



WPI

Aerial Drone Control Networks

Autonomous Control of Multi-Drone Systems

A MAJOR QUALIFYING PROJECT
SUBMITTED TO THE FACULTY
of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the
Degree of Bachelor of Science

on

April 30, 2015

by

Conor Geary
cwgeary@wpi.edu

Zachary Jasensky
zpjasensky@wpi.edu

Christopher Keane
cmkeane@wpi.edu

Zachariah Lovett
zdl@wpi.edu

icu@wpi.edu

Approved:

Prof. Alexander Wyglinski

Prof. Taskin Padir

Project number: MQP-AW1-ADCN

This report represents work of WPI undergraduate students submitted to the faculty as evidence of a degree requirement. WPI routinely publishes these reports on its web site without editorial or peer review. For more information about the projects program at

WPI, see <http://www.wpi.edu/Academics/Projects>.

Abstract

The goal of this project is to create an easy-to-expand Unmanned Aerial Vehicle (UAV) platform capable of conducting coordinated wide-area reconnaissance. Our system uses a combination of off-the-shelf components and open-source software to enable custom mission creation. Each drone packages and sends image, position, and orientation data over a WiFi connection to a centralized ground station computer for processing. Potential applications for this system range from search-and-rescue to surveying and inspection.



Figure 1: Drones offer an exciting way to interact with the world. With an endless selection of parts and powerful software, the possibilities are endless. Image taken from a Hexacopter drone built for this MQP near WPI campus.

Contents

List of Acronyms	ix
Executive Summary	xi
1 Introduction	1
1.1 Project Goals	3
1.2 Paper Organization	4
2 Drone Building Guide	5
2.1 Basic Parts	5
2.1.1 Propeller	5
2.1.2 Motor	9
2.1.3 Electronic Speed Controller and Voltage Regulator	10
2.1.4 Servo	12
2.1.5 Radio	14
2.1.6 RC Receiver	15
2.1.7 Autopilot	16
2.1.8 Video transmitter	17
2.1.9 Batteries	18
2.1.10 Frame	24
2.2 How to Assemble	25
2.2.1 Setting up the software	28
2.3 Warnings and final notes	30
2.4 Other Resources	30
2.4.1 Part Suppliers	30
2.4.2 Information and Instructions	31
2.4.3 Battery Information	32

2.5	Summary	32
3	Planning and Logistics	34
3.1	Individual Design	34
3.2	Logistics	41
3.3	Camera and Image Processing	44
3.4	Summary	46
4	Implementation	48
4.1	System Architecture	48
4.2	Software	51
4.2.1	Ground Station	52
4.2.2	Drone	56
4.2.3	Ground Station Visualization	57
4.3	Summary	63
5	Results	64
5.1	Drone Fitness Testing	64
5.2	Wireless Range Testing	67
5.2.1	Test Design	68
5.2.2	Test Results	68
5.3	Image Collection testing	69
5.3.1	Test Design	69
5.3.2	Test Results	70
5.4	Full System Test	72
5.4.1	Test Design	72
5.4.2	Overall Results	72
5.5	Issues	72
5.5.1	Weather	73

5.5.2	Loss of Assets	73
5.5.3	Image Stitching Setbacks	74
5.6	Results Summary	80
6	Conclusion	82
6.1	Future Work	82
6.2	Final Thoughts	83
7	Appendix A	88
7.1	Additional 3D Reconstructions	88

List of Figures

1	Drones offer an exciting way to interact with the world. With an endless selection of parts and powerful software, the possibilities are endless. Image taken from a Hexacopter drone built for this MQP near WPI campus.	
2	Each propeller has a marking indicating its diameter and pitch values. The diameter and pitch values for these propellers are 12in x 4.5in	6
3	Pitch varies along the length of the propeller to keep aerodynamic loads balanced. [34]	6
4	Propellers come in different materials and types, each with their own distinct benefits.	7
5	Propellers come with different profiles, each suited for different kinds of aircraft.	8
6	The most obvious difference between brushed and brushless motors is the number of wires. A brushed motor has two wires, and a brushless motor has three wires.	9
7	Brushless motors come in a number of distinct types.	10
8	Electronic Speed Controllers take signals from a microcontroller and convert them into motor output.	11
9	Linear and switching regulators are used as BECs to provide clean power to small drone control electronics and servos.	11
10	Servos provide rotational motion with positional awareness.	12
11	Servo horns provide mounting customization when using servos for an application.	13
12	Servos come in a variety of different sizes, proportional to output power. . .	13
13	RC radios provide a direct connection between the pilot and the aircraft. . .	15
14	Single antennas are lower cost, while multiple antennas offer better reliability.	16
15	Video transmitters allow for instant viewing of any onboard camera systems present on the vehicle.	17

16	LiPo batteries provide high density power in a small package.	19
17	Lithium is explosive when exposed to moisture. Over-voltage or punctures in a LiPo cell can be catastrophic. [18]	19
18	LiPo batteries contain several cells in series and parallel to increase nominal voltage and current supply.	20
19	Balance connectors allow for the individual access of each battery cell.	21
20	Battery chargers include balance ports to keep each cell evenly charged. This is important for battery health.	22
21	Battery monitors allow for an audible alarm to indicate when battery voltage drops below a programmed threshold.	23
22	Mission Planner allows for easy setup, deployment, and customization of UAVs.	29
23	A hex frame was designed in CAD to provide a lightweight, minimalistic hexarotor frame.	35
24	The assembled hex frame provided high performance due to its lightweight, minimalistic design.	36
25	Manufactured frames are available that provide easy construction and good performance. The Tarot 680 Pro is one of them. [39]	36
26	The Tarot 680 Pro is an easy to construct off-the-shelf hex frame solution. .	37
27	CAD Model of laser cut wood electronics bay and printed hood for weather resistance.	38
28	Multiple frames were designed in CAD to test performance.	38
29	The Y6 Hex frame is another frame type experiment.	39
30	The yellow hexacopter frame was inspired after the aluminum hex.	40
31	CAD Models showing the landing gear in the raised and lowered positions. The system is powered by small servos in the base of each leg.	40
32	CAD Models showing the landing gear toggle points.	41

33	The overall architecture of the system is dependent on a ground station computer and multiple drone subsystems.	48
34	The drone system architecture is dependent on both real time and non-real time components.	50
35	The ground station software involves running a python script that facilitates wireless connections to the drone. An SSH link is present to the drone to initiate the drone software if it fails to launch.	53
36	A screen capture of Mission Planer running. This is the software used to set the mission waypoints and configure the autopilots on the drones.	55
37	Data being processed using the GUI in VisualSFM [7].	59
38	Big-O-Complexity, number of items being processed vs. number of operations required to process those items [11].	60
39	Drone fitness testing required strapping the system down to simulate full load on all components and sensors for calibration.	67
40	The range test consisted set up a WiFi connection between two points 1.3miles away. The ground station was located at 42°16'33.8"N 71°48'23.5"W and the drone flew at 42°16'44.0"N 71°46'53.5"W.	68
41	This picture was transmitted and received over a 1.3mile WiFi link. GPS location 42°16'44.0"N 71°46'53.5"W.	69
42	A few of the many images collected during a flight over the Worcester Polytechnic Institute campus. These were taken from about 60m altitude at 42°16'25.4"N 71°48'33.9"W.	70
43	The 3D reconstruction generated from the image collection test flight over WPI campus.	71
44	Months of excessive snowfall made it difficult to test the system during the winter terms.	73
45	Mechanical failures resulted in loss of assets.	74

46	When two projectors are aimed at geometry on Unity the light from the multiplies. This means that the images are only visible where both projectors overlap. The projected images are also much darker than the original photos.	76
47	The edge pixels of projected images in Unity continue onto the rest of the geometry being projected on.	78
48	Construction of a quad in OpenGL is done with two separate triangles. This causes a seam to be visible when the quad is re-sized to irregular shapes. . .	80
49	The 3D reconstruction generated from the image collection test flight over WPI campus.	90

List of Tables

1	Energy and Momentum	26
2	The Tier 1 goal list represents the bare-minimum requirements for the completion of this project. These are the "must-haves".	42
3	The Tier 2 goals represent the "nice to haves" for this project. They expand on the bare requirements but are not mandatory for the completion of this project.	43
4	The Tier 3 goal list represents the unrealistic goals. These are the "super nice to haves" which show how this project's scope can quickly exceed realistic expectations.	44

List of Acronyms

APM ArduPilot Mega, the real time controller in charge of stabilization and navigation

ARM An ARM foundation CPU architecture

AVR An Atmel embedded controller brand

BEC Battery Eliminator Circuit, an RC term for a Linear Voltage Regulator

CPU Central Processing Unit, the main brain of a computer

CTFA Calzone, Tasty Food of Angels. Comins a 300g 10"x10"x3" box.

DOF Degrees of Freedom

ESC Electronic Speed Controller

FOV Field of View

FPV First Person View. Normally refers to flying a drone from an onboard camera

GPU Graphics Processing Unit

GPS Global Positioning System

HLC High-Level Controller. Onboard computer responsible for non-real time tasks

JPEG Joint Photographic Experts Group. A common method for compressing images.

LCM Least Common Multiple

LiPo Lithium Polymer Battery

MEMs Micro Electrical-Mechanical Device. Accelerometer, Gyroscope, etc

Pose 6-DOF position and orientation vector of a robot

RT Real Time

SFM Structure From Motion

TEC Thermo Electric Cooler. Also known as a Peltier chip

UAS Unmanned Aerial System. Used interchangeably with drone and UAV

UAV Unmanned Aerial Vehicle. Used interchangeably with drone and UAS

WPI Worcester Polytechnic Institute

XML Extensible Markup Language. A set of rules for creating documents that are both machine and human readable.

Executive Summary

Airborne drones can be used in a large number of applications from agriculture to military operations. Many of these applications could be performed better by groups of drones working together. Using a multiple drone system allows for improvements in effective range and task completion time. In addition, this increases flexibility with respect to carrying various hardware on different drones. To control these drones, a networked system is necessary to coordinate their activity and aggregate the collected data. Use of such a system for commercial applications in the US is, of course, predicated on the reformation of U.S. UAV policy.

Recently, unmanned aerial vehicle technology has become significantly less expensive. The sensors, computational power, and software required to create a drone were previously only available to government entities. However, mobile computing technologies originally developed for laptops, cellphones, and tablets, have been adapted for use in UAVs. Additionally, several open source projects have enabled the average consumer to build their own drone from scratch, as well as kick-start several consumer drone businesses. Due to all of these factors, a completely autonomous system can be bought for under one thousand US dollars. This makes UAV technology accessible to smaller entities as well as enabling cost-effective multi-drone systems.

The goal of this project is to set up a simple system for controlling multiple drones and receiving their aggregate data. The use of multiple drones allows for quicker acquisition of data over a large target area. As the drones fly, the data that they collect will be received in real-time, allowing the operator to not only complete the task faster, but also re-task units as needed to further investigate areas.

This project is divided into two main tasks:

- Creation and Modification of the UAVs

The design and building process for each drone used in the project. This included assessment of necessary features for each drone. This process continues throughout the project as features and part designs are reviewed and improved.

- Software Design of Image Transfer System

This step included determining the necessary hardware to capture and store images, and researching platforms on which to develop the software.

Development of the physical drone hardware was nearly completed before this MQP was initiated. However, major changes and additional parts needed to be made in order to meet the the project goals for payload and flight time. These changes included mounting systems for the image and data collection payload, new flight configurations, and the creation of additional drones for testing the multiple drone system.

The next two tasks of this project involved developing software for the drone system. In order for the data from the drone to be collected and transmitted, an extensible, modular system of programs was developed for both the UAV and ground station. These programs access the data collection system, and automatically transmit results to the ground station. In addition to capturing the data, the script is also able to access the location and position information from the real-time flight controller. This position information is paired with the captured data such that, in the future, the ground station can use this information to better display the captured data.

The approach to creating a system for controlling multiple drones in flight took several avenues of research. One approach was implementing a control scheme using ROS nodes and assembling navigational messages. This was ultimately abandoned in favor of using existing open source software, i.e. MissionPlanner. Plans were made for developing a better control scheme with easier access protocols, but not implemented.

In addition to creating the system, a subsequent goal of this project was to provide an interested reader a simple explanation for how to create autonomous aerial vehicles. Through the knowledge gained from various designs, flights, and crashes, a well informed “beginners guide” to drones was developed and implemented in this report. This chapter covers all the concepts involved from defining the basic requirements to constructing and flying a working UAV.

Over the course of this project, software and hardware were developed to meet the agreed upon specifications. When tested, the system was able to transmit and receive both image and positional information from a drone in-flight from a range of 1.3 miles.

1 Introduction

Situational awareness is a key requirement for safety in any professional operation. The ability to have a quick, complete understanding of the surrounding environment can be a major advantage in many jobs. However providing this knowledge can be expensive or time consuming using traditional methods. This project strives to provide a cheap and simple solution for situational awareness through the collection of aerial imagery in real-time. Using networks of multiple unmanned drones for collecting overhead images offers a solution that can be deployed fast and reduces logistical strain.

There are a number of industries that could benefit from this technology. Specific use cases have been compiled for search-and-rescue and agriculture. The technology could also be used for the entertainment industry, law enforcement, and several other scenarios.

Using this technology for search-and-rescue is an obvious approach, and the benefits are readily recognizable [2]. For example, allowing responders to quickly assess a wide area for signs of a lost party can potentially cut down the time needed for searching. Time is critical in search-and-rescue [24]. The chance for success decreases the longer a search goes on. Deployment of these drones is much faster and less expensive than the deployment of a helicopter for the same purpose. Not only will they increase probability of success for an operation, the convenience and reduced price allows for smaller organizations to utilize aerial search methods that are usually not available to them.

Agriculture is one sector in which drone technology is already being used, but could be improved with a networked system such as the one proposed [8]. A farmer using a single drone for surveying their fields will not be able to cover their entire property in one run. The proposed system's increased range and possible use of multiple units means a single run survey.

Over the past five years drone technology has changed from something that only the military or nation states could afford, to a technology so available that any home tinkerer can create a drone in their garage. These consumer drones are vastly different than their

military predecessors: smaller, cheaper, and generally less capable. However, they still possess the ability to perform impressive missions.

The DJI Phantom would be considered by many to be the current state-of-the-art in the prosumer and consumer drone marketplace. The quadcopter design offers features such as autonomous flight, full stabilized video recording, and live video streamed back from the drone. The DJI Phantom is available for under one thousand dollars, and has been popular with both amateur filmmakers and hobbyists alike [15].

For a slightly higher premium, a professional can purchase a product from senseFly [40]. These light 1.5 lbs fixed wing drones are designed specifically with mapping in mind. They fly completely autonomously and carry an interchangeable suite of cameras or mapping sensors for the job being carried out. Since they are small and entirely autonomous, these drones can be operated by anyone with little to no training, with no need for piloting skills or even a controller; the entire system is operated using a tablet computer. The system outputs both 3D and color maps of the area it has flown over .

Finally, one of the most interesting, complete, and affordable autonomous aerial vehicle products is the DroneDeploy system. This system integrates with a variety of drone controllers and can complete many of the same operations that senseFly products can [17]. Integration with multiple controllers means that the user is not limited to a single airframe and can even use custom airframes. In addition, the drone deploy system uses 4G cellular technology for both data and control. This means that the system is able to send back real-time images from the vehicle in flight to the operator.

Although the various commercially available UAV systems have great features, each has its own shortcomings. Even though the senseFly system is able to create high resolution maps of the areas it flies over, none of this data is available in real-time. After the vehicle lands the images must be uploaded to a web server, processed externally and only then is the model generated and returned. In a real world search-and-rescue environment, the data needs to be available as soon as possible, preferably while the drones are still in flight [24].

DroneDeploy has an advantage in this case because the vehicles are able to relay the data the collected live over a 4G data connection. The images are then able to stream to a handheld computer to be analyzed and viewed. However, once again in a search-and-rescue operation this system would be of limited use. This is because many of the locations where search-and-rescue operations are carried out are in remote locations without reliable cellular service.

1.1 Project Goals

The project focused on a specific set of goals from the beginning. Key areas of development were targeted in order to create a standard for minimum expected functionality. Extensions to these goals are listed in the planning section of the report.

- Minimum Drone Specification

The drones used in the project must be safe to operate. This included a means of manually overriding any drone at any time, as well as fail-safes put in place to automatically return to launch point upon loss of communications. The drones had to be able to carry a minimum payload allowing the necessary instrumentation and fly for a minimum time of 10 minutes.

- Communications Distance

The project required that data transmission range match the operating range of the drones, or that the system employs a means of handling a lost connection. This allows the real time data flow emphasized in the projects use cases.

- Communications Speed

Data transmission had to be established at speeds which allowed transfer of images in a timely manner. This is to prevent backups in the transmission of data from each drone.

- Group Control

The system must allow simultaneously controlled autonomous flight of at least two drones. These drones must be operating legally, i.e. with one backup manual operator for each drone.

- User Interface

Some manner of user interface must be established to represent the data gathered by the network of drones. This data should be organized in a manner easily interpreted by the user.

1.2 Paper Organization

This report leads with a guide on assembly of UAVs aimed at later MQP teams attempting to build off the work done during this project. The guide presents similar methods to those used for the design and construction of the UAVs utilized during this project. Following this guide are the traditional sections covering the course of the project. The planning section presents the goals and timeline developed for the project and establishes the preparations made prior to the project. Implementation covers covers the actual software and hardware designs. The results section includes test planning and results for each system, as well as an overall system review and evaluation of goals met. The conclusions section examines the successes and failures of the project and proposes future work necessary to meet the remaining project goals.

2 Drone Building Guide

The following section aims to provide an introduction to the terms and concepts necessary for understanding the technology used in small UAVs. The reader will be introduced to the of the basic parts found in both UAVs and RC hobby projects. The key decisions involved in designing a system along with guidelines to help with the decision process will be provided. Once a foundation for the mechanical and software aspect of the project has been provided, some of the challenges and regulatory concerns associated with flying small UAVs will be discussed.

2.1 Basic Parts

An electric UAV is normally made up of only a few basic components. These components are mostly the same between electric planes and multirotors, with the main difference being simply the number or size of the given parts. These parts are what make hobby scale aircraft possible, and play a big part in the versatility of small UAVs.

2.1.1 Propeller

The first part that we are going to talk about is the propeller. Propellers come in all shapes and sizes, with the ones that we are using coming primarily from hobby part manufacturers. Propeller sizing is labeled using the format: *diameter* x *pitch*.



Figure 2: Each propeller has a marking indicating its diameter and pitch values. The diameter and pitch values for these propellers are 12in x 4.5in

The pitch is the number of inches forward that the propeller would travel in one rotation. For example, the propellers used in this project are 11 inches in diameter with a 4.7 inch pitch. The size information can normally be found in small print on the top surface of the blade. The pitch of the blade varies along the profile to account for different tangential velocities at different distances from the center of the blade. For example, because the tip is moving so much faster than the center of blade, its pitch is correspondingly reduced.



Figure 3: Pitch varies along the length of the propeller to keep aerodynamic loads balanced. [34]

Propellers come in a number of different blade configurations, with each one having different advantages and disadvantages. The most common configuration is a fixed two

bladed propeller. These are used in the vast majority of small drones and RC aircraft due to their high efficiency and low cost. They are also the most commonly available propeller and come in a wide range of profiles to fit various applications. However in applications where space is more limited, multi-bladed propellers may be used in order to increase the amount of thrust produced without increasing the area. Tri and quad bladed propellers can be found in scale model aircraft, but are not normally used for multirotors. Some downsides to these propellers are that they take up more space than two bladed propellers when they are not in use, and are less efficient. Folding propellers are another type of propeller, and are commonly used for motor gliders and airplanes that do not have landing gear. While flying, if the motor is not turning, the propeller will fold up along the plane which reduces the aerodynamic drag and helps to protect it during landing.



(a) Normal Propeller



(b) Tri-Bladed Propeller



(c) Plastic Folding Propeller



(d) Carbon Folding Propeller

Figure 4: Propellers come in different materials and types, each with their own distinct benefits.

Plastic propellers are normally the least expensive, but they do not have the rigidity or precision of either wood or carbon fiber propellers. Propellers also come in a number of

different blade profiles to match different applications. The most common shapes are thin electric, electric slow-fly, multirotor, and paddle. Each profile is suited to a specific type of aircraft, and the profile can have a major impact on performance.



(a) Thin Electric Propellor [43]



(b) Electric Slow-Fly Propellor



(c) Multi Rotor Propellor



(d) Paddle Propellor [29]

Figure 5: Propellors come with different profiles, each suited for different kinds of aircraft.

Thin profile propellers are normally used on high speed performance airplanes, and paired with very high RPM motors. Slow-fly propellers are used when the air speed of the aircraft is relatively low, and move more air but at lower speeds. These are what we use on our multirotors because the airspeed along the axis of the propeller is normally very low. The multi rotor and paddle props are optimized for very large multirotors. They are designed to move as much air as possible while turning very slowly. This is to improve efficiency and helps to achieve long flight times. Most of the multirotors with more than forty five minute flight times use this type of propeller. The downside of large slow moving propellers is that they are much less responsive to control inputs, which can lead to loss of stability.

2.1.2 Motor

The type of motor used plays a huge role in the way that a aircraft performs. Although there are a few different families of motors, we are using electric brushless motors. The “brushless” part comes from the motors’ lack of brushes that are used to control the polarity switching in normal brushed motors. The advantage of brushless motors is much greater efficiency and longer life as there are no brushes that are constantly rubbing on the moving parts of the motor. The downside is that they need much more complex circuitry for control.



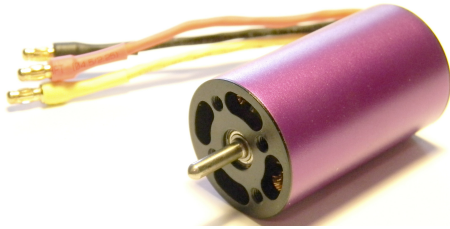
(a) Brushless Motor



(b) Brushed Motor

Figure 6: The most obvious difference between brushed and brushless motors is the number of wires. A brushed motor has two wires, and a brushless motor has three wires.

The hobby motors that we are using come in two standard forms, inrunners and outrunners. The inrunner is more like a normal motor in that it has a shaft that rotates inside of an outer case that remains stationary. An outrunner type motor, on the other hand, is normally mounted using the bottom or top surfaces of the motor, and has powerful magnets attached to the outer wall of the motor or “bell”. Furthermore the entire outer casing of the motor spins around the central core of electromagnets. This type of motor works well for our application as they can pack more magnets and wiring inside the same space, and tend to have better cooling ability compared to an inrunner of the same size.



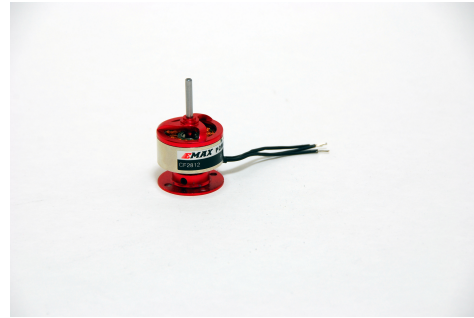
(a) Inrunner Motor [23]



(b) Outrunner Motor



(c) "Pancake" Motor [30]



(d) Bell Motor

Figure 7: Brushless motors come in a number of distinct types.

When picking a brushless motor, there are three key numbers to be kept in mind. These are the weight, normally measured in grams, the power measured in watts, and how fast the motor turns per volt applied at zero mechanical load. This last measurement is normally referred to as the “kV” rating of the motor. The power and kV ratings of the motor determine how much current the motor will draw and the maximum safe supply voltage. As a general rule for brushless motors; slow motors with big propellers will give long flight times, while fast motors with smaller propellers will give high performance at the cost of endurance.

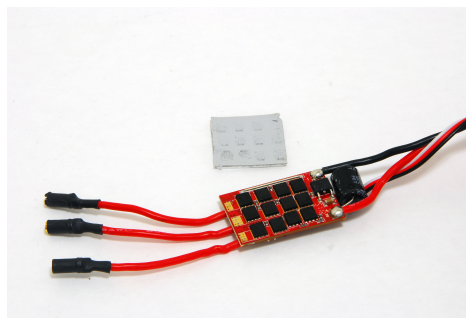
2.1.3 Electronic Speed Controller and Voltage Regulator

The Electronic Speed Controller, or ESC, is a key component of small drones. The ESC is what allows the drones to use brushless motors instead of the much less efficient brushed motors. The ESC is normally controlled using a Pulse Width Modulation (or PWM) signal that is supplied from either the radio receiver or the autopilot. Based on the PWM signal,

the ESC then modulates the power from the battery to energize the correct motor winding to pull the motor forward at the desired speed.



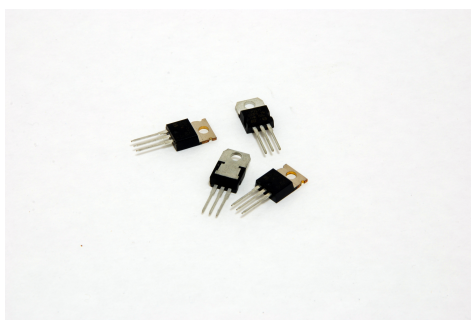
(a) ESC with Cover



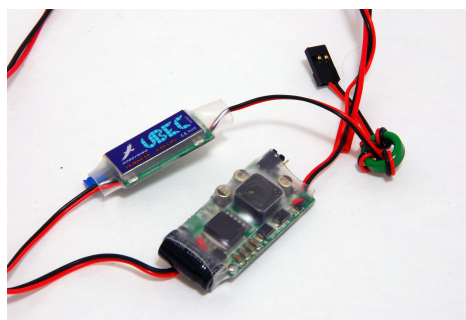
(b) ESC without Cover

Figure 8: Electronic Speed Controllers take signals from a microcontroller and convert them into motor output.

The ESCs are used for powering the motors, but the rest of the system normally operates at 5V DC. Since the main batteries output a wide voltage range that is normally higher than 5V, a Battery Eliminator Circuit (BEC) in the form of an additional voltage regulator is needed. There are two types of voltage regulators that are commonly used; switching regulators and linear regulators.



(a) Linear Voltage Regulators



(b) Switching Voltage Regulator

Figure 9: Linear and switching regulators are used as BECs to provide clean power to small drone control electronics and servos.

Linear regulators regulate voltage by dissipating excess energy as heat. These regulators are simple, very durable, and can be found as part of some ESCs. They are normally used for low power applications because they are very inefficient. Linear regulators can get hot

when used without proper cooling due to the large amount of heat that they dissipate from the regulation process.

Switching regulators use a very fast electronic switch to turn power on and off to regulate voltage. A large capacitor is then used to smooth the output voltage. Switching regulators are much more efficient than linear regulators, and are used for higher power applications such as powering servos.

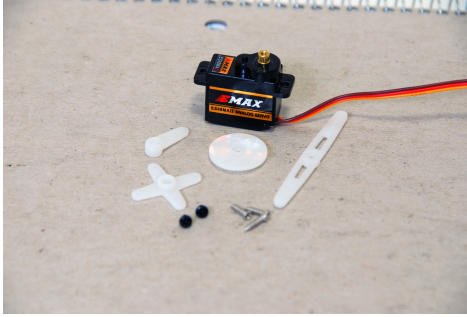
2.1.4 Servo

A servo is a type of motor that is used for position control, and normally has a limited range of motion. Servos that are used in small drones and RC aircraft typically consist of a small motor and gearbox with a potentiometer or similar position feedback mechanism. The servo receives a target position somewhere within its range of motion either through PWM or over a serial link, such as Futaba's S-BUS. The servo uses the position feedback mechanism to move to, and then maintain, the set position.



Figure 10: Servos provide rotational motion with positional awareness.

Hobby servos all use something called a servo horn to interact with the mechanical system that they are actuating. The servo horn is the little arm or disk that screws onto the small spline gear that serves as the output for the servo. These horns come in many shapes and sizes and can be customized for almost any application.



(a) Servo without Horn



(b) Servo with Horn

Figure 11: Servo horns provide mounting customization when using servos for an application.

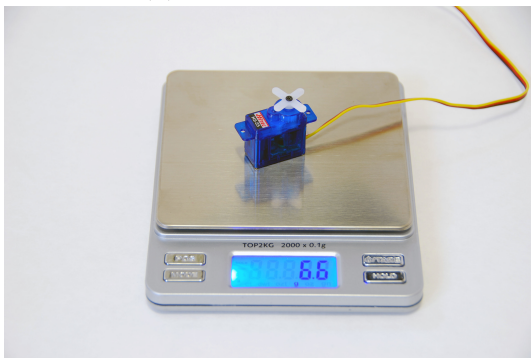
Servos are for actuating accessories such as cameras or drop pods as well as for operating control surfaces, such as ailerons and elevators on airplanes. Servos have a number of different types and ratings that are associated with them. The simplest and the most important are the weight, torque, and size.



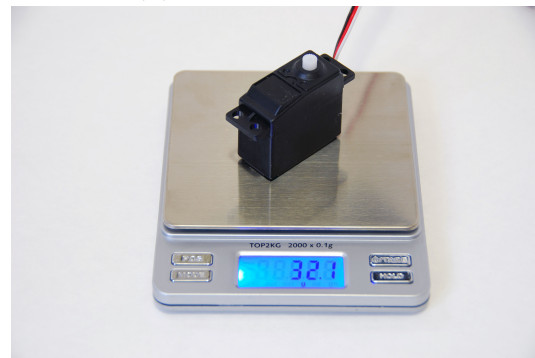
(a) Sub-Micro Servo



(b) Metal Micro Servo



(c) Micro Servo



(d) Standard Servo

Figure 12: Servos come in a variety of different sizes, proportional to output power.

The most common size for servos that are used on airplanes is the micro servo. There

is also the sub-micro size for very small models, and the standard servo size that is used on larger airplanes as well as for other applications such as small robotic arms. The weight of the servo is tied to the size of the servo as well as features such as if the gearbox uses metal or plastic gears. When picking anything that goes in the air its always a good idea to pick the lightest part that will meet the requirements.

The torque rating is a measure of how strong the servo is, and is normally measured using a force-distance combined unit. An example of this would be a servo with 50oz-in of torque. What this means is that the servo could move up to a 50oz weight 1 inch away from the pivot, or a 1oz weight 50 inches away from the pivot (assuming a weightless arm). How strong your servos need to be depends on how big the plane is, and how aggressively it will be flying, and thus how much wind load is on your control surfaces.

2.1.5 Radio

The most commonly used main pilot interface for operating small drones is the hobby radio controller. The radios have two control gimbals or “thumbsticks” that are mapped to four channels of control. Normally these are mapped as pitch and roll on one of the sticks, and throttle and yaw on the other stick. The radios normally have a number of other switches and knobs that can be mapped to additional channels. These channels can control things such as flaps, camera triggers, landing gear, and other such accessories. Controllers come in two main formats determined by the location of the main sticks. The Mode 1 type controllers put the throttle/yaw stick on the right and the pitch/roll stick on the left. The Mode 2 controllers reverse this. Mode 2 is the most common layout, but most controllers can be switched.



Figure 13: RC radios provide a direct connection between the pilot and the aircraft.

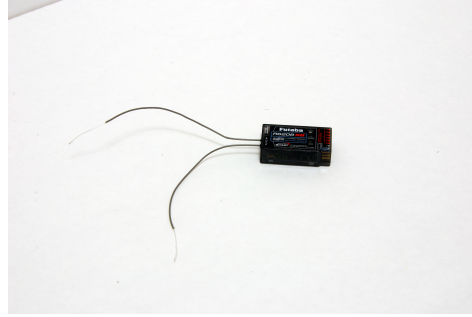
The most common frequencies used in hobby radio controllers are 2.4GHz and 433MHz. The 2.4GHz radios are used for short to mid range, and normally work well out to about a mile of range. The 433MHz radios are used for longer range flights, and are commonly called Long Range Systems (LRS). These systems are used for long flights and can have ranges exceeding ten miles.

2.1.6 RC Receiver

The receiving module that is used on the airframe communicates to the rest of the drone using pulse width modulation (PWM) or pulse position modulation (PPM). Both systems use a three wire interface to transmit information. The three wires are ground, five volts power, and data. The signal is either used to actuate controls directly, or to provide input to a flight computer such as the APM or Pixhawk which then calculates the correct control outputs for the desired behavior.



(a) Single Antenna Receiver



(b) Diversity Receiver

Figure 14: Single antennas are lower cost, while multiple antennas offer better reliability.

Like all radio systems, the antenna or antennas used with the RC receiver have a large impact on the performance. Most of the modern receivers use two separate antennas for receiving radio signals, and will switch which one is used based on which has the best signal strength. This is called a diversity receiver, and it is very useful in aircraft as it means that the antennas can be placed in different orientations so that as the aircraft maneuvers the signal remains consistent.

2.1.7 Autopilot

In order to have a flyable multirotor, some level of computer assistance must be provided by the flight electronics. These assistance systems range in complexity from simple sensors and a basic micro controller all the way up to sophisticated multi-computer systems. The majority of these systems are based around MEMs accelerometers and gyroscopes, due to the low cost and wide availability of that type of sensor. The simplest systems only make use of these two sensors and can not provide any autopilot functionality. An example of this type of system is the KK2 [25] board. More advanced systems such as the open source ArduPilot [10] or the DJI NAZA [1] use better processors and add a barometer, a magnetometer, and a GPS. The additional sensors and processing capability allows those systems to provide full autopilot functions.

Though there are many different autopilots and control boards available, the Pixhawk

[33] and the ArduPilot from 3D Robotics were chosen for this project. This is because they are one of the only products that has the flexibility to control a broad range of airframe types, and is open source. The ArduPilot, or APM for short, is the older of the two controllers and is based on a 8-bit AVR CPU. The Pixhawk is the modern replacement for the APM, and is based on a 32 bit ARM M4 architecture. The Pixhawk runs faster and has a more advanced sensor package, but for most applications the two autopilots perform equally.

2.1.8 Video transmitter

Many drones are used to carry cameras for fun, surveying, or film making. In many of these cases, a live video feed from the camera is very useful and is used to ensure that the shot is setup correctly or to fly the drone from the point-of-view of the camera in what is called First Person View (FPV) flying. It is important to note that in some places FPV flying is currently banned, so make sure to check local regulations before flying. In any of these cases where a live view from the camera is needed, a video transmitter is used to provide a live feed.

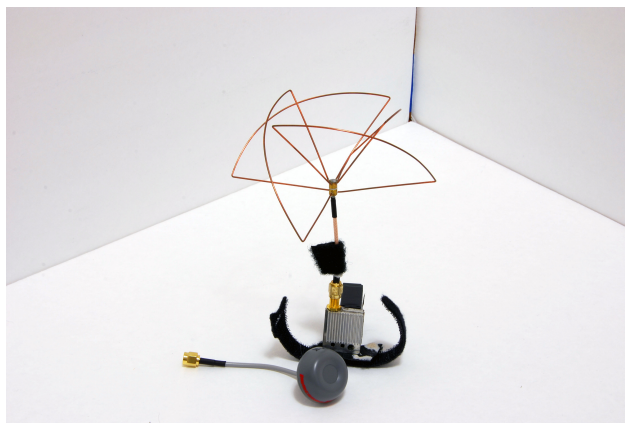


Figure 15: Video transmitters allow for instant viewing of any onboard camera systems present on the vehicle.

There is a large variety of transmitters available in a number of different frequencies. The costs normally range from around twenty or thirty dollars for a low power analog transmitter to multi-thousand dollar high power digital radios that can send HD video streams for miles.

The analog systems are the most common in hobby level flying, while the digital HD systems are used for professional productions. The analog systems come in a number of frequencies, with the most common being 900MHz, 1.2GHz, 2.4GHz and 5.8GHz. They also come in a range of power levels, though anything above 1W (1000mW) requires an amateur radio license to use in the US.

The most important factors when picking a video transmitter are frequency and antenna. Many people just look at transmitter power, but more power means more weight, more battery usage, and a higher likelihood for causing interference with the other radios on the drone. In general, you can improve performance more through careful selection of the frequency and the antenna than you can by just increasing the transmitter power.

The frequency of the video transmitter should be a frequency other than the frequency used by the radio controller in order to avoid interference and loss of control. If possible, harmonics of the control radio should also be avoided.

In addition, selecting the correct antenna can also have a huge impact on wireless performance. There are two types of antennas in a basic video transmission system: linear polarized, and circularly polarized, each coming in omnidirectional or directional configurations. For the high frequencies of video transmitters, which are typically in the 2.4GHz-5.8GHz ranges, linear polarized antennas are not a good choice, as these high frequencies tend to bounce off anything and everything in their way. These bouncing signals will eventually make it to the viewing station, causing multi-path interference and degraded image quality. Circular polarized antennas reject this interference, leading to better overall performance at high frequencies. However, these antennas are typically more expensive than their linearly polarized counterparts.

2.1.9 Batteries

The batteries that are used in most modern drones use the lithium polymer battery chemistry. These batteries are similar to the batteries found in most laptops and cell phones, and are

used for their combination of low weight and high power density. Lithium Polymer (also known as LiPo) batteries use the same basic chemistry as Lithium Ion batteries, but have soft cases and normally come in square sheet shaped cells that are easy to integrate into a drone frame. However, the soft case makes LiPo batteries more susceptible to puncture or crush damage.



Figure 16: LiPo batteries provide high density power in a small package.

If a LiPo is damaged either through puncture, crush, or impact, it can catch fire and will burn aggressively and with toxic smoke. LiPos can also self combust if overcharged. This is caused by excessive vaporization of the electrolytic solution in the battery, causing it to balloon and then rupture. If a battery is puffy, or has been over charged or over discharged it should not be used, and should be disposed with other hazardous waste.



Figure 17: Lithium is explosive when exposed to moisture. Over-voltage or punctures in a LiPo cell can be catastrophic. [18]

Like all batteries, the nominal, minimum, and maximum voltages are set by the battery chemistry. A single lithium polymer battery “cell”, regardless of capacity, has a nominal voltage of 3.7V can only be safely charged up to 4.2V and discharged down to 3.2V. Manufacturers make higher voltage batteries by wiring multiple battery cells in series.

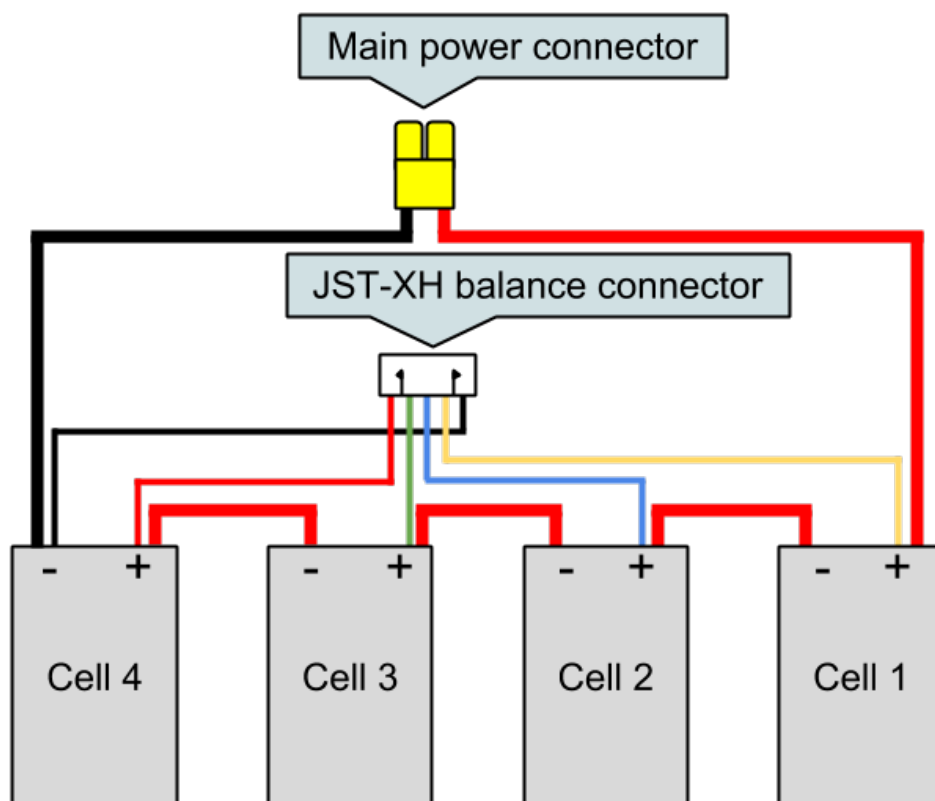


Figure 18: LiPo batteries contain several cells in series and parallel to increase nominal voltage and current supply.

Cells wired in series need to be kept at the same voltage to avoid uneven wear on the battery. It is important to make sure that every cell stays within a safe voltage range both while charging and discharging the battery. This can not be done by simply reading the output voltage of the battery, as one of the cells may be at a lower voltage than the rest. This means that a single cell may be outside of the safe range even if the average voltage of the pack is still safe. All lithium polymer batteries have a connector called a “balance port” or “balance connector” for this reason. The balance port is wired to each of the cells in the

battery to allow each cell's voltage to be read independently.

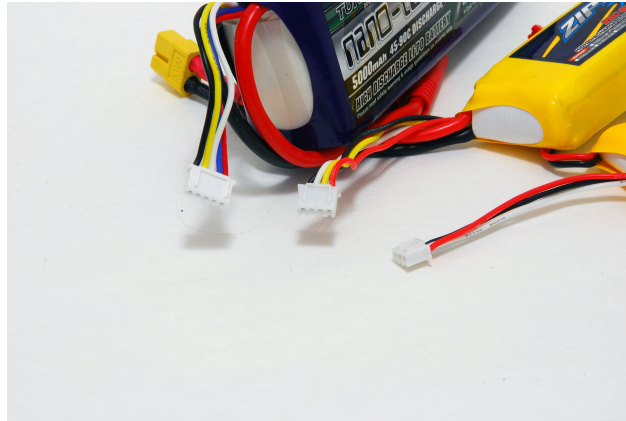


Figure 19: Balance connectors allow for the individual access of each battery cell.

Computerized battery chargers should be used to maintain balance during charging. These chargers have connectors for both the main power connector and the balance port. While charging a battery the charger monitors the voltage of each cell, and will charge and discharge individual cells to maintain balance. This is also called balancing a battery, and is an important part of maintaining a LiPo.

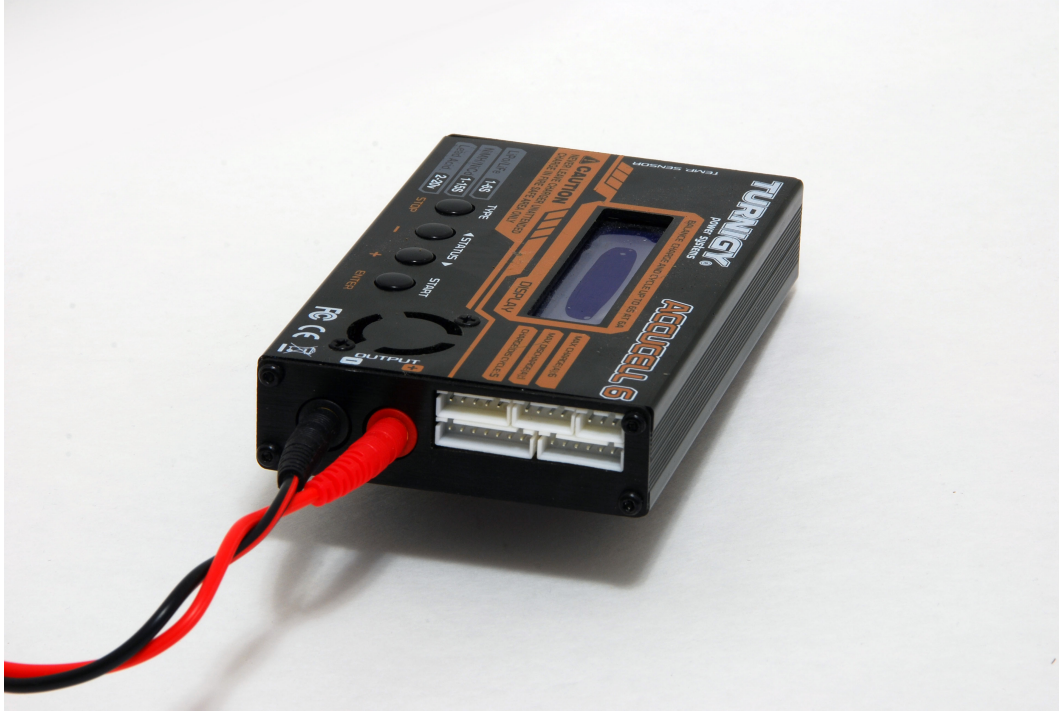


Figure 20: Battery chargers include balance ports to keep each cell evenly charged. This is important for battery health.

Batteries cell voltages need to be monitored while the battery is in use for the same reasons as while charging. The devices that do this are commonly called "battery monitors" and normally cost a few dollars. Battery monitors have two main functions, to display the voltage of each cell, and to alert the operator if the voltage of any cell of the battery is outside of safe values. Battery monitors work by connecting to the balance port of the battery and use a built in display to show the battery voltage. If any cell voltage is outside of the safe range, the monitor will alert the operator using a loud alarm.

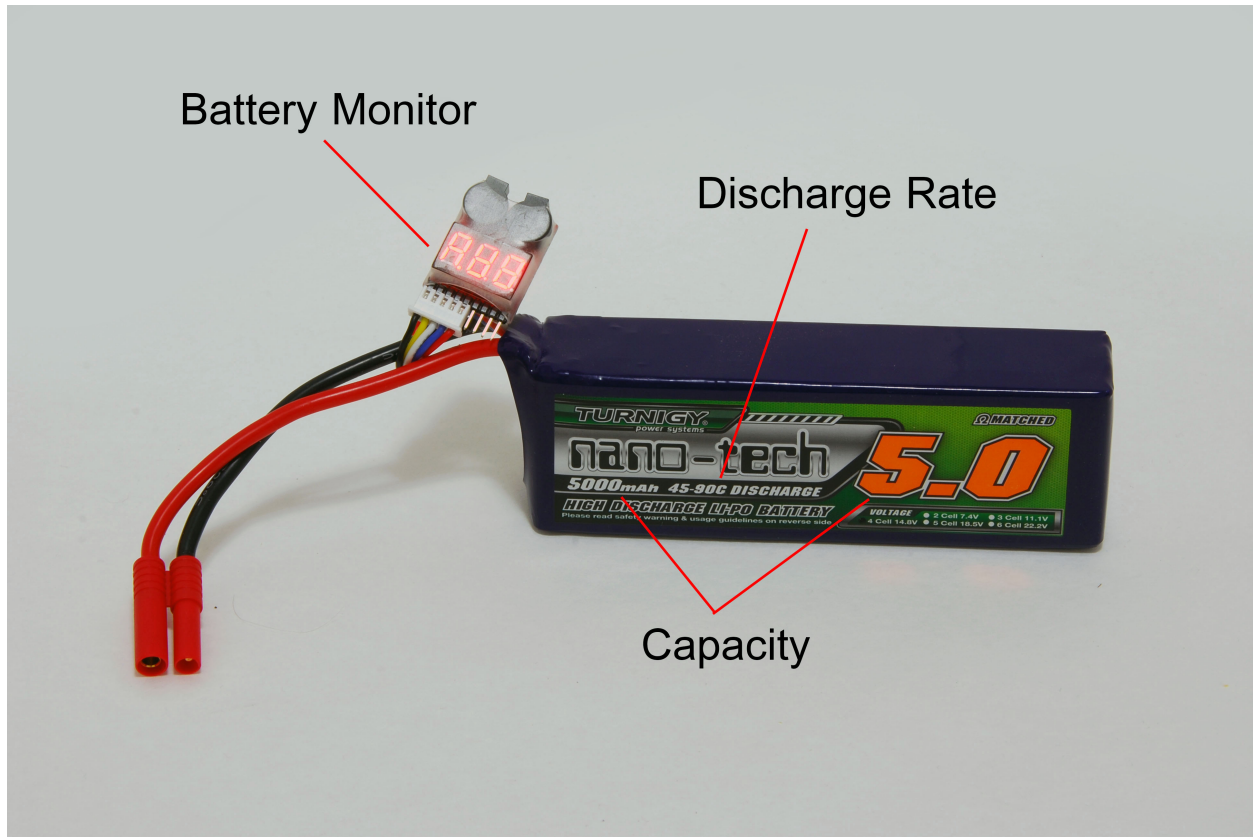


Figure 21: Battery monitors allow for an audible alarm to indicate when battery voltage drops below a programmed threshold.

LiPo batteries used for hobby aircraft have a number of terms that describe the battery construction (S and P), current ratings (C) and current capacity (in Ah or mAh). An example of this would be the flight batteries that we use, which are labeled 5000 20C 4S2P.

The 5000 on the battery indicates the capacity. The units are normally milliamp-hours with the symbol mAh. A milliamp is one thousandth of an Amp, and so one mAh is one thousandth of an Ah. 1mAh means a draw of 1mA for an hour, so our 5000mAh (or 5Ah) batteries can supply 5000mA for an hour, or even a much larger amount of current but for a proportionately shorter amount of time. The battery life for a given current supply can be estimated by using Equation (1):

$$T = \frac{C}{I} \quad (1)$$

where T is time in hours, C is battery capacity in Ah, and I is current in Amps. Note that

1A which is more commonly used as unit of current for small drones, is equal to 1000mA. Another important thing to note is that it is best to never fully discharge batteries, so plan on only using 80% of the time calculated above.

The current rating 20C indicates that the battery can safely supply up to twenty times the capacity rating. In the case of our batteries, which have a 5Ah capacity, this means that each battery can safely supply up to 100A of current. However, at that current draw they would be discharged in about three minutes.

The battery construction terms S and P are used to refer to the number of cells in series and parallel, respectively. For example, 4S indicates a battery with four cells in series, and a nominal voltage of 14.8V (this is because $3.7 * 4 = 14.8$). Wiring cells in series increases the voltage, which increases the amount of power can be delivered for a given current draw because of the relation for electrical power $P = VI$, or *power = voltage * current*. The 2P indicates that there are two 4S 5000 packs that are wired in parallel. Wiring cells in parallel increases both the total current capacity and the maximum current that can be supplied. In our case, because each battery can supply up to 100 amps, this increases our maximum current supply to:

$$5Ah * 20 * 2 = 200A \quad (2)$$

(note 20 comes from the C rating, and 2 comes from the P rating.)

and our maximum power to approximately:

$$200A * 14.8V = 2.96kW \quad (3)$$

(note 14.8V comes from having four 3.7 volt cells in series)

2.1.10 Frame

Drone airframes can be made out of materials ranging from foam poster boards and hot glue all the way up to titanium and advanced carbon composites. This variability and diversity

is one of the things that makes designing airframes for drones so interesting. There are two main categories of drone; fixed wing aircraft (airplanes) and rotor craft (helicopters, multirotors). There are others, such as ornithopters or planes that transition into rotor craft for taking off and landing, but these make up a very small set of drones.

The best advice for people that are just starting to get involved with drones or RC planes is to start small and cheap and work your way up from there. If you are building or buying an airplane, get one that is made out of foam and is easy to repair. If you are starting with a multirotor, get one of the many excellent micro quadcopters that will connect to a full size radio. They cost about a hundred dollars and fly just like a much larger and much more expensive quadcopter and are far more durable.

2.2 How to Assemble

The process of assembling a drone can be daunting and complex depending on the level of experience of the builder and the complexity of the target mission. There are simple platforms available for those that would like to get started flying but do not want to learn to build a drone from the basic components. The two most popular of these systems are the DJI Phantom and the 3DR Solo [15].

If one does actually want to construct a drone using basic parts, the first thing that needs to be decided is what do they actually want to do with the finished device. The parts used for a FPV racing multirotor are very different from those that are used for a stable camera platform that would fly with a DSLR or similar large cinema camera. For some applications, a fixed wing platform makes more sense as it would normally allow for much longer endurance. The key factors that need to be chosen at this stage are: flight envelope, payload, endurance, and size.

The flight envelope is how the aircraft will fly. This consists of the speed it will fly at, how fast it can climb, how sharp it can turn, whether it can hover or not, and other similar parameters. This set of decisions will have the largest impact on the type of airframe that

needs to be designed and constructed, and so should be settled first.

The payload capacity is the next thing that needs to be selected, as it will impact all of the other performance attributes of the aircraft, and may add additional constraints that will need to be met. For example, an aircraft that is designed to carry a small camera will require far less power than one that needs to carry a full size cinema camera and the necessary stabilization equipment.

Endurance is the next key factor because meeting the endurance requirement can have a very large impact on the components selected. The efficiency that is required for good endurance while lifting anything more than a very light payload has a large impact on the flight envelope and general performance characteristics of the airframe. This is primarily due to a relation in physics between energy and momentum. A simplified version of this relation is shown in Table 1 for a given mass and speed.

Table 1: Energy and Momentum

Action	Energy	Momentum
Double the mass	Double the energy	Double the momentum
Double the speed	Quadruple the energy	Double the momentum

This chart is generated from the equations for kinetic energy and momentum, show below:

$$E_K = \frac{1}{2}mv^2 \quad (4)$$

$$p = mv \quad (5)$$

where E_K is energy, p is momentum, m is mass, and v is velocity. The reason this is important for our aircraft design is that the thrust that we generate is directly related to the momentum, while the cost in battery power or fuel consumption to make that thrust is related to the energy. This means that to lift the same amount of weight, a quadcopter with propellers that move half as much air by mass, will need to move that air twice as fast,

which will cost four times the energy and give about a quarter of the endurance. This is a simplification, but it can provide us with a couple of useful rules of thumb. These are:

1. Slower, larger props and motors give better flight times.
2. Minimizing weight is always good.
3. Weight can be thought of as a currency. For example, you can choose to “spend” weight saved from improving the frame on a larger battery.

The size of the airframe is the last major factor that must be decided on, and it has the largest impact on the endurance and cost of the system. The main things that govern the size of the airframe are budget, payload, portability, and endurance. As the frame gets larger, the costs go up very quickly. This is because a larger airframe needs a larger propeller, which needs a larger motor, and so on. Large frames can also be a huge hassle to transport to the location that they will be flown at, and this can introduce additional costs as well. The best frames are a balance between what is necessary to carry the payload and meet the flight goals, and what is cheap to build and easy to transport and store. For example, if you are looking to build an airplane that can fly for half an hour and carry a small camera and video transmitter there is no reason to build an eight foot wingspan drone that can fly for hours and carry a small child, as all this does is increase costs and difficulty of storage and transport. That said, one of the best things about building your own is that you can decide to do it anyway, just because it is cool.

Once the key factors have been determined, the process of deciding on the airframe type, picking components, and assembly can begin. The first thing to decide is the frame size, as from this the motor, propeller, ESC, and battery sizes can all be calculated. Once the size and performance requirements are known, specific parts can be selected. At this point it is normal to go through a few iterations of selecting parts, adjusting the design parameters and other assumptions, and then selecting new parts. This will eventually converge to a valid design that meets the goals.

With all of the parts selected, the actual assembly is normally very straightforward. Basic electronics knowledge and solder skills, in addition to the other information in this document is all that is necessary to build a drone. There are many websites that provide useful information, build tips, and experimental data to people looking get into the RC and drone hobby. See section 2.4 for more information.

2.2.1 Setting up the software

The software used for the initial setup of the drones is called Mission Planner, and is part of the open-source Arducopter project. Mission Planner provides an easy to use interface to flash firmware to the autopilots for multirotor vehicles and several types of other vehicles. This software is particularly powerful in that most popular multirotor frame types are supported out of the box. In addition, Mission Planner simplifies the entire calibration procedure to a few button clicks, with informative system dialogue pop-ups to make it extremely easy for non-technical individuals to set up their own aerial vehicles. However, every system parameter is available for tweaking, which allows for infinite customization. There is documentation readily available online for correct setup and use of the Mission Planner software [28].



Figure 22: Mission Planner allows for easy setup, deployment, and customization of UAVs.

2.3 Warnings and final notes

One of the most important things that goes along with building your own drone is a respect for the technology. Regardless of if you build custom or buy off the shelf, these devices are not toys and can cause serious harm and property damage if they are used unsafely or if things go wrong. If one of these devices fails while above people, or hits a car on the highway, the impact can be lethal. There are also numerous privacy concerns from the use of camera equipped drones in urban environments. Most of these concerns are from only a few users flying irresponsibly, but their actions disproportionately affect public opinion. For example, the city of Richmond, British Columbia, Canada banned drones within the city limits [16] in response to safety and privacy concerns raised from a drone crashing near a playground and another drone that was seen flying up and down apartment buildings.

Some people see drones as a way provide both a fun and engaging hobby for consumers, as well as a new and powerful tool for everything from law enforcement and search and rescue to delivery and inspection. Public opinion and legislation is currently balanced between those that see the promise and those that see the risk. Because of this delicate balance it is critical that anyone who wants to fly a drone make sure that they do so in a responsible way. This means that you must be aware of all applicable laws and regulations that govern flying in your area, as well as make sure to avoid situations that are potentially dangerous for yourself or others, and anything that can be seen as a privacy concern [13]. By handling this technology with care the drone community can continue to remain in good standing with the community and with law enforcement.

2.4 Other Resources

2.4.1 Part Suppliers

- HobbyKing

Hobbyking provides reasonable quality parts at a low cost. The downside is that some

of their parts are very poor quality or just complete copies of other work. That said, in recent years they seem to be making an effort to provide good parts at a discount. Another important factor in ordering from Hobby King is to make sure to order from a warehouse that is in the same region as you. Otherwise shipping can take a very long time [21].

- Ready Made RC

Ready Made RC is another online hobby shop, and provides a good selection of parts at a reasonable price. Their parts are also a bit higher quality than most of what Hobby King sells [38].

- Horizon Hobby

Horizon is a big name in the RC world. Like Ready Made RC, the quality and prices of their products are normally higher than Hobby King [22].

- Amazon

You can get everything on Amazon. If you have a Prime membership then getting parts very quickly is easy if they are in stock. The industrial and scientific portion of the Amazon storefront also carries a wide range of fasteners, tools, tape, and other equipment that can be very useful for constructing a drone [9].

2.4.2 Information and Instructions

- DIYDrones Blog

DIYDrones is a blog style site where members post relevant news articles and information on both new technology and personal projects. There are also more links here for people who are just getting into building [14].

- FliteTest

FliteTest is an Ohio based group that is dedicated to growing the RC hobby. They are extremely friendly and down to earth, and have designed all of their products with the

goal of making them as cheap and easy to build as possible. They also have forums and video productions that go into detail on everything from very basic build and flight principles to advanced flying tricks and antenna designs [19].

- **RC Groups**

A large online community that is very knowledgeable about everything RC related. The site contains forums for everything from rockets to sail boats. There are some very technical and knowledgeable users on the site, however the information available is not well curated.[36].

2.4.3 Battery Information

- **RC Helicopter Fun**

This website provides an excellent description of the various battery types used in RC vehicles. The article outlines battery safety, compares different battery types, and also provides an understanding for how batteries are built [37].

2.5 Summary

UAVs are comprised of several basic parts, and an understanding of all of them is necessary in order to construct a flyable vehicle. These basic parts include batteries, electronic speed controllers, motors, propellers, autopilots, RC radios, and a frame. Frames come in different configurations, such as quadroters or hexarotors, and the pros and cons of each need to be weighed before choosing.

Building a custom drone using off-the-shelf components is generally cheaper than buying a pre-manufactured model, but there is a slight learning curve when building one from scratch. Out-of-the-box solutions are ready to fly and come with powerful software that has tremendous functionality [42], however there are compromises made in the design to keep the prices reasonable [12].

There are safety concerns with flying drones as well, and hobbyists need to be aware of the risks and laws in their respective areas. The FAA ruled that all fliers must have line of sight to their drones and must remain in complete control of their vehicles at all times. In addition, flying within 2 miles of an airport or above the 400ft limit is prohibited. Finally, these drones are usually a few pounds worth of weight carrying explosive batteries. Misuse or irresponsible behavior while flying can cause personal injury, property loss, or death.

3 Planning and Logistics

This project involves the use of drones to survey a wide area for different applications. In order to realize this goal, two major sections of the project were realized: the frame design, and payload design. In order to limit the scope of the project, a tiered list of project objectives was defined. From this tiered list, a set of core requirements were specified, and formed the foundation for each iterative frame design, and payload implementation. This section explores the project's core requirements and the tiered list of goals. In addition, individual frame designs are explored, as well as the requirements for an effective payload.

3.1 Individual Design

Over the course of this project there were a number of revisions and experiments relating to the frame designs used for the multirotors. These designs ranged from a store bought platform with minor modifications and additions to completely custom designs manufactured using a variety of techniques. However throughout the evolution of the frame designs, the goals for the system remained the same. These goals were determined from the goal tiers detailed in the logistics section of chapter 3.2. The core requirements were:

- Sufficient payload capacity for all of the MQP related electronics. Based on the estimated weight of the desired components, a target weight of 0.75kg was selected.
- A minimum of 10 minutes of flight time. 10 minutes was chosen as the minimum endurance needed to survey an area large enough to yield meaningful data. Minimizing this number allowed for lower cost parts to be used and reduced the challenge presented by the target payload. This allowed for the use of lower capacity and thus lighter, less expensive, batteries.
- Efficient use of already owned parts. For example, design or specify new parts that are compatible with existing systems to reduce cost and time required to integrate new components.

- Flexibility to accommodate changing goals based on new results. As the project continues, and the basic goals are accomplished, the system needs to be able to be extended to add new functionality per the listed project goal tiers.

From here the list of parts that was already owned was consulted, and a hexacopter configuration was chosen. Hexacopters are able to carry a larger payload and battery than quadcopters using the same size motors and ESCs. The control electronics and sensor requirements are the same, and thus cost of the control system is the same, for both frame layouts. Because the team members had already built quadcopters, simply adding two motors to each setup was more cost effective than buying a completely new set of motors and controllers for the desired payload capacity. The electronics were the same for all the frames, and the primary difference was simply the material and layout of each frame. The first frames that were used for this project were based on designs that were made for earlier explorations into multirotor aircraft.

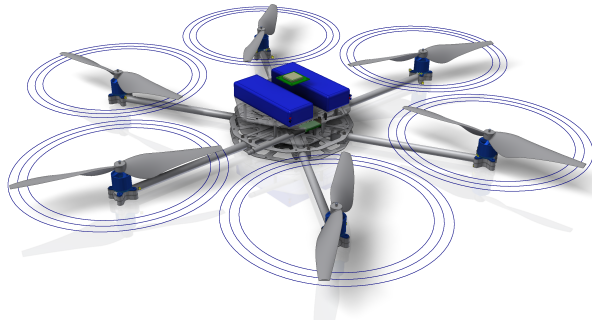


Figure 23: A hex frame was designed in CAD to provide a lightweight, minimalistic hexarotor frame.

The first frame that was used for the project was a hexacopter design that used aluminum as its primary material. The design was completed D-Term 2014 and the parts were milled in Washburn shops. The frames were built from a combination of 7075 aircraft aluminum for the main plates and 6061 T6 architectural aluminum for the spars and the milled clamps. One of the main features of this design was a method for folding the front and back spars so that the frame occupied far less space for storage and transport.



(a)



(b)

Figure 24: The assembled hex frame provided high performance due to its lightweight, minimalistic design.

The main problem with the design came from the 3D printed clamps that were used for the folding spars. The finished parts did not handle the loads that were put on them while flying, and cracked while under load. To solve this problem quickly a commercial frame was purchased. Another frame was designed out of wood that could be manufactured on the Washburn laser cutter. This started the parallel development of the drones used in this project, which continued through to the end and allowed a larger range of experimentation.



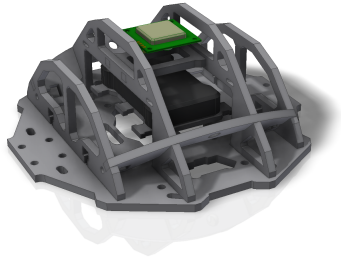
Figure 25: Manufactured frames are available that provide easy construction and good performance. The Tarot 680 Pro is one of them. [39]



Figure 26: The Tarot 680 Pro is an easy to construct off-the-shelf hex frame solution.

The commercial frame that was chosen was a Tarot 680 Pro. The 680 Pro is a hexacopter frame that comes with features such as integrated power distribution, folding spars for easier storage, and vibration isolated camera rails. Integrated power distribution is a useful feature as it keeps the wiring clean and helps to reduce weight by eliminating excess wire runs. The camera rail was used to mount the MQP electronics and camera, and helped to eliminate artifacts caused by vibration in the images captured. One issue that we encountered with this frame was that the clips used to secure the spars in the open or closed positions were sensitive to temperature, and so were prone to breaking in cold weather. The work around that the team used to address this issue was to simply use zip ties to secure the spars in the open position, allowing the frame to still be flyable, but making it impossible to fold for storage.

Another challenge that needed to be addressed was the lack of mounting points for the flight electronics. These electronics consisted of the APM 2.6, GPS and compass modules, and the radios for control and telemetry. The solution was designed to be manufactured out of plywood using a laser cutter. This frame worked well, and was later improved by the addition of a 3D printed hood for better weather resistance.



(a) CAD Model of Electronics Bay



(b) Printed Electronics Hood in Red

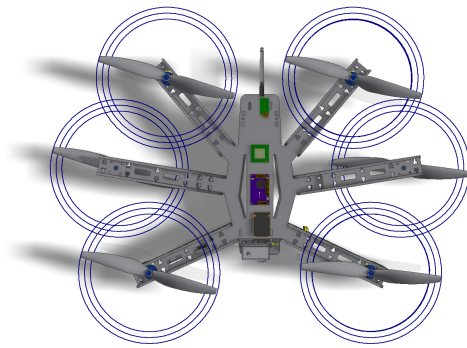
Figure 27: CAD Model of laser cut wood electronics bay and printed hood for weather resistance.

Once the hood was printed, this frame was tested by flying in moderate rain and heavy snow. The frame handled this well, with no apparent loss of performance or other ill effects. The larger challenge encountered when flying in adverse weather was the comfort and willingness of the pilots, and the weatherproofing of the ground station equipment.

The second hexacopter that was used on this project followed a more experimental development cycle. The first frame design that was used was made out of laser cut plywood and was designed to simply provide as much room as possible for the electronics. This frame made tradeoffs in favor of simplicity and low cost as opposed to efficiency and sleekness. That said, the frame performed well, and was extremely easy to fly.



(a)



(b)

Figure 28: Multiple frames were designed in CAD to test performance.

The reason that the wood frame was eventually retired was the the difficulty of weatherproofing the electronics, as well as its poor performance in windy conditions due to its large surface area. The frame that replaced it was an experiment with an alternate propeller configuration. Instead of mounting the blades in a hexagonal pattern around the center of the frame, this design mounted the blades in pairs with one blade above the other. This is known as a “Y6” configuration. The advantages of this design are a reduction in sail area, and wider spacing between each of the spars, which helps to keep the camera view clear. This frame was custom designed to make use of parts that were left over from the original aluminum hexacopters that had been made in D-Term of 2014. The additional parts that were needed for the new frame configuration were manufactured out of plywood on a laser cutter.



Figure 29: The Y6 Hex frame is another frame type experiment.

However this design did have major drawbacks. The motor configuration meant that there was a roughly ten percent loss of efficiency compared to the normal hexacopter layout, which led to a proportionally shorter flight time. The motor layout also meant that there were effectively fewer points of control for pitch and roll, which led to reduced stability compared to the normal hexacopter layout. These drawbacks led us to develop the final frame design for this hexacopter.

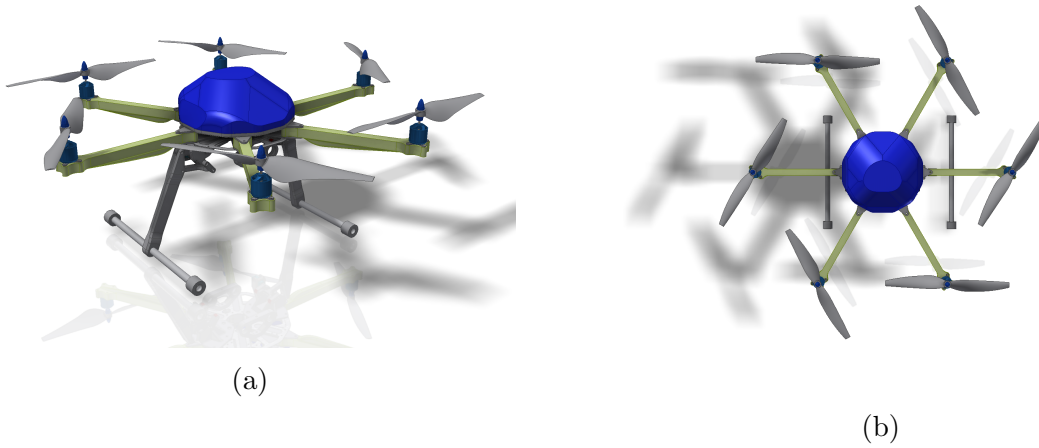


Figure 30: The yellow hexacopter frame was inspired after the aluminum hex.

The final design that we are using uses the lessons that we have learned from the previous frame designs, and makes extensive use of 3D printed parts. The layout is the standard hexacopter layout, with booms sized to allow up to 12 inch propellers. The electronics use the same type of mounting bay as was designed for the commercial frame, and a similar 3D printed cover for weather resistance. The payload is mounted on camera rails that are compatible with the rails used on the commercial hexacopter. An additional feature that was added with this frame design is the ability to fold the landing gear out of the way of the camera.

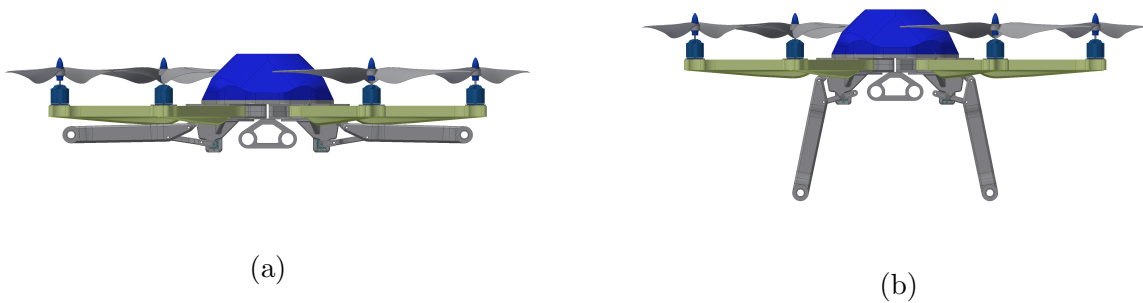


Figure 31: CAD Models showing the landing gear in the raised and lowered positions. The system is powered by small servos in the base of each leg.

This mechanism is designed so that over the range of motion of the servo used to actuate it, there are two “toggle points” where the load is entirely held by the configuration of the mechanism, and the servo motor experiences no load. This is useful as it helps to reduce the power consumption of the servos as well as minimize the risk of damage to the servo during landing.

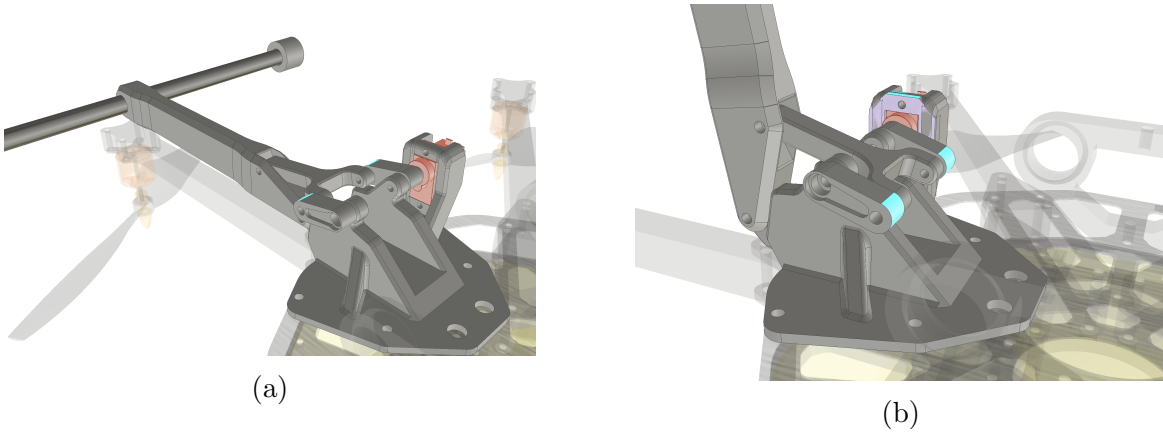


Figure 32: CAD Models showing the landing gear toggle points.

One point of improvement on this frame would be the spar design. The current design gets its strength using a relatively high infill, which makes each spar somewhat heavier than it needs to be. Additionally, better wire management of the ESCs and motor wires would improve the look and performance of the frame and help with weatherproofing. This is a point where minimal additional work would yield noticeable results.

3.2 Logistics

At its core, this project strives to create an easy to use and expandable drone network for search and rescue, agriculture, and other applications. Successfully catering to one application is a large undertaking, not to mention developing a dynamic system that can cover all three. Therefore, in order to maintain focus and ease the development process, three different tiers of goals were defined at the start of the project. Each set of goals covers a subsequently larger section of the features in the use cases features list.

The first tier consists of several features that are considered “basic functionality”. In other words, our project needed to achieve these goals at the minimum in order to be considered successful. Basic functionality would be achieved when:

Table 2: The Tier 1 goal list represents the bare-minimum requirements for the completion of this project. These are the ”must-haves”.

Done?	Tier 1
X	Drones can be manually overridden at any time
X	A telemetry link to drones is present at all times
X	Missions are planned and executed for at least two multi-rotor airframes
X	Both drones capture down looking video/images
X	Capture time is matched with corresponding global pose of drone
X	Data is automatically downloaded to ground station when in WiFi range
X	Images and drone location can be reviewed

Once these goals were achieved, the next two tiers of goals were designed to greatly expand and in some cases exceed the various requirements for each use case. The second and third tier goal sets are listed below.

Table 3: The Tier 2 goals represent the "nice to haves" for this project. They expand on the bare requirements but are not mandatory for the completion of this project.

Done?	Tier 2
	Functional UI for controlling multiple drones
X	Weather-resistant drone designs
X	Real-time video/data link to ground station
X	Abstracted support for multiple types of drone platforms
X	Abstracted vision support system
	Advanced mission control
	Auto takeoff/land
	Spline paths
	In-flight replanning/reassignment
	Automated battery time estimation planning
	Onboard video stitching
X	Complete a delivery mission
	Video tutorials on drone construction

Table 4: The Tier 3 goal list represents the unrealistic goals. These are the "super nice to haves" which show how this project's scope can quickly exceed realistic expectations.

Done?	Tier 3
	Advanced intuitive UI with touch integration
	Virtual reality interface for Oculus Rift/Razer Hydra
	Real-time onboard feature recognition
	Group tasking/automated formations
	3D generated environments
	Integration with other MQPs
	Battery swapping
	Communications system
	Path planning for ground vehicles
	Depth information integration
	Lab guides, detailed educational experience

Tier 2 is focused mainly on upgrading tasks completed in Tier 1, while Tier 3 contains features that are could each be project on their own.

3.3 Camera and Image Processing

An effective payload consists of camera and image processing capabilities. These capabilities of the drone network are paramount for agricultural, search and rescue, and many other applications. The prototype system developed for this project relies on a 5MP "webcam" type camera manufactured specifically as an accessory for the Raspberry Pi using the name "PiCam". However, the system was designed to be modular, so higher resolution or different types of cameras could be added. In addition, the Raspberry Pi does not have the compute performance needed to do advanced image processing, thus a better high-level controller

could be substituted in, although there would need to be some software calibration in order to accommodate other platforms.

Images captured by the PiCam are stored locally on the drone, and are sent off to the ground station computer for processing once a WiFi connection can be reliably established. Once the connection is determined to be stable, the ground station will begin receiving files as they are captured, and present them to the user. Due to the direct link between the Raspberry Pi and the APM, mission information and images can be simultaneously presented to the user, and can be incorporated into the post-processing pipeline.

Mission information is sent between the APM and MissionPlanner through a protocol called MAVLink, which was created specifically for the use of micro-air vehicle communication [26]. Information is passed via MAVLink messages. Thus, incorporating mission information from the APM into image data involves two processes:

1. A serial link exists between the APM and Raspberry Pi
2. A software module called MAVProxy actively receives and decodes MAVLINK messages received over the serial link [27].

The higher-level drone software then sorts through the received messages to locate and store drone pose and GPS information, which is then appended to the image data stream from Raspberry Pi to the ground station computer.

The ground station computer runs two pieces of software: 1) MissionPlanner, which receives and decodes MAVLink data through a dedicated telemetry stream, and 2) GSNetlink, our own ground station software. GSNetlink communicates over WiFi to each drone, receiving position information and images as they are taken and stored by the drone's high level controller. Pose data is stored in the form of an XML data file, while each image is received in JPEG format.

Once the images are downloaded to the ground station computer, all information is present to do as much post-processing as required. Pose information and image data are

stored in conjunction with each other, making it trivial to correlate the two. For this project, three processing schemes were explored.

The first was VisualSFM [7], which iterates through images to create a 3D reconstruction of the surveyed area. This is done by performing feature recognition on the entire image set, and then comparing each image to every other image for possible matches. Once possible features and images are correlated, a ray-tracing algorithm is used to back-calculate the camera position and create 3D points. However this process could not be preformed in real-time.

The second was a custom C# application developed in the game engine Unity. The goal of this method image processing is to make use of the pose information to project the images onto a virtual plane. This would leverage the existing graphics engine running Unity to reduce the development time needed to implement the system. Unfortunately this method proved impossible. See section 5.5.3 for more details.

The third and final method was a custom C++ application using the OpenGL graphics library to perform image stitching based on the received images. In a similar fashion to the Unity visualization system, the images and meta data are used by the system to determine where and how to align the images. Without the Unity game engine the trigonometry and image stretching operations were performed by hand in the application. Unfortunately due to time constraints this line of development was halted. Preliminary results and work can be found later in this report.

3.4 Summary

The planning of this project was critical to its success. Since each of the project members had experience in building and flying drones in the past, there was little overhead in getting units in the air.

Several frame designs were experimented with to see how they affected stability and flight performance, with the flat hex design proving to be the most robust. In addition, a three-

tiered list of goals were created to guide the project's completion. Finally, the necessary software and hardware of the camera and image processing system were defined in order to realize the system's basic functionality.

Several post processing techniques were explored and their respective advantages and disadvantages were noted. The complexity of developing a efficient image processing system quickly exceeded the scope of the project.

4 Implementation

The final implementation of the drone system has come together as a merging of each individual module. Each drone contains multiple systems which interface with the ground computer in different ways. The ground computer hosts the rest of the system, containing software for control, data transmission, and data utilisation. This section describes the implementation of each of these systems along with the architecture of the entire system.

4.1 System Architecture

The system architecture consists of a network of nodes with a single hub controller. The ground computer is the main hub of the network, which directs and receives data from each node. The general system architecture is shown in figure 33.

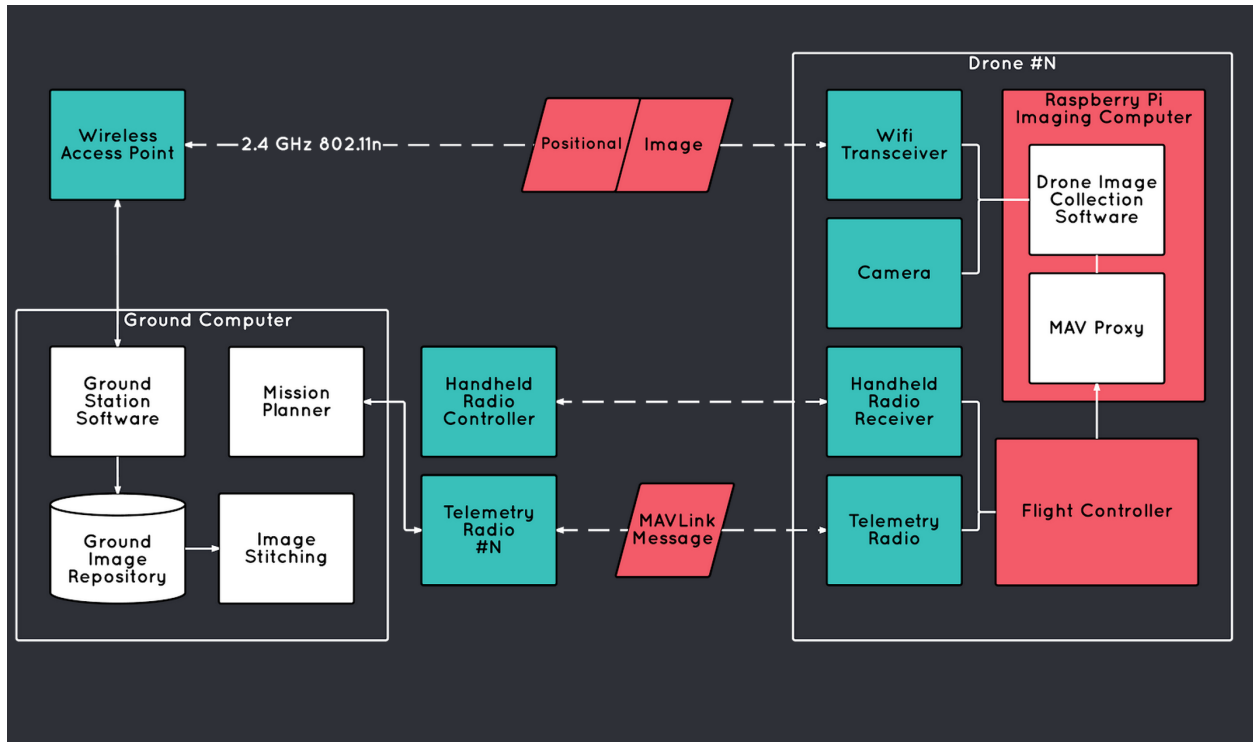


Figure 33: The overall architecture of the system is dependent on a ground station computer and multiple drone subsystems.

The ground computer acts a host for the software components which form the basis of

the two main control chains for the entire system. One of these is Mission Planner, a piece of open source software that provides flight control direction to each drone in the system. The other is the Ground Station software designed in this MQP to receive and organize images and data received from all drones in the system. This software compiles a database of images and data, which is accessed by stitching software to create a single map overlay. The individual design of these modules is discussed in the Software Section. The link between the drones and the ground station computer is bidirectional.

Mission Planner communicates to each drone over a 900MHz wireless telemetry link. Each separate drone in the system requires a separate telemetry radio for communication. The telemetry radio on each drone pairs to its own unique telemetry radio connected to the ground station. Possible improvements to this part of the system are discussed in the Conclusion. The telemetry connection to each drone carries messages containing waypoint information as well as sensor data for each drone. These messages are encoded using the MAVLink protocol, an open-source messaging system used for UAV communication. The drone sensor data tracks position, airspeed, and battery health information. The telemetry on the drone is connected directly to the flight controller.

The other communication link with each drone is over an 802.11n wireless link. This link is how the ground station software receives images and data from each drone. The WiFi connection is established using a 2.4 GHz dish transmitting with 28 dBm of power. The drones each carry a USB network adapter with dual 6bBi dipole antennas. This connects to the Raspberry Pi on each drone and is used by the image collection software. The network created for this link uses static IP addresses for each drone. This design could be improved to allow for better expandability by using DHCP to assign addresses to each drone instead. The system on each drone can be seen in figure 34.

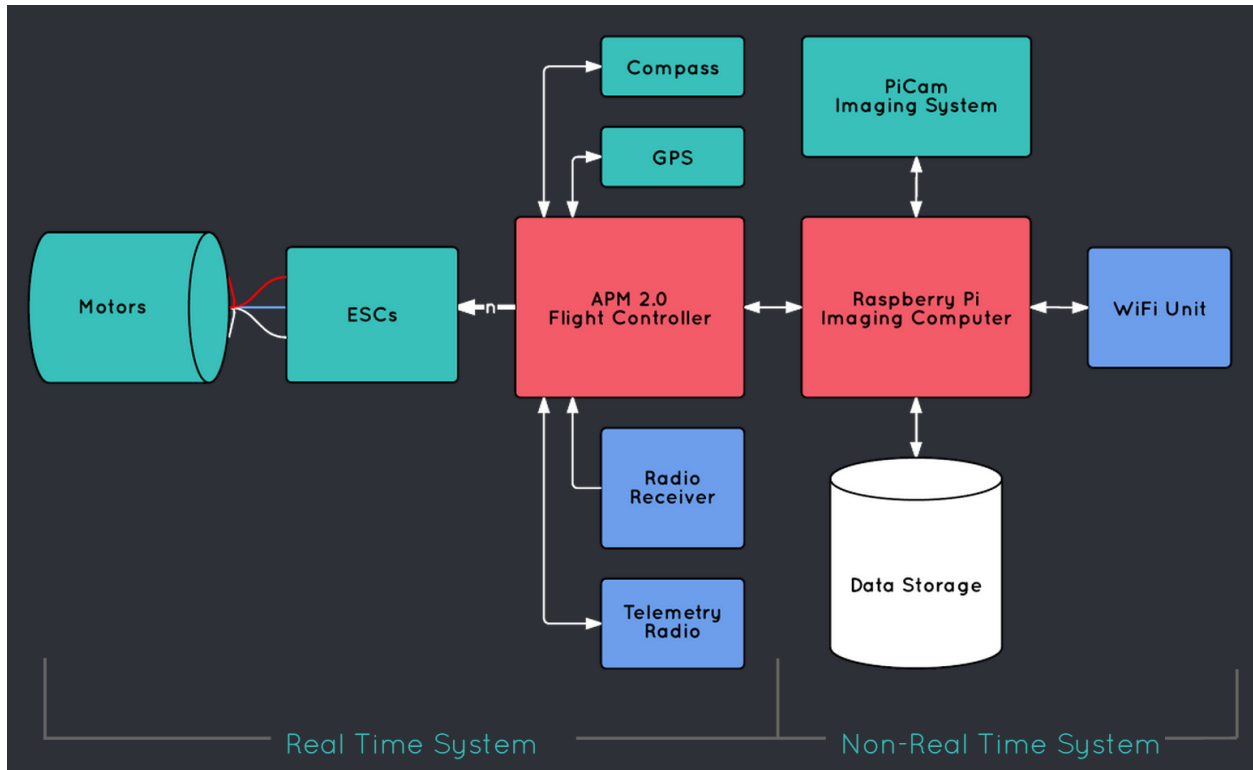


Figure 34: The drone system architecture is dependent on both real time and non-real time components.

The ground station architecture is made of software modules dedicated to presenting information collected from each drone to the user. The hardware of the ground station consists of a high-powered WiFi antenna, a medium power telemetry radio, and a general purpose personal computer running our software and Mission Planner.

At the fundamental level, each drone is comprised of a low-level controller, which runs the flight control algorithms. These algorithms output signals to each motor controller, and subsequently each motor. Data from compass and GPS modules provide the necessary information to the APM to maintain stable flight, while a telemetry link provides real time system status for the end-user at the ground station. In addition, a radio receiver provides manual override for safety in case of system failure.

In order to gain imaging functionality, a high-level controller was added with a bridge to the APM. Several different options were considered for this device including the BeagleBone Black, Raspberry Pi, Intel Edison, and using a mobile phone. The Raspberry Pi was chosen

because it was easy to acquire, simple to develop for and also has several open source libraries for interacting with cameras and networking. Connected to the Raspberry Pi is the imaging system, which consists of a 5MP camera. In addition, local storage is provided in the form of a high capacity SD card, while a WiFi module enables wireless connectivity from the Raspberry Pi to the ground station. The bridge between the APM and the Raspberry Pi is used to associate useful information from the APM's mission data to pictures collected by the imaging system, i.e., GPS coordinates and drone pose data to simplify search area reconstruction.

Each subsystem was designed to be as modular as possible, meaning that any given component can be upgraded to yield higher performance. For example, higher powered transmission systems for both the telemetry and WiFi links would provide longer transmission distances and higher data throughput. In addition, since the drone subsystem is based off the ArduPilot platform, nearly every component can be swapped out for better performing (more expensive) parts with little to no configuration needed.

The Raspberry Pi can also be swapped out for a better performing platform. However, a function equivalent to the software written for the PiCamera would need to be written for the new device.

4.2 Software

The software written for this system consists of two separate modules, each with their own set of dependencies. Each subsystem has its own specific software. The drone software is responsible for aggregating data from the imaging and real-time systems and transmitting it to the ground station. The ground station software is responsible for facilitating the connection between each drone and storing received data.

4.2.1 Ground Station

The ground station software, GSNetlink, is responsible for facilitating the transfer of image and positional data from each drone in the system. The ground station software is made up of three python modules: gsMain.py, gsNetClass.py, and DroneData.py. The gsMain python module is dependent on gsNetClass and DroneData, and is executed by the user.

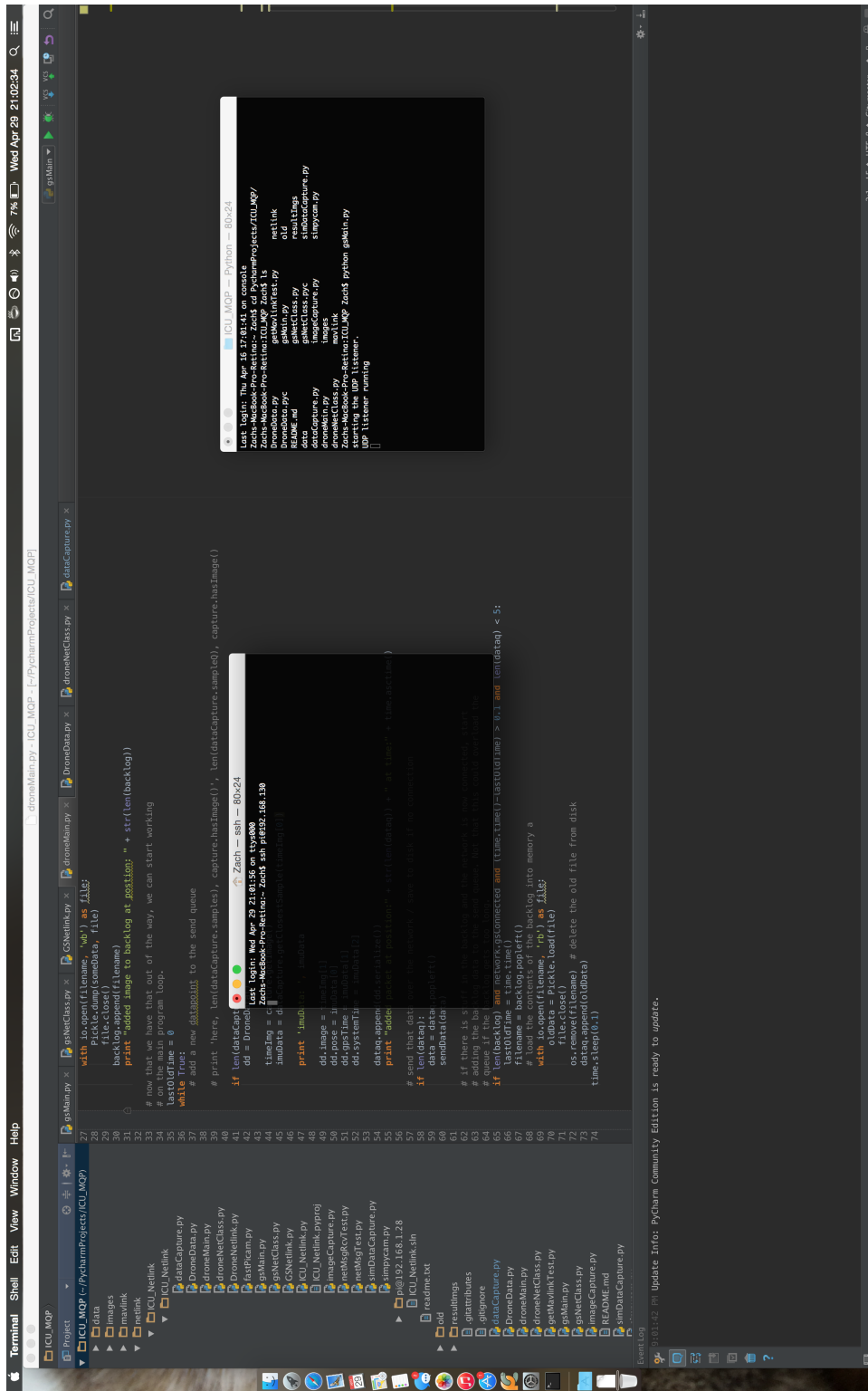


Figure 35: The ground station software involves running a python script that facilitates wireless connections to the drone. An SSH link is present to the drone to initiate the drone software if it fails to launch.

The gsNetClass python module handles all networking configuration between the ground station and each drone. This is accomplished by listening for UDP traffic containing a TCP port and address sent by a drone. If the received TCP address is not already connected, the gsNetClass spawns a TCP connection to that address on its own thread to handle incoming data streams. Multiple threads were used to ensure that multiple drone connections and data streams could be handled simultaneously. In addition, the gsNetClass python module contains methods for receiving and storing incoming data streams, as custom packets containing serialized image and positional data are used for communication. Received messages are stored in system memory.

The DroneData python module contains serialization and deserialization methods for transmitted and received data between the ground station and each drone. By defining a custom packet, the amount of data that can be sent in one message became a variable that could be adjusted to achieve maximum efficiency for the wireless network.

The gsMain python module handles all file I/O and data storage. It executes gsNetClass to obtain the received data streams from each drone stored in system memory. If there are received messages stored in memory, gsMain creates a directory for the respective drones and calls on DroneData to deserialize the message. Once the data is deserialized, the image data is separated from the positional information. The image data is written to an image file while the positional data is written to an XML file. The file names are set to the system time at reception. By doing this, the relationship between the image data and pose information is maintained, simplifying post-processing.

In addition to GSNetlink, another piece of ground station software is necessary for smooth system operation. This piece of software is called Mission Planner. For this project, Mission Planner connects to each drone over a dedicated telemetry link to display each drone's GPS position on a GoogleMaps-like interface, in addition to displaying system status conditions such as pose, velocities, and navigation waypoints. Mission Planner provides a simple at-a-glance presentation for each drone's system status.

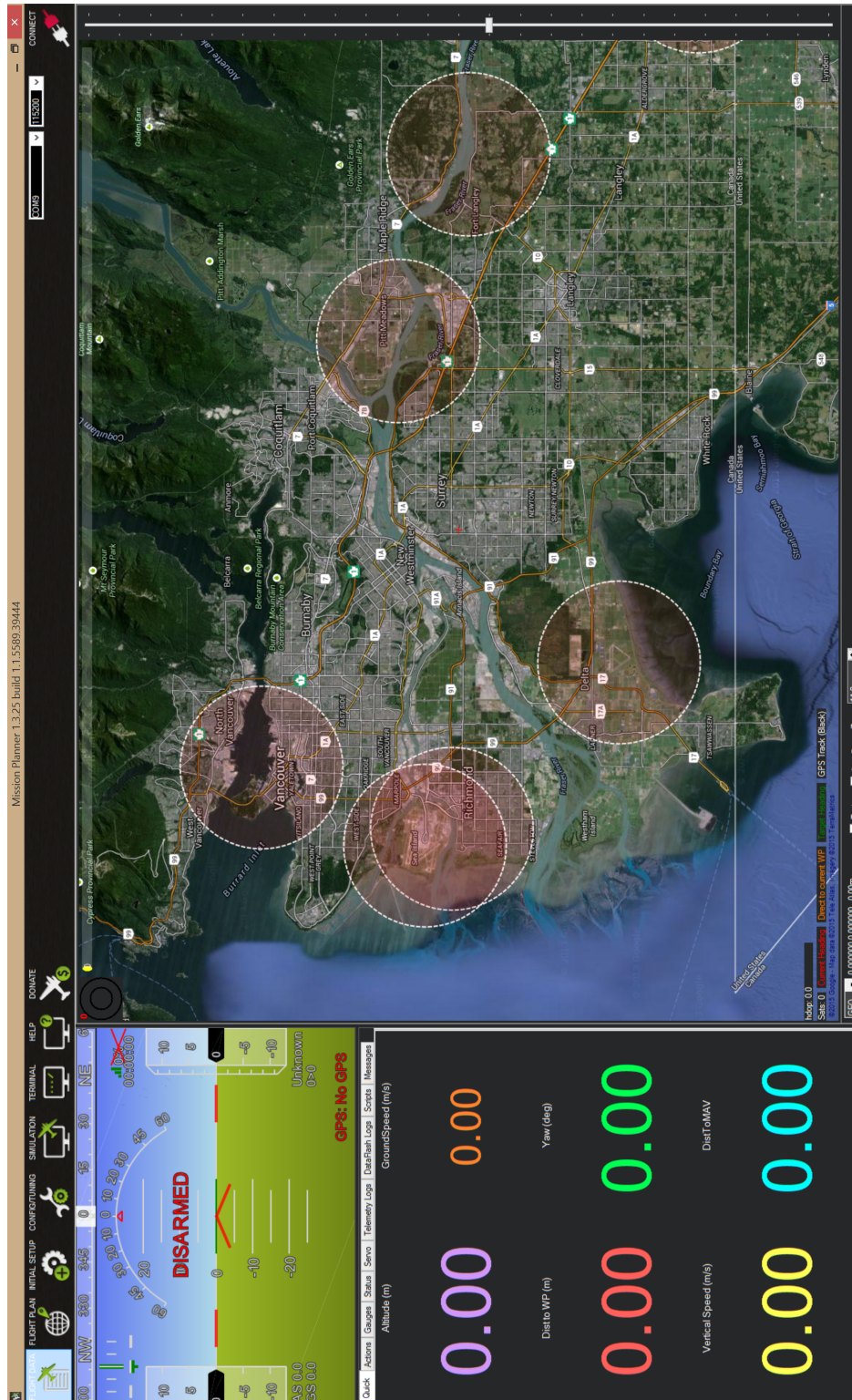


Figure 36: A screen capture of Mission Planner running. This is the software used to set the mission waypoints and configure the autopilots on the drones.

4.2.2 Drone

The drone software, DroneNetlink, is responsible for initiating wireless communications to the ground station for data streams, capturing images from the PiCamera module, and collecting positional data from the autopilot. The DroneNetlink software is split into several python modules: `droneMain.py`, `DroneData.py`, `droneNetClass.py`, `dataCapture.py`, and `imageCapture.py`. The `droneMain` module is dependent on all other modules in addition to an external piece of software, MAVProxy, and is executed by the user.

As with the ground station software, the `droneNetClass` python module manages the connection to the ground station over the 802.11 wireless link. The `droneNetClass` opens a UDP port and then begins to send TCP connection requests repeatedly through that port. The TCP connection details are automatically generated by the built-in socket library so that unused TCP sockets are chosen [35]. This module keeps sending TCP requests over the UDP port until a response is acknowledged by the ground station, and a TCP connection is established. This python module also includes methods for sending data through the connection once it is established.

Both `dataCapture` and `imageCapture` collect data from their respective sources. `ImageCapture` spawns a thread to handle capturing image data from the PiCamera, and stores captured images in system memory. Once the `imageCapture` python module is started, it will continue to capture images into a queue until told to stop by `droneMain`. `ImageCapture` uses a python module provided by the Raspberry Pi project to interface to the PiCamera [31]. This module contains methods to initialize the PiCamera, capture data, and tweak capture parameters such as resolution, exposure, and data output type.

The `dataCapture` python module operates in a similar way. However, instead of capturing image data to a buffer in system memory, it reads positional data to a buffer in system memory. When executed, `dataCapture` spawns a thread to read MAVLink messages through MAVProxy. MAVProxy is a separate module developed by the QCGroundControl project [27], which provides a way to read MAVLink messages over a serial connection to the au-

topilot. DataCapture then uses a MAVLink library containing message definitions to sort through the received message IDs to find the corresponding messages containing positional data. The MAVLink library is provided by the QCGroundControl project [26]. As data is received, it is stored in a circular buffer in system memory so the most recent positional data is maintained. This module also contains methods to retrieve data samples from this buffer.

The DroneData python module contains serialization and deserialization methods for transmitted and received data between the ground station and each drone. This is the same module used on the ground station. Since it is executed on the drone, only the serialization methods are used for transmitting data to the ground station.

The last piece of software run on the drone is the droneMain python module. This module initiates the droneNetClass, dataCapture, and imageCapture modules. DroneMain serializes image data stored in system memory from imageCapture, serializes positional data stored in system memory from dataCapture, and prepares messages for transmission in another buffer. If a connection is present as determined by droneNetClass, droneMain will then begin transmitting messages, starting with the oldest. If no connection is present, droneMain will save the oldest messages in the buffer to disk once the maximum buffer length is exceeded. This backlog is prioritized once a connection is re-established.

4.2.3 Ground Station Visualization

Several different methods were explored to view the images acquired by the drones on the ground station. These included 3D point visualization, adapting existing 3D software, and creating a custom visualization software. Each of the implementations explored required different processing capabilities and input data. The first and most computationally demanding of these methods was 3D point reconstruction.

Structure from motion or SFM, is a process by which a series of images can be used to recreate the topology they observed. This process was performed after a flight was completed and required large amount of computation time. In order to perform SFM this team selected

the VisualSFM program published by Changchang Wu for its simple interface and GPU optimizations [7]. The process runs in several discrete steps: feature detection, feature matching, and then finally bundle adjustment. In feature detection an algorithm called SIFT iterates over each image in the set, selects unique groups of pixels, and marks their location and properties. It then stores that information in a text file for each image [44].

This data is then used to match images that have overlapping content. In order to determine adjacent images every set of features for each image must be compared pairwise with every other, causing this section of the algorithm to have $O(n^2)$ time complexity. After images have been determined to be adjacent the 3d points are reconstructed using a system called bundle adjustment. During this process the matched features are used to align the images spatially while also predicting location and orientation of the camera. This is done by solving a system of matrices for each image and 3D point [20].

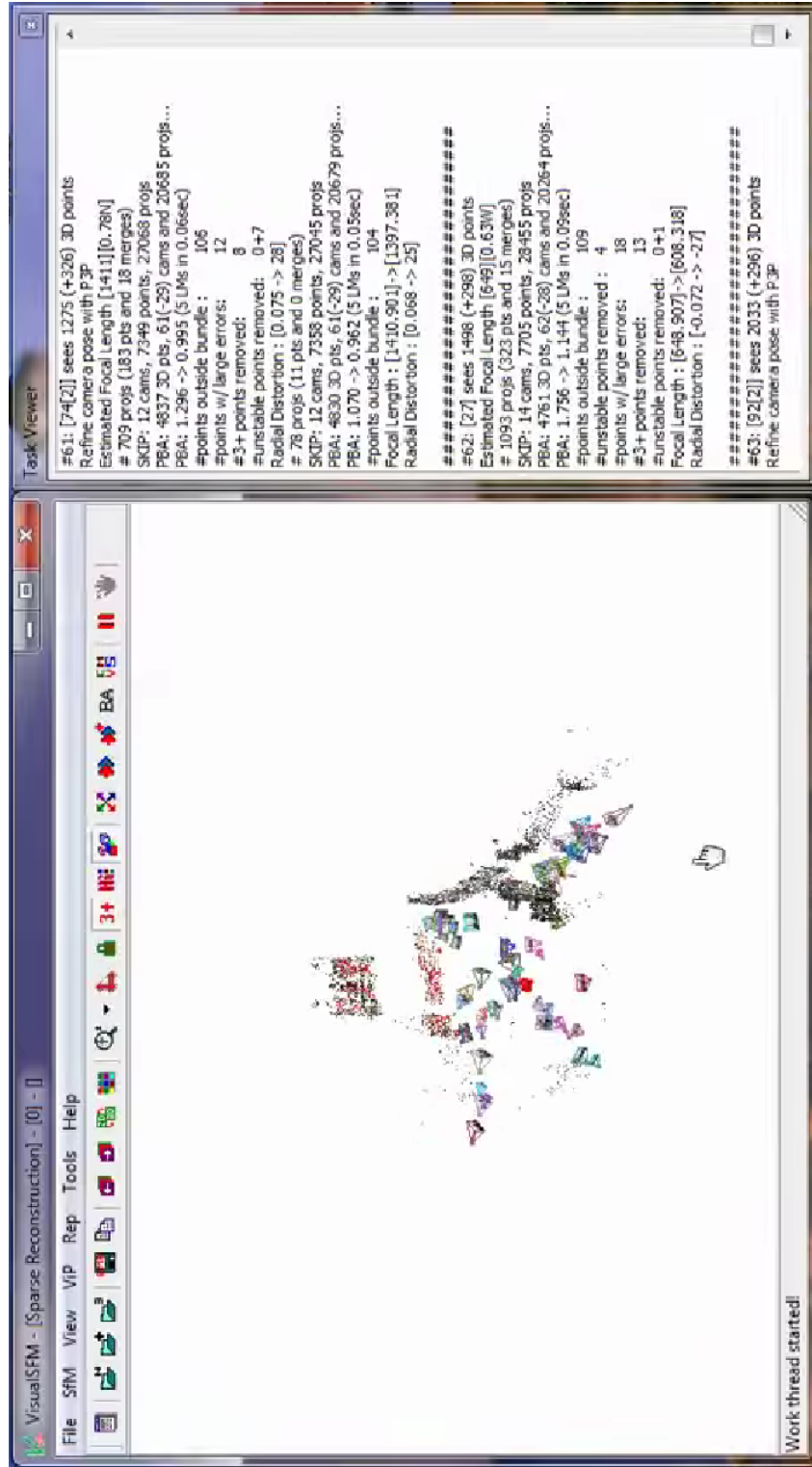


Figure 37: Data being processed using the GUI in VisualSFM [7].

Although the VisualSFM package is optimized for graphics processor unit (GPU) and multicore processing, the process is still very computationally expensive for large data sets. This is due to feature pairing having time complexity $O(n^2)$ and the bundle adjustment and feature recognition processes having time complexity $O(n)$ [41]. The feature matching search that VisualSFM performs also entirely uninformed and therefore cannot make assumptions about distance to objects or camera orientation. Additionally VisualSFM calculations require that the entire data set is imported at once and therefore cannot run while the vehicles are flying. For these reasons additional means of displaying the received image data were investigated.

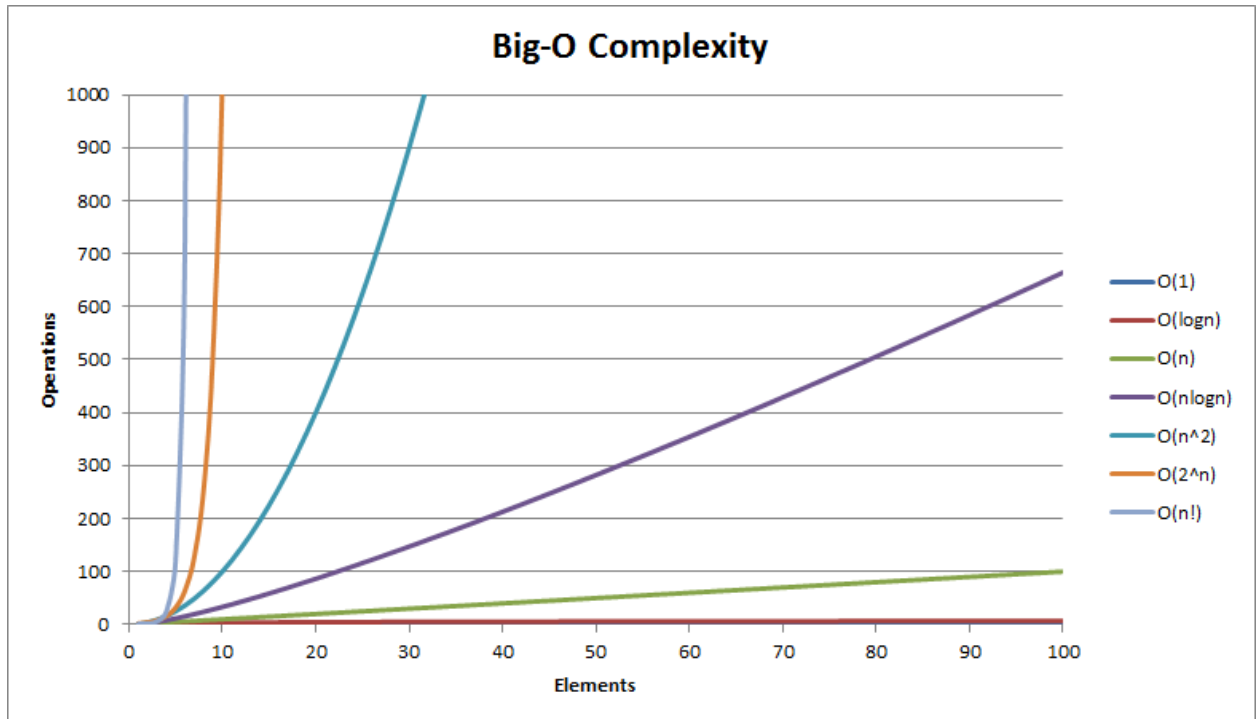


Figure 38: Big-O-Complexity, number of items being processed vs. number of operations required to process those items [11].

The second display method investigated was using an existing game engine, Unity [6]. A C# program was written to read a folder as the images arrived and load not only the image itself but also an XML file describing the GPS location, altitude and orientation the photo was taken at. This data is then used to orient a projector onto a surface representing

the ground being flown over. Traditionally the projector is used to project shadows or other game elements around a character in the game, but for this project it was re-purposed to project the acquired images [5]. By simply orienting this projector in the same manner and location as the drone was when the image was taken the image is correctly displayed on the plane in a 1:1 relation to the actual area the UAV was flying over. Unfortunately during the development process several issues with this method were discovered. These issues are further discussed in section 5.5.3 of this report.

The final method for displaying the images returned by the drones explored in this project was a custom implementation using C++ and the OpenGL graphics rendering library [4]. When it was realized that the Unity method would not be viable, OpenGL was chosen as the graphics library for a custom solution. In addition to being an industry standard library, OpenGL is also cross platform, which meant that any solution developed for it could be carried to mobile or other devices with little to no changes.

OpenGL only provides a bare bones interface to create planes and other basic geometry features, so the math to calculate the proper image size, orientation, and offset had to be derived. Assuming that UAV is flying over an area that can be approximated to a level plane this math can be simplified to a trigonometry problem. There are several important variables in this equation. The field of view of the camera or FOV describes the extent of the observable world visible to the camera. This is measured as two angles, both vertical and horizontal, and for the Pi camera used in this project they were found to be 41.41° (θ) and 53.5° (ϕ) respectively [32]. This value, combined with altitude above the ground z can be used to calculate the distance over the ground the image covers, x with equation (6) below.

$$x = 2 * z * \tan\left(\frac{\phi}{2}\right) \quad (6)$$

However this equation will be invalid if the drone is not perfectly aligned with the ground plane it is flying over. Fortunately the accelerometer and gyro on board provide highly accurate orientation data for the drone. The pitch and roll data for the drone can be directly

inserted as an offset for this equation. However in order to do this the distance over the ground must be broken into two components, in the case of the horizontal dimension the distance to the left of the drone and the distance to the right of the drone. Assuming a roll offset α , the equations now become equations (7) and (8).

$$x_{left} = z * \tan(\phi + \alpha) \quad (7)$$

$$x_{right} = z * \tan(\phi - \alpha) \quad (8)$$

Similar equations can be generated for the distance the image covers in the fore and aft directions around the drone. These four equations can now be used to define the four corners of the image as it would be projected onto the plane, with only two minor alterations. Firstly compass orientation, γ of the drone must be taken into account when calculating actual x and y locations for the image. This is done simply taking the entire value calculated above and multiplying it by the cosine of the compass orientation and adding sine times the value of the orthogonal distance. A simplified example of this can be seen in equation (9) below, where x_{right} represents the distance the image includes to the right and y_{fore} represents the on ground distance included in the image forward of the drone, solving for the actual distance relative to the ground, x_{north} .

$$x_{north} = \sin(\gamma) * x_{right} + \cos(\gamma) * y_{fore} \quad (9)$$

Finally, the offset for the GPS location must be incorporated into the calculation. This would be implemented by using an external library to convert the GPS location into a foot distance offset from the first received GPS location. Then this distance offset can be included as an offset in the equation. The two equations (10) and (11) for calculating one corner of the image are shown below. The GPS offset is represented by d_{lat} and d_{long} , and the pitch orientation offset is represented as β .

$$x_1 = \sin(\gamma) * (z * \tan(\theta + \beta)) + \cos(\gamma) * (z * \tan(\phi - \alpha)) + d_{long} \quad (10)$$

$$y_1 = \sin(\gamma) * (z * \tan(\phi + \alpha)) + \cos(\gamma) * (z * \tan(\theta + \beta)) + d_{lat} \quad (11)$$

Once equations necessary to calculate the image corners the creation of the software began. A OpenGL environment was setup and custom shaders were written. These shaders were applied to a quad object which would display the image. However complications in displaying the image on the quad and time constraints forced development on this section to end. An in depth description of the issues faced in continuing this line of development are discussed in the Results, section 5.5.3.

4.3 Summary

Each subsystem came together for the final implementation. Each drone consisted of a real time system responsible for keeping the UAV in the air, while a non-real time system consisting of a Raspberry Pi, PiCam, and WiFi antenna aggregated the drone's pose and current image data into packets. The software responsible for the collection and transmission of the data was written and implemented using Python, and open-source projects MAVProxy and MAVLink in custom-written software. The packets received over the wireless transmission were received on the ground station computer, decoded, and stored for future post-processing.

In addition, different kinds of post-processing schemes were investigated, consisting of the open-source VisualSFM project, Unreal Engine 4, and custom OpenGL development. Each of these investigations had their own strengths and weaknesses.

5 Results

Testing was done over the course of the project for each stage of development. The tests for each component as well as the final test are discussed in this section. However, the nature of the testing vehicles for this project necessitated a further degree of testing. Whenever a drone was created or altered additional testing was always necessary to verify the drone's safe operation.

5.1 Drone Fitness Testing

Each drone was required to go under a set list of tests before being declared flight worthy. A sample checklist for checking flight worthiness on each drone consists of the following:

☐ Firmware and parameter check

Check that each drone is running the same version of the firmware by connecting them to MissionPlanner. Parameters such as controller gains can be checked in this way as well. This ensures that MissionPlanner can communicate over its embedded multi-drone protocol, and will actively receive data over the telemetry link.

☐ Radio calibration

Check the status of each radio channel to ensure the expected response. This confirms that each radio channel is both assigned to the correct functionality, and that the range of each channel corresponds to the correct response. For example, a switch is mapped to loiter instead of autonomous flight, or throttle is mapped to elevator rather than a rudder control.

☐ Compass Calibration

The flight controller's compass is calibrated by rotating the airframe about its' axes. The offsets are saved by the flight controller.

☐ Accelerometer Calibration

The flight controller's accelerometer is calibrated by positioning the airframe sequentially along all 3D axes in both directions.

☐ Check Flight Modes

The flight controller parameters are checked from the computer and the correct flight mode settings are verified.

☐ Battery and battery monitor check

All battery voltages are checked. Any batteries supplying the flight controller and airframe power distribution network are given attached battery monitors to alarm if battery voltage drops below safe levels. This check should happen before every flight.

☐ Prop rotation and tightness check

The correct orientations of propellers on the airframe are verified. The propeller caps are checked for tightness. Caps should be tightened to firm but not overtightened.

☐ Secure all loose wires and electronics

Ensure all wires are secured to the airframe and will not become entangled with any rotors or become a snag risk. Verify electronics connections. This check should happen before every flight.

☐ Current Offset Test

This tests the effects of max current flow on the system. To calibrate this, a diagnostic test is run on the flight controller and the motors on the drone are run up to full throttle. For this test, the drone is strapped down to a fixed surface or rack. Safety goggles should be worn while performing this test in case of catastrophic propeller malfunction.

☐ Check GPS accuracy

Ensure on the ground computer that the drone's flight controller has the proper location information.

☐ Check altimeter accuracy

Ensure the altimeter values are reading correctly.

These prerequisites determine whether the drone is ready to run a preliminary test flight. Running a preliminary test flight should only be done in an area at least 300ft from any people or roadways. When the test flight launches, attention should be paid to how the drone is controlling and flight modes should be verified to be functional. If any problems arise, a landing should be attempted while still at a low altitude.



Figure 39: Drone fitness testing required strapping the system down to simulate full load on all components and sensors for calibration.

With a preliminary test flight complete the drone can be ruled airworthy and ready to undergo other testing, running autonomously or otherwise. The drone should only operate at height and location that allows it to set down in a timely manner in case of battery voltage dropping beneath a safe threshold.

5.2 Wireless Range Testing

One of the most important parts for the entire system was the communications module. One of the original goals of the project included a minimum range requirement for transmitting data images from the drone network to the ground computer. Thus, testing the range of the wireless systems was crucial. Tests needed to be carried out in order to assess the abilities of the dish WiFi antenna as well as the WiFi adapters on board the drones.

5.2.1 Test Design

The test of the wireless communications system was to follow a simple protocol. One team of operators would carry a flight-worthy drone equipped with the Raspberry Pi module and a ground computer (used only for drone telemetry by the operators) to a remote location. Another team would be based at a different location with the dish antenna and ground computer running the imagery collection software. For this test the ground computer team with the dish antenna were based next to WPI in Institute Park. The team launching the drone deployed from an empty football field on a nearby hill to allow line of sight between the drone and antenna. The distance between the locations was approximately one mile.

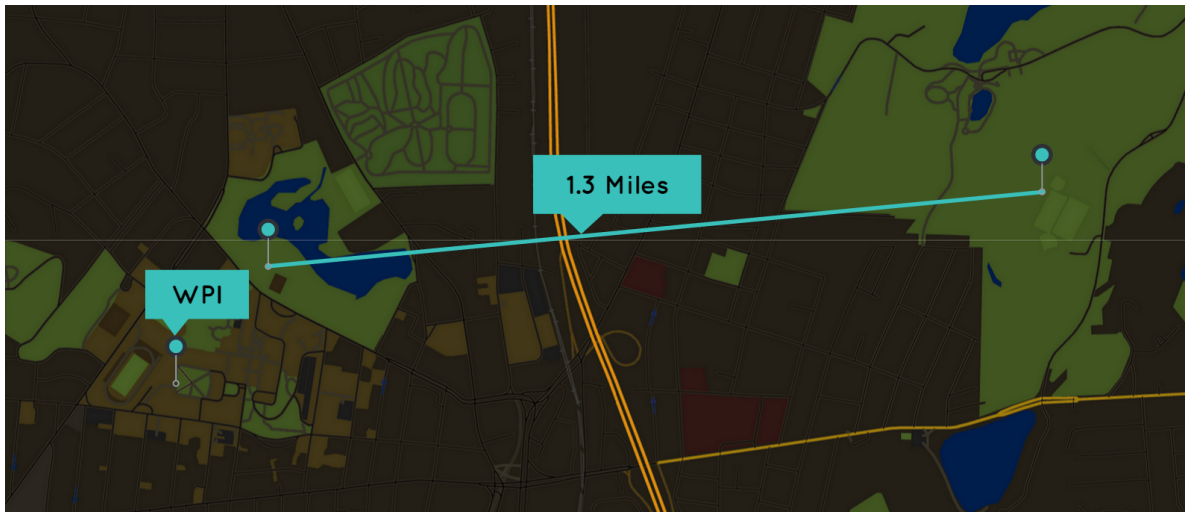


Figure 40: The range test consisted set up a WiFi connection between two points 1.3miles away. The ground station was located at $42^{\circ}16'33.8''\text{N}$ $71^{\circ}48'23.5''\text{W}$ and the drone flew at $42^{\circ}16'44.0''\text{N}$ $71^{\circ}46'53.5''\text{W}$.

5.2.2 Test Results

The first run of the test was a failure due to weather conditions. Though not windy, a frozen rain/hail mix was coming down in light volume. These conditions allowed the drone to be launched by the control team, but impeded the visibility of the drone by the antenna team. Further, precipitation increased shortly after launch from light to moderate. The precipitation greatly reduced signal strength over the test distance, revealing a possible

drawback of the communications system design based on 2.4 GHz WiFi. A connection could not be maintained between the drone and ground computer and the test was aborted.

The test was run again shortly after on a day with significantly improved weather conditions. The same procedures were followed and a reliable connection was established between the drone and ground computer, allowing transmission of data, as well as remote control of the Raspberry pi module.



Figure 41: This picture was transmitted and received over a 1.3mile WiFi link. GPS location $42^{\circ}16'44.0''$ N $71^{\circ}46'53.5''$ W.

5.3 Image Collection testing

5.3.1 Test Design

Testing the image collection software was done more incrementally than the other testing. The software was easily tested without the need for any flights at first. The software could be tested on the ground using a WiFi link in the same room. These tests simply sought to verify that the software module running on the Raspberry Pi was correctly collecting and

buffering images, then sending them to the ground station. Further testing of this software happened during the WiFi range test to verify that it would work during flight.

5.3.2 Test Results

The PiCam captured images as programmed. Data taken from MAVProxy successfully was paired with each image and transmitted from the buffer. Some selected imagery from testing can be seen below.



Figure 42: A few of the many images collected during a flight over the Worcester Polytechnic Institute campus. These were taken from about 60m altitude at $42^{\circ}16'25.4''$ N $71^{\circ}48'33.9''$ W.

Additionally the images collected in this test were run through the 3D point reconstruction pipeline to create the 3D point cloud. This was done using VisualSFM and the resulting construction can be seen in Figure 43 below.

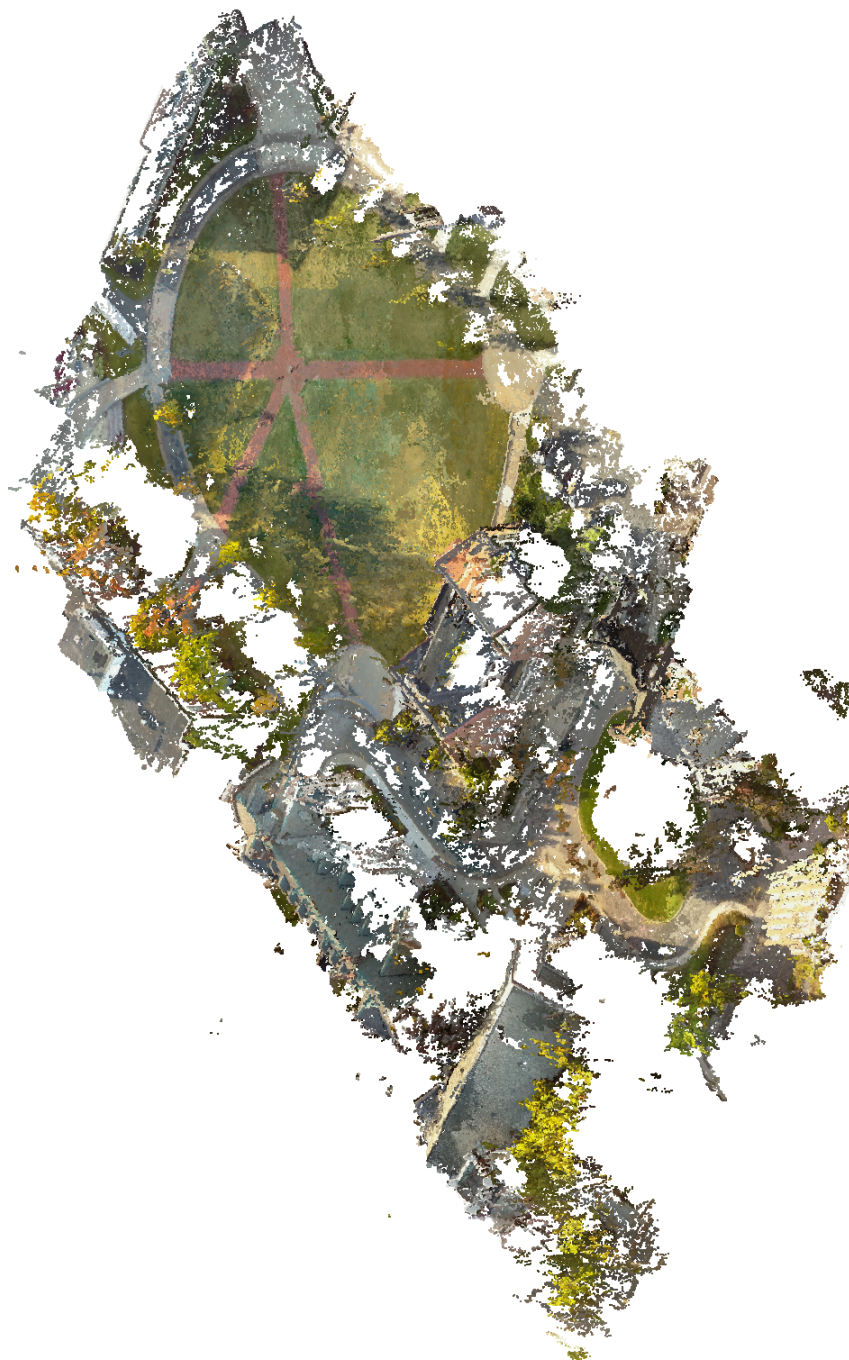


Figure 43: The 3D reconstruction generated from the image collection test flight over WPI campus.

5.4 Full System Test

5.4.1 Test Design

The final test plan was for a trial run of the entire system working in concert. A safe space for flight would be established, i.e., a field without people nearby and roadways sufficiently distant. This test called for flying at least two of the drones in formation across a grid while running the imaging software and connecting over the WiFi dish. Unfortunately, weather constraints delayed other testing enough that there is no longer enough time to plan and run this final test. So while all modules have been individually tested, a final system test has not yet been attempted.

5.4.2 Overall Results

The system as it is should be theoretically be functional. However, it definitely requires more testing. The overall output of the system is an unsorted database of images and associated positional metadata. This is a point of development that easily allows for the implementation of a processing program to take the data and make it usable. The modular nature of everything currently in the system means improving and expanding any part of it is relatively simple.

5.5 Issues

Over the course of the project, issues arose both incidentally during testing and with the overall design of individual subsystems. These delays prevented the project from fully realizing all the planned features. In this section, some of the most important issues faced will be discussed.

5.5.1 Weather

The timing of this project was an inevitable but unfortunate source of setbacks. Most of the work took place between the late fall and winter months. Unprecedented levels of snowfall during these times made flight tests difficult to execute. Often planned flight tests would need to be canceled due to high winds or further precipitation. These conditions severely limited the ability to test ideas on the drones while still developing each subsystem. Only with the transition to Spring during the final stretches of the project time line has the weather reattained any sort of consistency allowing planned flight tests. Even more recently some test have had to be aborted or cancelled to due to rain or wind.



Figure 44: Months of excessive snowfall made it difficult to test the system during the winter terms.

5.5.2 Loss of Assets

Another problem that arises due to the nature of testing this project is the loss of resources due to equipment failure during testing. In testing aerial drones, equipment failure can lead to loss of control. When drones lose control at altitude, an unplanned landing at high speed usually ensues; that is, a crash. These failures are typically safe as tests are conducted away

from unaware civilians and operators are constantly vigilant in positioning themselves safely, but often end with the drone severely damaged.

Since procedure dictates that only drones verified to be airworthy can be flight tested, these incidents have been kept to a minimum. However, unforeseen problems do still occur despite the testing regimen. During one flight in the late stages of testing, imagery was being gathered by a single drone under manual operation. The drone underwent the loss of a single prop at high altitude and crashed. A large amount of hardware was lost. This crash demonstrated that equipment can fail at any time, and reinforced that proper safety procedures and launch guidelines should be followed no matter what.



Figure 45: Mechanical failures resulted in loss of assets.

5.5.3 Image Stitching Setbacks

While creating the modules for image stitching several large set backs were encountered with all of the methods attempted. The first of these was the massive time complexity of structure from motion. During testing it was found that the system is simply not tractable for any sort of system that requires quick results. A single ten minute long flight of campus, resulted in 1500 images. Using VisualSFM on a computer with a Nvidia GTX580m and Intel i7 2720QM the program had to run for 32 hours before producing a result. Although this time

could be decreased by using a cluster of servers to process the images instead of a personal computer, that would still require a large amount of time as well as exterior hardware that would necessitate an Internet connection, which is not always available in environments like search-and-rescue. Additionally the data generated, although visually interesting and somewhat useful for surveying was not high enough resolution to discern individuals, which limits its usefulness.

Creating a 2D map of the environment proved to also be challenging. The first attempt at this was made using the Unity game engine. Although using a projector object and interacting with the game environment itself proved to be easy, projectors had several problems. Projectors work by taking a bitmap and projecting that onto whatever ever game geometry the object is aimed at. When it hit that surface the light from the projector is multiplied with the light from the environment to create the image. Unfortunately this means that if one tries aim multiple projectors at a surface they will only be visible where the two projectors overlap and the image becomes darker with each additional projector added to the environment. This issue can be seen in Figure 46 in which two projects are aimed at the same plane.

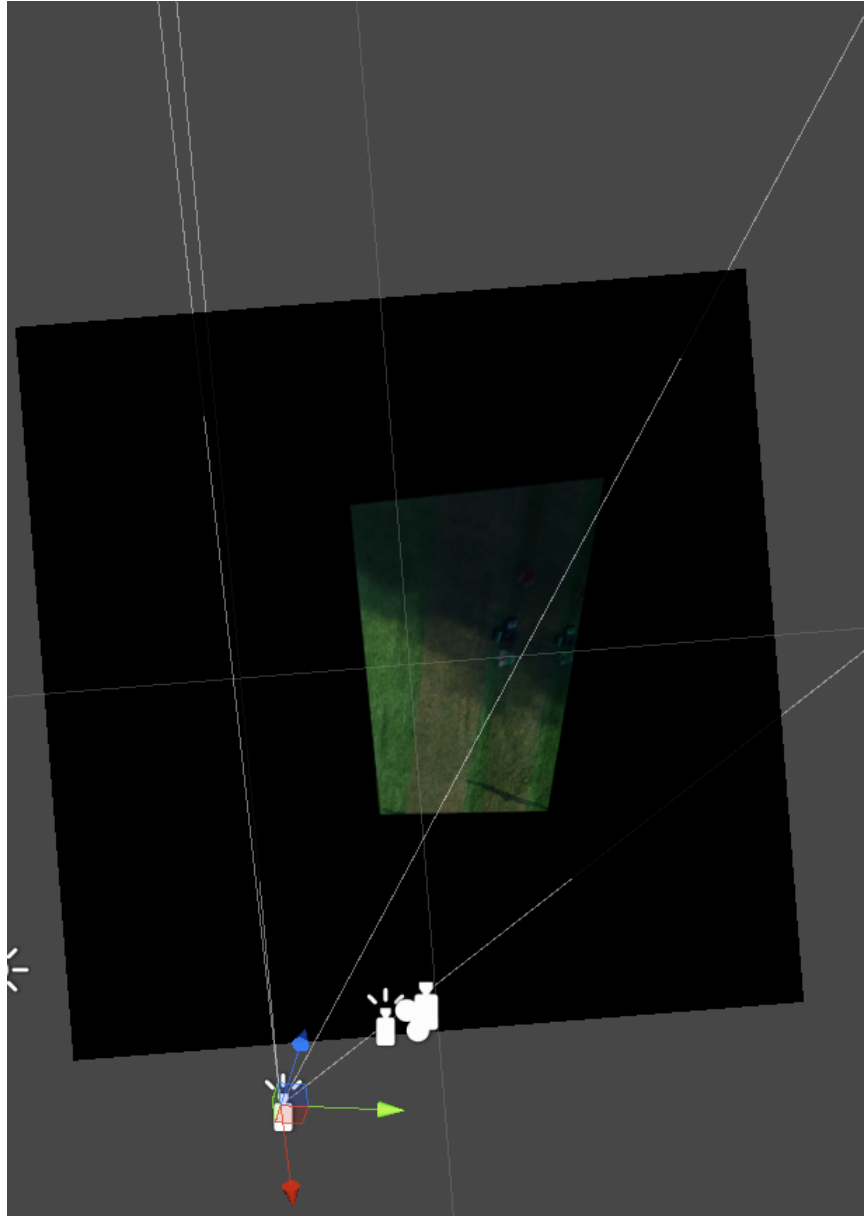


Figure 46: When two projectors are aimed at geometry on Unity the light from the multiplies. This means that the images are only visible where both projectors overlap. The projected images are also much darker than the original photos.

Additionally, projectors also appear to continue the outer pixels of the image being projected over the rest of the geometry. This means that if an image is projected at a plane the entire plane will change to the colors of the outside of the image. The only means of resolving this issue that was found was to place a transparent border around the edge of the image being supplied to unity sufficiently wide for this issue to go away. In order to implement this solution every image coming back from the system would need to be edited automatically to add the border. This would not only increase the complexity of the system, but also add to the size of the images and memory used for blank padding.

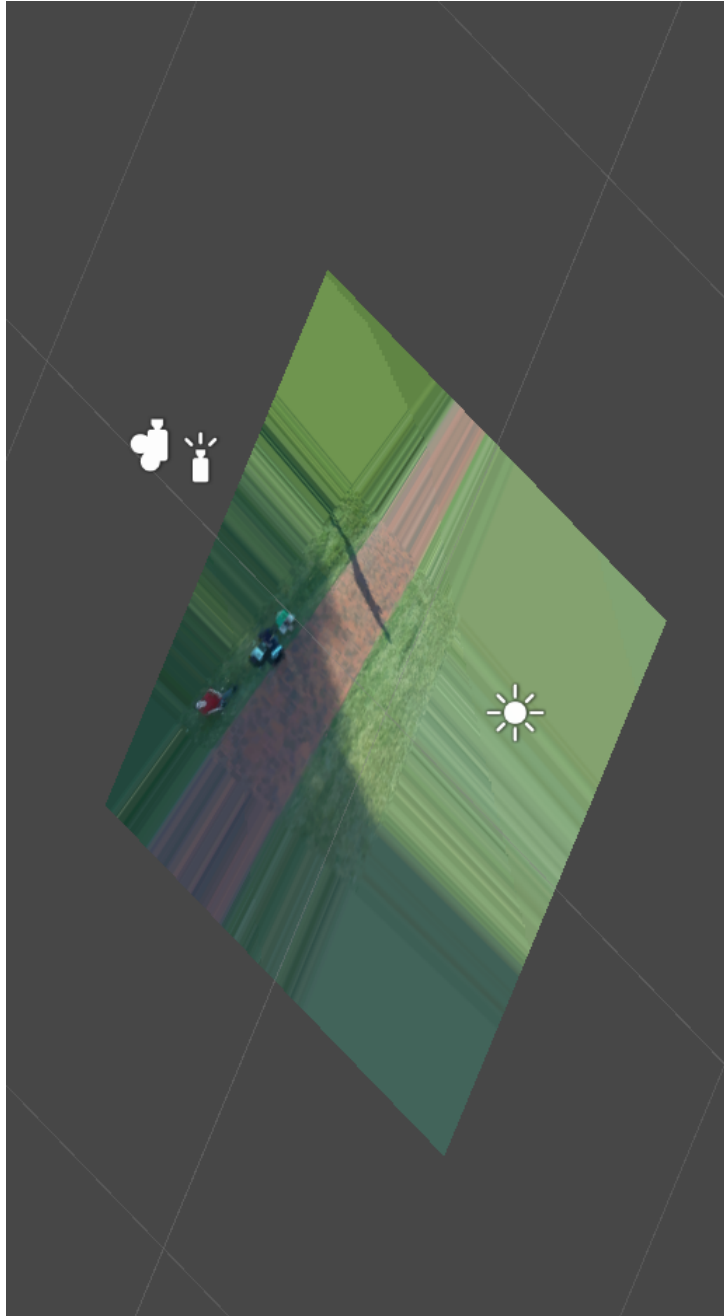


Figure 47: The edge pixels of projected images in Unity continue onto the rest of the geometry being projected on.

The final nail in the Unity coffin was attempting to dynamically load the images as they arrived from the drones. In order for this system to be truly useful images needed to be loaded in real-time as they were downloaded from the drone network. From the standpoint of Unity this meant it needed to dynamically load game assets while it was running. Although this is possible in theory in practice it was found to be unreliable. Assets would take a long time to load and often not load at all. At this point it was determined that using Unity was unfeasible and the transition to OpenGL was made.

There was only one setback discovered in implementing OpenGL. Which creating a quad, or flat figure with four points in OpenGL, the library models it as two connected triangles. Although this makes sense from a graphics pipeline perspective it causes unfortunate stretching of an image if the quad is not drawn as a parallelogram [3]. The image is mapped to each triangle separately causing an obvious seam down the middle, as seen below in Figure 48.



Figure 48: Construction of a quad in OpenGL is done with two separate triangles. This causes a seam to be visible when the quad is re-sized to irregular shapes.

A process for solving this problem was being researched [3], but due to time constraints and previous failures to create image stitching software project priorities were reevaluated and the implementation of image stitching software was canceled.

5.6 Results Summary

The system isn't tested enough to simply brand it as a success or failure, but it's certainly proven itself as a platform for development. Each component of the system is modular in an extremely convenient way. As it is now, the system is capable of limited tasks involving capturing ground imagery across a wide area. With the replacement and expansion of a few modules it could easily serve as a user friendly platform for any of the originally discussed use cases. Testing went well, and when correct procedures were followed, there were no safety incidents during operation of the system. The test results for each module are positive, and

demonstrate a solid platform for further development.

6 Conclusion

The main goal of the project, to construct an expandable drone based data collection system, was achieved, and the team learned a number of useful skills. Over the course of this project, the team learned methods in system design, mechanical design, electrical debugging and software engineering. Each of the skills learned helped to overcome either an expected technical challenge, or an unexpected problem that arose as part of the testing or data collection portion of the project. The project team had to work around a number of unexpected barriers. Developing government regulations and changing drone policy on campus created a turbulent environment for testing these technologies. Yet, the final product functions within the constraints of shifting regulation and still fulfills many of its original goals. The goals that went unmet mainly concern usability and utilization of captured data. The system is functional but not usable in a professional setting. The project managed to achieve most of the functions necessary, but has not provided a convenient interface for users without the necessary technical background to operate each component individually. There also remains a need to process the system output and put it in a useful form for review. These shortcomings provide an opportunity for future teams to continue the work done by this project.

6.1 Future Work

The future work remaining for this project is very exciting, as there are a number of places where even small improvements in performance could open entirely new applications. The place where the project team has left off is a turning point, where any module could be expanded to increase functionality without compromising the overall cohesiveness of the system.

On the hardware side, improvements in airframe efficiency could lead to longer flight times and more complex missions. Better cameras would allow for more detailed scans, or the possibility of hyperspectral data collection by capturing a wider range of spectrum. The

radios are another area where there is a lot of room for improvement. A combined telemetry and data link operating on custom designed hardware could have far longer range or much higher data throughput. Alternatively, the radios could be operated in a mesh network mode allowing for longer range missions by using intermediary airframes as relays. There is also the possibility of ASIC or FPGA based sensor add ons for applications such as stereo depth information that can accelerate the 3D reconstruction process.

A lot of the really interesting software problems remain only tangentially explored. Some of these problems relate to the user interface challenges of coordinating multiple UAVs. Another user interface task is improving the work flow for data capture and management, as well as creating better data visualization tools. There is a lot of work that could be done to integrate with new trends in computing, such as integration with mobile devices such as tablets and smart phones or VR devices such as the Oculus Rift or HTC Re Vive. On the algorithms side, some of the hardest problems in the field remain in finding solutions for the vision processing, obstacle avoidance, and efficient real time 3D mapping.

Regardless of what aspect of the project is chosen as a focus for future work, the system described in this report, or one like it, would make an excellent starting point. It provides a flexible framework which can be expanded by the addition of new software or hardware. This project provided not only a physical platform for development, but a workable set of guideline and directions for safely building and operating a fleet of UAVs for research.

6.2 Final Thoughts

Multiple robotic systems are an existing concept, but haven't been applied extensively in aerial drone research yet. The project didn't manage to test multiple drone flight thoroughly, but it did prove the feasibility of using multiple data streams simultaneously. Multiple drone systems are a field with a lot of potential work left remaining. The field may have to wait for changes in FAA regulation to continue expanding but it clearly presents an exciting opportunity for improving data gathering by aerial drones.

References

- [1] DJI, *Naza-M V2*, Accessed: April 29, 2015, <http://www.dji.com/product/naza-m-v2>
- [2] NBC News, *Drones in Search and Rescue*, Accessed: March 15, 2015, <http://nbcnews.to/1mLNu5e>
- [3] Nathan Reed *Quadrilateral Interpolation, Part 1* Accessed: April 11, 2015, <http://www.reedbeta.com/blog/2012/05/26/quadrilateral-interpolation-part-1/>
- [4] OpenGL Organization, *OpenGL Documentation*, Accessed: March 15, 2015, <https://www.opengl.org/documentation/>
- [5] Unity Technologies, *Projectors in Unity* Accessed: March 10, 2015, <http://docs.unity3d.com/Manual/class-Projector.html>
- [6] Unity Technologies, *Unity Engine 5 Documentation* Accessed: March 10, 2015, <http://docs.unity3d.com/Manual/index.html>
- [7] Changchang Wu, *VisualSFM Open-Source Project Documentation*, Accessed: October 10, 2014, <http://ccwu.me/vsfm/doc.html>
- [8] Luke Runyon, NPR News All Tech Considered, *As Rules Get Sorted Out, Drones May Transform Agriculture Industry* Accessed: February 18, 2015, <http://n.pr/1Eaz8PU>
- [9] Amazon Website, www.amazon.com.
- [10] 3D Robotics, *ArduPilot Autopilot Suite*, Accessed: December 25, 2014, <http://www.ardupilot.com/>
- [11] Eric Rowell *Big-O Cheat Sheet* Accessed: April 20, 2015, <http://www.bigocheatsheet.com/>

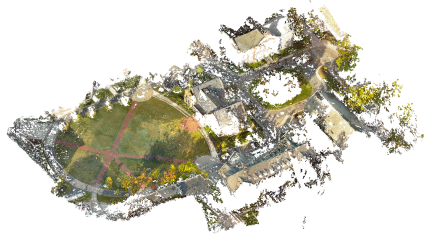
- [12] RC Helicopter Fun Blog, *Parrot AR Drone / 2.0 Quad Rotor Review*, Accessed: April 29, 2015 <http://www.rchelicopterfun.com/parrot-ar-drone.html>
- [13] Gregory S. McNeil, Forbes, *Woman Faces A Year In Jail For Beating Drone Operator, Assault Caught On Video*, Accessed: October 12, 2014, <http://onforb.es/SwZidF>
- [14] DIYDrones Website, www.diydrones.com.
- [15] Brent Rose, *DJI Phantom 2 Vision+ Review: Buttery Smooth Quadcopter Video*, April 6, 2014, Accessed: November 7, 2014, <http://gizmodo.com/dji-phantom-2-vision-review-buttery-smooth-quadcopter-1559567158>.
- [16] CBC News, *City of Richmond bans drones and remote-controlled planes*, Accessed: April 17, 2015, <http://www.cbc.ca/1.3036764>
- [17] DroneDeploy, *DroneDeploy - Simple Fast Drone Software for Business*, Accessed: April 28, 2015, <https://www.dronedeploy.com/>
- [18] i.ytimg.com, *Image Credit for Exploding LiPo*, Accessed: April 28, 2014, <http://i.ytimg.com/vi/bZ8IsMRFM5o/maxresdefault.jpg>.
- [19] Flite Test Website, www.flitetest.com.
- [20] Changchang Wu, Sameer Agarwal, Brian Curless, Steven M. Seitz, *Multicore Bundle Adjustment*, 2011 Accessed: April 29, 2015, <http://grail.cs.washington.edu/projects/mcba/pba.pdf>.
- [21] Hobby King Website, www.hobbyking.com.
- [22] Horizon Hobby Website, www.horizonhobby.com.
- [23] www.headsuphobby.com, *Image Credit for brushless inrunner motor.*, Accessed: April 28, 2014, <http://www.headsuphobby.com/Firepower-450-PLUS-2200kv-Inrunner-Brushless-Motor-H-715.htm>.

- [24] Ajai Sechgal, *Skype Interview*, November 11, 2014.
- [25] Hobby King Website, *Hobbyking KK2.1.5 Multi-rotor LCD Flight Control Board With 6050MPU And Atmel 644PA* Accessed: April 25, 2015, http://www.hobbyking.com/hobbyking/store/__54299__Hobbyking_KK2_1_5_Multi_rotor_LCD_Flight_Control_Board_With_6050MPU_And_Atmel_644PA.html
- [26] MAVLink Open-Source Project, Accessed: October 28, 2014, <http://qgroundcontrol.org/mavlink/start>
- [27] MAVProxy Open-Source Project, Accessed: October 28, 2014, <http://tridge.github.io/MAVProxy/>.
- [28] Mission Planner Open-Source Project, <http://planner.ardupilot.com>.
- [29] i.aliimg.com, *Image Credit for paddle propellers*, Accessed: April 28, 2014, http://i00.i.aliimg.com/wsphoto/v0/2046556362_1/2-Pairs-1755-17X5-5-High-Quality-Carbon-fiber-CW-CCW\ -Propeller-Props-Accessory-for-RC.jpg.
- [30] www.getfpv.com, *Image Credit for pancake style motor*, Accessed: April 28, 2014, <http://www.getfpv.com/nx-4008-620kv-multi-rotor-motor.html>.
- [31] Raspberry Pi Foundation, *Camera Module*, Accessed: February 13, 2015, <https://www.raspberrypi.org/documentation/usage/camera/README.md>
- [32] Raspberry Pi Foundation, *PiCamera Documentation*, Accessed: April 29, 2015, <https://www.raspberrypi.org/documentation/hardware/camera.md>.
- [33] PX4 Pixhawk Website, *The Pixhawk PX4*, Accessed: April 29, 2015, <https://www.pixhawk.org>

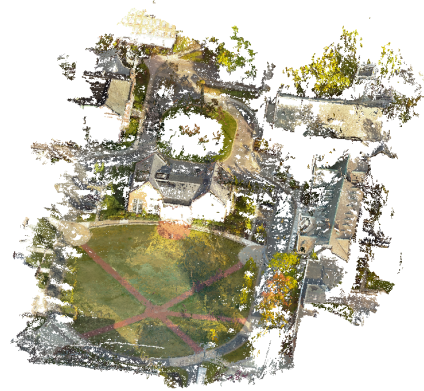
- [34] <http://www.mikrokoetter.de/en/home>, *Image Credit for propeller profile.*, Accessed: April 28, 2014, http://mikrokoetter.altigator.com/images/propeller/Carbon-propeller-15-16x5_5-multirotor.jpg.
- [35] Python Software Foundation, *17.2. socket — Low-level networking interface*, Accessed: November 15, 2014, <https://docs.python.org/2/library/socket.html>
- [36] RC Groups Website, www.rcgroups.com/forums/index.php.
- [37] RC Helicopter Fun Website,
www.rchelicopterfun.com/rc-lipo-batteries.html.
- [38] Ready Made RC Website, www.readymaderc.com.
- [39] RobotShop, *Image Credit for Tarot 680 Pro*, Accessed: April 28, 2014, <http://www.robotshop.com/en/tarot-680-pro-folding-hexacopter-frame.html>.
- [40] senseFly, *senseFly: Drone for Professionals, Mapping & Photogrammetry, Flight Planning & Control Software*, Accessed: April 28, 2015, <https://www.sensefly.com>
- [41] Changchang Wu, *Towards Linear-time Incremental Structure from Motion* 2013, Accessed: April 29, 2015, <http://ccwu.me/vsfm/vsfm.pdf>.
- [42] 3DR *3DR Introduces Solo, the World's First Smart Drone*, April 13, 2015, Accessed: April 29, 2015, <http://3drobotics.com/2015/04/3dr-introduces-solo-worlds-first-smart-drone/>.
- [43] www.horizonhobby.com, *Image Credit for thin electric propeller.*, Accessed: April 28, 2014, http://s7d5.scene7.com/is/image/horizonhobby/APC13065E_a0?wid=1400&hei=778.
- [44] David G. Lowe, *Distinctive Image Features from Scale-Invariant Keypoints*, January 5, 2004, Accessed: April 29, 2015, <http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>.

7 Appendix A

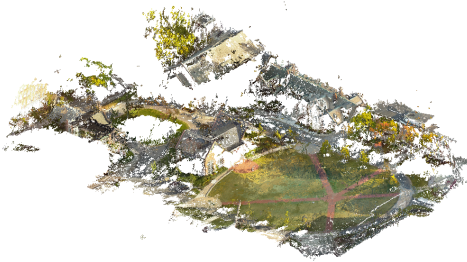
7.1 Additional 3D Reconstructions



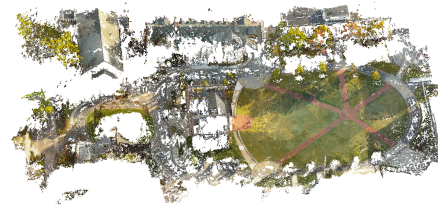
(a)



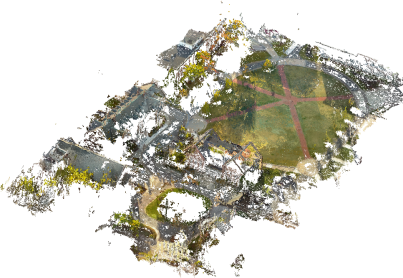
(b)



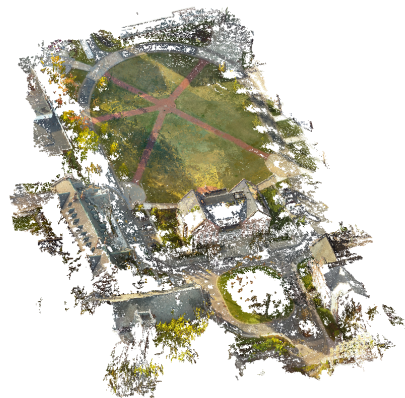
(c)



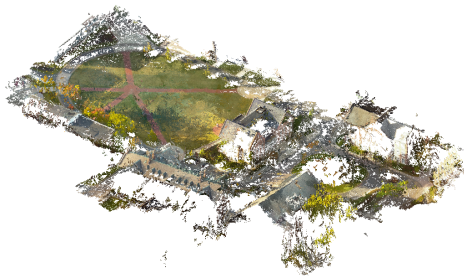
(d)



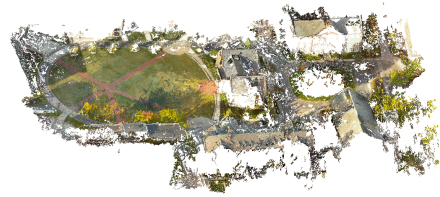
(e)



(f)



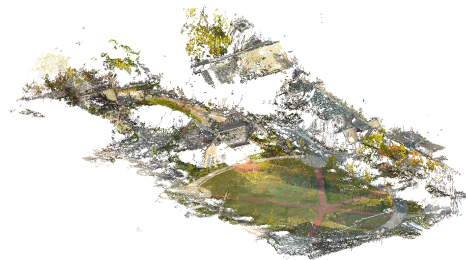
(g)



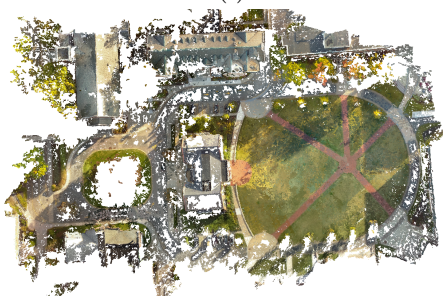
(h)



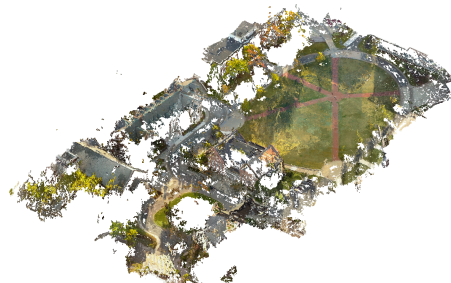
(i)



(j)



(k)



(l)

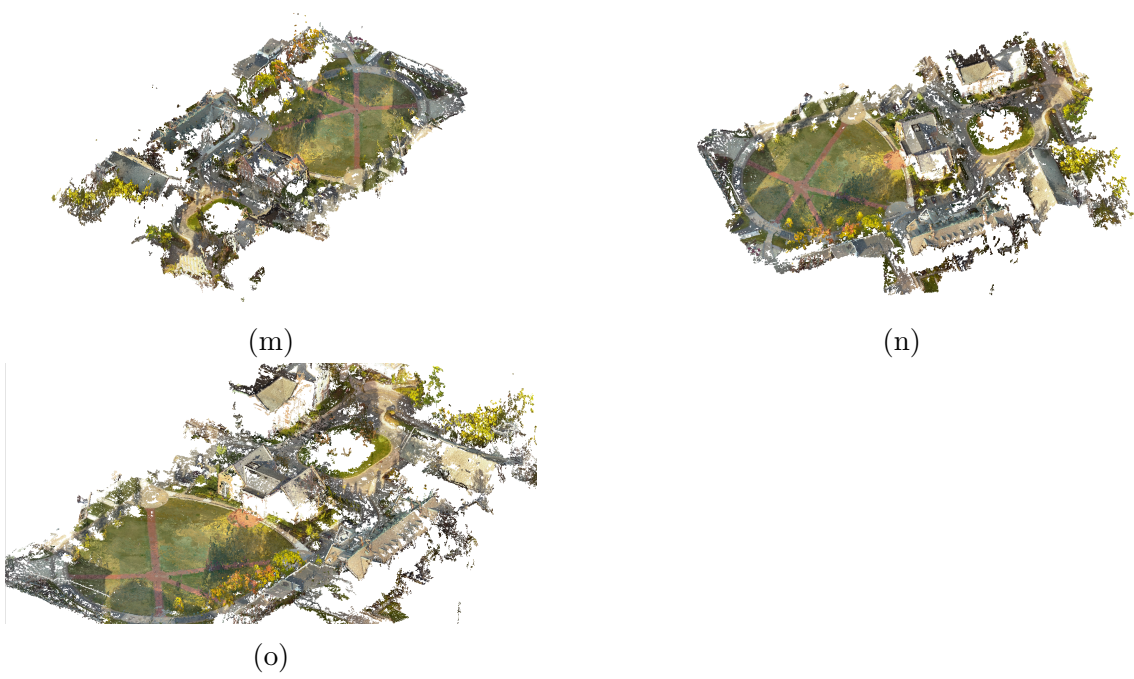


Figure 49: The 3D reconstruction generated from the image collection test flight over WPI campus.