Individual Lab Report #10

Sumit Saxena

Team F: Rescue Rangers

Teammates:

Juncheng Zhang

Karthik Ramachandran

Xiaoyang Liu

April 6, 2017

Contents

1. Individual Progress 1
1.1. Overview 1
1.2.1. Investigate issues with GPS location estimation 1
1.2.2. Outdoor testing and data collection 2
1.2.3. Integration of Signature detection and GPS location estimation algorithm 3
1.2.4. Camera calibration 3
1.2.5. Mount for package drop mechanism and camera 3
2. Challenges 4
3. Teamwork 5
4. Future plans 5

1. Individual Progress

1.1. Overview

During the past two weeks, I worked on the following tasks:

1. Investigate issues with GPS location estimation
2. Outdoor testing and data collection (as a team)
3. Integration of Signature detection and GPS location estimation algorithm
4. Camera calibration
5. Design mount for package drop mechanism and camera

1.2.1. Investigate issues with GPS location estimation

As I discussed in my last ILR, I had observed some pretty bad results from our GPS location estimation algorithm on a test video. I did a thorough review of the whole algorithm and explored many online resources to be sure that the algorithm was correct.

One hypothesis I had in mind was that probably we were not using the flight data correctly. To be specific, I thought the way we were using the reported yaw as the heading was incorrect. I did a lot of online research on this and found that the algorithm was correct. I found the following links very useful to arrive at this conclusion:

http://diydrones.com/profiles/blogs/the-difference-between-heading

http://forum.dji.com/thread-14103-1-1.html

https://developer.dji.com/mobile-sdk/documentation/introduction/flightController_concepts.html

Now, I was sure that the algorithm is correct. I went on to plot the GPS locations on Google Earth to get a better understanding of what was happening. The plot is shown in Figure 1 and summarizes the whole problem very conveniently. We consider three cases of signature GPS location reporting, represented by green, red, and yellow markers, respectively (same colored markers in the figure represent GPS locations for one case). Each case has three types of markers:

a) Actual signature location (<signature>_actx), x = case number
b) Calculated signature location (<signature>_calx), x = case number
c) Drone location for that case (Drone_x) , x = case number

Looking at this plot, I quickly realized what was going wrong. I was the subject 'Human_act3) for the green markers case and I knew from the flight that expected location of the drone for that frame had to be in the opposite direction, somewhere near the star in the figure. I found out that the GPS logs were missing some data. There was no GPS data logged for about 30 seconds for

the green and yellow markers' cases and thus, drone locations reported for those cases were the locations for the closest available timestamps (many seconds away) and thus, incorrect.

Figure 1: Three cases considered for signature GPS location estimation. The three sets of same colored markers (green, red, and yellow) represent the different cases considered

For the red markers' case, for which we had correct GPS data, the estimated signature (mattress in this case) location was found to be only 1 m away from the actual mattress location, hinting that the algorithm was indeed correct. But, we needed to perform more tests.

1.2.2. Outdoor testing and data collection

Though the thorough investigation helped me confirm that the algorithm was not at fault, it brought into light another problem – missing GPS data logs for long durations of time. We decided to do another round of testing to see if the issue was consistent and also to test the accuracy of our GPS location estimation algorithm.

We conducted 8 flights and did not see any GPS logging issues in any of them. But, upon testing the GPS location estimation algorithm on the new data, we still did not get good results. Upon further investigation, we found that the camera recording start time being reported by the video filename was off by about 6 seconds from the actual time. This caused an offset of about 6 seconds between the GPS log and the camera timestamps and thus led to incorrect results.

We now know that we can fix this by connecting the camera to a phone before the flight and syncing the time with the phone. We will have another round of testing using this soon.

1.2.3. Integration of Signature detection and GPS location estimation algorithm

I defined what the outputs of signature detection algorithm should be and how they could be fed into the GPS location estimation algorithm. I defined the output for each signature to be a list of the following sort:

['Frame_id' 'Timestamp' 'x_Pixel_location' 'y_Pixel_location' 'signature_type']

Xiaoyang implemented this part in the signature detection part to output a log file containing this information for all the signatures detected. I implemented the part in the GPS location estimation algorithm which reads the logs created by signature detection algorithm to output GPS locations for each of the signatures in the form:

['Frame_id' 'signature_type' 'Latitude' 'Longitude']

The outputs are dumped in an output file, which will be read by the data processing pipeline. Further work needs to be done narrow down these locations to unique locations.

1.2.4. Camera calibration

The RGB and IR images we get from our FLIR duo camera are of different resolutions and the objects in the corresponding frames seem to be shifted in both x and y directions. Finding the transformation between the two corresponding frames requires calibrating both the sensors to get the respective camera matrices.

Juncheng and I performed RGB sensor calibration using OpenCV but could not find any feasible way to perform IR sensor calibration. We talked with NEA people about this and they will help us calibrate the IR sensor.

1.2.5. Mount for package drop mechanism and camera

Since the time we decided to implement all the sensing on our drone instead of NEA's drone, we had removed the package drop assembly and were just mounting the camera and the microphone on the drone. Now, I have designed a mount on which we will be able to mount both the camera and the package drop assembly together. The SolidWorks model is shown in Figure 2. I took a 3D print of the model as well but it did not turn out to be very good. I may have to print it again.

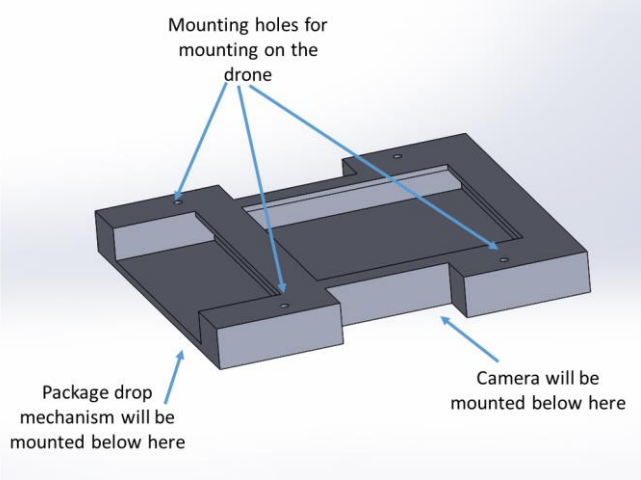

Figure 2: Model of mount for mounting the package drop mechanism and the camera together

2. Challenges

1. Determining the drone's heading type:

As I mentioned in my last ILR, I had found that the drone's yaw has to be taken as the heading. Now, in the GPS location estimation algorithm, I use this heading and the drone's GPS location to estimate the signature location. I am using 'geographiclib' library available for this purpose. This library provides a way to calculate endpoint GPS location given the starting GPS location and magnetic heading.

To be sure of the algorithm's correctness, I had to ensure that the yaw being reported was magnetic heading instead of true heading. It took me a lot of time to confirm this. Reading one of the discussions (link) on the DJI forum ultimately gave me confidence that the reported yaw is indeed the magnetic heading.

2. The issue mentioned in section 1.2.2:

After even confirming that the algorithm was correct, when I found that the algorithm could not give good results on the new test data, it took me some time to figure out the problem. Ultimately, I checked when the flight starts in the video and compared it against the drone's altitude log to find the discrepancy.

3. IR camera calibration:

We could not find a feasible way to do IR camera calibration. We tried applying hot glue from a glue gun to the checkerboard corners and then taking pictures using IR camera. We also tried using a heater to heat a cardboard and taking its images to somehow be able to find a transformation between the RGB and IR images. But, we realized that our methods were either insufficient or incorrect.

3. Teamwork

As a team, we did another round of outdoor testing and data collection, and also discussed how we should integrate different parts.

Work done by individual team members:

- Juncheng Zhang:
  - Migrated human detection algorithm for IR images to python
  - Fixed bugs in the MATLAB implementation of human detection algorithms
- Sumit Saxena:
  - Investigated issues with GPS location estimation
  - Worked on integrating signature detection and GPS location estimation algorithms
  - Worked on Camera calibration
  - Designed mount for package drop mechanism and camera
- Karthik Ramachandran:
  - Worked on data processing pipeline GUI
- Xiaoyang Liu:
  - Modified the signature detection algorithm to account for the offset between the RGB and thermal images
  - Added functionality to output a signature detection log

4. Future plans

Following are the tasks I plan to work on until the next PR:

1. Rigorously test the signature GPS location estimation algorithm
2. Camera calibration
3. System integration
4. Fabricate/print mounting for camera and package drop mechanism
5. Try to implement Faster R-CNN for human detection