Project 1
Visual Odometry

[ENN583] Foundations of Robotic Vision

Cameron Stroud (n11552123)

Abstract

A visual odometry pipeline has been implemented, based on ORB feature detection with brute force matching and 2D-3D correspondence. Performance issues are present with mitigation strategies presented. An evaluation has been performed against a publically available visual odometry system alternative.

1 Introduction

With the advent of autonomous vehicles comes a revolution of the transport industry. This paper will investigate an implementation of a visual odometry pipeline for use in taxi and delivery services in urban environments. The contents include a design overview, evaluation of performance, and further case study against an existing solution. The focus is on providing a functional, baseline visual odometry to better understand the effects of the subsystems, and as such performance in the presence of noise is outside the scope.

2 Method

2.1 Feature Detection

The visual odometry pipeline implements ORB as the chosen feature detector. ORB provides invariance to both rotation and scale in its application of the FAST detector, and provides unique descriptors through BRIEF for repeatability, all while maintaining decent computational speed, making it a solid choice for real-time visual odometry.

2.2 Feature Matching

Feature matching just applies a basic brute force with Hamming distance, applicable due to the BRIEF descriptors output by the ORB detector.

2.3 Motion Estimation

To estimate motion, the visual odometry pipeline applies a 2D-3D correspondence approach. This gains the advantage over 3D-3D correspondence of minimizing reprojection error, resulting in a more accurate estimation [1].

At time i, a disparity map is generated across the stereo pair of I_i and converted into a depth map. The matched $I_{i-1} \rightarrow I_i$ feature points are then converted into 3D points (ie. non-homogenous) and passed into OpenCV's `solvePnPRansac` function to gain the translation and rotation (the output rotation vector is passed through Rodrigues' equation to get the SO3 representation).

2.4 Outlier Removal

The outlier removal present in the visual odometry pipeline is very basic. Firstly, at time i, the $I_{i-1} \rightarrow I_i$ matched features are sorted by distance and the top 15 features are used as a means to limit outlier matches. Secondly, when points are triangulated and converted to 3D coordinates, points above a certain depth are ignored to mitigate the degeneration of stereo to monocular triangulation. A max depth value of 500 is used.

2.5 Local Optimisation

No local optimisation is currently applied as part of the visual odometry pipeline.

2.6 Strengths/Weaknesses

While there are fast solution for feature detection and tracking, ORB provides a solid blend of efficiency and invariance to adverse conditions. The model is expected to output high accuracy measurements at reasonably high speeds. However, without any form of local optimisation the resulting pose graph will inevitably drift from the ground truth, as no corrective measures are taken in the presence of measurement error.

3 Evaluation

Evaluation is performed on the KITTI residential set, as there is less dynamic activity compared to the city and highway sets, while maintaining an urban environment where taxi and delivery services are performed. The city provides many scenarios with dynamic objects, and while the outlier removal should mitigate the effects of dynamic objects, there is too much noise for such a primitive odometry system to deal with. Likewise, the highway set has dynamic objects more easily identified as inliers, as the relative position of other vehicles remains consistent with the camera, but vary enough to result in a lot of noise with scale. The specific sets being investigated are drives '0035' and '0061', selected for their lower sizes based on the constraint of restricted compute power and non-iterative data loading.

3.1 Residential Drive '0035'

Mean Processing Time: 0.04726581940284142

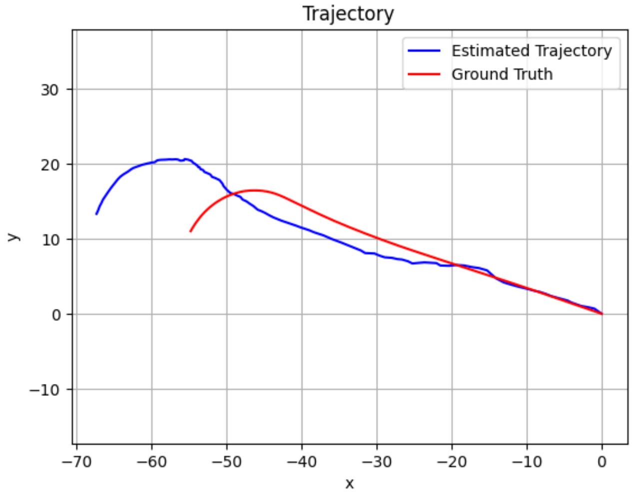
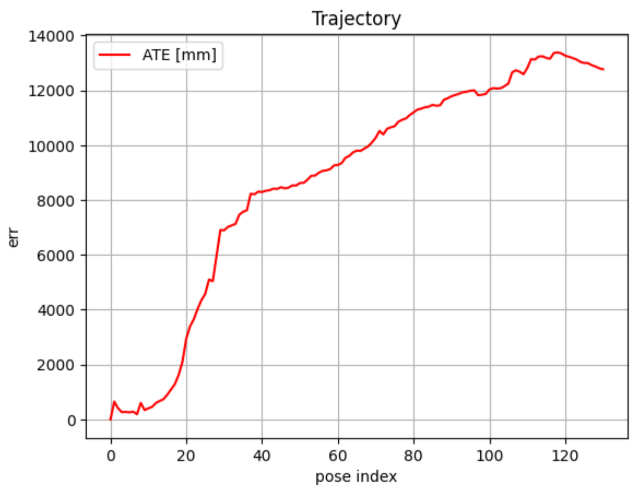
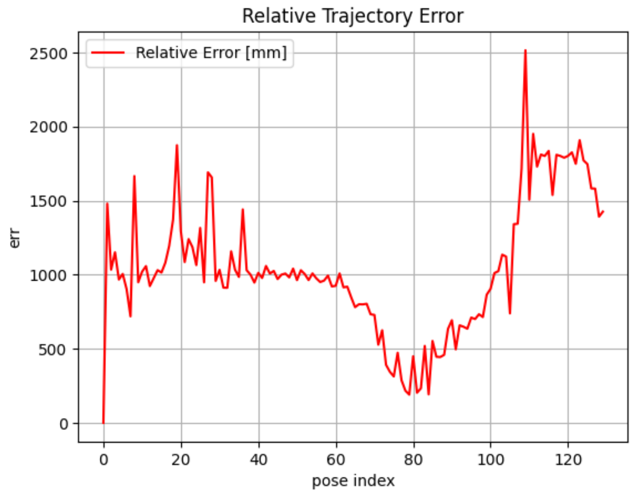


Figure 1: Residential Drive '0035' Trajectory Estimation vs Ground Truth



(a) Absolute Error



(b) Relative Error

Figure 2: Residential Drive '0035' Error Plots

Err Type	ATE	Relative
Mean:	8681.64 mm	1038.23 mm
Median:	9716.31 mm	1000.01 mm
Max:	13348.60 mm	2515.24 mm
RMSE:	6801.76 mm	1077.93 mm

Table 1: Residential Drive '0035' Performance

3.2 Residential Drive '0061'

Mean Processing Time: 0.054439835398964735

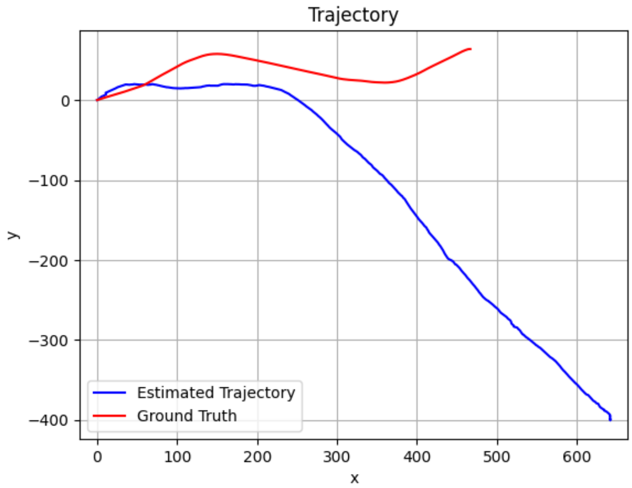
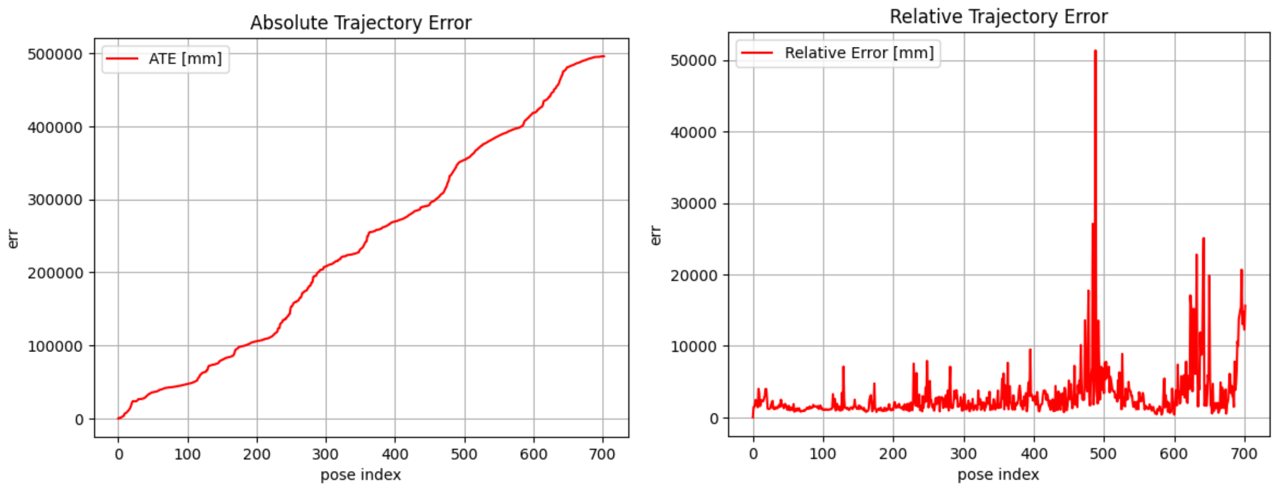


Figure 3: Residential Drive '0061' Trajectory Estimation vs Ground Truth



(a) Absolute Error

(b) Relative Error

Figure 4: Residential Drive '0061' Error Plots

Err Type	ATE	Relative
Mean:	237091.75 mm	2969.49 mm
Median:	232321.94 mm	1837.62 mm
Max:	496151.25 mm	51333.24 mm
RMSE:	140544.81 mm	8817.90 mm

Table 2: Residential Drive '0061' Performance

3.3 Summary

The odometry system works reasonably well within '0035' but fails drastically in '0061'. The overall trend of the trajectory is reflected within the estimates in '0035', with some drift and overshoot. The minor drift associated with '0035' could be an indicator of insufficient outlier rejection, while the overshoot seems to reveal issues with correct representation of scale.
With no baseline for typical processing times, the resulting time does not inform much about the efficiency of the pipeline. It does seem to be on the correct order of magnitude for real-time processing, being in the tens-of milliseconds range.

4 Experimentation

Performance of the implemented visual odometry system will be contrasted with an implementation by Nico Nielsen [2]. Nielsen applies FAST feature detection directly, in conjunction with OpenCV's `calcOpticalFlowPyrLK` for tracking. There is also no local optimisation implemented, and outlier removal is applied in much the same way of sorting keypoints and selecting the top performing, 10 in this instance.

4.1 Nielsen: Residential Drive '0035' Performance

Processing Rate: 12.15poses/s

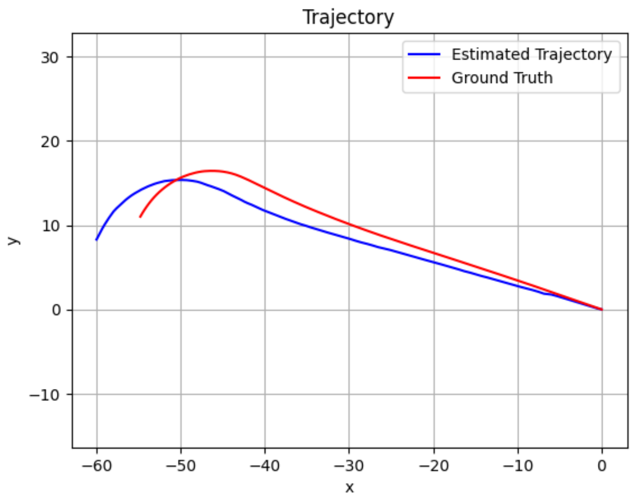
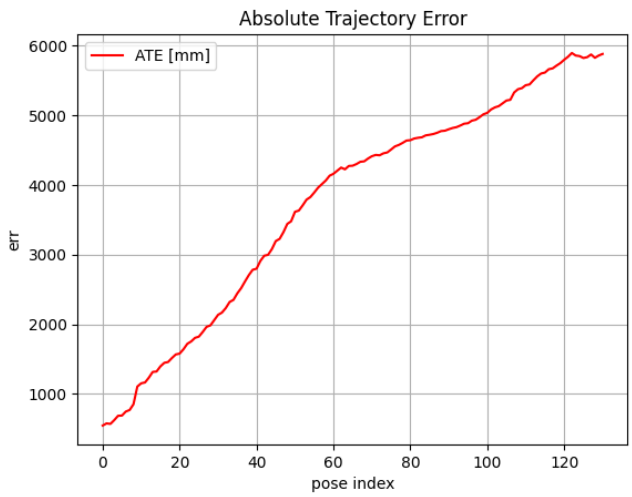
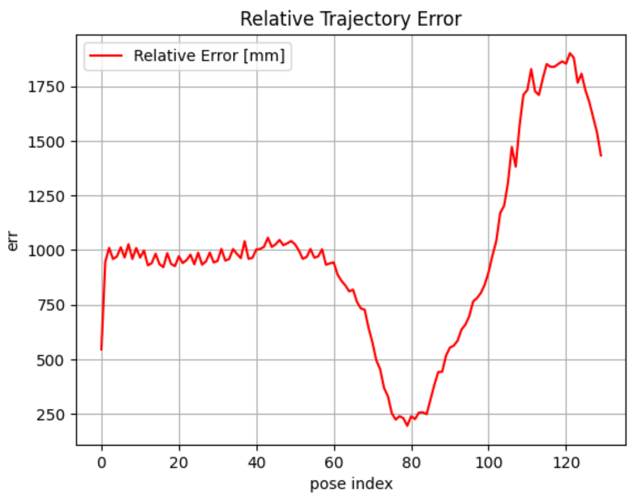


Figure 5: Nielsen Residential Drive '0035' Trajectory Estimation vs Ground Truth



(a) Absolute Error



(b) Relative Error

Figure 6: Nielsen Residential Drive '0035' Error Plots

Err Type	ATE	Relative
Mean:	3734.59 mm	983.52 mm
Median:	4274.20 mm	963.48 mm
Max:	5891.72 mm	1900.43 mm
RMSE:	2035.62 mm	967.31 mm

Table 3: Residential Drive '0035' Performance

4.2 Residential Drive '0061'

Processing Rate: 12.35poses/s

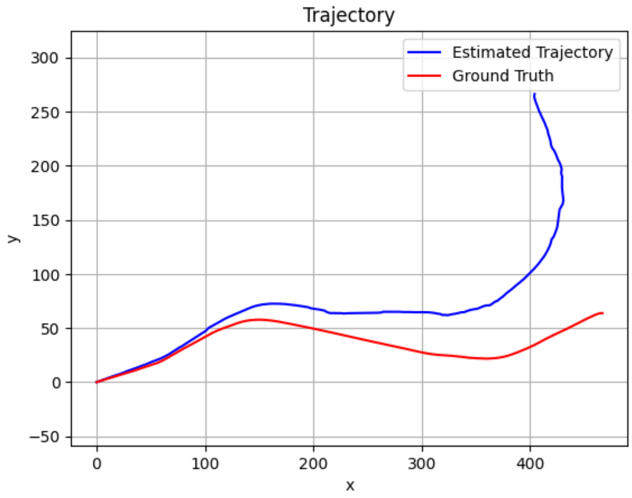
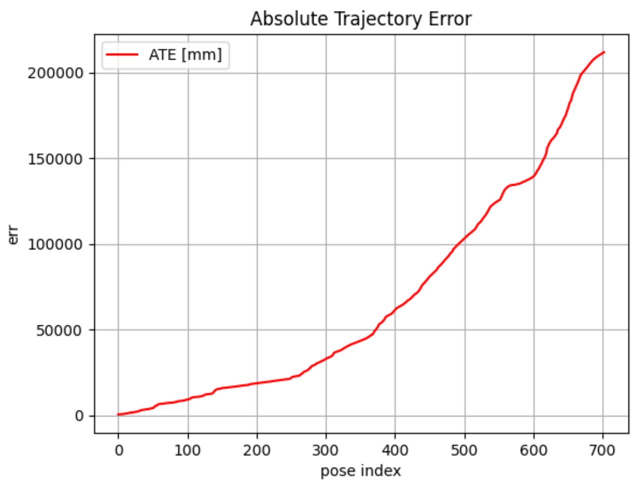
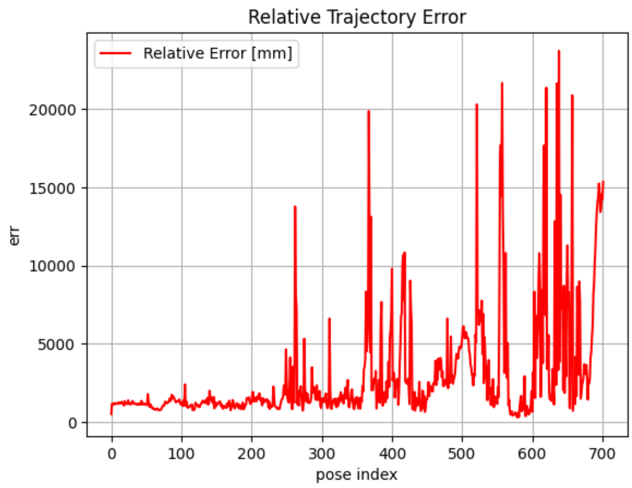


Figure 7: Nielsen Residential Drive '0061' Trajectory Estimation vs Ground Truth



(a) Absolute Error



(b) Relative Error

Figure 8: Nielsen Residential Drive '0061' Error Plots

Err Type	ATE	Relative
Mean:	67909.94 mm	2944.62 mm
Median:	43549.13 mm	1501.40 mm
Max:	211760.45 mm	23702.99 mm
RMSE:	45752.32 mm	8670.77 mm

Table 4: Nielsen Residential Drive '0061' Performance

4.3 Summary

In both test cases we see the Nielsen implementation performing better than the proposed visual odometry system. The main drawback with the Nielsen implementation appears to be heading drift, and seems to deal with the scale changes well.

5 Conclusions

While local optimisation may help in mitigating issues with drift and creating loop closure, we see that it is not necessary for gaining accuracy improvements, as Nielsen's implementation, while similar, provides enhanced performance.

References

[1] D. Scaramuzza and F. Fraundorfer, *Visual Odometry Part I: The First 30 Years and Fundamentals.*

[2] N. Nielsen, "Stereo visual odometry." [Online]. Available: https://github.com/niconielsen32/ComputerVision/blob/master/VisualOdometry/