Individual lab report #9

Mar,23rd, 2017

Xiaoyang Liu

Team F Rescue Rangers

Juncheng Zhang (Henry)

Karthik Ramachandran

Sumit Saxena

Individual Progress

- Collect image and sound data from Matrice100 in NREC
- Modify the integration algorithm for RGB and thermal detection
- Test the modified algorithm on self-collected data

Collect image and sound data from Matrice100 in NREC

We collected data in an open area in National Robotics Engineering Center. Figure 1 shows the exact location where we did all the flights. All the details of settings are listed below and Figure 2 give us an example of the image data we collected in NREC. In total we collected 30 3-min videos and several recording examples by using our own drone.

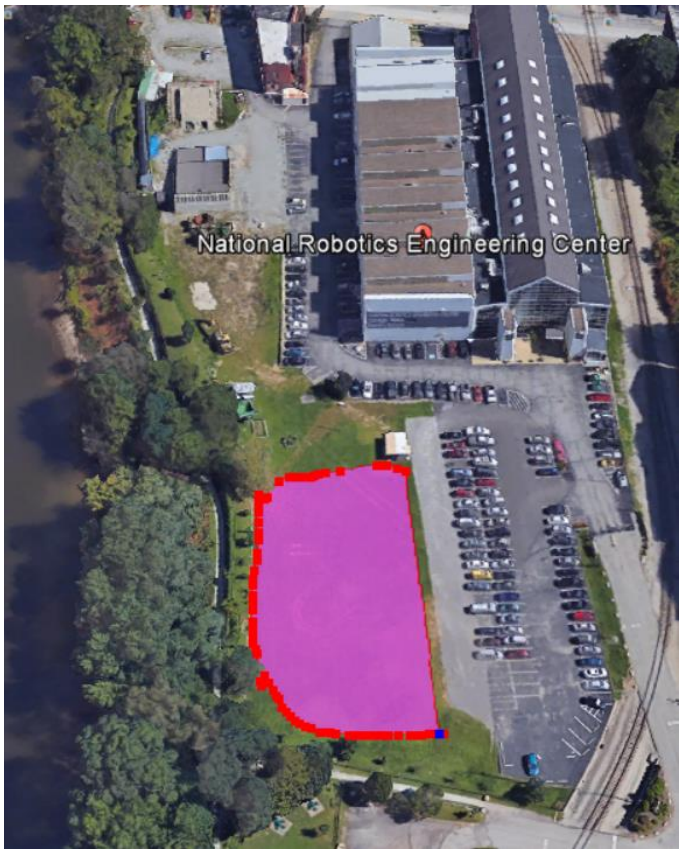

Figure.1 Testing site location

Visual part setting up:

- FLIR-Duo camera
- Matrice100
- 2 bright color mattresses.

- 2 containers with heated water.
- 3 human figures.
- 1 bag

Sound part setting up:

- Microphone
- Matrice100

Figure.2 Image example taken by FLIR-Duo

Modify the integration algorithm for RGB and thermal detection

Last time, I implemented a weight updating integration algorithm**.** The main idea of it is to combine the classification result of both algorithms, trying to add more weight to those bounding boxes which is classified to be human by both algorithms, meanwhile reduce the weight of those which are determined to have no humans by both algorithms. The details of the algorithm are shown below.

Steps:

a) Map the thermal bounding boxes to RGB images
b) Set some weights on each of the positive and negative bounding boxes for both the RGB and thermal images.
c) Calculate the overlapped area of RGB and thermal bounding boxes with humans. Add weights on corresponding positive bounding boxes based on the different overlapping situations.
d) Designate a threshold for choosing the bounding boxes whose weight is over than it, then count those bounding boxes as ROIs with human and return them.

Description on all the weights and threshold

Table.1 Threshold settings in the previous algorithm

Weight Parameter	Description	Value
α1	RGB positive	3
α2	RGB negative	-1
α3	Thermal positive	2
α4	Thermal negative	-2
α5	RGB positive overlaps Thermal positive	5
α6	RGB negative overlaps Thermal positive	-1
α7	RGB positive overlaps Thermal negative	-2

Threshold Parameter	Description	Value
θ1	% of RGB positive overlaps Thermal positive	0.4
θ2	% of RGB negative overlaps Thermal positive	0.4
θ3	% of RGB positive overlaps Thermal negative	0.4
θ_weight	Choose the bounding boxes above this threshold to be human locations	1 - 3

However, the real case is that most of the ROIs coming from RGB or thermal algorithms will not overlap. So, it seems that the weight updating rules for the integration is not that useful. Changing the threshold can only determine which algorithm to be chosen but not to increase the ability of detecting humans or to eliminating false positives. We need a better way to combine the two algorithms

Modification:

1. Combine ROIs from both algorithms
2. Classify all ROIs by both RGB and thermal classifiers
3. Choose those bounding boxes classified as humans by both of the algorithms
4. Integrate intensity threshold algorithm into the thermal system

In other words, we used **OR** for choosing ROIs and **AND** for determining human bounding boxes. In this way, we get a bigger chance of considering potential human candidates and smaller chance of getting false positives.

Input and Output:

Input:

RGB and Thermal videos taken simultaneously.

Output:

1. Coordinates of human bounding boxes in each frame
2. Timestamps which are in accordance with the bounding boxes
3. Images with human bounding boxes.

This time, I consider more about how to design my algorithm so that it can fit into our whole end to end system. The input should be RGB and Thermal videos which will be converted into frames of images later before entering the integration algorithm. The output should be the coordinates of human bounding boxes in each frame. Since the next step is to calculate the absolute GPS locations humans, we also need to output the timestamps which are in accordance with the bounding boxes (as shown in figure 3). We can get the GPS locations of our drones through DJI SDK, which is corresponding to certain timestamp. By matching the output timestamps and DJI timestamps, we will be able to get a one-to-one relationship between the GPS location of our drone and the human bounding boxes. Finally, we can calculate the exact GPS locations of detected humans.

	1	2	3	4	
1	164	1	17	32	11:44:33.100
2	163	1	20	38	11:44:33.200
3	160	1	23	46	11:44:33.300
4	158	1	27	54	11:44:33.400
5	159	4	25	54	11:44:33.500
6	159	11	25	54	11:44:33.600
7	159	11	25	54	11:44:33.700
8	159	17	26	56	11:44:33.800
9	159	24	26	56	11:44:33.900
10	158	31	26	57	11:44:34.000

Figure.3 Timestamps of human bounding boxes

Tests on self-collected data

Video

I've test the modified integration algorithm on our new dataset. However, we didn't change the training set. So hopefully after adding the training set, the performance would be better. The video below shows the result of RGB and thermal integration with thermal intensity threshold. We haven't added other signature detection algorithm in RGB yet.

Challenge

1. RGB classifier seems not to be very reliable. It continuously detects grass as human when the grass reflects strong sunlight. We will see the result again after adding our new examples into the training set.
2. When mapping thermal bounding boxes to RGB image or vice versa, the bounding boxes shifted a lot. We have to develop a warp function for shifted bounding boxes.

Team Work

In this week, we collected RGB, thermal and sound data together. Individually, Karthik worked on sound detection by using the new microphone which is hanged up to the drone. Sumit mainly focused on testing other signature detection on new data and testing signature GPS location reporting algorithm. Juncheng worked on migrating all the RGB and thermal codes from Matlab to Python.

Future Work

Before the next progress review, we should have a complete end to end software running all of our algorithms together. So, Juncheng will still work on migration the visual part to python. To train on our own data is very time consuming, not sure we can achieve it before finishing the software. I will work on training data and add other signature detection part in the RGB algorithm.

Also, I want to explore on deep learning methods of object detection which might be useful in the future. Sumit will work on finetuning the GPS location reporting algorithm and integrate his code into our pipeline. Karthik will work on CNN and integrate all the codes into the pipeline.