2013

Systems and image database resources for UAV search and rescue applications

David Christopher Macke Jr.

Department:

SYSTEMS AND IMAGE DATABASE RESOURCES FOR UAV SEARCH AND RESCUE APPLICATIONS

by

DAVID CHRISTOPHER MACKE JR.

A THESIS

Presented to the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

2013

Approved by

Steve E. Watkins, Advisor
R. Joe Stanley
Maciej Zawodniok

ABSTRACT

Aerial search and rescue applications using unmanned aerial vehicles (UAVs) can incorporate image processing technologies to locate targets faster and with a higher degree of accuracy. The task of developing such systems involves the development of both hardware and software components. Examples of both hardware and software approaches in the design process for an aerial imaging system are shown for small UAV applications. This project contains resources to help facilitate the incorporation of UAV applications into educational ventures and design projects. A power supply design is done for a lightweight imaging system and a large database of aerial images is assembled for image recognition development. The image set is all based in a small subset of geography, mainly lightly wooded pastureland, with a variety of lost-hiker target objects. The work describes an imaging system design, a power distribution design, target image collection and categorization, and basic image processing approaches.

ACKNOWLEDGMENTS

First I would like to thank my friends and family for the guidance, encouragement and support I received throughout my entire college career. Without the help I received from my family this project I would not have been possible

I would like to express my deepest appreciation to the following people by name:

Dr. Steve E. Watkins	Mr. Chris Mark	Mr. Ryan Bales
Dr. R. Joe Stanley	Dr. Maciej Zawodniok	Dr. Randy Moss

Each of you helped me in a significant way toward completing this master's thesis.

I would also like to thank BECS Technology Inc. for the time and facilities they donated toward the completion of this project. Without their assistance, the power distribution module for this project would not be as professionally finished as it is.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

Table Page

GLOSSARY

Geotag (noun): An indicator attached to a computer file that indicates geographic information related to the file (it can be used as a verb too, e.g., "to geotag a photo")

Global Positioning System [GPS](noun): a navigational system using satellite signals to fix the location of a radio receiver on or above the earth's surface; also : the radio receiver so used

National Marine Electronics Association 0183 [NMEA 0183] Specification for communication between marine electronic devices, in this case GPS.

1. INTRODUCTION

This thesis addresses image processing for target recognition in search-and-rescue unmanned aerial vehicle (UAV) applications. The scope of the project includes both hardware and software components for UAV target recognition. The selected targets were those related to a lost-hiker scenario. An example image collection system is shown for small UAVs. The system can be used for surveillance or search and rescue when mounted on a UAV or on a remote-controlled aircraft. The categories for targets within the images include people, signs, vehicles, shelters, clothing, bags, and empty images. All of the images were captured and tagged with the GPS location. The images were all captured in a defined environment. The environment for the project was pastureland with sparse trees in the fields. The pastureland commonly has a defined tree line around some of the targets.

Development of rescue imaging software can potentially help save a number of lives. When people are reported lost in nature, if the searchers can send up UAV's the area covered by aircraft exceeds the area covered by ground-based searchers. If the aircraft are setup to recognize rescue signs then all of the aircraft can be operated autonomously, meaning the amount of manpower required to search a large area is reduced. The manpower freed up can then be sent to the locations the drones send back as potential sites. Thus searching for people can become a more targeted operation and less of a brute force approach where the rescuers are physically trying to cover as much land as possible.

When people are lost they should try to find an open area that will allow their distress signal to be more readily observed. Hence, the best chance for aerial search

would not be dense forest land, since nothing would be seen. Lightly wooded pastureland was used as opposed to desert because pastureland was easily accessible.

This project was done to help facilitate the development of rescue sign image recognition software. Traditionally rescue sign image processing can only be developed after the airframe it will be used on is completed. The development of this image set allows people to develop the image processing software in parallel with the development of the airframe. The image set developed as a result of this project is available from the authors. The images are provided with the goal of facilitating development of search and rescue imaging software. The database of these images will proved a starting point for further software development.

The purpose of this project was to develop a system to capture images of various rescue signs at altitudes below 152.4 meters (500 ft) and to assemble a large database of aerial images. The image environment was lightly wooded prairieland with lost-hiker targets. First, an image capture system that was appropriate for small UAVs was developed and tested. As part of the system development, a custom power distribution board was designed. After collection of the images from between 0 and 152.4 meters (0 to 500 ft), the images were categorized by target type. Once categorized, basic processing methods were implemented. The images were processed using basic image processing techniques to provide insight into useful processing methods. The image database is a resource for future educational development projects.

2. REVIEW OF LITERATURE

The project of creating a search-and-rescue image set has three components that have been worked on previously. The components are the UAV, the embedded hardware, and the image processing. The image processing component, specifically the recognition of targets in a cluttered environment has been worked on for a variety of applications, from surveillance to quality control. The embedded hardware aspect has made great advancements in recent years, as an example taking the world from the cell phone to the smart phone. Lastly, the UAV technology has also been advanced such that UAVs are common aircraft.

2.1 UAV

As soon as man had developed the means for flight, people have tried to control the planes from the ground. In both World War I and II the control of aircraft was severely limited. By the time the Vietnam War came about, the aircraft had advanced to remotely piloted vehicles, or RPV. In the 1980's the military started to miniaturize the aircraft and develop more autonomous functionality. With the advancement of electronics the military finally decided to give a contract to develop UAV's in the 1990's. Since the first contract, the drone technology has advanced significantly[1].

The drones of today have advanced to be both larger and smaller with more features than the first ideas[1]. Some of the larger drones today can fly for more than a day with cruising altitudes above 20,000 feet [2]. At the other end of the spectrum, drones have gotten lighter and more powerful. Many of the drones in use by the military can be carried by a single soldier and deployed by hand. The more portable drones can also fill

civilian needs, like searching for lost people. The smaller drones could be employed by state conservation departments to search for lost outdoorsmen. The ease of operation of the smaller drones will allow for searches to take place much more quickly than on foot. Civilian UAV's have been developed and used for protecting wildlife, monitoring crops, and a number of other purposes[3][4][5][6].

A student competition called the Outback Challenge encourages student projects with UAV's[7][8]. The competition revolves around a dummy, ‘Outback Joe’, lost in the outback of Australia. The competition is for students to launch an autonomously-controlled aircraft to search for ‘Outback Joe’. Once located in the outback, the plane drops a rescue package to the dummy, one half liter of water. The advancement of the drone technology has allowed for the competition to take place. Now that technology has lowered the cost to develop a drone, and the technology is more readily available, students and universities have access to the technologies needed to research drone applications[9][10].

2.2 EMBEDDED SYSTEMS

An embedded system is a computer that was built to perform a specific task very quickly, often at near real-time speeds. An embedded system typically has a processor sized for the required application and various memory and outputs to perform a specific task. One example would be the hardware inside of a vending machine. [11][12][13].

Embedded systems are also found in vehicles. One embedded system would be the engine control unit (ECU) of a car controls the engine. Other embedded systems might be a tire pressure sensor, which warns the driver when there is a problem. Both of

the systems are tied directly to the operation of the vehicle and need to operate in real-time. Similar to the vehicle applications, aircraft have embedded systems as well. One embedded system is the autopilot. The autopilot takes in sensor data and provides real-time control of the aircraft. Another embedded system on aircraft are the de-icing electronics; they warn the pilot when a dangerous amount of ice has built up on the wing and the system can also try to get rid of the ice. The icing system also operates in near real-time to keep the aircraft from crashing[11][12].

The embedded systems that are in the class room today are much more general. Development systems often revolve around the ARM core and various single-chip computers. A few of the common single chip computers are the BeagleBoard series and the RaspberryPi series of single chip computers. These have become popular due to low cost and large development communities[14][15]. Other embedded system development in the classroom involves the development of a power supply for a specific application. For example, a power supply that is as compact and lightweight as possible.

2.3 IMAGE PROCESSING

Image processing is a subclass of signal processing where the object being processed started as an image. The processing of an image has many parallels to the processing of other signals. The goals of processing can include noise reduction, image enhancement, image restoration, and object detection. The processing of images within MATLAB allows for easy testing of various methods. The MATLAB functions allow for processing without the large amount of code development that other languages can demand[16].

The processing of images to detect targets within an image has been developed for a wide variety of applications. Some of the common applications include detection of a target, like a vehicle, on a bridge. The detection of a vehicle within a cluttered environment has many parallels to this project[17]. The images returned for the vehicle detection must place an object into a reference. The detection of a target may require some noise reduction and image enhancement[18].

One method that was studied for the project was the process of non-uniform color thresholding[16]. The thresholding of an image with different values for each color plane allows for processing based on mean color values. Along with the nonuniform color thresholding, the processing of images can be done based on the edges of objects. The edge detection of the various targets can be optimized to detect the larger targets within the image set. The various methods of edge detection, from Canny to Sobel, all have advantages and disadvantages[16]. A part of the project was the selection of the consistently successful edge detection algorithm. The final main challenge with the processing of images within the project was dealing with the clutter within the images. The clutter from trees and shrubbery create a significant challenge to detection of targets[19].

3. AERIAL IMAGING APPROACH

The technology for unmanned aerial vehicles (UAVs) provides useful tools for many types of aerial surveillance and reconnaissance. As the size and cost of UAVs have decreased and the capability of miniaturized electronics for image capture, storage, and processing have increased, educational projects and related-resources regarding aerial cameras and imaging systems have become more important.

A particularly flexible and practical application is that of search and rescue. For as long as people have been venturing out into nature, people have been getting lost. Before the 1900's, the lost adventurer's only hopes were to walk out on their own or be lucky enough to be found. Even in later years, a lost person's chances of rescue during an intentional, ground-based search, depended upon available searchers and the area being searched. These searches are more difficult the more remote and expansive the search area. The ability for searchers to sweep larger areas from the air with small inexpensive aircraft offer obvious advantages of efficiency and speed over comprehensive ground-based searches. Searches based on technology need to recognize signs, either intentional or unintentional, of human presence in cluttered environments. Autonomous image-processing approaches can identify potential targets for further aerial examination and possible ground search. Also, aerial images with potential targets can be tagged with GPS coordinates so the searchers know exactly where to go and search.

3.1 SELECTED COMPONENTS FOR AERIAL IMAGING

One such competition is the Outback Challenge Competition. The outback Challenge involves teams flying either a fixed wing aircraft or a helicopter from an

airport to a search area. There are a number of restrictions to the size, weight, flight time, etc, of the aircraft. The aircraft must be able to take-off, land, fly, search, and drop a rescue package autonomously. The challenge is conducted over a few square miles of Australian outback; an analog in the United States would be cluttered pastureland.

The Outback Challenge has a number of constraints contestants must stay within. The first constrain is the limitation on the flight time of one hour while covering 5nmi and searching a 2nmi by 2nmi area. The other influential constraint is the maximum weight of 100kg for rotary and 150kg for fixed wing aircraft. The competition revolved around locating Outback Joe, a dummy setup in a resting position. The target, Outback Joe, has many similar aspects to the image processing done in the project. Looking for a specific color or a specific pattern in an effort to locate some target. The needed aircraft must provide a stable platform for image collection and be capable of carrying the needed imaging and processing instrumentation in additional to the avionics and water bottle payload. The instrumentation includes the camera, the on-board processing (i.e. a computer), communication hardware, and associated electrical power systems.

For the aircraft system design a number of trade-offs have to be considered. One big consideration would be where to do the image processing. If the image processing is done onboard the aircraft, the system must be light enough for the aircraft to lift thus there will not be as much processing power available. If the image processing would be done on the ground, a high bandwidth communications link would be required but the processing limitation on the ground would be all but removed. Another option for the image processing would be a middle ground, where some preliminary image processing,

done onboard the aircraft, selects the photos to send to the base-station for advanced processing.

For the purposes of this project, all of the specifications were kept within the parameters of the outback challenge. The weight of the system was well below the project specifications. The height of the system was kept below the competitions maximum altitude. The only major point of the competition that differed in this thesis work was the method of image processing. For the competition, the image processing would be a hybrid between onboard and ground based processing. For the purposes of this thesis the image processing was done solely after the collection. The targets used for the collection were both relevant to the outback challenge, with a person as a target, and other potential uses, like vehicles and shelters.

3.2 UAV RESOURCES FOR SEARCH-AND-RESCUE

UAV and search-and-rescue applications are valuable subjects for educational projects. They provide opportunities for exploring interdisciplinary considerations, design optimizations, and laboratory experience. These projects can include joint design of an airframe, construction of power system for avionics, construction of system avionics, and even assignments to find optimized processing methods.

In the context of this design project, an important challenge with the development of the image processing algorithms is the collection of aerial images. For course-relate classroom projects, only aerial images are needed to investigate image processing methods. The creation of rescue image sets allows either the development of the imaging payload in parallel with the airframe or independently of a specific airframe.

The choices for images were guided by typical search-and-rescue scenarios. Lightly wooded pastureland was chosen for the target environment. Clearings in forested land is where lost people will often try to go. When lost in heavy woods with low visibility, people will travel and try to find a more open area for signaling. At the more open area, people will construct ground-to-air signals and leave around other signs of human activity. Some signs of activity include evidence of a fire, shirts, pans, packs, disabled vehicles, and trash. At chosen location, various signs of distressed were placed into the environment and photographed. Once a variety of rescue signs were placed into the environment the imaging hardware was launched. While the hardware system addresses small UAV fixed-wing aircraft, the images were taken from a stable balloon platform for convenience.

A few examples of educational uses for the image set would be as follows. A homework assignment in an image processing class could be to threshold each RGB plane of an image at different levels and see what happens. Another project could be to take an image with some blurring and try to correct the image. Alternatively the images could be processed to try edge detection using various methods to try and find the strengths of each algorithm. Any number of assignments could be created using one or more images from the set.

The work shows a possible hardware implementation for on-board camera support and provides an image database, i.e. sets of images, for testing search-and-rescue algorithms. The hardware implementation uses off-the-shelf systems with the exception of the on-board power distribution board. A custom designed power distribution module is described which provides a constant five volt output at current levels up to two amps.

The image sets include 8271 image frames in outdoor cluttered environments. Some of these frames show just the environment, i.e. no targets are present. Others show a variety of potential targets such as vehicles, objects related to human activity, and ground-to-air signals.

4. HARDWARE SYSTEMS

The second-generation Missouri S&T UAV [20] airframe is the radio-controlled (RC) plane shown in Figure 4.1. The RC plane was designed as an RC project in the senior design class AE 281 for Aerospace Engineering majors and was designed to allow modification for the requirements of the Outback Challenge Competition [8]. In particular, the plane was intended to be a stable platform for aerial imaging with a one and one half hour flight-time capability with a three pound payload capacity. The intended payload includes the autopilot, on-board computer, camera, water bottle, etc. Details on the airframe are given in Appendix A.

Figure 4.1: Second-Generation Missouri S&T UAV

The systems diagram for the UAV can be found below in Figure 4.2. The camera module provides the aerial imaging capability. The GPS module provides location information for tagging the images. The on-board computer runs the code to capture the images and annotate each image with the cameras location when the image was captured.

The battery supplies electrical power to the mobile computer, GPS module, camera module, autopilot, and communications link. The power distribution module takes the variable DC voltage of the batteries and provides a constant 5 volt line to the components. All but the last system was specified as off-the-shelf systems for integration. The power distribution power was custom designed for the UAV application.

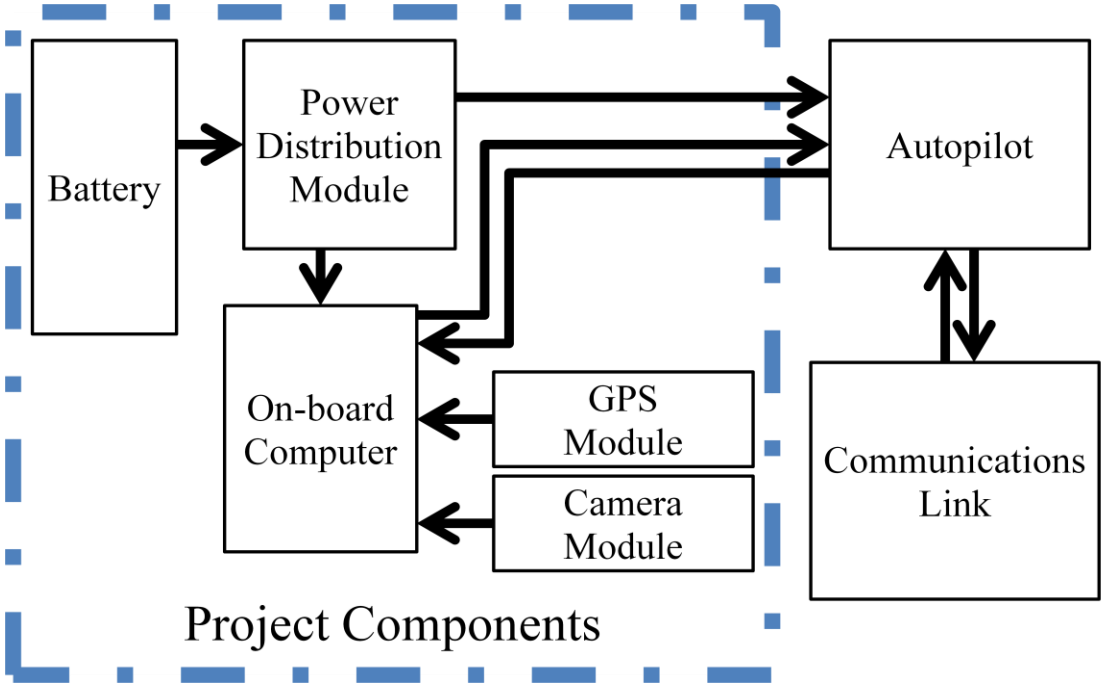

Figure 4.2: Systems Block Diagram

4.1 CAMERA MODULE

The camera module for the image capture system (see Figure 4.1) is connected to the on-board computer. The main concern of the camera module was capturing the highest resolution possible without weighing down the system. One solution to the problem was to use Webcams. In recent years webcams have become very popular for

internet use. Webcams are inexpensive and fairly light weight. To find the best option for the project a number of webcams were tested for image quality.

The camera selected was the Logitech C510. The C510 can capture 1280x720 images and only costs ~$40. The webcam was available at local stores as well as online retailers. The availability was attractive due to the risk of damage if the system fell. Being able to replace the camera quickly for a low cost would allow the system to be repaired quickly.

While the C510 was selected to be used for the build, the system could use a large variety of cameras. The libraries used for the system will accept a number of webcams and that flexibility was one of the desired features for the project. The code was written in a manner that allowed most webcams to be integrated into the system without code changes.

Most of the cameras tested only provided images with resolutions of 180x75 or similar resolutions. With that low of a resolution the images were not detailed enough to be used for the project. The C510 was selected due to its relatively high resolution of 1280x720, and low cost of only $40. Deciding on the resolution for the system involved selecting a high enough resolution to capture enough detail yet selecting a low enough resolution to be affordable.

4.2 GPS MODULE

The GPS module provides location information and is connected to the on-board computer. GPS devices have become very popular in recent years; everyone with a smart phone has a GPS module in their phone. As GPS technology has entered the cell phone

market, the modules have gotten smaller and use less power. The first step in finding a suitable GPS module was to research the options available. Various GPS Receivers are available as shown by the GPS Buying Guide produced by Sparkfun Electronics Inc®[21]. The GPS modules have a number of features, like number of channels, update rate, size and power requirement, that are important to consider. Other important characteristics of GPS modules are the communications protocol, startup time, antenna type, coverage and accuracy.

Also on the Sparkfun® site were examples of GPS modules with the features of each module listed. The module prices ranged between $40 and $80 per module. One of the most important features was the update rate, by default, most receivers update at a rate of 1 Hz, some modules are capable of update rates above 5 Hz. An adjustable update rate allows the system to change the distance covered between each update. The higher the update rate, the more often the module will be able to tell the computer the systems location. Another critical aspect of each module was the amount of power required. Most modules operate on less than 500mW. While 500mW was not a large amount of power, selecting a module with a lower power requirement was important. Another important feature of the modules was the number of satellites the module can track simultaneously, depending on the module that number can range between 14 and 66. The higher the number of satellites tracked the lower the time to acquire a first fix.

With all of the features of GPS modules defined and prioritized, the GPS module needed to track at least 15 satellites at a time. Have a positional accuracy of better than +/- 10m. Then select the module by the amount of power required. The search revealed a module that could track 20 satellites and has a power consumption of 0.325 watts. A nice

feature of this particular GPS module was the variety of output options available to the user. The GPS module provides the NMEA sentences over both TTL and RS-232 interfaces. With the additional output, the RS-232 output gives the option to use the BeagleBoard-Xm's serial port to input the GPS's information. With the GPS module selected, the final component for the system was USB storage.

4.3 ON-BOARD COMPUTER

The system to collect the images during flight required a low-power processor to run the collection program. The processor component of the system can be found in Figure 1, the part labeled On-board Computer. The market of low-power single chip computers has exploded in recent years. The market went from a single major item, the gumstix® computer, to several commercially available options like the Raspberry Pi®, and the Beagleboard® series. Each low power computer series has advantages and disadvantages. Selecting the correct computer for the project required researching the options determining the best fit.

The BeagleBoard Series of single chip computers was selected as the mobile computer to drive the system. The BeagleBoard-Xm series had an MSRP of $149 and several attractive features. The Xm had an ARM Cortex-A8 running at 1GHz and 512MB of onboard memory. The Beagleboard-Xm includes an Ethernet port, a serial port, and 4 USB ports. The Xm runs on 5 volts and has a maximum power consumption of ~2 watts and weighs only ~37 grams. The large number of physical ports on the BeagleBoard-Xm allows for a range of cameras and GPS modules to be used [15]. The large support

community behind the BeagleBoard family of products provided a number of avenues that can help to trouble shoot problems encountered throughout the project.

One concern about the BeagleBoard-Xm was the storage medium. The system operates on a MicroSD card inserted into a slot. The card serves as the main drive for the computer. The drive has a significant amount of read/write time due to the limited system memory. The read/write activity means the MicroSD card must have some free space at all times to store swap. If the card ever fills, the system will wedge and crash. To prevent the system from crashing a separate drive was used to store the images. The images were written to a thumb drive connected to a USB port. A secondary benefit of the thumb drive is the collected images can be accessed quickly at the end of a flight. At the time of thumb drive removal an empty thumb drive can be inserted so return to flight does not depend on the rate images can be downloaded from the memory.

An alternate to the BealgeBoard series was the Gumstix computer. The Gumstix computer series was started in 2003 with one of the first extremely small computers with modern processing powers. The Gumstix computer series bases the computer on a main computer chip, normally ARM A8 and A9's. The main boards range between $99 and $230 depending on included functionality. Some common specifications of the main boards are operating frequencies between 700MHZ and 1GHz. A number of the main boards are equipped with 512MB of RAM. Many of the main boards come with Wi-Fi, Bluetooth, and some digital signal processing (DSP) Capabilities. In combination to the main boards of the system, expansion boards can be added on to provide additional functionality. Some of the expansion boards that are relevant to the project are a GPS module for $130 and various camera modules each costing $75 and having a resolution of

752x480. Totaling up all of the parts needed for the project, a Gumstix computer would cost between $300 and $435 per setup. The most relevant information for the Gumstix computer was built into a table[22].

Another alternative was the Raspberry-Pi. The Raspberry-Pi was initially released February 29th, 2012. There are two models of the Pi, a model A and model B. The model A, which costs $25, uses 2.5W of power, 256MB of memory, and 1 USB port. The model B, which costs $35, uses 3.5W of power, can come with 512MB of memory, and 2 USB ports. The Pi models, both using an ARM11 core, generally operate around 700MHz. The Raspberry-Pi foundation has also released a camera module. The camera module costs $25 and can take pictures up to 2592 x 1944[14] [23].

The concept for the project was to make the system as flexible as possible with respect to components. The flexibility would allow the system to be recreated with alternate components if a part used in this design was no longer available or outside the budget, another component could be substituted. The next concerns for the project were weight and cost together. With those project considerations in mind, the first elimination was the Gumstix computers. The Gumstix computers required the entire system to be made from the parts they sold and did not allow for outside components. Along with the highest price of any setup, the Gumstix were deemed not the best fit for the project.

The next step was to eliminate either the Raspberry-Pi or the BeagleBoard-Xm. The flexibility of the two systems was comparable but with four USB ports, an Ethernet port, and a serial port, the Xm had a slight advantage. The Raspberry-Pi was less expensive, by more than four times. The weight was comparable, at ~60 grams for the Raspberry-Pi and ~75 grams for the BeagleBoard-Xm. Both computers have an active

development community that can help with troubleshooting. In the end, the flexibility of components the BeagleBoard offers makes it the better choice for the project.

4.4 SOFTWARE

The software to capture the images needed to be able to have some specific functionality. One capability the software needed to provide was to capture images from the camera. As seen in Figure 1, both the camera and GPS module attached directly to the On-board Computer. Through that physical connection, the software would need to save images from the camera. Another capability the system needed to provide was to read GPS information over some connection port, either USB or Serial. Reading the GPS information would allow the software to annotate each image with the exact location of the system when the image was captured. The last main capability of the system was the ability to program C++ code within the system[24].

With all of the features in mind, the image capture code was developed on the BeagleBoard running an Ubuntu 11.04 distribution. The GPS and camera were connected to the computer over USB interface. The compiler used was G++. A number of libraries were used to provide all the required functionality.

The OpenCV library was used to facilitate the image capturing from the camera. The OpenCV project was started by Intel® and is now supported by Willow Garage and ItSeez. The code is under the open source BSD license meaning it is free for public use. The OpenCV library was used because of the cross-platform nature of the project and the support community around the code gives a wealth of experience that can be drawn upon.

The Video4Linux2 library, also known as V4L2, was used to interface with the camera on the system. The library was used due to its adaptability with supporting a large number of devices. The V4L2 software was distributed under the GNU General Public License. Much like the OpenCV library, the code has a large support community.

The EXIF2 library was used to add the location information to the images. The EXIF2 library was distributed as open source under the GNU General Public License. The library can be purchased and used in closed source projects as well. The library used for the project was open source freeware. The EXIF2 library gives access to the Exchangeable Image File Format, or EXIF, information used by digital cameras. The EXIF information that can be included in images ranges from date and time information to focal length and aperture settings. The project uses the GPS information that the EXIF format specifies. The specific information included was the latitude, longitude, altitude, speed, status, and number of satellites. The information contained in the images will allow people working with the image to extract the location of potential rescue locations.

The code was written to be as simple to follow as possible. As shown below in Figure 4.3, the code initializes all of the components and then it enters a WHILE loop. The code sits in the while loop until the escape button gets pushed. A full copy of the image capture program can be found in Appendix B.

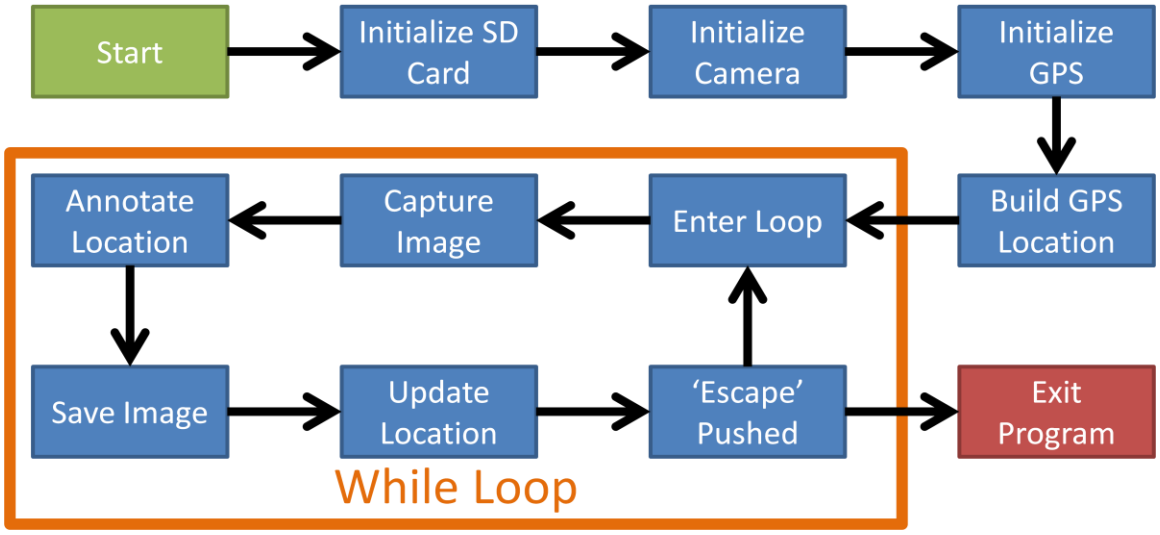

Figure 4.3: Image Capture Code Flow

4.5 BATTERY

Part of the airborne project involved developing a mobile power distribution module for the BeagleBoard-xM and accessory components. The battery and power distribution module can be found in Figure 4.1. The batteries would provide the power for the system in flight and the power distribution module would convert the power from the battery to the correct form for the BeagleBoard-Xm. The first step in creating a mobile power distribution module for the project was the selection of the system batteries. Selecting the batteries involves first selecting a battery technology. Previous experience provided a base of knowledge about each technologies advantages and disadvantages. The technologies considered were Lead Acid, Nickel–Metal Hydride (NiMH), and Lithium Polymer (LiPo). Each battery technology specifications were built into Table 4.1, as seen below.

Table 4.1: Battery Technology Breakdown

Battery Technology	Lead Acid (Pb)	Nickel-Metal Hydride (NiMH)	Lithium Polymer (LiPO)
Nom. Cell Voltage (V)	2.1	1.2	3.7
Specific Energy (Wh/kg)	30-40	60-120	130-200
Energy Density (Wh/L)	60-70	140-300	300
Power Density (W/kg)	180	250-1,000	Up to 7,500
Life Cycle (Avg)	500-800	500-1,000	750
Hazardous (Yes/No)	No	No	Yes

Lithium Polymer, or LiPo, batteries were selected to power the imaging system. Lithium batteries were the newest technology of the three. LiPo technology had saturated the cell phone due to the high cell voltage, 3.7 volts, high energy density, 130 - 200 Wh/kg, and the very high power density, up to 7,500 W/kg. The excellent performance of LiPo batteries was a main reason for the rapid acceptance into the cell phone market. In addition to the cell phone market, LiPo batteries can be found in cordless tools, cameras, cars, and remote control toys. All of the LiPo specifications were placed into Table 4.1.

The major drawback of LiPo batteries was risk of failure resulting in fire. The concern over fire and explosion can be greatly mitigated with safe handling and attentive charging. Part of safely handling LiPo batteries involves not discharging the battery fully. When fully depleted, the LiPo cell will register ~3.2 volts. At the low voltage level, the LiPo batteries internal chemistry can be damaged and result in lower battery charge capacity. The general rule for LiPo cells in the RC community was to not let the battery

voltage go below 3.6-3.7 volts. Stopping at 3.6-3.7 volts keeps the battery above an approximate discharge state of 80%. Therefore, RC groups recommend sizing the battery capacity and then dividing by the 80% usable capacity received. The precaution of only discharging to 80% of the capacity is not required but does help to preserve the usable life of the battery.

The last step in roughing out the battery selection was sizing the battery pack for the aircraft. For the sizing calculation, the desired flight time was one hour. In that one hour the onboard equipment could draw up to 2 amps continuously. The 2 amp load on the power distribution module would be at 5 volts, totaling 10 watts of power. With the estimation that the power distribution module designed would be at least 80% efficient, the power available from the battery would need to be 12.5 watts. With the power requirement being 12.5 watts, the configuration of the LiPo cells offers a few different configurations, most commonly, two or three cells in series, offering either 8.4 or 12.6 volts respectively. A two-cell battery would need to be at least 1500mAh and a three-cell would need to be at least 1000mAh. With the two capacities determined, dividing by the usable 80% rule, the capacities become 1875mAh and 1250mAh respectively. Both of the two-cell and three-cell batteries were available from hobby stores.

An alternate to the LiPo battery was the Lead Acid battery. The most common commercial battery technology was Lead Acid. Lead Acid batteries were invented over a hundred years ago and since then the technology has fully matured. Lead Acid batteries have a variety of uses, from the automotive industry to the solar industry. Lead Acid batteries have a nominal cell voltage of approximately 2.1 volts. The energy density is between 30 - 40 Wh/kg and the power density is approximately 180 W/kg. The average

cycle life of a lead acid battery is 500-800 cycles. Lead Acid batteries are very inexpensive compared to other battery technologies. Finally, the operating temperature for longest life, needs to be between 0 - 50° Celsius. With no obvious problems involving the lead acid technology, all of the technical specifications were placed into a table which can be found below[25].

Another battery technology researched was Nickel-Metal Hydride, or NiMH. NiMH batteries have been used in cell phones for several years and the technology has been very well researched. NiMH batteries were used in remote control vehicles for years before lithium batteries came to market. NiMH batteries have a nominal cell voltage of 1.2 volts. The technology has energy densities between 30 - 80 Wh/kg and power densities between 250 - 1000 W/kg. The life cycle of a NiMH battery averages between 500 - 1000 cycles. NiMH batteries average cost is more than double the cost of Lead Acid batteries. Lastly, NiMH batteries optimum operating temperature ranges is between 0 - 50° Celsius. All information about NiMH batteries was placed into a table to be compared to the Lead Acid batteries[25].

From the research into the various technologies, the first step was to eliminate the Lead acid batteries from the list of potential technologies. There were several reason Lead acid batteries needed to be eliminated from consideration. First, Lead Acid batteries have a very low energy density, which means that to power the onboard system for a full flight, several small batteries or a single large battery would need to be used. Any battery large enough to power the system for a full flight would also be too large for the aircraft to lift. The next problem was the low cell voltage, each Lead Acid cell only providing ~2.1 volts. So to provide the needed voltages of >5 volts, at least three cells would need

to be connected in series. The challenge with having a three cell Lead Acid battery was the physical dimensions of the outside of the battery; they tend to become very large, again due to the relatively low energy density. For all of the reasons listed previously, Lead Acid batteries were determined to be an inappropriate power source for the onboard system.

Both Nickel-Metal Hydride and Lithium Polymer batteries have been used in RC applications for years. More recently, people have been using more LiPo batteries and less NiMH. There are a few reasons for the shift from NiMH to LiPo. Both LiPo and NiMH batteries have similar specific energies and energy densities. LiPO batteries have a lower self discharge rate than NiMH. LiPo batteries can also have up to 7.5 times larger power density than NiMH. The largest advantage NiMH batteries have over LiPo's was their safety factor. NiMH batteries are easier to charge and discharge compared to LiPo. Recent advances have been made to make LiPo batteries much safer to work with. After taking all of this into consideration, LiPo battery technology was selected for the project. The risks while evident can be substantially mitigated with the proper precautions.

4.6 POWER DISTRIBUTION MODULE

Selecting the batteries to power the system provided a source for the system, but a power distribution module was the link between the batteries and the various systems of Figure 4.1. The power distribution module takes one form of power and converts it to another form; one common example takes AC power from a wall outlet and converts it to DC power for your cell phone. The power distribution module needed to take a variable DC voltage and convert it to a constant 5 volts DC. For the purposes of the power

distribution module section, assume the use of a two cell LiPo battery running at between 7.6 and 8.4 volts. There were a number of components that needed power from the batteries. The two main loads on the batteries were the BeagleBoard-xM and the GPS module. The BeagleBoard-xM, when being powered by the DC port, requires between 4.8 and 5.2 volts. The current required for the board alone was specified at up to 750mA. The current required by the board can increase as a result of the USB devices used on the board. The maximum current the board would supply to the four USB devices was 1500mA. During operation, only two of the USB ports will be in use. The USB specification dictates a maximum current of 500mA per port. Thus the maximum additional current was 1000mA. The 500mA per port was used because that was the maximum amount of current a USB device can draw[26]. With the current draw of the BeagleBoard-xM, 750mA, added to the maximum current drawn by the two USB ports, 1000mA, the total current for the BeagleBoard-xM was estimated at 1750mA.

The other device that was drawing power from the distribution board was the GPS module. The GPS module could be powered with between 3.3 to 5 volts, a much wider range than the BeagleBoard-xM. The typical current drawn by the GPS module was 65mA. The low draw of the GPS module was then added into the previous total resulting in a maximum current consumption of 1815mA. The power distribution board was also built to serve other project. Changing two of the resistors on the board will make the board capable of powering the autopilot and communications link used onboard the aircraft.

The final requirement for the power distribution board was additional current capacity for future expansion. A number of projects were being worked on for the

airframe such as locator beacons and radar altimeters. To allow some capacity for future components, another 500mA were added to the running total, resulting in a final current capacity of 2315mA. With the current total at 2315mA, the circuit designed to fill the role would be built to output 2500 mA.

4.7 POWER DISTRIBUTION BOARD DESIGN

The power distribution module designed was a switching regulator. The regulator takes input between 7.6 volts and 40 volts and outputs a constant 5 volt line at up to 2.5 amps. As designed the switching regulator has a switching frequency of 2.2MHz. A few additional features of the switching regulator are a low voltage cutoff, to prevent damage to the lithium cells, an indicator LED to signal when the regulator is active, and a large ground plane within the PCB to help prevent overheating of the integrated circuit chip and to lessen the noise generated by the circuit. Finally several through hole connectors were provided on the output line to allow for future expansion of the system. An image of the final circuit can be seen below in Figure 4.4.

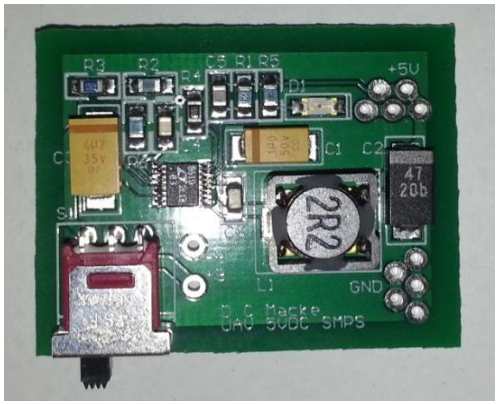

Figure 4.4: Power distribution module

The realization of the switching regulator was not an easy process. There were several problems along the way that delayed the completion of the power distribution module. The start of the process was an initial search looking into different types of power converters. There are a number of types of dc-dc converters, for the voltage and current range of this project, two main converter types stood out. The Linear Regulator and the Switch-Mode regulator each fit well enough that they deserved consideration for the power distribution module.

The first type of power converter researched was the linear regulator. A linear regulator takes a range of input voltage above the desired output voltage and outputs a steady voltage. The inside of a linear regulator is an adjustable voltage divider to output the desired voltage as the input voltage changes. The device dissipates any power above the desired output power as heat. As a result linear regulators tend to be used in other applications . One of the attractive parts about using a linear-regulator was the parts cost, parts suppliers listed several 5 volt linear regulators, almost all under $0.50 per part. The downside of the linear regulator was twofold, the low efficiencies and the need to have the input voltage above the output voltage. Also linear regulators cannot have voltages higher than the input voltage nor can they invert the input voltage. The low cost and ease of implementation ensured linear regulators were considered.

The other main type of power converter researched was a switching regulator. A switching regulator operates much differently from a linear regulator, where the linear regulator was passive, the switching regulator was active. The switching regulator turns on the output for a short period of time, during this time the inductor on the output reaches a quasi steady state. After a specific amount of time the output turns off, the

current keeps flowing through the inductor. After the output was off for a specific amount of time, the output would turn back on and the process would start again. The active nature of a switching regulator allows them to reach efficiencies above 90% with a wide range of input voltages. The output of a switching regulator can be electrically noisy however there are components on the output that reduce the ripple and high frequency noise that may be present in the output signal.

There were advantages to both types of power converters. Linear regulators tend to be inexpensive and simple to implement. The simplicity of implementation, often only requiring one chip, allows for the total size of the board to remain small, thus negligible weight. The tradeoff with linear regulators was the amount of power wasted in the conversion process. The more power wasted in the conversion process the larger the battery system needed to be to account for the lower efficiency and the increased heat sinking needed to prevent regulator damage.

Switching regulators tended to cost a little more, on the order of $5 per chip, and are more difficult to implement, requiring more components in the design. For the trade-off in price and board size they had much higher efficiencies, often above 90%. The high efficiency of the conversion process allowed the battery system to be smaller.

With the trade offs in mind, the initial design was to use a 5V linear regulator to convert the battery voltage to the required system voltages. With the linear regulator selected, the process to verify safe operation of the chip began. Knowing the battery voltage would range between 8.4 and 7.6 volts, and the maximum designed current would be 2.5 amps, the calculation for power dissipated through the linear regulator showed the chip would need to dissipate between 8.5 and 6.5 watts. For the chip to be able to

dissipate 8.5 watts of power, a prohibitively large heat sink would need to be added to the chip. The heat sink would need to be very sizeable and would have a substantial negative impact on the flight characteristics of the airplane. The cooling requirement eliminated the linear regulator from the pool of potential power converters

$$Power = (Voltage_{Battery} - Voltage_{System}) * Current_{MAX}$$

$$8.5W\ to\ 6.5W = ([8.4\ to\ 7.6] - 5.0) * 2.5$$

With the elimination of the linear regulator from the list of potential power converters, the only other option was a switching regulator. The switching regulator involved more component chips and thus a larger board but the benefit was efficiency. Searching for a switching regulator lead to one of many companies , Linear Technologies. A product search returned tens of chips that could provide a constant 5V output from a variable DC input. After searching through the results, the LT1374 looked to be the best option based on efficiency, size, weight, and cost.

Looking at the LT1374 datasheet revealed the standard configuration for a 5V buck converter. The LT1374 would have efficiencies between 89% and 90% depending on the current load. Another benefit of the chip was the shutdown input on the chip. By default the shutdown pin does not need to be connected but if implemented, did allow for more robust circuit protection. The circuit protection would allow the circuit to shutdown when the battery voltage dropped below a specific value. For the circuit, it was determined that when the battery system dropped below 7.6V the power distribution module should be turned off. Turning the power distribution module off at 7.6V protected the batteries from being over drawn and damaging the cells.

To activate the shutdown pin on the LT1374, a circuit based on an open collector comparator was designed. The open collector comparator would compare the battery voltage to a fixed voltage and would set its output, high impedance or grounded, based on this comparison. To fill the comparator need, another search resulted in finding the LT1011, a Linear Technologies voltage comparator. To provide a fixed voltage for comparison, a resistor and reverse biased 6.2 volt zener diode was connected between the positive and negative battery terminals. The node between the resistor and diode, a fixed 6.2 volts, was connected to the inverting terminal of the LT1011. A voltage divider was designed and placed between the positive and negative battery terminals. The node between the resistors was connected to the non-inverting terminal of the LT1011. The implementation of the LT1011 set the output to ground when the battery voltage was above 7.6 volts and set the output high, when the battery voltage was at or below 7.6 volts. The high output of the comparator connected to the shutdown pin turned off the LT1374.

A circuit was needed for the power distribution board to prevent the battery voltage recovery from reactivating the power distribution module. Batteries operate with an internal chemical reaction; the reaction provides some voltage potential across the terminals. If the current required exceeds what the chemical reaction could provide, the voltage will drop. The voltage can also drop as a result of the battery discharging under normal operation. If the load was then removed from the battery, the voltage could recover a small amount. As a result of the recovery, the battery voltage might go above the cutoff voltage turning on the power distribution module again and drawing the battery below a safe state of discharge.

To prevent the voltage recovery from turning the converter back on, some hysteresis was added to the comparator. Hysteresis helps to prevent the circuit from powering back on until a voltage higher than the recovery voltage was applied to the circuit. To provide the hysteresis, a resistor was placed between the output and the non-inverting terminal of the LT1011. The resistor was designed to provide 0.4 volts of hysteresis. The 0.4 volts of hysteresis requires the battery voltage to go above 8.0 volts, a voltage well above the recovery potential for the LiPo batteries. The result of the circuit was a comparator that would only turn on when a charged battery pack was used with the power distribution module.

The final additions to the comparator circuitry were an active indicator and a main power switch. An LED was placed in series with a resistor and then placed between the positive battery voltage and the output of the LT1011 chip. The LED would be on when the converter was on, any battery voltage above 7.6 volts, and the LED would be off once the battery voltage dropped below 7.6 volts. The switch was placed in between the positive battery terminal and the rest of the circuitry. The main power switch will be turned on only once the system was setup in the airplane ready for flight. The switch will prevent the system from turning on while the batteries were on the charger.

With the initial design of the power distribution module completed, Linear Technologies was contacted to have the design verified by an FAE, or Field Application Engineer. The Linear Technologies FAE reviewed the design and suggested a few changes. The FAE also pointed out another LT chip, the LT8610, that had higher efficiencies and built in comparator shutdown functionality. The trade-off of the 8610 was the maximum current capability. The LT1374 had a maximum current of 3.5 amps,

where the LT8610 had a maximum current of 2.5 amps. The 2.5 amps output provided by the 8610 was within the maximum current rating of the system. Another benefit of the LT8610 was the higher efficiency compared to the LT1374. Where is LT1374 had an efficiency of ~89%, the LT8610 had an efficiency of ~93%. With the LT8610 appearing to be a better choice, a new power distribution module was designed around the LT8610[27].

The new power distribution module was designed around the LT8610. In the datasheet for the LT8610, the typical application showed the 5 volt step-down converter without the voltage shutdown implemented. Keeping in mind the normal operating load of 1.5 Amps, the parts selection was performed. Selecting the resistor for the switching frequency was one challenge. The LT8610 switching regulator could operate at fixed frequencies between 0.2 and 2.2MHz. Placing certain resistor values between the RT port and ground caused the regulator to operate at different frequencies. With some guidance from Linear Technologies, the switching frequency of 1MHz was selected, corresponding to a resistance of 41.2 kΩ. The final schematic for the LT8610 implementation can be seen in Figure 4.5. The GERBER layout file can be seen in Figures 4.6 and 4.7

The final step in the parts selection for the LT8610 was the implementation of the low voltage cutoff. The low voltage cutoff happens when the EN/UV pin drops below 1 volt. Given the cutoff needs to happen when the battery voltage reaches ~7.6 volts, the resistor combination of 33.2kΩ and 4.99kΩ were selected. With the parts selection complete, the schematic was sent to a contractor to be used to create a layout file.

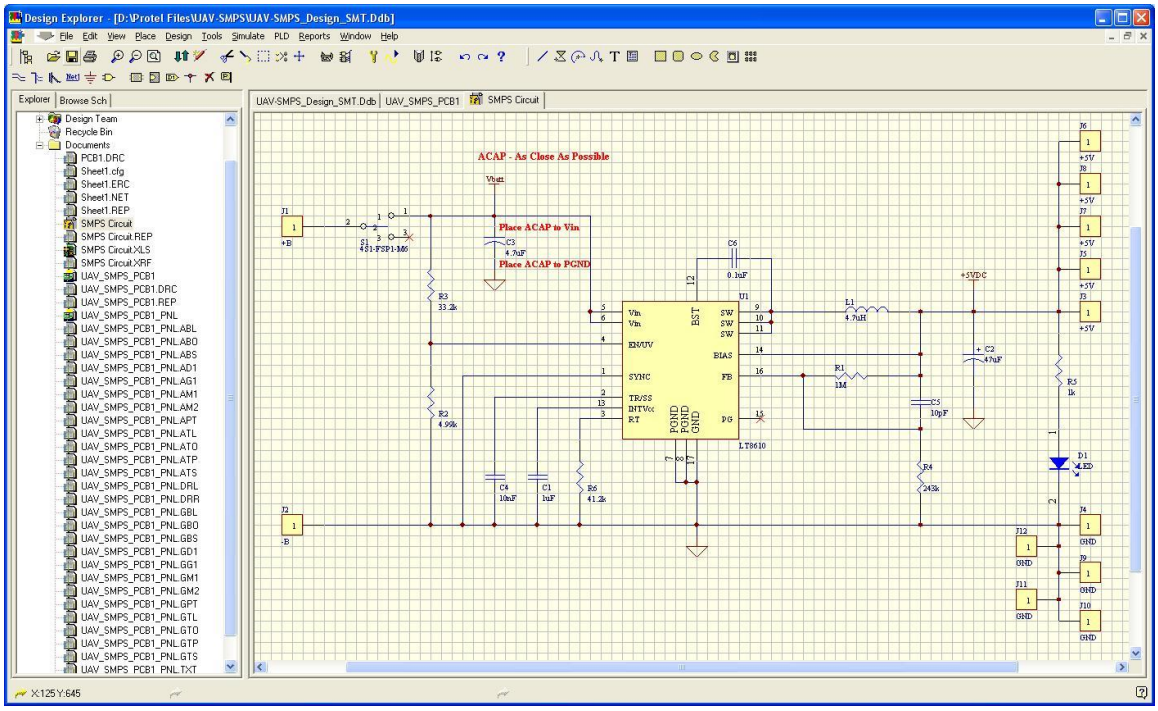

Figure 4.5: Circuit Schematic

The PCB layout had a very small footprint, 27.94mm by 35.56mm (1.1" by 1.4"). There were a few design considerations to note on the PCB design. The first design aspect was the large ground plane on the bottom of the PCB. The ground plane helped to limit the EMI sent from the chip and created a sizeable heat sink for the LT8610. Another design consideration was the power switch. The power switch was placed in the bottom left corner of the PCB. The small slide switch shown can be thought of as the default option, the switch will turn off the power distribution module. The next option would be to short the switch ports so the power distribution module always activates whenever enough voltage is supplied to the system. The elimination of the switch was useful while verifying the circuit. Shorting the switch could also be useful when the switch is unnecessary, then the cost to build can be reduced by a few dollars. Finally the switch could be replaced with a remote switch. For example if the system were permanently

mounted inside of an aircraft without easy access to the power distribution module, a remote switch could be run and mounted to an accessible location on the aircraft. With the power distribution module board using 993.54 mm^2 (1.54 in^2), the contractor was able to place 30 modules onto a single composite PCB. The benefit of placing several copies of the same layout onto a composite PCB was seen in dollars. With the fabrication houses cost spreadsheet, it made more sense to order one composite board with 30 modules and pay the extra fee of $50 vs order 10 module boards at $20 apiece. After verifying the design files, they were sent off to the fabrication house.

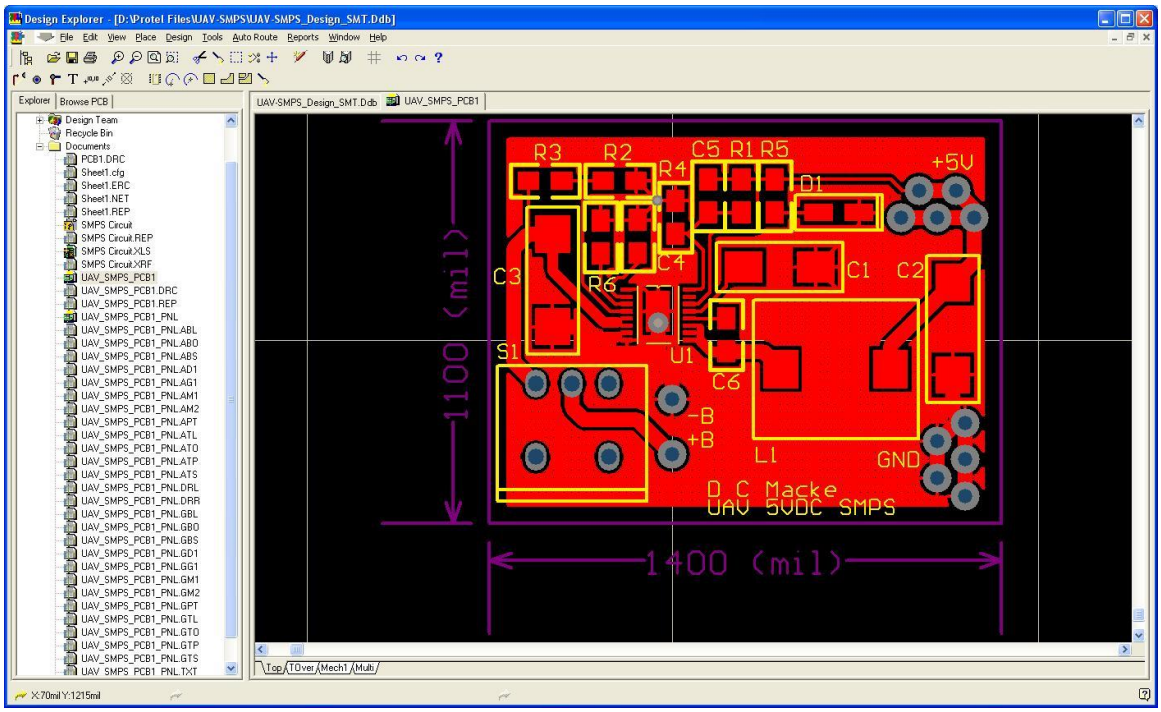

Figure 4.6: GERBER Layout File

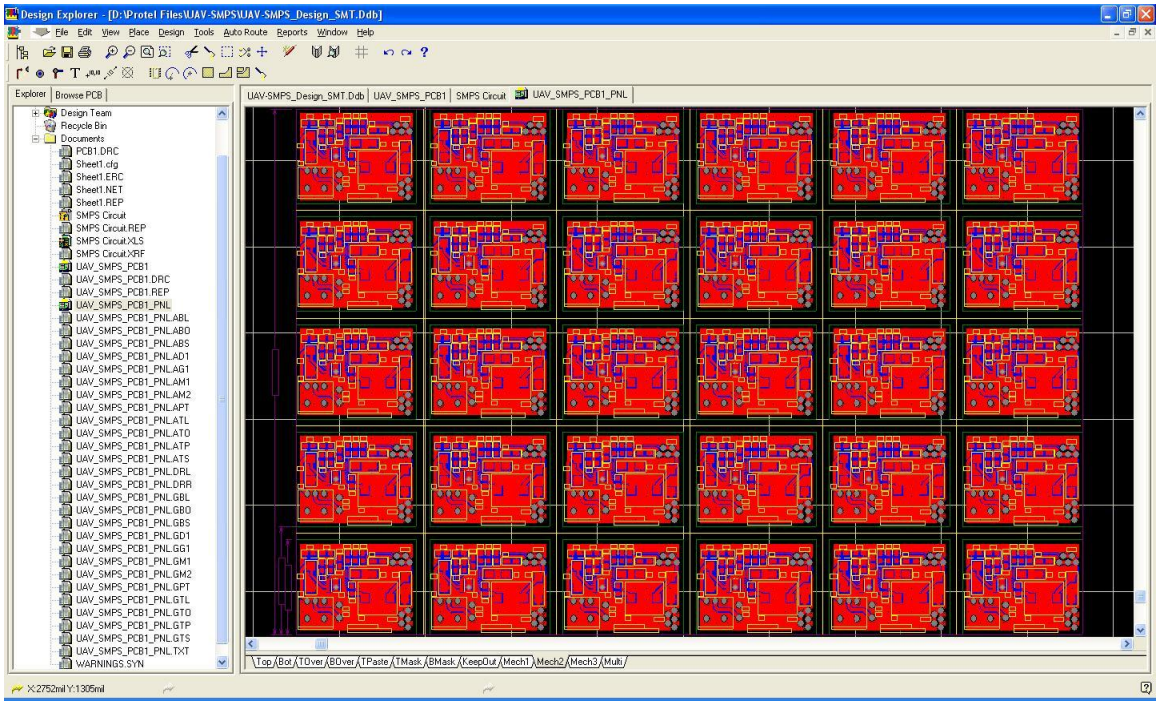

Figure 4.7: GERBER Panel Layout (30 Boards)

The board was received from Advanced Circuits and assembly was started. Most of the parts could be put down by hand if needed. The one part that would be difficult was the LT8610. The part has 16 connectors and a bottom ground pad in an area 5.23mm by 2.59 mm(.206 inches by .102 inches). Due to the LT8610's size, BECS Technology Inc. was contacted to use their reflow oven. The reflow oven provided a higher probability of successfully soldering the LT810 to the PCB and a more consistent soldering result if compared to any hand soldering process.. After putting the first batch of 5 LT8610's down, the chips had a lot of bridging, or solder connecting adjacent pins. The first batch was cleaned up and all the bridging removed. The next batch was put down with less solder paste but bridging remained an issue with each batch of LT8610's.

While at BECS, all of the LT8610's available were placed, 25 in total. In total, 10 boards were fully assembled that day. The choice of only finishing 10 boards was done so

that if reworking was necessary, all of the boards would not have to be reworked. The process of reworking the boards involved changing R1, R2, R3, R4, and R6 to adjust various voltages and the switching frequency. There was a need for a few completed boards for testing purposes and those 10 would fulfill that need.

5. POWER DISTRIBUTION MODULE VERIFICATION

The power system is shown in Figure 5.1. It consists of the Lithium Polymer batteries and the custom power distribution board. The power distribution module takes the variable voltage from the batteries and provides a constant voltage to the other systems. The purpose of verification of the power distribution module was to test the circuit for correct operation and to profile the circuit performance. Checking for correct operation involved testing the capability of the circuit to output the desired voltage over the expected current values. Profiling the circuit involved connecting the circuit to an oscilloscope to view the output from the circuit and any noise on the output.

Figure 5.1: Power System

5.1 TESTING PROCEDURE

After assembling the first few boards, they needed to be tested. Testing the boards allowed for the verification of various functions. The board needs to be able to produce a constant 5 volt output with a range of input voltages, between 7.6 volts and 12.6 volts. In

addition to maintaining the 5 volts output at no load, the board needed to be able to maintain the 5 volt output at current levels up to 1.75 amps. The process of trimming the board involved changing a few resistor values. The resistor values being adjusted were R1, R2, R3, R4, and R6. Tuning the board was an iterative process, adjusting the output current, then the output voltage, then returning to the output current.

With several power distribution boards completed the verification process was started. The first step in the verification process was seeing the indicator LED turn on once the input voltage went above 7.6 volts. The next step was to place a very small load onto the output and verify 5 volt output at low currents. For the test a 50Ω load was placed on the output. The output voltage was as at 5 volts but some ringing was observed on the oscilloscope.

The next step was to design a method to test the output of the power distribution module over the expected use currents, between 0.25 and 1.75 amps. It was decided that testing the output voltage every ~0.25 amps could either verify the functionality or indicate the need for more testing. The power distribution module tested fine up until the load current reached 0.75 amps. At the 0.75 amp load level, the switching regulator was not maintaining the required voltage. After some investigation, it was determined the switching regulator was not switching fast enough, as a result the current through the inductor was dissipating. The RT resistor was changed to 30kΩ and retested. The higher switching frequency yielded a greater operation range, but the chip had troubles maintaining 5 volts at a 1.0 amp load. The switching frequency was increased and the output retested a few more times until a stable output voltage for loads between 0.25 and

1.75 amps. The final RT value used was 16.6kΩ, corresponding to a switching frequency of 2.15MHz.

After a stable output was obtained over the expected current load, the process of fine tuning the feedback resistors began. Initially the feedback resistors were 1MΩ and 243kΩ, these resistors yielded an output voltage of 4.95 at low current loads. At higher current loads, the voltage would drop low enough that the BeagleBoard-Xm would turn off. Fixing the output voltage level required adjusting the feedback resistor combination. After reworking the resistor divider the new combination was 1MΩ and 226kΩ, giving an output of 5.15 volts. After getting both the output voltage and current capability in range, both parameters were tested again to verify functionality.

The last major hurdle to the verification of the power distribution module was tracing a 60Hz signal seen earlier on the oscilloscope. While testing the power distribution module at the assorted current loads, there was a 60Hz ripple seen on the output. The ripple had a peak to peak voltage of ~200mV and caused the power distribution module output to oscillate between usable and unusable voltages. The ripple voltage can be seen below in Figure 5.2.

Given the signal was at 60Hz, the likely culprit was some signal leaking through the DC power distribution module. To help determine if the DC supply was the source of the 60Hz signal, the power distribution module was connected to a 12 volt battery. Using the battery as the source and testing all of the load currents, the 60Hz signal disappeared from the output. The circuit test setup can be seen below in Figure 5.3. Another benefit to the battery supplying the power was the cleaner output signal. When the DC power

supply was the source, the output had a large amount of noise within the output. After switching to a battery, the output of the signal was much less noisy.

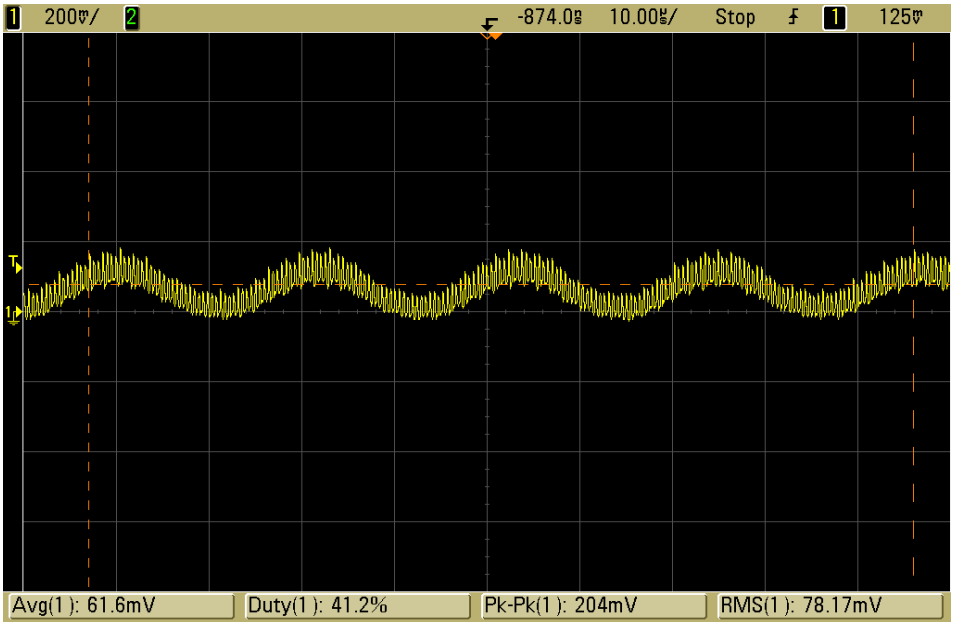

Figure 5.2: 60Hz Phantom Signal

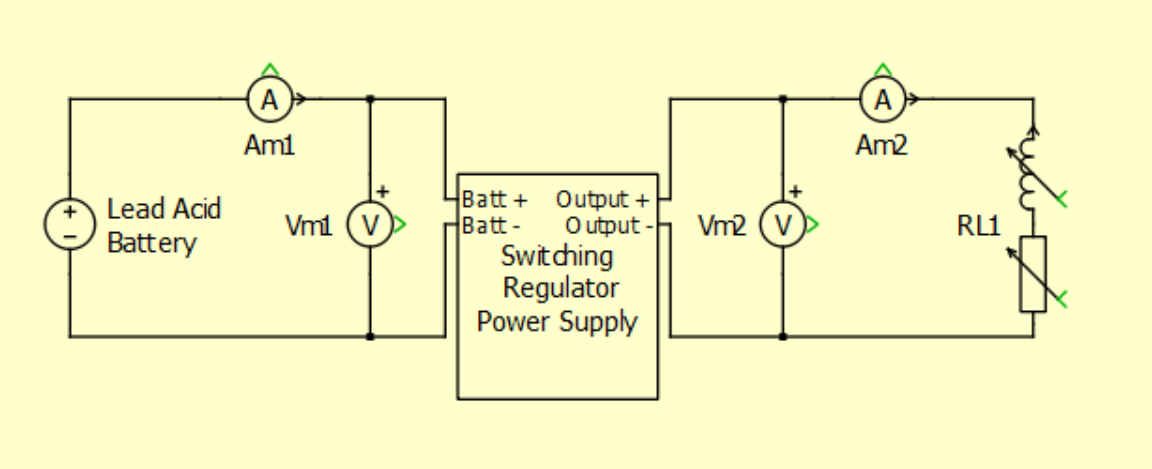

Figure 5.3: Testing Circuit Setup

With the power distribution module operating as expected, a more complete test of various load currents was performed. For the verification process, the power was supplied by a 12 volt Lead Acid battery. The battery was used to remove the 60Hz bleed

through seen earlier in the verification process. The test swept over likely current loads for the power distribution module during normal operation. The measurements were recorded and can be found in Table 5.1 and 5.2. One challenge of the testing setup was the load resistors in use. The resistors were sand filled wire-wound and thus had an inductive component. Each of the resistors was measured with an HP LCR meter and added into an Excel spreadsheet which can be found below in Figure 5.4. At times the inductance of the various resistors exceeded the inductor placed on the output of the power distribution module . The challenge of the high inductance resistors created problems with observing the output of the power distribution module on the oscilloscope. The inductance would cause the peak-peak voltage to be larger than under normal operation. To help gain a better idea how large the impact of the inductance was, oscilloscope images were captured with similar output loads for both the resistors and the BeagleBoard-Xm. The images show larger and more frequent oscillation with the resistive load.

Table 5.1: Verification Resistor Values

Resistance				Inductance	
Desired	DMM	LCR	Calculated	1 kHz (µH)	120 kHz (mH)
30.00	31.40	31.35	30.03	3.60	0.032
25.00	27.55	26.48	25.06	1.20	0.025
20.00	21.33	20.51	20.03	1.40	0.010
15.00	19.16	15.63	14.96	3.10	0.010
12.50	16.75	13.09	12.47	2.30	0.006
10.00	11.25	11.43	9.99	2.70	0.006
7.50	9.06	9.58	7.53	1.20	0.005
5.00	6.10	5.88	5.10	0.30	0.003
3.30	5.64	4.22	3.35	0.10	0.003
2.50	4.57	3.16	2.52	0.40	0.003
1.67	4.93	2.99	1.70	0.10	0.003

Table 5.2: Verification Results

Resistance	Input			Output			Power Lost (%)	
Calculated	Voltage (Volts)	Current (Amps)	Power (Watts)	Voltage (Volts)	Current (Amps)	Power (Watts)	Total (Watts)	Percent
30.03	13.076	0.0842	1.101391	5.108	0.1701	0.8689	0.1595	0.1448
25.06	13.055	0.0947	1.236309	5.077	0.2026	1.0286	0.1347	0.1089
20.03	13.042	0.1194	1.557215	5.019	0.2506	1.2578	0.2264	0.1454
14.96	13.021	0.1557	2.02737	4.975	0.3326	1.6548	0.2996	0.1478
12.47	13.047	0.1880	2.452836	5.078	0.4073	2.0682	0.3117	0.1271
9.99	13.043	0.2261	2.949022	4.963	0.4968	2.4655	0.4106	0.1392
7.53	12.978	0.2992	3.883018	4.999	0.6637	3.3180	0.4920	0.1267
5.10	12.925	0.4334	5.601695	4.927	0.9654	4.7561	0.7726	0.1379
3.35	12.765	0.6621	8.451707	4.860	1.4491	7.0426	1.3361	0.1581
2.52	12.740	0.8694	11.07616	4.729	1.8746	8.8650	2.1382	0.1930
1.70	12.620	1.3020	16.43124	4.633	2.7254	12.6262	3.7320	0.2271
							Avg Efficiencies	
							84.94%	

5.2 POWER EFFICIENCIES

Once the board was trimmed, the boards overall efficiency needed to be determined. The efficiency was needed to determine the battery capacity. Having an exact efficiency allows for the battery to be sized appropriately, large enough to provide power but small enough to not increase weight un-necessarily. After determining the efficiency of the power distribution module, it would be possible to determine if weight would need to be reevaluated. The biggest influence on the boards efficiency was the switching frequency. If the efficiency was determined to be too low, the switching frequency could be lowered, providing a little more efficiency.

The verification process collected input and output voltage and currents. From that information the input and output power was calculated. The power distribution module, over the range of load currents, had an average efficiency of 82%. The efficiency of the power distribution module tracked with the circuit that was constructed. The data sheet gave an average efficiency of ~95% for the test condition, or 5% losses. Most of the losses with a switching-regulator are due to switching losses. The data sheet had the

losses shown at a switching frequency of 700kHz. The power distribution module was operating at 2.15 MHz, or approximately 3 times faster than the datasheet. With 3 times the switching frequency, the switching losses should be about 3 times larger, approximately 15%, or 85% efficient.

The final part of the process was determining where the remaining 3% of the power was lost. An audit of all the components within the power distribution module, excluding the switching-regulator, revealed the missing power. The various resistors within the power distribution module and the indicator LED added up to ~2.8% of the power used. The power distribution module has an average efficiency of 82%. The efficiency of the switching regulator did create some thermal concerns. Those concerns were mitigated by the large ground plane connected to pad 17 of the switching regulator. The large ground plane connected to the chips acts like a heat sink, drawing the thermal load away from the chip. All of the internal losses and efficiency calculations were built into tables and can be found below in Table 5.3 and 5.4.

Table 5.3: Internal Losses

Device	Resistance	Voltage	Power
R1	1,000,000	3.930	0.00002
R2	4,999	1.662	0.00055
R3	33,200	11.120	0.00372
R4	229,000	0.969	0.00000
R5	1,000	3.320	0.01102
R6	16,500	0.968	0.00006
LED	0	1.920	0.05760
	Power Total		
	0.07298		

Table 5.4: Efficiency Table

Input	Output	Power Lost (Input/Output)							
Power (Watts)	Power (Watts)	Total (Watts)	Percent	Switching Losses	Percent				
1.101391	0.8689	0.2325	0.2111	0.1595	0.1448				
1.236309	1.0286	0.2077	0.1680	0.1347	0.1089				
1.557215	1.2578	0.2994	0.1923	0.2264	0.1454	Avg Efficiencies		Avg Efficiencies	
2.02737	1.6548	0.3726	0.1838	0.2996	0.1478	82.09%		84.94%	
2.452836	2.0682	0.3846	0.1568	0.3117	0.1271				
2.949022	2.4655	0.4836	0.1640	0.4106	0.1392	Internal Losses			2.86%
3.883018	3.3180	0.5650	0.1455	0.4920	0.1267				
5.601695	4.7561	0.8456	0.1509	0.7726	0.1379				
8.451707	7.0426	1.4091	0.1667	1.3361	0.1581				
11.07616	8.8650	2.2112	0.1996	2.1382	0.1930				
16.43124	12.6262	3.8050	0.2316	3.7320	0.2271				

5.3 DISCUSSION

There was a need to keep the aircraft electronics as light as possible. One way to save weight was to make every component as efficient as possible. For example the power distribution board switching frequency. The higher the frequency the lower the effecincy. While testing the board the switching frequency was only set as low as possible while maintaining functionality. Another step to increase the performance was the removeal of excess weight, for example each cable was trimmed to be just long enough to make the connection. While measures like trimming cable might seem unnecessary, every little step to help efficiency helps the system. Allowing one component to go unscrutanized would lower the systems overall performance. Figure 5.4 found below, shows the entire setup when mounted into the airframe.

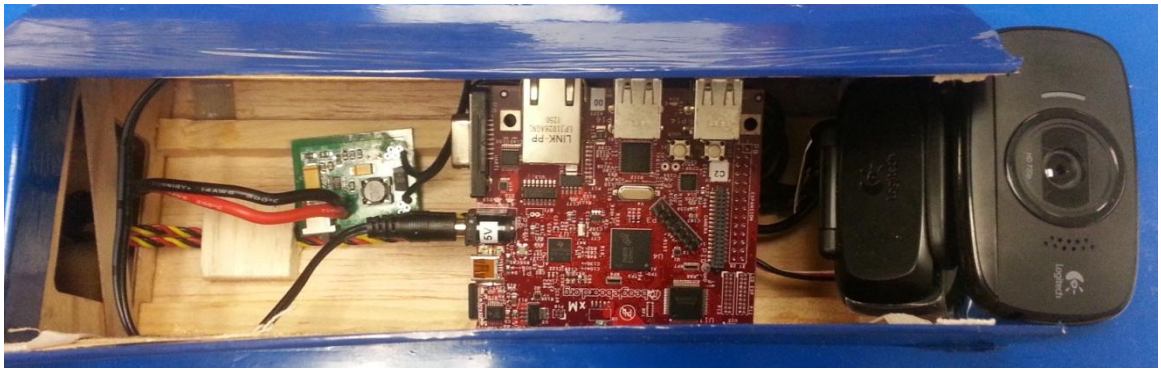

Figure 5.4: Aircraft Setup Table

6. IMAGE SET FOR TARGET IDENTIFICATION

A library of images was created to provide a variety of scenes and targets in a complex, cluttered environment. The intent of the library is to be a resource for testing image processing approaches to search-and-rescue target identification. The images are representative of aerial views from low altitude as could be obtained from a UAV camera platform. The Image Set section describes the image environments, the targets, and the library. Further information on the image sets is given in Appendix D.

6.1 OVERVIEW OF ENVIRONMENT AND TARGETS

The aerial images were obtained for lightly wooded pastureland. They contain a variety of non-target features including sky, shadows, tree lines, isolated trees, brush, roads, and ponds. The images were taken during the summer of 2013 in two locations near Rolla, Missouri. The camera altitude ranged up to 152.4 m (500 ft.). Some of the images contain single or multiple targets and other images contain no targets.

Targets related to recent human activity were placed in the environment for a subset of the images. These targets include person or persons, ground-to-air signals, vehicles, shelter, clothing articles, and bags. Figure 6.1 shows examples of these targets (image LOC1-PRSN-00005 from the set location one). The red circle in the center of the image highlights a person. A vehicle is shown directly above the person. The two red squares highlight ground-to-air rescue signals, i.e. an “X” mowed in the grass and an “X” made of tree branches. The red triangle show a lean-to structure made of tree branches. Note the image cluttering that includes sky in the lower left-hand corner of the image and the fisheye effect of the camera lens.

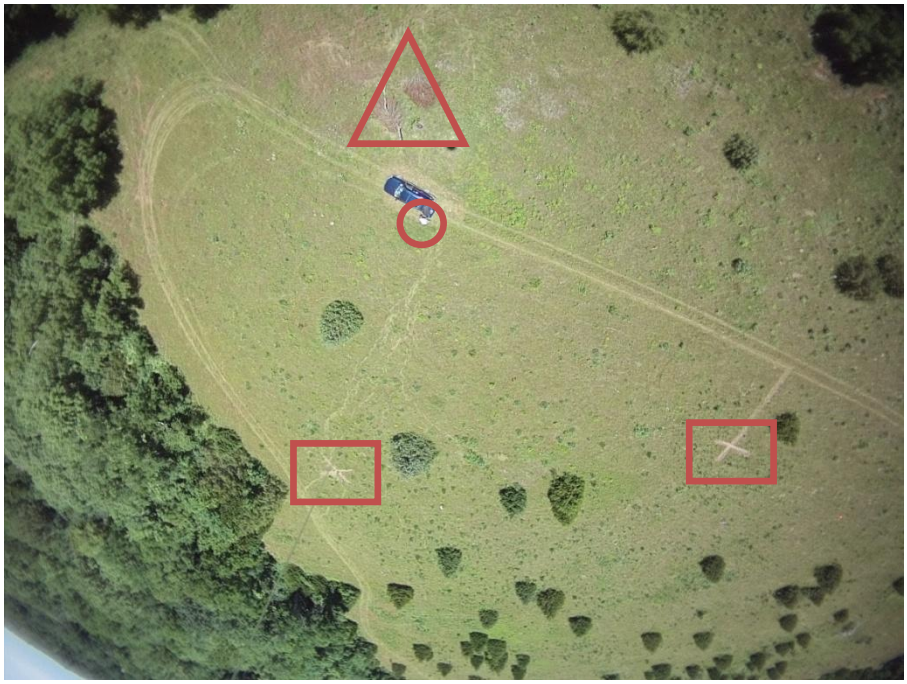

Figure 6.1: Example Targets in the Environment (LOC1-PRSN-00005)

The next image in Figure 6.2 (image LOC1-CLTH-00004 from the set location one) shows more subtle targets from a lower altitude. The red circle highlights a green shirt. The red square highlights a blue bag. Note that the image variation included a different altitude and rotational orientation. These differences further complicate any image processing.

Figure 6.2: Example Target Image (LOC1-CLTH-00004)

The noise categories identified for the project were the following: sky, altitude, rotation, shadowing, and lens effects. Also, note that the scene includes include shades of green, brown, and blue; varied surface textures; and non-target features such as roads. Each of the types of noise created a different problem for processing the images.

6.2 IMAGE COLLECTION

The images were obtained using a lighter-than-air balloon to lift the image collection system. The five foot-diameter balloon was a latex helium balloon. It provided a safe, stable platform to collect a large quantity of images without the challenge and expense of operating a remote-controlled aircraft. There were two balloon configurations used for the project. One configuration to fly the developed system used two balloons to provide enough lift. The other configuration used a single balloon to fly a

Contour GPS unit. Figure 6.3 shows the two balloon system in operation. The camera system could freely rotate so the image computer orientations are random. The balloon was flown over the area collecting images with a ground tether. Some of the images have rescue signs in them and some images have nothing but nature. These empty frames that contain no intentional target, is a control for false positives from target recognition processing. Each location was photographed at various times of day. (Later work may add images from different seasons to further expand the library variability)

Figure 6.3: Image Collection using a Double Balloon.

The balloon payload consisted of five components total, all shown in Figure 6.4. The main part was the BeagleBoard-Xm. The BeagleBoard was the computer that ran the code to collect the images. Another component was the custom power distribution board. The power system took the DC power from the batteries, another component, and provided a constant 5 volts output to power the BeagleBoard. The batteries for the system

were custom constructed LiPo packs. Each battery pack operated at 12.6volts and had a capacity of 5000mAh. The other two components were a GPS module, used to get the exact location of the imaging system, and the Logitech C510, used to capture the images. The system weighed 0.204kgs (0.45lbs) total. Along with the hardware used to capture images, there was an addition to the JPEG format used to store the images. The addition to the JPEG images was the EXIF add-on. The EXIF format allows for a variety of information to be stored within the image. The information that can be stored ranges from exposure information to GPS location information. Figure 6.4 shows the camera system.

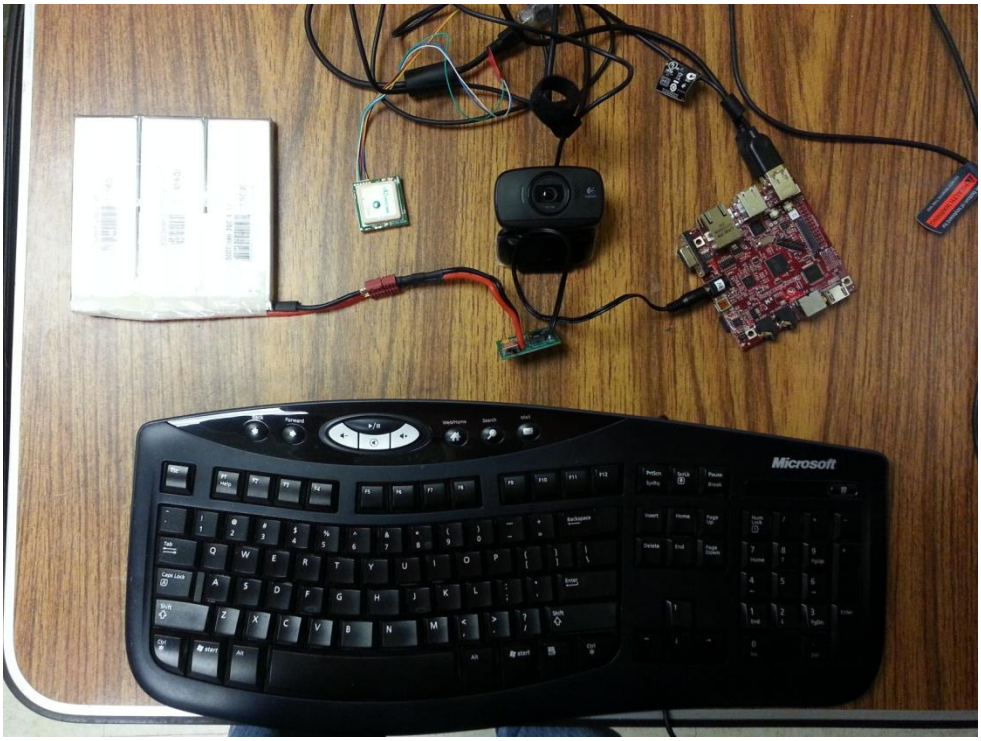

Figure 6.4: Camera for Image Collection.

An alternate system was also built around a ContourGPS camera. The ContourGPS camera was used to capture a number of the images due to the lower cost of operation. The ContourGPS only took a single balloon to fly. The downside to using the

ContourGPS camera was the presence of the fisheye on the lens. Other than the lens effect, the camera provided a simpler and more cost effective option compared to the system built. Both systems were used to collect images. Those images including a fisheye effect were taken with the ContourGPS system.

6.3 DESCRIPTION OF IMAGE SETS

The image library is a collection of images that might be seen from the air when searching for lost people. Some of the images include a number of rescue signs. The altitudes ranged between 0 and 152.4 meters (0 to 500ft). The image set was created over a variety of terrains but the primary description of the terrain would be sparsely wooded pastureland. In the various locations a number of rescue signs were erected/placed into the environment and then photographed and geotagged. The various targets used are all described below and shown in a few example images.

The images are classified by target type for single target scenes or the highest priority target for multiple target scenes. The order of the target priority was person, signal, vehicle, shelter, clothing, bags, and none. If an image with a person, that image is placed into the person set regardless of whether there are other target types present. If an image contains both a signal and a shelter without a person or vehicle, the image would be classified in the signal set. For instance, the image in Figure 6.1 is in the person set and the image in Figure 6.2 is in the clothing set. An image without a target is the seventh non-target subset.

For verification of the ability of a target recognition approach to detect a target and not falsely trigger, the image library will need to be searched to find a similar target and empty images. The tree lines and roads can assist with matching two images.

Each of the images was classified first by the location the image was taken and second by the highest priority target in the image. Setting the location information was done by the geographic difference and the time difference. For example a set of images taken in the same geographic location but on different days is classified as different locations. The images for the project were taken across two locations. After setting the location name, the image is tagged with the highest priority target. The target list in order from highest to lowest priority is a person, Ground-to-Air Signal, Vehicle, Shelter, Clothing, Backpacks, and Nothing. After classifying the photo by the location and the target type, the image is tagged with a sequential number ranging from 00001 to 99999. Using the sequential numbering for the images gives each photo a uniquely identifiable name while maintaining a naming scheme.

Figure 6.5 shows the highest priority target, a person, in the upper center of the image, filename LOC2-PRSN-00247, surrounded by a red circle. Figure 6.6 shows an example of a ground-to-air signal, image LOC2-SGNL-00009. Another target found in Figure 6.6 was an example of a vehicle. Figure 6.7, image LOC2-STRC-00005, shows an example of one of the shelters captured in the project. Figure 6.8 shows what an example of what clothing looks like in image LOC1-CLTH-00131. Also in Figure 6.8 is an example of a backpack. Appendix D gives further information and examples for the image sets. Figure 6.9 shows an example image with no targets.

Figure 6.5: Lake with Person

Figure 6.6: Field with Signal and Vehicle

Figure 6.7: Field with Shelter

Figure 6.8: Field with Clothing and Backpack

Figure 6.9: Field with No Target

7. IMAGE PPROCESSING FOR TARGET IDENTIFICATION

The image set includes a variety of targets in an outdoor cluttered background. Basic image processing strategies were applied to selected images as examples approaches to facilitate target identification. One common feature of many of the images was the presence of sky. Images containing sky have a potential noise source, e.g. the sky could have similar color to a blue vehicle. A sky-removal algorithm was applied to preprocess the example images. Processing the images was done with both color thresholding and edge detection algorithms.

MATLAB was used to implement the image processing code. MATLAB has advantages of rapid, straightforward coding, especially with the image processing toolbox. Other programming languages could be used to optimize performance or tailor a system to a given hardware. The goal for the image processing was to detect some of the major target types within the image set. These target types include ground-to-air signals, vehicles, and clothing. All of the operational functions were built into the standard MATLAB image processing toolbox.

7.1 PREPROCESSING TO REMOVE SKY

The preprocessing algorithm of the image processing removed the sky from captured images. Consider image LOC1-VHCL-00267 from the set location one. In Figure 7.1, the original image has some sky in the bottom left corner and in the preprocessed image, the sky has been removed by replacing pixels with a minimum value (0).

Figure 7.1: Original Image and Final Preprocessed Image

The sky is a noise source from the perspective of target identification. No targets of interest are in the sky. Hence, removal of the sky prevents false positives and allows faster run time. A block diagram of the sky removal preprocessing code can be found in Figure 7.2. The algorithm consists of examining the blue image plane, applying a thresholding operation, and applying additional filtering operations. The altered image can then be processed for potential targets. The MATLAB code is given in Appendix E.1.

The process of thresholding the image involves a pixel by pixel operation across the entire image. The image first gets divided into the Red, Green ,and Blue component images. At each pixel, the blue pixel value is compared to the image threshold value. When the pixel value is above the threshold the value is set to one when the pixel value is below the threshold, the pixel was set to zero. The selection of the threshold value was done with a trial and error approach. In addition to the threshold function, there were two filtering functions used. The two filters used were the wiener2 function and the medfilt2 functions within MATLAB. The filters were adaptive noise-removal functions and a median filtering function respectively. Along with the filtering functions that were used,

the bwareaopen function was used to remove blobs that were below a certain size. The blob size was determined through trial and error, and finally settled on a 10,000 pixel blob. The final function that was used within the image processing was the imclearboarder function. The imclearborder function removes blobs that do not touch the border of the image. Using the imclearborder function removes the sky because blobs of a certain size that touch the border are sky. The blob must touch the border because there will not be a section of sky in the center of an image that does not touch a border.

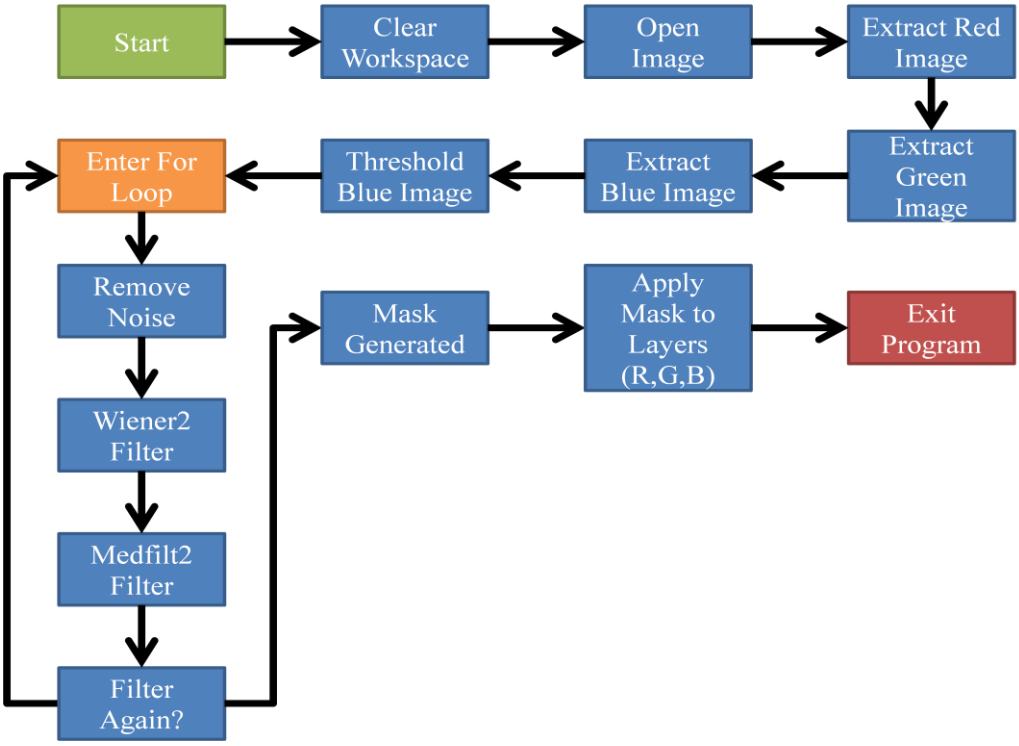

Figure 7.2: Sky Removal Code Flow Chart

Figures 7.3 through 7.11 show the step-by-step process of removing the sky from the image in Figure 7.2. Note that there are pixels with similar blue values to the sky, e.g. the blue Jeep Grand Cherokee. These pixels are eliminated from the sky mask by removing blobs with less than 10,000 pixels as in Figure 7.9 and by removing blobs not

on an edge as in Figure 7.10. The preprocessed image preserves all information on the ground.

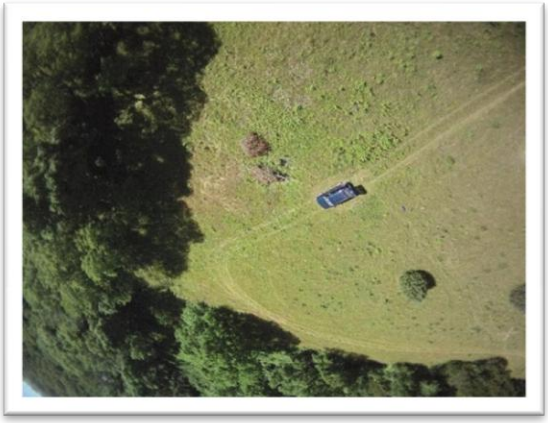

Figure 7.3: Original Image

Figure 7.4: Blue Image Plane

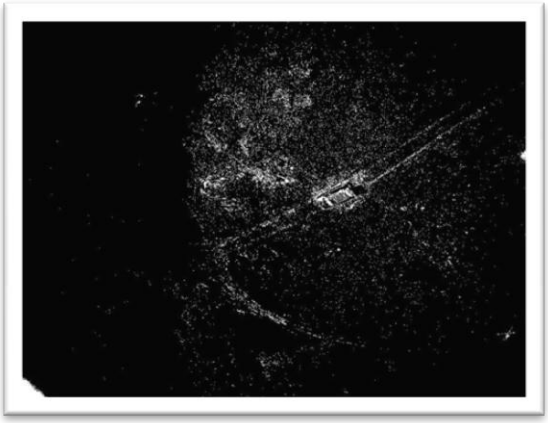

Figure 7.5: Black and White Image

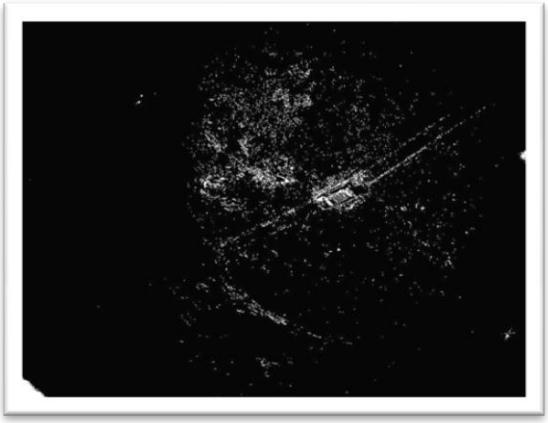

Figure 7.6: Remove <10 Pixel Noise

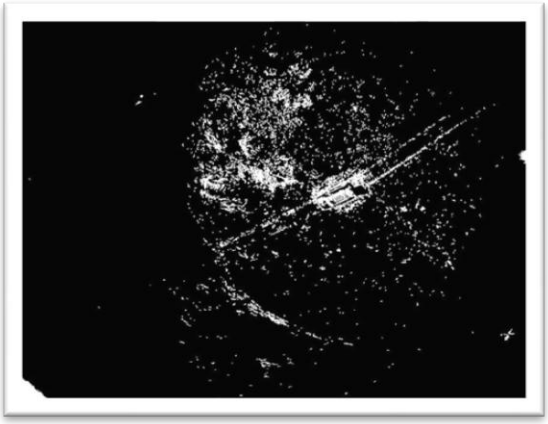

Figure 7.7: Wiener Filter 3X3 Area

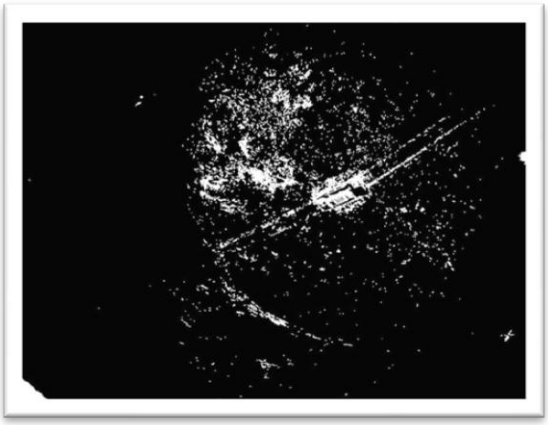

Figure 7.8: Median Filter 4X4 Area

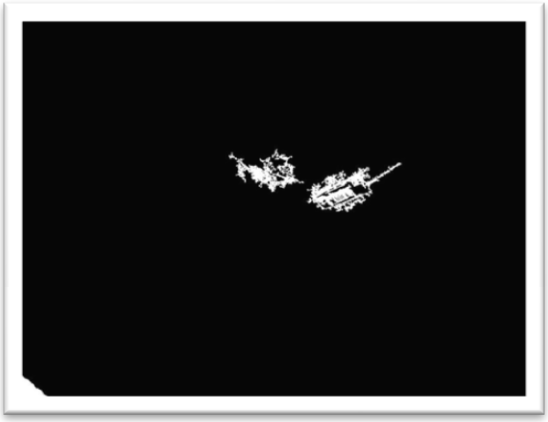

Figure 7.9: Remove Blobs <10K Pixels

Figure 7.10: Remove Blob Not on Edge

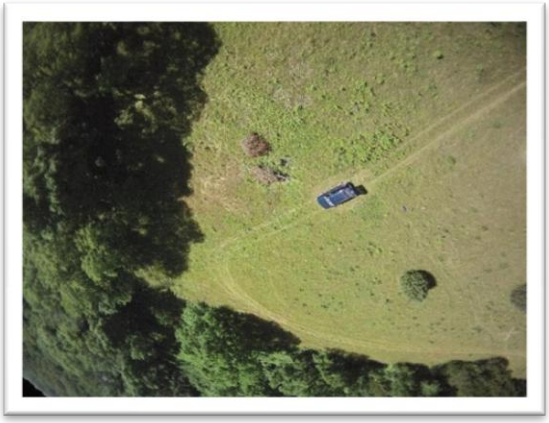

Figure 7.11: Final Image with Sky Removed

Figure 7.12 shows the algorithm applied to a) image LOC1-NONE-00933 from the set location one, b) image LOC1-CLTH-00789 from the set location one, c) image LOC1-PRSN-00008 from the set Location one, d) image LOC1-STRC-00024 from the set location one, e) image LOC1-SGNL-00056 from the set location one. The set of five images below shows the performance of the process of removing the sky from the images. As shown below the process has good results and the process was orientation invariant.

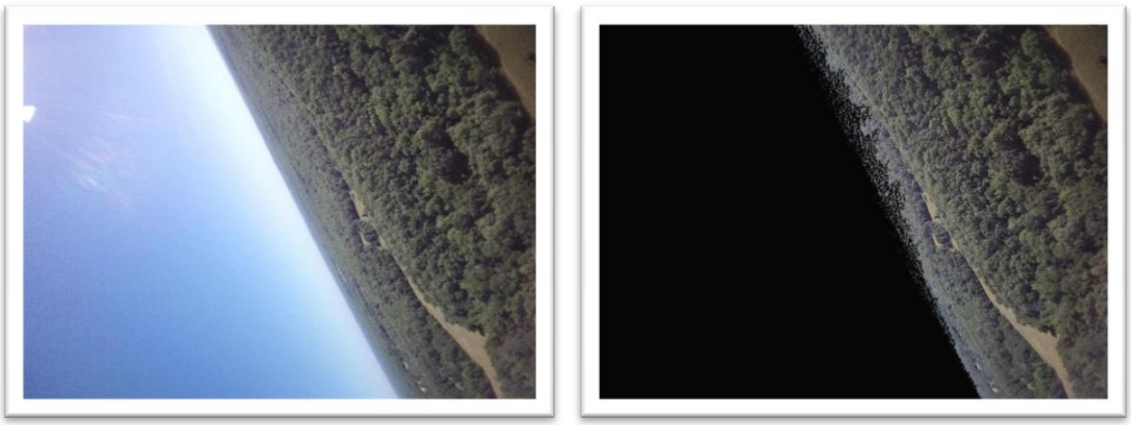

Figure 7.12 a

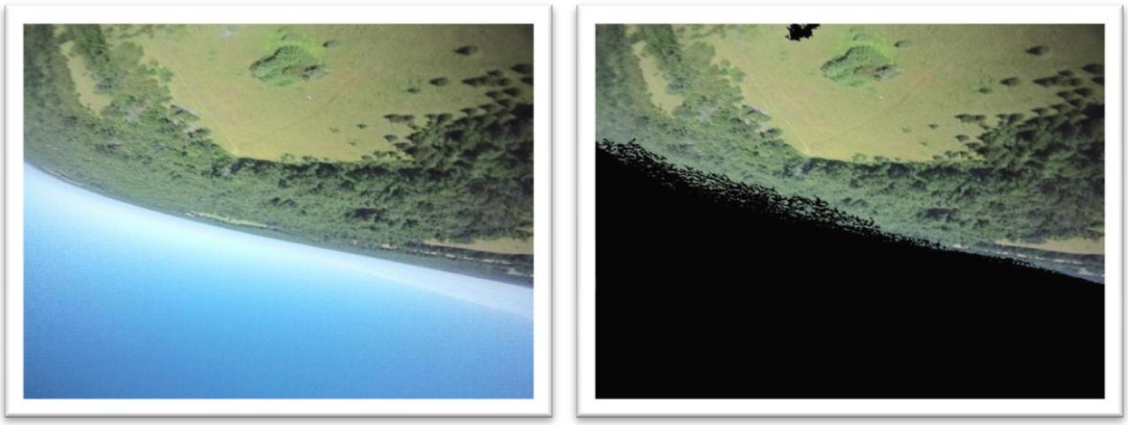

Figure 7.12 b

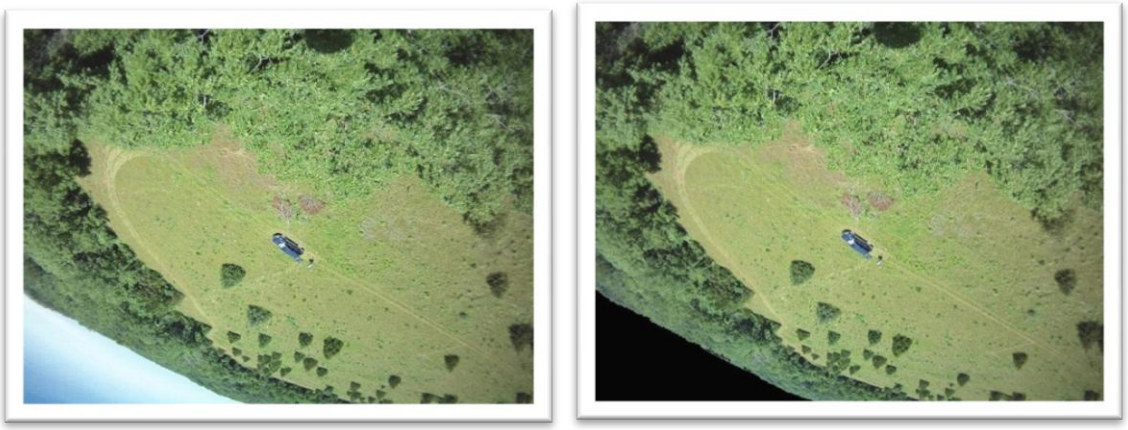

Figure 7.12 c

Figure 7.12: Examples of Sky removal on a) Image LOC1-NONE-00933 b) Image LOC1-CLTH-00789 c) Image LOC1-PRSN-00008 d) Image LOC1-STRC-00024 e) Image LOC1-SGNL-00056

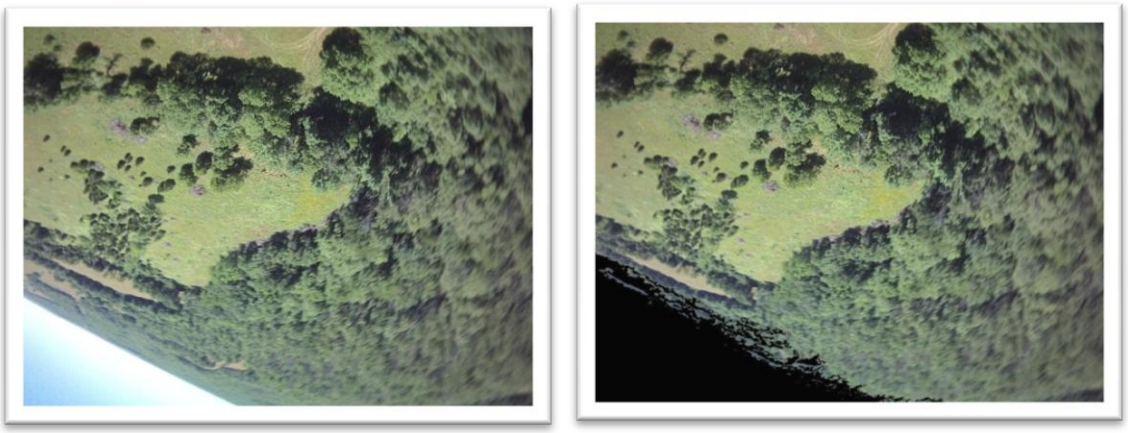

Figure 7.12 d

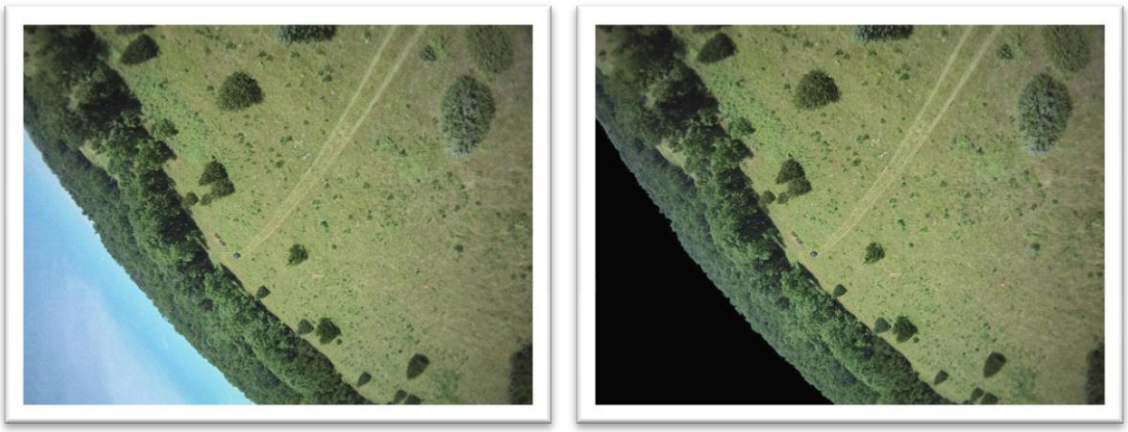

Figure 7.12 e

Figure 7.12: Examples of Sky removal on a) Image LOC1-NONE-00933 b) Image LOC1-CLTH-00789 c) Image LOC1-PRSN-00008 d) Image LOC1-STRC-00024 e) Image LOC1-SGNL-00056 (cont.)

7.2 TARGET PROCESSING BY COLOR THRESHOLDING: NON-GREEN OBJECTS

Color discrimination is a potential method for target identification. In the collected outdoor image set the dominant color was different shades of green. Image features of other colors are potential targets. Again using image LOC1-VHCL-00267 from the set location one, the desired target is a blue Jeep Grand Cherokee. The target is highlighted in the original image as shown in Figure 7.13. The approach can apply to remove green content to find non-green objects or, if a specific target is known, apply color thresholding for the known color, like a lost child's shirt color. Color thresholding for blue objects leaves just the Jeep target as shown in the final processed image. Only the desired target has a non-zero pixel values.

Figure 7.13: Original Image and Resulting Processed Image

A block diagram of the color thresholding processing code can be found in Figure 7.14. The sky removal steps are shown. The algorithm consists of examining the blue image plane after sky removal and iterating over a three stage filter multiple times. The number of iterations could eventually be linked to the altitude of the aircraft. After the filtering has been run *n* times, a mask was created to only show potential targets. The

mask is then applied to the multiple image layers and the layers were merged. For the example images, the multiple filters were applied over three iterations. Note that the loop count may be tailored for better performance as described earlier. In particular, increased altitude can reduce the number of loops run and vice-versa. MATLAB does not have the ability to read the EXIF information to obtain altitude values; other coding tools could retrieve information. The MATLAB code is given in Appendix E.2.

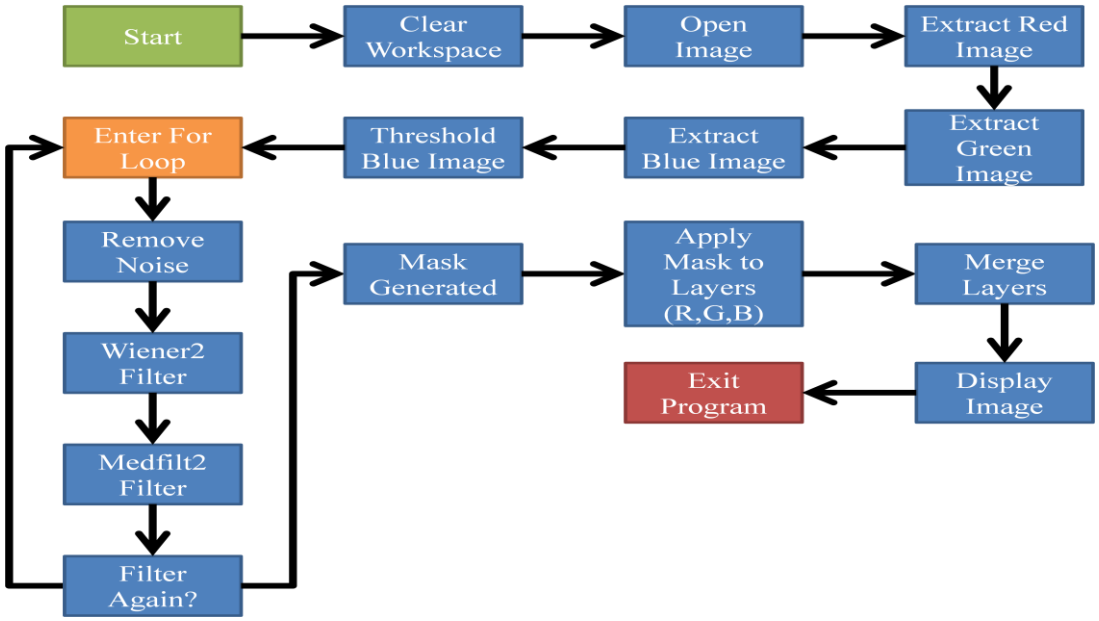

Figure 7.14: Color Thresholding Code Flow Chart

Figures 7.15 through 7.20 show the results of each step of processing for the image in Figure 7.15. The result of each filtering step removes a certain amount of noise. The first pass removes noise blobs smaller than 10^1, the next pass removes blobs smaller than 10^2, the process continues for each iteration. The result filter can change the blob detection size based on altitude. Note that the ground around the Jeep had some

blue color content that was not removed by the filtering due to the proximity of the Jeep blob.

Figure 7.15: Original Image

Figure 7.16: Blue Image Plane

Figure 7.17: First Filter Output

Figure 7.18: Second Filter Output

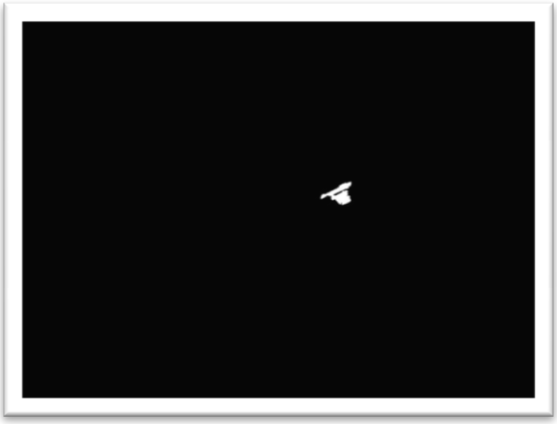

Figure 7.19: Third Filter Output

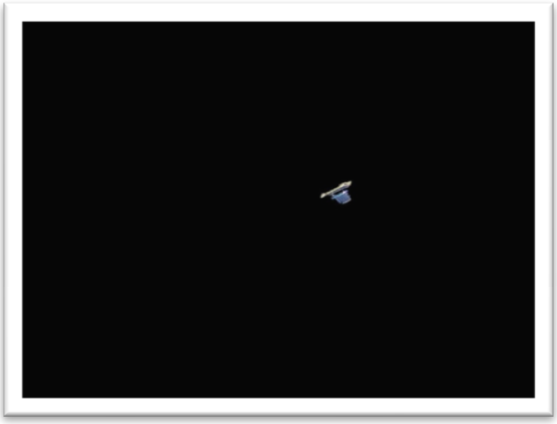

Figure 7.20: Filter Mask Original Image

Figure 7.21 shows the algorithm applied to a) image LOC1-SGNL-01302 from the set location one, b) image LOC1-STRC-00009 from the set location one, c) image LOC1-PRSN-00017 from the set location one, d) image LOC1-PRSN-00094 from the set location one, e) image LOC1-CLTH-00807 from the set location one.

For the purpose of the color thresholding, the same method was used for each of the example images. The color thresholding was done with the goal of lessening the intensity of natural colors. The thresholding was done to the Red Green Blue image plane. For each of the image planes the threshold was set to .55, .90, and .55 respectively. By removing the green from the environment, the colors that are less common in the scene are preserved.

Figure 7.21 a

Figure 7.21: Examples of Color Thesholding on a) LOC1-SGNL-01302 b) Image LOC1-STRC-00009 c) Image LOC1-PRSN-00017 d) Image LOC1-PRSN-00094 e) Image LOC1-CLTH-00807

Figure 7.21 b

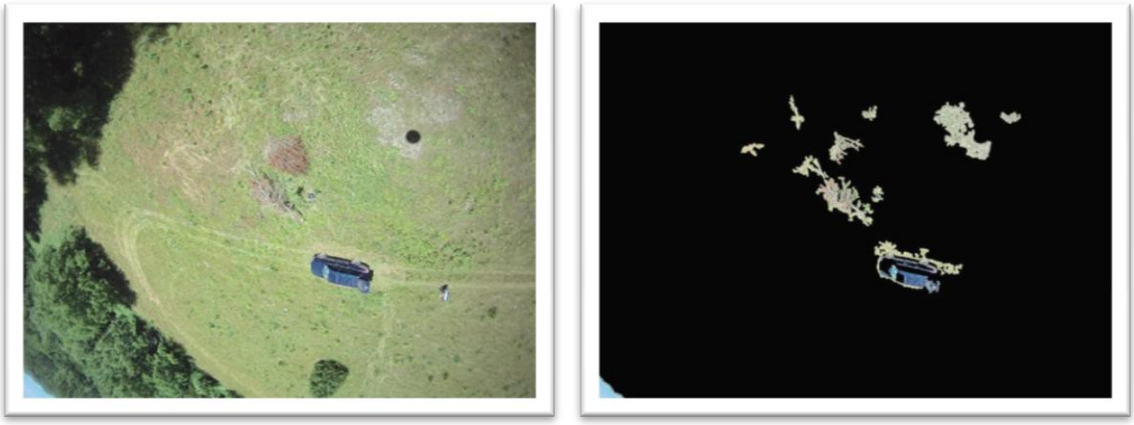

Figure 7.21 c

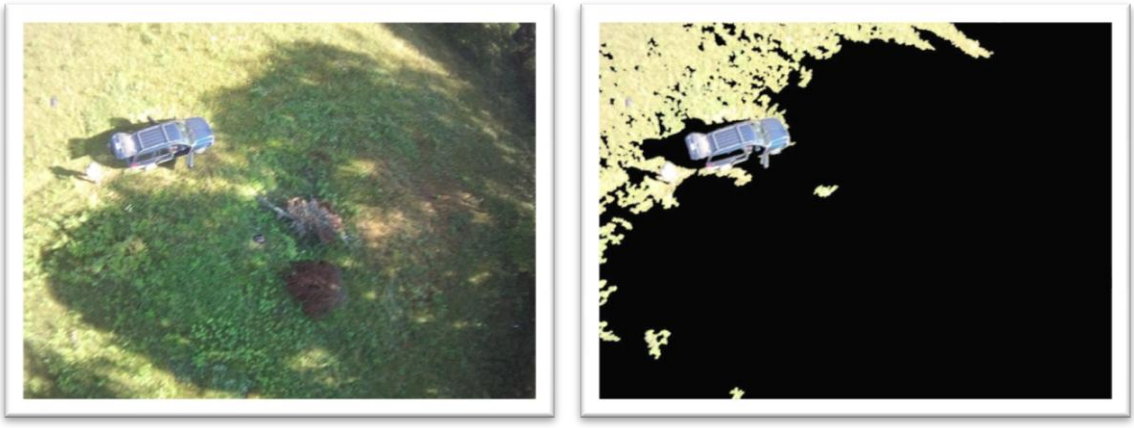

Figure 7.21 d

Figure 7.21: Examples of Color Thesholding on a) LOC1-SGNL-01302 b) Image LOC1-STRC-00009 c) Image LOC1-PRSN-00017 d) Image LOC1-PRSN-00094 e) Image LOC1-CLTH-00807 (cont.)

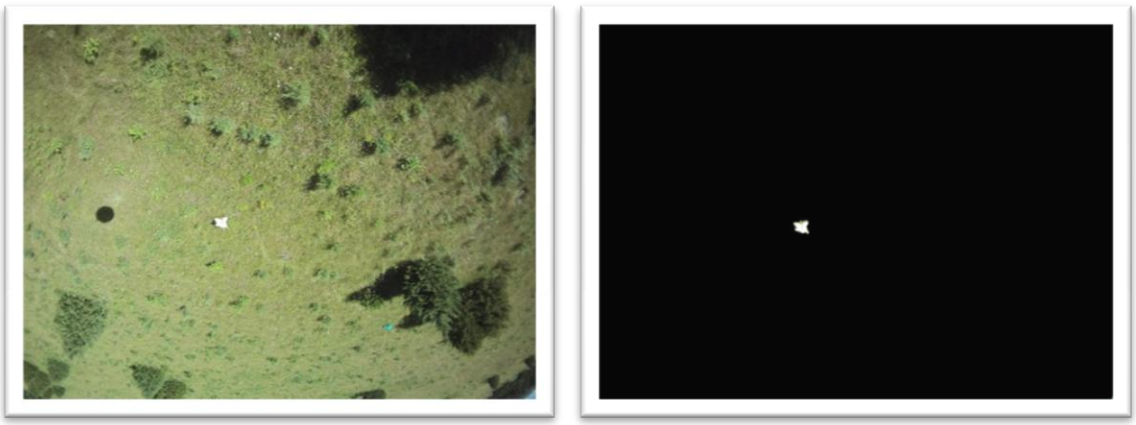

Figure 7.21 e

Figure 7.21: Examples of Color Thesholding on a) LOC1-SGNL-01302 b) Image LOC1-STRC-00009 c) Image LOC1-PRSN-00017 d) Image LOC1-PRSN-00094 e) Image LOC1-CLTH-00807 (cont.)

7.3 TARGET PROCESSING BY EDGE DETECTION: LINEAR OBJECTS

Edge detection is another method for target identification. Man-made features such as roads, vehicles, shelters, emergency signs, etc. often have different edge characteristics than natural objects. Some of these features, such as roads, are not useful for search-and-rescue situations, unless they assist in orienting the user to map locations. Other features may indicate the presence of a desired target. In particular, a ground-to-air signal of an "X" is the universal sign of distress for those lost in the wilderness. Consider image LOC1-SGNL-01302 from the set location one. Figure 7.22 shows the original image with an "X" mowed in the grass and the image after being processed to detect the edge of the "X". Note that the processing was set such that it eliminated the linear road object.

Figure 7.22: Original and Final Image

A block diagram of the edge detection processing code can be found in Figure 7.23. (The sky removal steps are not shown since they are unneeded for Figure 7.24. For a general image, sky removal would be included.) The process of edge detection of a ground-to-air signal was as follows. First the image was processed with a non-uniform

threshold. The non-uniform thresholding helps to remove the common colors from the image so non-natural combinations of colors stand out more. For the project, most of the green was removed from the image, leaving red and blue within the image. Then the image was threshold and filtered. The filters removed noise and smoothed the image. The filtered image was then edge detected, which outlines any blobs in the image. The Edged image was then filtered and averaged to smooth out the edges. Finally the image was displayed and can be seen in Figure 7.31. Figures 7.24 through 7.31 show the result of each step of the edge detection algorithm.

The reason the road was eliminated from the image but the "X" was not removed was as follows. The "X" was cut into a field to a point where the dirt was visible. The road was removed from the image because the road still had a significant amount of green. The reason it showed up in the images was due to the bending pattern of the grass. The bending did change the texture of the image but not the colors of the road.

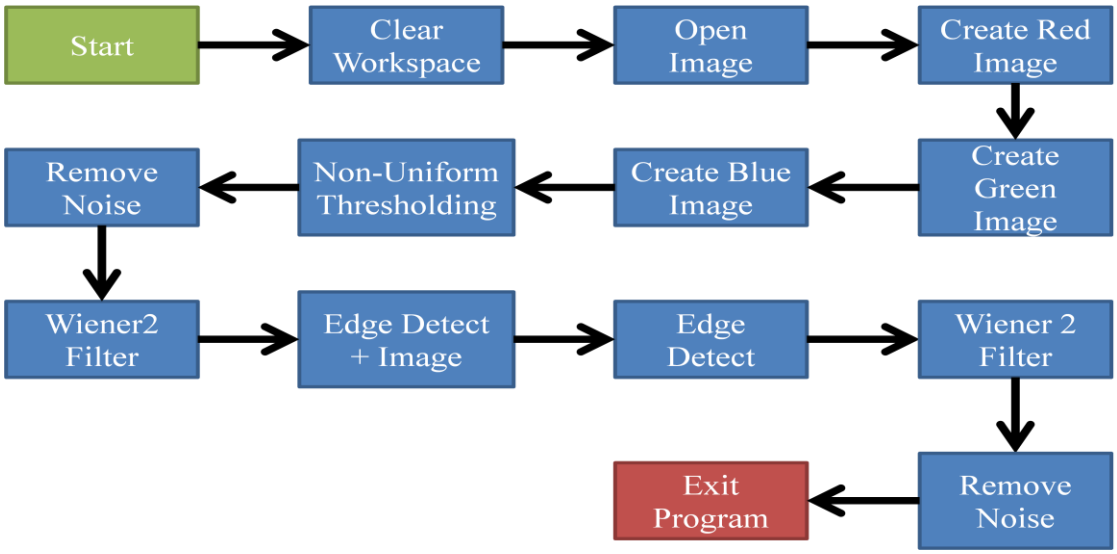

Figure 7.23: Edge Detection Processing

Figure 7.24: Original Image

Figure 7.25: Non-Uniform Thresholding

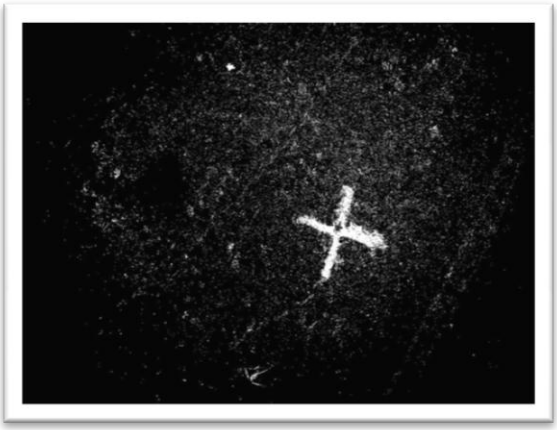

Figure 7.26: Black and White Threshold

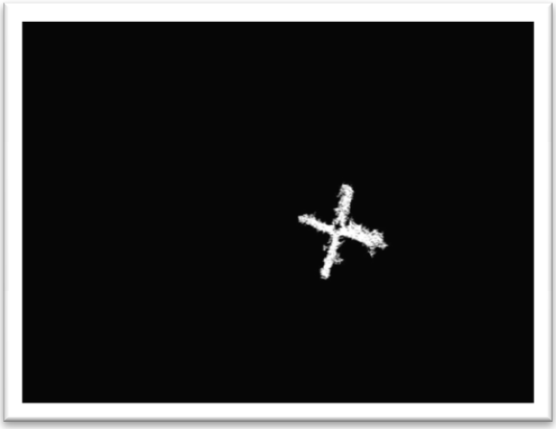

Figure 7.27: Remove Under 1000 Pixels

Figure 7.28: Wiener Filter

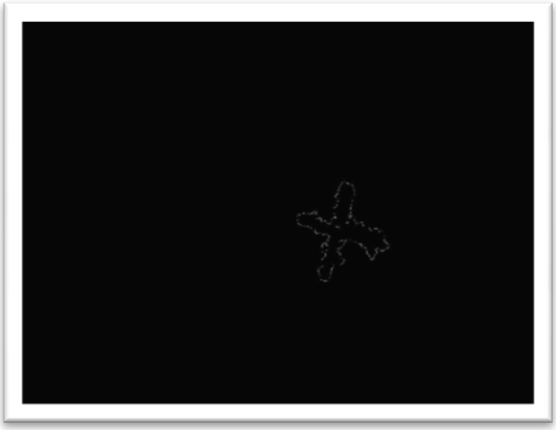

Figure 7.29: Edge Detection

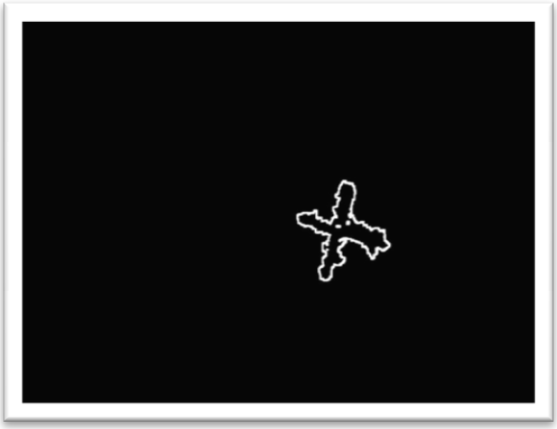

Figure 7.30: Wiener Filter Again

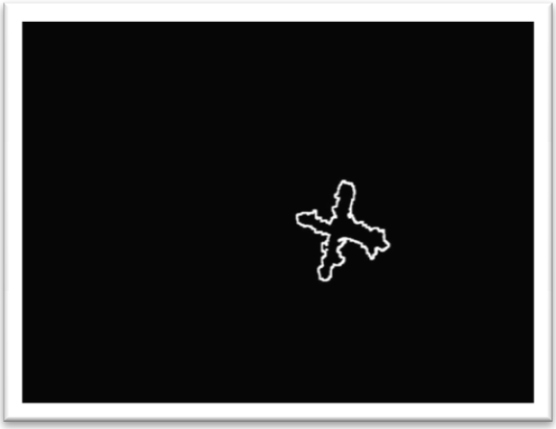

Figure 7.31: Under 1000 Pixel Removed

After designing the edge detection code, it was applied to various target types. Figure 7.32 show the before and after images for five different images using the edge detection algorithm. These images are a) image LOC1-VHCL-00267 from the set location one, b) image LOC1-PRSN-00159 from the set location one, c) image LOC1-SGNL-01241 from the set location one, d) image LOC1-CLTH-00803 from the set location one, e) image LOC1-CLTH-00807 from the set location one. Note that some of the images detect the target along with other noise in the image. Additional processing could be applied to identify an "X" or to match potential object from edge detection with a potential object from color thresholding. Also, the edge detection could be used to find differences at a particular location, e.g. the outline of an object is in a current image, but it is not in an earlier image.

Figure 7.32 a

Figure 7.32 b

Figure 7.32 c

Figure 7.32: Example Edge Detection Algorithm on a) Image LOC1-VHCL-00267 b) Image LOC1-PRSN-00159 c) Image LOC1-SGNL-01241 d) Image LOC1-CLTH-00803 e) Image LOC1-CLTH-00807

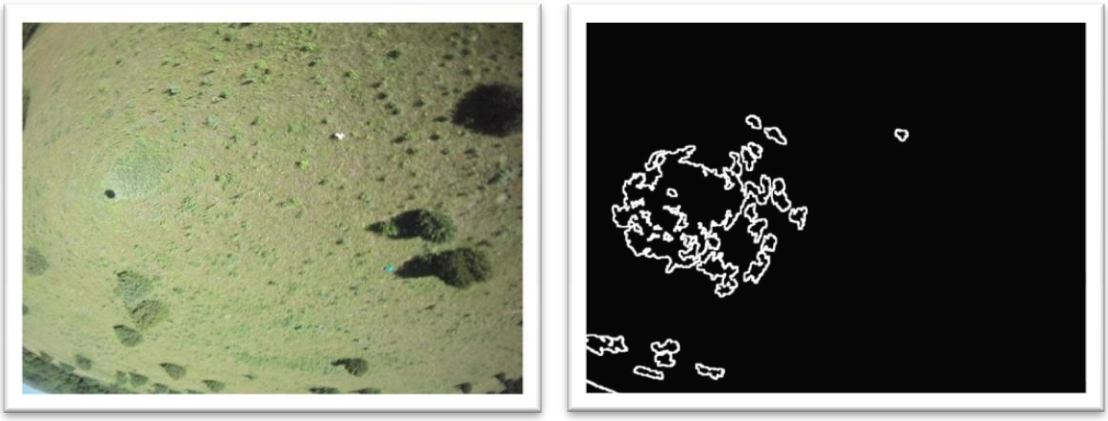

Figure 7.32 d

Figure 7.32 e

Figure 7.32: Example Edge Detection Algorithm on a) Image LOC1-VHCL-00267 b) Image LOC1-PRSN-00159 c) Image LOC1-SGNL-01241 d) Image LOC1-CLTH-00803 e) Image LOC1-CLTH-00807 (cont.)

7.4 TARGET PROCESSING BY TEXTURE

Some processing was attempted using textures functions within MATLAB. The thought was that textures like clothing would appear vastly different from the texture of a tree or field. The problem encountered in the testing was a step tradeoff. The first problem with trying to see clothing in a field was to see the small target, the area processed needed to be sensitive to small objects. That sensitivity led to problems when the large trees were processed because they were giant targets every time. If the texture

functions were changed to process a larger area, in an effort to not detect trees, the clothing and other targets were processed over because they were two small.

7.5 ADVANCED TARGET PROCESSING

With an overview of each of the basic target processing techniques seen here, it is now possible to discuss some advanced methods. The thing to note is that any processing technique will not be 100% correct all the time. The entire idea behind the processing involves high percentages and being able to account for the times the processing is wrong.

One way to get a higher degree of accuracy would be to have the aircraft perform sweeps. Starting at a high altitude, the aircraft could go up using a processing method, and detect any points of interest. Then descending in altitude, the aircraft can pass over the points and process the new images with some other processing method suited for the target. The process can be repeated multiple times to increase the likelihood of finding a target.

A part of the image processing that should not be overlooked would be advanced information. If for example you are searching for someone with a red shirt on, using processing methods that have high returns with red objects help to find the person. Similarly, if you are searching for a blue car, use techniques that have been proven to work on blue. Conversely, if you are searching an environment that has a flower that is bright yellow, you might not trigger on collections of that flower. Information about the target and environment allows for tuning of the processing algorithms.

Another method of advanced processing would be to filter the image with either a high pass or low pass filter depending on what qualities are being searched for. Another few operational methods would be to use dilation or erosion on the image to either expand or shrink bright areas. Another filtering technique could be using sigma filters. There are a number of advanced processing methods, and combinations of those methods that could result in very high accuracy results.

8. SUMMARY

This thesis addressed four tasks related to image processing for UAV search-and-rescue applications. First, a system for capturing images and annotating with GPS information was developed. Second, a power distribution module was developed for a light-weight imaging system that was suited for UAV and other similar applications. Third, the most significant accomplishment of this project was the creation of an image database of aerial search-and-rescue images. The images are in a cluttered outdoor environment and they contain varied noise sources for target recognition processing. The database will allow the further development of image processing software. The database will also allow design teams and student organizations to work on parts of design competitions, like the outback challenge, in parallel with the development of the airframe. Fourth, basic image processing techniques were applied to the database.

The image database consist of 8271 images. Some of the images contain a target or targets including people, signals, vehicles, shelters, clothing, and backpacks. Some images have no targets. Processing challenges include handling sky pixels, rotational orientation, shadowing, and fisheye lens distortion.

The sky noise was eliminated do to the challenge it created with color thresholding. The sky would look like a large blob of non-green and that artifact would throw off the thresholding, e.g. the sky could be confused with a non-green target. Removing the large blue sky allows for the color detection algorithm to run with a low incidence of false positive on the sky. Additionally the sky was removed because there were no valid targets within the sky for this image set.

The next source of noise was the rotational independence of each image. If each image had the same compass orientation and perspective, the image processing could be much simpler. Since the targets in the images can be in any compass rotation and any perspective, processing the images was much more difficult. The rotational independence makes mask matching challenging and beyond the scope of the project. Also, the mask size would need to change based on the elevation of the image.

Another source of noise was the shadowing in a number of images from trees and targets. The shadowing in the images creates a problem with both color and edge processing methods. The shadows in the images can cause the edge detection to trigger and outline a tree, causing a false positive. The shadow can also obstruct a target.

The final main source of noise found in the image set was the fish eye effect of the camera. The lens on the camera provides a large field of view and fisheye distortion.. The distortion makes the outer edge of the images more difficult to process with regards to edge detection. Each of the sources of noise makes the processing of the images challenging. Removing the various sources of noise makes the processing much easier and less prone to error.

The basic processing done on the image set had decent results for the small number of images tested. The sky removal processing was applied to 100 images with varying amount of sky. The processing removed all of the sky on 83% of the images. The color thresholding was applied to 150 images with various targets and successfully detected 78% of the targets. Finally the edge detection was run on 100 images. The results were detection of the target at 76% but the detection of noise was 32%.

The work done in this thesis can be applied in a number of different ways. The image set could be expanded upon by student to include more seasons and land types. The images could be used in artificial intelligence and image processing classes as homework assignments and projects to explore the automatic detection of targets. The hardware development could be expanded in an embedded systems class to build a smaller and more efficient system. The possible academic applications for this project are only limited by the creativity of the person using the work. The image database will be made available to the educational community.

Each location was photographed at various times of day. More variability in season, time, and environment will help to improve the imaging software's universality. Other future work could involve using more powerful computers to run image capture program or adding more information into the EXIF tag. On the processing side of the problem, more advanced image processing techniques could be applied to the image set to try and find better results. On the hardware side of the problem, higher resolution cameras can be used to try and get better quality images. Similarly, a more accurate GPS unit could be implemented to provide higher accuracy location information.

APPENDIX A.

AIRFRAME SPECIFICATIONS

The project revolved around the design of an imaging system for use on an RC aircraft. The airframe was built by a group of aerospace students as a part of a senior design course. The senior design team consisted of Caleb Plumlee, Scott Replogle, Aaron Pederson, Andrew Graham, Nicholas Fehner, Christopher Ward, and Stuart Schneider The airframe was built to be very stable and have a moderate cruise speed. The airframe was powered by an FG-14B four cycle gasoline engine. The aircraft has a total weight of 16 lbs and a top speed of 35 knots.

The aircraft was designed with a conventional style having a straight leading edge , single propeller, and cargo bay located behind the landing gear. The wing uses a NACA 4415 profile and was made as a single piece. The materials used on the aircraft were balsa, spruce, and plywood, then the aircraft was skinned using monokote. The aircraft has an estimated range of 44km and an endurance of 1.73 hours. Both the range and flight time of the aircraft far exceed the minimum estimations of what would be required for the Outback Challenge. Figure A.1 shows the internal structure of the aircraft. Figure A.2 shows the placement of the fuel tank and payload within the airframe, the electronics get mounted aft of the payload.

Figure A.1: Airframe Dimensions

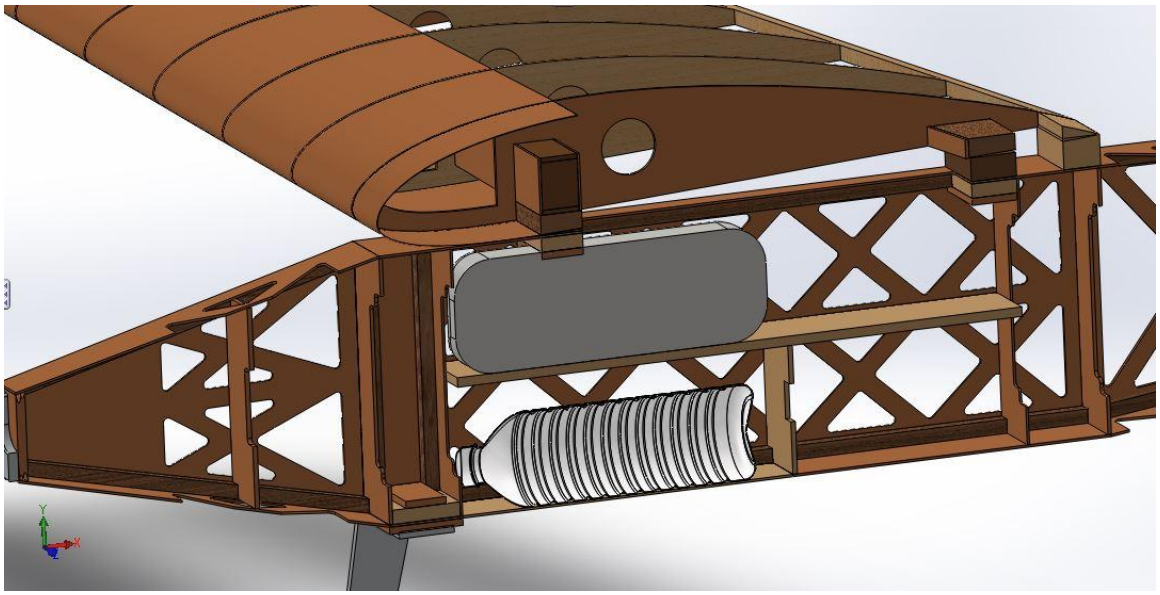

Figure A.2: Airframe Internal Payload

APPENDIX B.

IMAGE COLLECTION CODE

The image collection code consists of two main parts, the main and some accessory structures. The first part listed was the code used to structure the GPS information. The second code block contains the main file for the image collection code. The initial makefile system was setup by M.R.Bales as a part of his maters. The code was used with permission and modified extensively from what was provided[24]. The image collection code first initializes the camera, USB Drive, and GPS module, then the code enters a for loop. The for loop runs, capturing images, adding GPS annotation, and saving to the USB drive, until the program is stopped. All of the code was written in C++, there are no input arguments, and all files will be saved to the USB drive specified within the program.

```
#ifndef NMEA_h_
#define NMEA_h_

struct GPGGA {
   std::string timeStamp,HDOP;
   std::string latitude, latDirection;
   std::string longitude, lonDirection;
   int numberSatelites;
   int altitude;
   int fixQuality;
};

struct GPRMC {
   std::string timeStamp;
   bool validSignal;
   std::string latitude, latDirection;
   std::string longitude, lonDirection;
   std::string groundSpeed;
   int trueCourse;
   std::string dateStamp;
   double variation;
   char variationDirection;
   double checksum;
};
#endif
```

Code B.1: NMEA.h

```
// These libraries are needed for V4L2 API
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>
#include <getopt.h>
#include <fcntl.h>
#include <unistd.h>
#include <errno.h>
#include <malloc.h>
#include <sys/stat.h>
```

```
#include <sys/types.h>
#include <sys/time.h>
#include <sys/mman.h>
#include <sys/ioctl.h>
#include <asm/types.h>
#include <linux/videodev2.h>

// These libraries are needed for OpenCV API
#include "cv.h"
#include "highgui.h"

// These are used for the GPS Information
#include <termios.h>
#include <iostream>
#include <fstream>
#include <sstream>
#include <csignal>
#include "../Demo/NMEA.h" //NMEA Stentence Structures

// These libraries are needed for the EXIV Outputting
#include "exiv2/exiv2.hpp"

// This library is used for signals
//#include <signal.h>

// V4L2 device pointer
static int fd = -1;
// Name of device to open
static char *dev_name = NULL;

/***  START V4L2 Control Setup   ***/
struct v4l2_control control;
struct v4l2_queryctrl queryctrl;
struct v4l2_querymenu querymenu;

static void open_device(void)
{
        struct stat st;
        dev_name = "/dev/video0";

        if(-1 == stat(dev_name, &st)) {
                        fprintf(stderr, "Cannot identify '%s': %d, %s\n", dev_name, errno, strerror(errno));
                        exit(EXIT_FAILURE);
        }
        if(!S_ISCHR(st.st_mode)) {
                        fprintf(stderr, "%s is no device\n", dev_name);
                        exit(EXIT_FAILURE);
        }
        fd = open(dev_name, O_RDWR /*required*/ | O_NONBLOCK, 0);
        if(-1 == fd) {
                        fprintf(stderr, "Cannot open '%s': %d, %s\n", dev_name, errno, strerror(errno));
                        exit(EXIT_FAILURE);
        }
}

int main()
{
  IplImage *frame;
  CvCapture *capture;
  int counter = 0;
  char filename[32];
  char c;
  unsigned char *img;
  int width, height, step, channels;

  unsigned int bright_val = 100;
  unsigned int contrast_val = 30;
  unsigned int sat_val = 20;
  unsigned int auto_white_bal_val = 1;
  unsigned int white_bal_temp = 2800;
```

```
    unsigned int exposure_val = 333;
    int gain_val = 100;

    // Open device
    open_device();

    control.id = V4L2_CID_BRIGHTNESS;
    control.value = bright_val;
    ioctl(fd, VIDIOC_S_CTRL, &control);

    control.id = V4L2_CID_CONTRAST;
    control.value = contrast_val;
    ioctl(fd, VIDIOC_S_CTRL, &control);

    control.id = V4L2_CID_SATURATION;
    control.value = 20;
    ioctl(fd, VIDIOC_S_CTRL, &control);

    control.id = V4L2_CID_AUTO_WHITE_BALANCE;
    control.value = auto_white_bal_val;
    ioctl(fd, VIDIOC_S_CTRL, &control);

    control.id = V4L2_CID_WHITE_BALANCE_TEMPERATURE;
    control.value = white_bal_temp;
    ioctl(fd, VIDIOC_S_CTRL, &control);

    control.id = V4L2_CID_GAIN;
    control.value = gain_val;
    ioctl(fd, VIDIOC_S_CTRL, &control);

    // Exposure Controls
    control.id = V4L2_CID_EXPOSURE_AUTO;
    control.value = V4L2_EXPOSURE_AUTO; //V4L2_EXPOSURE_SHUTTER_PRIORITY;
    ioctl(fd, VIDIOC_S_CTRL, &control);

    /***   END V4L2 Control Setup   ***/

    /***   START GPS Setup ***/
    std::ifstream ifs;
    std::string sentence;
    ifs.open( "/dev/serial/by-id/usb-FTDI_FT232R_USB_UART_A600eH4L-if00-port0" , std::ifstream::in);
    if (!ifs.is_open() ) {
      perror("open_port: Unable to open /dev/ttyS# â€“ ");
    } else {
      printf("Port 1 has been sucessfully opened \n");
    }
    /***   END GPS Setup  ***/

    /***   START OpenCV Setup   ***/
    capture = cvCaptureFromCAM(0);

  /* Ask the camera what its max resolution is */
    int qWidth = (int)cvGetCaptureProperty( capture, CV_CAP_PROP_FRAME_WIDTH);
    int qHeight = (int)cvGetCaptureProperty( capture, CV_CAP_PROP_FRAME_HEIGHT);
    printf("Width %d Height %d\n",qWidth,qHeight);

  /* Explicitly Set the Cameras window size */
    cvSetCaptureProperty(capture, CV_CAP_PROP_FRAME_WIDTH, 1280);
    cvSetCaptureProperty(capture, CV_CAP_PROP_FRAME_HEIGHT, 720);

    frame = cvQueryFrame(capture);

    /* Get frame info */
    width = frame->width;
    height = frame->height;
    step = frame->widthStep;
    channels = frame->nChannels;
    printf("\n\rW=%u, H=%u, Step=%u, #Chan=%u' \n", width, height, step, channels);

    GPRMC currentGPRMCLocation;
```

```
GPGGA currentGPGGALocation;
currentGPRMCLocation.validSignal = 0; //Initialize the validSignal to 0 to prevent bad data

while( /* This Could Change to Interrupt Stop */ 1) {
  /*This method of frame grabbing grabs the frame quickly.
  This could be important when grabbing from several cameras.
  The frame is not decompressed since that would take time. */
  if(!cvGrabFrame(capture)) {
    printf("\n\rCould not capture frame\n");
    exit(0);
  }
  frame = cvRetrieveFrame(capture, 1);

  /* Move image into familiar byte format for manipulation */
  img = (unsigned char *)frame->imageData;

  for(int i=0;i<3;i++){
    std::getline( ifs, sentence);
    std::istringstream iss( sentence );
    std::string result;
    if( std::getline( iss, result , ',') ){
      // If result has a GPRMC string update the GPRMC information
      if( result == "$GPRMC") {
        GPRMC tempGPRMC;
        std::getline( iss, tempGPRMC.timeStamp , ',');
        std::getline( iss, result , ',');
        tempGPRMC.validSignal = 1;
        std::getline( iss, tempGPRMC.latitude , ',');
        tempGPRMC.latitude.erase((tempGPRMC.latitude.length()-5),1);
        std::getline( iss, tempGPRMC.latDirection , ',');
        std::getline( iss, tempGPRMC.longitude , ',');
        tempGPRMC.longitude.erase(tempGPRMC.longitude.length()-5,1);
        std::getline( iss, tempGPRMC.lonDirection, ',');
        std::getline( iss, tempGPRMC.groundSpeed, ',');
        std::getline( iss, tempGPRMC.dateStamp, ',');
        if(tempGPRMC.validSignal == 1) {
          currentGPRMCLocation = tempGPRMC;
        }else { std::cout << "Invalid GPS Data" << std::endl;}
      // If result has a GPGGA string update the GPGGA information
      } else if( result == "$GPGGA") {
        GPGGA tempGPGGA;
        std::getline( iss, tempGPGGA.timeStamp , ',');
        std::getline( iss, tempGPGGA.latitude , ',');
        tempGPGGA.latitude.erase((tempGPGGA.latitude.length()-5),1);
        std::getline( iss, tempGPGGA.latDirection , ',');
        std::getline( iss, tempGPGGA.longitude , ',');
        tempGPGGA.longitude.erase(tempGPGGA.longitude.length()-5,1);
        std::getline( iss, tempGPGGA.lonDirection, ',');
        std::getline( iss, result , ',');
        if( result[0] == '0') { tempGPGGA.fixQuality = 0; }
        else if ( result[0] == '1') { tempGPGGA.fixQuality= 1; }
        else { tempGPGGA.fixQuality= 2; }
        std::getline( iss, result , ',');
        tempGPGGA.numberSatelites = atoi(result.c_str());
        std::getline( iss, tempGPGGA.HDOP, ',');
        std::getline( iss, result , ',');
        result.erase(result.length()-2,1);
        tempGPGGA.altitude = atoi(result.c_str());
        currentGPGGALocation = tempGPGGA;
      } // Seen other strings
    }
  }

  sprintf( filename, "/media/FB00-3002/Images/image%d.png", counter++);
  cvSaveImage( filename, frame, 0);

  Exiv2::Image::AutoPtr image = Exiv2::ImageFactory::open(filename);
  assert(image.get() != 0 );

  Exiv2::ExifData exifData;
```

```
    if(currentGPRMCLocation.validSignal == 1) {
    // Could include a timestamp check for image vs GPS information
      exifData["Exif.GPSInfo.GPSLatitudeRef"] = currentGPRMCLocation.latDirection;
      exifData["Exif.GPSInfo.GPSLatitude"] = (atoi(currentGPRMCLocation.latitude.c_str()))/1;
      exifData["Exif.GPSInfo.GPSLongitudeRef"] = currentGPRMCLocation.lonDirection;
      exifData["Exif.GPSInfo.GPSLongitude"] = (atoi(currentGPRMCLocation.longitude.c_str()))/1;
      exifData["Exif.GPSInfo.GPSAltitude"] = (currentGPGGALocation.altitude)/1;
      exifData["Exif.GPSInfo.GPSStatus"] = 'A';
    }
    image->setExifData(exifData);
    image->writeMetadata();
  }

  printf("\n\rAll Done...\n\n");
  cvReleaseCapture(&capture);
  cvDestroyWindow("CamView");
  /***   END OpenCV Setup   ***/
  return 0;
}
```

Code B.2: Main.cpp

APPENDIX C.

IMAGE SET INFORMATION

The image set contains the date and size information for each location and the seven target files within each location. Each location has targets for all seven of the target types. The information below contains the size and quantity of each location subset.

Location 1 (LOC1): . . . . . 5,019 Images (12.9GB)

Date: June 13th, 2013

- Person (PRSN) . . . . . 439 Images (1.20GB)
- Ground-to-Air Signal (SGNL) . . . 1,906 Images (4.86GB)
- Vehicle (VHCL) . . . . . 277 Images (0.72GB)
- Shelter (STRC) . . . . . 33 Images (0.09GB)
- Clothing (CLTH) . . . . . 921 Images (2.32GB)
- Backpack (BAGS) . . . . . 25 Images (0.07GB)
- Nothing (NONE) . . . . . 1,418 Images (3.67GB)

Location 2 (LOC2): . . . . . 3,252 Images (8.89GB)

Date: July 19th, 2013

- Person (PRSN) . . . . . 2,648 Images (7.26GB)
- Ground-to-Air Signal (SGNL) . . . 15 Images (0.04GB)
- Vehicle (VHCL) . . . . . 81 Images (0.27GB)
- Shelter (STRC) . . . . . 33 Images (0.09GB)
- Clothing (CLTH) . . . . . 16 Images (0.04GB)
- Backpack (BAGS) . . . . . 0 Images (0.00GB)
- Nothing (NONE) . . . . . 459 Images (1.23GB)

APPENDIX D.

THE IMAGE SET

The images were categorized by a systematic naming convention. In addition to the organizational method, an overview of the types of images can be found for each location. The project collected over 8,000 images and needed a way to catalog the images for use later. The process of categorizing each photo entailed first grouping based on location and then grouping based on target type.

The location for the images, not only refers to the physical location but also the time of the flight. This means while two flights might have been in the same geographic location, if taken on separate days, the location will be classified differently. The location aspect was built into the first part of each filename by starting each image name with LOC#. The LOC indicates the location, and the number is incremented for each unique location.

Table D.1 Location Identification Abbreviations

Location	Date	Abbreviation
Site A	June 20th, 2013	LOC1
Site A	July 22nd, 2013	LOC2

The next part of each images name had to do with the targets in the image. The targets used for the project were classified into seven separate groups. The groups were, Person, Ground-to-Air Signal, Vehicle, Shelter, Clothing, Backpacks, None. Now the target types were also prioritized in the order listed previously. This means if an image contains a person and a vehicle, the image will be labeled with the person abbreviation. The table below lists each target type, priority from first to last, and the abbreviation for that target type. After adding the highest priority target to the name, the filename

resembles the following. LOC# - PRSN - #####.jpeg. The five hash tags after the target type are used to indicate the category can hold up to 100,000 images of a specific target type at a location.

Table D.2 Target Identification Abbreviations

Priority	Target	Abbreviation
1st	Person	PRSN
2nd	Signal	SGNL
3rd	Vehicle	VHCL
4th	Shelter	SHTR
5th	Clothing	CLTH
6th	Backpacks	BAGS
7th	Nothing	NONE

A few examples of the naming convention and the images associated with each name can be found below. The first image, LOC1-PRSN-00097, shows the highest priority target, a person, highlighted with a red circle. Additional targets in the image include a vehicle, shelter, and backpack. Due to the priority of the targets, the image was labeled with PRSN. The next image, Image LOC1-SGNL-00693, shows a ground-to-air signal. Also within the image are three pieces of clothing. Due to the priority of the naming convention, the image was labeled with SGNL as opposed to CLTH

Figure D.1: The image on the left, Image LOC1-PRSN-00097, contains the following targets, a person, vehicle, shelter, and backpack. The image on the right, Image LOC1-SGNL-00693, contains a single ground-to-air target and three pieces of clothing.

APPENDIX E.

IMAGE PROCESSING CODE

The image processing code is given for processing the various targets. Appendix E.1 shows the removal of any sky from a target image. Appendix E.2 which shows color thresholding of a vehicle, a Jeep Grand Cherokee, based on color and size within the image. Appendix E.3 which shows the code for locating a Ground-to-Air signal, specifically an "X", based on non-uniform color thresholding. All of the image processing was written in MATLAB R2010.a, installed with the basic image processing toolbox. The input parameters for each of the scripts was just an input image. The resulting displayed images were saved off from the figure displays.

E.1: REMOVING THE SKY

Subsection E.1 shows the code for sky removal. With no valid targets appearing in the sky, removing that part of the image reduces the chance of a false positive. The decision to process the sky away was also made to help with thresholding within the blue plane. If not removed, the sky can adversely affect the performance of other image processing methods detailed below. Figure E.1, shown below, has a before and after running code E.1 on example image LOC1-VHCL-00267

Figure E.1: Original Image and Final Image

```
clear all;
close all;
clc;

%Open file
files = dir('FILE1042.jpg'); % List files in a directory
if( ~isempty(files) > 0 )
   [cdata,map] = imread(files(1).name);
   if ~isempty( map )
      cdata = ind2rgb( cdata, map);
   end;
   figure('NumberTitle', 'off', 'Name', 'Original Image'); %label figure components
   imshow(cdata); %Display Image

%Convert to RGB Images
inputImg = cdata;
doubleImg = im2double(inputImg);
Image_Red = im2double(inputImg(:,:,1)); %Red frame to Double
Image_Green = im2double(inputImg(:,:,2)); %Green frame to Double
Image_Blue = im2double(inputImg(:,:,3)); %Blue frame to Double
figure('NumberTitle', 'off', 'Name', 'Blue Image Matrix'); %label figure components
imshow(Image_Blue);

%Sky and Horizon Elimination
Sky_Check = im2bw(Image_Blue, .50); %Threshold at a high value to get sky
figure('NumberTitle', 'off', 'Name', 'Image to Black and White'); %label figure components
imshow(Sky_Check);
Sky_Check = bwareaopen(Sky_Check,10); %Remove groups under 10 pixels
figure('NumberTitle', 'off', 'Name', 'Remove Noise under 10 Pixels Large'); %label figure components
imshow(Sky_Check);
Sky_Check = wiener2(Sky_Check,[3 3]); %Wiener filter image over 3x3 areas
figure('NumberTitle', 'off', 'Name', 'Wiener filter over 3x3 area'); %label figure components
imshow(Sky_Check);
Sky_Check = medfilt2(Sky_Check,[4 4]); %Median filter at 4x4 areas
figure('NumberTitle', 'off', 'Name', 'Median Filter over 4x4 area'); %label figure components
imshow(Sky_Check);
Sky_Check = bwareaopen(Sky_Check,10000); %Find groups larger than 10,000 pixels
figure('NumberTitle', 'off', 'Name', 'remove blobs under 10,000 pixels'); %label figure components
imshow(Sky_Check);
Sky_Check = im2double(Sky_Check); %Convert to a double
Sky_Check = Sky_Check - imclearborder(Sky_Check);
figure('NumberTitle', 'off', 'Name', 'Subtrack Large Blobs that touch the Border'); %label figure components
imshow(Sky_Check);

%Remove Sky from all RGB Component Images
if(NUM > 0) %Only apply Wiener2 filter if there are blobs
Image_Red = Image_Red.*~Sky_Check;
Image_Green = Image_Green.*~Sky_Check;
Image_Blue = Image_Blue.*~Sky_Check;
end;
figure('NumberTitle', 'off', 'Name', 'Sky Free Image'); %label figure components
imshow(cat(3,Image_Red,Image_Green,Image_Blue));
```

Code E.1: Sky Removal

E.2: COLOR THESHOLDING

Subsection E.2 shows the detection of a vehicle. One reason people can become stranded is the failure of a vehicle, being able to detect that vehicle can help to locate the stranded persons. Figure E.2 shows the before and after of running code E.2, color thresholding, on an example image.

Figure E.2: Original Image and Target Image

```
clear all;
close all;
clc;

%Open file
files = dir('FILE1042.jpg'); % List files in a directory
if( ~isempty(files) > 0 )
   [cdata,map] = imread(files(1).name);
   if ~isempty( map )
      cdata = ind2rgb( cdata, map);
   end;
   figure('NumberTitle', 'off', 'Name', 'Original Image'); %label figure components
   imshow(cdata); %Display Image

%Convert to RGB Images
inputImg = cdata;
doubleImg = im2double(inputImg);
Image_Red = im2double(inputImg(:,:,1)); %Red frame to Double
Image_Green = im2double(inputImg(:,:,2)); %Green frame to Double
Image_Blue = im2double(inputImg(:,:,3)); %Blue frame to Double
%Large Primairly Blue Targets
Blue_Check = im2bw(Image_Blue, .65);
for i=1:3
   Blue_Check = bwareaopen(Blue_Check,2*10^i);
   [l NUM] = bwlabel(Blue_Check,4); %Get Number of Blobs
   clearvars l;
   if(NUM > 0) %Only apply Wiener2 filter if there are blobs
      Blue_Check = wiener2(Blue_Check,[3 3]);
```

```
    end;
    Blue_Check = medfilt2(Blue_Check,[4 4]);
    figure();
    imshow(Blue_Check);
end;
%Apply Filter to all image slices
Image_Red = Blue_Check.*Image_Red;
Image_Green = Blue_Check.*Image_Green;
Image_Blue = Blue_Check.*Image_Blue;
figure('NumberTitle', 'off', 'Name', 'Red Filtered Frame'); %label figure components
imshow(Image_Red);
figure('NumberTitle', 'off', 'Name', 'Green Filtered Frame'); %label figure components
imshow(Image_Green);
figure('NumberTitle', 'off', 'Name', 'Blue Filtered Frame'); %label figure components
imshow(Image_Blue);

Target = cat(3,Image_Red,Image_Green,Image_Blue);
figure('NumberTitle', 'off', 'Name', 'Large Blue Vehicle Target'); %label figure components
imshow(Target);
```

Code E.2 Color Thresholding

E.3: EDGE DETECTION

Subsection 4.3 contains the code for locating a rescue sign, an “X”, which has been carved into the dirt of a field. Figure E.3 shows the before and after images of running the Edge detection code found in Code E.3

Figure E.3: Original and Final Image

```
clear all;
close all;
clc;

%Open file
```

```
files = dir('FILE0615.jpg'); % List files in a directory
if( ~isempty(files) > 0 )
   [cdata,map] = imread(files(1).name);
   if ~isempty( map )
      cdata = ind2rgb( cdata, map);
   end;
   figure('NumberTitle', 'off', 'Name', 'Original Image'); %label figure components
   imshow(cdata); %Display Image

%Convert to RGB Images
inputImg = cdata;
doubleImg = im2double(inputImg);
Image_Red = im2double(inputImg(:,:,1)); %Red frame to Double
Image_Green = im2double(inputImg(:,:,2)); %Green frame to Double
Image_Blue = im2double(inputImg(:,:,3)); %Blue frame to Double

%Ground-to-Air Target "X"
level = [.55 .90 .55]; %Thresholding Value Options
GtASignalX = cat(3,double(im2bw(Image_Red,level(1))),...
                           double(im2bw(Image_Green,level(2))),...
                           double(im2bw(Image_Blue,level(3))));
figure()
imshow(GtASignalX);
GtASignalX = im2bw(GtASignalX,.3);
GtASignalX = bwareaopen(GtASignalX,1000);
GtASignalX = wiener2(GtASignalX,[10 10]);
GtASignalX = edge(GtASignalX,'canny')+GtASignalX;
GtASignalX = edge(GtASignalX,'canny');
figure('NumberTitle', 'off', 'Name', 'Ground-to-Air Signal');
imshow(GtASignalX);
GtASignalX = wiener2(GtASignalX,[10 10]);
GtASignalX = bwareaopen(GtASignalX,1000);
figure();
imshow(GtASignalX);
```

Code E.3: Edge Detection

BIBLIOGRAPHY

[1] Newcome, Laurence R. *Unmanned Aviation: A Brief History of Unmanned Aerial Vehicles*. Reston, VA: American Institute of Aeronautics and Astronautics, 2004. Print.

[2] "Predator® UAS." *Predator UAS*. N.p., n.d. Web. 19 Aug. 2013.

[3] L. David. (2007, November) UAVs Seen as Ideal Civilian Research Tools. [Online]. Available: http://www.aviation.com/technology/071106-specenews-civilian-uses-of-uavs.html

[4] Yu X, Hoff LE, Reed IS, Chen AM, Stotts LB (1997) Automatic target detection and recognition in multiband imagery: a unified ML detection and estimation approach. IEEE Trans Imag Proc 6:143–156

[5] Lareau AG (1991) Flight performance of an airborne minefield detection and reconnaissance system. Photogam Eng Remot Sens 57(2):173–178

[6] "Unmanned Flight." : The Drones Come Home. N.p., n.d. Web. 26 Aug. 2013.

[7] J. Miller, P. Minear, A. Niessner Jr, A. DeLullo, B. Geiger, L. Long, and J. Horn, "Intelligent unmanned air vehicle flight systems," in AIAAInfoTech Aerospace Conference, no. AIAA 2005-7081, pp. 26–29.

[8] Search and Rescue Challenge." UAV Challenge. 11 Aug. 2013 <http://www.uavoutbackchallenge.com.au/index.cfm?contentID=5>.

[9] Gu I Y-H, Tjahjadi T (2002) Multiresolution feature detection using a family of isotropic bandpass filters. IEEE Trans Sys Man Cybern 32(4):443–454

[10] Gu I Y-H, Tjahadi T (2002) Detecting and locating land mine fields from vehicle- and air-borne measured images. Patt Recog 35(12):323–337

[11] "Unmanned Aerial Vehicles: Embedded Control." *Wiley:*. N.p., n.d. Web. 19 Aug. 2013.

[12] "1.1 Real Life Examples of Embedded Systems." *1.1 Real Life Examples of Embedded Systems*. N.p., n.d. Web. 26 Aug. 2013.

[13] Mitchell, Kyle, Steve E. Watkins, James W. Fonda, and Jagannathan Sarangapani. "Embeddable Modular Hardware for Multi-functional Sensor Networks." Smart Materials and Structures 16.5 (2007): N27-34. Print.

[14] "Adafruit Ultimate GPS on a Raspberry Pi « adafruit industries blog." Adafruit industries blog Adafruit Ultimate GPS on a Raspberry Pi RSS. 09 Apr. 2013 <http://www.adafruit.com/blog/2012/08/28/adafruit-ultimate-gps-on-a-raspberry-pi/>.

[15] "BeagleBoard-xM." BeagleBoard Wm. BeagleBoard.org. 06 Apr. 2013 <http://beagleboard.org/Products/BeagleBoard-xM>.

[16] Solomon, Chris, and Toby Breckon. *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab*. Chichester, West Sussex: Wiley-Blackwell, 2011. Print

[17] Zitova, Barbara, and Jan Flusser. "Image registration methods: a survey." *Image and vision computing* 21.11 (2003): 977-1000.

[18] Polesel, Andrea, Giovanni Ramponi, and V. John Mathews. "Image enhancement via adaptive unsharp masking." *Image Processing, IEEE Transactions on* 9.3 (2000): 505-510.

[19] Crane, Randy. *simplified approach to Image Processing: classical and modern techniques in C*. Prentice Hall PTR, 1996.

[20] Erdos, David. "An Experimental UAV System for Search and Rescue Challenge." *IEEE Aerospace and Electronic Systems Magazine* 28.5 (2013): 32-37. Print.

[21] "GPS Buying Guide - SparkFun Electronics." GPS Buying Guide - SparkFun Electronics. 06 Aug. 2013 <https://www.sparkfun.com/pages/GPS_Guide>.

[22] "Gumstix - Dream Design Deliver." Gumstix® Open Source Products. Gumstix. 10 Apr. 2013 <https://www.gumstix.com/store/index.php>.

[23] "2nd Raspberry Pi US Roadshow is go!" Raspberry Pi. 06 Aug. 2013 <http://www.raspberrypi.org/>.

[24] Bales, Micheal R. *A Video Compression Algorithm for Security Applications*. Thesis. University of Missouri--Rolla, 2006., 2006. N.p.: n.p., n.d. Print.

[25] (2011, Oct. 10). *Battery Life|Climber.Org Home Page.* [Web]. <http://www.climber.org/gear/batteries.html>.

[26] Http://www.usb.org/developers/docs/usb_20_070113.zip. USB 2.0 Specification. 27 Apr. 2000.

[27] Linear Technologies. "LinearTech LT1374." E-mail to the author. 19 Mar. 2013.

VITA

David Christopher Macke, Jr. was born in Kirkwood Missouri on March 31st, 1990. He graduated from SLUH in St.Louis Missouri in 2008. He received a Bachelors of Science in Electrical Engineering from Missouri University of Science and Technology in May 2012. During his undergraduate work he focused on power electronics and was a 2102 recipient of the Grainger Power Engineering Award.

Continuing on, he received a Master's of Science in Electrical Engineering in December 2013. Along with his class work, David was an active member in Eta Kappa Nu and Tau Beta Pi. He served as the Student Representative of the IEEE Region 5 for the 2011-2012 term. Also during his graduate study. David was the President of the Missouri S&T IEEE UAV Team, a design team focused on the advancement of UAV technology in education.