A L I S

Aerial Land Inspection System

Vermeer Corporation

Project Plan

Team Members

Brian Gillenwater

Nathan Kent

Quinn Murphy

Bryce Poellet

Jonathan Schlueter

Revision: 3

Date: 12/13/2015

Table of Contents

List of Figures and Tables

1 Introduction

1.1 Purpose

The Aerial Land Inspection System (ALIS) is a solution to remotely and autonomously map the terrain of a future work site from the air. Once generated, a 3D terrain map will allow people to view the terrain interactively through a head-mounted display. This document introduces the proposed design of ALIS developed by the team.

1.2 Scope

When given a location and area, ALIS will be responsible for generating and executing a quadcopter flight path. From this flight path, ALIS will capture images to generate a 3D model of the terrain. This model will be uploaded into a game engine that allows exploration with a head-mounted display. The user is responsible for knowing the area and size of the desired location. The documentation provided here details the design decisions made while developing the entire ALIS system. These details include: the system description with use cases, constraints, and assumptions; functional requirements and nonfunctional requirements; and the top-level system design.

1.3 Definitions

Quadcopter - A rotorcraft with four propellers.

Photogrammetry - The use of multiple two-dimensional photographs of a subject to reconstruct the third-dimension.

Virtual Reality System - A piece of multimedia used to simulate physical presence in a virtual environment.

Oculus Rift - A head-mounted display that renders stereoscopic images to produce a 3D environment.

LIDAR - A method of measuring distance by using lasers and analyzing the reflected light.

1.4 Acronyms

ALIS - Aerial Land Inspection System

UAV - Unmanned Aerial Vehicle

HMD - Head-Mounted Display, often used for virtual reality

GUI - Graphical User Interface

FAA - Federal Aviation Administration

1.5 Problem and Needs

As it currently stands, examining a job site is an expensive and time-consuming task. It requires sending someone out to the area and having them take notes and photographs. This process can potentially involve several days' worth of time and work. By finding a way to automate this process, the cost of inspecting an area will dramatically decrease and the ability to view the designated area will not be limited to the individuals physically sent to the work site.

1.6 Overview

ALIS is intended to enable the mapping of a worksite remotely. To accomplish this, a quadcopter equipped with a high resolution camera is sent out to autonomously take pictures of the worksite from the air. By taking enough images, photogrammetry techniques can be applied to the images to generate a three-dimensional model. This model can then be sent to a virtual reality system allowing people to examine the worksite without needing to travel there. The goal is that ALIS will require very little user interaction and will operate under its own control as much as possible.

1.7 Deliverables

The end result of this project will be the prototype of a system that enables the remote mapping of worksites and will be composed of three main parts:

1. A drone capable of autonomously traversing the designated work area, capturing photographs as needed.
2. A piece of desktop software that allows the user select the area they wish to designate as the work area.
3. A pipeline to convert the captured photos into a three-dimensional model that can be viewed inside of a virtual reality system

2 Overall Description

2.1 Functions

- Specification of the mapping location through an intuitive and GUI-based user application
- Automatic generation of a flight path for the quadcopter based upon the user-specified mapping area
- Automated flight of the quadcopter including travelling to the mapping location, photographing the mapping location, and flying back to the starting location
- Automated transfer of photographs of the mapping location from the quadcopter to a workstation
- Minimal user interaction to generate a three-dimensional model reconstruction of the mapping location from the photos taken by the quadcopter
- Simple importation of the model into a virtual reality engine
- Ability to walk/fly through the mapped space in the virtual reality engine, while wearing HMD (such as Oculus Rift)

2.2 Constraints

- The FAA limits the altitude and locations at which Unmanned Aircraft Systems can fly to 400 feet and at least 5 miles from an airport respectively
- Photogrammetry software tends to require a large number of photos with a significant amount of overlap and detail
- Significant time is required to render a 3D model from captured photos
- Quadcopter must fly within range of the controller (~2 km)
- Quadcopter battery only lasts ~23 minutes
- The DJI Phantom 3 Advanced SDK must be run on either Android or iOS

2.3 Assumptions

- There will be sufficient variation in the terrain for the photogrammetry software to accurately construct a model
- The end user will have access to the internet to load the mapping solution
- The area to be modeled is of a size that can be fully scanned both within a single flight and without moving the controller/operator
- No obstacles will interfere with the flight of the quadcopter
- The PC application will be able to communicate over a standard computer network with the Android application
- The PC application will run on a powerful workstation with multiple CPU cores and a dedicated Nvidia graphics card.

3 Requirements

3.1 Functional Requirements

- The system must be able to support sustained flight for at least 20 minutes, under all flyable weather conditions
- The quadcopter will be able to fly a maximum distance of ½ a mile from the controller/operator
- The system will take 70+ images, each with >50% overlap, to maximize model accuracy and minimize unneeded image information
- Photogrammetry software will generate a model in under 6 hours
- Generated model must be natively supported by a virtual reality environment

3.2 Non Functional Requirements

- User interaction with the system shall be simple and error free.
- A clear and accurate model will be generated, regardless of terrain.
- The device shall work in clear and windy environments (< 20 mph).
- The system shall not crash or become unstable during flight.
- The model will be viewable in a virtual reality platform.
- The system will scale altitude and number of images based on given map area
- After flight plan is generated and started, minimal user intervention will take place
- The system will be safe to operate, and include emergency stop or halt controls.

4 Design

4.1 Market and Literature Survey

4.1.1 UAV Selection

Throughout our planning phase, we looked at multiple different UAV solutions to find one that had a programmable interface (which includes flight controls and the ability to take pictures programmatically), a long flight time, and good control algorithms. We looked at the following:

- Parrot Bebop
- Lumenier QAV250 with OpenPilot Control System
- DJI Matrice-100
- DJI Phantom 3 Advanced

The Bebop was lightweight and had a programmable SDK, but the camera was not as configurable as we wanted to take different types of photos. The QAV250 would have been nice, but the community that runs OpenPilot was going through some trouble at the time, and it was impossible to acquire the flight controllers. The Matrice-100 met all of our criteria, though it was too expensive to justify. We ended up selecting the DJI Phantom 3 Advanced because it allowed us to program waypoints for the quadcopter to automatically fly to, the camera was on a gimbal that can be rotated any way we want, we could fly for more than 20 minutes, and DJI's control algorithms are among the best in the industry.

4.1.2 Photogrammetry Software Selection

For a short amount of time, we looked at a LIDAR solution for generating a 3D model, which would give us a model shortly after the flight with minimal computation. However, we dismissed this solution early on because we would still have to take photos in order to colorize the model and the resolution would be fairly low from the heights that we would be scanning from.

We ended up deciding on a technique called photogrammetry that takes a bunch of pictures from different perspectives, analyzes them, and generates a 3D model and corresponding texture from them. This solution does take longer to analyze compared to the LIDAR solution, though it will give us much better accuracy and allows us to scan a larger range because we can spend less time at each location scanning and just grab a picture instead. We initially looked into three different photogrammetry software solutions:

- Pix4D
- Ames Stereo Pipeline
- VisualSFM combined with CMP-MVS

Pix4D is the ideal solution, as it is enterprise-level software with UAV environment mapping in mind. However, this software is thousands of dollars and outside of our budget. The Ames Stereo Pipeline is developed by NASA's Ames Research Center, and is completely free, open source, and can run on servers in parallel to allow for future acceleration. However, this software does not natively run on Windows and we would have to spend some fair amount of time to modify it so that it does. Additionally, the software is a research project, and thus doesn't have the greatest user experience. We ended up picking VisualSFM and CMP-MVS as, while they are not open source and cannot run on multiple servers in parallel, they are free, have nice user interfaces, and natively runs on Windows.

Since our initial comparison of photogrammetry solutions, we have discovered that the author of CMP-MVS has started a company called Capturing Reality. Our initial assessments have found that their product, RealityCapture, is far superior to our current solution: it runs faster, produces better models, and can be run on multiple servers in parallel. However, there is little licensing information available. We have contacted them to enquire about the possible licenses but have not yet received an answer.

4.2 Operating Environment

Our Windows application will be targeting a 32-bit Windows 7 machine with a wireless adapter. Our Android application will be targeting Android SDK Level 16, which is required by the DJI Mobile SDK. Our system will be compatible with both the DJI Phantom 3 Professional and Advanced models.

4.3 System Diagram

ALIS is made up of three components. The first of these components is the ALIS Command Center, which runs on a Windows PC and is responsible for the planning of a route, transferring that route to the Android application, and then waiting for the Android application to respond with a set of pictures. The second of these components is the Android application, ALIS LITE (Location and Image Transfer Entity), a thin client which is responsible for translating the location requests from the Windows application into something that the quadcopter can read and then transferring the images from the quadcopter back to the PC. Finally is the DJI Phantom quadcopter itself, which is responsible for flying around and gathering photographs of the environment. Figure 4-1 is the system diagram for how we expect the pieces of our final solution to communicate.

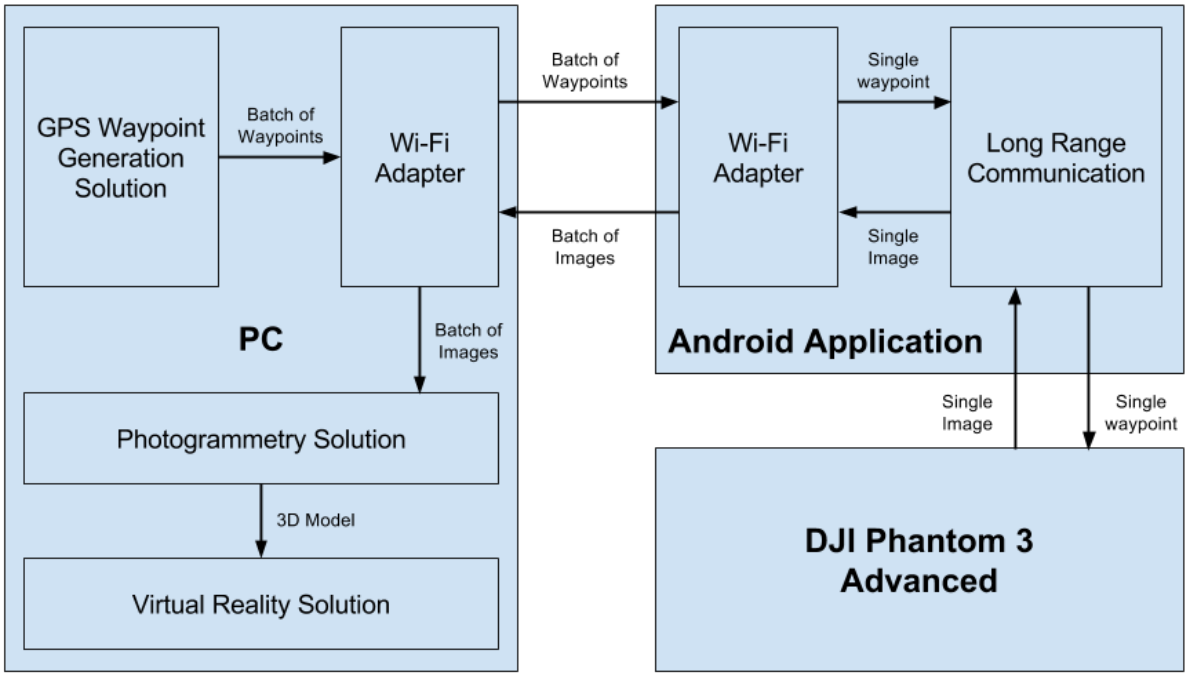

Figure 4-1: System Diagram

4.4 State Diagram

Figure 4-2 is the state diagram for the steps that ALIS will go through in order to complete its task.

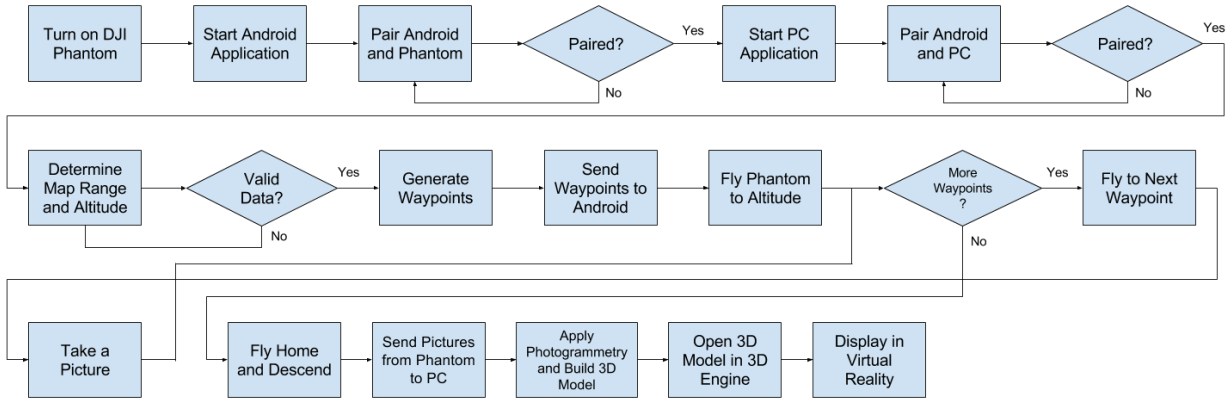

Figure 4-2: State Diagram

4.5 Photo Taking Strategy

In order to grab information about the environment from different vantage points and still have a large overlap, we will need to take a lot of pictures. Figure 4-3 shows the strategy that we will be using to take photos. The black area is the area we wish to capture, the blue circles are the aircraft, and the yellow vectors are the photos we will be taking, with an implicit vector facing downwards on each circle.

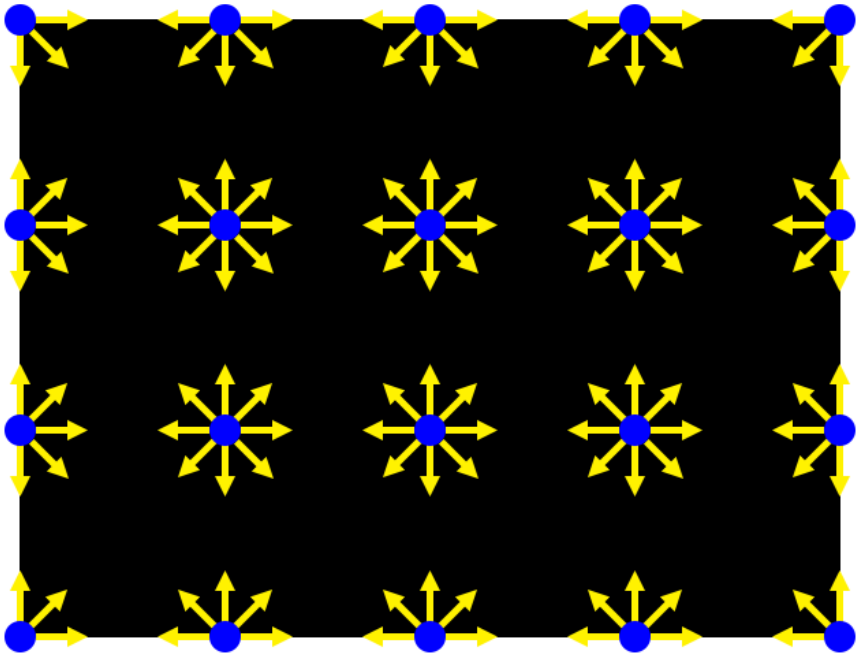

Figure 4-3: Photo Taking Strategy

4.6 Windows Application Design

Because the Android application is a thin-client, it will not have much in the way of screen sketches. It will simply report back the current status of the quadcopter and the transfer of data to and from the Windows application. Figures 4-4 and 4-5 show some screen sketches for the application.

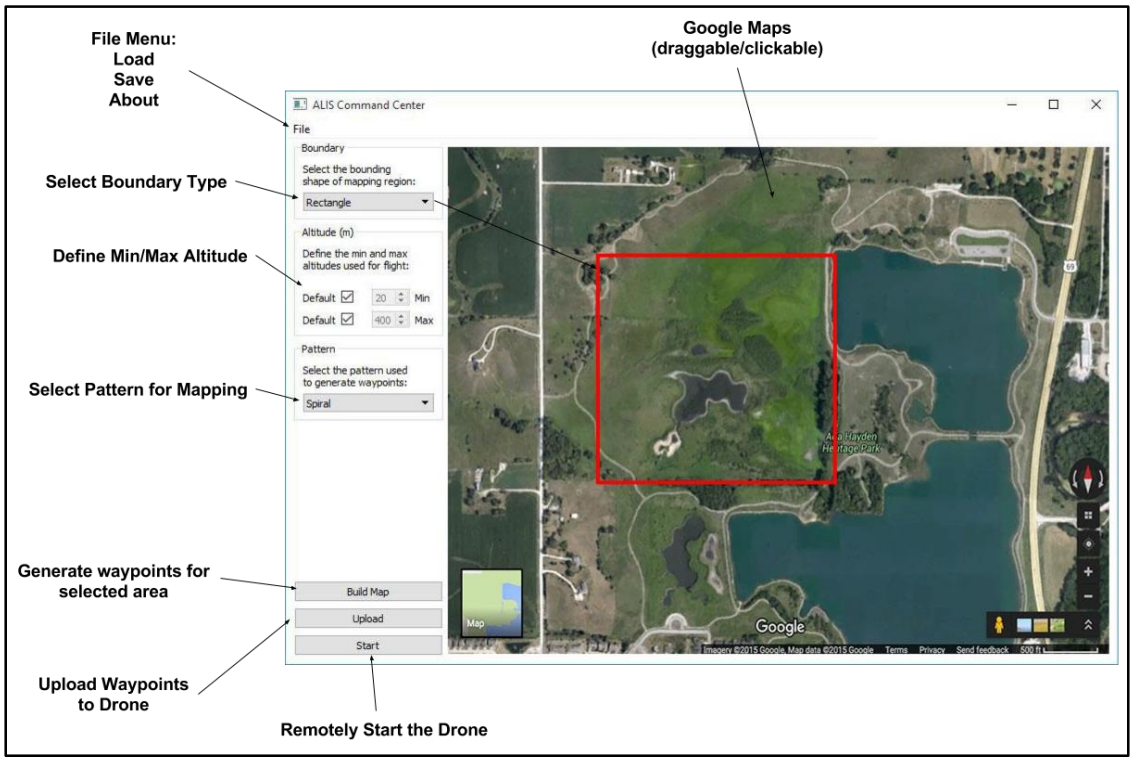

Figure 4-4: Windows Application Main Window

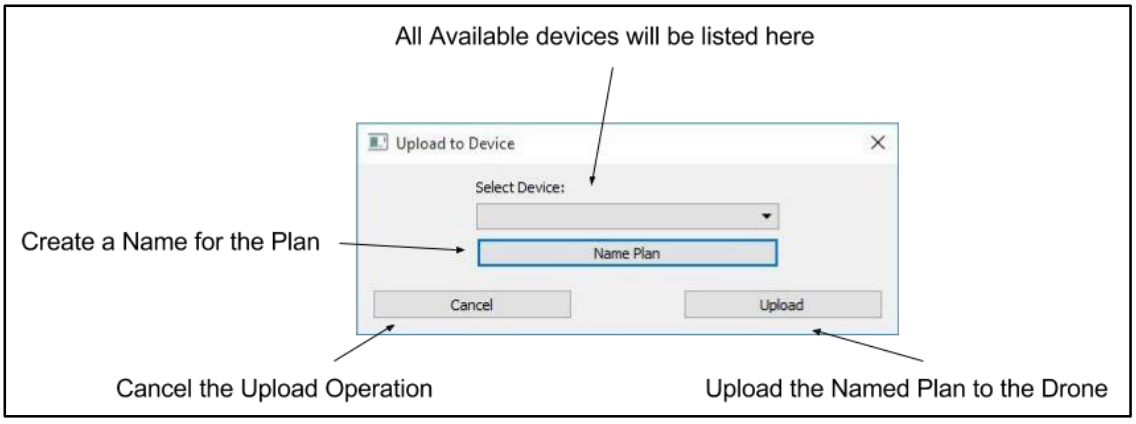

Figure 4-5: Windows Application Device Upload

5 Project Management

5.1 Cost Estimate

The only cost we plan on having is the cost of purchasing the quadcopter. The DJI Phantom 3 Advanced is the quadcopter we decided to use for the project and the cost of purchasing it is $1000. Currently we believe the free, open-source photogrammetry software we have found will work for the project. However, if this software does not work, we will likely need to purchase a commercial software solution which is likely to have a cost around $1000.

5.2 Work Breakdown

5.2.1 Windows Application

This part of the project is likely to require the most work. We plan to start working on the application as soon as the user-interface and quadcopter interface are determined. We expect this development will take two or three people two months to get a working version and another month to get a full prototype of the application.

5.2.2 Android Application

We expect the development of this application to be very straightforward and easy to implement. The goal is to make this application as thin as possible and we expect it to only take one person around a month to complete. If we find that this goal is optimistic, we should be able to transfer another person to the development and we have some extra time before we need a prototype to be ready.

5.2.3 Photogrammetry Pipeline

The development of this pipeline is more of an integration task to get the existing photogrammetry software working with the files transferred from the quadcopter. However, we will still need to develop and interface for the file transfer and then develop a method to run the photogrammetry software. We expect that this task will take one or two people about two months to complete. We will likely start this development later in the timeline as we will need time to determine the best photogrammetry software that works for our purpose.

5.3 Resource Requirements

5.3.1 Quadcopter

We will need a quadcopter for this project. We have selected the DJI Phantom 3 Advanced as the model we will use.

5.3.2 Android Device

Because we selected the DJI Phantom 3 Advanced as the quadcopter we will use, we will need an Android device to control the quadcopter. Ideally our project would have its own on-board flight computers, but with our limited budget and this project being a prototype, we decided that having the quadcopter rely on an external device would be fine. We are looking at using a Nexus 7 tablet for this purpose.

5.3.3 DJI Mobile SDK

We will need the DJI Mobile SDK in order for our Android device to communicate with the quadcopter. The data received from the PC will be translated into something that the quadcopter can read and all commands will be sent to the quadcopter using this SDK.

5.3.4 Photogrammetry Workstation

We will need a workstation to perform the photogrammetry on to get the three-dimensional reconstruction. The reconstruction can take many hours to complete, and so we will want a machine with a high-powered GPU that we can dedicate to this purpose.

5.4 Project Timeline

Table 5-1: Timeline of Project

September	• Begin planning the project (develop requirements, system architecture, and determine constraints) • Perform research into the options for different parts of the project such as quadcopter models and photogrammetry software
October	• Purchase quadcopter • Plan Android application and the Windows application • Begin development of the Windows application
November	• Begin work on the Android application • Work with quadcopter and photogrammetry software to find the mapping strategy that yields the best three-dimensional model
December/January	• Prototype of the autonomous quadcopter control including both the Windows application and the Android application
February	• Prototype the transfer of images from the quadcopter to the photogrammetry software and the construction of the three-dimensional model
March	• Full prototype with the Windows application, Android application, and photogrammetry software all working
April	• Fix bugs in the solution
May	• Final project is complete

5.5 Risks

5.5.1 Photography Strategy

Currently we do not have a full understanding of the best method for taking photographs to use for our projects. We have performed research into how other groups have done similar projects and developed plans based upon the result of the research, but until we are able to test our approaches by taking photographs with the quadcopter and seeing the result of running those photographs through the photogrammetry software we will be unable to know if we are on the right track. We plan to mitigate this risk by purchasing a quadcopter as soon as possible and testing the different methods we have developed to determine which one will work the best for our project.

5.5.2 Photogrammetry Software

Similar to the photography strategy, we are not familiar with any photogrammetry software, yet we will need to select one to use in the project. We have performed research into the different options available and tested them out on the ground, but until we have the quadcopter we will be unable to determine if the selected software will work in our use case. We will mitigate this risk in a similar way to the photography strategy; we will fly the quadcopter as soon as we have it available and run the captured photographs through the selected photogrammetry software to determine if it will work for our purposes. If we decide that the selected software does not work, we should still have time to choose a different application to use.

6 Conclusion

The final goal of this project is to develop a system that uses a quadcopter to autonomously map a remote worksite. We plan to have as little user-interaction as possible, and limit that interaction to the user specifying a worksite to map and starting the process. The product of the mapping will be a three-dimensional model of the worksite that can be imported into a virtual reality system to allow users to examine the worksite. By creating this virtual worksite, we will allow people to examine the worksite without travelling to the physical location which could potentially be in a very remote area that is difficult to access.