ENGN2560 LAB 3 REPORT

Neil Xu

Introduction

In this lab, we perform absolute pose estimation of a camera path using a sequence of images. We use two different methods of absolute estimation: propagation of sequential estimated poses, and computation of an absolute pose using triangulated points. Problem 1 sees us perform absolute pose estimation on a third image given two. Problem 2 utilizes the propagation method aided by a set of ground truth poses to reconstruct the camera path, and Problem 3 uses absolute pose estimation on triangulated world points.

Problem 1

In this problem, we perform absolute pose estimation of a third image given two images. We use Ransac to compute an estimated pose between the first and second image. We use Symmedian triangulation to triangulate the inlier features on the two images. Next, we create 2D and 3D feature correspondences between these triangulated features and the matching SIFT features between image 2 and image 3. Afterwards, we use Ransac4AbsPose function, which makes use of P3P Lambda Twist algorithm to compute the absolute rotation matrix and translation vector. We evaluate the results using the equations provided, yielding the following errors: 1

	Error
Rotation Error	0.0066
Translation Error	9.54×10^{-4}

Table 1

Comparing this method with the process of propagating the relative poses of images 2 and images 3 yields the following results: 2

Method	Rotation Error	Translation Error
Absolute	0.0066	9.54×10^{-4}
Propagation	0.01	7.933×10^{-4}

Table 2

It appears that both methods are quite accurate, with the absolute method doing slightly better on rotation. These methods are really comparable because we have only propagated across one image, meaning that any error in the estimation has compounded only slightly. If we propagate across many images, we should see that the error increases substantially compared to the absolute method.

Problem 2

In this problem, we estimate the camera trajectory across a sequence of images by computing the relative pose from one image to the next, and propagating the poses over the sequences of images such that they share a coordinate system. We correct for metric ambiguity by scaling the translation vector using a value computed from the corresponding ground truth poses. We iterate through each image in the sequence and estimate a relative pose using Ransac. Afterwards, we propagate this relative pose to the world coordinates and scale the translation vector. This is our estimated pose. Doing this across all 180 images yields the following trajectory and error: 1 3

	RMSE
Rotation Error	0.1271
Translation Error	0.1316

Table 3

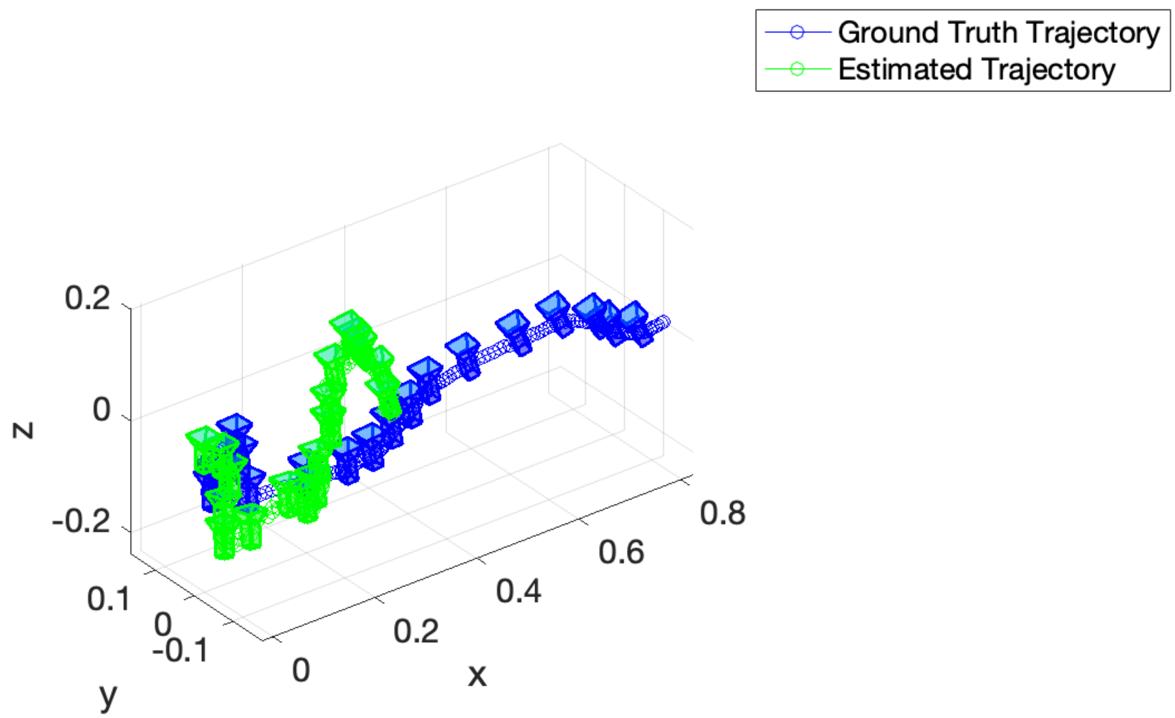

Figure 1: Ground truth trajectory vs. computed trajectory

Problem 3

In this problem, we estimate the camera trajectory using the method from problem 1. It should be noted that I was not able to achieve a perfect implementation of this method in time. This method uses the absolute estimation algorithm to sequentially estimate the absolute pose of the camera for every image in the sequence, given the initial set of triangulated 3D world points. In each iteration, we compute the matches based on the previous keyframe image, and create correspondences between the triangulated points and the 2D matches. Afterwards, I determine whether this new image should be a keyframe based on some selection criteria. If it should be made a keyframe, I perform Symmedian triangulation to get the new set of world points and continue iterating. The results I achieved are displayed below: 4 2

	RMSE
Rotation Error	0.3048
Translation Error	0.7496

Table 4

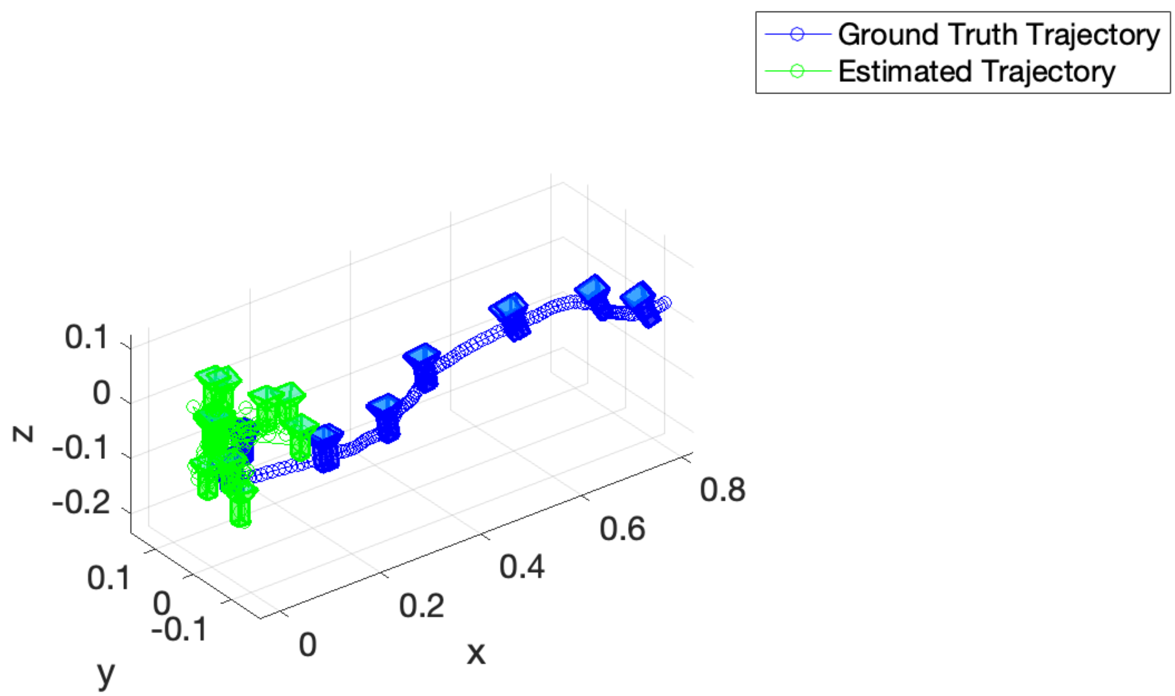

Figure 2: Ground truth trajectory vs. computed trajectory