System for Open-Water Person Detection

By

Justin Toth

A Thesis submitted to the Graduate Faculty of Fairfield University in partial fulfillment of the requirements for the degree of A Master of Science in the Electrical and Computer Engineering program.

Advisor: Professor Douglas A. Lyon, Ph.D.

Electrical and Computer Engineering Department

Fairfield University, Fairfield CT 06824

1. Introduction

This section contains four subsections that describe the problem statement, the motivation for solving this problem, the approach taken and a discussion of the engineering ethics involved. These subsections will set the foundation for this thesis which will be expanded upon in later sections.

1.1 Problem Statement

In order to locate a person requiring rescue we seek to design and implement a system that makes use of mesh networking with multiple unmanned aerial vehicles equipped with image sensors to locate the person.

1.2 Approach

Object detection through computer vision and the use of mesh networks are technologies with a rich history of research and development. Thus, the first step was to perform a historical review (2) to leverage previous work relating to the problem. This informed the development of a system level design (3.1), which was further refined with review, comparison and selection of hardware (3.2). Software integration (3.3) of selected components followed, and a prototype system was built and tested against the parameters outlined in section 1.2.1. The results of the prototype test are discussed in section 4.

1.2.1 Testing and Evaluation

In order to validate the system design, experimentation was used to test and quantify the system's effectiveness. System performance was measured by successful identification of a person in the given search area and the time it takes the system to identify a person within the sensor's vision. For the test, the device will be placed in an area where it can view an open scene free of obstructions. The system will be run for a set amount of time (30 min) and all stored images will be reviewed after the test. The scene should include the sparse distribution of people which when present should register a detection by the system. These instances should be monitored and any noted failures recorded. When the system detects a person in a captured image, a number of previously captured images are reviewed post-test. The reviewed images will be used to determine how many images the person was in before the system responded. This image count and the number of images per second of the system are used to calculate the time to detection. The recorded data of the system's response to targets in frame is used to calculate success rate.

1.2.2 Alternate Approaches

Additional methods for testing and evaluation will be built full scale. Issues for power plant and robustness of the vehicle will need to be solved. A full scale build will also run the risk of losing drones over water or to bad weather, but the results should be improved.

1.3 Motivation

This section is divided into four subsections. 1.3.1 talks to the novelty of the solution and 1.3.2 concerns the contribution of the thesis to the field. Section 1.3.3 discusses the impact of this problem on society, and 1.3.4 details the monetization potential.

1.3.1 Novelty of the Solution

Identifying a person in distress over a large area offers many challenges for image processing. Today this task is performed by a low number of manned aircraft with expensive high resolution infrared sensors. With the high cost of the infrared sensors, it is unlikely that a lower cost drone could be used as a platform. The approach of this thesis is to use a lower cost infrared sensors used on multiple drones with sensor fusion for overall performance enhancement. Alone, the resolution of lower cost sensors cannot match the range and precision of the higher cost units. The drones are spread out over the search area and communicate over a network.Sensor fusion with many low-cost sensors perform better than a few high-cost sensors without any fusion.

1.3.2 Contribution

The contribution to the field of sensor fusion includes a method of image processing of data from IR sensors on device nodes and communicating results over a mesh network in order to locate people at reduced cost compared to current techniques.

1.3.3 Societal Impact

Through government spending, society funds groups which will perform lifesaving operations such as firefighting, emergency medical response or search and rescue. This research looks to have a positive impact on the search and rescue groups by decreasing cost of their tools while maintaining efficiency. The Bureau of Transportation Statistics [32] lists the number of lives saved by the US Coast Guard at an average of 5,879 per year for 1985 to 2013 and 842 lives lost at sea

1.3.4 Total Addressable Market

To estimate the total addressable market we will review groups which could utilize this technology and what their current costs are. For search and rescue (SAR) occurrences over water there is a small number of groups with a small amount of aircraft. A typical customer would be a government entity or in some cases a private company such as the case with the Bristow Group which have a contract to provide SAR for the UK. According to The World Factbook [10] there are 107 countries with coastlines greater than 350 km and of those countries 74 have military aircraft. The Bristow Group uses S-92's which cost $27 million each with upkeep at $900 k per year. Cost in this sector would be $2 billion in assets and $66.6 million spent each year on upkeep.

1.4 Ethics

While developing new or novel systems aspects of ethics need to be assessed in order to have a positive impact on society. Sections will be dedicated to the review of Virtue (1.4.1) and Utility Ethics (1.4.3) as well as reviewing IEEE code of ethics in section 1.4.2.

1.4.1. Virtue Ethics

The basis of this thesis has a positive and ethical proposed application, but the algorithm may have some application for a weapon or military use. These other applications need to be assessed ethically if proposals for these implementations arise.

1.4.2. IEEE Code

IEEE Code of Ethics 3 [31] states that when creating a system it is important to be honest, critical and thorough with the testing process and reporting of the test results. Since the proposed application of this system will be to detect persons in distress if the system does not perform as stated and the user puts absolute trust in the detection a failed detection may result in loss of life.

According to IEEE Code of Ethics 7 [31] with this work building on the advances and problem discovery of others and building the knowledge base of the author through readings and discussion, it is important to properly cite and give credit where it is owed.

1.4.3. Utility Ethics

The proposed application of the thesis is a good utilization of its capabilities, but other utilizations may create additional opportunities to maximize its utility for the greater good. The author and users should be willing to embrace additional implementations, but be mindful of uses which may conflict with virtue ethics.

2 Historical Review

This section will review the existing works of research and development in the field of computer vision and drones as they relate to this thesis. Section 2.1 looks into the guidance systems for aircraft by use of visual sensors. Section 2.2 reviews works relating to multiple drone system and specialized drones. While Section 2.3 deals with research in visual sensor technology.

2.1 Visual Guided Systems

“Vision Systems for Autonomous Aircraft Guidance”[1] goes in depth in adapting autonomous drone technology to mimic aspects of observed biological solutions in the animal kingdom. A large focus of this cited thesis is the control of a UAV through inputs from a stereo visual sensor system which would replace or augment typical UAV sensors such as GPS or radar. A smaller portion of the cited thesis is more relevant to the problem statement of this thesis which goes into target detection and tracking from an aerial vehicle. The problem of how to identify and track ground based targets from a moving aerial platform remains open. The cited thesis’ proposed solution is significant as it is successful in identifying targets based on common colors on the target and the color differences between the target and the background and through on aircraft processing continuously track targets frame-to-frame by predicting target and aircraft movements and the expected change in target position from input sensors. After the cited thesis’ experiment was completed, R. J. D. Moore observed system limitations. It was noted that the resolution of the system’s sensors limited the system’s ability to track targets at longer ranges and were negatively impacted by visual imparities due to environmental factors such as weather. Ultimately the proposed solution was an acceptable and computationally-efficient solution. Article is cited by two other works. [1]

2.2 Fleet Composition Networked Drones

With the emergence of drones to the commercial market resources have been dedicated towards researching different applications for them. In 2014 a competition was held in the

Netherlands for a drone search and rescue scenario. Cui et al. [15] outlined their winning methodology. The scenario was an urban environment used to portray a town after a disaster. Utilizing multiple drones, they stitched many images together to output a map of the town, navigated and mapped the interior of a previously unknown building, land on a rooftop and decipher a digital number displayed on a building. Their application of image stitching and multiple drone utilization brings an interesting approach to search and rescue. Opportunities may exist for applications of a drone fleet for ocean search and rescue. Cui et al. [15] however bring up the downside of drone search and rescues limits. Computational power available on board, as a result of this they reduced the output quality of their stitched images eliminating certain post processing to reduce overall time to completion. In the applications described for this thesis accurate outputs in real time are required, so the computational power of a drone may not be sufficient enough. Also the communication range of drones would be called into question.

Similar to the previous competition, Lee et al. [16] are investigating the application of drones in disaster relief efforts. They are utilizing the drones LIDAR and IR cameras to detect victims within buildings. For their initial application the drone was not autonomous, instead a pilot navigated the interior. Their experimental results show effective use of the sensor fusion in different lighting conditions. However the sensors chosen are specialized in short-range detection. For ocean applications range is a priority. The 10m range of the sensors used in this experiment are unusable for this thesis' applications.

Lee et al. [18] present their ground control station (GCS) drone network. Their design assumes that high connection uptime to the network by a drone is required for drone control, but that the dynamic nature of the drones motion will often hinder this and the network will need to adapt to the movements of the drones. With the GCS controlling the movements of the drones, they theorize that the changes to network routing and be predicted by the GCS before a disconnect occurs offering faster recovery time. During network type selection for this thesis, uptime of the connection would need to be compared against the autonomous nature of the drones would need to be accounted for.

In additional works Lee et al [18] cite the popular method of ad hoc networks for UAVs. Specifically the works of Lin et al [19]. Similar to Lee et al [18] problem statements regarding UAV mobility and its effect on the network performance. The standard network routing example showed the capability of a given network routing path to not be optimal for number of node jumps nor geographic distance. The result was an impact on network performance. To resolve this their ad hoc network relies on Mobility Prediction based Geographic Routing (MPGR), which attempts to predict the UAVs movement and properly set up the network to increase overall performance.

For a drone fleet a top level system is required to ensure separation to avoid collision and master flight / search plan is required. S. Yoo et al.[24] describes such a system. They utilize drone networks, gps, IR and acoustic sensors to avoid collisions. The reasoning for using so many sensors is their system must work indoors and outdoors. Certain sensors such as GPS has shortcomings indoors and other sensors must compensate. Additionally the precision of off the shelf GPS does not meet their requirements. The system described in this thesis will only operate outdoors and the ideal situation would have drones at max range from each other to cover the most area as possible for the search. This distance would be much larger than the GPS resolution. For this thesis a higher-level system controlling the flight areas and maintaining drone distances utilizing standard GPS will sufficient for future works.

2.2.1 Specialized Drones

Stating inherent issues with drones being short range and overburdened with sensors, Alex and Vijaychandra [17] start to address some of these problems. In their approach, they split the roles of the drones into 3 categories; pathfinders which map the location in 3 dimensions and pass the information to other drones on the network, human detection drones which utilize IR cameras to identify victims and cargo drones which are heavier lifting and carry supplies. Although the mapping and cargo drones aren't really necessary for this thesis, the concept of different takes allocated to specialized drones to improve range and duration of missions is promising as well as the application of a LAN to pass information between drones. With the given range of a search and rescue field each drone would be encumbered by weight of higher power transmitters and range reduced by the power required to transmit. The task might be better suited for a heavier lift drone which reduces weight and power consumption by not be burdened with actual person detection. Alternatively if a well dispersed search pattern is used, information could be passed drone to drone in shorter distances, but may consume similar power consumption for the overall fleet. Alex and Vijaychandra [17] didn’t conduct physical experiments to validate their theories, but instead opted for simulation. They state the use of Gazebo software to simulate all activity.

2.2.2 Target Sharing and Confirmation

In a networked sensor system with a large number of sensors, transmission time and power resources can be stretched thin. E. Masazade and A. Kose [21] propose a proportional time allocation algorithm to determine the amount of transmission time a sensor will get. The algorithm will favor those sensors which are near the desired targets. With reducing the amount of sensors transmitting the unconstructive bandwidth on a given frequency is reduced and so would error rates due to crosstalk. E. Masazade and A. Kose [21] also discuss limiting bandwidth to subsets of sensors based on proximity to target. In this thesis we are using multiple drones, with limited power resources communication between drones could be limited to minimal information and follow-up information on potential target spotting. Once a target is spotted and communicated drones without confirmation of target, outside of range, could limit their communication to only self-confirmed targets.

2.3 Visual Sensor

2.3.1 Target Detecting and Tracking

"Applied Techniques in Tracking Moving Targets in Marine Using Image Processing,"[2] samples different target tracking methods and how they apply to a marine environment. The problem addressed in this article is what are the different methods available and what are the positives and negatives for the different methods. Similar areas of improvement and existing problems are listed for this article also include environmental variables such as weather and time of day with reduced light. Although this article has not been cited by others, it does provide a good foundation to direct further investigations in which tracking methodologies would be best suited for my problem statement [2].

From J. Han et al [12] cited work we further look into attention shifting with L. Itti et al [13] look at mapping a scene based on saliency topology, ranking pixels based on the features that stand out against the scene. Three major feature categories were assessed; colors intensity and orientations. Each major feature is mapped by multiple subcategories. For example intensity feature category contains six unique maps. Four maps are based on the color intensities of red, blue, green and yellow; two additional maps are created for color intensity contrasts of neighboring pixels between opposite colors visually observed in biological vision of mammals, such as red/green and yellow/blue. Although L. Itti et al [13] takes note of color opponency from mammals, utilization of manufactured/fabricated opponency within the IR spectrum based on animals with some visual range in the IR spectrum may have some benefit towards this thesis. 36 other maps are combined after a map normalization operator is applied. This operator enables the system to suppress maps with multiple points of interest in favor of maps with low numbers. Theory behind this normalization is that there may be unique features identified by few maps which if not normalized would get washed out by maps with a large number of points of interest. In testing the pattern analysis showed promise, but L. Itti et al [13] acknowledge shortcomings of the analysis if the object features are not properly represented in the feature maps. With the foreknowledge of the object we wish to detect for this thesis, the simple approach shown may be able to be modified to have multiple maps that would be assessing multiple unique features of the given object.

2.3.2 Camera GPS Fusion Target Localization

G B Chatterji et al. [22] used sensor fusion to aid in aircraft landing. These sensors include cameras, altimeter and GPS to estimate the aircraft's position. Their calculated approach to estimating the aircraft’s location based on identifying lighting features with the camera and processing it with known parameters of the camera and mounting location on the physical aircraft and comparing it to a predicted light image based on a known light model. Their calculations and approach can be repurposed for target tracking and identification for this thesis. For future works expanding on this thesis these approaches could be leveraged. With multiple drone if a target is acquired the reverse process of G B Chatterji et al. [22] can be used to estimate the IR target's location. This location can then be transmitted to other drones. If receiving drone's position and camera range allow, they can estimate target's location on their image and confirm its presence.

3. System Design

This section will discuss the design process and prototype system developed. The first section (3.1) will outline the devices which make up the system, the function of each, and the physical characteristics. The second section (3.2) will discuss the hardware selection process and the final selections. The software (3.3) utilized in this thesis is discussed in the final section.

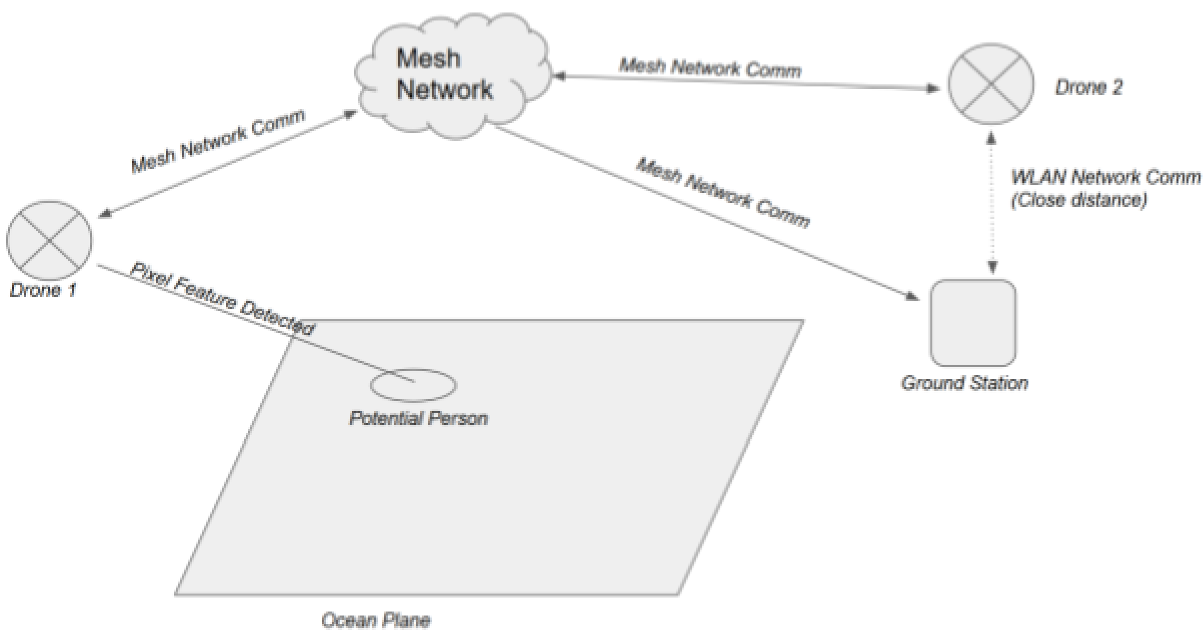

Figure 3-1: System Diagram

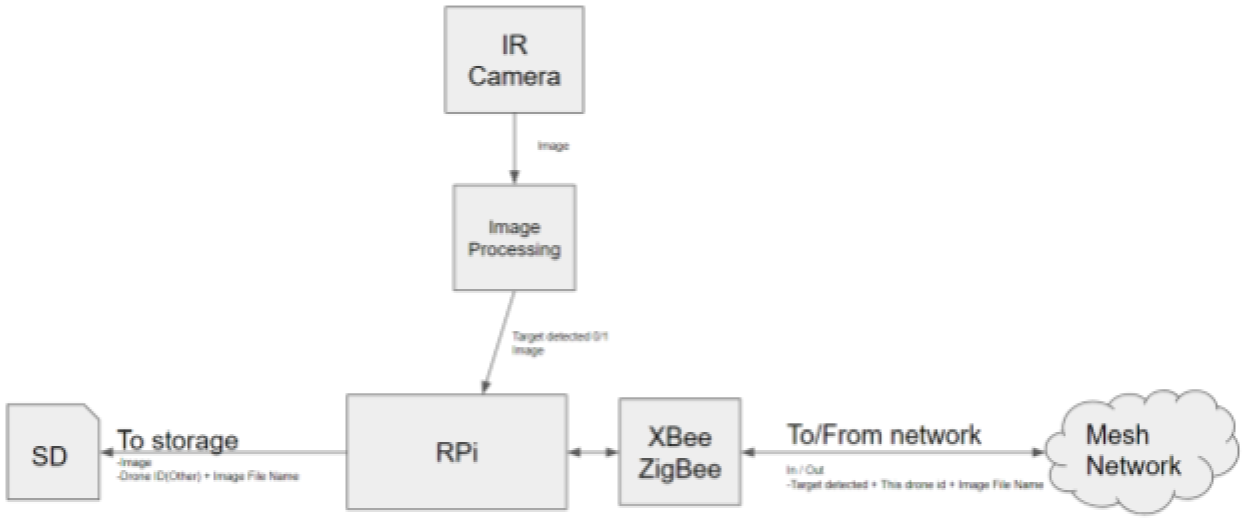

Figure 3-2: Image Capture Device Diagram

3.1 System Devices

The system developed for this project consists of two types of devices, each acting as a node in a mesh network. The first and primary device is for image capturing (3.1.1). The system is expandable and is intended to have multiple image capturing devices, each being an additional node on the network. The second device is a ground station (3.1.2). This device is optional, but improves the ease of setting up the image capturing devices and accessing information on the ground either prior to or after a search.

3.1.1 Image Capturing Device

The image capturing device is intended to be mounted to an unmanned aerial vehicle and transported across a search area at altitude. This device is capable of capturing infrared images and processing them to detect infrared emitting objects in the water. In the event that an object is detected an alert will be sent through the mesh network. This alert message will include the devices unique serial number and the image file name. The image file itself is stored in onboard memory for later extraction at the ground station. Each image capturing device will monitor the network and record any transmitted messages regarding object detection. This will allow for each drone to have a full list of objects detected by the system, as well as information regarding which device detected them so that the images can be pulled from the select devices. If a device is lost, this information can be used to determine if the device's search area contained any items of interest. If there were items of interest other devices could be reallocated to that search area.

3.1.1.1 Enclosure Design

The enclosure is a 3D printed box with a lid held on by bolts. An isometric view of the design is shown in A8 with photographs of the build in A9. The FLIR Lepton sensor is mounted to the lid to give room for the header wires to be routed to the pins on the Raspberry Pi. With the location of the USB and power ports on the Raspberry Pi, 90 degree cords and a low profile adapter needed to be used in order to fit in a smaller enclosure. The routing of the cables in this way allowed for space for the XBee ZigBee to be mounted on one of the box walls. Finally, the Raspberry Pi was stack mounted with the battery pack as intended by the manufacturer. Improvements that could be made to this box design would be access ports for turning on and off the battery pack and recharging. Currently, the enclosure needs to be opened and the Raspberry Pi removed to access these ports.

3.1.1.2 Thermal Concerns

The initial design of the enclosure did not offer anything in the way of thermal management. None of the boards or chips had any heat sinks and, as the box was completely enclosed except for a penetration for the FLIR lens, there would be little to no air movement across these chips. Without knowing the exact thermal properties of the components and materials used to construct the device, a test was devised to measure the reactions of the system under load. The enclosure was sealed and placed in a scene where it would regularly witness objects it would be expected to report. A software process (A1) was created for this test. This process would sample the Raspberry Pi's processor temperature through integrated sensors and write the data to storage. This process along with the standard OS (Operating System) was run for the first 24 minutes of the test. At that point all processes described in the software section were run and left running on battery power until the 1 hour mark. The chart in Figure 3.1-1 was generated from this data. The temperature during the first part of the test was stable around 43C. The spike at the 24 minute mark is attributed to the increased processor load of the additional programs running. This thermal spike stabilized around 50C for the remainder of the test. During this test approximately 2000 images were captured, processed and stored. The conclusion drawn from this test is that the boards and chips selected will not put off enough heat during the processes that we intend to run to justify the need for an improved thermal management design. For future designs if the components are consolidated onto single PCBs thermal management would need to be readdressed.

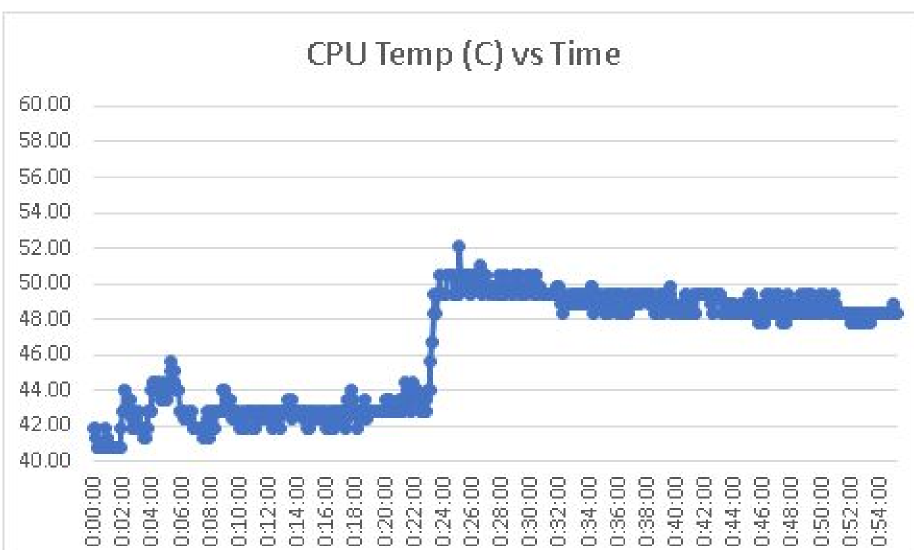

Figure 3.1-1

3.1.2 Ground Station

The ground station in this project provides two key processes for this system. First, it acts as an additional node in the mesh network system similar to any other node, with the exception that it does not capture or process images. It will only monitor the network for reported objects and record the sightings and serial number of the detection device. Secondly it allows for interaction with each of the system devices when they are within the enclosures. The ground station cannot power up the devices as there is a hardware switch. The ground station can access the operating system's interface without removing the device from the enclosure or needing to directly connect peripherals such as a monitor, keyboard and mouse. From this interface all the functionality is accessible. Programs and processes can be executed or modified, and live feeds from the infrared camera can be accessed. The file storage on the device can be accessed and all records and saved images can be transferred to the ground station. Programs used for this interface are outlined in the Program UI Section (3.3).

3.2 Hardware

This section will discuss the hardware selection process and the final outcome of those selections in developing the system.

3.2.1 Computer / Processor

For this project the Raspberry Pi with Rasbian OS [38] was selected. Although a bit bulky, the Raspberry Pi offers decent processing capabilities while being flexible with connected devices and having a wide range of available libraries and resources. During project execution the Raspberry Pi was able to be adapted to work with various camera types without much modification aside from locating the appropriate libraries for that communication scheme. Alternative selections would have been an Arduino or other commonly available ARM Cortex based microcontrollers, but, as previously stated, the libraries and processing capabilities steered the decision to use the Raspberry Pi. Additionally, the communication scheme library used by the FLIR Lepton was available [37] and the supplier of the board included documentation specifically for Raspberry Pi project setup [36].

3.2.2 Mesh Network Adapter

Communication between devices within the system requiring a main router / hub would have been cumbersome. Mesh networks would not require a router and offered the benefit of being dynamic. If they are put in an environment where nodes may disconnect or be added, the network would adapt. Digikey offered a very good kit [42] which offered an assortment of XBee Zigbee [41] chipsets. The standard and pro were tested and, due to the increased range capabilities of the Zigbee pro, it was included in the system as the mesh network adapter. No special libraries were required other than the base Raspbian libraries for serial. Described further in software mesh network section 3.3.2, it required very little in terms of software, but has ranges upwards of 2 miles with line of sight.

3.2.3 Infrared Camera

A large portion of this project relied on the selection of an appropriate camera. A few inexpensive near-infrared cameras were tested along with a mid-infrared camera. Ultimately, due to the drawbacks of near-infrared described in section 3.2.3.1, the selection of the mid-infrared FLIR Lepton [36] was made. This camera offered a simple setup with readily available libraries for communication as well an appropriate infrared spectrum for this systems application. Although the FLIR Lepton does have a less desirable resolution of 80x60 [36], it captures near to mid range images well. Other drawbacks of detecting far objects with low resolution imaging are discussed in section 3.2.3.3. It should be noted that since the final hardware selection for this project was completed, there have been improvements with the commercially available sensors. Some of these sensors have been tested and observations made in section 3.2.3.2 (FLIR One and Seek Thermal Observations).

3.2.3.1 Near versus Mid-Infrared Camera

Most common and cheap infrared sensors on the market for low light imaging or night vision with high resolution are near infrared. This band, despite its shared name, is not related to the frequencies emitted due to temperature. This range of EMR, just outside the band of visible light, shares many of the same characteristics of visible light spectrum, such as relying on a source to illuminate objects. However, since the illumination source is beyond the human visual spectrum it would still be considered night vision. The image in Figure 3.2.3.1-1 was taken at 75' with a mid-infrared. A similar image was captured with near-infrared, but despite the relatively close range the near infrared camera with its standard illumination source was not able to discern any objects in the scene. The resulting image was a uniform low level across all pixels.

Figure 3.2.2.1-1: FLIR Lepton 80x60 NIR - Person in wooded area.

Theoretically, a high powered illumination source could be used to illuminate targets at further ranges. This poses two issues: first, this illumination source would have similar power requirements to using the visual spectrum and normal cameras, so it would offer no benefit for search and rescue. The only unique feature is the illumination would be undetectable to the human eye. However, this is also the second downside. The human pupil does not respond to infrared and in the dark would be fully enlarged to take in the most light and give the best visibly. This would leave the eye vulnerable to damage as the power of the illumination would not be mitigated and the person may not react and gaze away like they would for a similar high powered spotlight.

Chart 3.2.3.1-1 shows the spectrum usage of various cameras researched for this project. It should be noted that the military grade FLIR has closer spectrum coverage to the sensor chosen for this thesis, the major difference being resolution and optics. Surprisingly the military FLIR Star SAFIRE III does not have an extreme resolution with only 640x512 [43] but instead relies on specially designed high quality magnification optics with a 71x zoom ratio [43]. Although similar optics for photography may not seem out of the ordinary, the unique characteristics of the infrared spectrum drive up the costs of similar quality infrared optics.

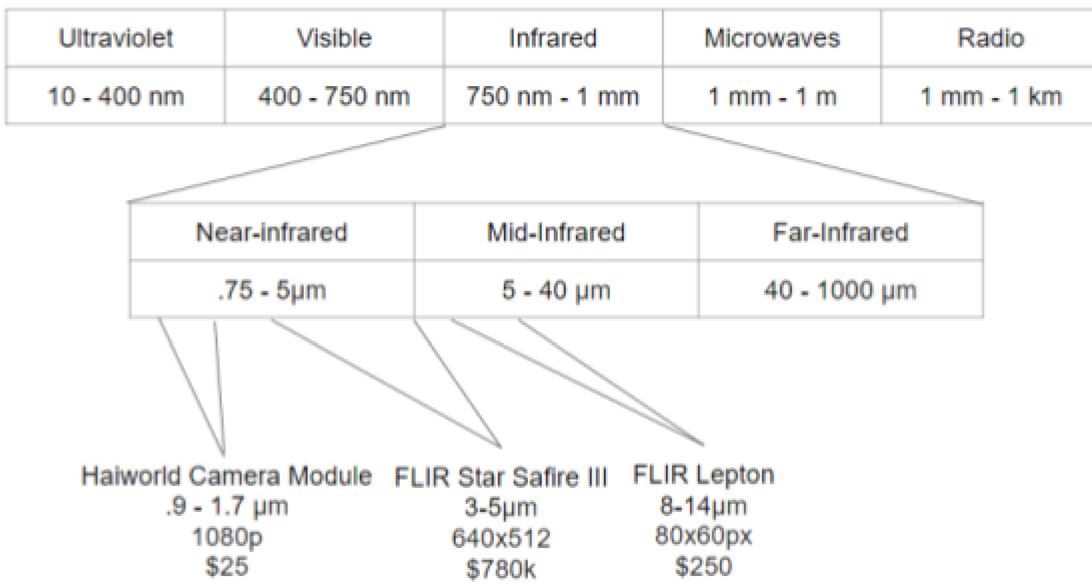

Chart 3.2.3.1-1: Infrared spectrum sensor coverage of available sensors

3.2.3.2 FLIR One and Seek Thermal Observations

Since the original build of this project, FLIR has released a new version of their infrared camera. Additionally they have a product which integrates their camera to be used as a smartphone attachment. This product, along with third party product Seek Thermal, were purchased and tested since both products have the higher resolution camera. One interesting feature of the FLIR One is the existence and fusion of two sensors with their proprietary FLIR MSX (Multi-Spectral Dynamic Imaging). The product has both a visual spectrum camera and the mid-infrared sensor. Reviewing the captured images, the visual spectrum camera has object edge detection which is then overlaid on the thermal output. Although the software does not have object detection and reporting, the overlaid edges allow for the user, or more specifically the user's brain, to make associations between the displayed edges and objects in their environment. This works very well in well lit and close range scenes, but without illumination the visual spectrum camera offers no impact on the product's displayed image output.

3.2.3.3 Issues observed with the use of infrared imaging

By far, the biggest hindrance to this project has been the unique characteristics of the infrared spectrum compared to normal visible light. The largest challenge was the IR spectrum's inability to penetrate certain objects and materials. Water, for example, absorbs a very large percentage of the EMR spectrum belonging to infrared. As seen in Figure 3.2.3.2-1 ,although the water blocks very little of the visual spectrum, the hand submerged in water is nearly invisible to the infrared imaging sensor. Even when removed from the water the dampness on the skin reduces the magnitude of the detected IR. Both of these issues hinder detecting people in water greatly. People were approximated to be 2x1 pixels for an average sized person at 100' given the resolution of the IR sensor. Given the attributes of water, however, if the target is swimming the detectable surface is reduced from the entire body to just the head. Additionally, if the head is damp it would reduce the heat signature further. From a distance of 100' the visible surface of a head emitting infrared would be less than the size of a pixel, thus, the magnitude of the signal would be averaged with its surroundings to get the value of the individual pixel.

Similar to water there are other materials which IR does not penetrate, but may be commonly transparent in the visual spectrum. Glass, for example, typically blocks all infrared. The Figure 3.2.3.2-1 shows the blocking of IR by a glass bowl. The first image shows a hand in the bowl from the side through the glass and then the top down view shows a hand partially submerged in water.

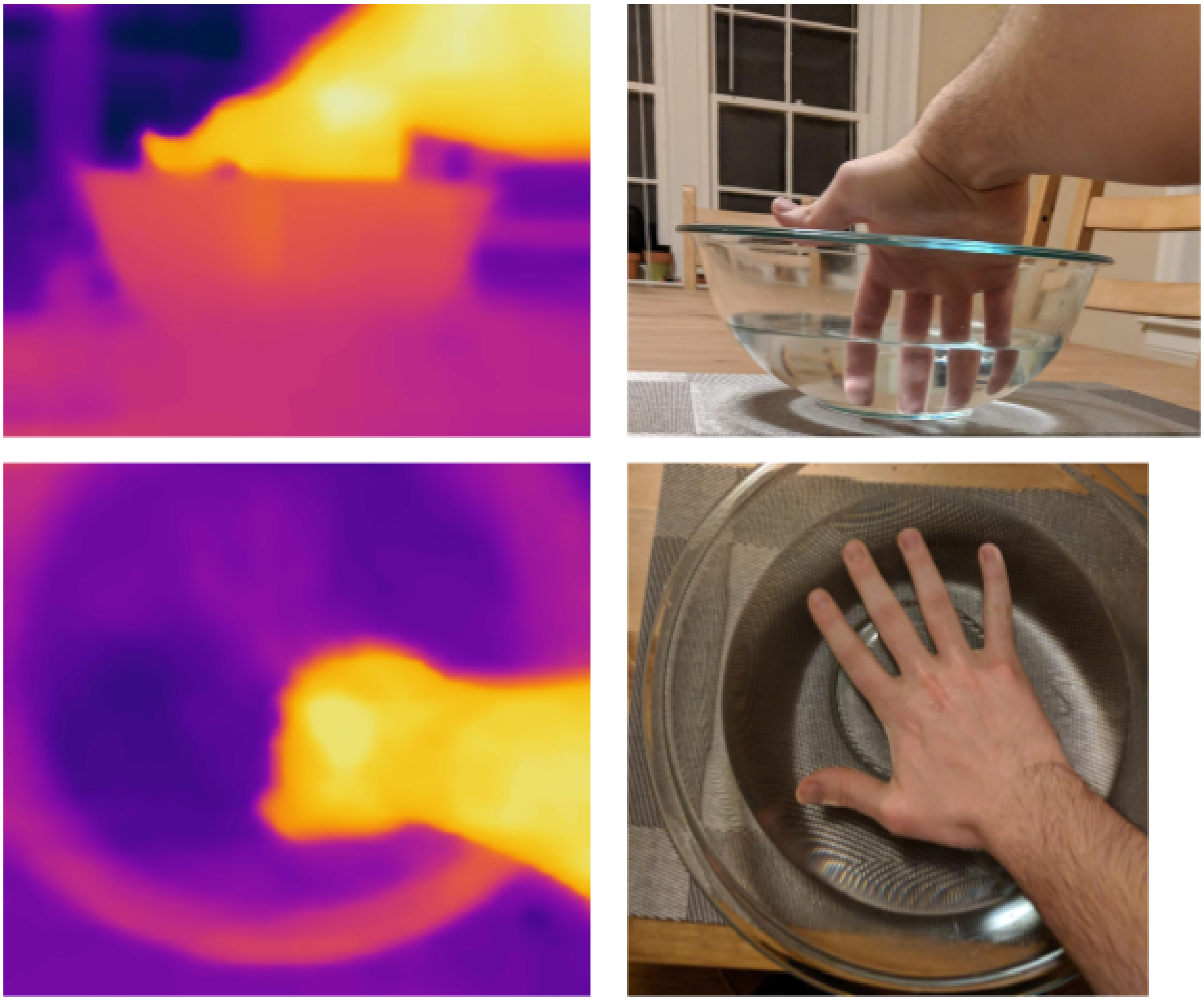

Figure 3.2.3.2-1: Near Infrared Blocked by glass and water
FLIR One 160x120 (Left)
Google Pixel 2 3024x4032(Right)

Consumer photography lens are another commonly transparent material that blocks IR. With the low resolution of IR cameras on the market magnification would greatly benefit target detection. This can also be seen in the price of high end FLIR products which contain multiplication optics as they not only need to be high quality, but also need to be specially made to not block the infrared spectrum.

3.2.4 Power Storage

For this project, the power source required mobility, so battery power storage was considered. To simplify the design tasks, an off the self-expansion board battery pack for the Raspberry Pi was looked into. Ultimately, the MarkerFocus Raspberry Pi Battery Pack was selected. According to the MarkerFocus spec sheet [35] this 3.7V 3800mAH lithium battery pack can provide the 5.1V, 2.5A power source required [34]. Based on Figure 3.2.4-1 a fully charged battery with max power draw will last for over one hour. However, independent benchmarking from Raspberry Pi Dramble [33] offers a better understanding of the power consumption of a Raspberry Pi 3B with a max power consumption on an overloaded CPU at 3.7W, which would have the battery lasting for 3.8 hours.

$$Hours = \frac{Battery\ WattHours}{Source\ Watts} = \frac{Volts(3.7) * AmpHours\ (3.8)}{Volts\ (5.1) * Amps\ (2.5)} = 1.1\ Hours$$

Figure 3.2.4-1

3.3 Software

3.3.1 Computer Vision

In this section we will review all the code associated with image capturing and processing. Code described in this section is intended to be run on the Raspberry Pi Image capturing device.

3.3.1.1 Image Capture

Image capture primarily relies on the ‘pylepton_capture’ [37] library import. This is a third party program recommended by the FLIR Lepton supplier in their setup guide [36]. The main function in Appendix A.4 will call this function in the main loop once every second and stores the newly captured image in a temporary file until it is processed.

3.3.1.2 Image Processing

The main process outlined in A.4 also handles the task of processing all images. It carries a value pixelmap_base[ ] which is to represent the expected baseline scene. Each image builds on this baseline scene. After every new image a new baseline scene is calculated using a weighted average of 90% the previous scene and 10% of the new image. Prior to calculating a new baseline, the captured image is compared to the current baseline. The process looks for the pixel in the captured image that has the largest difference over the baseline scene as calculated from equation 3.3-1.

$$Pixel\ Difference\ = (Captured\ Pixel)_{[x,y]}\ -\ (Baseline\ Pixel)_{[x,y]}$$

Equation 3.3-1

It should be noted that this is not the largest absolute change, because sudden low value pixels, representing low thermal energy, in the captured image would not be of interest. Once this pixel is found the difference is compared to a set threshold. Currently in appendix A4 the threshold is set to '20' which is based on a scale of each pixel having a value from 0 to 255. The scale of the FLIR Lepton is from -10C to 140C [44], so a difference of 20 from the pixel’s value is approximately +-11.75C. All images containing a pixel which pass this threshold are stored locally with unique names. These files can be accessed at a later time by the ground station, outlined in section 3.1.2, through the use of the Remote Access (3.4.1)

3.3.2 Mesh Network

When a new image is processed in the main process A.4 and a potential object is detected in that scene, an alert is sent out over the mesh network. The message which is to be sent will include the device's CPU serial number which is unique to each Raspberry Pi and the image file name stored to that device. Subroutine XBee_Send.py A.2 was created to handle transmissions send to the network and XBee_Receive.py A.5 handles transmissions received from network. XBee is connected via USB and communicates over it via standard serial communications. To handle collecting the serial number information for the device, the getserial.py function accesses the cpuinfo file and returns the serial number value.

3.4 Ground Station

The ground station is an optional device within the system outlined in section 3.1.2 which is capable of remote access to the image capturing devices by the process described in 3.4.1. This will allow the ground station to display the user interfaces of processes of 3.4.2.

3.4.1 Remote Access

For remote access of the image capturing devices from the ground station we used the LAN based remote access program VNC. This install is included in the base Rasbian operating system. When your ground station and image capturing devices are connected to the same LAN, the image capturing device can be access via VNC Viewer. The peripherals of the ground station will then work for the image capturing device. Processes can be started, ended, and files transferred.

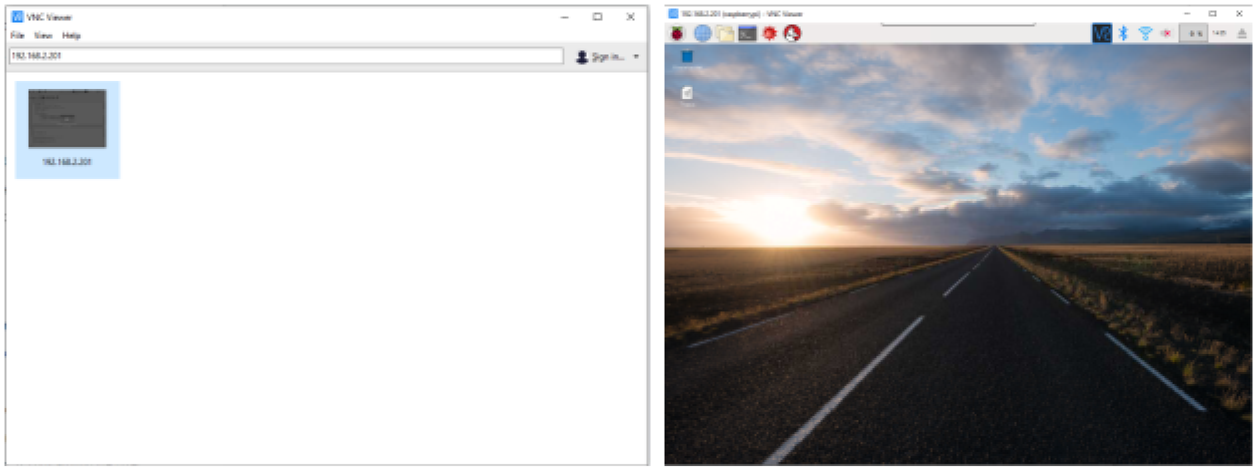

Figure 3.3.1-1: VNC Connection between ground station and image capturing device over WLAN

3.4.2 Program UI

During normal operation of the image capturing device there are no peripherals connected for a user interface to be displayed, but when the ground station is connected to a device the interfaced shown in Figure 3.3.1-1 can be utilized.

The main process A.4 also has a user interface. Shown in Figure 3.3.2-1 the interface displays the baseline scene as described in the software image processing section (3.3.2.1). Additionally the last captured image with an object detected is also displayed. This image has a 5x5 red box drawn around the object. The pixel which the box is centered around is that which has the largest difference from the baseline scene as outlined in Section 3.3.1.2 and Equation 3.3-1. Additionally there are console outputs from the main process A.4 these include: when an image is stored and when there is a received transmission from the network (A5).

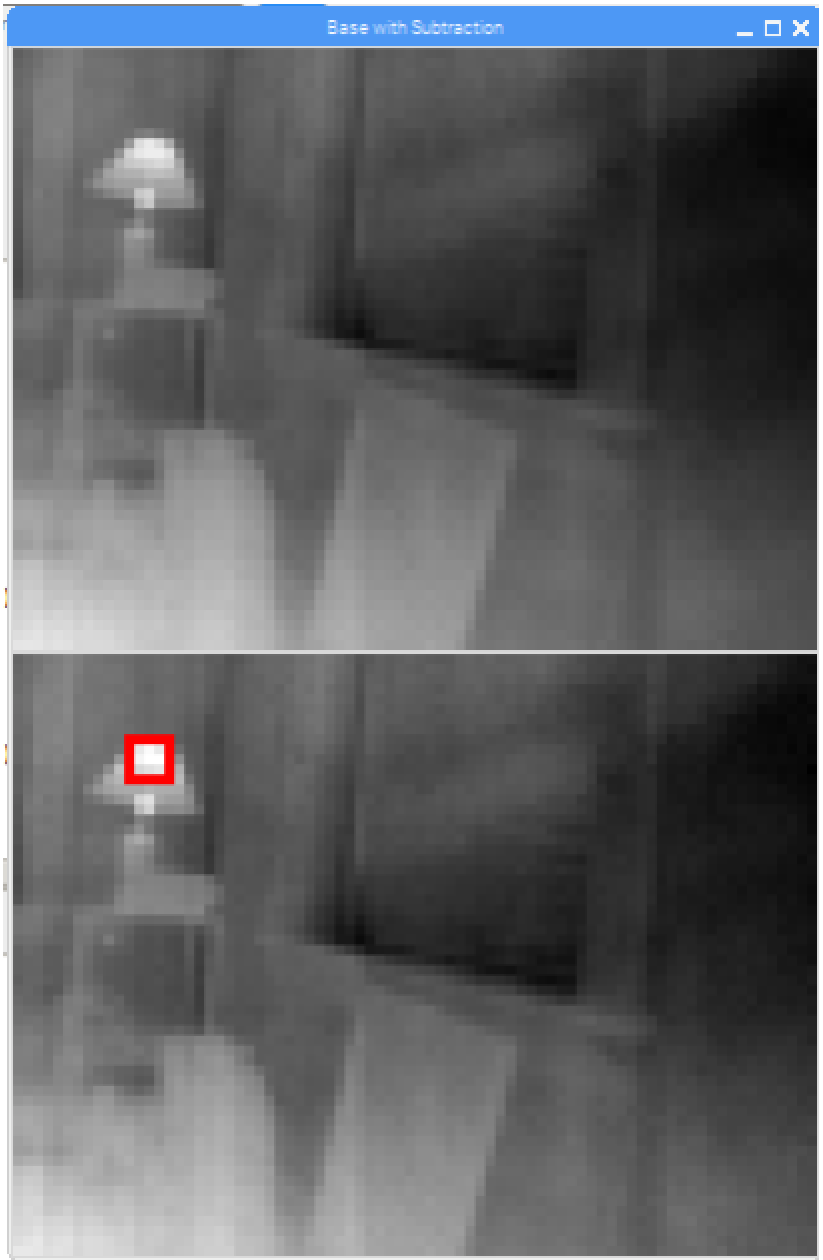

Figure 3.3.2-1: Main Process

Instead of running the main process (A.4) two other processes can be run to provide direct outputs on command from the IR camera. A.6 when run has an interface which is shown in

Figure 3.3.2-2. The 'snap' button will execute the subroutine described in the previous section for image capture. The captured image will then be displayed in the interface. Alternatively the FLIR Lepton capture software from [37] contains a program which provides a live video feed from the IR camera (Figure 3.3.2-3).

Figure 3.3.2-2: Service_Test.py

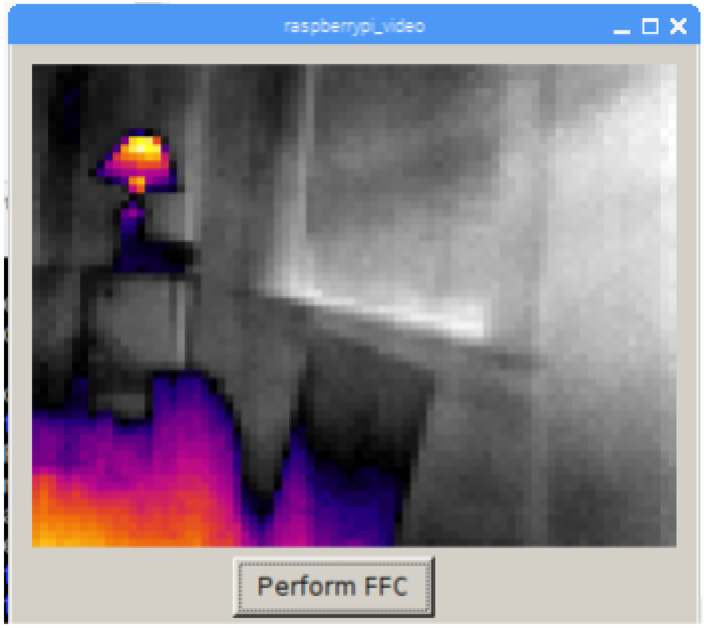

Figure 3.3.2-3: raspberrypi_video

3.5 Cost

When compared to the cost of current search and rescue vehicles outlined in the Total Addressable Market (1.3.4) the components of our system are much lower cost. Table 3.5-1 lists the price of the selected components for a single image capturing device which totals $603.52. The system as described in Section 3 would require multiple image capturing devices along with drones capable of lifting a device (~$1500 ea). Additionally the optional ground station can be run from any windows operating computer (~$800). For the cost of a single month of upkeep on an S-92 search and rescue helicopter ($75,000), a system of 35 prototypes devices including drones and ground station could be built.

Raspberry Pi	$36.95
FLIR - Lepton Chip	$259.00
FLIR - Lepton Board	$39.99
XBee Zigbee Pro Chip	$39.45
XBee Zigbee Board	$13.99
Battery	$14.99
Enclosure	$179.15
Misc Parts (Wire, fasteners)	$20.00
Total	**$603.52**

Table 3.5-1

4. Prototype Testing

A prototype system was constructed by the specifications outlined in Section 3. Of the devices listed in Section 3.1 two devices were constructed; one image capturing device (3.1.1) and one ground station (3.1.2).

The prototype was tested by having the image capturing device placed in a third story window facing a courtyard and street. A model of the scene is shown in Figure 3.4-3, distance from the device to the street is ~115’ and to the courtyard is ~60’. The ground station was setup to receive messages over the mesh network as well as access the program’s UI as it ran. The system ran for 30 min and five staged instances of people walking into the scene were performed. The system was able to detect all five staged instances of people walking (100% from sample). An example output from a staged event is shown in Figure 3.4-1. The images prior to the detection of each of the five staged instances were reviewed post-test. Four of the instances detected the person on the first available image. The fifth instance was detected on the second image as the first image was a partial capture where the person was not fully in frame. Given an image capture rate (code in appendix A.4) of 1 second and assuming objects entered the image frame in a uniform distribution over time the test averaged .6 seconds to detect a person. There were some noted anomalies. The system erroneously detected heat fluctuations near the chimney of the building Figure 3.4-2. In addition, the system would detect the occasional passing car's exhaust. These situations were noted, but not included in the results as failures as they technically meet the criteria of items of interest, but they were moving significantly faster and are hotter than the intended targets. During the test 140 detection flags were raised out of 2000 processed images. The post test image review noted 21 of the 140 (15%) flags were erroneous.

Figure 3.4-1: Person Walking (~100')

Figure 3.4-2: Erroneous Object

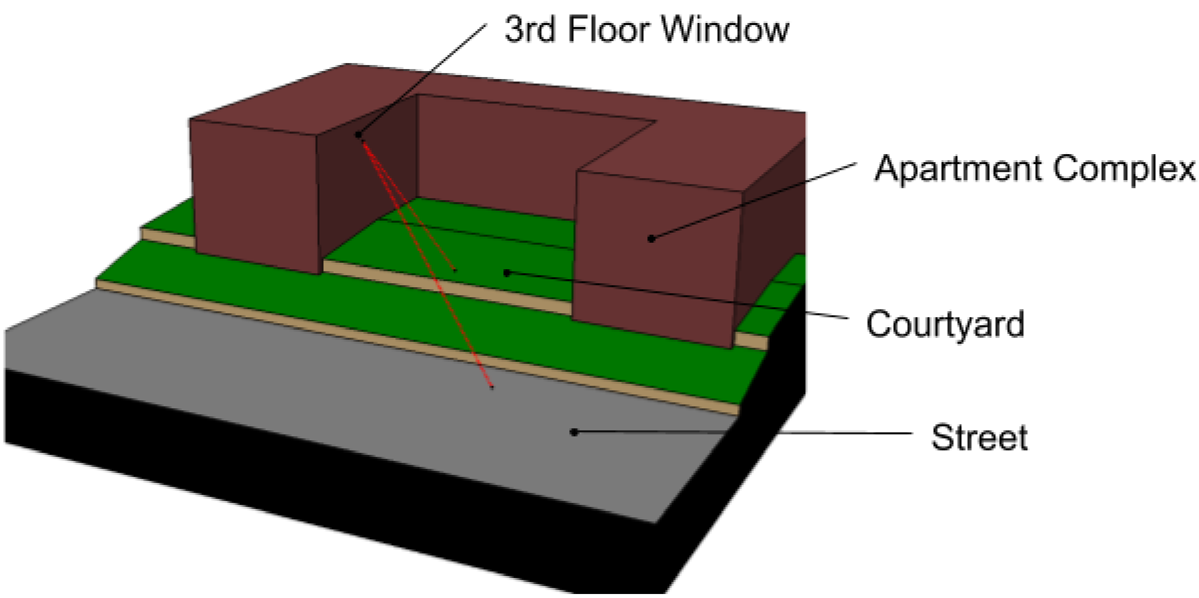

Figure 3.4-3: Test Scene

5. Conclusion

The mid-infrared range is an excellent spectrum in which to detect people. In medium ranges and controlled conditions the system was shown during experimentation in Section 4 to be able to detect people with acceptable success rates. This along with the greatly reduced costs compared to existing systems outlined in Section 3.5 shows that the commercially available sensors are capable of similar forms of people detection. However, the low resolution of cameras available commercially today hinder using them at greater distances >100’. These ranges don’t support typical operating altitudes of aerial vehicles which, in order to cover larger areas, will typically fly higher than 100’. With the uniqueness of lens for infrared it is unlikely that the current commercially available products could be used to develop a realistic system without first solving some of the issues described and undertake improvements listed in section 6.

6. Future Designs

6.1 Device Design

There is room for device design improvement. Development boards were used for each of the individual components of the prototype design. This led to certain redundancies in design and inefficient use of physical space. If the design process were to proceed after testing the development board designs could be dropped and the system redesigned onto a single printed circuit board. This would reduce redundancy of things like the XBee Zigbee and the on board Raspberry Pi WiFi chip performing similar tasks. Although the Raspberry Pi is very versatile for a developer there are numerous features which are unused for this project. Two examples are the Ethernet port and the USB ports. If the design were to go to a PCB the USB ports, or at least the physical jacks, would be able to be eliminated.

6.2 Infrared Camera

As described in Section 3.2.3.2 (FLIR One and Seek Thermal Observations) there are already new cameras available on the market with higher resolution than the one used in this project. However the resolution increase is still not high enough for consistent detection of people at range. To further increase image clarity the operating altitude could be reduced and a non-fixed camera could be utilized. This would get the camera closer to the objects without reducing the breadth of each search path.

6.3 Software & GPS

With improved sensors and larger objects in scene more advanced image processing techniques could be used. The main disadvantage of the current software was the inputs available. There would also be benefits to adding additional sensors such as GPS. This data could be used to harness such techniques as stereo target localization as described in Section 2.1.

6.4 Additional System Usage

While the resolution criteria of current consumer grade sensors may not be able to detect small targets such as people from significant altitude, that is not to say that a larger target may not be able to be detected with the current available sensors. There is currently an ongoing issue of wildfires in the US West. They have adapted many emerging technologies into their firefighting methods including many early warning systems to detect fires. They typically react to a wild fire via containment method in which they create or use existing fire breaks. These fire breaks are areas which have had combustible materials removed to prevent the fires progression. However embers can ride winds and thermal rises from the fire and start smaller fires past these breaks. Although firefighters are constantly on the lookout for these jumps in order to put them out when they are small, manpower and sheer size of risk areas limit their capabilities. If an automated system similar to the one described in this thesis could be adapted to navigate a set path and detect these small fires it could provide benefit to the firefighting operations. Additionally with fires being such a large target the drones could follow the edge of the fire and report the information back to a central hub to provide live updates on the current status of the fire.

Figure 4.3-1: Fire shown with thermal imaging
Seeker Thermal 206x156 (Left)
Google Pixel 2 3024x4032(Right)

As you can see in the Figure 4.3-1 images fire under an infrared sensor has a very extreme signal and is easily detectable due to the large temperature difference between it and it's surroundings. Even at the beginning and ending stages the thermal difference is great as shown in Figure 4.3-2 after the fire had died down the embers were still emitting a great amount of IR.

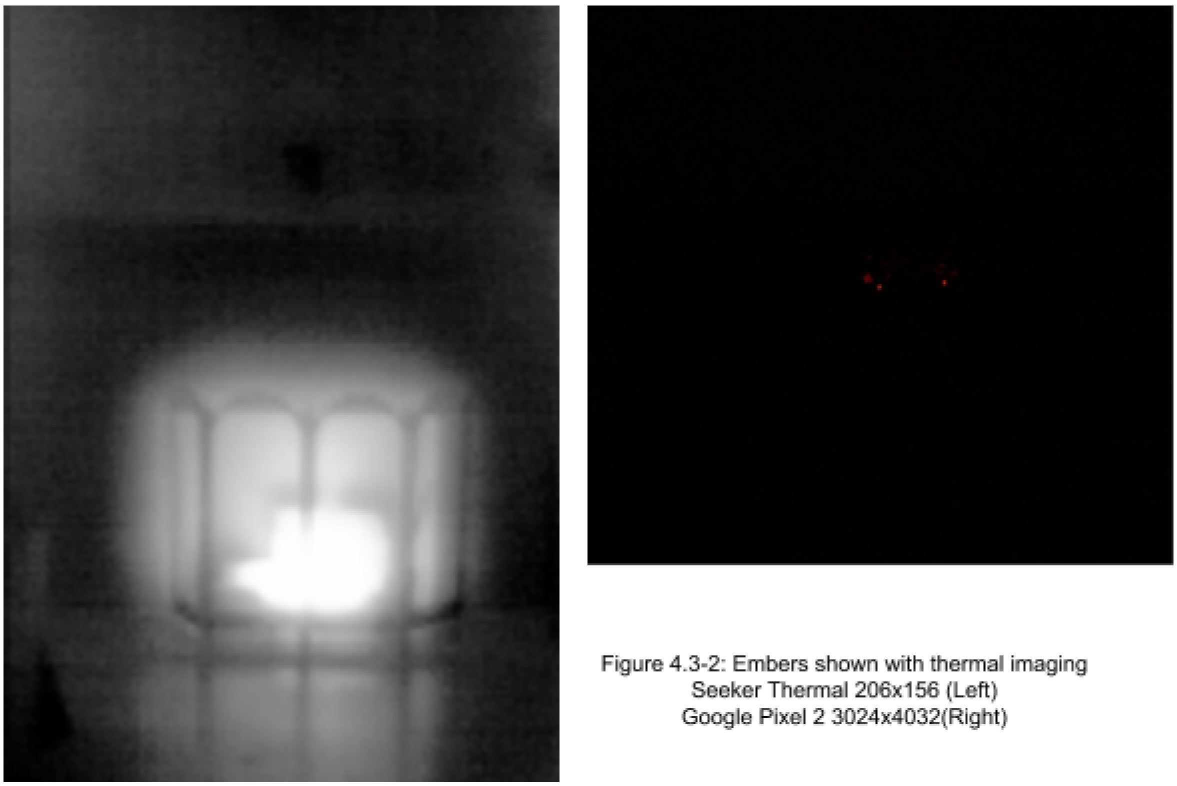

Figure 4.3-2: Embers shown with thermal imaging
Seeker Thermal 206x156 (Left)
Google Pixel 2 3024x4032(Right)

7 Literature cited

[1] R. J. D. Moore, “Vision Systems for Autonomous Aircraft Guidance”, Ph.D. thesis, Queensland Brain Institute, Univ. of Queensland, Brisbane, 2012

[2] H.K. Moghaddam and A.A.Y. Fard, "Applied Techniques in Tracking Moving Targets in Marine Using Image Processing", Open Journal of Marine Science, vol. 6, no. 04, pp. 524.

[3] S. Parameswaran et al, "Marine object detection in UAV full-motion video," Proc.Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications XI, vol. 907608, pp. 1-8, 2014.

[4] R. Amin, R. Gould, W. Hou, R. Arnone and Z. Lee, "Optical Algorithm for Cloud Shadow Detection Over Water," IEEE Transactions on Geoscience and Remote Sensing, vol. 51, no. 2, pp. 732-741, Feb. 2013. doi: 10.1109/TGRS.2012.2204267

[5] J. J. Simpson and J. R. Stitt, "A procedure for the detection and removal of cloud shadow from AVHRR data over land," IEEE Transactions on Geoscience and Remote Sensing, vol. 36, no. 3, pp. 880-897, May 1998. doi: 10.1109/36.673680

[6] K. L. Carder, P. Reinersman, and R. F. Chen, "AVIRIS calibration using the cloud shadow method," Proc. Summ. 4th Annu. JPL Airborne Geosci. Workshop, R. O. Green, Ed., 1992, vol. 1, pp. 26–28, JPL Publication 92-14.

[7] Z. P. Lee et al. "Water and bottom properties of a coastal environment derived from Hyperion data measured from the EO-1 spacecraft platform," J. Appl. Remote Sens., vol. 1, p. 011502, 2007.

[8] P. Reinersman, K. L. Carder, and F. R. Chen, "Satellite-sensor calibration verification with the cloud shadow method," Appl. Opt., vol. 37, no. 24, pp. 5541–5549, Aug. 1998.

[9] R. Amin, R. W. Gould, W. Hou, R. Arnone, Z. Lee, "Automated system and method for optical cloud shadow detection over water" US8509476 B2, Aug 13, 2013.

[10] Central Intelligence Agency, 2017. The World Factbook 2017. Available: https://www.cia.gov/library/publications/the-world-factbook/fields/2060.html

[11] X. Wang et al, "Infrared Small Target Detection via Nonnegativity-Constrained Variational Mode Decomposition," IEEE Geoscience and Remote Sensing Letters, vol. 14, (10), pp. 1700-1704, 2017. . DOI: 10.1109/LGRS.2017.2729512.

[12] J. Han et al, "A Robust Infrared Small Target Detection Algorithm Based on Human Visual System," IEEE Geoscience and Remote Sensing Letters, vol. 11, (12), pp. 2168-2172, 2014. . DOI: 10.1109/LGRS.2014.2323236.

[13] L. Itti, C. Koch and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, (11), pp. 1254-1259, 1998. . DOI: 10.1109/34.730558.

[14] Lijun Jiang et al, "Perceptual-based fusion of IR and visual images for human detection," in Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004. 2004, . DOI: 10.1109/ISIMP.2004.1434114.

[15] J. Q. Cui et al, "Drones for cooperative search and rescue in post-disaster situation," in 2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), 2015, . DOI: 10.1109/ICCIS.2015.7274615.

[16] S. Lee, D. Har and D. Kum, "Drone-assisted disaster management: Finding victims via infrared camera and lidar sensor fusion," in 2016 3rd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE), 2016, . DOI: 10.1109/APWC-on-CSE.2016.025.

[17] C. Alex and A. Vijaychandra, "Autonomous cloud based drone system for disaster response and mitigation," in 2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA), 2016, . DOI: 10.1109/RAHA.2016.7931889.

[18] J. Lee et al, "Constructing a reliable and fast recoverable network for drones," in 2016 IEEE International Conference on Communications (ICC), 2016, . DOI: 10.1109/ICC.2016.7511317.

[19] L. Lin et al, "A geographic mobility prediction routing protocol for ad hoc UAV network," in 2012 IEEE Globecom Workshops, 2012, . DOI: 10.1109/GLOCOMW.2012.6477824.

[20] J. Colorado et al, "Low-altitude autonomous drone navigation for landmine detection purposes," in 2017, . DOI: 10.1109/ICUAS.2017.7991303.

[21] E. Masazade and A. Kose, "A Proportional Time Allocation Algorithm to Transmit Binary Sensor Decisions for Target Tracking in a Wireless Sensor Network," IEEE Transactions on Signal Processing, vol. 66, (1), pp. 86-100, 2018. . DOI: 10.1109/TSP.2017.2759080.

[22] G. B. Chatterji, P. K. Menon and B. Sridhar, "GPS/machine vision navigation system for aircraft," IEEE Transactions on Aerospace and Electronic Systems, vol. 33, (3), pp. 1012-1025, 1997. . DOI: 10.1109/7.599326.

[23] F. Fooladgar et al, "Geometrical Analysis of Localization Error in Stereo Vision Systems," IEEE Sensors Journal, vol. 13, (11), pp. 4236-4246, 2013. . DOI: 10.1109/JSEN.2013.2264480.

[24] S. Yoo et al, "Empowering drones teamwork with airborne network," in 2017, . DOI: 10.1109/AINA.2017.101.

[25] C. G. Sărăcin, I. Dragoş and A. I. Chirilă, "Powering aerial surveillance drones," in 2017, . DOI: 10.1109/ATEE.2017.7905185.

[26] E. Mingkhwan and W. Khawsuk, "Digital image stabilization technique for fixed camera on small size drone," in 2017, . DOI: 10.1109/ACDT.2017.7886149.

[27] F. Königseder, W. Kemmetmüller and A. Kugi, "Attitude Estimation Using Redundant Inertial Measurement Units for the Control of a Camera Stabilization Platform," IEEE Transactions on Control Systems Technology, vol. 24, (5), pp. 1837-1844, 2016. . DOI: 10.1109/TCST.2015.2510324.

[28] F. Zhang, "Simultaneous Self-Calibration of Nonorthogonality and Nonlinearity of Cost-Effective Multiaxis Inertially Stabilized Gimbal Systems," IEEE Robotics and Automation Letters, vol. 3, (1), pp. 132-139, 2018. . DOI: 10.1109/LRA.2017.2734968.

[29] M. L. Psiaki, T. E. Humphreys and B. Stauffer, "GPS lies," IEEE Spectrum, vol. 53, (8), pp. 26-53, 2016. . DOI: 10.1109/MSPEC.2016.7524168.

[30] W. Lu et al, "Design and implementation of a gasoline-electric hybrid propulsion system for a micro triple tilt-rotor VTOL UAV," in 2017, . DOI: 10.1109/DDCLS.2017.8068112.

[31] IEEE, 2017. IEEE Code of Ethics. Available: https://www.ieee.org/about/corporate/governance/p7-8.html.

[32] Bureau of Transportation Statistics, 2017. Table 2-49: U.S. Coast Guard Search and Rescue Statistics, Fiscal Year 2013. Available: https://www.rita.dot.gov/bts/sites/rita.dot.gov.bts/files/publications/national_transportation_statistics/html/table_02_49.html

[33] “Power Consumption Benchmarks | Raspberry Pi Dramble.” [Online]. Available: https://www.pidramble.com/wiki/benchmarks/power-consumption. [Accessed: 05-Feb-2019].

[34] “Raspberry Pi 3 Model B+.” [Online]. Available: https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf [Accessed: 05-Feb-2019].

[35] “New version Raspberry Pi UPS Battery Pack-Document.pdf - Google Drive.” [Online]. Available: https://drive.google.com/file/d/12_UCy6u5iaznpgSu-4YFTY0w-D2REM_4/view. [Accessed: 05-Feb-2019].

[36] “FLIR Lepton Hookup Guide - learn.sparkfun.com.” [Online]. Available: https://learn.sparkfun.com/tutorials/flir-lepton-hookup-guide?_ga=2.88974616.2006017150.1549426746-298940717.1549426746. [Accessed: 05-Feb-2019].

[37] “Pure Engineering - LeptonModule - GitHub.com” [Online]. Available: https://github.com/groupgets/LeptonModule. [Accessed: 05-Feb-2019].

[38] “Raspbian for Raspberry Pi.” [Online]. Available: https://www.raspberrypi.org/downloads/raspbian/. [Accessed: 05-Feb-2019].

[39] “Seek Thermal Compact Specifications.” [Online]. Available: https://www.thermal.com/uploads/1/0/1/3/101388544/compact-sellsheet-usa_web.pdf. [Accessed: 05-Feb-2019].

[40] “FLIR One Specifications.” [Online]. Available: http://www.flirmedia.com/MMC/THG/Brochures/IND_036/IND_036_EN.pdf. [Accessed: 05-Feb-2019].

[41] “XBee/XBee-PRO S2C 802.15.4 Radio Frequency (RF) Module User Guide,” 2018. [Online]. Available: https://www.digi.com/resources/documentation/digidocs/pdfs/90001500.pdf. [Accessed: 05-Feb-2019].

[42] “Digi XBee ZigBee Mesh Kit,” 1996. [Online]. Available: https://www.digi.com/pdf/ds-xbee-zigbee-mesh-kit.pdf. [Accessed: 05-Feb-2019].

[43] “FLIR Star SAFIRE III Datasheet.” [Online]. Available: https://www.flir.com/globalassets/imported-assets/document/star-safire-iii-datasheet.pdf. [Accessed: 05-Feb-2019].

[44] “FLIR LEPTON® with Radiometry Datasheet.” [Online]. Available: https://cdn.sparkfun.com/assets/8/b/a/2/d/lepton-engineering-datasheet---with-radiometry.pdf. [Accessed: 05-Feb-2019].

[45] “Getting Your Raspberry Pi Serial Number Using Python” [Online]. Available: https://www.raspberrypi-spy.co.uk/2012/09/getting-your-raspberry-pi-serial-number-using-python [Accessed: 05-Feb-2019].

Appendix A

A.1 Temp_Monitor.py

```
import os
import time

def measure_temp():
        temp = os.popen("vcgencmd measure_temp").readline()
        return (temp.replace("temp=",""))



while True:
        file = open('/home/pi/Thesis/Output/temp_history.txt','a+')
        print(measure_temp())
        file.write(measure_temp())
        file.close()
        time.sleep(10)
```

A.2 XBee_Send.py

```
import serial

def XBee_Send(xbee_out):
    ser = serial.Serial('/dev/ttyUSB0', 9600,timeout=.5)
    string=xbee_out
    ser.write(string.encode())
```

A.3 getserial.py [45]

```
def getserial():
    #Extract serial from cpuinfo file
    #credit stackexchange Raspberry Spy

    cpuserial = "0000000000000000"
    try:
        f=open('/proc/cpuinfo','r')
        for line in f:
            if line[0:6]=='Serial':
                cpuserial = line[10:26]
        f.close()
    except:
        cpuserial = "ERROR000000000"

    return cpuserial
```

A.4 Test_Image_Rolling_Average_w_Capture.py

```
from PIL import Image, ImageTk, ImageOps, ImageDraw
import tkinter as tk
import subprocess,time
from XBee_Send import XBee_Send
from getserial import getserial

root = tk.Tk()
root.title("Base with Subtraction")

im_base = Image.open("/home/pi/Thesis/OutputWorkspace/Latest.png")
pixelmap_base = im_base.load()

photo1 =
ImageTk.PhotoImage(im_base.resize((im_base.size[0]*5,im_base.size[1]*5)))
l1 = tk.Label(root, image = photo1)
```

```
l1.pack()
l1.photo = photo1
#root.mainloop()

photo2 =
ImageTk.PhotoImage(im_base.resize((im_base.size[0]*5,im_base.size[1]*5)))
l2 = tk.Label(root, image = photo2)
l2.pack()
l2.photo = photo2
print("Main Loop Start")
for a in range(0,10000):
    print("Loop ",a)
    output_file = "OutputWorkspace/Latest.png"
    subprocess.run(["pylepton_capture", output_file])

    image_path = "/home/pi/Thesis/OutputWorkspace/Latest.png"
    im = Image.open(image_path)
    pixelmap = im.load()

    max_pix = (0,0,0)

    for x in range(0,80):
        for y in range(0,60):
            if(pixelmap[x,y]-pixelmap_base[x,y] >0):
                if(pixelmap[x,y]-pixelmap_base[x,y]>max_pix[0]):
                    max_pix = (pixelmap[x,y]-pixelmap_base[x,y],x,y)
            pixelmap_base[x,y] = int(pixelmap_base[x,y]*.75 +
pixelmap[x,y]*.25)
                #pixelmap[x,y]=(pixelmap[x,y]-pixelmap_base[x,y])*2
            #else:
                #pixelmap[x,y]=0

    #print(max_pix)
    if(max_pix[0]>20 and a>10):
        print("Item Found in: /OutputWorkspace/",str(a),".png")
        XBee_Send(getserial()+" " + str(a) +".png\n")

        im = im.convert('RGB')
        draw = ImageDraw.Draw(im)
        draw.rectangle(((max_pix[1]-2, max_pix[2]-2), (max_pix[1]+2,
max_pix[2]+2)), outline="red")
        im.save("/home/pi/Thesis/OutputWorkspace/"+str(a)+".png")

        photo1 =
ImageTk.PhotoImage(im_base.resize((im.size[0]*5,im.size[1]*5)))
        l1.config(image = photo1)
        #l1.pack()
        l1.photo = photo1

        photo2 =
```

```
ImageTk.PhotoImage(im.resize((im.size[0]*5,im.size[1]*5)))
        l2.config(image = photo2)
        #l2.pack()
        l2.photo = photo2

        root.update()


    time.sleep(1)
```

A.5 XBee_Receive.py

```
import serial

ser = serial.Serial('/dev/ttyUSB0', 9600,timeout=.5)

while True:
    incoming=ser.readline()
    print(incoming)
    file = open('/home/pi/Thesis/Output/XBee_Record.txt','a+')
    file.write(incoming)
    file.close()
```

A.6 Service_Test.py

```
import FLIR_Snap
from PIL import Image, ImageTk
import tkinter as tk

def snapbutton_press(root,photolabel):
    FLIR_Snap.Snap()
    im = Image.open("/home/pi/Thesis/Output/Latest.png")
    photo1 = ImageTk.PhotoImage(im.resize((im.size[0]*5,im.size[1]*5)))
    photolabel.config(image = photo1)
    photolabel.photo = photo1
    root.update()



root = tk.Tk()
root.title("UI Demo")

frame = tk.Frame(root)
frame.pack()
```

```
im = Image.open("/home/pi/Thesis/Output/Latest.png")
photo1 = ImageTk.PhotoImage(im.resize((im.size[0]*1,im.size[1]*1)))
photolabel = tk.Label(frame, image = photo1)
photolabel.pack(side=tk.TOP)
photolabel.photo = photo1

snapbutton = tk.Button(frame,
                   text="Snap",
                   fg="green",
                   command=lambda: snapbutton_press(root,photolabel))
snapbutton.pack(side=tk.BOTTOM)

qbutton = tk.Button(frame,
                   text="QUIT",
                   fg="red",
                   command=root.destroy)
qbutton.pack(side=tk.BOTTOM)

root.mainloop()
```

A.7 FLIR_Snap.py

```
import subprocess, time
from PIL import Image, ImageTk
from datetime import date

def Snap():
    output_file = "Output/" + date.today().isoformat()+ " " +
str(time.time())+ ".png"
    print(output_file," created.")
    subprocess.run(["pylepton_capture", output_file])
    subprocess.run(["pylepton_capture", "Output/Latest.png"])

    #im = Image.open(output_file)
    #im.show()

Snap()
```

A.8 Enclosure isometric view

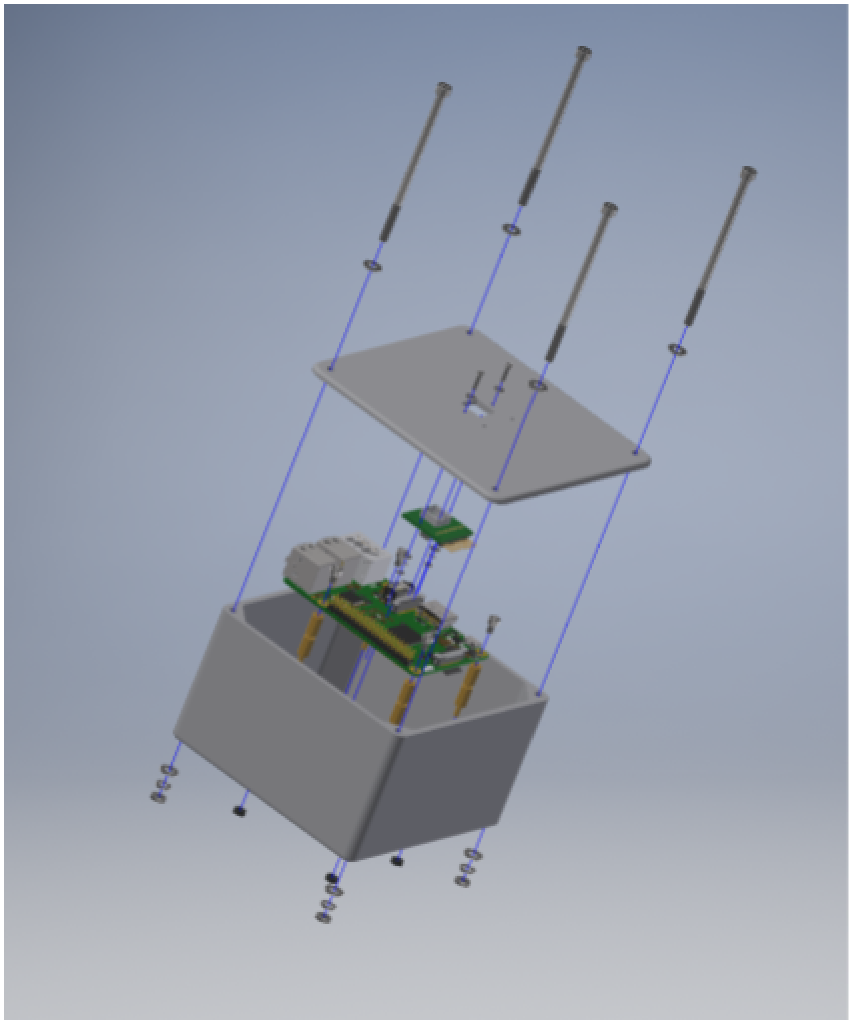

Figure A.8-1: Enclosure Exploded Isometric View

A.9 Enclosure Photos

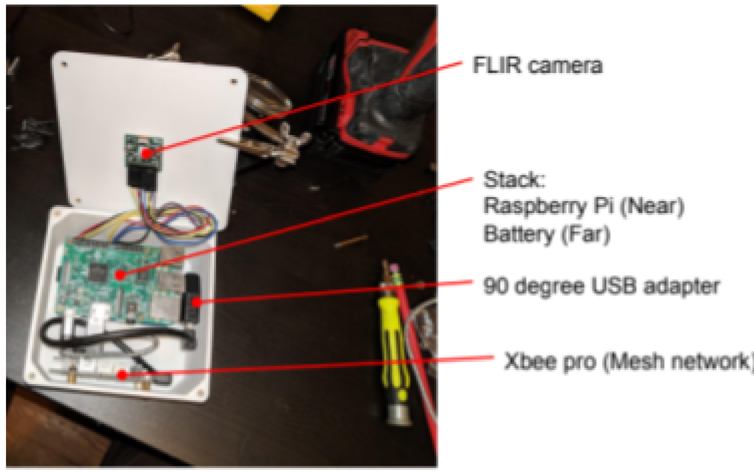

Figure A.9-1: Enclosure Interior View

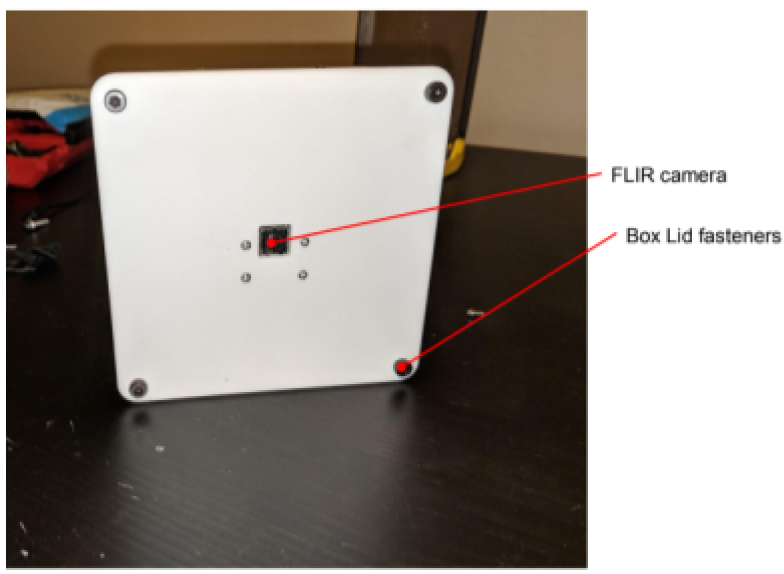

Figure A.9-2: Enclosure Exterior View