Individual lab report #10

Apr, 6th, 2017

Xiaoyang Liu

Team F Rescue Rangers

Juncheng Zhang (Henry)

Karthik Ramachandran

Sumit Saxena

Individual Progress

- Investigate and Resolve issues with bounding boxes on RGB/Thermal
- Add multiple signature detection in the integration algorithm
- Assist on building the whole pipeline

Investigate and Resolve issues with bounding boxes on RGB/Thermal

Last time, I found out 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. And we need a better way to combine two algorithms. So, we decided to do some modifications, and thus we can get a more accurate classification result.

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 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.

However, during the process of combining ROIs from both algorithms, we have to map RGB ROIs to Thermal images, and vice versa. There exists shifts in both x direction and y direction. Even if the RGB image is cropped and resize to be as the same size of the thermal image, the drift is still a significant obstacle in our case. As you can see in figure 1, this pair of images are collected at the same time, but the bounding box of human drift a lot in both images.

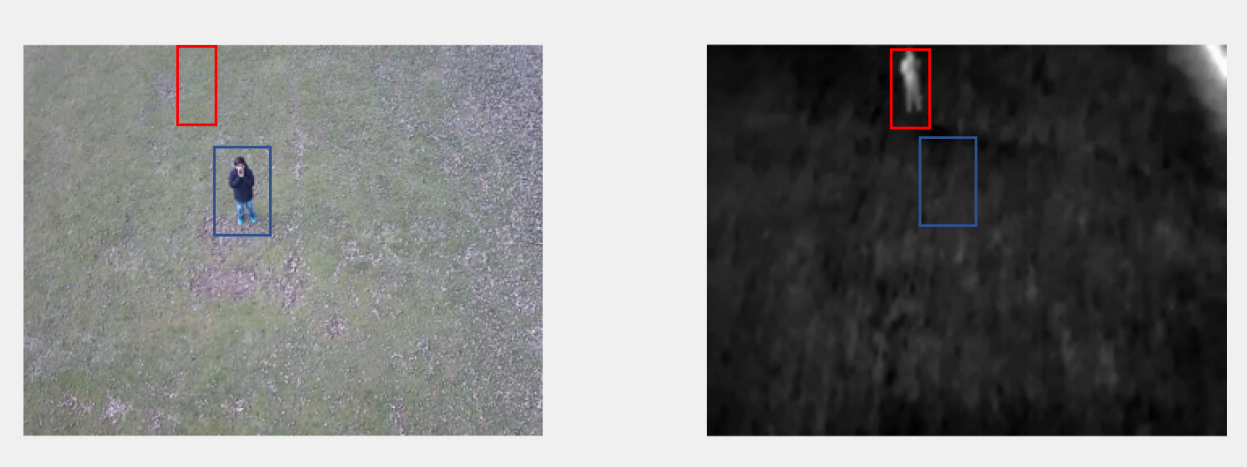

Figure.1 Drift exists in one pair of images

At first, we want to find the warp function between two image. Since the extrinsic and intrinsic camera matrix does not change for the two cameras, so that theoretically we can find the map relationship by computing a transferring matrix between two images. However, the fact is that right now we can only manually choose limited corresponding points which are not accurate enough, and the thermal images are kind of distorted. So, it's very hard to compute a united warp function for all images.

After that, we found out that the drift in both images are almost constant. So, the temporary strategy to solve this problem is to set the drift based on different height. Even though there might still be some drifting errors, we develop a mitigation policy which can maintain original bounding boxes if the mapped bounding boxes from the other image have a certain degree of overlap with the original bounding box.

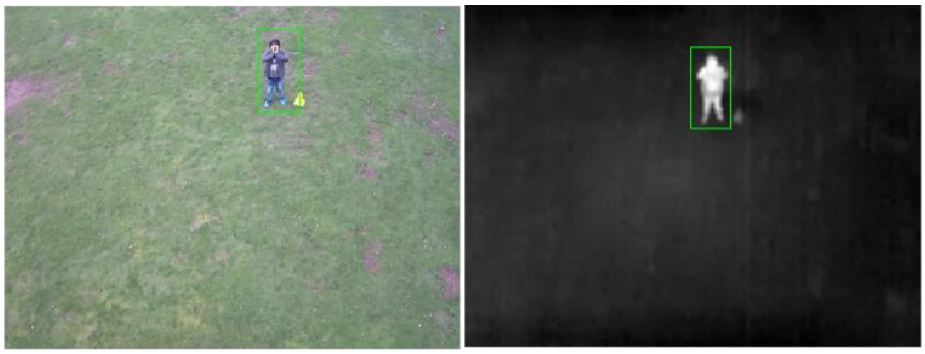

Figure.2 The result shows the bounding box after applying the shift mitigate algorithm

Add multiple signature detection in the integration algorithm

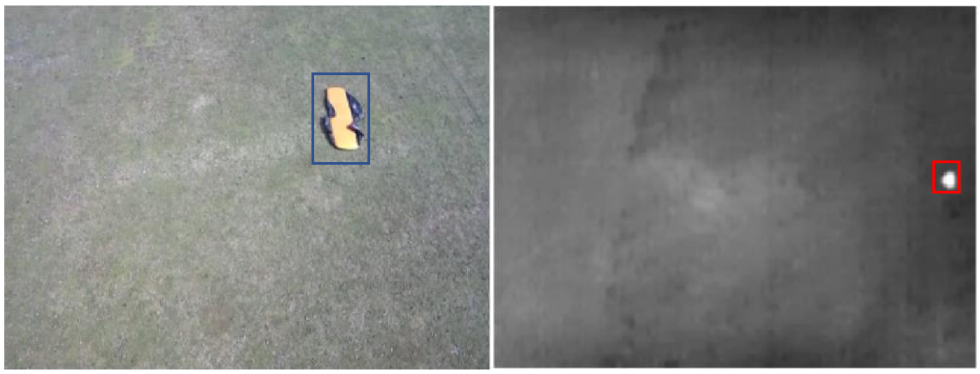

Figure.3 Bright objects in RGB image and high intensity object in thermal image

In addition, we want to detect some other signatures (mattresses, hot kettle, tents, etc.) except for humans. In this way, we will be able to know places where humans are more likely to appear.

Therefore, this time I add the algorithm which can detect bright objects in RGB image and high intensity object in thermal image into the integrated object detection system. So, after combining the thresholding algorithm:

In RGB images:

- We are now able to detect other objects except for humans

In thermal images:

- We can find out the object with high intensity which can be used to eliminate false positives in thermal detection algorithms. To illustrate, we only output the bounding boxes which also belong to high intensity objects in thermal images.

Assist on building the whole pipeline

1) Generate systematic output for the whole pipeline
   In order to do our end to end test, we need the integrated signature detection system to generate an output file in a format that can be fed to the next GPS estimation system. After discussing the conversion of the two systems, we decided to design the output to contain the name of output image, the timestamp which corresponds to each output image, the pixel location of detected signature, and the type of signature ('H' represents for human, and 'B' represents for bright objects). You can see the details in figure.3

```
RGB/00145.bmp 11:24:02.133 1524 63 B
RGB/00146.bmp 11:24:02.167 1545 84 B
RGB/00147.bmp 11:24:02.200 1578 122 B
RGB/00148.bmp 11:24:02.233 1611 160 B
RGB/00149.bmp 11:24:02.267 1632 198 B
RGB/00150.bmp 11:24:02.300 1642 238 B
RGB/00151.bmp 11:24:02.333 1645 267 B
RGB/00152.bmp 11:24:02.367 1589 356 H
RGB/00152.bmp 11:24:02.367 1663 312 B
RGB/00153.bmp 11:24:02.400 1602 396 H
RGB/00153.bmp 11:24:02.400 1667 345 B
RGB/00154.bmp 11:24:02.433 1607 414 H
RGB/00154.bmp 11:24:02.433 1658 369 B
RGB/00155.bmp 11:24:02.467 1635 392 B
RGB/00156.bmp 11:24:02.500 1612 410 B
RGB/00157.bmp 11:24:02.533 1532 392 H
```

Figure.3 Output text file of signature detection system

2) Run Matlab in Python code

```python
import subprocess
file = 'C:\\Users\\xiaoy\\Desktop\\RGB+thermal\\Integration\\readVideo.m'
cmd = "run('" + file + "');"
print cmd



subprocess.Popen(['matlab', '-nodisplay', '-nosplash', '-nodesktop', '-r',
    cmd])
```

Figure.4 code for running Matlab script in Python

Since our software for running the pipeline is written in Python, we need to find a way to fun Matlab code in Python project.

Step 1:

Add Matlab installation path into the environmental variables

Step 2:

Run the code in Figure 4, and the only thing needs to be changed is the “file” path which includes your Matlab projects

Challenge

The biggest challenge this week is to find the warp function to map thermal image back to RGB coordinates. We only find a temporary way to solve the drifting problem right now. But to make our algorithm more robust, we still have to explore on the calibration of RGB and thermal camera so that we can get a correct warp function.

Team Work

In this week, Juncheng (Henry) devoted most of his time migrating Matlab code to Python, and he also helped me to explore on the drifting problem. Karthik worked on YOLO and building the software to integrate all systems together. Sumit mainly focused on refining test signature GPS location reporting algorithm and integrate GPS estimation with Signature detection.

Future Work

Before the next progress review, we need to test our end-to-end system in NREC and see the overall performance of the whole system. We should get prepared for the SVE.