Team F: Rescue Rangers

Juncheng Zhang

Teammates: Karthik Ramachandran
Sumit Saxena
Xiaoyang Liu

ILR08

3/02/2017

1、Individual Progress

1.1、Overview

During the past few weeks, my primary role was exploring the tracking method for human beings in aerial videos. I tried to analyze three different tracking methods, including Lucas-Kanade tracker with template update, KLT feature tracker, and Kalman Filter tracker.

Also, I implemented the first two methods on a online video taken from the drone and tried to track a human being inside. I compared these two methods and discussed their feasibility on our use case.

1.2、Exploration of tracking methods in aerial videos

Lucas-Kanade tracker basically performs template tracking between movie frames based on optical flow[1]. It assumes that the optical flow is essentially constant in a local neighbourhood of the pixel under consideration, and solves the basic optical flow equations for all the pixels in that neighbourhood, by the least squares criterion. However, it may have the problem of drifting due to the accumulated error of each frame. Even though the drifting could be somewhat migrated using template update methods[2], the occlusion problem is hard to solve for the LK tracker since it tracks the object based on the whole template. In another word, if humans are occluded in the template, it is very likely that the tracker will lose the human in the next frames.

KLT feature tracker makes use of spatial intensity information to direct the search for the position that yields the best match. One of its main advantage is that it is faster than traditional techniques for examining far fewer potential matches between the images. Also, it deals with occlusion much better than Lucas-Kanade tracker because instead of tracking the whole template, it only tracks several feature points and their neighbourhoods. However, as the KLT tracker progresses over time, points can be lost due to lighting variation, out of plane rotation, or articulated motion. In addition, KLT tracker doesn't perform as well as LK tracker does when the tracking object is moving fast.

Kalman filter-based tracker can predict object's future location and reduce the noise introduced by inaccurate detections [3]. In addition, it is very helpful to the data association when dealing with multiple object tracking. However, in our use case, since the camera is also moving, it is extremely hard to build the mathematical model for the tracking object in the image. Consequently, we don't plan to use it in our application until we find other ways to address this issue.

1.3、Implementation of tracking methods in aerial videos

I implemented the LK tracker using the template update method described in [4], and the KLT tracker based on the corner features. The video clip comparing these two methods is in the following link:
https://drive.google.com/file/d/0B19Cauta5rmFUEo1RlROMTVCc3c/view?usp=sharing

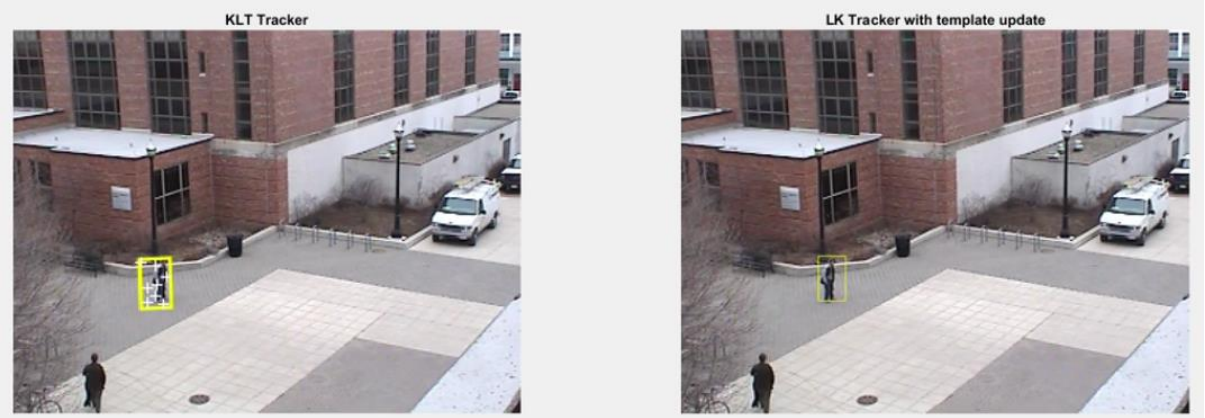

Figure1 Beginning of the tracking

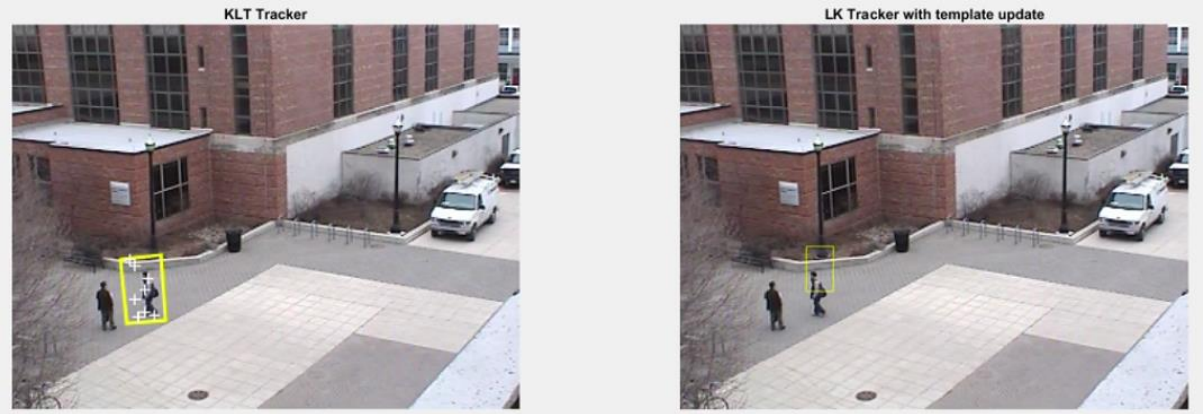

Figure2 Middle of the tracking

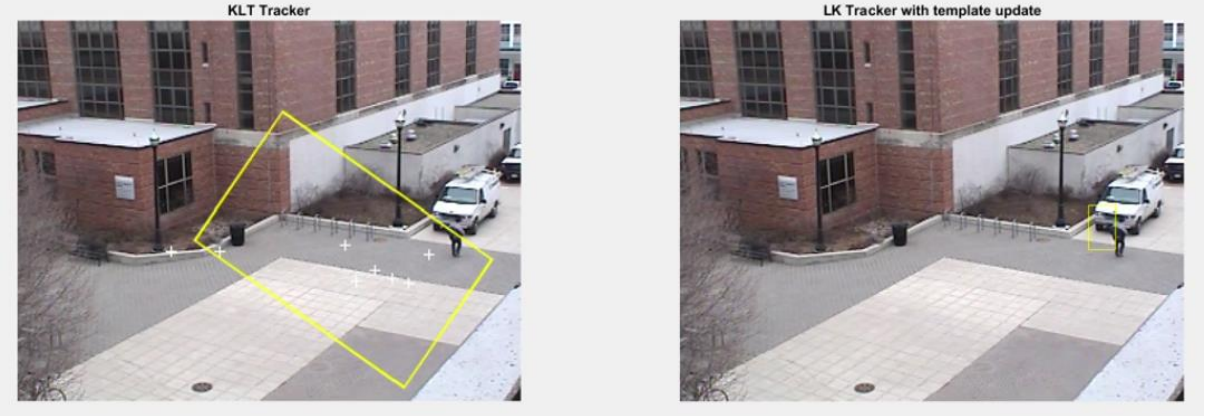

Figure3 End of the tracking

From the figures above, we can see that both trackers don't perform well in the long-term tracking. The LK tracker will have the drifting problem during the tracking, while the KLT tracker will lose the feature points during the process. In the end of tracking, the LK tracker fails because the huge truck influences the template matching, and KLT tracker is not able to track the human when he starts to move in a fast speed.

However, since our application only requires to track the detected human beings in the next several frames to help with the detection and prune the false positives, we don't need the tracker to work in the long tracking process. We will evaluate these two tracking methods again after we collect our own video and implement them on that.

2、Challenges

The main challenge I faced last week was to implement the KLT tracker based on the existing code from Computer Vision Toolbox in Matlab. The example of KLT tracker in Matlab is face detection and tracking [5]. This open source example gives me the basic idea about how to apply KLT tracker in the real case. However, since most codes are hidden in one function, it is hard for me to understand every step of this algorithm and customize it so that it can be used in my application. Finally, after thoroughly understanding the algorithm based on comments and related literatures, I am able to apply the KLT tracker on tracking the human beings in aerial videos.

3、TeamWork

After the previous progress review, our team discussed the plan for the next few weeks, and broke the work down as follows:

Table1 Work distribution form

Member	Work
Karthik Ramachandran	Initial work on data processing pipeline
Sumit Saxena	Triangulating GPS location of signatures
Juncheng Zhang	Explore tracking methods in aerial videos
Xiaoyang Liu	Integration of RGB and thermal signature detection algorithm

The team worked with great coordination during execution of the entire task. We worked on different components of our whole system separately, and each of us did a good job.

4、Future Plans

Before the next progress review, we plan to have a working system based on the data processing pipeline that Karthik has built. In order to achieve that, we will mount our Flir DUO camera and microphone on the drone and collect data at NREC, and implement the whole algorithm based on those data.

For my personal task, I will start to migrate the current Matlab code to the Python code so that it can be run in our processing pipeline. If I still have time after this task, I plan to further explore the tracking method and test it using our collected data.

5、Reference

[1]. S. Baker et Al. "Lucas-Kanade 20 Years On: A Unifying Framework"

[2]. D. Schreiber, "Robust template tracking with drift correction"

[3]. http://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html

[4]. https://www.ri.cmu.edu/publication_view.html?pub_id=4433

[5].https://www.mathworks.com/help/vision/examples/face-detection-and-tracking-using-the-klt-algorithm.html