ALIS

Aerial Land Inspection System

12

Vermeer

Dr. Philip Jones and Dr. Joe Zambreno

Kit Kohl - Communication Leader I

Thiem Nguyen - Communication Leader II

Anthony Khounlo - Key Concepts Holder I

Caleb Redman - Key Concepts Holder II

Christopher Smith - Webmaster

Joseph Randazzo - Team Leader

Revised: 11-2-16/1.0

http://may1712.sd.ece.iastate.edu/

All emails for the team may be sent to:
ecpemay1712@gmail.com

Contents

1 Introduction

1.1 Project statement

The Aerial Land Inspection System (ALIS) is a solution to remotely and autonomously map the terrain of a potential work site from the air by taking several aerial photos and later stitching them into a Google Maps-like program. This document introduces the proposed design of ALIS developed by the second development team.

1.2 Purpose

The work done here will be used in an ALIS application that can be used onsite to survey an area using a drone connected with the mobile device, which has the application installed. The drone will be able to survey autonomously and collect photos to produce a 2-dimensional map of the area. This map can be used in the field and cross-referenced with other materials and tools to aid the workers in understanding the area.

1.3 Goals

We hope to create a working application that will utilize and add onto the previous version of the ALIS project. The project will be reworked to autonomously capture photos and then stitch them together to make a 2D map similar to Google Maps. Some of our research and development goals are to learn more about GIS and OpenLayers, then adeptly apply them to this project.

- ➔ Addition to the existing/previous ALIS project
    - ◆ Improve existing Android application by adding features for customization to waypoint generation, reduce in-app crashes, more user friendly with easier to understand instructions
    - ◆ Autonomously capture photos at specific areas on a map using a DJI Phantom 3 Advanced drone which is determined by an Android device and application
    - ◆ Stitch the photos captured by the drone to create a 2D map similar to Google Maps, all generated on the Android device and application
- ➔ Research and Development Goals
    - ◆ Learn more about GIS and existing GIS applications, mainly OpenLayers. Also research into integrating the GIS application with Android
    - ◆ Learn more about the DJI SDK for Android, communication between an Android device and the DJI Phantom 3 Advanced drone

2 Deliverables

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

1. An autopilot survey with the drone using the Android Application
2. The resulting pictures stitched together in a GIS application, preferably OpenLayers
3. A high quality 2D map of the area photographed by the drone with an offline mode

3 Design

We decided as a group to split up into smaller teams to split the problem into two parts. One team's goal is to provide a functioning automated drone with information sent from a mobile application. The other team's goal is to provide an OpenLayers application that would take pictures and stitch maps together.

3.1 System specifications

3.1.1 Non-functional

List and explain the non-functional requirements of the project.

- Maps are generated in an automated fashion.
- Map can be shown using OpenLayers.
- Map can be viewed while not connected to the internet (will still need GPS/Location).
- Map will be accurate down to ten meters when referenced with other material
- Application is easy to use and responsive (buttons/options are readable 2' from device, actions take less than 2 seconds to complete).
- The drone can work in fair weather consistently (under 20 mph wind, no rain/snow).

3.1.2 Functional

- Generate 2D map using the Android application on the device
- Three pictures will be taken at each waypoint by the drone. (This is done as a redundancy measure to better ensure a high-quality image at each waypoint.)
- One picture will be used for each fly-by waypoint in map stitching
- Waypoint generation can be set by location of device or by touch location (selecting a place on the device's map in application)
- Drone flies autonomously after launch from application
- Failsafe measures by having a button on the application to send the drone back to the launch location at anytime within the automation flight

3.2 Proposed Design/Method

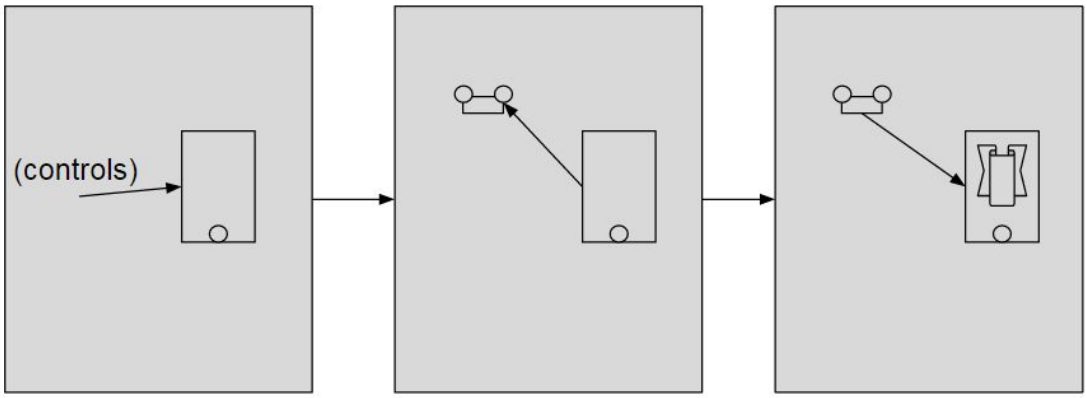
Our team has decided to implement a mobile application that will send information to a DJI Phantom 3 Advanced drone and retrieve pictures from it. The drone will receive 2 GPS coordinates and autonomously fly over the specified area and take pictures at automatically generated waypoints.

We also decided to implement an OpenLayers web application which we will use to put together the images we collect from the drone and display them as a map. After getting these two parts to work separately we are hoping to integrate our OpenLayers solution into the mobile application in order to have one single process.

3.3 Design Analysis

The team working on the functioning automated drone has modified the code provided by Vermeer that sends waypoint information to the drone from an Android application. The existing waypoints were to facilitate the creation of a 3D image and it has been changed to increase the rate at which a mission can be accomplished for a 2D map. The team also reworked the UI so that waypoints can be centered around the user or around a location chosen by the user. As of now the drone can perform a mission taking a single picture straight down at each waypoint.

The team working on the Map stitching team did a research survey on the existing technologies that are available to do 2D map stitching. They reviewed several programs and decided that OpenLayers would be the best fit for this project. They have managed to provide an OpenLayers application onto our website and uploaded images to the application. The Map stitching team has implemented and tested uploading a single image to OpenLayers and having that image appear on the screen. The team has also tried to upload multiple pictures but could not get the images to appear. The drone team has tested and succeeded in sending waypoint information from a mobile device to the drone.



First, the phone is loaded with data for the drone, on/off-site.

The drone autonomously flies and scouts the area, taking pictures of the ground in a series of geotagged, altitude-tagged images.

The pictures are stitched onto an empty map on the phone, with waypoints in a higher quality.

4 Testing/Development

4.1 Interface Specifications

The hardware for the project is the DJI Phantom 3 Advanced. It interfaces with the RC that attaches to a smartphone that will be running the application. Our Java application will use the DJI API to interface with the drone hardware.

4.2 Hardware/Software

Hardware:

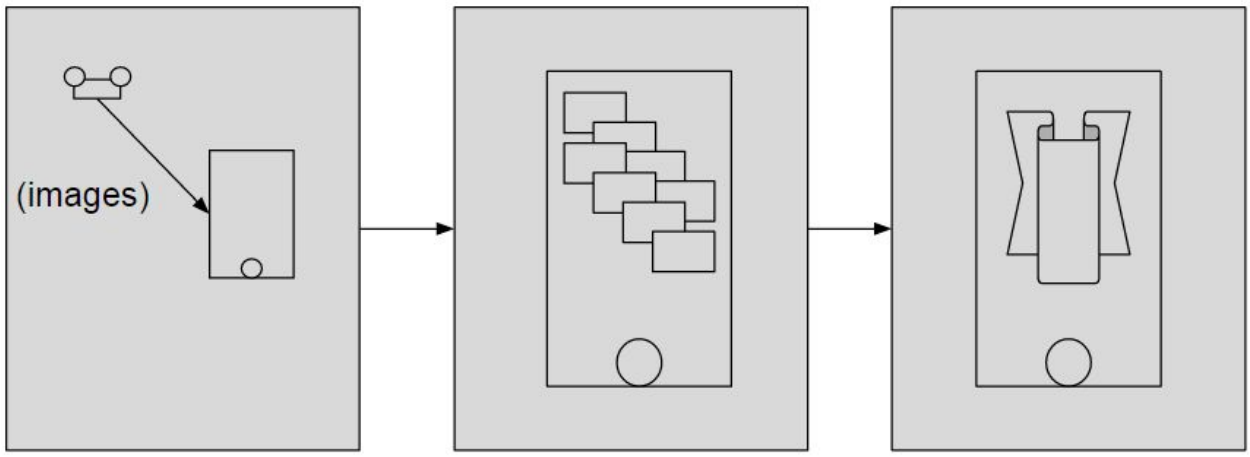
- Android Mobile Device - The Android mobile device is used to send and receive information from the DJI Phantom 3 Advanced drone. Allowing users to send waypoints to the drone and potentially receiving pictures to generate a 2D map in an unmapped location.
- DJI Phantom 3 Advanced Drone - The DJI Phantom 3 Advanced Drone will take pictures at waypoints specified by the user from the Android Mobile Device, transfer pictures taken to the Android Mobile Device, and fly autonomously after initial launch with device.

Software:

- OpenLayers - Software used for stitching the 2D map from the pictures and geolocations
- Chrome/Firefox Developer Tools - Software used to debug Javascript and analyze HTML elements.

- Android Studio - Allows for development and debugging of Java code for the Android platform.

4.3 Process



The phone receives the images from the quadcopter.

The images are sent in a queue to be attached to the empty offline map, modified by geolocation and altitude.

The map is redrawn and shown on the phone with the new images.

The picture above is the basic overview of our testing. At first, we plan to test our 2D stitching and drone communication separately. By isolating these two methods, we can easily determine where the bugs are located and to prevent any merge errors. After merging, we would know the issues lies within our merging rather than our mobile to drone communications or our 2D map stitching.

We have to test our drone communication using the DJI Phantom 3 Advanced drone provided by Vermeer and our own mobile devices. By making sure that each of our Android Mobile devices can communicate with the drone, we can ensure that most users should be able to do the same. Google Map is able to be generated in application with all devices using the hard-coded Google Maps API Key, the map should be able to generate as well without network data and only with GPS/Locations on.

To test our OpenLayers application, we will use the Developer Tools from Chrome/Firefox. This will allow us to dynamically test and edit our Javascript code and verify that everything in our HTML page is as expected. We will also be using OpenLayers' access to OpenStreetMap to check that our GPS positioning is correct. This is done by overlaying our images on the OpenStreetMap layer.

5 Results

So far both teams have had some success. The drone can fly autonomously and take pictures and the application team has implemented the creation of waypoints on touch. We still are experiencing unknown application crashes that we plan to work out. We've learned that the DJI API and the Google Maps Android API are fairly intuitive and easy to work with. We plan to incorporate the use of Fabric.io which was mentioned by one of our Vermeer contacts to help root out crashes.

6 Conclusions

So far the drone can autonomously fly and take pictures. The code from the previous team has been inspected, edited, and integrated into our current work. Research has been done on what technologies are available to do map stitching and help with the project goals. The overall goal is to create an application that will direct the drone to autonomously take pictures of a given area and then stitch those images into a 2D map using a map stitching software. The stitching should be done offline. Our plan to accomplish these goals is to split into two teams (which we have done), and have one team focus on the mobile application and the other on map stitching software. Once each team is confident in the functionality of what they're working on then the teams will merge their work into a single project/codebase. Once that is complete the finer details will be fixed and the project will move into testing phase. The testing phase should last 1-2 months and will cover field tests, performance, and correctness.

7 References

- OpenLayers 3 API
- DJI Phantom 3 Advanced
- DJI SDK
- Android Developer Guide
- Google Maps Android API
- Android Studio User Guide
- Git ECE IASTATE - GitLab
- Fabric.io
- Vermeer
- Phillip Jones
- Joe Zambreno
- Quinn Murphy

8 Appendices