsystem are as follows:

- Our human detection algorithm is designed to work well only for upright, unoccluded humans. It cannot handle occlusion and other human poses.
- Due to the limitation of HOG+SVM method, we can only detect humans with similar poses as those in the training set. Since we can only collect limited samples for the training set, our method cannot cover all poses.
- Right now, we are not able to associate multiple detections (across different frames and across different algorithms) of the same signature with each other
- Since we are using only a single microphone, we are only able to detect the presence of sound, not the direction of the source. Building a robust sound signature detection would require using a microphone array to estimate the direction of the sound source

7.4 Rescue Package Drop Subsystem

Rescue package drop subsystem is responsible for the following tasks:

(a) Rescue location estimation: Estimating the likely locations for rescue based on the outputs of the signature detection and analysis subsystem

(b) Autonomous package drop: Delivering a rescue package at the rescue location chosen from reported rescue locations. Due to payload limitations, our system has the capability to deliver a rescue package to only one location in one flight.

7.4.1 Rescue Location Estimation

After we identify human signatures in the data, we need to determine the rescue locations of the detected signatures and convey them in an efficient manner to enable in-time and successful rescue. Our rescue location estimation algorithm has following two steps:

1) Estimate GPS location of each signature for all the frames in which that particular signature is detected
2) Combine the GPS locations estimated for numerous frames to report the likely rescue locations in a meaningful way

Following process describes how we accomplish (1). The process has also been summarized as a graphic in Figure 7.4.1.

(a) Using the timestamp of the frame in which the signature was identified, get the flight data for the corresponding timestamp from the flight log. We get following information from the flight log:
   (i) Drone's GPS location
   (ii) Drone's altitude with respect to the flight starting point
   (iii) Drone's pose (yaw, pitch, and roll). Yaw represents the drone's heading.

(b) Get drone's altitude above sea-level (ASL) for the flight starting point and also for the timestamp when the signature was found by making a query to Google Maps API with drone's GPS locations for both the cases. Getting these ASLs helps us compute the drone's altitude above ground level (AGL) for that timestamp.

$$AGL_t = (ASL_t - ASL_s) + h$$

Where, AGL_t = altitude above ground level for timestamp t
ASL_t = altitude above sea level for timestamp t
ASL_s = altitude above sea level for flight starting point
h = altitude of the drone with respect to starting point reported by DJI SDK

(c) Using pixel location of signature in the image and information about drone altitude, drone pose, and camera pose, compute the displacement vector of the signature from the drone

(d) Using the displacement vector identified in (c), drone's heading (yaw) and drone's GPS location we get from flight data, compute the signature's GPS location.

After we determine the GPS locations for signatures for all the frames they are detected separately, we cluster the GPS locations based on distance to combine hundreds of identified locations to only a few locations. This clustering provides us two benefits:

(a) It helps us reduce the no. of locations to report while using all the information we have
(b) It automatically filters out false positives. For the purpose of GPS location estimation, a false positive is harmful only if it is found far away from the actual signatures. Since a false positive is not consistently detected at one specific location and if that false positive is far away from any real signature, we get a separate cluster for that false positive and this cluster has much fewer frames as compared to clusters we get for well-detected signatures.

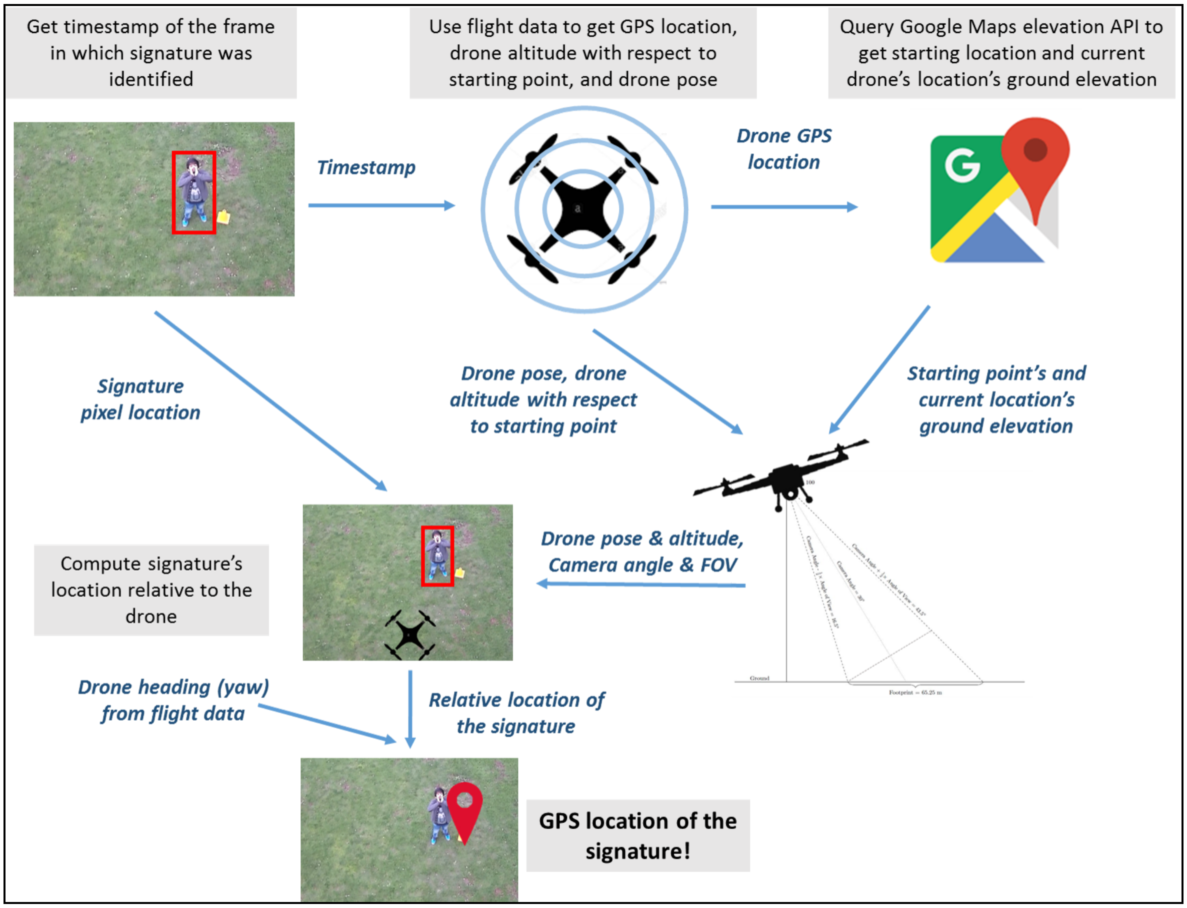

Figure 7.4.1. GPS location estimation of a signature using only one frame in which the signature was identified

7.4.2 Autonomous package drop

To achieve autonomous package drop, we designed a package drop mechanism and controlled it using an onboard Raspberry-Pi to initiate package at the provided rescue location.

To effectively fulfill the system functional requirements M.F.7 and M.F.8, our package drop mechanism is required to meet the following basic requirements. The mechanism should:

- Be able to carry a 100g package
- Enable easy package attachment
- Have a good grip on the package throughout the flight
- Release the package easily without causing any damage

Keeping these requirements in mind, we designed a simple yet robust mechanism, shown in Figure 7.4.2. The mechanism is basically a slider-crank mechanism mounted on a 3-D printed ABS body, facilitating required motion of the slider. The mechanism consists of the following main components:

1. Servo motor: To actuate the mechanism
2. Servo motor attachment: acts as the crank
3. Connecting rod: 3-D printed (ABS); to connect crank to slider
4. Slider rod: made of wood to keep it lightweight while providing sufficient strength
5. Body: 3-D printed (ABS); to provide mounting for the servo motor, passage for the slider rod, and space for attaching a package to the mechanism

Control of the mechanism was implemented using a Raspberry Pi mounted on the drone. Raspberry Pi and its housing are shown in Figure 7.4.3. During the rescue mission, the base station commands the Raspberry-Pi to initiate the package drop when the drone reaches within a certain distance from the rescue location and a certain altitude (to avoid dropping the package from a very high altitude). The Raspberry Pi commands the servo motor of the mechanism to rotate by the specified angle to open the mechanism and drop the package. The drone is then commanded to fly back to the home location.

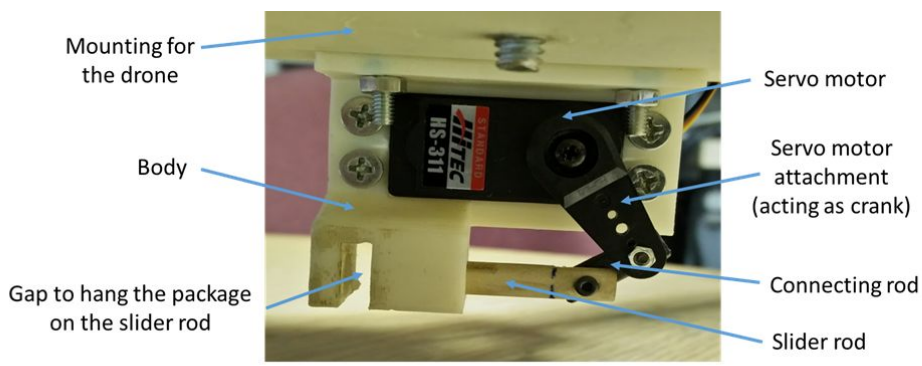

Figure 7.4.2. Package Drop Mechanism

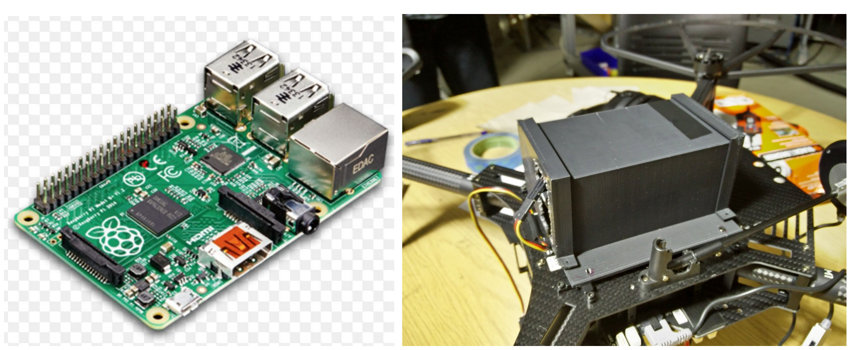

Figure 7.4.3. (a) Raspberry Pi used to command the mechanism
(b) Housing for Power supply and Raspberry Pi

7.4.3 Modeling

The Package drop mechanism was first modeled in SolidWorks to test the feasibility, to finalize exact design specifications of different components to be manufactured, and to estimate the angles the servo motor needs to rotate to open/close the mechanism Figure 7.4.4 shows the SolidWorks model for the mechanism.

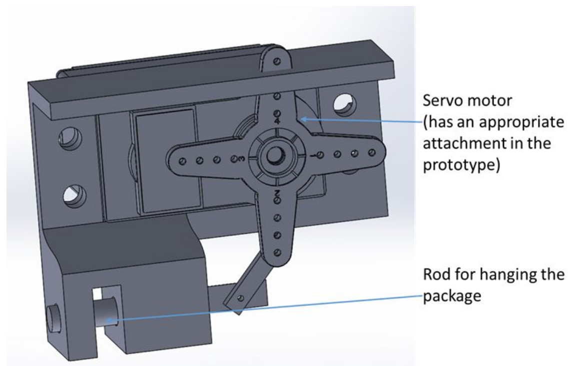

Figure 7.4.4. SolidWorks model for the Package Drop Mechanism

7.4.4 Testing

Testing of this subsystem was done in four phases:

Phase 1: Testing the reliability of package drop mechanism

Rigorous testing was done to ensure that the mechanism worked, as required. A package weighing ~160 g and dimensions 10cm x 10 cm was made for this testing. A lot of in-flight testing and 'attach/release' testing was done to ensure reliability of the mechanism.

Phase 2: Testing GPS location estimation for individual frames

In this phase, we took individual frames from the videos and tried to estimate the GPS locations based on the pixel location of the signatures in the images. We did this on frames from videos for different flights and made improvements.

Phase 3: Testing GPS location estimation clustering

In this phase, we took flights with the signatures set up in the same way as our Spring Validation Experiment (SVE). We kept human, bright and hot signatures in a 50m x 50m area and obtained the ground truth GPS locations by placing the drone at the locations of these signatures and taking multiple GPS readings from the drone's GPS. An average of the readings for each location was taken as the ground truth.

We conducted numerous flights, collected the data, and ran the GPS location estimation algorithm to obtain the likely rescue locations. Then, we compared the obtained rescue locations with the ground truth to validate the correctness of the algorithm. The algorithm was tested on about 25 flights over a period of one month with continuous improvements to make it more robust.

Phase 4: Testing the complete autonomous package drop

This phase was very similar to Phase 3, with just an addition of the part of launching a rescue mission after getting the likely rescue locations. Given that the main part, the GPS location estimation had been rigorously tested during Phase 3, we did the complete testing for only 10 flights and got consistently good results.

7.4.5 SVE Performance Evaluation

The following table describes the Rescue location estimation subsystem's performance on the relevant SVE steps.

Table 7.4.1. Rescue Package Drop Subsystem: SVE performance evaluation

Step	Procedure	Verification Criteria	SVE Performance	SVE Encore Performance
6	Run software to report GPS coordinates of the probable rescue locations.	The system should be able to report GPS coordinates of the rescue locations with a margin of error of +-8m	We got likely locations within +/- 5m of all signatures just except for the human. Human was reported to be 8.65 m away from actual	We got likely locations within +/- 3m of all the signatures
9	UAV flies to the provided GPS location, lands, and autonomously drops the package.	Rescue package should be dropped with distance less than 8m from the signature. (assuming the signature has not moved)	Autonomous package drop did not work, but the drone landed 2 m away while testing for a bright mattress, and 9 m away for the human	Autonomously dropped the package 2.3 m away from a human location

The subsystem could not give its best performance during the SVE as the onboard computer stopped working and we were not able to drop the package autonomously. Also, location estimation for the human did not work well enough.

After some improvements, the subsystem started giving consistently good results. In the SVE Encore, the subsystem worked better than expected, as can be seen from the Table 7.4.1.

7.4.6 Conclusions

Strong points:

- The rescue location estimation is pretty robust to false positives in the sense that it separates false positives into separate clusters easily identifiable by the user.
- The rescue location estimation is robust to change in drone's altitude above ground level (AGL) during the flight.
- The package drop mechanism is simple and easy to control: this prevents any complex issues. Also, grip on the package is not dependent on any electrical system, but rather on structural strength: this ensures we never lose grip on the package even if other systems fail.

Weak points:

- The subsystem is highly dependent on proper synchronization between camera recording and flight data and we do not have complete control over this synchronization. The maximum extent to which we have control over this synchronization is that we can sync the camera time with a phone's before every flight. We have noticed some lag between flight data and camera recording in some flights and it has the potential to affect the subsystem's performance drastically. A potential solution to this problem is to have an onboard clock that can trigger all operations (for pose tracking and camera) to have exact timestamps recorded for each of the systems.
- The GPS location estimation assumes while estimating location of a signature from an individual frame that the area the camera's field of view covers at any time instance is a horizontal flat plane. This assumption can degrade the subsystem's performance in regions with a lot of terrain change.

7.5 Backend processing console

7.5.1 Description

The backend processing console is not an independent subsystem but provides an interface to the user of the system. It allows the user to interact with the system and accomplish the search and rescue task. The key interfaces of the backend processing console are shown in Figure 7.5.1 below. As can be seen from the figure, the console consists of a sequence of screens for accepting the input, analyzing RGB, thermal and sound signatures, along with estimated human locations, and plotting the locations on the map.

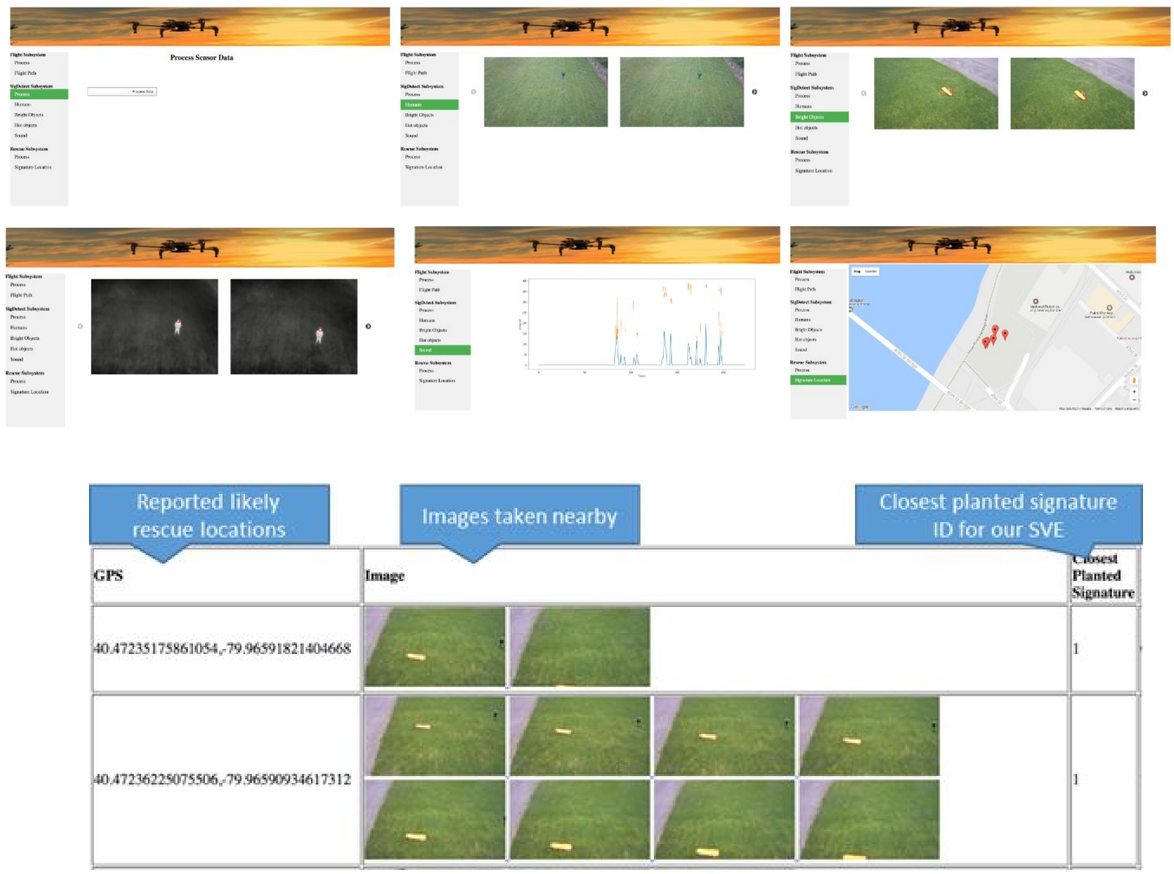

Figure 7.5.1. Backend processing console

7.5.2 Conclusions

- It provides an intuitive and clean interface for the user to interact with the system.
- It reports the likely rescue locations along with what is seen around those locations. The user has to use this to choose the best rescue location for package drop.

7.6 System SVE Performance Evaluation

This section summarizes the overall system performance during the SVE and SVE Encore. For detailed analysis, refer to analysis done for each of the subsystems.

S. No.	Procedure	Verification Criteria	SVE Performance	SVE Encore Performance
1	Place seven signatures specified above, at various locations in a 50m x 50m area.			

2	Place UAV on the ground, turn it and the payload power ON. Create and launch a search mission through the Mobile app.			
3	UAV sweeps the area (50m x 50m) collecting sensor data.			
4	Transfer data from the payload to the laptop.			
5	Run integrated human detection software to detect human signatures.	The system should be able to detect at least 5/7 human signatures planted	6/7 signatures detected. The system was able to detect 1 out of 2 humans and all the other signatures	7/7 signatures detected. The system was able to detect all the signatures
6	Run software to report GPS coordinates of the probable rescue locations.	The system should be able to report GPS coordinates of the rescue locations with a margin of error of +-8m	We got likely locations within +/- 5m of all signatures just except for the human. Human was reported to be 8.65 m away from actual	We got likely locations within +/- 3m of all the signatures
7	Select one location out of the reported rescue locations for the package drop			
8	Detach the microphone from the payload, attach the rescue package. Launch rescue package drop mission through the Mobile app.			
9	UAV flies to the provided GPS location, lands, and autonomously drops the package.	Rescue package should be dropped with distance less than 8m from the signature. (assuming the signature has not moved)	Autonomous package drop did not work, but the drone landed 2 m away while testing for a bright mattress, and 9 m away for the human	Autonomously dropped the package 2.3 m away from a human location
10	UAV flies back to the home location and lands.	Total time < 25 minutes	Total time > 25 minutes	Total time < 25 minutes

8. Project management

8.1 Schedule

The schedule of our project is shown as Figure 8.1. As you can see, our schedule is created based on our subsystems. We have four subsystems and each of them will have some detailed task in the schedule, including autonomous flight subsystem, sensing subsystem, signature detection and analysis subsystem, and package drop subsystem. Additionally, system integration and testing, as well as project management are also important parts of the project, thus they are also listed in the schedule.

Figure 8.1 Schedule

In our schedule, the blue blocks represent the plan for the project and give us a general idea about when we will work on this subsystem. The green blocks are the tasks that we have already finished, while the yellow blocks stand for what we haven't achieved yet.

Overall speaking, we followed our schedule pretty well. The autonomous flight system was finished by the beginning of February, and we spent most of the time on signature detection and rescue package drop subsystem during the rest of the semester. However, the sound sensor mounting and the electronics for the power distribution board in the rescue package drop system were not completed on schedule. The main reason was that we realized that these two tasks are not necessary because we could directly use the package drop mechanism to carry the microphone during the flight instead of designing a new sensor mounting, and a mobile power bank could simply replace the power distribution board due to the simple power requirements of all the components aboard. Consequently, we skipped those two tasks in the schedule, but focused more on other crucial tasks to make the whole system robust.

8.2 Parts list and budget

Table 8.1 Part list I(Sponsored by Near Earth Autonomy)

Description	Manufacturer	Model	Unit	Weight (g)	Cost
Dual Camera	FLIR	FLIR Duo R	1	84	$1299

Table 8.2 Part list II(Sponsored by Near Earth Autonomy-Parts still need to be finalized)

Description	Manufacturer	Model	Unit	Weight (g)	Cost
Aerial Platform	DJI	Matrice 100	1	680	$3250
Battery Heater	DJI	Inspired 1	1	100	$20
Battery Sticker	DJI	Inspired 1	1	0.2	$2
Audio Recorder with Shotgun Microphone	Tascam	DR-10SG	1	50	$199.00
Mount for Hero 4	Gopro		1	80	$28.99
10 feet rope	Paracord Planet		1	20	$6.79
Wind muff for microphone	DR-10SG		1		$12.99
Total budget $5000/ Total cost $3519.77					

Table 8.1 shows the items provided by our sponsor Near Earth Autonomy, and Table 8.2 lists those that need to be purchased using our own budget. In all, we have a $5000 budget, and our total cost in the fall semester was 3519.77$, around 70.4% out of the total budget. The key item for us was the DJI Matrice 100, which cost $3250.

8.3 Risk management

The risk analysis for the project is listed in Table 8.3.1 below and depicted as a risk template in Figure 8.3.1 There are 4 major risks related to the availability of flight locations for the drone, ability of our signature detection techniques to detect humans accurately, impact of lack of time synchronization between drone pose data and camera frames, and impact of weather on our drone components. Mitigation strategies were devised for each of them and were pursued aggressively and the risks were mitigated

Table 8.3.1 Risk Analysis

ID	Description	Likelihood of Occurrence	Level of Impact	Area of Impact	Mitigation Strategies
1	Difficult to find location for scheduling outdoor flying tests.	3	High	Time, Reliability	1. Talk to multiple flying clubs to find a location where the drone can be flown. 2. Explore NREC location suggested by sponsor for flights
2	Difficulty in achieving high accuracy with signature detection	2	Medium	Reliability	1. Collect more sensor data to improve training sets. 2. Explore alternative algorithms for improving accuracy. 3. Explore options of getting sound sensor closer to the ground.
3	Impact of time synchronization issues between drone pose and camera frames	3	High	Accuracy of human location estimation	1. Add validation to ensure camera is synchronized with the mobile app before each flight.
4	Impact of weather conditions on drone components.	3	High	Time, Reliability, Cost	1. Order additional backup components 2. Ensure all components and backups are preheated and ready before each flight.

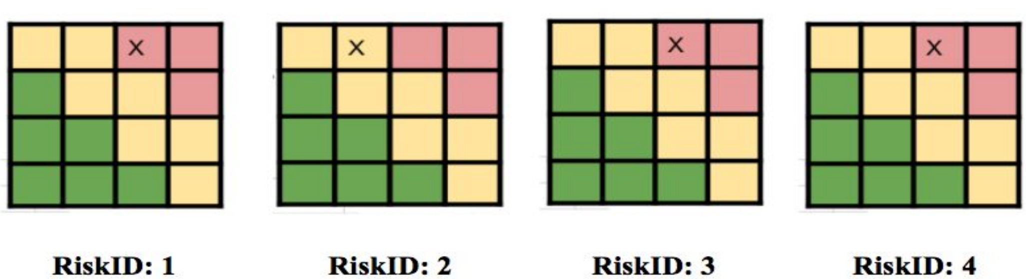

Figure 8.3.1

9. Conclusions

9.1 Lessons Learned

This project, exposing us to development of a complete system, helped us learn several important lessons about building a system and demonstrating its capabilities.

- **Preparation for demonstration: System end-to-end testing:**
  Before the SVE, we had rigorously tested our subsystems and had conducted end-to-end testing as well a few times, but in a casual manner. After the SVE, we realized the importance of rigorous end-to-end testing, and rehearsing and thinking through various steps of the demonstration. After careful planning, we were able to give a smooth and successful demonstration for the SVE Encore
- **Planning and following risk mitigation strategies:**
  At the beginning of the Spring semester, we mitigated an important risk of not being able to get enough time to work on our sponsor's drone to work on the project, by making our own drone as the primary drone and buying relevant sensors of it. Though it was a difficult decision to make at that time given the additional amount of effort we had to make to start everything afresh with our own sensors, it was a great decision in the hindsight as it gave us a lot of testing freedom, which we would not have gotten otherwise.
- **Proper planning of testing procedures:**
  Initially, we did not plan our tests very well and at times, missed some important aspects of testing which led to repetition of those tests. Since we had to go to NREC to do all the testing, this unplanned testing cost us hours and delayed our progress. With time, we learnt the importance of planning our tests and were able to test our system/subsystems much more efficiently later.
- **Thorough research on the newly launched products before purchase:**
  We decided to get the newly launched FLIR Duo R camera for our project. We did as much research as we could and the camera turned out to be fine for the essential aspects. But, there were some features of the camera for which we could not get direct details anywhere, like ability to access the radiometric data. After we tested the camera, we found that the radiometric data could be accessed only through proprietary FLIR software. Also, the camera data could not be accessed while recording. Though these features were not essential for us and did not affect our project. We got the idea that we should do a really thorough research before buying a newly launched product.

9.2 Future work

Although our system met all the requirements we defined and set out to achieve, we feel that it can be improved further to be a great product. Following are some of the ideas we have:

1. **Human detection algorithms robust to occlusion and human pose**
   Our human detection algorithm currently is able to detect only unoccluded upright humans while this may not be the case in many search and rescue scenarios. It would be better if we develop algorithms more robust to occlusions, use of deep learning techniques can also be explored.
2. **Initial planning algorithm**
   While giving waypoints to cover a search area can work well for areas which the user has good knowledge about, it would be better to have a planning algorithm which could create a navigation plan based on the terrain of the given search area.
3. **Onboard vision and adaptive planning**
   The search can be further optimized and the whole mission time can thus be reduced if we have onboard vision processing and adaptive planning. Then, the drone would be able to modify it search based on what it observes. We could start with a high altitude flight, and based on what is observed in different areas, the system would go down and have a closer look at the areas in a priority order. Also, we might not even need to cover the whole area in this case and the search can be terminated if the system is able to find the required signature earlier.
4. **Better data logging**
   Since our system is highly dependent on time synchronization between the sensor data and the drone flight data, it would be better to have a central data logger, which would log various sensor data and flight data together so that there are no synchronization issues.

10. References

[1]https://www.researchgate.net/publication/26795932_Dead_Men_Walking_Search_and_Rescue_in_US_National_Parks
[2]https://www.mathworks.com/help/images/examples/detecting-a-cell-using-image-segmentation.html
[3] http://www.pyimagesearch.com/2014/11/10/histogram-oriented-gradients-object-detection/
[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. Computer Vision and Pattern Recognition (CVPR), 2005. 1
[5] A. Gaszczak, T. P. Breckon, and J. Han, "Real-time people and vehicle detection from UAV imagery," in Proceedings of the SPIE Conference Intelligent Robots and Computer Vision XXVIII: Algorithms and Techniques, 2011
[6] Jan Portmann, Simon Lynen, Margarita Chli and Roland Siegwart, "People detection and tracking from aerial thermal views"
[7] Use case has been designed after reading cases "Yosemite National Park, California". Refer to links: https://www.nps.gov/yose/blogs/psarblog.htm, https://www.nps.gov/yose/getinvolved/sar_jobs.htm, https://www.nps.gov/yose/blogs/Three-Distressed-Hikers-Rescued-Tuolumne-Meadows-Area.htm, http://www.climbing.com/news/10-things-you-didnt-know-about-yosar/