Visual odometry with and without object detection

Abstract

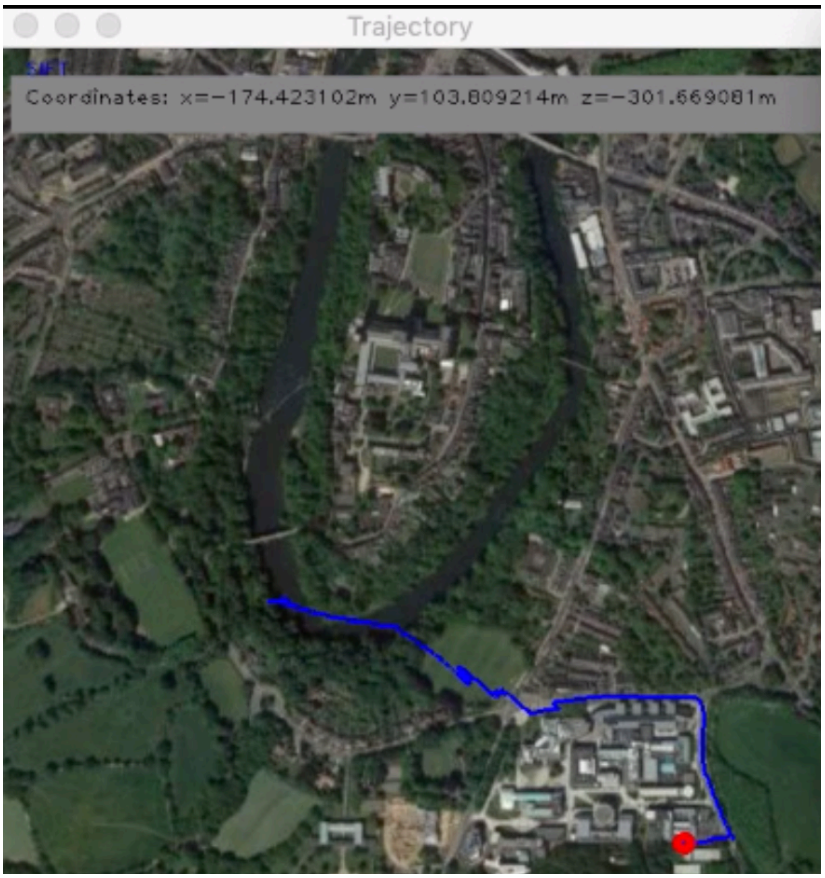
In this paper we attempted to implement a visual odometry system with and without considering dynamic objects in the scene. There has been a great improvement when using object detection however the results are still far from the ground truth. The biggest challenge happened when the vehicle turned left or right, as these turns were not accurate in the VO trajectory compared to the ground truth.

Our approach

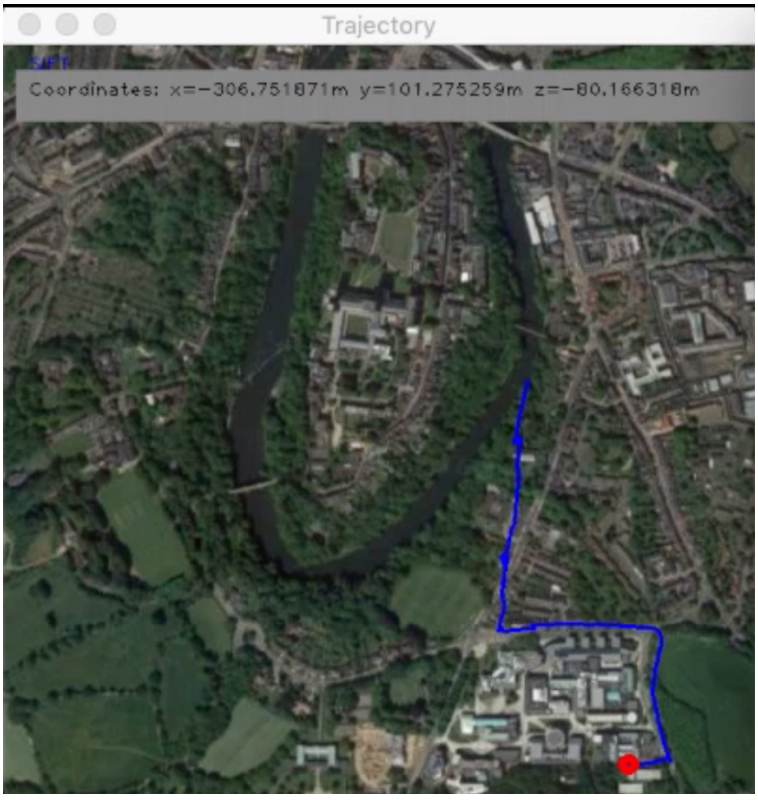
In our approach we modified a simple existing monocular VO obtained from github *(https://github.com/uoip/monoVO-python)* to fit our dataset. In addition to the existing FAST feature point detection, we used SIFT as well. To run our code, run the file *mono_stream.py* in command line. To change the parameters, open the code and see these parameters on the top.

Due to time limit we did not manage to derive an accurate ground truth trajectory from GPS/IMU, however we designed the code that if it is started without any skipping forward, then the Durham satellite picture will be the background of the trajectory mapping, where the red line indicates the ground truth, and the blue one the VO trajectory.

Below one can see the results without using the object detection from *yolo.py:*



And the next figure shows the result using the dynamic object detection method:



As one can see the results are much better compared to one another, however turns are still not accurate (straight driving causes no problem). Further fine tuning of the VO code could resolve these issues with curves. As seen above, we used the satellite map of Durham as our ground truth.

In the video attached, additionally to the detected objects by yolo, we added an extra box as a permanent object in the scene, namely the front of our car. This is because before adding it, our code sometimes detected feature points in this area, which just makes the data noisier, as the reflection on this part of the car was high.