UNMANNED AERIAL VEHICLE PATH PLANNING AND IMAGE PROCESSING FOR ORTHOIMAGERY AND DIGITAL SURFACE MODEL GENERATION

By

Mark W. Bailey

Thesis

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

December, 2012

Nashville, TN

Approved:

Professor Julie A. Adams

Professor Gautam Biswas

ACKNOWLEDGMENTS

I was fortunate to be advised by Dr. Julie A. Adams throughout my work on this thesis. Her incredible patience and understanding with every issue that arose was crucial to the completion of this thesis. I would also like to thank Dr. Gautam Biswas for agreeing to be on my committee and for being incredibly flexible in the weeks leading up to this thesis's defense and final deadline.

Much of the ground work and interface development would not have been possible if not for a fellow graduate student in the Human-Machine Teaming Laboratory at Vanderbilt University, Eli Hooten. His knowledge and expertise frequently sped up the application development process and was an essential resource for completing the work in this thesis. The image processing discussed in this thesis was aided by Dr. Bennett Landman. He allowed us to use one of his powerful servers from the Medical-image Analysis and Statistical Interpretation Laboratory at Vanderbilt University for performing the image processing, which dramatically reduced the amount of time necessary to process our large image sets.

I would like to thank all of my family and friends for being so incredibly supportive throughout the process of writing this thesis. Without their constant moral support and motivation, this thesis would not have been completed by the final deadline.

This research was partially supported by a Vanderbilt University Discovery Grant and by an NSF CAREER Award IIS-0643100.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

Introduction

Unmanned aerial vehicles (UAVs) provide a unique platform that facilitates novel solutions for orthoimagery and digital surface model (DSM) generation. Due to the high costs of low-altitude, manned aerial photogrammetry and mapping and the potentially exorbitant amount of time required to use ground-based manual methods, UAVs are a potential low-cost alternative. High-resolution orthoimagery and DSMs are fundamental aspects of archaeological research that can aid in documentation, excavation and restoration. This thesis describes a new path-planning algorithm for using UAVs to map archaeological sites, which accounts for various parameters that have been considered individually, but have not been combined into a single algorithm.

Current methods for obtaining low-cost, high-resolution imagery for above-ground archaeological sites involve a manual process, such as using kites or balloons with cameras attached that can be quite time consuming and tedious, which diverts resources away from the more important aspects of the archaeological research. Furthermore, maps at different time points during, say an excavation, can potentially provide archaeologists with information that cannot be feasibly obtained for larger sites (i.e., greater than 10 ha.) using the manual methods. There has been some prior research using UAVs to obtain high-quality aerial imagery, but most of the literature focuses on only one or two aspects of doing so optimally—such as taking into account wind speed/direction or sun position, while other parameters are controlled manually. However, a more robust and autonomous system is desirable when the intended users are archaeological researchers and not the UAV specialists.

The primary contributions of this thesis are: the new path-planning algorithm that considers multiple parameters with simulated results showing effectiveness, and an exploration of the efficacy of Agisoft's Photoscan software package (Agisoft, 2012) to generate high-quality orthophotos and DSMs of imagery obtained from a UAV flown manually. The path-planning algorithm takes as input the camera specifications, desired ground sample distance (GSD) and image overlap percentage, the current wind vector and a convex polygonal region that specifies the area to be mapped. The algorithm generates a path of waypoints that the UAV flies to capture photographs of the entire site. The system provides the user with a suggestion for the optimal time of day that will produce images with the smallest shadows for the location in question. Due to technical difficulties, the algorithm was not tested on the UAV at the archaeological site in 2012 (Mawchu Llacta, near the town of Tuti in Peru); however, simulation-based results are provided.

Images of Mawchu Llacta were collected over three manually controlled flights. The images from these

flights have been used to generate orthoimagery and DSMs using Agisoft's Photoscan. Generating high-quality and accurate orthoimagery and DSMs is still somewhat of an art form, thus the presented results are based on exploring the software parameters in order to better understand how to optimize the image-processing pipeline and produce the highest-resolution results.

The relevant background information is presented in Chapter II. Following that literature review, the path planning algorithm and simulated results are presented in Chapter III. Lastly, the various methods used to generate the highest quality orthophotos and DSMs using Photoscan and the imagery captured via manual UAV flights are presented in Chapter IV.

The simulations presented in Chapter III show that considering only the geometry of a rectangular flight region provides the quickest flight paths. There are also fewer path oscillations when a path is generated considering only region geometry, which is desirable to generate higher-quality DSMs and orthophotos from the captured images. When there are more path oscillations, the required image overlap between each image is not necessarily guaranteed. Furthermore, the image processing results presented in Chapter IV show that the best DSMs and orthophotos are generated when the least amount of images are used that still ensure the necessary overlap. Subjective image quality standards were also established that proved to guarantee a higher probability of correct image matching in the absence of GPS/attitude information for every captured image.

CHAPTER II

Background Information

II.1 Mawchu Llacta Archaeological Site

Figure II.1: Mawchu Llacta — archaeological testing site located near the village of Tuti in the Colca valley of Peru.

The broader purpose of the research presented in this thesis is to develop a photogrammetry system that increases the efficiency of archaeological research. There are several sites in Peru being studied by archaeological students and professors at Vanderbilt University, but the focus for this thesis is on the Mawchu Llacta site because of its size and logistical constraints. Mawchu Llacta is a planned colonial town (*reducción*) with several hundred standing structures spread out over an approximately 25 ha. area (or approximately 500m $\times$ 500m) in the highlands of the Andes Mountains. A section of the site can be seen in Figure II.1 and an aerial view of the entire site can be seen in Figure IV.1. Located at approximately 4000m above sea level, the structures were built throughout the Viceroyalty of Peru in the 1570s. The site is inaccessible by ground vehicle, requiring a moderately arduous forty-five to sixty minute hike to reach from the village of Tuti. Archaeological interest in this site is primarily the study of the spatial organization of the 100 urban grid blocks and standing structures. Therefore, it is necessary to have high-resolution and accurate maps and 3D models.

II.2 Brief History of UAVs

Gaspard-Félix Tournachon captured the first aerial image in 1858 from a hot-air balloon of the village of Petit Bicêtre near Paris, France (Newhall, 1969). Since then, aerial imagery has been captured from a wide array of devices including balloons, kites, poles, pigeons, rockets, planes, and more recently, UAVs (see section II.3 for a more detailed explanation of this category) (Batut, 1890; Berlin Correspondent of the Scientific American, 1909; Newhall, 1969; Verhoeven, 2009; Eisenbeiss, 2009; Mozas-Calvache et al., 2012). Other than pigeons, which were used by German engineer Julius Neubronner to capture the path they flew using small cameras mounted to their breasts (Berlin Correspondent of the Scientific American, 1909; Verhoeven, 2009), and rockets, which were used during both world wars for reconnaissance (Newhall, 1969), all of the aforementioned devices have been used with varying success for archaeological documentation and mapping.

The first aerial image of an archaeological site was taken by Giacomo Boni from a balloon (Ceraudo, 2005) in 1899 of the *forum* in Rome (Verhoeven, 2009). However, it was not until around the 1970s that tethered balloons began to surge in popularity for documenting archaeological sites (Whittlesey, 1970). Balloons continue to be used for aerial photography due to their relative ease of use and ability to provide precise control over the orientation of the captured images. Another popular choice for archaeological documentation, kites, can be less stable (Mozas-Calvache et al., 2012).

The use of kites for aerial photography started in 1888 when Arthur Batut captured the first documented aerial photographs from a kite (Batut, 1890; Verhoeven, 2009). They are an obvious choice due to their portability and low cost (Anderson, 1980; Aber et al., 2002). Aber et al. demonstrated the versatility of kites using various film, digital and stereo camera systems in a variety of applications including forest cover, wetlands, property surveys, architectural designs, golf course management, cemetery survey and glacial geomorphology (Aber et al., 2002; Eisenbeiss, 2009). Kites complement balloons in that they are cheaper and easier to use, whereas balloons can be more stable and are less dependent on the presence of wind to function properly.

Although balloons and kites are particularly popular among archaeologists for their cost-effectiveness and ease of use, they both suffer from the inherent issue of needing to be controlled from the ground by up to four people (in the case of balloons) and are limited by the speed at which the human operator(s) can traverse the archaeological terrain. Total Stations (Trimble, 2012) can also be used to survey a grid of geographical points to generate a DSM; however, the resolution of the DSM depends on the spacing between each of the sampled points. Due to the size and difficult-to-travel terrain of our site, Mawchu Llacta, it was estimated by Dr. Steve Wernke (Wernke, 2012) (the archaeologist principal investigator for this project) that using traditional technology, such as a total station, to map the entire site will require approximately three field seasons (i.e., three summers), whereas simulations estimate that it will take an UAV approximately ten

minutes to capture the images. Kites are not a viable option for Mawchu Llacta because of how difficult it is to traverse the terrain, which has significant standing architecture. UAVs also have the advantage of potentially requiring only a single operator and are not physically tethered to that operator. These time and operational advantages that UAVs have over other methods resulted in a sizable subcategory of UAV research known as *UAV Photogrammetry* (Eisenbeiss, 2009).

UAVs were first developed for military applications (Bone and Bolkom, 2004) (which still makes up a significant portion of current UAV research), but recent advances in computing performance, and computer vision techniques have made UAVs a viable option for archaeological photogrammetry as well as other civillian surveillance and mapping applications. Sauerbier et al. (Sauerbier et al., 2011) identified two criteria that must be satisfied for UAVs to be generally accepted by the archaeological community as a viable photogrammetric platform compared to other mapping techniques:

1. UAVs must provide the required accuracy.
2. UAVs must be competitive in terms of economic application compared to other measurement technologies.

Indeed most of the literature focuses on the first criterion (Skarlatos et al., 2004; Eisenbeiss et al., 2005; Eisenbeiss and Zhang, 2006; Furukawa and Ponce, 2010; Remondino et al., 2011; Wallace et al., 2012; Harwin and Lucieer, 2012), while the second appears to remain an open issue (Sauerbier et al., 2011). There are various trade-offs between using UAVs and other competing technologies, such as global satellite navigation system measurement, terrestrial laser scanning and other low-altitude aerial imaging platforms previously mentioned, such as kites and balloons. UAVs are more time-efficient whereas other low-altitude systems can potentially provide more accurate and higher resolution results, but as sensors decrease in size and increase in effectiveness, UAVs will prove to be the system of choice for photogrammetric applications (Eisenbeiss and Zhang, 2006; Lambers et al., 2007; Sauerbier et al., 2011).

II.3 UAV Classification

An UAV in the context of this thesis is a powered aircraft system that is remotely piloted either manually or semi-autonomously by remote control or autonomously through the use of an on-board computer navigation system or a ground control station (GCS) that sends commands wirelessly to the aircraft. Some of the literature uses *UAV* to generally refer to both powered and unpowered, tethered and untethered aerial systems (Eisenbeiss, 2009), but typically, UAVs are powered, remotely piloted aircraft. Various other terms are used in attempt to solve this discrepancy. Remotely operated aircrafts (Verhoeven, 2009), unmanned aircraft systems (Kendoul, 2012) and remotely piloted vehicles (Chiabrando et al., 2011) are all terms used to refer to UAVs.

However, our definition is most similar to that of the American Institute of Aeronautics and Astronautics', which defines a UAV to be:

> An aircraft which is designed or modified not to carry a human pilot and is operated through electronic input initiated by the flight controller or by an on board autonomous flight management control system that does not require flight controller intervention

(American Institude of Aeronautics and Astronautics, 2004).

II.3.1 Types

UAVs fit into two categories: rotorcraft and fixed wing. Due to their vertical takeoff/landing[1] and hover capabilities, rotorcraft UAVs, also known as rotocraft unmanned aircraft systems, seem to be the most popular research platform. Kendoul (Kendoul, 2012) provides an excellent review of the recent advances in guidance, navigation and control as applied to rotocraft UAVs.

(a) Survey Copter's Copter 1B - Single rotor example

(b) RC Coaxial Helicopter - Coaxial example

(c) Microdrone MD4-1000 Quadrotor - Quadrotor example

(d) Oktokopter - Multi-rotor example

Figure II.2: Examples of rotocraft UAVs

Additionally, rotocraft UAVs can be further subdivided into four separate categories (see Figure II.2): single rotors, coaxial, quadrotors and multi-rotors (Eisenbeiss, 2009). Single-rotor rotocraft UAVs have a main rotor on top and a tail rotor for stability. Coaxial rotocraft UAVs have two, counter-rotating blades mounted to the same shaft that allow them to carry heavier payloads and fly at higher altitudes. Quadrotors

[1] Some fixed wing UAVs have vertical takeoff and landing capabilities

are very popular in the research and do-it-yourself/hobbyist communities, because of their smaller size and increased maneuverability. However, quadrotors typically have a smaller payload capacity and are affected more by wind. Multi-rotor rotocraft UAVs are usually designed with either six or eight rotors and are able to carry larger payloads than quadrotors and continue flight unaffected, even if a motor fails due to the multiple-rotor redundancy.

Fixed wing UAVs are unable to fly as close to objects as rotocraft UAVs and are not as agile in maneuverability (Bendea et al., 2007), but a significant advantage is their ability to fly at higher altitudes (Eisenbeiss, 2009). Rotocraft UAVs require the downward thrust generated by their rotors to take off and remain in flight, whereas fixed wing UAVs move forward through the air by means of a propeller (or multiple propellers), or a jet engine and generate lift as the air passes over the wings according to Bernoulli's principle. Another significant advantage is that fixed wing UAVs are able to remain in flight for much longer periods of time without requiring refueling/recharging. That time advantage is especially important for M-class (see Table II.3) electric-powered UAVs that perform missions in remote locations, where access to electricity to recharge the batteries may be unavailable.

II.3.2 Sizes

Depending on the application, there are some reported discrepancies in the literature over how to classify UAVs by their size. For military/tactical applications, Bendea et al. (Bendea et al., 2007, 2008) classifies UAVs by their range, endurance and weight, as shown in Table II.1. Kendoul (Kendoul, 2012) presents five categories based on size and payload capacity for rotocraft UAVs specifically, with examples of each category (see Table II.2). Lastly, Eisenbeiss (Eisenbeiss, 2009) presents a three-level classification that appears to collapse the five categories presented in Table II.2 into two categories (M-Class and L-Class) and adds the OM-Class to distinguish between manual and autonomous systems (see Table II.3).

This thesis focuses on Mini (Table II.2) or M-Class (Table II.3) UAVs, also known as mini/micro aerial vehicles (micro UAVs) (Ceccarelli et al., 2007). However, though most of the presented research is tested on micro UAVs, many of the techniques and algorithms developed will work just as well with larger platforms.

II.3.3 Levels of Autonomy

Standards on the autonomy levels of UAVs are much more clearly presented than aircraft sizes. The National Institute of Standards and Technology has published the Autonomy Levels for Unmanned Systems classification framework, which is generally used to rank UAVs. The specified UAV autonomy levels are categorized based on sensing/perceiving, analyzing, communicating, planning, decision-making and acting/executing capabilities available to achieve a goal specified by the human operator(s) (Huang, 2008). According to this

Tactical UAV Sub-categories	Acronym	Max height [m]	Autonomy [hours]	Weight [Kg]
Micro	μ	250	1	<5
Mini	Mini	150–300	<2	150
Close Range	CR	3000	2–4	150
Short Range	SR	3000	3–6	200
Medium Range	MR	5000	6–10	1250
Medium Range Endurance	MRE	8000	10–18	1250
Low Altitude Deep Penetration	LADP	50-9000	0.5–1	350
Low Altitude Long Endurance	LALE	3000	>24	<30
Medium Altitude Long Endurance	MALE	14000	24–48	1500

Table II.1: Classification of Tactical UAVs (Bendea et al., 2007)

Category	Size	Payload	Example
1	Full Scale	Significant	Piasecki/CMU optionally piloted autonomous helicopter
2	Medium Scale	> 10kg	NASA/army autonomous rotocraft UAS (Yamaha RMAX)
3	Small Scale	2-10kg	CSIRO-ARCAA robotic helicopter
4	Mini	< 2kg	MIT autonomous indoor quadrotor
5	Micro	< 100g	Epson micro-flying robot

Table II.2: RUAS Size Classifications (Kendoul, 2012)

Category	Explanation	Limitation
OM-Class	**O**pen source and **M**annual controlled systems	Manually controlled
M-Class	**M**icro & **M**ini systems	<5kg payload
L-Class	**L**arge payload UAVs	>5kg payload

Table II.3: Categorization with respect to price and payload of UAV systems (Eisenbeiss, 2009)

classification framework, there are five levels of UAV autonomy, where the scale increases in mission and environmental complexity and decreases in the amount of human interaction necessary from level one to level five.

Level one is dedicated to purely remote controlled UAVs. This level of autonomy (or rather lack thereof) requires 100% human interaction and as such, is reserved for those missions that are least complex. Level two systems require a high amount of human interaction, but the missions performed may have more of a purpose than simply flying around. The third level of autonomy defines a moderate level of human interaction with moderate mission and environment complexity. Level four systems have a minimal amount of human interaction and are used for highly complex and/or collaborative missions possibly in a more demanding environment where human reaction times are insufficient. The final level of autonomy describes systems that require absolutely no human interaction. These systems are capable of performing the highest complexity missions in the most extreme environments without any human interaction. Level five systems do not currently exist, so that category serves merely as a goal and an eventual category for future systems .

Even though the Autonomy Levels for Unmanned Systems framework seems to provide enough categories for classifying UAVs by their autonomy levels, Kendoul (Kendoul, 2012) believes the framework is insufficient in providing enough granularity to categorize the existing rotocraft UAV systems and research. Kendoul proposed a new, eleven-level scale called Autonomy Levels for Unmanned Rotocraft Systems as an extension to the existing framework. However, Kendoul argues that the new framework can be applied to UAVs in general (Kendoul, 2012). Kendoul's classification framework also added two additional metrics for determining autonomy: situational awareness and real-time capabilities. Situational awareness measures how capable a UAV is at reacting to the environment and potential environmental changes. The real-time aspect determines if the UAV can reason and plan as a mission is carried out, or does the UAV have to pre-plan or stop in the middle of a mission to re-plan before continuing?

UAV photogrammetry and aerial mapping/surveillance requires researchers to consider whether or not the UAV system needs to plan and reason in real-time to adapt to complex environments, as well as how much human interaction is necessary. Full autonomy (according to the autonomy levels framework) is not necessarily mandatory for all cases. The person operating the UAV for most photogrammetric missions is ideally not the person/team that designed it—it is the police officer deploying the UAV to find a missing person or the soldier using it for surveillance/reconnaissance or the archaeologist launching it to map an archaeological site. Current systems require additional enhancements before they can be operated entirely by these potentially non-technical users. Therefore, a primary concern is how much interaction is needed from the operator, with the autonomy level of the system following based on the answer to that question.

The current state of UAV systems typically allows the operator/pilot to operate in three navigation modes:

manual, semi-automated or assisted, and autonomous (Eisenbeiss and Sauerbier, 2011). The manual mode corresponds exactly to level one of the autonomy levels framework, where all motion of the UAV is controlled remotely by a human pilot with the possibility that system status information (such as battery/fuel levels and link status) is monitored by the pilot or a secondary operator. The semi-automated, or assisted mode, corresponds roughly to level two or three of the autonomy levels framework. This mode typically only requires the pilot to control flight altitude, velocity and heading, while the other complexities of flight stability are controlled by an on-board controller or the GCS. The autonomous flight mode controls all degrees of freedom using an on-board controller or the GCS. This mode corresponds roughly to level three or four of the autonomy levels framework, but not directly to level five due to the complexities of take-off and landing. Take-off is typically completed in manual mode, the system is switched over to autonomous mode to perform the actual mission and then switched back to manual mode to land (Sauerbier and Eisenbeiss, 2010; Remondino et al., 2011; Eisenbeiss and Sauerbier, 2011; Rinaudo et al., 2012). Furthermore, for aerial mapping and photogrammetry missions, the autonomous mode is usually simply navigating a series of pre-defined waypoints and is therefore only autonomous in the sense that it does not require human interaction, but not in the planning/reasoning/adaptability sense.

II.4 UAV Path Planning

This section provides an overview of the state of the art in UAV path planning techniques for various problem domains.

II.4.1 Point to Point Planning

The point to point path planning problem for UAVs involves planning the optimal path to get the UAV from its current location, point A, to a desired location, point B, while accounting for a variety of environmental factors. This problem statement differs slightly from the general point to point planning problem for ground robots in that the added third dimension of altitude is potentially taken into account. The following two sub-sections describe scenarios where point to point algorithms have been developed.

II.4.1.1 Avoiding Threats and Obstacles

One of the main environmental-based considerations is the avoidance of obstacles and threats. Bortoff (Bortoff, 2000) considered the scenario where a UAV attempts to reach a specified destination, while navigating through a field of static, known enemy radar sites. The goal is to reach the destination via an optimal path without being detected by one of the radars. The solution requires a two-step process. The first step performs a graph search on a graph based on Voronoi polygons, where the polygonal “seeds” are the known

locations of the radars. The edges of the polygons are assigned a weight value based on their length and probability of detection by neighboring radars and any of a number of well-known graph-search algorithms can be used to find an optimal path. This solution, however, suffers from dimensionality, but Bortoff proposed an iterative solution of increasing the granularity of the graph at each iteration. The second step attempts to solve the steady-state equilibrium of a Lagrangian mechanical system driven by virtual forces (Bortoff, 2000). Solving that system smooths out the path generated by the graph search in the first step by optimizing the distance of the path from any nearby radar site. The solution can account for radars of either homogeneous or heterogeneous signal strengths.

The downside to Bortoff's solution is that it requires the enemy radar locations to be known in advance and that they also be static. Assuming that enemy radars do not move may not be such an outrageous assumption, but if the problem domain were to be generalized to threats or obstacles, then the potential for the threats to move may be more of an issue. Jun (Jun and D'Andrea, 2003) proposed an alternative solution that accounts for uncertainty in the location of enemy radars. This solution builds a probability map, similar to grid-based occupancy maps (Elfes, 1987), from *a priori* surveillance data where occupancy values for each grid cell are calculated from sensor readings and by applying the conditional probability of occupancy using Bayes' rule (Jun and D'Andrea, 2003). The probability map is converted to a digraph on which a shortest path is calculated and smoothed out and refined based on the minimum turning radius of the UAV. More recently, Fu and Zhu (Fu and Zhu, 2012), attempted to use Korf's Learning Real-Time A* (Korf, 1990) algorithm, combined with a Five-fork Tree node expansion method (Thrun, 1998; Fu and Zhu, 2012), under the assumption that enemy radars or threats are unknown in advance, but can be detected by on-board sensors. Simulated results make this method appear to be effective for re-planning in real-time when a new threat is encountered.

Unfortunately, none of the above solutions account for altitude. De Filippis et al. (De Filippis et al., 2012) used the Theta* algorithm to generate paths that account for known orographic information from DSMs or known object locations in urban environments. Theta* is an extension of the A* algorithm in that it allows paths to be generated in three dimensions. Simulated results were compared to an A* solution and showed that the Theta* approach produced much smoother paths and eliminated useless altitude changes. However, the simulations did not account for vehicle kinematics.

Archaeological photogrammetry likely does not require the consideration of obstacle avoidance, since the UAV typically flies above all of the buildings and structures, but other photogrammetric applications, such as urban search and rescue, may necessitate the implementation of obstacle avoidance algorithms. Furthermore, the Theta* approach may be useful if a lower resolution DSM of the archaeological site is somehow known in advance.

II.4.1.2 Harnessing Wind

Another particularly important environmental factor when planning routes is wind—specifically, harnessing wind to enable longer flights without requiring refueling.

Langelaan and Bramesfeld (Langellan and Bramesfeld, 2008) developed a dynamic model that detects wind gusts from atmospheric turbulence and uses the energy from those gusts to reduce the energy spent by the UAV. Wind gusts typically last for such a short amount of time that human pilots are unable to react fast enough to utilize that extra energy, but the feedback controller can react in real-time. Simulations showed a significant improvement in performance using the developed controller when compared to a controller that did not attempt to react to wind gusts.

Continuing with the theme of using wind to aid long-distance path planning, Chakrabarty and Langelaan (Chakrabarty and Langelaan, 2009) considered the case where the wind field of the area the UAV will be flying through is known (perhaps from a meteorological forecasting tool). The problem can then be seen as that of finding a least-cost energy path through an energy field. The minimum cost path is computed using wavefront expansion under the constraint that node transitions must be made in a direction towards the goal. Using that constraint generated quick, but potentially less-efficient solutions. An A^* approach was used instead of wavefront expansion to calculate optimal solutions, which proved to be a better approach in the resulting simulations (Chakrabarty and Langelaan, 2010).

Using wind for long distance path planning may not be as relevant for mapping smaller areas, but the situation may arise where there are several locations to be mapped, each separated by some distance from each other. The UAV can then utilize wind fields in between each area in an attempt to conserve energy so that it can map as many areas as possible before requiring refueling.

II.4.2 Surveillance

The surveillance problem involves controlling a UAV to view target(s) or target area(s) using an on-board camera, while attempting to optimize image quality and path efficiency, and potentially accounting for environmental issues.

Consider the problem of obtaining imagery of a set of stationary ground targets, all with varying levels of occlusion. At a certain altitude h, each target has a polygonal region in which the UAV is within some distance d of the target and can view it without any obstruction. The goal is then to find the minimum-distance path to fly, such that the UAV can view every target. That problem sounds like a variation of the traveling salesman problem, which is exactly how it was modeled by Obermeyer et al. (Obermeyer et al., 2012). This problem is called the Polygon-Visiting Dubins Traveling Salesman Problem. Obermeyer et al. (Obermeyer et al., 2012) present two algorithmic solutions based on sampling-based roadmap methods. A

slight variation of this problem deals with a set of target areas, each with their own probability of containing a desired target where the goal is to navigate the UAV in a path that finds the desired target in the shortest time possible. This new problem becomes applicable to search and rescue and was considered for the wilderness search and rescue problem (Lin and Goodrich, 2009). The problem is modeled as a discretized combinatorial optimization problem and presented algorithmic solutions based on Local Hill Climbing and Evolutionary Algorithms, which generated descent approximate solutions.

Wind is also a factor for surveillance and requires different considerations than point to point planning. Due to the small size of micro UAVs, wind is generally a factor that needs to be considered when generating path plans, especially if that path requires the micro UAV to capture images of ground targets—if wind is not accounted for, the ground target may not be within the field of view (FOV) of the on-board camera. Ceccarelli et al. (Ceccarelli et al., 2007) considered the problem of obtaining imagery of a set of known ground targets, but with a preferred azimuthal viewing angle and in the presence of wind. The on-board camera is fixed, so the azimuthal viewing angle is controlled by selecting a waypoint in which the line of site of the camera from the UAV to the target produces the desired azimuthal angle. Relaxing the assumption that the ground target is stationary into an assumption that the ground target is moving in the direction and speed opposite of the current wind vector allows the controller to compensate for wind. If the planner assumes the ground target moves opposite the wind vector, while the ground target actually remains stationary, the controller forces the UAV to converge on the necessary location to obtain the desired imagery.

When observing a stationary ground target from an angle other than the nadir angle (the angle pointing in the direction of the force of gravity), there is the potential that sun glare can overexpose and ruin imagery (Stolle and Rysdyk, 2003). Therefore, Stolle and Rysdyk (Stolle and Rysdyk, 2003) developed a controller for fixed-wing UAVs with a nose-mounted camera that can pan and tilt, and considered wind as well as the position of the sun as it maintained video surveillance of a stationary ground target.

If however, the ground target being observed is rather large (such as a bridge), it is possible that peculiar wind patterns (such as the Venturi effect or Karman vortexes) occur that affect the UAVs path (Guerrero and Bestaoui, 2012). Guerrero and Bestaoui (Guerrero and Bestaoui, 2012) developed a solution to this problem by solving a Zermelo-Travelling Salesman Problem. The problem Guerrero and Bastaoui were concerned with was inspecting a structure (such as a bridge) for damage, weak spots, etc. in an optimal manner, while factoring in wind effects. Meshing techniques are used to automatically generate the minimum number of inspection points necessary to inspect the entire structure and then solve the Zermelo-TSP over those points to find the optimal route. The solution accounts for general wind fields, but does not specifically address alternative wind patterns (such as the Venturi effect or Karman vortexes) or the potential for sun glare ruining the images. This solution also requires the geometry of the structure to be known in advance.

UAV photogrammetry is not necessarily confined to capturing images from a nadir viewing angle. The situation may arise where an archaeologist wants a high-resolution 3D model of a particular building or structure. Simply flying over the structure may not capture all of the geometric intricacies. The UAV needs to generate a flight plan that allows it to photograph all sides of the structure, while accounting for the position of the sun. The discussed surveillance algorithms can be used in this scenario.

II.4.3 Simultaneous Localization And Mapping

Simultaneous Localization And Mapping (SLAM) for UAVs is an interesting research area because it has the potential to solve many of the issues related to achieving level five autonomy (see Section II.3.3). Operating autonomously requires UAVs to know their altitude, attitude and position relative to the ground with high precision, as well as the location of objects relative to itself in all three dimensions. Various sensors have been tested to estimate these values, but many either do not provide high enough accuracy or are too heavy of payloads for micro UAVs. Recent approaches to this problem have, therefore, been primarily vision-based.

Hrabar (Hrabar, 2012) tackled the problem of avoiding obstacles and compared results using both stereo imagery and a laser range finder. He designed a reactive system that builds a 3D occupancy map (Moravec and Elfes, 1985) from a stereo camera (Videre-Design STH-DCSG-STOC-C) and a Hokuyo UTM-30LX laser range finder. Simulated and real-world experiments focused on avoiding larger obstacles, such as trees, and showed that the laser range finder produced better performance with an 84% success rate, whereas the stereo camera only had a 42% success rate. The laser range finder has a limited 30m range, so it is suitable either for slower-flying UAVs that can react fast enough to avoid the detected obstacle, or for take-off and landing to estimate the distance to the ground. Stereo cameras are better for detecting the UAVs distance to the ground at higher altitudes, but the accuracy decreases the higher the UAV flies because of the decrease in image resolution (Eynard et al., 2012). Camera systems are also typically lighter than laser range finders and are more suitable for micro UAVs that have a smaller payload limit.

Eynard et al. (Eynard et al., 2012), used a plane-sweeping approach (Collins, 1996; Gallup et al., 2007) to estimate altitude. This approach works better for fixed-wing UAVs as it can be used both in-flight as well as during take-off and landing. Plane-sweeping also works well for rotocraft UAVs, but since take-off and landing is performed vertically, it may be better to switch over to using a short-range laser range finder instead. Furthermore, the plane-sweeping algorithms cannot produce accurate results in outdoor environments where uniform textures, such as grass, are present (Eynard et al., 2012). Eynard et al. also provide methods for estimating attitude and motion using stereo vision techniques, thus attempting to eliminate the need for any sensors other than a camera for autonomous flight. Inertial measurement units are great for providing attitude estimates, but when used in conjunction with vision-based estimates, overall accuracy can be improved (Meier

et al., 2012).

GPS sensors also have their pros and cons. GPS has an accuracy of less than fifteen meters in the latitude and longitude directions, but between twenty-five and fifty meter accuracy in the vertical (altitude) direction (Eynard et al., 2012). GPS provides descent latitude/longitude positional accuracy when sub-meter resolution is not necessary or if being used with larger UAVs. However, GPS sensors can only be used in outdoor environments. Weiss et al. (Weiss et al., 2011) adapted Klein and Murray's (Klein and Murray, 2007) monocular visual SLAM algorithm to increase real-time performance, but unfortunately, their implementation and micro UAV platform has to stop and hover periodically as it performs certain calculations. Hardware specifications and algorithmic enhancements in the near future may allow these calculations to be performed in real-time. Furthermore, the poor resolution GPS provides in the altitude direction effectively renders it useless for take-off and landing. As mentioned previously, laser range finders can be used for this task, but they are still too heavy for many micro UAVs. Therefore, vision techniques (Saripalli et al., 2002; Mejias et al., 2006), as well as hybrid automaton (Cabecinhas et al., 2012) have been explored for take-off and landing. The Skate UAV has semi-autonomous landing, in that the operator sends a command to land and the system controls the landing. However, all of the landing parameters are pre-programmed into the system and do not adapt to different environments, such as the higher altitudes at Mawchu Llacta.

II.5 Photogrammetric Applications

II.5.1 Overview

Photogrammetry is the field of determining geometric properties from imagery. Therefore, UAV photogrammetry can be seen as the process of obtaining imagery from UAVs and processing those images to derive 3D data such as DSMs, digital terrain models and/or orthophotos. Micro UAVs can obtain aerial imagery for a variety of applications, such as forestry, agriculture, archaeology/cultural heritage, environmental surveying, traffic monitoring and 3D reconstruction (Remondino et al., 2011). The various UAV platforms that have been used for photogrammetry, the sensor(s) used and what application(s) they were used for are listed in Table II.4. It is clear that rotocraft UAVs are the predominate platform for photogrammetric applications. Although the techniques used in UAV photogrammetry can be applied to most of the listed applications, this research focuses on the application of archaeological photogrammetry.

II.5.2 Archaeological Photogrammetry

Orthophotos and 3D site renderings can aid archaeologists in documentation, recording, excavation and restoration (Baratin et al., 2000; Dorffner et al., 2000; Skarlatos et al., 2004). Satellite imagery (such as that which can be obtained from Google Earth™ (Google, 2012)) is insufficient in resolution to be of any

UAV	Type	Size	Sensor	Payload Capacity	Flight Time	Forward Overlap	Side Overlap	Software	Application	Source
Copter 1B	Rotocraft	2m	Laser Scanner / Nikon D2Xs	5 kg	30 min.	75%	75%	Custom, LPS	Archaeological & Rockslide Documentation	(Eisenbeiss and Zhang, 2006; Eisenbeiss, 2008; Sauerbier and Eisenbeiss, 2010)
Falcon 8	Rotocraft	?	Camera	650g	20 min.	75%	75%	LPS	General Photogrammetry	(Eisenbeiss and Sauerbier, 2011)
HexaKopter	Rotocraft	?	Sony Nex 5	1kg	36 min.	75%	30%	Custom	Archaeological Documentation	(Rinaudo et al., 2012)
ITHACA Pelican	Fixed-wing	?	Ricoh GR	?	1 hour	60%	20%	Custom, LPS	Archaeological Documentation	(Bendea et al., 2007, 2008; Chiabrando et al., 2011)
LTBT-Eagle	Fixed-wing	?	Canon 5D Mark II	?	2 hours	75%	55%	LPS	Mine Mapping	(Liu et al., 2012)
Microdrone MD4-1000	Rotocraft	?	Olympus EP-1	?	88 min.	70%	20%	LPS	General Photogrammetry	(Sauerbier et al., 2011)
OktoKopter AD-8	Rotocraft	950g	LiDAR / Canon IXUS 100 IS	2kg	3-5 min.	?	?	Custom, Photoscan, Arc3D	Forrest Inventory / General Photogrammetry	(Wallace et al., 2012; Nietzel and Klonowski, 2011)
Skate Gen. 2	Fixed-wing	907g	Go Pro Hero 2	227g	20-40 min.	60%	60%	Photoscan	Archaeological Documentation	This Thesis
TerraLum	Rotocraft	?	Canon 550D	?	6 min.	70-95%	70-95%	Custom, PMVS2	Coastal Monitoring	(Harwin and Lucieer, 2012)
weControl Heli	Rotocraft	30kg	Canon D60	10kg	90min	75%	75%	Custom, LPS	Archaeological Documentation	(Eisenbeiss et al., 2005)

Table II.4: UAVs used in 3D aerial mapping. A '?' means that information was not found in either the source paper(s) or the product website. Flight time represents how long the UAV can fly with the specified payload before needing to refuel/recharge its batteries. Forward overlap represents how much overlap there was in images captured in the direction of flight. Side overlap represents how much image overlap there was in the direction perpendicular to the direction of flight.

use to archaeologists other than as a general reference. That is why UAVs have been used for quick, high-resolution, low altitude aerial photography. 3D information can be extracted when there is enough overlap between the captured images. There is some research that looks into using laser scanners to generate high-resolution DSMs (Eisenbeiss and Zhang, 2006; Lambers et al., 2007; Wallace et al., 2012; Scherer et al., 2012), but laser scanners are still too heavy for many micro UAVs and orthoimagery is still needed in order to generate a 3D recreation of the archaeological site.

Archaeological photogrammetry is a two-step process—step one involves capturing the aerial imagery and step two involves processing those images to obtain the orthophotos and DSMs. Step one involves the path planning as well as how the images are actually captured. The entire process, from path planning to image processing is summarized nicely in (Nietzel and Klonowski, 2011), but the following sections (Sections II.5.2.1, II.5.2.2 and II.5.2.3) break down each step and discuss the current methodologies.

II.5.2.1 Path Planning

As mentioned in Section II.3.3, it is not necessary for UAVs to have level five autonomy for archaeological photogrammetry. It is perfectly sufficient to pre-plan the waypoints that the UAV will fly sequentially to cover the area it will be photographed. However, for these photogrammetric methods to be applied to more complex tasks in the future (such as mapping disaster areas for first responders in urban enviornments), more advanced path planning techniques (such as the algorithms described in Section II.4.1.1) will need to be incorporated into the overall system. Furthermore, there does not appear to be any photogrammetric systems in existence that handle take-off and landing autonomously. Autonomous take-off and landing is a crucial component in making these systems usable by the intended users (e.g., archaeologists, first responders, search and rescue workers, etc.).

Most of the literature focusing on archaeological photogrammetry presents results of flights that were either flown manually or where the waypoints were pre-calculated in a manner known as 2.5D path planning (Eisenbeiss, 2009). 2.5D path planning calculates unique latitude and longitude coordinates for each waypoint, but each point is specified at the same altitude. 2.5D planning makes sense for capturing nadir imagery, because it maintains a uniform ground sample distance (GSD)[2] between all of the images, assuming the terrain is flat. Using 2.5D planning for mountainous or sloped areas may cause a large variance in GSD between all of the images. The only research that appears to address sloped terrain is (Eisenbeiss, 2008). The solution uses a known topographic map of the mountainside being mapped to determine the altitude of each flight line as the UAV descends down the mountain.

[2]Ground sample distance represents the number of centimeters or meters captured per pixel specified by the edge distance of the square area of land covered by a single pixel in an image.

The archaeological sites being mapped either are, or can be defined as convex polygonal regions. The most efficient way to fill a convex polygon with rectangular regions (i.e., images) is to fly over the area in a back-and-forth motion where each flight line is parallel to the last—the two main parameters being the altitude of each flight line, uniform for all flight lines in the 2.5D case, and the distance between each flight line. Those parameters are calculated based on the image scale, camera parameters, maximum flying heght of the UAV, dimensions of the site, desired GSD and necessary image overlap (Eisenbeiss and Zhang, 2006; Eisenbeiss, 2009; Liu et al., 2012). The majority of the literature addresses the planning process in usually one sentence such as: waypoints were calculated manually before the flight (Eisenbeiss et al., 2005; Rinaudo et al., 2012). Eisenbeiss (Eisenbeiss, 2009) developed a 2/2.5D planning tool, but it is described as taking in the necessary parameters and then outputing a file specifying all of the waypoints using in-house software. No further details are provided. Nietzel and Klonowski (Nietzel and Klonowski, 2011) describe their flight planning tool a bit more thoroughly, but with certain aspects still being left up to the imagination. However, Nietzel and Klonowski's program determines waypoint locations, but requires flying altitude to be specified manually and the employed autopilot, the MikroKopter autopilot, can only manage 12 waypoints, with more waypoints having to be split over several files.

II.5.2.2 Image Acquisition

Where the images are captured, how much overlap there is from one image to the next as well as the altitude above ground at which images are captured, determine how high a resolution the end results will be and potentially how much time the images will take to process. There are some standards that have been defined specifying requirements for different photogrammetric parameters in aerial projects (ie., DIN 18740-1:2001-11 and DIN 18740-4:2007) (Mozas-Calvache et al., 2012), but there is still a wide variance of parameters used in the literature.

The desired GSD, the parameter used in conjunction with the camera specifications that determines the altitude at which to fly above ground, is a matter of preference for the specific application. The average GSD range is around 1-4cm (Bendea et al., 2007; Sauerbier and Eisenbeiss, 2010; Harwin and Lucieer, 2012), but can be up to 10cm (Eisenbeiss and Zhang, 2006) or even 2m (Liu et al., 2012).

There also seems to be some discrepancy as to what image overlap percentage produces the best results. Some of the varying percentages used in the literature are shown in Table II.4. All of the experiments from the literature seem to produce high-quality DSMs and orthophotos, so it is therefore most likely that the necessary image overlap is determined by which software is used to perform the image matching and point-cloud generation (see Section II.5.2.3) as well as how much contrast in lighting and texture there is in each image.

Lastly, there are various methods used to actually capture the images. Rotocraft UAVs have the option of either stopping and hovering at each pre-defined image-capture location, or to fly through the point and capture the image in-flight. The former option will force longer flight times, but can potentially produce images with less motion blur, whereas the latter option will shorten flight times, but can introduce motion blur in the images. When using the Falcon 8 from Ascending Technologies, Eisenbeiss and Sauerbier (Eisenbeiss and Sauerbier, 2011) found that capturing images in the stop mode produced less stable flights than when capturing images while flying through the waypoints. However, fixed-wing UAVs do not have the option to stop and hover for capturing imagery. There is also the issue of how the shutter is actually triggered. Systems that can stop and hover to capture an image usually also have the ability to trigger the shutter remotely from the GCS, but not all systems have this capability. When the shutter cannot be triggered remotely, the camera is placed in an intervolameter mode where it takes images every x seconds. Necessary overlap is then ensured from a combination of varying x and the flight velocity.

II.5.2.3 Image Processing

Although some researchers use in-house developed software for processing the captured images, the Leica Photogrammetry Suite (Intergraph, 2012) and Agisoft's Photoscan (Agisoft, 2012) are the two most popular commercial software packages for DSM and orthophoto generation (see Table II.4). LPS and Photoscan are each full-featured applications, but this thesis uses Photoscan because it does not require GPS and attitude information for each image is required by LPS. Having GPS and attitude information does speed up the processing time in Photoscan, but it is not necessary, and the UAV system used in this thesis is unable to record that information for each image.

DSM generation is a three-step process. First, all of the images are matched up and stitched together using a matching algorithm. The reason LPS requires GPS and attitude information for each image is to allow the matching algorithm to narrow down the images to stitch together to only those that actually do have overlapping information. That matching reduction process dramatically speeds up and increases the accuracy of the image matching process. Since Photoscan does not require GPS and attitude information for each image, it attempts to match each image to all of the other images, which can be a lengthy process when processing several hundred images. Once the image mosaic has been generated, it can be exported to a geographic information system software package, such as ArcGIS (ESRI, 2012), to be orthorectified[3].

The second step generates a 3D point cloud using a multi-view stereopsis algorithm (Furukawa and Ponce, 2010). Microsoft Photosynth (Microsoft®, 2012) is a popular and free application that demonstrates how

[3]Orthorectification requires ground control points to be setup on the ground using either a total station or digital GPS prior to the aerial imagery being captured. Those known points are identified in the image mosaic and ArcGIS rotates/translates/skews the image to match up those points geographically.

point cloud generation works. LPS and Photoscan generate the point cloud while matching the images, since both processes require the knowledge of shared points between images. The resolution of the images and the accuracy of the multi-view stereopsis algorithm determine the density of points in the point cloud and ultimately the resolution of the final DSM.

The final step is to build a 3D model around the point cloud and fit the image mosaic to that model if desired. Depending on the density of the point cloud, this step is fairly straight forward—connect the points in the point cloud with polygon surfaces and then decimate the polygon mesh by removing redundant polygons to fit the specified resolution parameters. The resolution of the 3D model is only limited by the point cloud density and the amount of polygon faces the graphics card can feasibly render.

II.6 Conclusion

Due to their small size and maneuverability, micro UAVs have a large potential for photogrammetric applications. Systems have been developed that produce descent results, but are lacking in robustness and are only really usable by trained pilots or the developers of the system. The research in this thesis attempts to bring together the state-of-the-art in UAV photogrammetry research to produce a more usable and accurate system.

UAV type, size and autonomy classifications were presented in Section II.3 with various uses and benefits of each classification. This thesis is concerned primarily with micro UAVs. Section II.4 presented an overview of the state-of-the-art in UAV path planning techniques applicable to photogrammetric applications. If a more robust system is desired that can perform flights for a wider range of photogrammetric applications, autonomous take-off and landing algorithms, such as those presented in Section II.4.3 need to be incorporated in the planning system. Furthermore, obstacle avoidance algorithms for either known obstacles (Section II.4.1.1) or detected obstacles in flight (Section II.4.3) and algorithms optimizing the flight path that take into account wind (Sections II.4.1.2 and II.4.2) and the locations of disjunct regions (Section II.4.2) all need to be incorporated into a unified system. The ability of a micro UAV to perform those operations is, however, limited by the payload capacity and processing power of the on-board computer.

The research presented in this thesis focuses on using micro UAVs for archaeological photogrammetry, which does not necessarily require the robustness of the ideal system just described, but the path planning algorithm presented can be applied to mapping for more domains than just archaeological sites. The full image processing pipeline involved in generating DSMs and orthophotos was described in detail in Section II.5.2. UAV photogrammetry research has made great advances in individual aspects of autonomous and robust systems, but before these systems can be placed in the hands of the users they are intended for, those advances need to be brought together in a unified system.

CHAPTER III

System Description

III.1 Introduction

This chapter describes a path planning algorithm developed for guiding a UAV to capture images over a pre-defined area for generating DSMs and orthophotos. The objective of the algorithm is to find an optimal, minimum-path that allows a UAV to photograph the entire pre-defined area with the necessary overlap between images to facilitate DSM and orthophoto generation. The current version of the algorithm allows a user to choose whether to optimize the path based on area geometry, wind speed and direction or both. Furthermore, the system provides the user with the time-of-day when the sun is at the highest point of its trajectory, and therefore, when shadows are shortest.

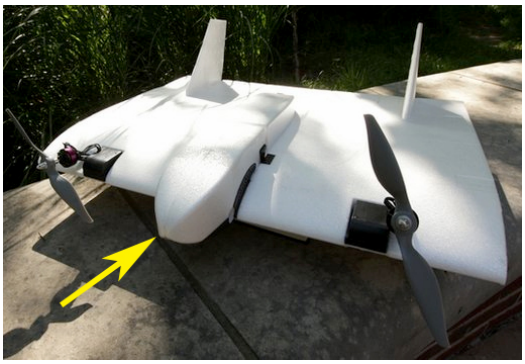
Due to technical difficulties with the Skate UAV system (see Section III.2), the developed algorithm went untested in a real-world application in 2012. Instead, the algorithm's performance was simulated. The results of those simulations are presented in Section III.4.

III.2 UAV Platform

Mawchu Llacta is located approximately 4000m above sea level and at least a forty-five minute hike from the nearest town, necessitating a UAV platform that is light weight, portable and can fly at that altitude. Those restrictions mean that most commercially-available rotocraft UAVs other researchers have used for their testing cannot be used. Instead, this project used the Skate fixed-wing UAV developed by Aurora Flight Sciences (Aurora, 2012), as shown in Figure III.1a. The Skate was engineered to be able to fly comfortably up to 13,000 feet (or 3962.4m) above sea level. The Skate's built in auto-pilot is based on the Paparazzi open-source project (Paparazzi, 2012). A flight plan is flashed to the UAV before flight and a ground control station (a Toughbook running the Paparazzi software) can monitor its sensors from the ground as well as alter waypoints in-flight. The Skate UAV was used with six-cell, 2600mWh batteries that can last approximately twenty minutes of flight time on a single charge. The Skate does have vertical take-off and landing capabilities, but the company does not recommend launching or landing it vertically other than at sea level.

The Skate is launched by hand, as shown in Figure III.1b and can fly in three different flight modes. Manual mode allows the operator to control every aspect of the UAV's flight including velocity, yaw, pitch and roll. Auto1 mode allows the Skate to fly semi-autonomously with the operator only needing to control pitch (at limited increments) and left or right turning (at limited increments). Auto2 mode allows the controller to specify higher-level instructions, such as follow a path of waypoints, and the autopilot controls every aspect of

the flight parameters to achieve to desired action. The Skate has an auto-landing feature that when activated, controls the UAV to fly in a downward spiral until it lands.



(a) The Skate 2 from Aurora Flight Sciences. The yellow arrow points to the payload pod.

(b) Launching the Skate.

(c) The Skate flying over Mawchu Llacta.

Figure III.1: The Skate 2 from Aurora Flight Sciences.

The camera used to capture images from the Skate was an HD Hero2 GoPro with an 11MP sensor and adjustable FOV setting at either 90° (narrow FOV), 127° (medium FOV) or 170° (wide FOV). The camera is fixed underneath the payload pod at the front of the Skate (pointed at by the yellow arrow in Figure III.1a) pointing downward in the nadir direction and can take up to ten photos per second. However, due to the nature of how the Skate flies (at a slightly upward-pitched angle as shown in Figure III.1c), the camera lens points slightly forward as the Skate flies.

III.3 Flight Planning Algorithm

The following subsections describe all of the parameters that are involved in calculating the path the UAV will fly to photograph the desired site.

III.3.1 Flight Altitude

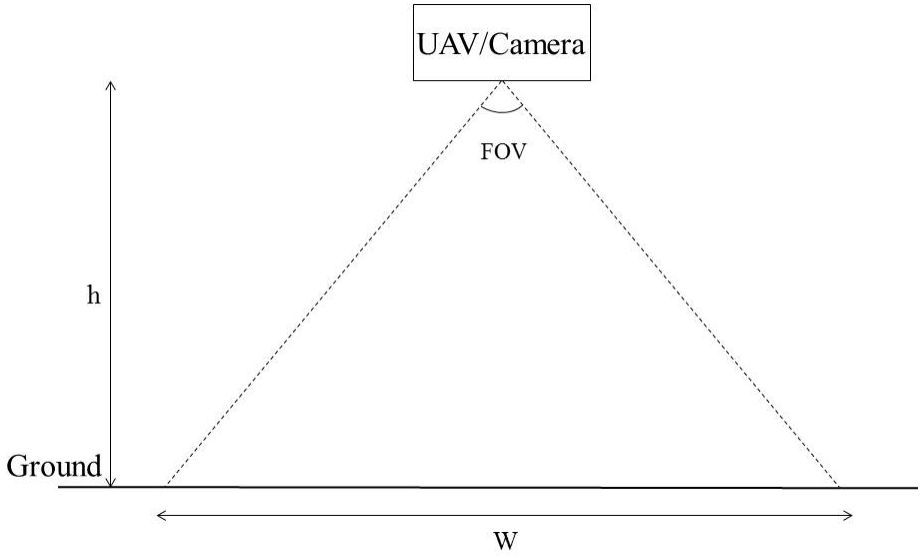


Figure III.2: Ground coverage (W) of an image captured at height h given camera Field Of View.

Determining the altitude above ground to fly at (h) involves specifying a desired GSD and the resolution, in pixels, of an image captured by the payload camera ($R_w \times R_h$), where R_w is the number of pixels in the width direction and R_h is the number of pixels in the length direction. The GSD is specified by the user (typically around 1-4cm per pixel, as stated in Section II.5.2.2) and can be described as follows:

$$\text{GSD} = \frac{\text{Land Covered per Image } (W)}{R_w}. \tag{III.1}$$

Based on Figure III.2, an equation for W can be stated as:

$$W = 2h \cdot \tan\left(\frac{\text{FOV}}{2}\right), \tag{III.2}$$

where FOV is the camera's Field Of View. Lastly, solving Equation III.1 for W by setting it equal to Equation III.2 and solving for h:

$$h = \frac{\text{GSD} \cdot R_w}{2 \cdot \tan\left(\frac{\text{FOV}}{2}\right)} \tag{III.3}$$

provides the optimum altitude above ground to fly at in order to achieve the desired GSD. If the GSD is specified in centimeters, then h will also be in centimeters.

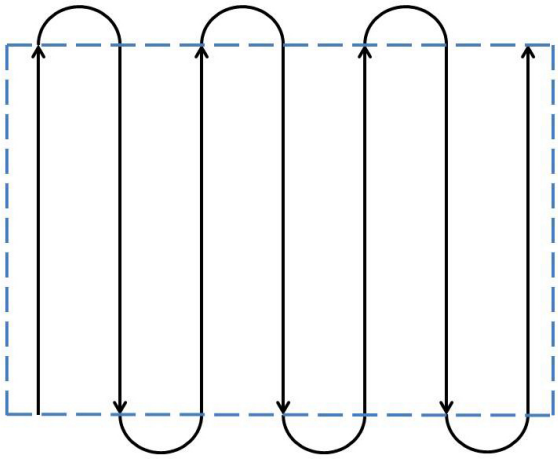


Figure III.3: Example of a back-and-forth flight path (solid line) to cover a convex polygon region (dashed line)

III.3.2 Area Geometry

The optimum path pattern to cover a convex polygonal region is a back-and-forth pattern (see Figure III.3) (Burkard et al., 1998). Every time the UAV has to make a turn, it increases the amount of power used and the total flight time. Therefore, minimizing the number of turns involves calculating the direction of the polygon that is smallest (also known as the width of the polygon) and specifying the flight trajectory to be orthogonal to the width direction. The polygon width is efficiently calculated using a rotating calipers algorithm (Houle and Toussaint, 1988).

After determining the direction of each flight line, the other variable to calculate is the separation between each flight line in meters (LS). Calculating the line separation requires the user to specify the image side overlap (O_s) percentage that typically ranges from 60% to 95%, as shown in Table II.4. After specifying the overlap value, the line separation amount is calculated as:

$$LS = (1 - O_s) \cdot \frac{\text{GSD} \cdot R_w}{100}. \tag{III.4}$$

Side overlap percentage is specified in decimal form and the GSD in centimeters. The same equation (Equation III.4), substituting R_h for R_w, can be used to calculate the distance to travel in the forward direction before taking another picture, or more applicably, to calculate the time separation, Δt, between each captured image:

$$\Delta t = \frac{(1 - O_f) \cdot \text{GSD} \cdot R_h}{100 \cdot \text{flight speed}}, \tag{III.5}$$

where flight speed is specified in meters per second.

III.3.3 Wind

Wind can potentially have a significant effect on a UAV's trajectory as well as image quality. As with aircraft takeoff and landing, a UAV is most stable when it is flying with or against the wind. Flying against the wind will consume more battery power, if that is the direction chosen, but the UAV will only fly against the wind half of the time—the other half of the time will be spent flying with the wind, allowing the motors to throttle back. Image quality is certainly more important than energy consumption, since blurred images caused by the UAV rocking back and forth from the wind will render the captured images useless. Therefore, in the presence of significant wind speeds, it makes sense to choose the direction of each back-and-forth path to be parallel to the wind vector. The wind speed and direction can be detected either through a preliminary test flight and using a hybrid model (Osborne and Rysdyk, 2005) or by using an anemometer. There are several, high-quality commercial anemometers that can send data over Bluetooth in real-time. This thesis used a Kestrel 4500 weather station (KestrelMeters, 2012) when wind information was needed.

This method, however, assumes that the wind vector does not change in-flight. The current version of the path planning algorithm only considers this scenario. The algorithm captures the current wind vector from a Kestrel 4500 weather station when the flight plan is generated. Future improvements to the algorithm will consider the more realistic case where the wind vector changes in-flight. The system will run while the UAV is flying and dynamically alter waypoints to better accommodate changes in wind.

III.3.4 Combining Geometry and Wind

The flight vector suggested by considering only site area geometry ($\vec{G}$) can be significantly different from the flight vector suggested by only considering wind ($\vec{W}$). Trying to satisfy both site area geometry and wind will almost always lead to a conflict. However, it is possible that choosing a flight vector that is a combination of the two individual vectors calculated by considering only geometry and only wind may lead to a more effective vector than considering just one of the two. A more effective flight vector is considered to be one that causes the total flight time to be less.

The current version of the path planning algorithm in this combination mode calculates the new flight vector ($\vec{C}$) as the average of $\vec{G}$ and $\vec{W}$:

$$\vec{C} = \frac{\vec{G} + \vec{W}}{2}. \tag{III.6}$$

The efficacy of taking the average of the two vectors suggested by considering geometry and wind independently is discussed in Section III.4.

Simply taking the average of the two vectors only considers the direction of each vector. Future improvements will give higher priority to the wind vector as wind speed increases. The *hypothesis* is that taking the

average of the two vectors when the wind speed is almost negligible (e.g., 1-5 m/s) or extreme (e.g.,10-15 m/s) will not be as effective as when the wind speed is moderate (e.g., 5-10 m/s).

III.3.5 Optimum Time-of-Day

Ideally, mapping a site will take place when the sun is directly above the structures being photographed, causing there to be no shadows in the images. However, since the sun is only perfectly overhead at the equator, there will almost always be some shadows in the images. The goal is to find the time of day where the sun is at its highest point in the sky, thus casting the smallest shadows. A suggestion can be provided by the planning system to the user as to the best time of day to ideally perform a UAV photogrammetry mission. There are several open source software projects in various programming languages that provide an estimate of the position of the sun given geographical location, time of day and various other parameters. The code can then be used in reverse to calculate time of day when the sun is highest by performing an iterative hill-climbing optimization search. Since the camera is fixed to the UAV pointing downward, issues involving sun glare washing out parts of the captured images will most likely not occur. However, if a pan/tilt system were to be used, other techniques for ensuring image quality will need to be explored.

Wind has already been stated to be a factor for path planning, but the time-of-day can also affect the wind vector. Observations of the wind patterns throughout the day at Mawchu Llacta showed that there are minor winds in the morning and then almost no wind until around 2-3pm when the wind speeds increase significantly. The current system does not account for these wind conditions. Knowing the expected wind conditions throughout the day in advance, either from forecast data or from observations could potentially be used to find a balance between the time-of-day when the sun is at the optimal position and the time-of-day when the wind is slowest. Similar to combining geometry and wind, combining temporal wind conditions and sun position can cause conflicts in the optimal time-of-day suggested. However, the two different time-of-day suggestions can also be averaged or weighted as suggested for combining geometry and wind in Section III.3.4.

III.3.6 The Algorithm

The hardest aspect of designing an algorithm to be used by non-technical people is making the algorithm inputs easily and intuitively specified. Since one of the broad goals of the system described in this chapter is making it usable by non-technical users, the current version of the system utilizes Google's map interface inside of a Webkit window. Google's map interface provides drawing mechanisms that are implemented to allow the user to specify the convex polygonal region that the UAV will fly over (see Figure III.4). However, map tiles pulled from Google's servers cannot be cached, restricting the usability of the system in Wi-Fi

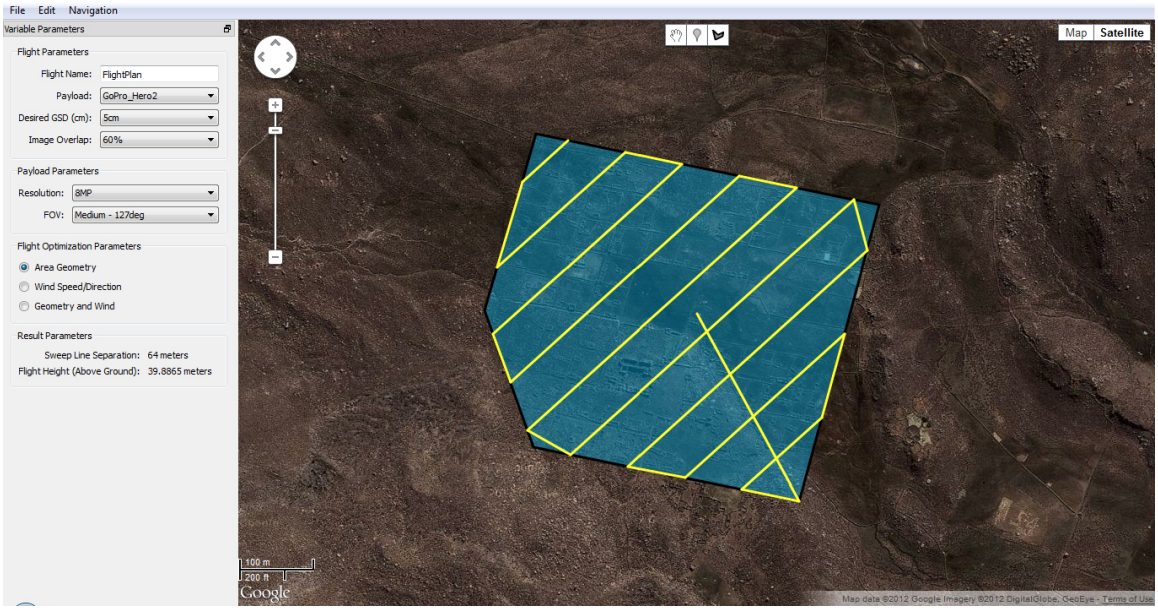


Figure III.4: Interface developed for retrieving input parameters from a user.

deprived areas. The side panel allows the user to specify all of the other flight parameters. Payload parameter options are specified in an XML file and the options associated with a specified payload update in the drop-down boxes when a new payload is selected. The path lines shown in Figure III.4 only connect the calculated waypoints and do not represent the actual path the UAV will fly. The UAV flies in semi-circles outside of the polygon when turning to fly the next path line, as illustrated in Figure III.3.

Once all of the algorithm parameters are specified by the user, a flight plan can be generated. The algorithm for calculating the series of waypoints is presented in Algorithm 1, where "P" represents the area polygon, "mode" represents which optimization mode (i.e., geometry only, wind only or average of both) is specified and "s_0" represents the current location of the UAV.

Algorithm 2 calculates the closest point in the set of polygon border points to the UAV's current location that is clamped by the parallel lines defined by the flight heading. As a means for visualization, consider two infinite lines that are parallel to the flight heading, such that the polygon is between those two lines. Move those parallel lines inward toward the polygon until they intersect with the polygon and those two intersection points are p_n and p_s. This procedure is followed because if the first point were selected to be simply the closet point in the set of polygon border points to s_0, then flying in the calculated heading direction may make the UAV fly through the middle of the area, forcing the UAV to have to backtrack to cover the rest of the site.

Each waypoint's height-above-ground value in the set of waypoints returned by Algorithm 1 is set to be the height value calculated by Equation III.3. All of the waypoints are written to an XML flight plan file in a

Algorithm 1 The algorithm for calculating the series of waypoints for the UAV's path.

CalcWaypoints[P, GSD, O_s, O_f, $\vec{W}$, R_w, R_h, FOV, mode, s_0]

plan := empty array of waypoints
heading := $\vec{G}$, $\vec{W}$ or $\vec{C}$ depending on mode
plan[0] $\leftarrow$ s_0
s := $\leftarrow$ StartPointOnPolygon(s_0, P, heading) (see Algorithm 2)
plan[1] $\leftarrow$ s
$LS \leftarrow (1 - O_s) \cdot \frac{\text{GSD} \cdot R_w}{100}$.
intersectLine := infinite line defined by s and heading
while intersectLine has at least 1 intersection point with P **do**
    intersectLine $\leftarrow$ intersectLine moved parallel into P, LS meters
    $[p_i, p_j] \leftarrow$ intersection points between intersectLine and P
    if p_i is closer to s than p_j **then**
        Add p_i to end of plan
        Add p_j to end of plan
        $s \leftarrow p_j$
    else
        Add p_j to end of plan
        Add p_i to end of plan
        $s \leftarrow p_i$
    end if
end while
return plan

Algorithm 2 The algorithm for calculating the first waypoint to fly to on the polygon border.

StartPointOnPolygon[s_0, P, heading]

Rotate points in P about the center, -heading degrees
p_n := northern most point in rotated set
p_s := southern most point in rotated set
d_n := great circle distance between p_n and s_0
d_s := great circle distance between p_s and s_0
if $d_n < d_s$ **then**
    return p_n
else
    return p_s
end if

format specified by the Paparazzi autopilot system. A flight plan can be simply specified as a plan that flies from one waypoint to the next. However, the Paparazzi system also has a built-in polygon survey flight mode. This flight mode requires the set of points defining the polygon border, the first point to fly to on the border, as calculated by Algorithm 2, the angle of the flight vector, the separation distance between each flight line and the altitude at which to fly. The polygon survey mode proved to be the better solution in preliminary tests because it calculates the semi-circle paths connecting each flight line (see Figure III.3) and tells the UAV to fly that path instead of simply connecting them with a straight line (see Figure III.4). Following the semi-circle paths between each flight line reduces waypoint overshooting and zenith behavior.

III.4 Simulation Data

This section presents the results from running simulations testing a UAV's performance in each of the three optimization modes presented in Sections III.3.2, III.3.3 and III.3.4 in the presence of varying levels of wind. All of the experiments used a rectangular area that was 380m $\times$ 658.179m, with a surface area of 250,108.137 m^2. The reason for choosing a rectangular area with these dimensions was so that one side was significantly longer than the other, but that it also approximates the same surface area as the Mawchu Llacta site. The rectangle was tested at four different rotation levels in the counter-clockwise direction—$0^\circ, 30^\circ, 60^\circ$ and 90°. Those four rotation levels were chosen so that the wind direction remains constant, while testing the effects of wind on the flight path. Rotating the polygon area instead of rotating the wind direction provides better visualization of the different flight paths for each simulation.

At each of those four rotation amounts, wind speeds were chosen to be 0, 2, 4 and 6 m/s (or 0, 4.5, 8.9 and 13.4 mph) in the east to west direction. Preliminary simulations showed that wind speeds of 10 m/s and above resulted in flight paths that are completely unusable (see Figure III.5). Therefore, wind speeds of 0, 2, 4 and 6 m/s were chosen to span the range of wind speed values that show the varing effects of wind speed on the different optimization modes, while still producing potentially usable flight paths. The range of wind speed values will vary depending on the capabilities of the specific platform being used, but for the aircraft model used in the simulations, the Microjet model supplied with the Paparazzi software, that range of values was sufficient to see the effects of wind speed on each of the optimization modes presented in Sections III.3.2, III.3.3 and III.3.4. Lastly, with the exception of the no wind case, each of the experiment combinations were tested with each of the three optimization modes. Forty simulations were performed in total. All of the trials are presented in Table III.1.

Simulations were performed using Paparazzi's built-in simulator. The wind speeds were specified under the environment parameters. The Microjet model was chosen as the simulation model, because it comes with Paparazzi, making replicating the results easier. The aircraft configuration file for each aircraft model

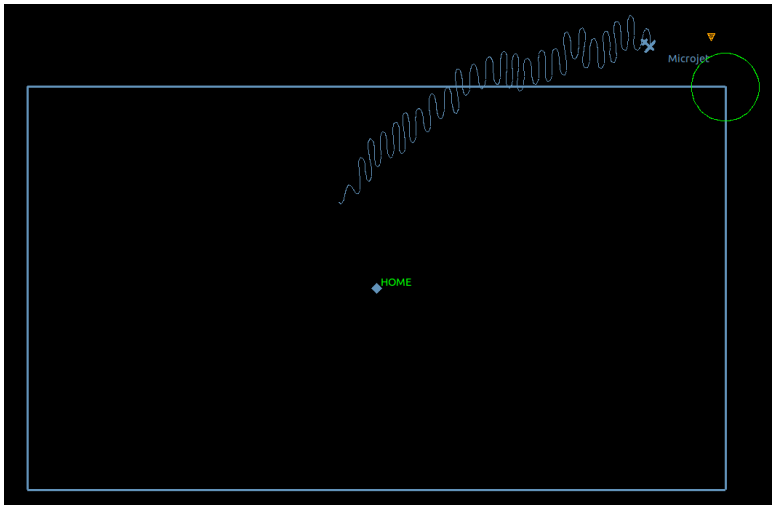


Figure III.5: Preliminary simulation with wind speed set to 10 m/s. Flight path shows UAV attempting to fly into the wind towards the corner of the polygon to start the survey.

specifies platform-specific parameters such as the battery specification, minimum and maximum power of the motors, minimum and maximum values for yaw, pitch and roll, etc. The Microjet configuration differs from the Skate configuration in a few different ways that may affect simulation results. First, the cruising speed for the Microjet is 13 m/s, whereas it is 8 m/s for the Skate. Second, the maximum roll value for the Microjet is 0.7, whereas it is 0.46 for the Skate. That means the Skate has a larger turning radius than the Microjet. Lastly, the gain values for the PID controller are different between the two aircrafts, but those values are fine-tuned to be specific to the dynamics of each aircraft.

When running a simulation, the simulator only shows a certain length of the total flight path, that seems to be different for each simulation, most likely due to arbitrary buffer lengths. Therefore, the simulation images presented do not show the entire path of the flight. Screenshot images were captured at particular times to show varying zenith effects. The simulation times reported in Table III.1 represent the amount of time the UAV was actually performing the image capture mission and do not include take-off and landing times. The simulator tracks the total flight time from take-off to landing, but since only the amount of time-in-flight that the UAV was performing its back-and-forth survey mission was desired, the flight time was recorded manually using a stopwatch and the total flight time was ignored.

The main *hypothesis* for these simulations was that the time saved by taking into account only geometry will not be significant enough to make up for the potential off-path behavior of the UAV in the presence of wind. Therefore, combining geometry and wind mode will provide a balance between path variance and total flight time.

#	Rotation	Wind Speed	Priority	Duration (seconds)	Speed Range (m/s)	# Flight Lines
1	0	0	Geometry	357.5	12.9	6
2	30	0	Geometry	357.3	12.9	6
3	60	0	Geometry	357.5	12.9	6
4	90	0	Geometry	357.6	12.9	6
5	0	2	Geometry	373.6	10.8 - 14.9	6
6	30	2	Geometry	370.5	11 - 14.7	6
7	60	2	Geometry	367.3	11.8 - 13.8	6
8	90	2	Geometry	369.7	12.8	6
9	0	4	Geometry	417.8	8.9 - 16.9	6
10	30	4	Geometry	405.4	9.3 - 16.3	6
11	60	4	Geometry	397.6	10.4 - 14.4	6
12	90	4	Geometry	386.9	12.3	6
13	0	6	Geometry	472.5	6.9 - 18.9	6
14	30	6	Geometry	459.4	7.3 - 17.7	6
15	60	6	Geometry	434.5	8.8 - 14.9	6
16	90	6	Geometry	420	11.4	6
17	0	2	Wind	373.4	10.9 - 14.9	6
18	30	2	Wind	459.4	10.9 - 14.9	10
19	60	2	Wind	488	10.9 - 14.9	12
20	90	2	Wind	412	10.9 - 14.9	10
21	0	4	Wind	415.9	8.9 - 16.9	6
22	30	4	Wind	522.8	8.9 - 16.9	10
23	60	4	Wind	540	8.9 - 16.9	12
24	90	4	Wind	455.2	8.9 - 16.9	10
25	0	6	Wind	477.1	6.9 - 18.9	6
26	30	6	Wind	598	6.9 - 18.9	10
27	60	6	Wind	618	6.9 - 18.9	12
28	90	6	Wind	519.6	6.9 - 18.9	10
29	0	2	Both	373.9	10.8 - 14.9	6
30	30	2	Both	451	11 - 14.8	8
31	60	2	Both	499.5	11 - 14.8	10
32	90	2	Both	500.4	11.4 - 14.3	11
33	0	4	Both	416.3	8.9 - 16.9	6
34	30	4	Both	478.2	8.9 - 17	8
35	60	4	Both	538.2	9.3 - 16.5	10
36	90	4	Both	543	9.7 - 15.5	11
37	0	6	Both	473.4	6.9 - 18.9	6
38	30	6	Both	587.7	6.9 - 18.10	8
39	60	6	Both	593.2	7.1 - 18.9	10
40	90	6	Both	599.1	7.8 - 16.6	11

Table III.1: Path Planning Algorithm Simulation Data

III.4.1 Rotated 0°

Rotated 0°, the flight vectors suggested by the geometry and wind optimization modes are the same because the minimum width of the polygon is perpendicular to the wind vector, which means the vector suggested by the combination mode is also the same. Therefore, this set of simulation experiments serves as the control set. Since the simulation times were almost identical for all three optimization modes, as can be seen in Figure III.6, it can be assumed that the simulation software is reliable. The slight variation in completion times is most likely due to human error in operating the stopwatch.

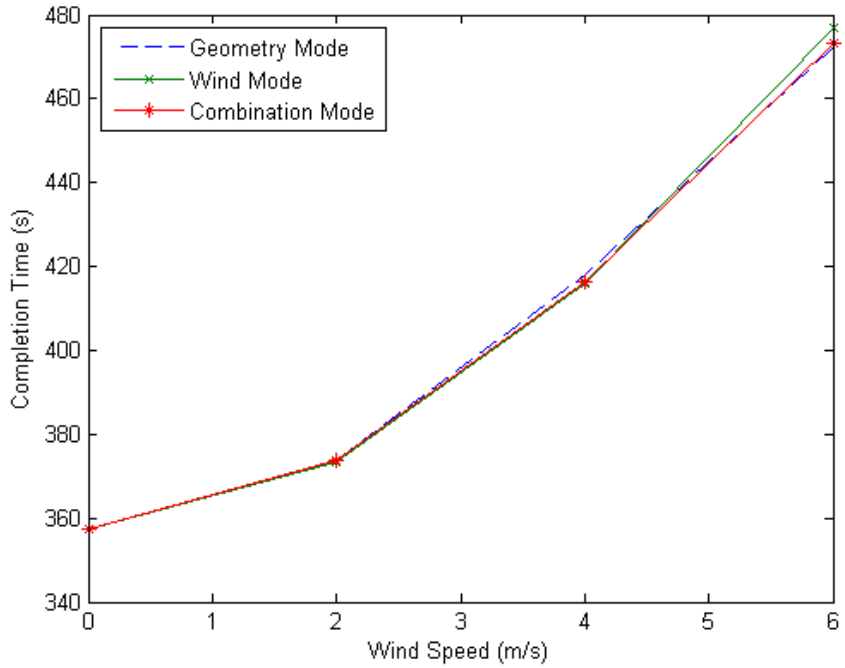


Figure III.6: Simulation completion times for each optimization mode with the rectanglular area rotated 0°.

III.4.2 Rotated 30°

Rotating the area polygon about its center 30° counter-clockwise allows the differences between each of the three modes to be observed. A comparison of the three modes subjected to 4 m/s winds is presented in Figure III.8. The paths in geometry mode and wind mode appear mostly the same, aside from the different flight vector. However, in the combination mode, the path oscillates the entire length of each flight line when flying into the wind. All of the optimization modes show anomalies in the semi-circles that the UAV is supposed to fly when turning around outside of the polygon. The simulator appears to perturb a small ($\leq$ 1 m/s), random wind vector even when no wind is specified. However, the wind simulated is so small that it should not cause the path blips. Therefore, the almost uniform anomalies seen in all of the simulations when the UAV

turns are most likely caused by the turn radius being larger than the UAV's minimum turning radius. The UAV undershoots the turn and so it has to overcompensate to correct, causing the second bump seen in the simulation paths. Geometry mode is clearly the best choice in terms of flight time (see Figure III.7). Wind and combination modes have roughly the same flight times, but the oscillations present in the combination mode are undesirable.

Although the flight lines in wind mode appear to have roughly the same levels of oscillation as geometry mode, there are a total of ten flight lines, compared to the six needed in geometry mode. The combination mode had only eight flight lines, which suggests that the flight time will be between the flight times for the geometry and wind modes, but since there are significant oscillations in the flight path, the flight time increases to be approximately the same as in wind mode.

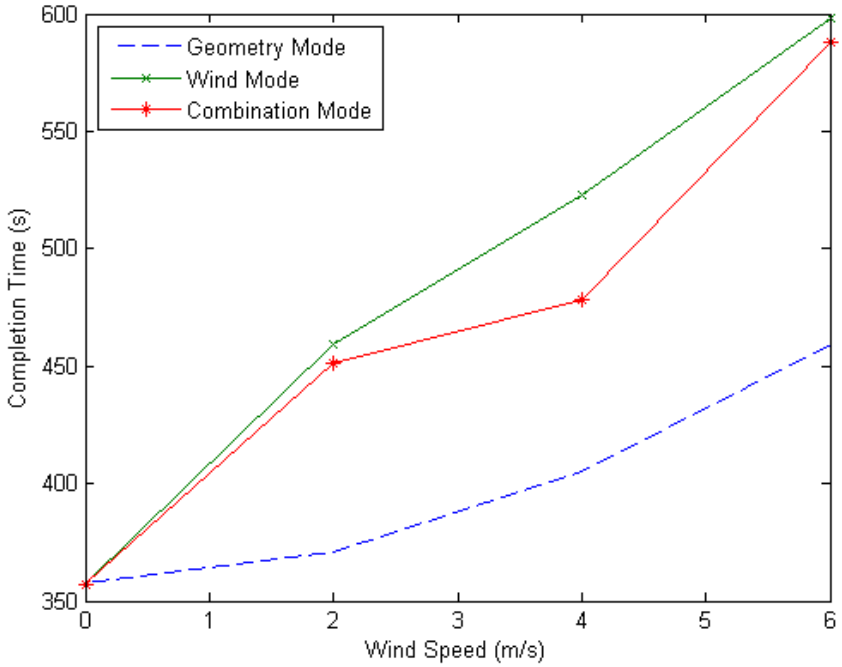
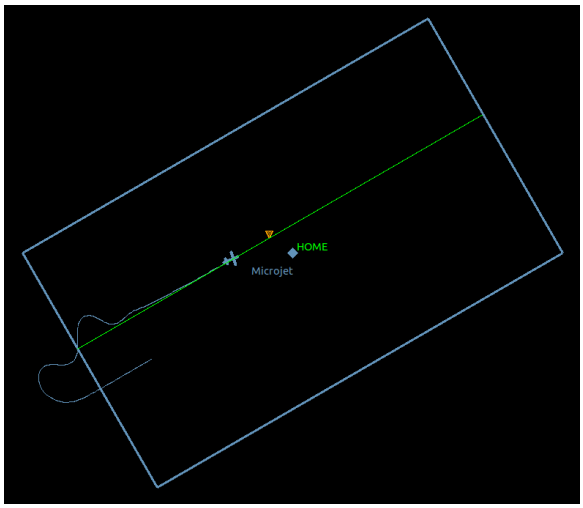


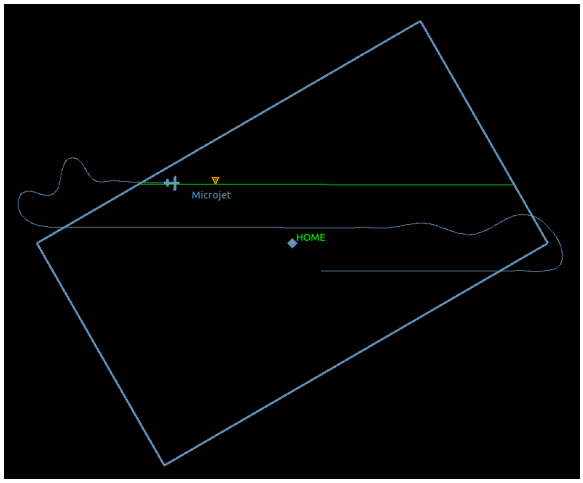
Figure III.7: Simulation completion times for each optimization mode with the rectanglular area rotated 30°.

III.4.3 Rotated 60°

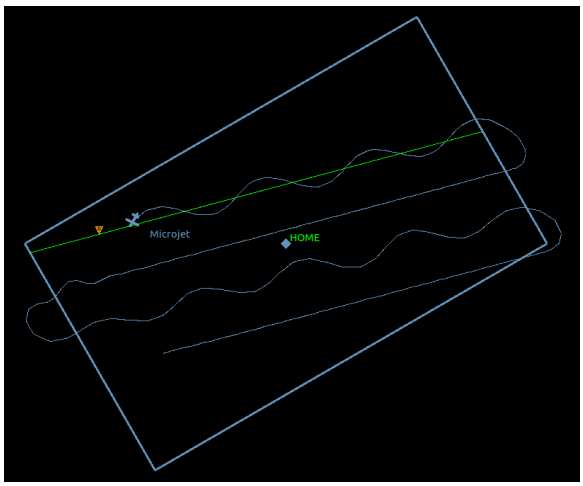
Rotating the polygon 60°, further increases the difference between the geometry mode flight vector and the wind mode flight vector. Since the wind mode will always suggest a flight vector that is rotated 0°, and the geometry mode flight vector will always be rotated the same amount as the polygon, the angle between the two vectors has increased to 60° from 30°. Geometry mode clearly provides the best result in terms of total flight time with wind and combination modes having similar performance to each other (see Figure III.9).



(a) 4 m/s wind in geometry mode.



(b) 4 m/s wind in wind mode.



(c) 4 m/s wind in combination mode.

Figure III.8: Area rotated 30° in all three modes.

The oscillations seen when the rectangle was rotated 30° in combination mode (Figure III.8c) are absent from the simulation shown in Figure III.10c, but is most likely due to the wind velocity being only 2 m/s in that trial. Oscillations were more pronounced when the wind was set to 4 and 6 m/s in combination mode.

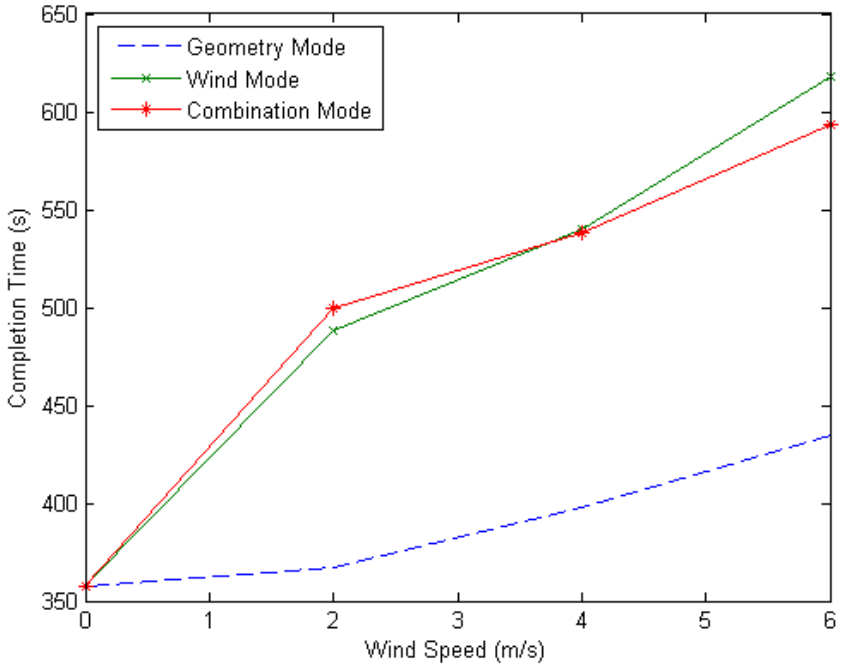
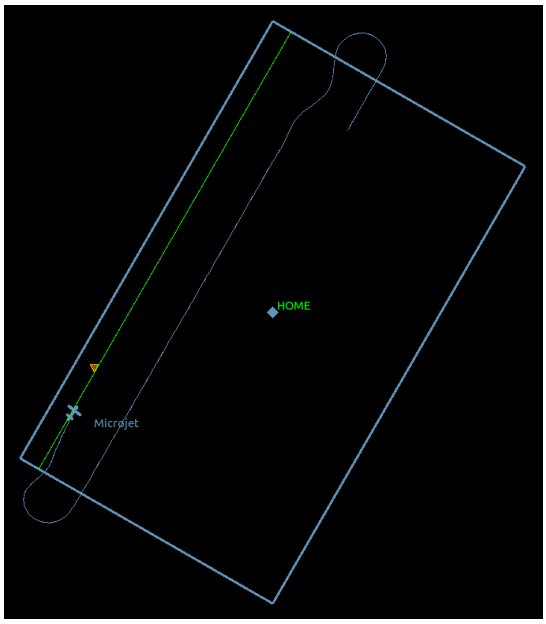


Figure III.9: Simulation completion times for each optimization mode with the rectanglular area rotated 60°.

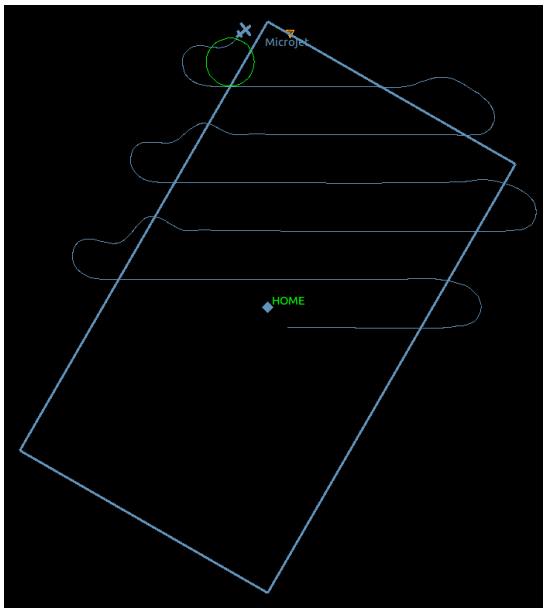
As with the set of simulations with the polygon rotated 30°, the increased number of flight lines and number of oscillations present in the wind and combination modes makes geometry mode outperform the other two modes. Geometry mode only required six flight lines, whereas wind mode required twelve flight lines and combination mode required ten flight lines. The differences in flight line counts can be seen in Figure III.10. Although not all of the flight lines are shown, the reader can see that there are clearly more total flight lines in Figure III.10b, than in Figure III.10a, for example.

III.4.4 Rotated 90°

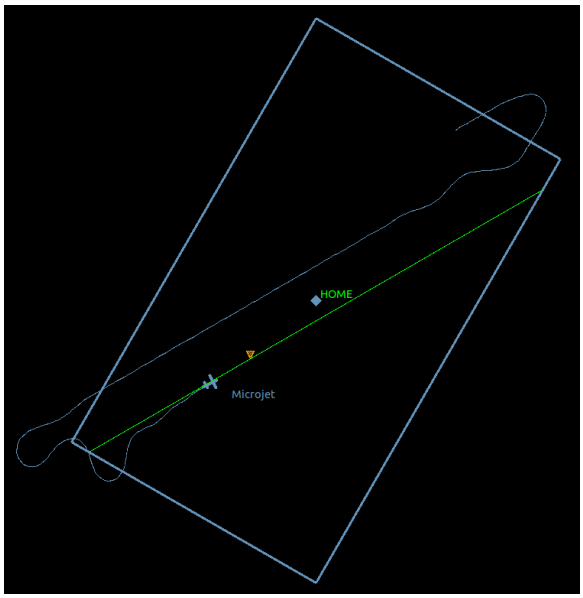
Rotating the rectangular area 90° causes the flight vector suggested by geometry mode to be perpendicular to the flight vector suggested by wind mode. Even under these conditions, geometry mode is the best in terms of flight time and path accuracy (see Figures III.11 and III.12). The simulations for wind set to 4 m/s and the rectangle rotated 90° in each of the three flight modes are presented in Figure III.12. The geometry mode simulations performed surprisingly well with only minor oscillations at the border of the area. However, both the wind and combination modes had significant oscillations in their flight paths. This set of simulations was



(a) 2 m/s wind in geometry mode.



(b) 2 m/s wind in wind mode.



(c) 2 m/s wind in combination mode.

Figure III.10: Area rotated 60° in all three modes.

the only one where the combination mode performed significantly worse than either the wind or geometry modes. All of the other sets of simulations had the combination mode performing either slightly better or about the same as the wind mode. Those results are most likely because the amount of flight path outside of the flight region is greater, as can be seen when comparing the turning path outside of the flight region between Figures III.12b and III.12a.

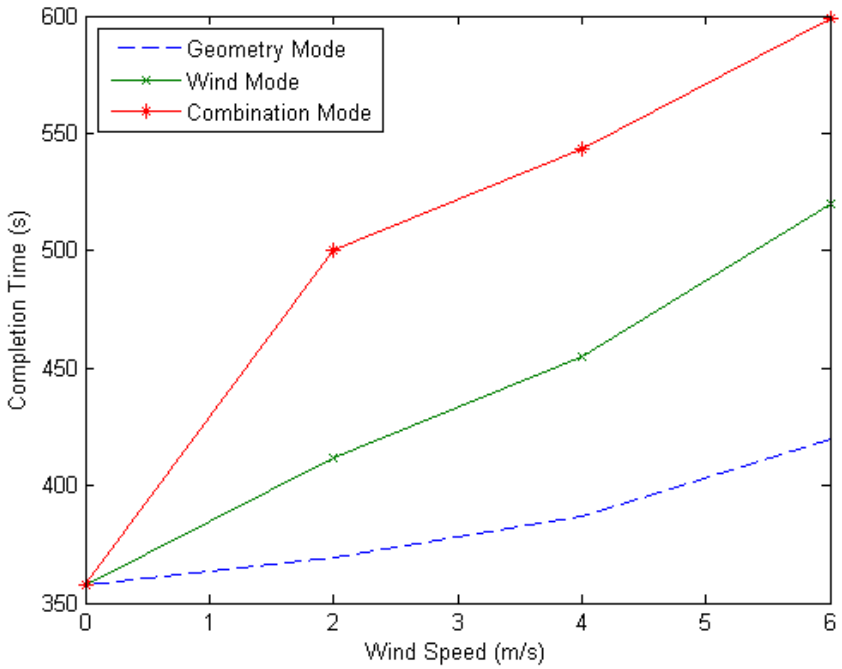
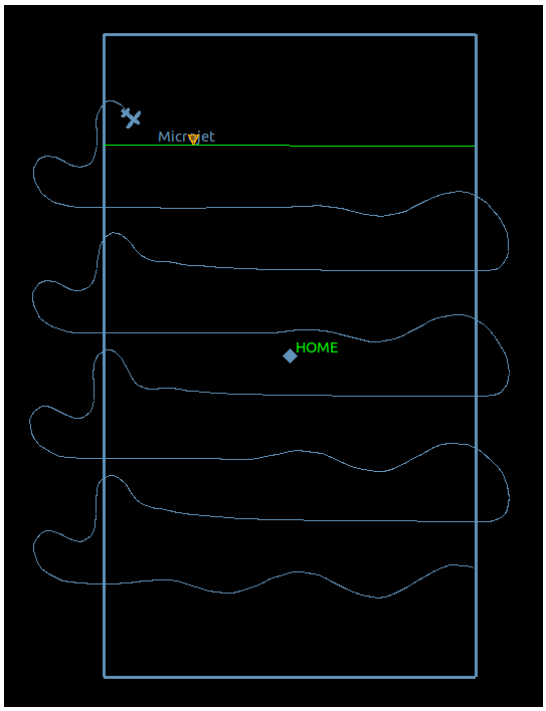


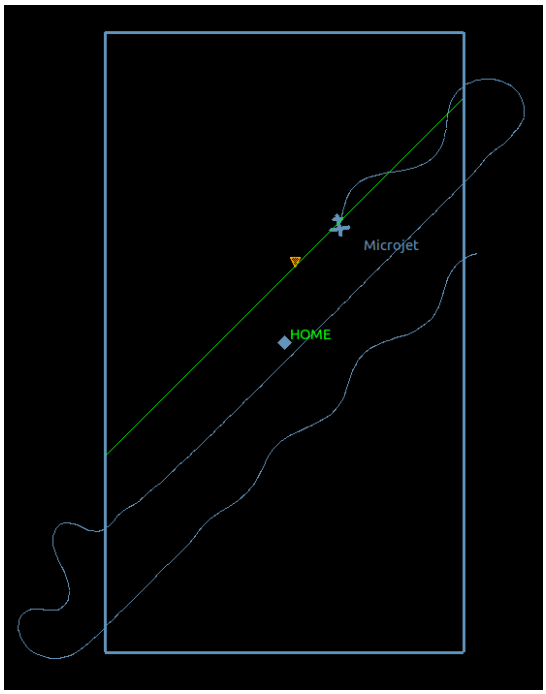
Figure III.11: Simulation completion times for each optimization mode with the rectangular area rotated 90°.

III.4.5 Simulation Summary

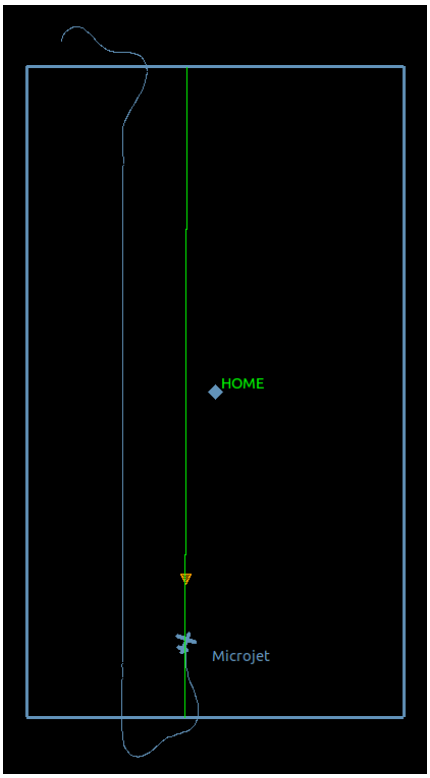
The initial *hypothesis* that an UAV will be more stable when flying with or against the wind was surprisingly inaccurate—at least when simulated using the Microjet model. Not only did flying parallel to the wind vector cause significant oscillations in the flight paths, but also significantly increased the overall flight time. Geometry mode performed the best in each of the rotation sets, even when the flight vector suggested was perpendicular to the wind vector. The number of flight lines significantly determined which optimization mode was the best in terms of flight time. Geometry mode always required only six flight lines, whereas wind mode sometimes required up to twelve flight lines (see Table III.1). The cause for the oscillations is probably in how the on-board controller logic is designed. It is unclear from the sparse documentation for Paparazzi what PID controller variant is used for correcting a flight path, but since the simulations seemed to have trouble converging on the actual path in the presence of wind, it is most likely just a PI controller.



(a) 4 m/s wind in wind mode.



(b) 4 m/s wind in combination mode.



(c) 4 m/s wind in geometry mode.

Figure III.12: Area rotated 90° in all three modes.

Since the motivation behind developing a combination mode had to do with the *hypothesis* that the UAV will perform better sometimes when flying parallel to the wind vector, it makes sense that this mode provided no additional value. Throughout the experiments, there seemed to be a correlation between variance in path velocities and the total simulation time. Even though the average wind speed may have stayed the same, when the variance in wind speed increased, so did the total flight time. Furthermore, when the maximum flight speed increased, it took longer for the oscillations along the path to dampen out. The UAV spends more time flying slower and less time flying faster when the variance in flight speed increases. Furthermore, when flying with the wind, the UAV does not have as much time to converge on its flight line before it has to turn around to start its next flight line. Although, there is less time to converge, there is still sufficient time for a normal PID controller, which makes it appear that the simulator is only implementing a PI controller.

The results from the simulations provide significant insight into what makes a better flight path for performing photogrammetric flight missions. The simulations were performed on a rectangular region, so it is possible that the results will change with geometrically different flight regions. However, these simulations provided sufficient data for evaluating the value of each of the three proposed optimization modes. Two of the primary goals when generating a flight plan for a photogrammetric mission are minimizing total flight time and ensuring the necessary image overlap to generate high-resolution DSMs. The simulations showed that using geometry mode is better for minimizing flight time as well as for ensuring the necessary image overlap. However, the overshooting seen in the simulation results when the UAV turns around, causes the UAV to be off-path consistently at the borders of the flight region, potentially ruining the necessary overlap. Therefore, assuming there is room to do so, the flight region should be expanded on all sides, such that the UAV converges on the desired path before it enters the actual flight region, guaranteeing the necessary image overlap.

CHAPTER IV

Image Analysis

IV.1 Introduction

This chapter presents and discusses the results of using Agisoft's Photoscan to process images captured by the Skate UAV in manual flight mode at Mawchu Llacta over three flights. The number of images, flight duration and time of day for those flights are shown in Table IV.1. The camera's FOV was set to 127° and two images were captured every second for each of the three flights with the UAV flying at an average ground speed of 10 m/s. Given that the Skate did not use the flight planning algorithm and autonomous mode, as presented in Section III.3, the methods presented in this chapter to obtain higher-quality DSMs and orthophotos are for non-idealistic image capture conditions. The images were not captured in a uniform, back-and-forth pattern nor at a uniform altitude above ground. However, idealistic conditions are rarely present in real-world applications, necessitating the development of methods to accommodate for aberrant system behavior.

Flight Time	# of Images	Flight Duration
11:50am	867	07 min. 13 sec.
1:30pm	1157	09 min. 38 sec.
1:50pm	2427	20 min. 13 sec.
Total	4451	37 min. 04 sec.

Table IV.1: Manual flights flown at Mawchu Llacta to capture images.

The processed orthophotos are visually compared to the most recent publicly available satellite image of Mawchu Llacta (see Figure IV.1) to determine quality. The satellite image is good enough to provide a general layout of the site and as a visual cue for specifying the border polygon for generating a path plan, but the resolution is insufficient for further use in the archaeological research. The number of images used and resolution settings for each of the processing trials presented in this chapter are shown in Table IV.2.

Figure	# Images	Source Flight(s)	Matching Resolution	Geometry Resolution
IV.3a	150	All three	High	High
IV.3b	150	All three	High	High
IV.4	1537	All three	High	Medium
IV.6	634	01:50pm	High	High
IV.7a, IV.7b, IV.8a, IV.8b	48	01:50pm	High	High
IV.7c, IV.7d, IV.8c	180	01:50pm	High	High
IV.7	27	None	High	High

Table IV.2: Parameter settings for all of the image processing trials presented in this chapter.

Figure IV.1: Satellite image of Mawchu Llacta captured May 22, 2012. Image obtained from Google Earth (Google, 2012)

.

IV.2 Lens Distortion

The camera used with the Skate UAV was the GoPro Hero2 HD camera (see Section III.2). This camera has a built-in fish-eye lens that distorts the captured image or video radially outward from the center. A potential benefit of using a fish-eye lens can be that it captures more information in a single image allowing fewer images to be captured to cover the entire site. However, since the sensor size does not change, more information is captured with the same amount of pixels, meaning the camera has to be closer to the ground to obtain the same GSD as a normal lens. Using the camera settings described in Section III.2, an ideal balance between height-above-ground and land area covered by a single image was determined. The FOV was set to 127° allowing the UAV to fly at a reasonable height above ground, while ensuring the necessary image overlap (set to 60%). The initial *hypothesis* was that the image distortion requires correction before processing the images in Photoscan—the thought process being that since the distortion increases towards the edges of an image, the point-matching algorithm used by Photoscan will have difficulty accurately matching the points at the edges, creating an overall distorted or inaccurate mosaic.

Removing the fish-eye distortion from the images involved using a third-party script (Weinhaus, 2012) in conjunction with ImageMagick (Magick, 2012). The fish-eye distortion was removed using the following command in a terminal window: `bash defisheye.sh -i 127 -f fullframe IMAGE OUTPUT`, where `IMAGE` is the file path to the image being processed and `OUTPUT` is the file path of where the result will be placed. The resulting images were cropped on all sides to remove the parabolic borders generated by removing the distortion. The following command was used to crop the images: `convert IMAGE -shave 185x295 OUTPUT`. An example showing all three phases of this process (original, distortion removed and cropped) can be seen in Figure IV.2. The resolution of the original image is 3200×2400 and the resolution after cropping is 2830×1810—a loss of 2,557,700 total pixels.

After processing the images with Photoscan, it was determined that Photoscan produced more accurate results when using the distorted images. 150 images were hand selected from all of the captured images and processed using both the distorted and undistorted versions as separate projects. The selected images appeared to cover the same sub-region of the site. Using Photoscan's high-resolution settings for both the image matching and geometry building steps, a DSM/orthophoto was produced using the original 150 images and another using those same 150 images after removing the fisheye distortion. The results from these experiments can be seen in Figure IV.3. It may not be immediately clear that one orthophoto is necessarily better than the other, but after careful visual examination, the orthophoto generated with the original images has fewer artifacts and misalignments than the other orthophoto generated with the undistorted images. The orthophoto generated using the distored images was identified to have eight anomoly areas whereas the or-

(a) Example image with fish-eye distortion

(b) Result of removing the fish-eye distortion.

(c) Cropped image after removing fish-eye distortion.

Figure IV.2: Example image with fish-eye distortion, with distortion removed and after cropped.

thophoto generated using the undistorted images was identified to have twelve anomaly areas, as indicated by the yellow rectangles in Figure **??**. The orthophoto generated with the distorted images also has more visual information, most noticeable at the top of the images. Half of another row of three grid blocks are present in Figure IV.3a at the top that is absent in Figure IV.3b. This is due to the loss of information when the images are cropped after removing the fish-eye distortion.

It is unclear why using the distorted images produces more accurate results, but that phenomenon is most likely caused by the pixel compression during the fish-eye removal process. When removing the distortion, the script can map several pixels of information to a single pixel in certain areas. That compression and loss of information can account for the loss of visual information at the top border of Figure IV.3b, and also for the inaccuracies in point matching.

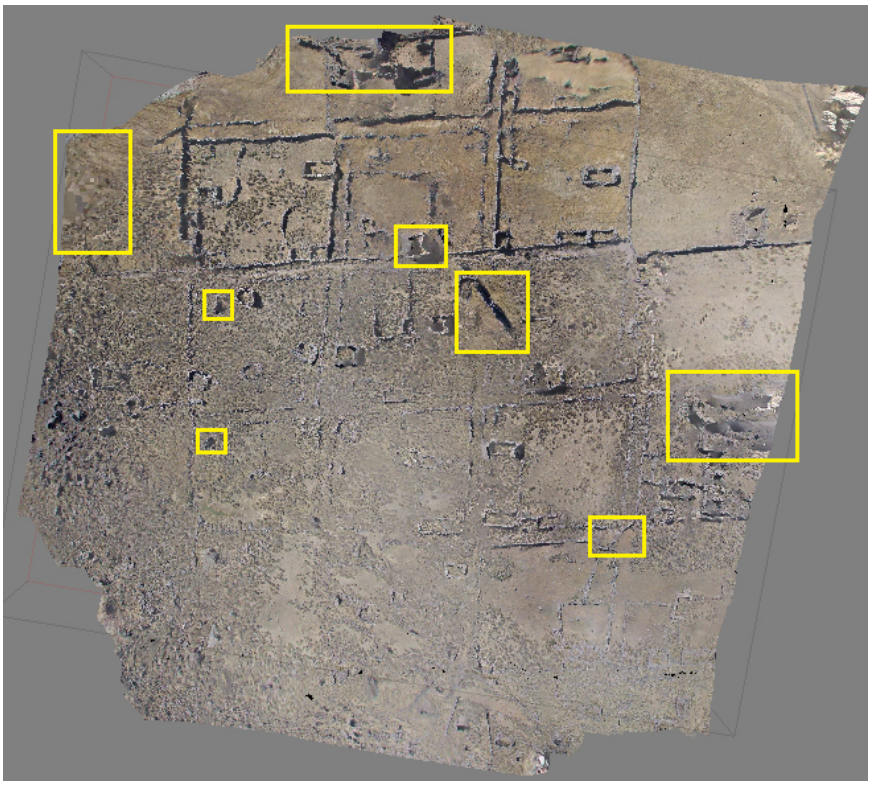
IV.3 Less is More

The intuition is that more images will produce higher-resolution and higher-quality results when it comes to generating high-resolution DSMs and orthophotos, because there is more information for matching between images and potentially more angles from which an object was viewed. That intuition was in fact the *hypothesis*; however, several experiments show that fewer images actually produce more accurate results—as long as there is a substantial overlap between each successive image.

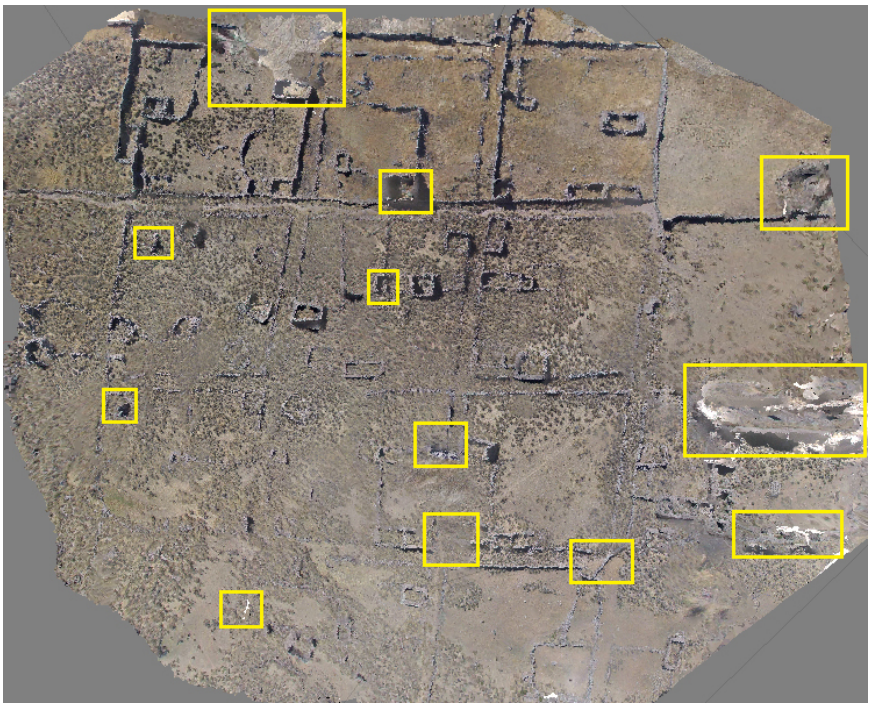
An experiment to test the initial *hypothesis* that more images produced better results processed all of the images captured across the three flights (see Table IV.1) into a single DSM/orthophoto. Every third image was processed to speed up the processing time. The total number of images used after removing every third was 1,537. Photoscan seemed to have a difficult time processing 1,537 images, even considering the technical specifications of the machine used for performing the image processing[1]. Therefore, in order to complete the geometry building process, a lower resolution setting of medium was used. The result is shown in Figure IV.4. Clearly Photoscan had a difficult time matching all of the images, because the result is a contorted mess with practically no useful segments. The estimated camera positions and orientations are shown in Figure IV.4b.

The result shown in Figure IV.4 is most likely caused by the fact that the images used were from all three flights without the guarantee of the necessary image overlap between all images. Also, examining the images used in that experiment led to the *hypothesis* that certain types of images did not match up properly. After examining those images carefully, a list of criteria for images that have a higher probability of being matched incorrectly was generated:

[1] A Linux machine running Ubuntu 12.10 with 64 gigabytes of RAM and twelve AMD Opteron 4184 (six core) processors was used to perform the image processing.

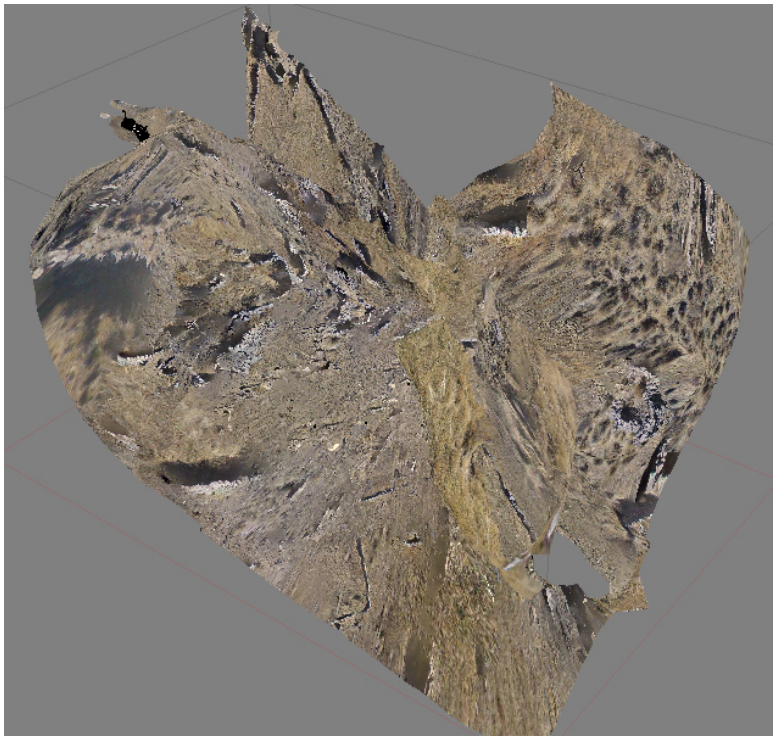


(a) Orthophoto generated using original, distorted images.

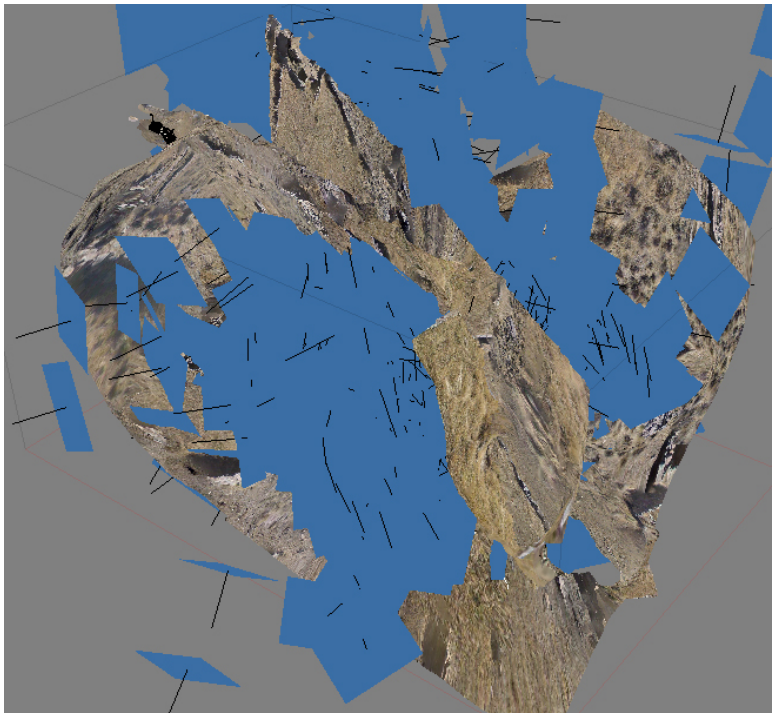


(b) Orthophoto generated using undistorted images.

Figure IV.3: Comparison of orthophotos generated using fish-eye distorted images (Figure IV.3a) and undistorted images (Figure IV.3b). The yellow rectangles indicate areas visually identified to be anomolies compared to the satellite image.



(a) DSM/orthophoto generated using every third image from all images (approximately 1500 images used).



(b) Same DSM/orthophoto as IV.4a, showing the approximated location and orientation of each image.

Figure IV.4: DSM/orthophoto generated using every third image from all images shown with (Figure IV.4b) and without (Figure IV.4a) Photoscan's approximations of the location and orientation of each image.

(a) Example image with uniform texture.

(b) Example image of ambiguous structure.

(c) Example image of far off the nadir axis.

Figure IV.5: Example images demonstrating criteria for images that do not match well in the image stitching process.

1. Images that have a mostly uniform texture—such as images that are solely of grass or dirt (see Figure IV.5a).

2. Images that contain ambiguous structure details—such as images that are of just a single straight wall (see Figure IV.5b).

3. Images that are further off the nadir axis than normal—such as images that contain the horizon or even sky[2] (see Figure IV.5c).

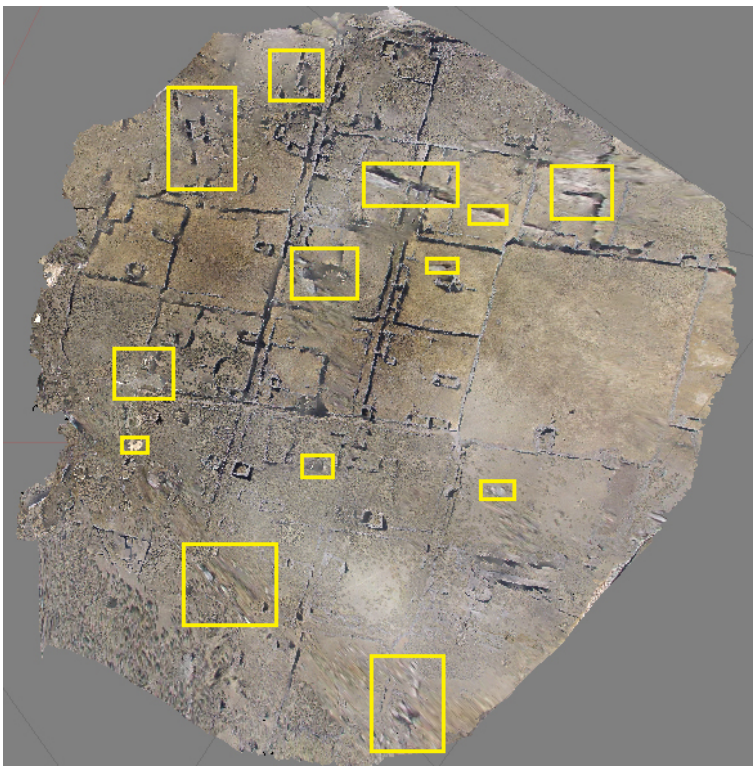
The next experiment was based on the two lessons learned from the previous experiment: fewer images produce better results and some images do not match very well. Therefore, this experiment used only images from a single flight and again only every third image. The images from the 1:50pm flight, the longest flight that also covered the largest region of the site, were used. There were 2,427 images from that flight, so using only every third image and removing all images that met the criteria listed for bad images left 634 images used in this experiment. The results from this experiment are shown in Figure IV.6. It is important to emphasize that every one of the 634 images used in Figure IV.6 were also used in the generation of Figure IV.4, but Figure IV.4 used approximately 58.75% more images where many of those images were either redundant or of poor quality in terms of the provided criteria. Reducing the number of images used and increasing the standard of quality produced a DSM/orthophoto that much more accurately represents the site. However, there are still many anomalies in the generated orthophoto, as indicated by the yellow rectangles in Figure IV.6. The height above ground varied significantly within each flight, so using only every third image does not necessarily guarantee the required 60% overlap. That reduction in overlap to speed up the processing most likely caused the anomalies pointed out.

IV.4 Ideal Performance

Section IV.1 stated that the captured images were not under ideal circumstances. However, certain subsequences of images followed an almost ideal pattern. Two subsequences were evaluated—one where the UAV flew in just a straight-line pattern and one where it flew in a back-and-forth pattern. The results of those experiments are presented in Figure IV.7. Figures IV.7a and IV.7b show the semi-straight line pattern results. Since image data is only captured from one side of the structures in this sequence of images, the geometry of the DSM is inaccurate in certain areas, but the orthophoto has fewer artifacts and misalignments than any of the other previously generated orthophotos. Figures IV.7c and IV.7d show the back-and-forth pattern results. The DSM generated from the back-and-forth sequence of images has significantly more accurate

[2]This occurred a few times throughout the flights when quick wind gusts caused the Skate to tilt abruptly before the pilot was able to counter the wind effects and return to normal flight.

(a) Orthophoto shown without anomaly areas indicated.



(b) Orthophoto shown with anomaly areas indicated by yellow rectangles.

Figure IV.6: Orthophoto generated from every third image of the longest flight after removing images matching the removal criteria for being most likely to not match properly with other images (634 images used). The yellow rectangles indicate areas that were not matched correctly and differ from the satellite image.

.

geometry, since most of the structures in the result were captured from at least two angles (see Figure IV.8). However, additional images not part of the back-and-forth section (the cluster of images present at the top in Figure IV.7d) were part of this sequence, which caused some artifacts to appear. Overall, the resulting DSM/orthophoto is close to what was expected if the UAV were to have been flown in autonomous mode using the path planning algorithm presented in Chapter III.

Figures IV.8a and IV.8b show the top down view of the same structure from the DSM generated using images captured in a straight-line path and a 3D perspective of the DSM showing how the structure geometry is flat. Figure IV.8c shows a 3D perspective of the DSM generated from the images captured in a back-and-forth pattern. The area those images covers is not the same area that the straight-line path covers, so a comparison of the same structure cannot be made, but a structure with similar geometry is still present and a more detailed DSM can still be seen. The orthoimagery of that particular section of the DSM in Figure IV.8c is a little blurry, suggesting that there may have been too much or not enough image information for that section.

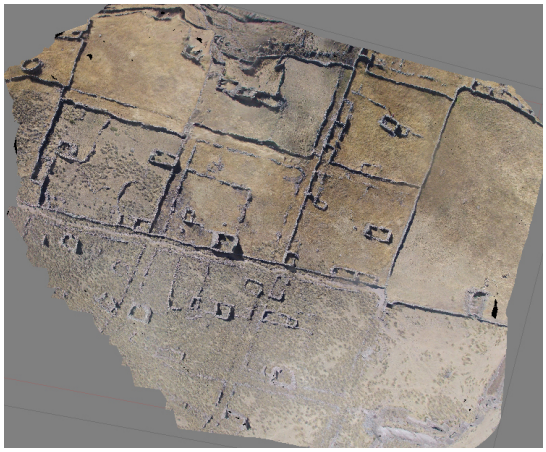
Lastly, to get a sense of what quality of results can be achieved using Photoscan under ideal image capturing conditions, images of a medium-sized rock from all sides were captured, while moving around it twice to capture it from different angles. The results of processing those images are presented in Figure IV.9. Using just twenty-seven images, a very good DSM/orthophoto of the rock was produced. Although this rock experiment may seem like a trivial example, it stands as a comparison and guideline for actual applications. Very few images were used (in comparison to the hundreds of images used in the other experiments presented) and although the images were not captured in a perfect pattern, they were captured close to what is the ideal sequence and that seemed to be good enough for Photoscan to produce a high-quality result.

IV.5 Image Analysis Summary

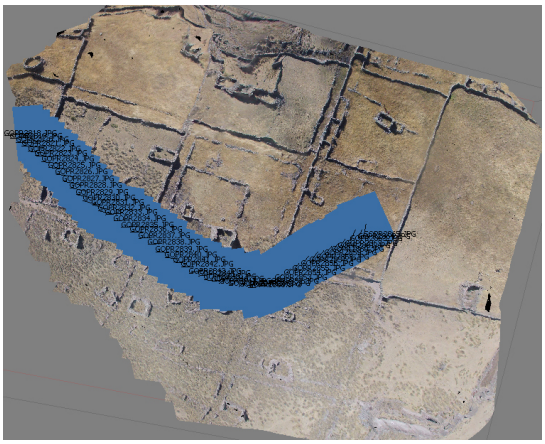
The results of the image processing experiments provided three significant insights into producing better DSMs and orthophotos. First, if the captured images have fisheye lens distortion, removing the distortion before processing the images will result in lower quality DSMs and orthophotos when using Agisoft's Photoscan. Second, the fewer the images used in DSM and orthophoto generation, the better the results. More image information does not mean better results. Furthermore, images that have a uniform texture, contain ambiguous structure details or that are off the nadir axis will most likely cause errors in image matching and should be discarded. Third, the back-and-forth pattern used in the literature and in the path planning algorithm presented in Chapter III does seem to produce the best results.

Once the Skate UAV is fixed to fly in autonomous mode, more thorough image processing experiments can be done at varying levels of image overlap to find the value for that parameter that produces the best

results.

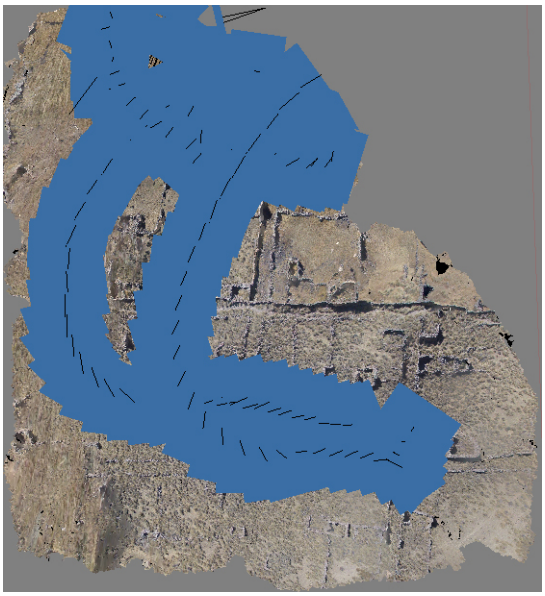


(a) Straight-line pattern orthophoto.



(b) Straight-line pattern orthophoto with estimated camera positions and orientations.

(c) Back-and-forth pattern orthophoto.

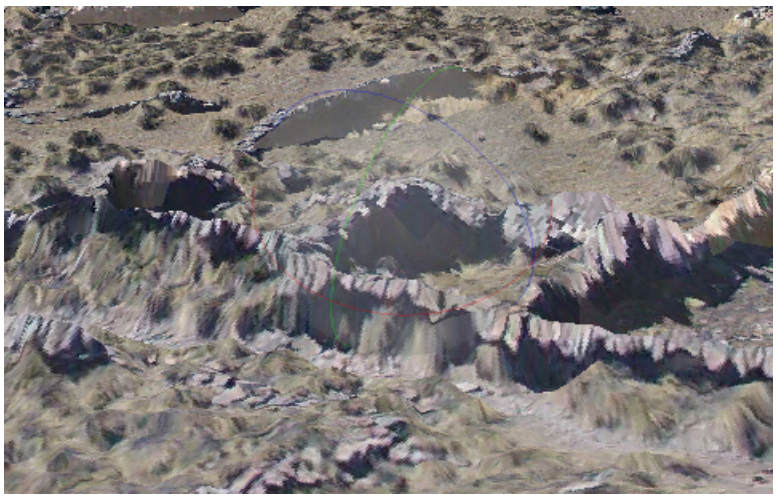


(d) Back-and-forth pattern orthophoto with estimated camera positions and orientations.

Figure IV.7: Straight-line and back-and-forth image capturing pattern orthophoto results.

(a) Top-down view of structure.

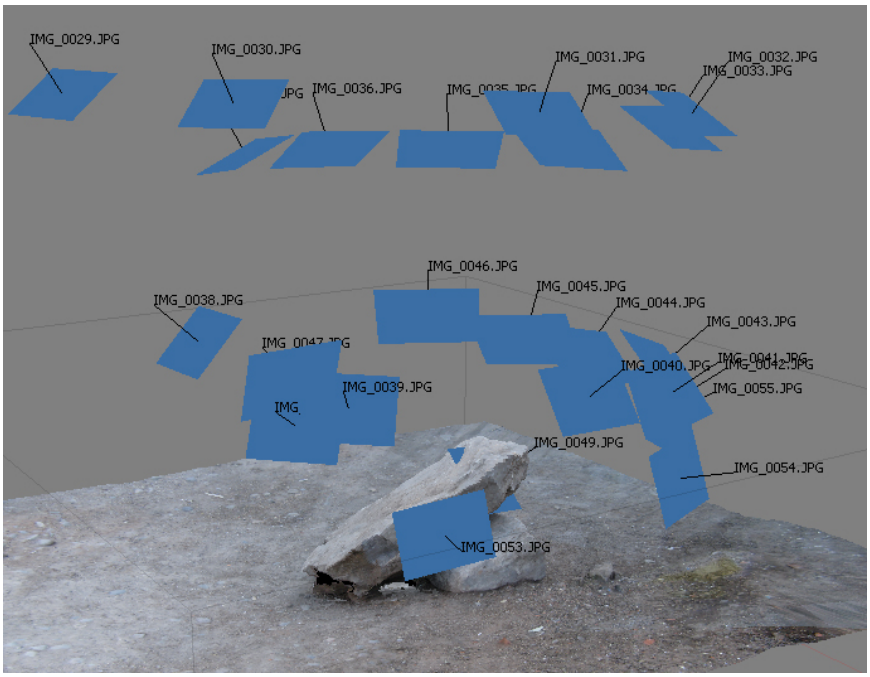
(b) DSM generated from straight-line image path.



(c) DSM generated from back-and-forth image path.

Figure IV.8: DSMs generated from straight-line image path and back-and-forth image path.

(a) Medium-sized rock DSM



(b) Medium-sized rock DSM with estimated camera locations and orientations.

Figure IV.9: Example DSM generated under ideal imaging conditions.

CHAPTER V

Concluding Remarks

V.1 Contributions

The efforts of the work presented in this thesis attempted to expand the efficiency and quality of current methods used for UAV photogrammetry. A new path planning algorithm was presented in Chapter III. Simulations contradicted the initial *hypothesis* that choosing a flight vector parallel to the wind vector will produce a more stable and efficient path. A series of simulations showed that for a rectangular region, optimizing the path based on the region geometry produced the most efficient path in all cases. Combining the suggested vectors from geometry and wind modes proved to be of no additional value and in some cases, produced worse results than choosing geometry or wind mode alone. Path oscillations were most pronounced in the wind and combination optimization modes. Since capturing images in a straight line was shown to produce the best DSMs and orthophotos in Chapter IV, those path oscillations are undesirable. Therefore, geometry mode should be used for rectangular regions.

Unfortunately, the path planning algorithm went untested in 2012 on an actual UAV, which is why simulated results were presented. However, images captured from an UAV flown manually over the archaeological site in the Colca valley of Peru were used as a basis for testing the capability of Agisoft's Photoscan software to generate high-quality DSMs and orthophotos. Image quality standards were established for choosing images that have a higher probability of matching more accurately. Both Photoscan and LPS, the other industry-standard DSM generation software, match images more accurately when GPS and orientation data is known for every image. However, not all UAV photogrammetry platforms have the capability to record GPS and orientation data for every image and so quality standards are necessary for generating more accurate results. Furthermore, example cases were presented to show the validity of using the back-and-forth flight pattern. Orthophotos generated from image sub-sets that follow the back-and-forth pattern were more accurate than orthophotos generated using image sets that were captured in semi-random patterns.

V.2 Conclusion

Although the path planning algorithm presented in this thesis was discussed for the application of archaeological photogrammetry, it can be applied to various other photogrammetry domains. Current systems still require a lot of work before they can be used by the intended users. Even though there has been a lot of research into SLAM techniques for UAVs and as well as performing various surveillance and photogrammetry tasks, there does not seem to be much research involving combining those research areas. An archaeologist

or a first responder cannot be expected to know how to operate a UAV manually for performing their respective tasks and so more advanced planning techniques need to be explored that combine the autonomy of SLAM techniques for operating in an unknown environment with path planning techniques for performing the photogrammetry tasks.

V.3 Future Work

The next step for this work involves confirming the simulated results on our UAV platform. The other largest priority for this system is implementing an interface that allows satellite maps to be cached and loaded offline. Unfortunately, Google's map API does not allow the caching of map tiles. Photogrammetric applications in urban areas where internet access is readily available does not necessarily require the ability to cache map tiles, but this system will most likely be used in areas where internet access is not available. Moreover, Chapter IV discussed the capabilities of both Photoscan and LPS to perform more efficiently when the GPS and attitude information is known for each captured image. Therefore, to speed up image processing and potentially increase the quality of the processing results, the Skate UAV and GCS can be improved to record and synchronize the GPS/attitude data for each captured image. Another system enhancement, discussed in Sections III.3.3 and III.3.5, will be to account for variable wind speeds. Wind is constantly changing, which cannot be simulated using Paparazzi's simulator. However, the autopilot does allow for waypoints to be moved in-flight, so a simple solution can be to move waypoints dynamically, forcing the UAV to change trajectory to compensate for the wind.

On top of the system enhancements just discussed, there are several algorithmic enhancements that can be made. Section III.3.4 discussed the potential need to weight the average vector when combining the geometry and wind mode vectors. Although simulations showed that geometry mode was always the best, real-world tests may produce different results and the combination mode may need to be revisited. The planning algorithm presented in Chapter III only allows flight regions that are convex polygons. The site to be mapped may not always be a convex region or surrounding landscape may not allow a path along a convex region to be followed. Therefore, the algorithm can be extended to allow non-convex polygonal regions. Furthermore, implementing autonomous take-off and landing in unknown environments is essential to the long-term goals of this research. The current system specifies take-off and landing parameters that are fine-tuned to a specific altitude. Using the system at a significantly different altitude nullifies those manually-specified parameters. Therefore, it is necessary to implement autonomous methods that can adapt to environmental changes. Lastly, photogrammetry does not necessarily mean mapping an entire site. It may be beneficial to generate 3D models of specific structures for use in computer simulations or model reconstructions. That application will require an entirely different path planning algorithm allowing a UAV

to capture every geometric intricacy of the structure.

BIBLIOGRAPHY

Aber, J. S., Aber, S. W., and Pavri, F. (2002). Unmanned small-format aerial photography from kites for acquiring large scale, high-resolution, multiview-angle imagery. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Pecora 15/Land Satellite Information IV/ISPRS Commission I/FIEOS*, Denver, CO, USA.

Agisoft (2012). Agisoft photoscan. http://www.agisoft.ru/products/photoscan/.

American Institude of Aeronautics and Astronautics (2004). Committee of standards "lexicon of UAV/ROA terminology".

Anderson, R. (1980). A kite supported system for remote aerial photography. *The Journal for Air Photography and Archaeology*, 4:4–7.

Aurora (2012). Aurora flight sciences. www.aurora.aero.

Baratin, L., Bitelli, G., Unguendoli, M., and Zanutta, A. (2000). Digital orthophoto as a tool for the restoration of monuments. In *International Archives of Photogrammetry and Remote Sensing. Vol. XXXIII, Part B5*, pages 62–69, Amsterdam.

Batut, A. (1890). *La Photographie Aérienne par Cerf-Volant*. Gauthiers-Villars et Fils.

Bendea, H., Boccardo, P., Dequal, S., Tonolo, F. G., Marenchino, D., and Piras, M. (2008). Low cost UAV for post-disaster assessment. In *The International Archives of the Photogrammetry, Remote Sensing and Information Sciences*, pages 1373–1380.

Bendea, H., Chiabrando, F., Tonolo, F. G., and Marenchino, D. (2007). Mapping of archaeological areas using a low-cost UAV. In *Proceedings of the CIPA International Symposium*.

Berlin Correspondent of the Scientific American (1909). Carrier pigeons as photographers. *Scientific American*, 100(27).

Bone, E. and Bolkom, C. (2004). *Unmanned Aerial Vehicles: Background and Issues*. Novinka Books, New York.

Bortoff, S. A. (2000). Path planning for UAVs. In *Proceedings of the American Control Conference*, pages 364–368.

Burkard, R. E., Deineko, V. G., van Dal, R., van, J. A. A., Veen, d., and Woeginger, G. J. (1998). Well-solvable special cases of the traveling salesman problem: A survey. *SIAM Rev.*, 40(3):496–546.

Cabecinhas, D., Naldi, R., Marconi, L., Silvestre, C., and Cunha, R. (2012). Robust take-off for a quadrotor vehicle. *IEEE Transactions on Robotics*, 28(3):734–742.

Ceccarelli, N., Enright, J. J., Frazzoli, E., Rasmussen, S. J., and Schumacher, C. J. (2007). Micro UAV path planning for reconnaissance in wind. In *Proceedings of the 2001 American Control Conference*, pages 5310–5315.

Ceraudo, G. (2005). 105 years of archaeological photography in italy (1899-2004). In Jean Bourgeois, M. M., editor, *Aerial Photography and Archaeology: A Century of Information (Archaeological Reports Ghent University)*. Academia Scientific.

Chakrabarty, A. and Langelaan, J. W. (2009). Energy maps for long-range path planning for small-and micro-UAVs. In *Proceedings of the American Institute of Aeronautics and Astronautics Guidance, Navigation and Control Conference*.

Chakrabarty, A. and Langelaan, J. W. (2010). Flight path planning for UAV atmospheric energy harvesting using heuristic search. In *Proceedings of the American Institute of Aeronautics and Astronautics Guidance, Navigation and Control Conference*.

Chiabrando, F., Nex, F., Piatti, D., and Rinaudo, F. (2011). UAV and RPV systems for photogrammetric surveys in archaeological areas: Two tests in the piedmont region (italy). *Journal of Archaeological Science*, 38:697–710.

Collins, R. (1996). A space-sweep approach to true multi-image matching. In *IEEE Computer Vision and Pattern Recognition*, pages 358–363.

De Filippis, L., Guglieri, G., and Quagliotti, F. (2012). Path plannig strategies for UAVs in 3D environments. *Journal of Intelligent & Robotic Systems*, 65:247–264.

Dorffner, L., Kraus, K., Tschannerl, J., Altan, O., Kulur, S., and Toz, G. (2000). Hagia sophia—photogrammetric record of a world cultural heritage. In *International Archives of Photogrammetry and Remote Sensing. Vol. XXXIII, Part B5*, pages 172–178, Amsterdam.

Eisenbeiss, H. (2008). The autonomous mini helicopter: A powerful platform for mobile mapping. In *The International Archives of the Photogrammetry, Remote Sensing and Information Sciences*, pages 977–984.

Eisenbeiss, H. (2009). *UAV Photogrammetry*. PhD thesis, Institut für Geodäsie und Photogrammetrie Eidgeössische Technische Hochschule Zürich.

Eisenbeiss, H., Lambers, K., Sauerbier, M., and Li, Z. (2005). Photogrammetric documentation of an archaeological site (Palpa, Peru) using an autonomous model helicopter. In *Proceedings of the CIPA International Symposium*, Torino, Italy.

Eisenbeiss, H. and Sauerbier, M. (2011). Investigation of UAV systems and flight modes for photogrammetric applications. *The Photogrammetric Record*, 26(136):400–421.

Eisenbeiss, H. and Zhang, L. (2006). Comparison of DSMs generated from mini UAV imagery and terrestrial laser scanner in a cultural heritage application. In *Proceedings of the ISPRS Commission V Symposium on Image Engineering and Vision Metrology*, pages 90–96.

Elfes, A. (1987). Sonal-based real-world mapping and navigation. *IEEE Trans. on Robotics and Automation*, RA-3(3):249–265.

ESRI (2012). ArcGIS. http://www.esri.com/software/arcgis.

Eynard, D., Vasseur, P., Demonceaux, C., and Frémont, V. (2012). Real time UAV altitude, attitude and motion estimation from hybrid stereovision. *Autonomous Robots*, 33:157–172.

Fu, L. and Zhu, K. (2012). UAV path re-planning of multi-step optimization based on LRTA* algorithm. In Deng, W., editor, *Future Control and Automation*, volume 173 of *Lecture Notes in Electrical Engineering*, pages 163–169.

Furukawa, Y. and Ponce, J. (2010). Accurate, dense, and robust multiview stereopsis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(8):1362–1376.

Gallup, D., Frahm, J.-M., Mordohai, P., Yang, Q., and Pollefeys, M. (2007). Real-time plane-sweeping stereo with multiple sweeping directions. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8.

Google (2012). Google earth. http://www.google.com/earth/index.html.

Guerrero, J. A. and Bestaoui, Y. (2012). UAV path planning for structure inspection in windy environments. *Journal of Intelligent & Robotic Systems*, pages 1–15.

Harwin, S. and Lucieer, A. (2012). Assessing the accuracy of georeferenced point clouds produced via multi-view stereopsis from unmanned aerial vehicle (UAV) imagery. *Remote Sensing*, 4:1573–1599.

Houle, M. E. and Toussaint, G. T. (1988). Computing the width of a set. *IEEE Transactions Pattern Analysis & Machine Intelligence*, 10:761–765.

Hrabar, S. (2012). An evaluation of stereo and laser-based range sensing for rotocraft unmanned aerial vehicle obstacle avoidance. *Journal of Field Robotics*, 29(2):215–239.

Huang, H.-M. (2008). Autonomy levels for unmanned systems (ALFUS) framework volume i: Terminology version 2.0. Contributed by the Ad Hoc ALFUS Working Group Participants, NIST Special Publication 1011-I-2.0, NIST.

Intergraph (2012). LPS. http://geospatial.intergraph.com/products/LPS/LPS/Details.aspx.

Jun, M. and D'Andrea, R. (2003). Path planning for unmanned aerial vehicles in uncertain and adversarial environments. In Butenko, S., Murphey, R., and Pardalos, P. M., editors, *Cooperative Control: Models, Applications and Algorithms*, chapter 6, pages 95–111. Springer.

Kendoul, F. (2012). Survey of advances in guidance, navigation, and control of unmanned rotocraft systems. *Journal of Field Robotics*, 29(2):315–378.

KestrelMeters (2012). Kestrel 4500 weather meter. http://www.kestrelmeters.com/products/kestrel-4500-weather-meter.

Klein, G. and Murray, D. (2007). Paallel tracking and mapping for small AR workspaces. In *International Symposium on Mixed and Augmented Reality*, pages 225–234, Nara, Japan.

Korf, R. (1990). Real-time heuristic search. *Artificial Intelligence*, 42:189–211.

Lambers, K., Eisenbeiss, H., Sauerbier, M., Kupferschmidt, D., Gaisecker, T., Sotoodeh, S., and Hanusch, T. (2007). Combining photogrammetry and laser scanning for the recording and modeling of the late intermediate period site of pinchango alto, palpa, peru. *Journal of Archaeological Sciences*, 34.

Langellan, J. W. and Bramesfeld, G. (2008). Gust energy extraction for mini-andmicro-uninhabited aerial vehicles. In *Proceedings of the American Institute of Aeronautics and Astronautics 46th Aerosciences Conference*, Reno, Nevada.

Lin, L. and Goodrich, M. A. (2009). UAV intelligent path planning for wilderness search and rescue. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 709–714.

Liu, X., Chen, P., Tong, X., Liu, S., Liu, S., Hong, Z., Li, L., and Luan, K. (2012). UAV-based low-altitude aerial photogrammetric application in mine areas measurement. In *Second International Workshop on Earth Observation and Remote Sensing Applications*.

Magick, I. (2012). Image magick. http://www.imagemagick.org/script/index.php.

Meier, L., Tanskanen, P., Heng, L., Lee, G. H., Fraundorfer, F., and Pollefeys, M. (2012). PIXHAWK: A micro aerial vehicle design for autonomous flight using onboard computer vision. *Autonomous Robots*, pages 21–39.

Mejias, L., Usher, K., Roberts, J., and Corke, P. (2006). Two seconds to touchdown—vision-based controlled forced landing. In *International Conference on Intelligent Robots and Systems*, pages 3527–3532, Beijing, China.

Microsoft® (2012). Photosynth. http://photosynth.net/.

Moravec, H. and Elfes, A. (1985). High resolution maps from wide angle sonar. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, St. Louis, MO.

Mozas-Calvache, A., Pérez-García, J., Cardenal-Escarcena, F., Mata-Castro, E., and Delgado-Garcí, J. (2012). Method for photogrammetric surveying of archaeological sites with light aerial platforms. *Journal of Archaeological Science*, 39:521–530.

Newhall, B. (1969). *Airborne Camera: The World from the Air and Outer Space*. Hastings House.

Nietzel, F. and Klonowski, J. (2011). Mobile 3D mapping with a low-cost UAV system. In *Proceedings of the International Society for Photogrammetry and Remote Sensing Conference on Unmanned Aerial Vehicle in Geomatics*, Zurich, Switzerland.

Obermeyer, K. J., Oberlin, P., and Darbha, S. (2012). Sampling-based path planning for a visual reconnaissance UAV. *AIAA Journal of Guidance, Control and Dynamics*, 35(2):619–631.

Osborne, J. and Rysdyk, R. (2005). Waypoint guidance for small UAVs in wind. In *Proceedings of the American Institude of Aeronautics and Astronautics Infotech@Aerospace Conference*, Arlington, VA.

Paparazzi (2012). Paparazzi the free autopilot. http://paparazzi.enac.fr/wiki/Main_Page.

Remondino, F., Barazzetti, L., Nex, F., Scaioni, M., and Sarazzi, D. (2011). UAV photogrammetry for mapping and 3D modeling – current status and future perspectives –. In *Proceedings of the International Society for Photogrammetry and Remote Sensing Conference on Unmanned Aerial Vehicle in Geomatics*, Zurich, Switzerland.

Rinaudo, F., Chiabrando, F., Lingua, A., and Spanò, A. (2012). Archaeological site monitoring: UAV photogrammetry can be an answer. In *The International Archives of the Photogrammetry, Remote Sensing and Information Sciences*, pages 583–588.

Saripalli, S., Montgomery, J., and Sukhatme, G. (2002). Vision-based autonomous landing of an unmanned aerial vehicle. In *International Conference on Robotics and Automation*, pages 2799–2804, Washington, DC.

Sauerbier, M. and Eisenbeiss, H. (2010). UAVs for the documentation of archaeological excavations. In *Proceedings of the International Society for Photogrammetry and Remote Sensing Commission V Symposium*, Tyne, UK.

Sauerbier, M., Siegrist, E., Eisenbeiss, H., and Demir, N. (2011). The practical application of UAV-based photogrammetry under economic aspects. In *Proceedings of the International Society for Photogrammetry and Remote Sensing Conference on Unmanned Aerial Vehicle in Geomatics*, Zurich, Switzerland.

Scherer, S., Rehder, J., Achar, S., Cover, H., Chambers, A., Nuske, S., and Singh, S. (2012). River mapping from a flying robot: State estimation, river detection, and obstacle mapping. *Autonomous Robots*, 33:189–214.

Skarlatos, D., Theodoridou, S., and Glabenas, D. (2004). Archaeological surveys in greece using radio-controlled helicopter. In *Proceedings of the FIG Working Week*, Athens, Greece.

Stolle, S. and Rysdyk, R. (2003). Flight path following guidance for unmanned air vehicles with pan-tilt camera for target observation. In *Digital Avionics System Conference*.

Thrun, S. (1998). Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99:21–71.

Trimble (2012). Trimble - survey - total stations. http://www.trimble.com/survey/Total-Stations.aspx.

Verhoeven, G. J. J. (2009). Providing an archaeological bird's-eye view – an overall picture of ground-based means to execute low-altitude aerial photography (LAAP) in archaeology. *Archaeological Prospection*, 16:233–249.

Wallace, L., Lucieer, A., Watson, C., and Turner, D. (2012). Development of a UAV-LiDAR system with application to forest inventory. *Remote Sensing*, 4:1519–1543.

Weinhaus, F. (2012). Fred's imagemagick scripts. http://www.fmwconcepts.com/imagemagick/defisheye/index.php.

Weiss, S., Scaramuzza, D., and Siegwart, R. (2011). Monocular-SLAM-based navigation for autonomous micro helicopters in gps-denied enviornments. *Journal of Field Robotics*, 28(6):854–874.

Wernke, S. (2012). Steve Wernke's webspace. http://www.vanderbilt.edu/wernke/.

Whittlesey, J. (1970). Tethered balloon for archaeological photos. *Photogrammetric Engineering*, 36:181–186.