Development of Mobile Applications for Crop Scouting with Small Unmanned Aircraft Systems

by

Shubh Chopra

B.E., Rajiv Gandhi Proudyogiki Vishwavidyalaya, India, 2015

A THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computer Science
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2017

Approved by:

Co-Major Professor
Antonio R. Asebedo

Approved by:

Co-Major Professor
Mitchell L. Neilsen

Copyright

Abstract

Small unmanned aircraft systems (sUAS) have been in commercial use since the1980's and over 8-12% of its current uses are in the agricultural sector, but only involving limited uses like surveying, mapping and imaging, which is expected to increase to 47% according to AUVSI with the association of Artificial Intelligence over the next decade. Our research is one such effort to help farmers utilize advanced sUAS technology coupled with Artificial Intelligence and give them meaningful results in a widely used and user friendly interface, like a mobile application.

The vision for this application is to provide a completely automated experience to the farmer for a repetitive and periodic analysis of his/her crops where all the instruction needed from the farmer is a push of a button on a one time configured application and ultimately providing results in seconds. This would help the farmer scout their crops, assess yield potential, and determine if additional inputs are needed for increasing grain yield and profit per acre.

For making this application we focused on user-friendliness by abstracting crop algorithms, minimized necessary user inputs, and automate the construction of flight paths. Due to internet connection not always being available at farm fields, processing was kept to on-board compute systems and the mobile device to give live results to farmers without reliance on cloud-based analytics.

The application is configured to work with DJI Aircraft using OpenCv for video processing and mobile vision, GIS and GPS data for accurate mapping, locating device, sUAS on the mobile application, and FFMPEG for encoding and decoding compressed video data. An algorithm developed by Precision-Ag Lab at the K-State Agronomy Department was implemented into the sUAS application for providing real time yield estimations and nitrogen recommendation algorithm for winter wheat.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to express my sincere gratitude to my Major Professor, Dr. Antonio Asebedo and Co-Major Professor Dr. Mitchell Neilsen for their encouragement and for trusting my abilities to complete this project.

I take immense pleasure in extending my heartfelt thanks to my committee members Dr. Doina Caragea and Dr. Pavithara Prabhakar for their encouragement and for taking the time to serve on my committee.

I take this opportunity to acknowledge the support and help received from the academic staff of the Department of Computing and Information Sciences.

I would like to thank my parents and friends for their immense love and belief.

Chapter 1 - Introduction

1.1 Project Description

It is common for a farmer to scout their fields many times during the growing season at different growth stages in crop fields ranging in 40 to 640 acres in size. Farmers conduct crop scouting in order to find any problem areas in the field, estimate yield, and estimate additional nutrient requirements such as Nitrogen (N), which can be very costly. This small unmanned aircraft system (sUAS) crop scouting iOS application is precisely designed for farmers in helping them to achieve these tasks with the help of sUAS connected to an iPhone via its Remote Controller. This application provides an interface to the user for configuring, monitoring and validating the results of the survey done by the sUAS in real time. This would help the farmers to survey their fields by sitting in the comfort of their home within a very short time and over a vast span of area with a better field view from the top, on a live feed from the camera.

Users can configure field boundary, flight path, waypoint distance, camera features and other sUAS actions for better filtering crop accuracy. Users have access to map view, camera view and configuration view to monitor the sUAS and users can validate the current GPS position and other attributes of flight on map view and image filtering results in the camera view.

This application is based on following modules, DJIMobileSDK, for controlling the sUAS position, missions, camera functions, and viewing sUAS location on the application. Opencv with its wrapper filters the image frame. The H264 decoder is for decoding the live feed in H.264 format to YUV frames. The mobile file storage is for saving the shape file and other configuration settings. Yield estimation and N recommendation algorithms are for calculating the estimated yield potential and nitrogen requirement based on the image filtering value. Initial testing and validation of this application was conducted on Winter wheat (*Triticum aestivum*).

1.2 Motivation

While working as a full-time intern with agronomists in the K-State Agronomy department and working with PrecisionAg lab on a mobile application for Kansas Wheat, I was introduced to Techno-Agriculture sector and came to know about the problems faced by farmers. The PrecisionAg lab have been working on remote sensing research utilizing sUAS for years and came very far in developing the algorithms on sUAS imagery for quickly giving results to help farmers, but there was the issue of global reach and accessibility to this information in Kansas and around the world. Therefore, we decided to make a mobile application, which can help implement these algorithms with instructions to the sUAS in a simple and easy mobile application.

Chapter 2 - Related Work

2.1 DJI GO

This application is compatible with all DJI Aircrafts and have all the customizable functions from DJISDK. It has many more flight plans such as, follow me, point of interest and home lock. You can record videos and shoot pictures from your device. But this application does not help the farmer with doing any analysis on the data captured and is harder to use for an average farmer because it has a lot of functions that are not required for field scouting.

2.2 Data Mapper Inflight

This application is made for agriculture purposes like counting the number of plants, scout view, mosaic image of the field, and delivering the image output in different formats. This application’s biggest drawback is that it is only available in the Android platform and it uses the cloud to map and survey the results. Internet connectivity is required and farmer will have to pay for the cloud services along with the application. The application only provides a raw data and provides no recommendation or analysis on the data.

2.3 Drone Deploy

This application works on the same principle as the Data Mapper Inflight and is also available in iOS platform and can perform additional features. The additional features include, shapefile and contour export, 3D mapping and export etc, along with providing very few analysis information on the data collected. It is very costly for an average farmer to spend money for just mapping the field.

Chapter 3 - Requirement Analysis

We gathered the requirements for this application, keeping in mind the practical use to user, for example, the farmers having huge fields, low internet connectivity, needing easy to use application, efficiency and needing the application to be robust. These requirements that we discussed, based on the experience of Dr. Antonio Asebedo and the students of PrecisionAg, over years of testing on different growth stages of the crop. Then, the functionalities of the application were discussed. We got ideas from the currently used sUAS application and tried to simplify it by adding only the features that were required. We then discussed the structure of the application. The requirements have been tweaked and refined after many iterations as we progressed.

The requirements are classified in to two categories, functional and non-functional requirements.

3.1 Functional Requirements

Functional requirements are the requirements which relate to the functional or behavioral aspect of the application. These requirements are visible to the user and are required by the application, defines what the system should do.

- A fully configured application is able to conduct the mission and get results on the touch of a button.
- The user is able to track the location of the aircraft.
- The user can see the live video feed as well as a filtered feed from the aircraft.
- The user can see the live calculation of the yield estimate.
- The user can configure the flight plan and application and is able to calculate the waypoints accordingly.

- The user gets live information about the aircraft and remote controller such as, altitude, horizontal distance, vertical and horizontal speed, GPS signal, Lightbridge signal, aircraft battery and RC battery.
- Users have access to configure the camera and OpenCV filtering settings like, white balance, feekes growth stage, luma, ChormaR and ChromaB components of YUV frame.
- It can return the problem areas to the user which they can see on the map screen.
- It captures the images, where the problem area resides, and allows user to see the see the captured images.

3.2 Non-Functional Requirements

Listed below, are the requirements that are quality attributes of the system. These attributes are hidden to the user and are highly recommended but are not necessary for the functionality of the application and defines how the system should work.

- Accessibility: The application doesn't require internet connection.
- Flexibility: The application is compatible with all the apple devices.
- Cost: The application is free for use and it saves a lot of capital for famer on just one initial investment of the aircraft.
- User friendly: The application only has functions, which are required and is very easy to operate.
- User interface: The user has access to all the information on one screen.
- Warning alerts: The user should get message alerts for low battery or if the mission is a fail etc.

Chapter 4 - APIs and Technology Used

4.1 DJI Mobile SDK

The application is using only DJI Aircrafts because it is using only DJI Software Development Kit[1]. This SDK acts as the middle layer for communicating between the DJI Aircrafts, remote controller and the mobile phone. This SDK provides features for controlling and monitoring every getter and setter function of the aircraft. The documentation for its SDK is provided in DJISDK Reference[2] and a sample application is also provided on their GitHub page[3].

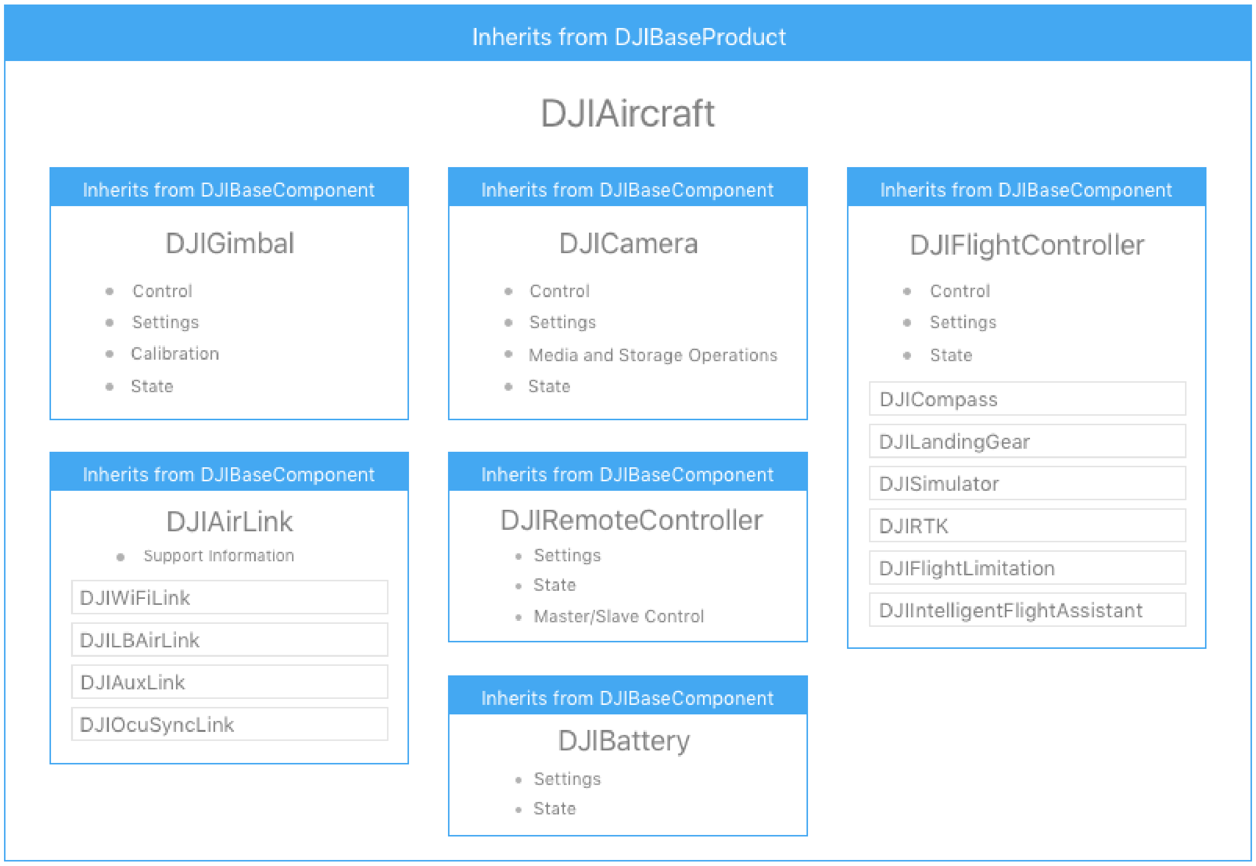


Figure 4-1 DJIAircraft Class API[5]

4.1.1 DJIAircraft

For getting the current GPS position and heading of the aircraft this class is used. We can provide instructions such as, increase the altitude or come back to the original position through this class.

4.1.2 DJIBattery

This class of the API is used to get the battery power percent left in the aircraft and Remote Controller. We can find how much a battery has depleted with time and also help in giving better warnings when there is low battery.

4.1.3 DJICamera

This class is used to control the camera operation, including the exposure, camera capture and record, playback, downloading the image and white balance etc. It is used for getting video feed from the aircrafts to the phone.

4.1.4 DJIMission

This class is used for setting the aircraft mission for a fixed flight path, it uses the information provided by the user like altitude, max speed, auto speed, action after finish and heading for setting a proper flight plan.

4.1.5 DJIFlightController

While the DJIMission save the flight path on the aircraft DJIFlightController gives the user information about the current location, horizontal distance, vertical distance, horizontal speed and vertical speed on the aircraft.

4.1.6 DJIRemoteController

This class is used to get the information from the remote controller, like the number of satellites connected to Aircraft GPS, RC battery status, Light bridge signal strength etc. This class provides the connection link between the aircraft and the mobile application.

4.1.7 DJIWaypoint

Using the calculation of the field of view of the camera we can define the end points around the area boundary we have selected in the application. These points are then saved in an array and used as input to DJIWaypoint class to set the path of the aircraft mission. This acts as an input to DJIMission class.

4.2 OpenCv for Objective-C/Swift

Open Source Computer Vision[4] is a software for image processing and machine learning. We use this library in our project to convert image color space (YUV Frame -> RGB Frame), filtering the image based on the color spectrum (RGB Frame -> filtered Monochrome Frame), overlaying the original image on the monochrome image (filtered monochrome AND original Frame -> Filtered RGB Frame), and converting the RGB to again YUV (Filtered RGB Frame -> Filtered YUV Frame). The application is built in Swift but the library is only in C/C++ therefore a OpenCVWrapper is made in the application to provide OpenCV functionalities.

4.3 QGIS

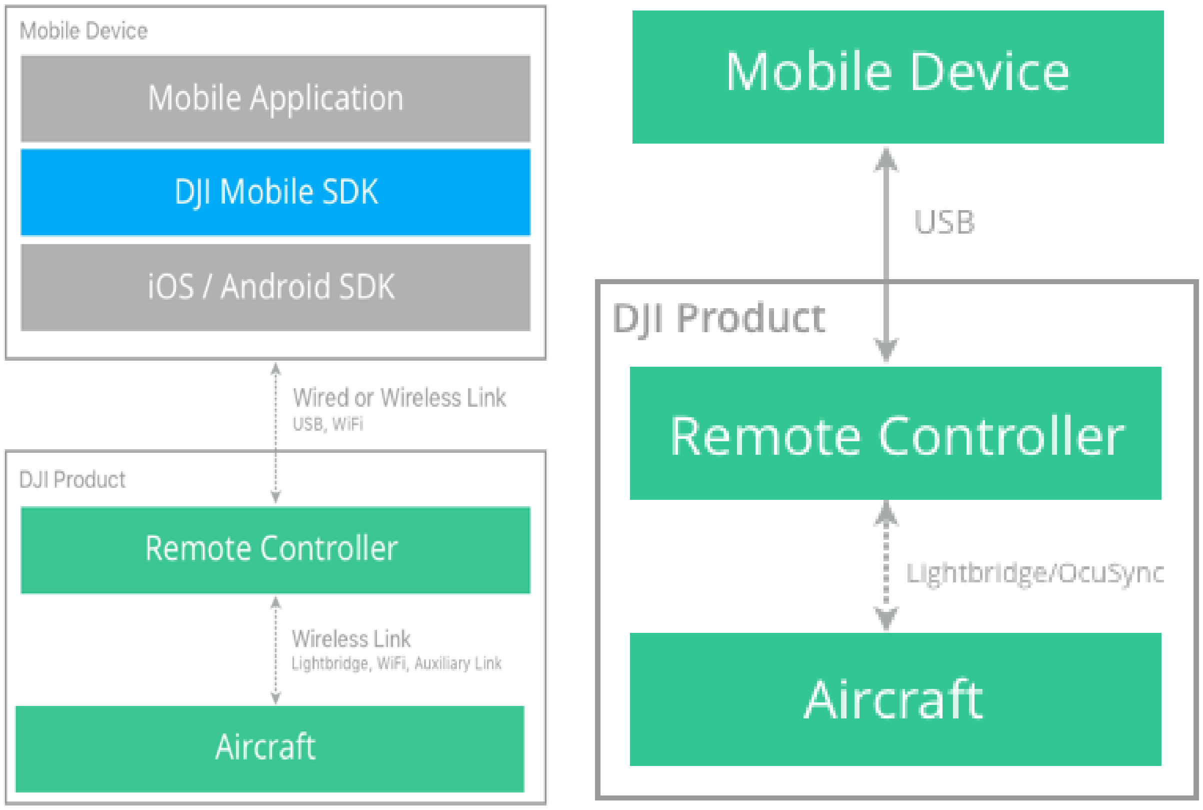
This tool is used to preprocess the shape file (the bounded field) to CSV file. Shape file is in the format of .ZIP which contains three or more compressed files having the coordinate, shape and position information. Such files have to be decompressed and read in to the application, which is why we use this software to decompress the shapefile, convert the bounded region to vector coordinates and export it as CSV file.

4.4 MPMapView

This API is used for viewing the world map on the phone. This helps to locate the users coordinate, aircraft coordinate and mark the waypoint of the path on the map so that users can easily track the location of all of the above. This class is given the CSV file on the startup as a GoogleDoc link, it then downloads the link and annotates the points using its coordinates and then uses DJISDK delegate methods to trace the path of the aircraft in real time.

Chapter 5 - Architecture

5.1 Connectivity



a. b.

Figure 5-1 Connectivity Diagram between iPhone and DJI Aircraft[6]

There are three devices working together (i.e. Figure 4-1 a). DJI Software Development kit is installed on the mobile device (compatible with iOS and Android) through an application through the app store. This mobile SDK connects to a DJI Remote controller through a wired or wireless link. The remote controller is implicitly connected to the aircraft with a wireless link, which may be a Lightbridge, Wi-Fi or Auxiliary link.

For this application i.e. the DJI-SDK is installed in an iOS application, which is connected to remote controller through USB for the actual working of the application. For testing purposes, another application on another iOS is used (DJI bridge) because there is only one Lightbridge on a phone to connect and debug. The connectivity between the device and the connection to the remote controller and the device connected to the Mac for debugging, is through Wi-Fi. More information is provided on this topic in the Chapter: Testing.

The connection between Remote Controller and the aircraft is through a DJI Lightbridge, which can provide 2.4G Full HD downlink from up to 1.7 km. Such a connection is vital to transferring high resolution video feed from the aircraft to the mobile device. Unlink the mobile SDK and you can connect to a different aircraft and its RC, the connection link through Lightbridge is for assigned RC-Aircraft pair only. One RC cannot connect to multiple aircrafts and vice versa. For connecting multiple aircraft through, a mobile application Wi-Fi connectivity is used.

5.2 System Architecture

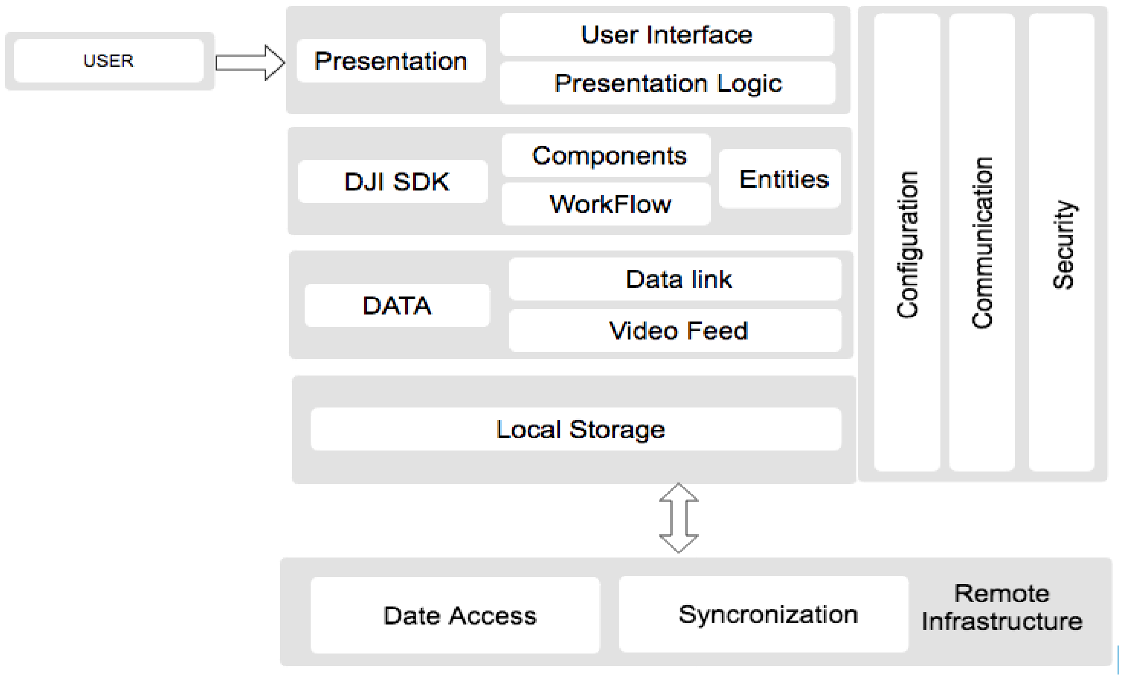


Figure 5-2 System Architecture

The application architecture is divided into four layers:

- Presentation Layer: This layer interacts with the user and provides GUI to the user. It enables the users interaction by actions such as tap, scroll, and swipe etc. Application delegate functions updates the application UI based on the implemented SDKs.
- API Layer: This layers provides the native application with additional functions. For this application, we added DJI SDK to control and configure the aircraft. We added MKMap for map and location services, video previewer for processing the video feed from the aircraft, and OpenCv for processing the frame. It adds components and entities and provides functions to control them as a middle layer between them.
- Data Layer: This is the layer which manages the data between the APIs and the components. It provides the data link for the video feed and capturing and saving the data.
- Local Storage: This is the layer which saves the data in the data storage in the phone. We use this layer for saving the coordinates for waypoints over the field boundary so that users don’t have to do it again, as well as saving the images of the problem areas for later analysis.
- All the layers above have three common layers; configuration to configure them based on requirement, communication so that each layer can communicate with each other, and security which is a future scope in our application.

- The application is connected to remote infrastructure. In this case, the Remote Controller of the aircraft with lighting cable, which provides the data link and synchronization to the mobile application.

Chapter 6 - Implementation & Design

The implementation of this project can be subdivided into two modules based on its functionality and User Interface. The aircraft has to first scout across the field as a waypoint mission and then take pictures and then process them to calculate yield estimation and N Recommendation in parallel.

6.1 Waypoint Mission

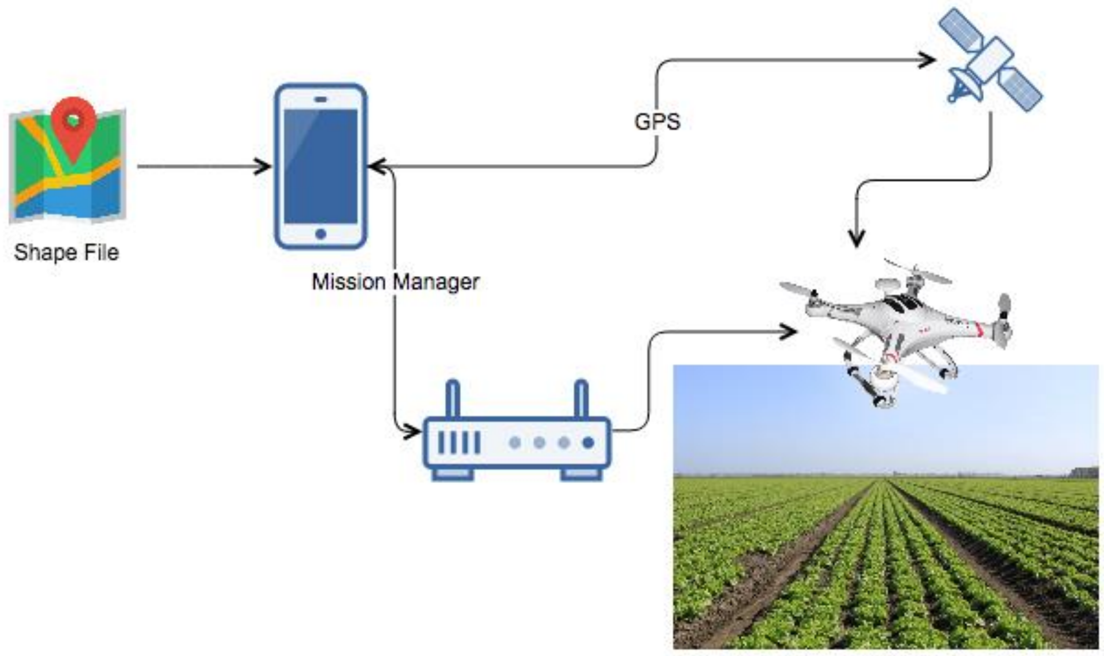


Figure 6-1 Waypoint Mission Management

First, the user has to make a shapefile for his field, to do so, there are many software programs like, ArcGIS and QGIS etc., he/she then converts the shape file into the vector coordinate points and exports it as a CSV file. This file is then uploaded to Google drive and the sharable link is copied and imported into the application. This process has to be done one time only and the shape file coordinates can be changed in the application again. Using the MKMap SDK in iOS, the coordinates are connected to make a polygon and overlay on the MapView. Based on the altitude and camera configuration the field of view of the camera is calculated and based on that the way points are calculated. Once the waypoints are marked on the map we use the Mission Manager in the DJI-SDK to create a mission for the aircraft. When the user clicks the start button, the mission is executed on the aircraft and is not interrupted even if the application crashes. The current location of the aircraft is located through the GPS and is sent to the application through the Remote controller. Once the mission is completed, the aircraft can be configured to return to the original position.

6.2 Yield Estimation and N-Recommendation

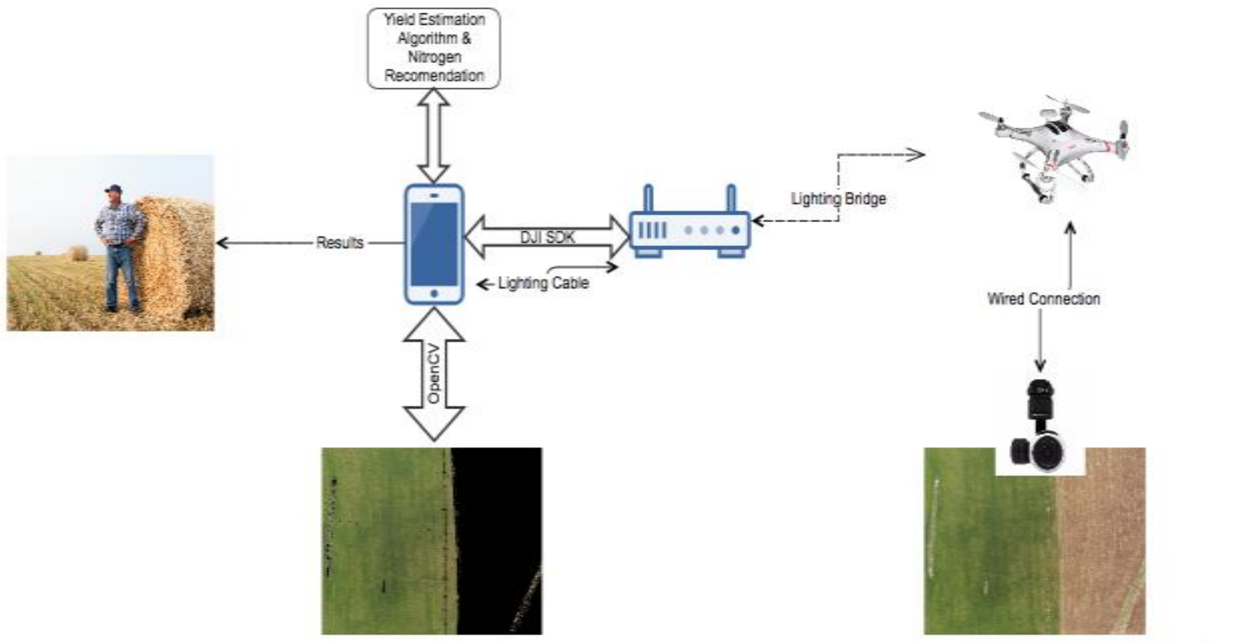


Figure 6-2 Yield Estimation and N-Recommendation

Once the aircraft is on its way, it starts collecting the video frame from its camera. Each video frame is collected by the Remote Controller through Lightbridge and sends to the application through the Lighting cable. These raw frames (H.246) format is then captured by the application and converted to iOS compatible video codec and then decompressed. This decompression is intercepted by another function which called the OpenCV API, this captures the frame in to YUV format. The filtering is done by limiting the values of Y(Luma), U(or ChromaB, difference of green and blue), and V (or ChromaR, difference of green and red) components. This setting can be manipulated through the camera view on the application as well. Based on the number of pixels being filtered and the number of pixels being passed were calculated by the percent of image that is in the threshold of green and then passes this information on to the equations for calculating the Yield potential and N recommendations. This is done for each frame hence the results are live to the user.

6.3 System Design

Unified Modeling Language (UML) is a standard way to visualize design of a system. It represents both the static and dynamic view of the system. The static view describes the relationships between entities, objects, operations, and attributes. The dynamic view describes the behavior of the system. The dynamic view also refers to instances of entities and how it changes and interacts. The following are the UML diagrams of this application's system.

6.3.1 Class Diagram

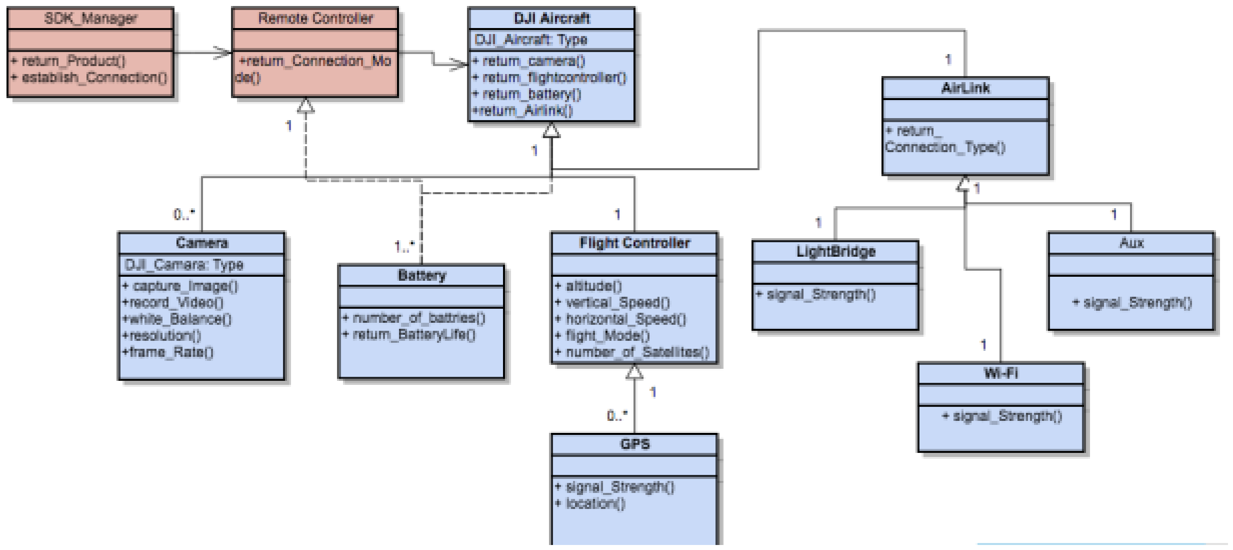


Figure 6-1 Class Diagram

Class diagram is a type of static view diagram. It defines the structure of the system in object oriented view. It is used to construct the executable code of the application. Class diagram describes the constrains of each class and its functionalities. Class diagram also defines the relationship among classes (generalization/specification) and multiplicity. Figure 6-1 represents the class diagram of the application. DJIAircraft is the parent class of the API and holds the instances of Camera, FlightController, Battery, and AirLink. Aircraft have a One-One multiplicity to all the classes except Camera (zero or more) and Battery (one or more).

6.3.2 Use Case Diagram

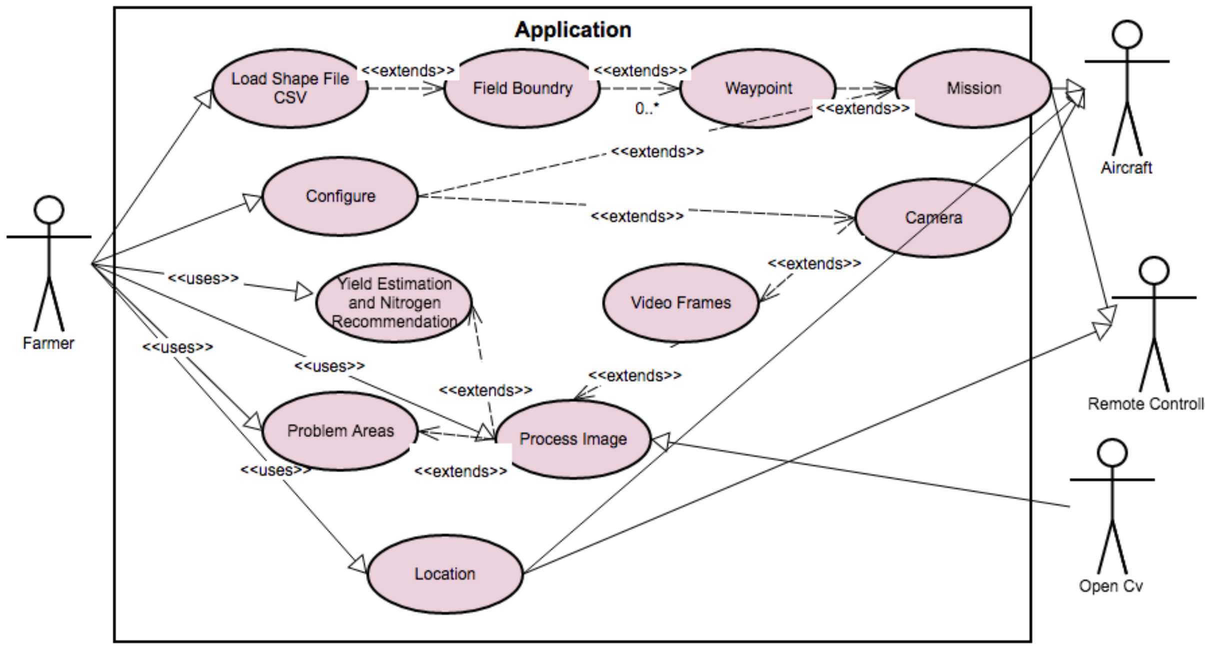
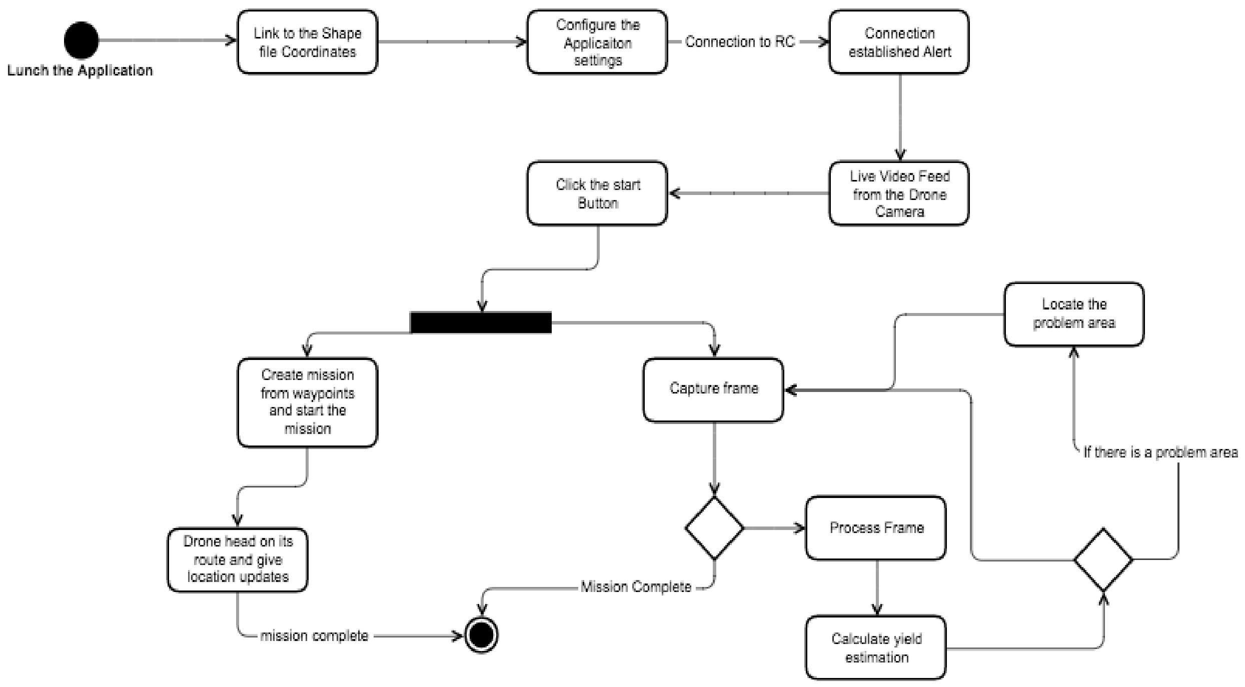


Figure 6-2 Use Case Diagram

The Use Case diagram is a dynamic view diagram. The Use Case diagram represents the behavior of the system and how the system interacts with the actors outside the application boundary. The Use Case diagram describes the activities responsible for each actor. Figure 6-2 represents the Use case diagram of the application. There are four actors in the system, Farmer (user), the Aircraft, the Remote Controller, and the OpenCV. A user proves the shapefile and configure the application and uses the Yield estimations, N-Recommendation, Problem areas, location of the aircraft, and the processed frames. OpenCV function is to process the frame and the Remote Controller handles the aircraft's mission. The aircraft executes the mission and captures the video frames.

6.3.3 Activity Diagram



Lunch the Application
Link to the Shape file Coordinates
Configure the Applicaiton settings
Connection to RC
Connection established Alert
Live Video Feed from the Drone Camera
Click the start Button
Create mission from waypoints and start the mission
Capture frame
Locate the problem area
If there is a problem area
Drone head on its route and give location updates
Process Frame
Mission Complete
mission complete
Calculate yield estimation

Figure 6-3 Activity Diagram

The Activity diagram is also a dynamic view UML diagram. The Activity diagram describes the flow of control from one operation to the other. The flow can be sequential, branched, conditional or parallel. It has a start point and an end point which defines the end of the applications all functions. Figure 6-3 represents the activity diagram of the application. The control flow of the application is parallel as the user hits the start button as one thread controls the aircraft’s location and mission, and the other thread controls the video frame capture. The video frame captures runs in a loop until the mission is complete and for each frame it processes it to get the result and decides if the frame has a problem area or not.

6.3.4 Sequence Diagram

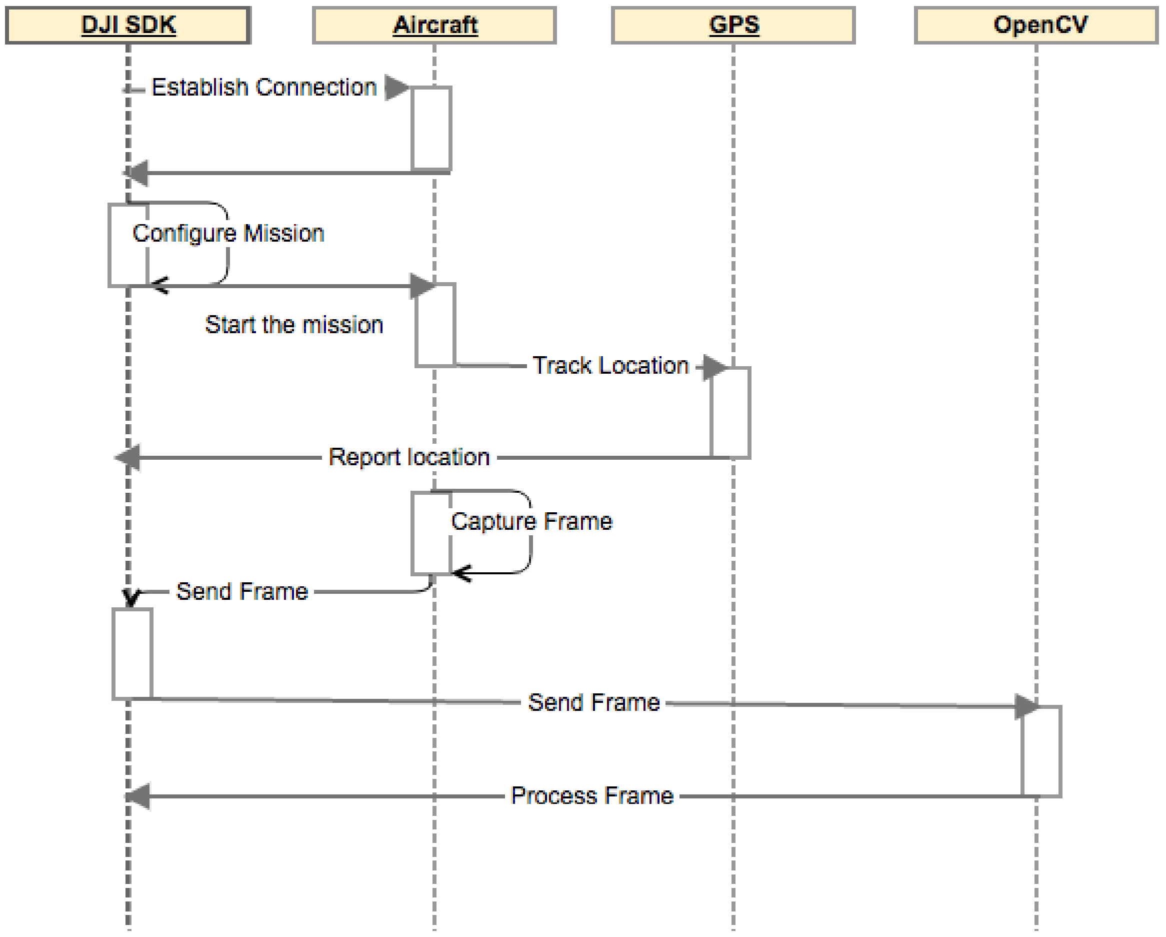


Figure 6-4 Sequence Diagram

The sequence diagram, or swim line diagram, is also a dynamic view diagrams, which describes the instance interactions with time. It also represents the message exchanged between the objects for each function and time taken for each object to respond. Figure 6-4 represents the sequence diagram of the application.

Chapter 7 - Application GUI and Working

The application is an iOS native application and in turn, the GUI is a restrictive to iOS platform, unlike a web application. We divide the GUI into three parts, Configuration, Monitoring and Validation.

7.1 Configuration

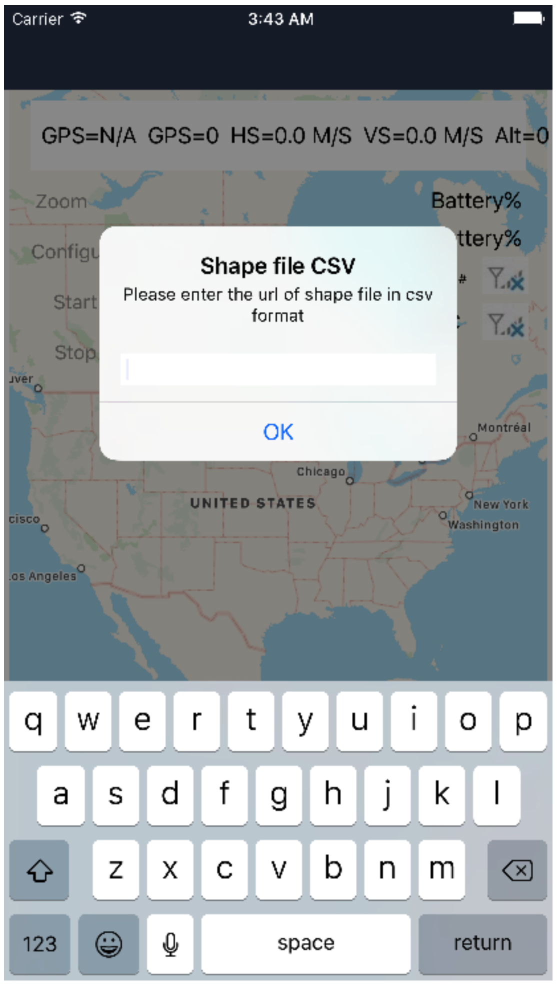


Figure 7-1 Shape File link Setup

Figure 6-1 represents the initial screen for the first time user. Once a correct CSV link is uploaded in the google drive, this alert won't appear when opening the application. This is a sample link of the field in which we have simulated and tested the application on: https://drive.google.com/file/d/0B_TLsJS5vbjzaXVWcE1HalZuY1k/view?usp=sharing.

Once the link is entered and the field boundary is set up, then we see the following screen on the Figure 7-2. On the top left side of the screen we see four buttons.

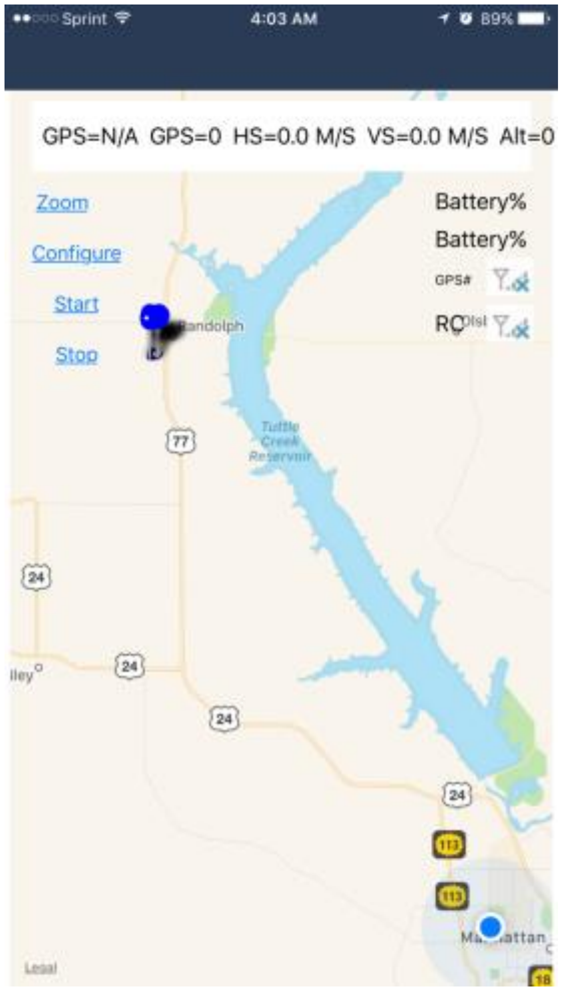
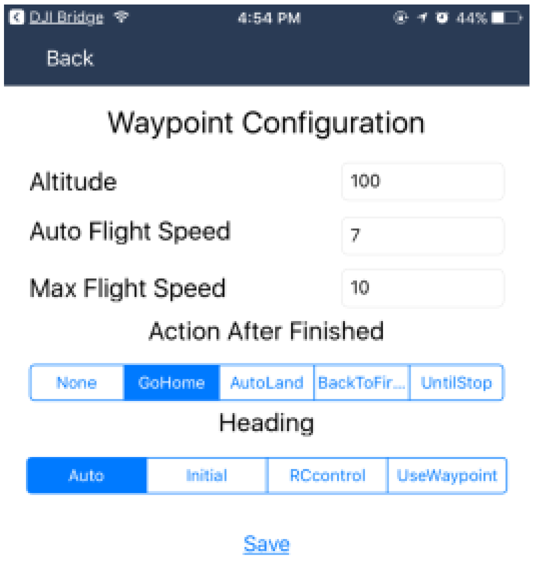


Figure 7-2 Map View Aircraft Not Connected

First button "Zoom" zooms the map view to the aircraft, if the aircraft is not connected then it zooms to the user location (bottom left corner of the screen). Second is the "Configure" button which is used to change the default setting of the mission. "Start" and "Stop" buttons are to start and stop the mission.

The “Configure” button takes you the following screen on the Figure7-3.



Waypoint Configuration

Altitude: 100

Auto Flight Speed: 7

Max Flight Speed: 10

Action After Finished

None [✓ GoHome] AutoLand BackToFir... UntilStop

Heading

[✓ Auto] Initial RCcontrol UseWaypoint

Save

Figure 7-3 Configure Screen

There are five options which can be changed. All these options have default values which are preset on the application.

- Altitude: The default altitude is set at a 100 meters. The higher the altitude, the bigger the field of view and the aircraft have less rotations of the field, which will also reduce the flight time, but the resolution of the field will be smaller and may not get correct results.
- Auto Flight Speed: The maximum speed of the fight is 17 meter per seconds. Auto fight speed is the speed at which the aircraft tries to fly.
- Max Fight Speed: On a windy day when the aircraft is flying with the wind, it may fly with higher speed than the auto speed to save battery life. This option restricts the maximum flight speed of the aircraft.
- Action After Finish: This is to set the action of the mission as completed, the options are to go to the initial position(goHome), land on the last waypoint (AutoLand), back to the first waypoint, and stop on the last waypoint without landing.

- Heading: It is the direction at which you want to make the aircraft face. It can be auto or fixed at the initial position, or to face the Remote controller or to face the waypoint.

7.2 Monitoring

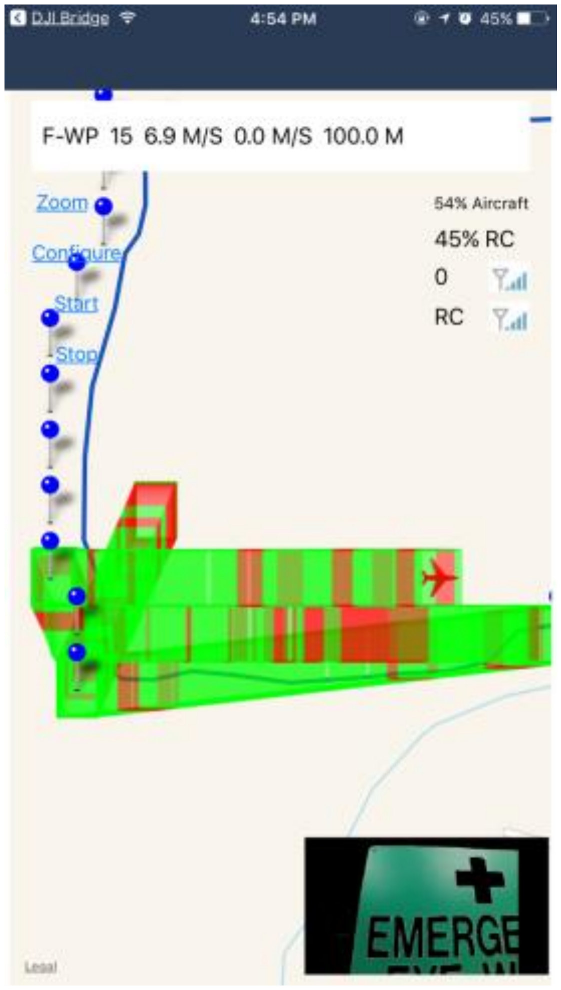


Figure 7-4 Map View Aircraft Connected

When everything is configured, mobile application is connected to the Remote Controller, both Remote Controller and the Aircraft is turned on and is connected as the user taps the “Start” button the aircraft starts the mission and Figure 7-4 represents the map view when the sUAS is connected.

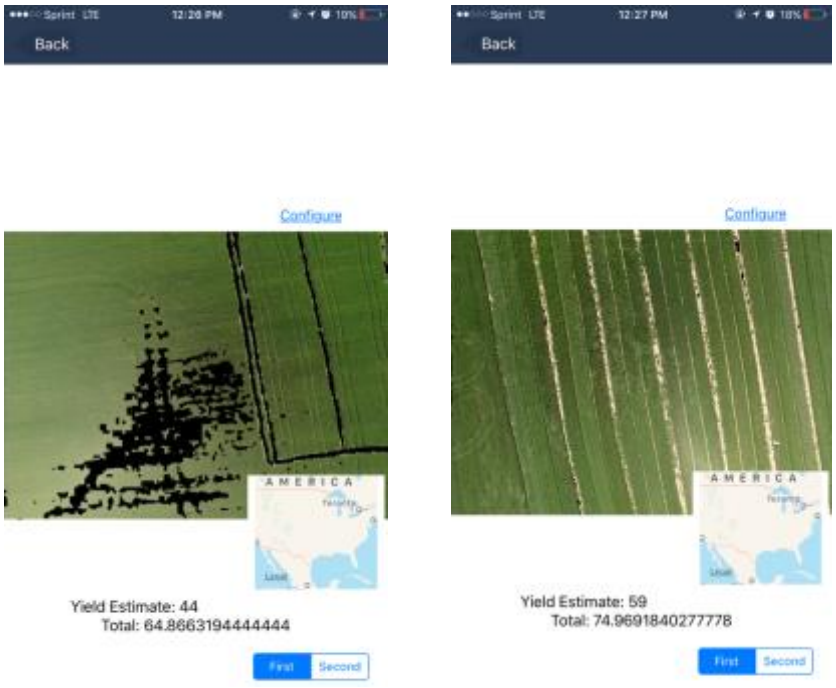
On the top, we see first the mode in which the sUAS is flying, it should be F-WP representing flight mode on waypoint mission. Next is the number of satellites connected to the aircraft locating the aircraft on the map, followed with horizontal speed, and lastly with the vertical speed and altitude.

On the right side we see the aircraft battery life remaining, followed by the Remote Controller’s battery life, followed by the GPS signal strength, and lastly the Remote Controller’s signal strength to the aircraft (in this case the Lightbridge).

If the aircraft is connected to a DJI camera, then the application automatically shows a small camera view on the map view. It is also a button to the full camera view and options. Note the camera is automatically configured to filter every color except green color.

On the center map we user can see the current location on the aircraft in real time with the representation on the bad part of the field (red color) and the good parts of the screen (green) and the field which is not yet analyzed (no color).

7.2.1 Camera View



Yield Estimate: 44
Total: 64.8663194444444

Yield Estimate: 59
Total: 74.9691840277778

Figure 7-5 Camera View

When the mini camera view on the Map view is tapped it brings the user to the camera view. As shown in the Figure 7-5 are the examples screenshots when a sUAS is flying over wheat field. User can see the yield estimate of the field in real time. User can also configure the camera settings using the configure button on the top left.

The camera view/configurations as seen in the Figure 7-5 has the following attributes that the user can control:

- The Luma, ChromaR and ChromaB values of the filtering.
- The growth stage of the crop.
- The camera white balance.
- Threshold.

On the camera view user can monitor the following:

- The yield estimate of the crop.
- Total amount of the camera pixels is filtered out.
- Live feed of the filtered camera view.

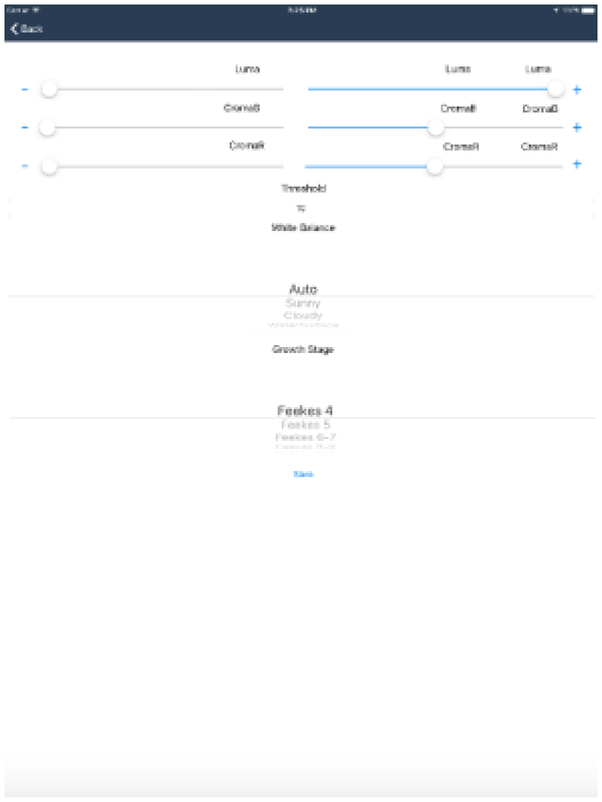
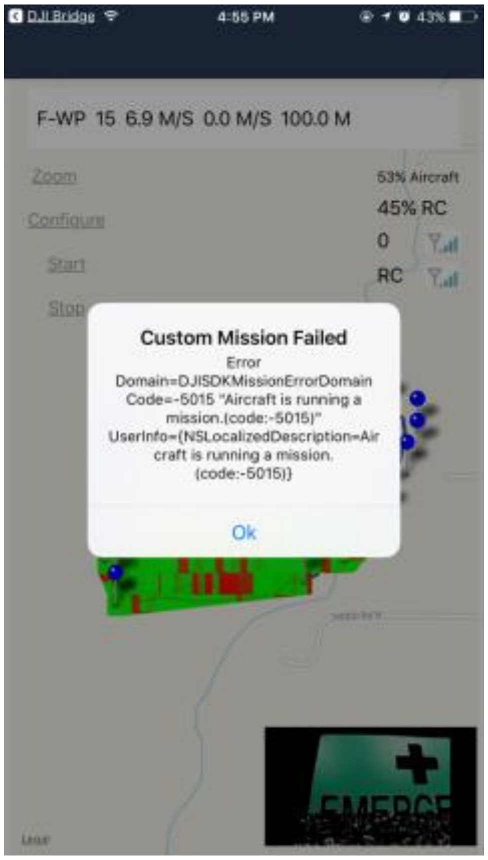


Figure 7-6 Camera view Configure

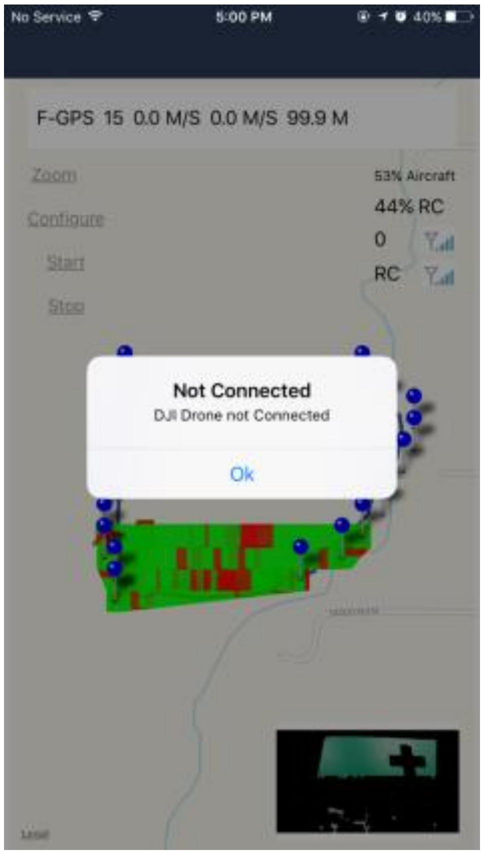
7.3 Validation



F-WP 15 6.9 M/S 0.0 M/S 100.0 M

Custom Mission Failed

Error Domain=DJISDKMissionErrorDomain Code=-5015 "Aircraft is running a mission.(code:-5015)" UserInfo={NSLocalizedDescription=Aircraft is running a mission.(code:-5015)}



F-GPS 15 0.0 M/S 0.0 M/S 99.9 M

Not Connected

DJI Drone not Connected

Figure 7-6 Mission Failed

When the mission is attempted to started there are variety of reasons that it can be failed, the user needs to know what might be the reason to a mission failed. For this reason, we have added notification alerts in the application.

The left side of the Figure 7-5 is an example of a mission failed when there is already a mission running, there might be other reasons for a mission to fail. There might be other reasons like, low battery, or the aircraft is too far from the first waypoint etc.

The right side of the Figure 7-5 is an example of a notification when the Remote Controller of the aircraft is disconnected from the mobile phone. Other notifications can be, the aircraft connected, or the mission started etc.

Chapter 8 - Testing

To ensure the correctness of the application, we have conducted tests on the simulator and on the test field. All the tests are done manually and verified on iPhone 7, iPhone 6 and iPhone 6+, DJI Mavic, and DJI M100 aircraft. The camera used is DJI X3. We conducted unit testes over each function of the application and exact location of the aircraft on the field. We conducted integration testes between each API integrating with the other. We also conducted compatibility tests using different iPhone and DJI drones.

8.1 Unit Testing

Unit testing is the testing of each function of the application independently to confirm that each function of the application is working correctly, independently.

Table 8-1 Unit Test

S. No.	Test Case	Expected Result	Result
1.	Application Lunch	Alert asking for link to shape file	Passed
2.	Enter correct link on Alert	Mark the boundary and waypoints based on default values.	Passed
3.	Enter incorrect link on Alert	No boundary and waypoints are marked	Passed
4.	Lunch application for the second time	No alert and, Map View lunch (Aircraft not connected)	Passed
5.	Check the user location	Correct user location on the map view	Passed
6.	Zoom button (aircraft not connected)	The map should zoom to user location	Passed
7.	Configure button	Open configuration screen with five options	Passed
8.	Check default values	Check if the default values are set in all five field.	Passed
9.	Altitude change (lower)	Check the way points after lowering the altitude and taping save. It should be closer than before (more waypoints)	Passed

10	Altitude change (higher)	Check the way points after higher the altitude and taping save. It should be farther than before (less waypoints)	Passed
11.	Connect aircraft	Aircraft annotation should be on the map	Passed
12.	Check flight information	Check the flight mode (as set on the RC), altitude, VS, HS should be zero	Passed
12.	Check the battery information	The Aircraft battery and the RC battery should up and should be correct	Passed
13.	Check GPS and Light Bridge signal information	The GPS signal should be correct and Light bridge signal should be high id RC and aircraft are near to each other	Passed
14.	Check the camera view mini screen on the map view	If the aircraft have a camera installed then there should be a mini screen pop up on the map view	Passed
15.	Tap the mini camera view	A new camera view screen should pop up will all the configuration controls	Passed
16.	Check the white balance control	Default white balance should be set to auto and changing the UIpicker should check the camera white balance	Passed
17.	Check the YUV filter controls	Checking the YUV filter range should change the camera filtration	Passed
18.	Check the Green, Yellow, Total values	Once the camera is connect Total, green and yellow pixel values should pop up out of 100	Passed
19.	Check the growth values	The default feekes value on the UIpicker should be set to 4. And should check the growth value when UIpicker values are changed.	Passed
20.	Tap the “Enable Control”	All the configuration controls should be hidden now and only the camera view should be clear.	Passed
21.	Change the orientation on the phone	The camera view should fit the screen in both landscape and portrait view.	Passed
22.	Start the mission	The mission should start and aircraft should be on its way	Passed
23.	Check the altitude and speed of the aircraft	The altitude and speed of the aircraft should not exceed the configured values	Passed

24.	Check the aircraft heading	The aircraft heading should be as configured in the application	Passed
25.	Check the aircraft finish action	The aircraft should respond the same way as configured after finishing the mission	Passed
26.	Check the problem areas	The aircraft while flying should detect the problem areas with red color on the map and the rest with green color on the map	Passed
27.	Check the stop button	The aircraft should stop flying	Passed

8-2 Integration Testing

There are modules in the application in which calls other modules and this is the test to see if those modules would be in cohesion with one another. These modules include the local storage, OpenCV API, DJISDK API, and video Processing API.

Table 8-2 Integration Testing.

S. No.	Test Case	Expected Result	Result
1.	Save the waypoint information	Once the correct shapefile is entered in the application, it should be saved in the application local storage.	Passed
2.	Change the flight plan	Changing the flight altitude, speed and heading should change the flight plan	Passed
3.	Change the camera filtration	Changing the filter values of Luma, ChromaR and ChromaB UIrangePicker change the filter values. (OpenCV API is working from the application)	Passed
4.	Change the shapefile	Changing the shapefile to confirm that the application calls the server and gets the new way point information.	Passed
5.	Change the user location and map information	Changing the user and aircraft location confirming the map API is working.	Passed
6.	Switch the page from camera view to map view and vice versa	Change the page integration from one page to the other.	Passed

8-3 Compatibility Testing

This application is for iOS devices only and we have tested this application with variety of devices ranging from iPhone 6 plus, iPhone 5S, iPhone 6s, iPhone 7-7 Plus and iPad Pro. For the small screen devices such as iPhone 4S and below we have used simulators to test the application and below are the following test results.

Table 8-3 Compatibility Testing

S. No.	**Test**	**Result**				
		iPhone 6 plus	iPhone 6S	iPhone 7	iPhone 7 plus	iPad Pro
1.	Install and lunch	Success	Success	Success	Success	Success
2.	Stress Test	Success	Success	Success	Success	Success
3.	Page Transition	Success	Success	Success	Success	Success
4.	Regression testing for older versions	Success	Success	Success	Success	Success

Chapter 9 - Conclusion and Future Scope

Smart phones are the most portable and widely used devices in the world and they are constantly increasing in their capabilities every day. This has given us the ability to think about thousands of possible ways to make the lives of these user easier. This is one such attempt to help users and in this case farmers to make use of their smart phones to perform an industrial task. The project in its testing and beta version has met all the functional and nonfunctional requirements and passed all the tests in the lab and the field. I have learned a lot working on this project including how to make, test and publish applications on iOS platform, image and video processing, using sUAS APIs, controlling and flying an sUAS, Precision Agriculture. We hope this application gets success in its purpose of being an essential tool for farmers, but our work here is not complete. We plan to include other features to this application in the future:

- Connecting multiple sUAS so that survey of a larger field can be done parallel with one device.
- Adding yield estimation and nitrogen recommendation algorithms for other crops.
- Merging this application with other Precision Agriculture applications so that user can get a complete experience with just one application.
- Effective results for this application is restricted to 20 m altitude in the future we are aiming to improve this distance to further above so that bigger field of view of the camera can analyze larger area.
- Include the iOS camera capability so that farmer can analyze the field but using the iOS device camera if he doesn’t have a sUAS

References

[1] DJI Developer (n.d.). *DJI Mobile SDK Documentation*. Retrieved November 11, 2015, from https://developer.dji.com/mobile-sdk/documentation/introduction/index.html

[2] DJI Developer (n.d.). *DJI Mobile SDK iOS Reference.* Retrieved November 20, 2015, from https://developer.dji.com/iframe/mobile-sdk-doc/ios/index.html

[3] DJI Developer (n.d.). *DJI Demo Application.* Retrieved November 1, 2015, from https://github.com/dji-sdk/Mobile-SDK-iOS

[4] OpenCV(n.d.). *OpenCV Documentation for iOS.* Retrieved January 10, 2016, from http://docs.opencv.org/2.4/doc/tutorials/ios/table_of_content_ios/table_of_content_ios.html

[5] DJI Developer(n.d.). *DJI SDK Architecture.* Retrieved March 2, 2016, from https://developer.dji.com/mobile-sdk/documentation/introduction/sdk_architectural_overview.html

[6] DJI Developer(n.d.). *DJI SDK Connection.* Retrieved February 12, 2016, from https://developer.dji.com/mobile-sdk/documentation/introduction/mobile_sdk_introduction.htm

[7] Gliffy (n.d.). gliffy for making diagrams. Used March 25, 2017, from https://www.gliffy.com