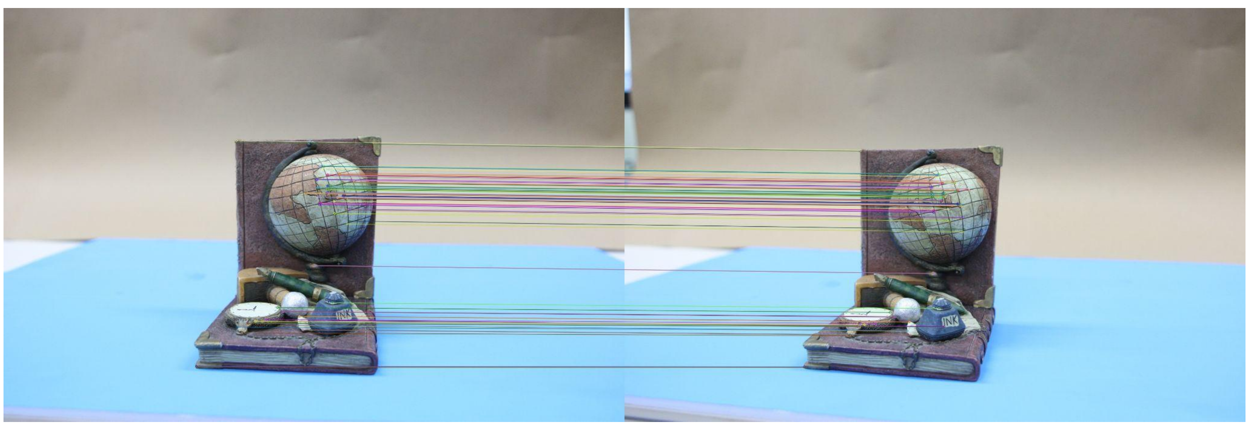
(Real Time) Autonomous system - (Fractal 2)

Assignment 1

Saurabh Shashikant Burewar (B18CSE050)
Rituraj Kulshresth (B18CSE046)

Finding Correspondences

We have written two functions using SIFT and ORB image matching algorithms to find the set of ground truth correspondences and either of them can be used. We get a total of 284 matching pairs of points. The below image only shows 100 of the best correspondences from our set. The set of ground-truth correspondences found from the pair of images was as follows:



Finding Essential matrix

Now, we normalize the coordinates and use the eight point algorithm to find the essential matrix. We take eight points from our list of correspondences and build the matrix A as given in the algorithm. We use SVD to solve the homogeneous equations we get and estimate the essential matrix.

The Essential matrix found from the provided setup was as follows:

-0.99994529	-0.00807781	0.00502644
-0.00134917	0.58170363	-0.03627318
-0.00933879	0.81097541	-0.0505361

As expected, the Determinant of the Essential Matrix was found to be 0.

Finding R and t

We get R and t by decomposing the essential matrix and using the below formulas -

$$\begin{aligned} \mathbf{R} &= \mathbf{U}\mathbf{R}_z^\top\left(\pm\frac{\pi}{2}\right)\mathbf{V}^\top \\ [\mathbf{t}]_\times &= \mathbf{U}\mathbf{R}_z\left(\pm\frac{\pi}{2}\right)\Sigma\mathbf{U}^\top. \end{aligned}$$

- The Camera Rotation matrix found from the decomposition of the essential matrix was as follows:

-8.62935098e-03	9.99962524e-01	-6.96918926e-04
5.31122458e-01	5.17392096e-03	8.47279273e-01
8.47251125e-01	6.94132092e-03	-5.31147201e-01

  As expected (since it is a rotation matrix), the determinant of R was found to be 1.

- The Translation Vector found from the decomposition of the essential matrix was as follows:

1.00000139

0.32871448

-0.94442788

Finding Corresponding 3D points

We found the corresponding 3D points using the following method -

$$\begin{aligned} \lambda_1\hat{\mathbf{p}}_1 &= \mathbf{M}\hat{\mathbf{P}} \Rightarrow [\hat{\mathbf{p}}_1]_\times \mathbf{M}\hat{\mathbf{P}} = \mathbf{0} \\ \lambda_2\hat{\mathbf{p}}_2 &= \mathbf{N}\hat{\mathbf{P}} \Rightarrow [\hat{\mathbf{p}}_2]_\times \mathbf{N}\hat{\mathbf{P}} = \mathbf{0} \\ \begin{bmatrix} [\hat{\mathbf{p}}_1]_\times \mathbf{M} \\ [\hat{\mathbf{p}}_2]_\times \mathbf{N} \end{bmatrix}\hat{\mathbf{P}} &= \mathbf{0} \Rightarrow \mathbf{A}\hat{\mathbf{P}} = \mathbf{0}. \end{aligned}$$

Here, M is the intrinsic matrix and while N is M*[[R t], [0] 1] as discussed in the camera matrix part in class. With this, we have our matrix A and we use svd to solve the homogeneous equations.

Example 3D points-
[-0.13727768425013234, 0.2510473785112311, -0.4943362719967822, 1]

[-0.13993093409682247, 0.251196426662701, -0.49458791707817823, 1]
[-0.14481961365115534, 0.2540727396170356, 0.4915418262742527, 1]
.
.
.

We couldn’t get a suitable example as to how we can plot the points that we generated using the algorithm taught in class.