Odometry results based on Cosine rule :

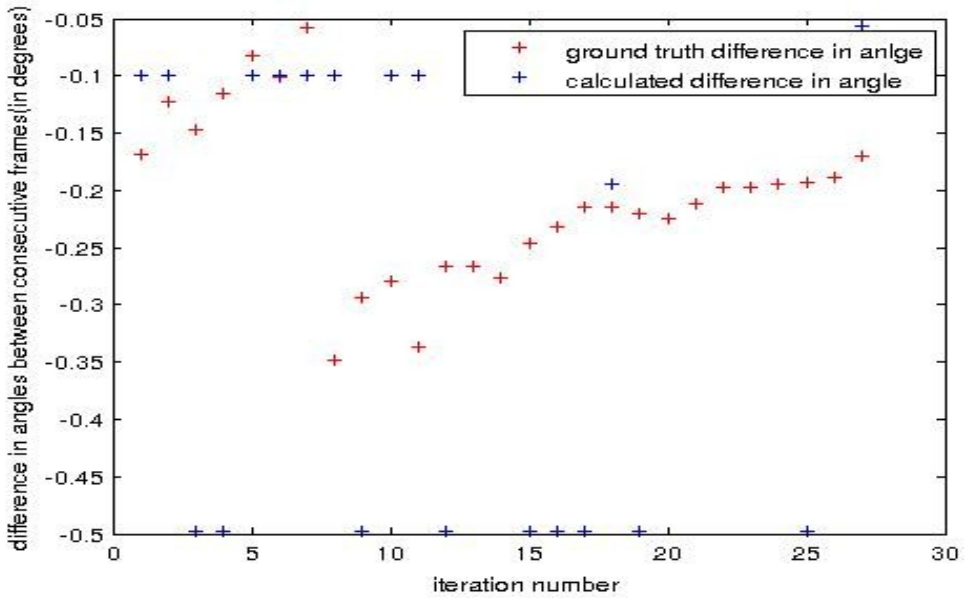
Features : Deep Matches, SIFT
Yaw angle: One Point Ransac
Translation: Cosine rule

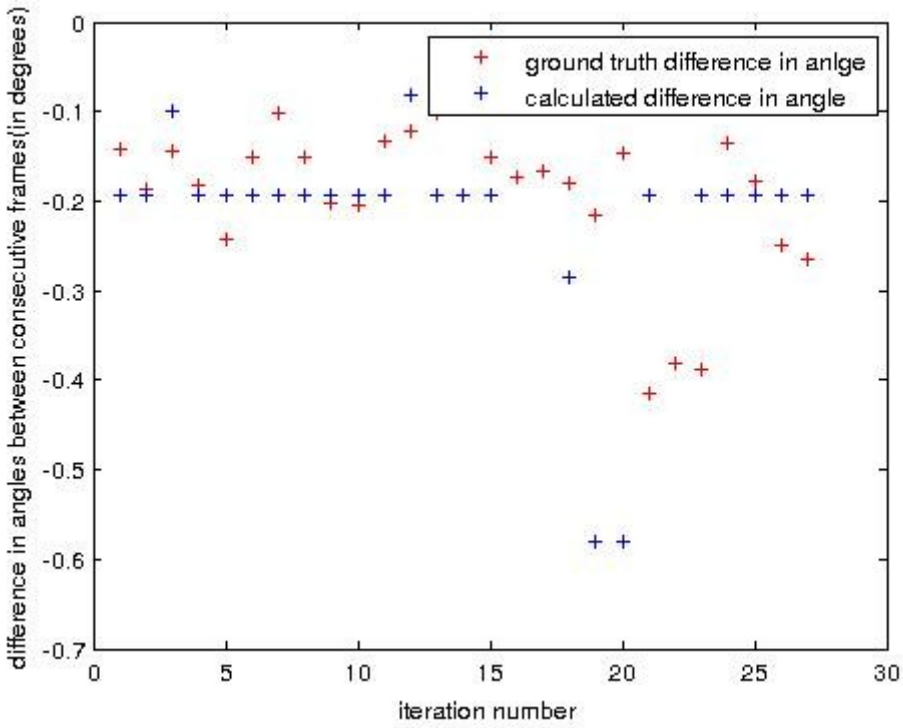
Feature Matches:

Following Image shows the deep matches between two typical frames. I used roipoly(matlab function) to select the features within a patch that usually gives good matches.

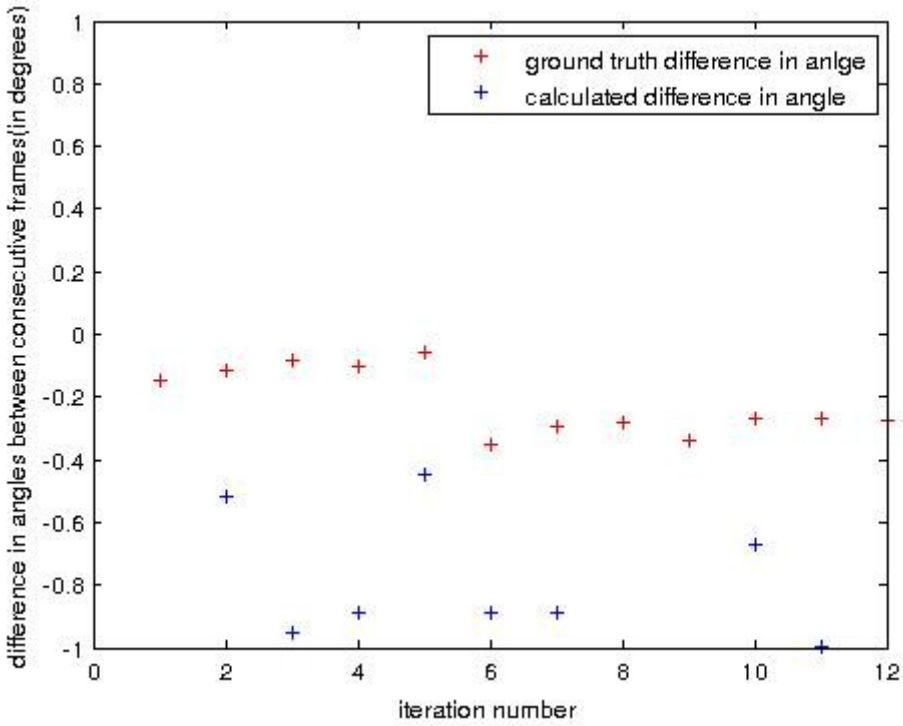
Yaw angle results:

Following results shows the yaw angle comparision on two sequances computed through one point Ransac on the corresponding points obtained through deep matching. In each sequance vehicle has approximately Tavelled 35m.





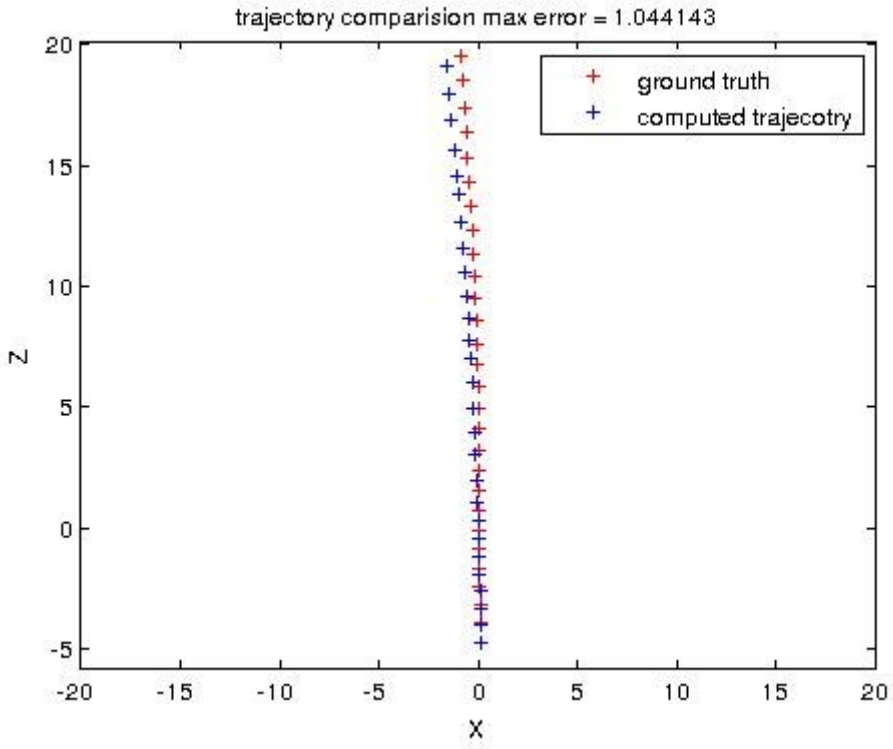
The following figure shows the yaw angle comparision computed through one point Ransac on the corresponding points obtained through SIFT matches.

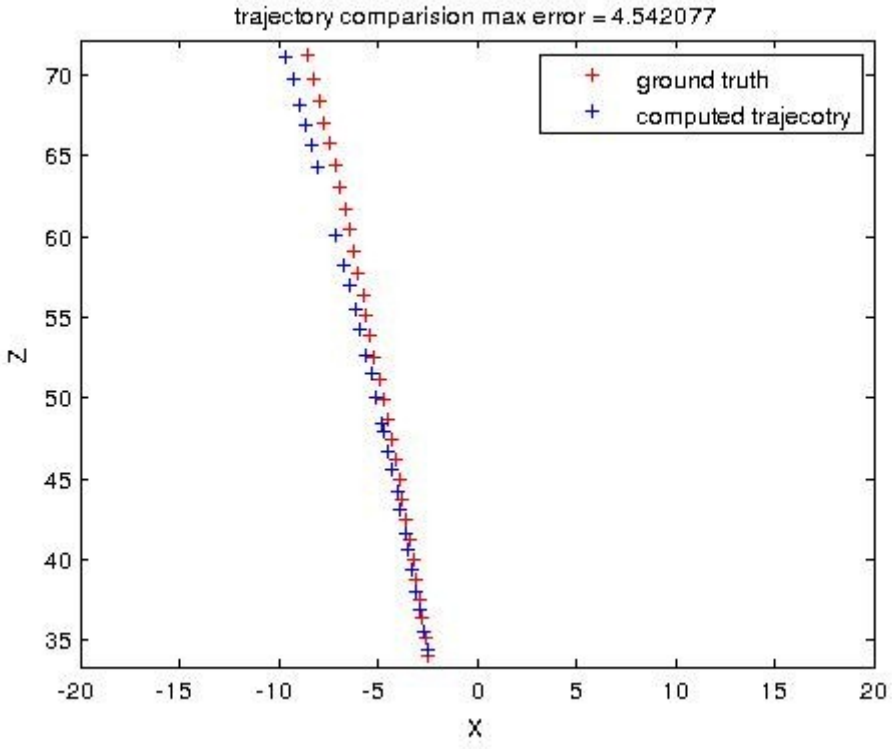


The error is sligthly more for the SIFT matches. Because of the Repetitive texture of the road we don't get good SIFT matches.

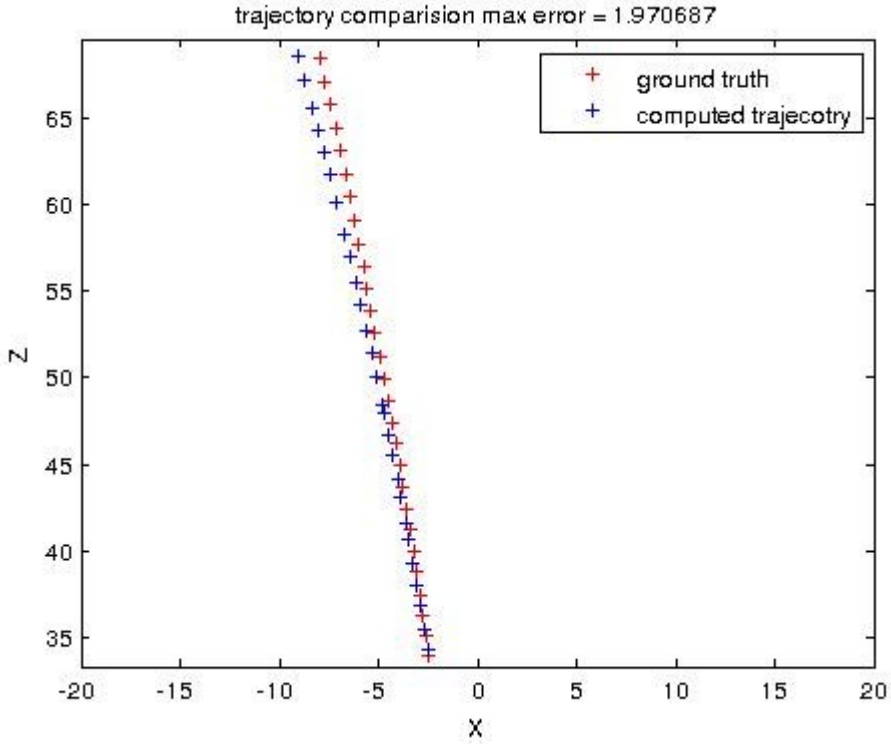
Trajectory Results:

Following figures shows the Tajectory comparision for two sequences:





The error shown above each of the figures is the maximum error between the trajectory points calculated though our method and the correspoding ground truth points. Note that in second case the error is increased drastically this is because during one of the iteraion two bad corresponding points got selected. I refined the point match manually for this particular iteraion (around (-5,60)) and got the following trajectory .



The error is reduced significanly which shows that we are having some good matches but still have to automate the process of selecting these matches, currently I am considering point match which is closest to the ground plane (y = 1.65) and computing Translation by applying cosine rule on the corresponding 3d points .

To calculate the translation first I get the disparity using libeals and then compute the 3d points corresponding to the good match (mentioned earlier) after multiplying the point in the second frame with the Rotation matrix(obtained through one point Ransac) I apply the cosine rule to get the magnitude of the third side which is also the translation of the camera(see the figure below) .