Hot Chicks: UAV-based Surveying System for Bird Nest Detection

Team: Monica Hung, Aniket Mathur, Dominique Meyer, Sriram Venkatesh, David Yang, Matthew Yu

Abstract

Habitat restoration and other land management practices are vital to the continued health of open spaces, ecosystem functioning, and provision of ecosystem services such as forestry and agricultural production. On-the-ground ecological land management activities oftentimes require extended and disruptive land modification, calling for a transition from land management to state-of-the-art UAV-based approaches. We propose a novel technique to improve and speed-up ground-based ornithological surveys of nesting migratory birds using mounted Long Wavelength Infrared (LWIR) and visual spectrum sensors on a multirotor UAS platform. Combining the ability to downlink data in different visual bands with GPS coordinates, we provide high-accuracy locationing of potential bird nests for in-field verification and ecological land assessment.

Table of Contents

1. Introduction

Bird nesting occurs during a majority of the calendar year and faces major threats from land restoration projects, especially on-the-ground ecological land management activities performed during breeding season. In this work, we propose a novel UAS-based bird nest detection and surveying technique to replace harmful ground-based ornithological surveys commonly carried out before land-management activities.

Current bird nest detection approaches involve sending ornithologists on foot through the habitat: these experts listen for nesting and breeding calls while also utilizing their experience and binoculars to locate bird nests. This method is not 100% accurate and can be time-consuming, expensive, and harmful due to unavoidable habitat damage caused by the surveyor. Recently, Unmanned Aerial Systems or UASs (also referred to as Unmanned Aerial Vehicles, UAVs, and drones) have been growing in popularity as an alternate solution because of their immense potential in surveying difficult-to-reach areas. By mounting cameras and additional sensing equipment on these aerial systems, we can take photos and videos while exploring hard to reach locations from views humans cannot normally reach on foot. This data can be recorded, geolocated, and mapped for further inspection by experts so that appropriate precautions can be taken to allow land management practices to proceed. In addition, UAS mobility combined with the ability to transmit wireless data from the UAS to the ground allow us to conduct real-time visual inspection of captured video and infrared data. This rapid processing time allows us to further investigate potential sites of interest and gather additional information about nesting areas – even mid-flight.

There are many obvious advantages that UAS-based bird nest identification approaches have over traditional on-the-ground methods. First, aerial surveys are minimally intrusive and do not disturb bird habitats or the birds nesting in them, which is an important consideration for protecting migratory or endangered birds. Second, we avoid ground-based environment challenges in areas that are difficult or impossible to access. Third, UAS-based surveys are faster and cheaper than human based surveys, providing more time and resources for more thorough and accurate inspections.

But while there are many advantages of UAS-based approaches, there are also disadvantages as well. Many times UAS cameras can only identify nests if the perfect viewing angle is achieved above plant-life, as bird nests are commonly located deep within brush and trees, and sometimes nests are impossible to identify from the skies. As a solution, we propose utilizing infrared filters as well as other visual spectrum sensors on our UAS platform. These sensors capture additional information in different light wavelengths unseen to the human eye to complement our visual and thermal data. For example, captured multispectral image data may identify particular ecological points of interest, such as water sources, plantlife, and terrain, which may help point us towards areas with higher probabilities of bird nests during land surveys.

In this report, we propose a novel technique to improve and speed ground-based ornithological surveys for nesting migratory birds using mounted Long Wavelength Infrared (LWIR) and visual spectrum sensors on a multirotor UAS platform. Our approach complements, improves, and speeds-up the ornithologist surveys, rather than displaces them. Our proposed system consists of a multirotor UAV with a Raspberry Pi mounted on it. The RPi is the main processing unit, and is connected to an Emlid Navio2 for flight control and GPS sensing. The multispectral camera and the visual camera are then mounted on the UAV for visual, thermal, and multispectral data. The aerial system is given long-range communication capabilities through Ubiquiti radios, which allow real-time communication with the base station. The base station computer gathers the data sent to the RPi from the cameras and sensors and processes them.

2. System Description & Results

2.1 Hardware and Software Components

Figure 1 shows our air-to-ground wireless connection design, with UAV components and ground station components. In the next two subsections, we describe these individually and explain the working of our communication system.

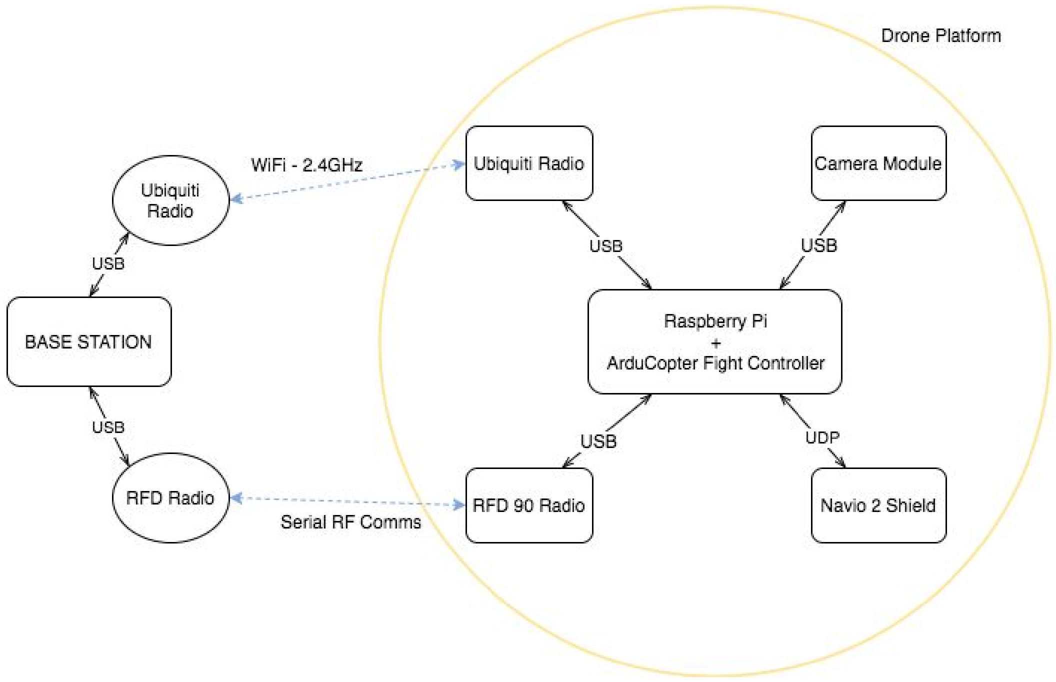
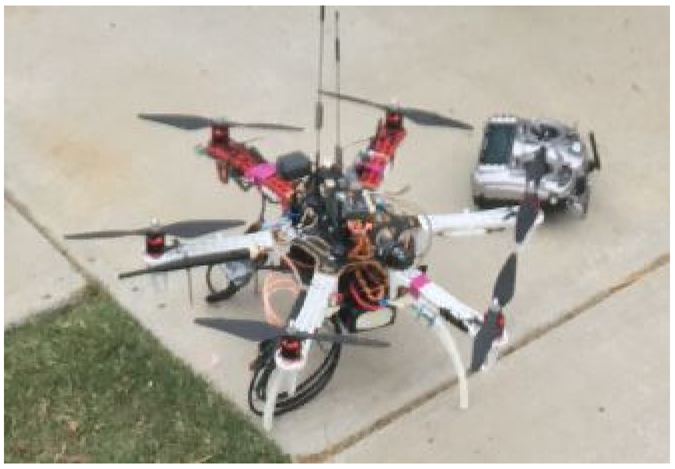


Fig. 1: Full System Diagram

2.1.1 UAV Components



Our drone consists of the following components:

- Raspberry Pi 3 with ArduCopter Flight Software
- Navio2 Shield for Raspberry Pi: contains U-Blox 8 GPS Chip
- See3Cam Visual Camera
- Multispectral Camera
- Ubiquiti Radio - UAV-end

The RPi is the main processing unit of the drone. With the help of GPS, altitude and attitude data from the Navio, the RPi doubles up as a flight controller. The Ubiquiti radio is plugged into the ethernet port of the RPi, and is interfaced to the Ubiquiti radio at the ground station-end wirelessly.

The visual and multispectral camera are then strapped on to the drone and connected to the RPi through USB. The GPS data is queried by the RPi through serial ports (using Mavlink protocol).

2.1.2 Ground Station Components

The ground station consists of the following components:

- Laptop Computer (Windows) with SSH, Mission Planner Software
- Ubiquiti Radio - Ground Station-end

The laptop is connected to the Ubiquiti radio through the ethernet port. We establish a 2-way connection from laptop to the RPi through these radios. We have an RSync script running on the laptop for receiving the files from the drone.

2.1.3 Multispectral Camera

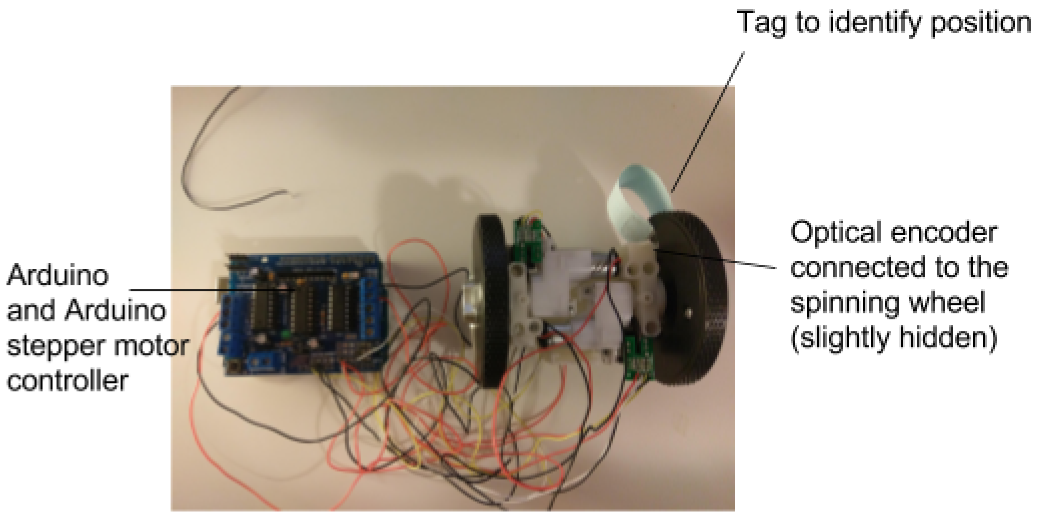
The multispectral camera is made up of a high-speed camera which continuously captures images while a rotating disk of different light wavelength filters spins in front of the camera lens. A closed feedback control system will handle the camera and disk coordination to ensure useful photos are taken in a multitude of different light wavelengths.

2.2 Implementation

2.2.1 Multi-spectral Camera

The multispectral camera system needs a precisely controlled camera and a rotating filter wheel to isolate the bands of light. One of the main problems is knowing when to trigger the camera at the correct moment such that photos are taken when the filter is in front of its lens. Additionally, another problem is identifying which filter is in front of the camera when the image is taken. We created an experimental setup for proof of concept of our multispectral camera design.

We used two optical encoders to address when the camera should be triggered and to address the spin rate of the filter wheel. To calculate speed, the first encoder calculated the time required for one rotation of the filter wheel based on passing spectral filters. To identify the current filter in front of the camera lens, the second encoder detected when a predetermined “start” filter was passing which allowed us to identify the following filters based on ordering. Based on the two encoders outputs’, we determined when to trigger the camera and which filter was being used. Proportional derivative (PD) control utilized the current rotational speed, the distance to the target speed, and the rate at which we were accelerating to determine how to adjust our motor input to reach our target speed quickly. In addition, a high camera rate and spin speed were required to capture enough overlapping photos to create an accurate stitched map of our filtered images.


Tag to identify position
Optical encoder connected to the spinning wheel (slightly hidden)
Arduino and Arduino stepper motor controller

We created a test system to implement an algorithm capable of controlling the speed of the wheel and the precise camera trigger signal. The current version in the above figure uses an Arduino Uno and an Arduino motorshield. This system allowed us to test our complete closed-loop feedback algorithm, but our algorithm will require tinkering when applied to our final multispectral camera solution.

2.2.2 System Integration

The system is setup as described in Section 2.1. We set up all the sensors and establish connectivity with the RPi. We then run the following code in a continuous loop to receive image and GPS data indefinitely.

Main function in Python for gathering data:

```
if __name__ == '__main__':
    ap = initGPS( 'udp:localhost:14550', 15200 )
    vid = initCam( "/dev/video0", 1920, 1080 )

    for i in xrange(20):
        # sleep to wait for data
        time.sleep(0.7)
        filename = "testgps/" + time.strftime("%Y_%m_%d_%H_%M_%S")
        saveGPSData( ap, filename + ".data" )
        saveImage( vid, filename + ".jpg", 3840, 2160 )
        # get image and save to the same filename, different folder
    vid.close()
```

Our main function is the driver of two functions, saveGPSData and saveImage. ‘saveGPSData’ queries the GPS sensor for latitude, longitude, altitude and attitude information and dumps it in a file, through the mavlink protocol. ‘saveImage’ function gets the image with the specified resolution and save it as a jpeg file.

Through many iterations of testing, we have determined that our code runs above the 1 Hz minimum requirement for image capture. We have a sleep time to allow for image to fully capture and save to the disk. Under saveImage, we invoke an instance of the video camera attached on the Raspberry Pi. In our case, that is the See3 12MP video camera. We set our images to capture in full 4K resolution. Under saveGPSData, we call functions that communicate with the Navio2 shield via PyMavlink, a library that talks in Mavlink via Python. Below is a sample of the GPS data received from PyMavlink.

```
GPS data::
        altitude: +2.36000, heading: +254.00000, speed: +1.11893, location: {'lat': 32.9893302,
        'lon': -117.1307326, 'fix_type': 4}, attitude: {'yaw': -1.8476104736328125,
        'roll': 0.008410630747675896, 'pitch': -0.08130650967359543}
```

Via our communications to the ground station computer, we setup rsync, a utility that enables file synchronization between two computers. This is a handy tool because we are able to get real-time data from the RPi, in the form of incremental file lists. The following is a snippet of the bash script running on the ground station computer, which maintains synchronization between folders in the RPi and the ground station.:

```
while [ 1 ]
do
 rsync -avzu pi@192.168.1.150:/home/pi/mavlink_stuff/testgps ~/data_from_pi/
 sleep 1s
done
```

The Ubiquiti Rocket M2 radios support long-range communication at 2.4GHz. We have tested our drone UAV at a 50 meters in the air with reliable constant connectivity.

3. Milestones

Since we had 2 sub-teams working separately on the system integration and the multi-spectral camera design, we created 2 sets of milestones.

3.1 Milestones for System Integration Team

The main milestone for the system integration team was to have a fully integrated system which captures real-time data from the GPS sensor and the cameras, and transmits it to the base-station. We divided the main milestone into 3 smaller tasks: (MS1) setup of individual components, (MS2) transmission of data from drone to ground station, and (MS3) timing analysis to ensure that all data is synchronized, and integrating the newly-designed multispectral camera into the system.

The deliverable for MS1 was to have working python code for saving GPS and image data onto the Raspberry Pi (RPi). First, this involved configuring the Raspberry Pi for SSH and installing the necessary libraries for capturing GPS and camera data. We then flashed the EMLID RTK GPS Sensor, the See3cam camera and the Ubiquiti wireless radios to the latest firmware. We were able to set all these up in 2 weeks, and have working code to dump time-tagged images and GPS data into files. We were also able to set up one of the Ubiquiti radios to be an emitter of a Wifi signal and get the other radio to connect to it.

The deliverable in week 7 for MS2 was to show a demo of data sync between the ground station and the drone in real-time. This primarily involved setting up RSync on the RPi and the computer, communicating through the Ubiquiti radios, for real-time syncing of data. We had issues with setting up the peer-to-peer network with the Ubiquiti radios and struggled with accessing the RPi through the radios. However, we were able to fix this by going through the Ubiquiti manual and fixing some of the settings. We also determined a methodology for obtaining raw GPS data from EMLID for future processing by the Pi, and configured necessary SW for porting the GPS raw data to the Raspberry Pi. By the end of week 6, we had a working demo where our ground-station was able to receive the images and GPS metadata from the sensors by querying for the incremental file list every 2 seconds.

Our final milestone MS4 involved ensuring that we have visual data every second, and have the images time-synced with the GPS data queried. MS4 also involved integrating the multi-spectral camera with the existing system. By week 9, we were able to do the first part of the milestone, but not the second part (the reasons for which are listed in Section 4.2). Through our python code, we ensured that the highest resolution images were being saved once every 0.7 seconds and GPS data much faster than that. We also ensured that when these files were transferred to the ground station, we were able to match the timestamps and correlate the image and GPS data.

We verified our testing at Black Mountain Open Space Park where we performed a flight test with live data feed in. We were able to achieve real-time data transfer to the ground station computer as the drone was in flight along with accurate to the meter GPS data information tagged. This completed our milestone for real-time data gathering.

3.2 Milestones for Multispectral Camera Team

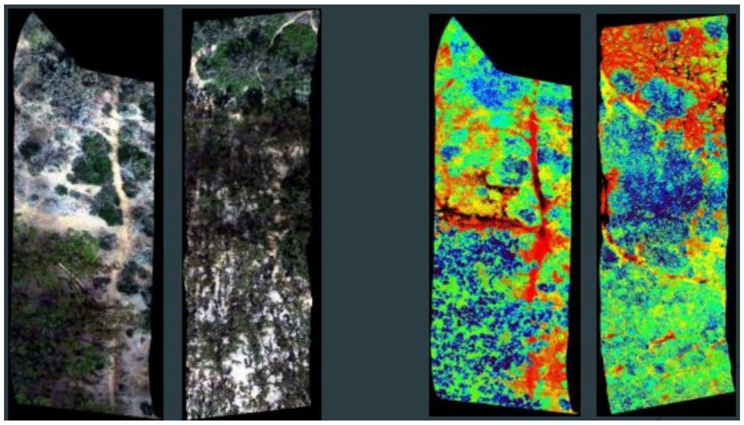
Multispectral camera milestone 1 centered upon research and selection of camera parts best fit for our project. We researched different types of motors, encoders, and control system approaches. Our goals for this project were to select parts which would provide a reliable robust system while still maintaining cheaper costs to maintain advantages over commercial multispectral cameras. For the motor to control the speed of our spinning IR filter wheel, we chose a DC brushless motor with built-in encoder due to its cheap cost and integrated ability to control speed. DC brushless motors provide more reliability and precision over brushed motors, and we found one for a cheap cost and integrated encoder output to make up for DC brushless motors' higher cost and more difficult control compared to brushed motors. For encoders we chose an optical gate array sensor over capacitive and magnetic encoders. The encoder is integral in ensuring the camera was triggered correctly in conjunction with the filter wheel and requires high reliability and accuracy. Although optical sensors are typically less accurate and less robust than magnetic and capacitive encoders, we figured optical encoders gave us enough control in addition to cheaper cost and simpler usage. Finally we researched different methods of closed-loop feedback algorithms and found PID control algorithms. We settled on a PD algorithm as we found the proportional and derivative elements gave us enough speed control for our project. We targeted our IR filter wheel to spin at around 1Hz and did not need exactly 1Hz (since the important part was triggering the camera on time and spinning at a fast enough speed to create a stitched map). Therefore we found that both the speed at which 1Hz was reached and the reliability of the system provided by proportional and derivative control were enough compared to the fine-grain tuning provided by the integrative portion. The camera itself was very important and we were unable to settle on a camera. More on those details below.

Multispectral milestone 2 centered upon building a test system upon which we could develop our PD speed control algorithm. Because we knew we wouldn't have our camera parts ready in the beginning of the quarter, especially the camera, we wanted to build a rig with which we could write an algorithm to control the speed. We were able to scrounge up parts and put together a system with two DC motors, two optical encoder gate arrays, an Arduino Uno, and an Adafruit motorshield unit. The Arduino provided an easy platform upon which to learn how to control motors and encoders as well as microcontroller digital inputs and outputs. The motorshield handled all the voltage step-ups and step-downs required for the motors and encoders. Additionally, the optical encoders and DC motors were close to what we would be working with in our final product, so we were able to utilize this test system to develop a practice PD algorithm to control the speed of the spinning IR filter wheel.

Multispectral milestone 3 centered upon implementing our PD speed control algorithm to reach and maintain a target speed for the IR filter wheel. Using the encoder output to our Arduino to determine rotational speed, we generated a script to analyze the current spinning speed and the rate at which the spinning speed was changing to adjust the output to the motor accordingly. The script read encoder values and measured the time to pass one “slit,” or later on filter, in the wheel. We then multiplied this time by the number of slits in the wheel as well as the reduction ratio to calculate the actual speed of the wheel. This speed value was compared to the target speed as well as the rate of change from the last measured speed to calculate a new speed output value to the motor. The difficulty in this part was determining how to weigh the distance to the target speed and the current rate of change of speed when determining a new speed output value.

Multispectral milestone 4 centered on taking our test system and test PD algorithm and porting it to our actual final system. The goal in this milestone was to build the final camera assembly with our chosen parts, and move the closed-loop feedback speed control system from our test system Arduino Uno to a RaspberryPi 3. This was the longest and most important milestone, as it involved finally building the final product we had prepared for. Dom was able to procure a Ximea xiC USB camera to use for the project and we spent some time attempting to connect it to the RaspberryPi 3 and incorporate it into our system using external hardware triggering. But unfortunately we decided last minute that we did not want to purchase the $3000 camera given the usability and specs for the price of the camera. Because of this we were left without the final product camera and tried to attach a separate camera to use in our system. This second camera (See3Cam USB camera) also had a difficult software interface, and ultimately we were unable to implement hardware-triggering in the short amount of time we had left. This last-minute camera change, in addition to the inability to produce a working IR filter wheel, prevented us from completing the final multispectral camera system. But we were able to produce a proof of concept test system.

Multispectral milestone 5 centered on combining our complete multispectral camera with the systems integration team and attach it to the UAV system. Ultimately we could not attempt this milestone as we did not complete our final multispectral camera system. Instead, we were able to attach a commercial multispectral camera onto the drone and test the basics of our feedback system to some success seen in the two images below. The photo on the left shows normal camera images, and the picture on the right shows images taken with LWIR filters simultaneously.



4. Conclusion

As humans continue to populate, we often look to expand into uninhabited land or repurpose natural habitats. In both cases, wildlife is at risk as we disturb or even destroy their environment. Laws and regulations require ecological surveys to be performed on habitats before land management projects may proceed, but current methods are oftentimes slow and limited. Ornithologists and experts on the ground must traverse through difficult terrain and inspect for wildlife that spend their entire lives protecting and hiding themselves from predators and threats such as humans. While these ecological surveys are a promising step forward in attempts to protect ever-dwindling wildlife, we believe that we can incorporate recent advances in technology to aid in our struggle to save ecosystems.

We propose an aerial-based UAV bird nest detection approach in our search for bird nests during ornithological surveys. Our UAV solution incorporates multiple sensors and cameras to capture visual and thermal data from the skies. Multispectral sensors acquire additional information to aid bird nest identification by identifying high-probability nesting habitats. Finally an integrated wireless and GPS system aggregates all of the image data captured, geotags the images, and sends the combined information down to the ground team in real-time.

We believe that our solution carries many advantages over current on-the-ground wildlife survey methodologies. Our integrated GPS and imaging UAV solution provides mobility, as drones can explore terrain that would slow down or inhibit experts on the ground. UAVs are less intrusive and damaging to habitats than manual surveys. Most importantly, our integrated solution examines and records large spectrums of data in shorter periods of time. This allows us to explore more ground and revisit areas of interest based on geotagged information in higher levels of detail. Ultimately, this rapid analysis time makes our UAV-based analysis more effective and cheaper as well.

While we believe we have developed a complete and effective solution to ecological surveys, we would still like to look into further improvements to our current integrated system. Such improvements include incorporating higher spec cameras with isolated bands, implementing EMLID dual GPS modules for centimeter GPS accuracy, and bigger antennas/radios for improved flight range and data transfer. In addition, we would also like to look into adapting our system to look for other wildlife such as turtles. This would require a refinement of design and software to tailor towards other wildlife and their habitats, but ultimately we believe our UAV infrastructure will provide immense help towards protecting the world’s habitats, wildlife, and ecosystems..

References

1. Dominique Meyer, Ryan Kastner.
   “Using Unmanned Aerial Systems for Nesting Bird Surveys: A Practical Handbook”.
   April 2017
2. Emlid Reference Manual. https://docs.emlid.com/navio2/
3. Ubiquiti Rocket M2 User Guide. https://help.ubnt.com/hc/en-us