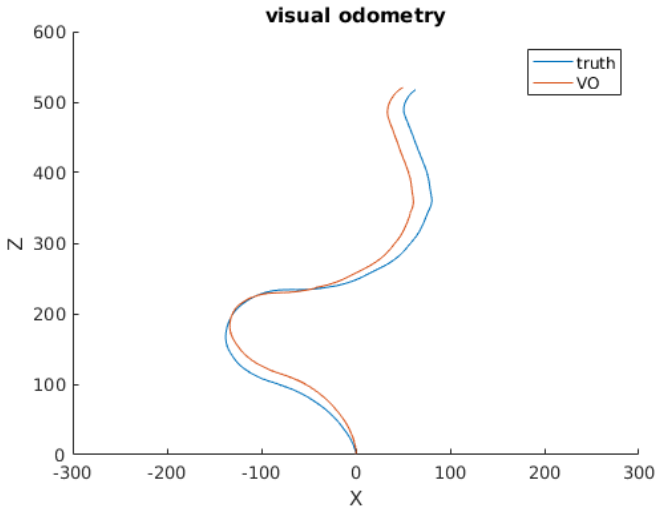
ECEn 631
hw 8 – Visual Odometry
Jae Lee

Task 1

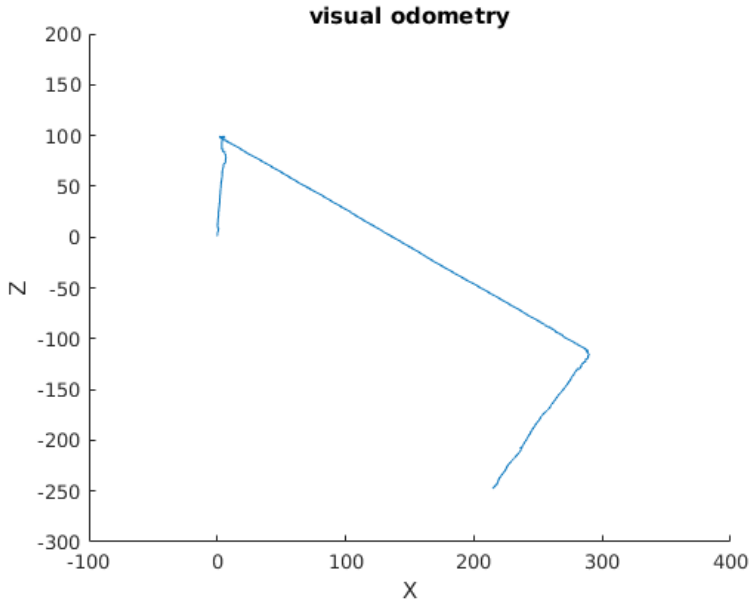
For feature matching, I initialize features with ‘goodFeaturesToTrack’ on the current frame and use ‘calcOpticalFlowPyrLK’ to find matching features on the next frame. I used ‘findFundamentalMat’ for the first time to reject outliers from these matched points. Then, ‘findFundamentalMat’ is used one more time to get the actual fundamental matrix. Once I know fundamental matrix, I can compute essential matrix which will be decomposed into rotation and translation matrices. These rotation and translation are just transformation from previous frame to current frame, but I accumulate rotations and translations from the first frame to current frame to find out the total transformation from the starting point to the current point.



As shown, the result seems to be quite accurate. I spent good amount of time tuning parameters. It turns out that the parameter that indicates maximum distance allowed from a point to an epipolar line for ‘findFundamentalMat’ function has the most influence on my visual odometry performance.

Task 2

This task is more difficult to get accurate visual odometry because there are some image sequences with not enough good features. I intentionally skip 2 frames to get more baseline because without skipping frames resulted in almost no baseline movement in motion vectors which caused a lot of confusion for forward movement. When there aren't enough good features(blank wall), I applied the same rotation and translation of the last time when there were enough good matched features.



After several hours of tuning and trying new methods, I decided this is the best result I can produce at this moment. Except for the first turn is estimated too much, it keeps straight line when camera is moving straight forward and also turning direction is estimated correctly. I didn't multiply by the scale factor 0.8 because for the task 1, the result was better without the scale factor and I assuming the same thing for the task 2.