Google Earth Visualization of 3D Trajectory Reconstruction of Flying Objects

Zhanar Osmonaliev, Young Won Choi
Department of Computer Science, State University of New York Korea - Stony Brook, Songdo, Incheon 21985,
zhanar.osmonaliev@stonybrook.edu, youngwon.choi@stonybrook.edu

This paper aims to provide an analysis of our implementation of a project presented by Jingtong Li and Jesse Murray, supervised by Cenek Albl and Konrad Schindler in the group of Photogrammetry and Remote Sensing, at ETH Zurich. Li et.al's method reconstructs the 3D trajectory of drones (despite unfavorable conditions) and our project builds on their method by implementing our own object tracking and conversion of the flying trajectories to KML for visualization on Google Earth. (Li et al. 2020)

Key words: feature detection, KML, Google Earth, drone

---

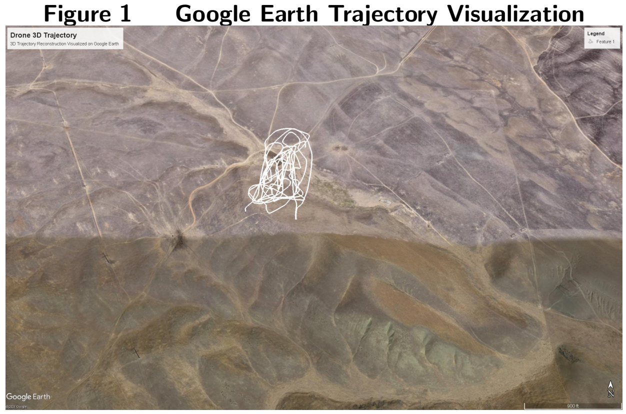
1. Introduction

Not too long ago, expensive cameras - and synchronization of cameras - were highly recommended to obtain quality data from computer vision projects. However, advancements in computer vision have allowed us to use everyday equipment to do the same work and furthermore visualize the data. Tracking objects on multiple cameras can be especially challenging. For example, feature detection mechanisms must be able to efficiently track an object - or multiple objects - across frames and withstand transformations or occlusion. Li et.al proposed a robust method that reconstructs the3D trajectory of flying objects (drones in this example) with inexpensive equipment and unsynchronized cameras. This paper aims to present and discuss our feature detection implementation of the data set provided by Li et.al and the visualization of the results through KML onGoogle Earth. (Li et al. 2020)

2. KML Visualization

Keyhole Markup Language, or KML, was developed by Google to display geographic data on applications such as Google Earth. One of the main features of KML is marking geographic locations on earth using longitude, latitude, and if needed, altitude. Our project takes trajectories from our detection sets and converts the grid coordinates into KML by adding them to the main coordinates on a fixed point, mainly 37 degrees North and 121 West. In order to accommodate for robustness, we adjusted the conversion taking the earth's curvature into account.

Figure 1 Google Earth Trajectory Visualization



3. Feature Detection

Unlike previous works on drone detection implemented in the CSE327 course, where detection of the drone was being done manually by selecting the bounding box, our implementation features object detection. The object detection is implemented with Haar Feature-based Cascade Classifiers which was initially proposed by P. Viola and M. Jones in 2001. The detection itself is performed with pre-trained model developed by M. Pawelczyk and M. Wojtyra (https://github.com/Maciullo/DroneDetectionDataset). Their model was trained with aim to detect a drone in different environments and was exported as a Haar Cascade XML file.

4. Object Tracking

Once the drone is detected, the tracking is performed with OpenCV's tracking library. Among various trackers, a Channel and Spatial Reliability of Discriminative Correlation Tracker (CSRT) is used. Since speed performance is not a priority, it was decided to use a CSRT tracker for better tracking accuracy. An alternative tracker in consideration was a Kernelized Correlation Tracker (KCF), however, this tracker did not handle the occlusion well. Other tracking algorithms were either not available in OpenCV 3+ or did not meet our standards for tracking the drone.

5. Challenges and Further Work

5.1. Challenges

One of the greatest challenges in our project was to solve the problem of occlusion. Whenever the drone approached a building and went behind it, our tracker would stop detecting the drone altogether and did not resume tracking once the drone was visible again. One of the attempted solutions for this issue was to implement a Kalman Filter, which would help us predict the direction of the drone during occlusion. However, we were not able to successfully incorporate the filter.

5.2. Further Work

We believe that with more time, we could implement the Kalman filter by removing the background using cv2.backgroundsubtractormog which uses a Gaussian Mixture Model.

6. Conclusions

Drone tracking implemented purely with Computer Vision technology offers a low-maintenance and low-cost solution. However, drone tracking based purely on imagery data has some inherent caveats. Due to the small size of the drone, tracking results can deviate from the ground truth. Additionally, improper model training can result in erroneous detections of birds or other flying objects. Nevertheless, it is not to say that the technology is not worth developing. In turn, in combination with other data sensors, a Computer Vision-based solution can offers additional information which might give a new perspective on drone movement analysis.

References

Jingtong Li, Jesse Murray, Dorina Ismaili, Konrad Schindler, Cenek Albl (2020) Reconstruction of 3D flight trajectories from ad-hoc camera networks. Retrieved June 14, 2021, https://arxiv.org/abs/2003.04784

Google Developers. Keyhole Markup Language. Retrieved June 14, 2021, https://developers.google.com/kml/documentation/kml