Autonomous Systems

Programming Assignment

Team	Roll No.
Ayush Gangwar	B20CS008
Harsh Sharma	B20CS017

Problem Statement :

With the provided intrinsic camera matrix K ("Intrinsic Matrix K.txt"), consider two photos I1 ("im1.jpg") and I2 ("im2.jpg") of a static scene taken by a single camera.

- ☐ **Objective 1:** With any of the current implementations, find a collection of ground-truth correspondences $\{(p_i, p'_i)\}^n_{i=1}$.Ascertain that n = 100 true correspondences are present.
- ☐ **Objective 2:** Suppose that the camera location's coordinate system and the global coordinate system are in alignment. Use the method that was given in class to locate the Essential matrix E.
- ☐ **Objective 3:** Decompose the obtained Essential matrix E into the camera motion rotation matrix R and the translation vector t.
- ☐ **Objective 4:** Let Pi be the corresponding 3D point for the pixel pair (pi,p′i). Find Pi, $\forall i \in \{1, 2, . . . , n\}$ using the triangulation approach learned in the class.
- ☐ **Objective 5:** Plot the obtained Pi, $\forall i \in \{1, 2, . . . , n\}$ and the camera center t.

Python and its dependencies :

Libraries	**Functions**
CV2	Operations on images.
Numpy	Matrix Math etc.
Matplotlib	3D Graphs plots etc.

Objective 1:

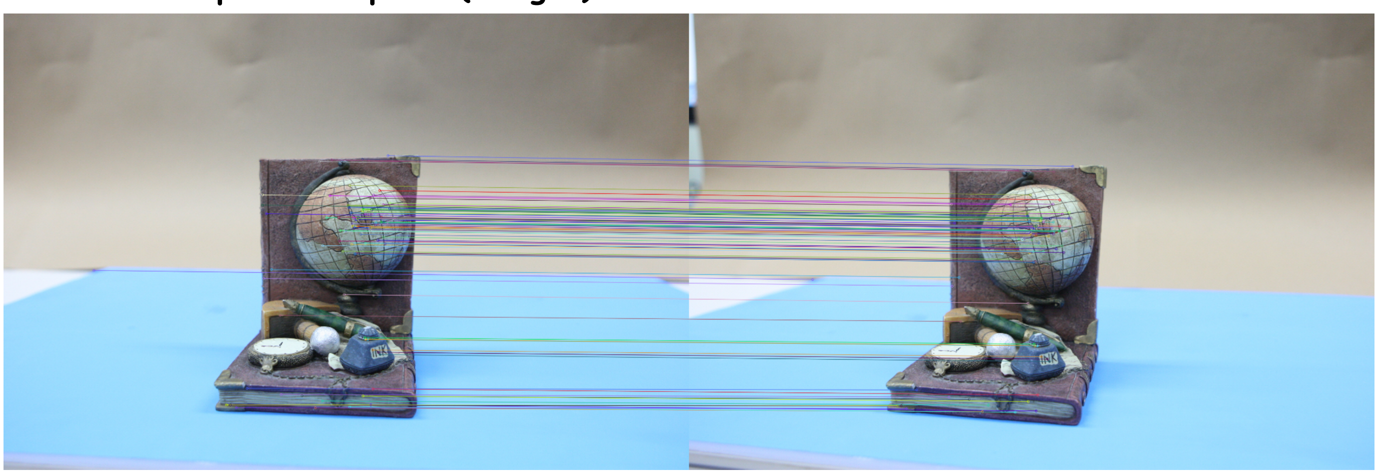
Approach

- **SIFT (Scale-Invariant Feature Transform)** method is used to find correspondences between images.

Implementation Explanation:

- To find a set of ground-truth correspondences using the SIFT method, we can follow these steps:
    - Extract SIFT features from both images. This involves detecting and describing local invariant features in each image using the SIFT algorithm.
    - Match the SIFT features between the two images. This involves finding corresponding features between the two images based on their descriptors.
    - Through BF matcher matching is found between descriptors.

Correspondence points(Images):



Objective 2:

Approach/Implementation:

- First we have normalized the correspondence as following:

$$x = K^{-1} * \hat{p}_i \quad \text{and} \quad y = K^{-1} * \hat{p}'_i \,.$$

- Now using camera equations for $\hat{p}_i$ and $\hat{p}'_i$ and doing some manipulation in the equations we can come to the following result:

$$y^T_i [t]_x R x_i = 0$$

$$y^T_i E x_i = 0 \qquad \text{where } E=[t]_x R$$

- Now doing some further manipulations we can get the following equation :

$$Ae=0$$

- e is the column matrix of of dimension 9x1
- We need to find e such that Ae=0 and e^t e=1 that is $||Ae||_2$ is minimized
- We can do this by finding eigen vector of A^T A corresponding to smallest eigen value of A
- The E we got from this does not satisfy the general properties of the essential matrix .
- Then, to enforce this property, we update the essential matrix as follows:

$$E = U\mathrm{diag}([\sigma\ \sigma\ 0])\ V^T \text{ with } \sigma = (\sigma_1 + \sigma_2)/2.$$

Essential Matrix:

E= [[0.00797246 -0.18799947 -0.11662825]
[0.1261202 -0.0248958 0.68643113]
[0.10079511 -0.67300107 -0.01200153]]

Objective 3:

Approach/Implementation:

- Now, we can decompose E as $[t]_x R$:

$$R = UR^T_z(+\pi/2)\ V^T \qquad R = UR^T_z(-\pi/2)\ V^T$$

$$[t]_x = UR_z(+\pi/2) \textstyle\sum U^T \qquad [t]_x = UR_z(-\pi/2) \textstyle\sum U^T$$

- There are four possible solutions . We have chosen any one of them .\

Rotation Matrix:

R = [[-0.8538945 -0.30002817 0.42526143]
[-0.29091454 0.9526931 0.08800337]
[0.43154712 0.04856914 0.90078195]]

Translation Vector:

T= [-0.9497246 -0.15662339 0.27109464]

Objective 4:

Approach/Implementation:

- For finding the 3D points we have used triangulation method as follows :

$$\lambda_1 \hat{p}_1 = M\hat{P} \Rightarrow [\hat{p}_1]_x M\hat{P} = 0$$

$$\lambda_2 \hat{p}_2 = N\hat{P} \Rightarrow [\hat{p}_2]_x N\hat{P} = 0$$

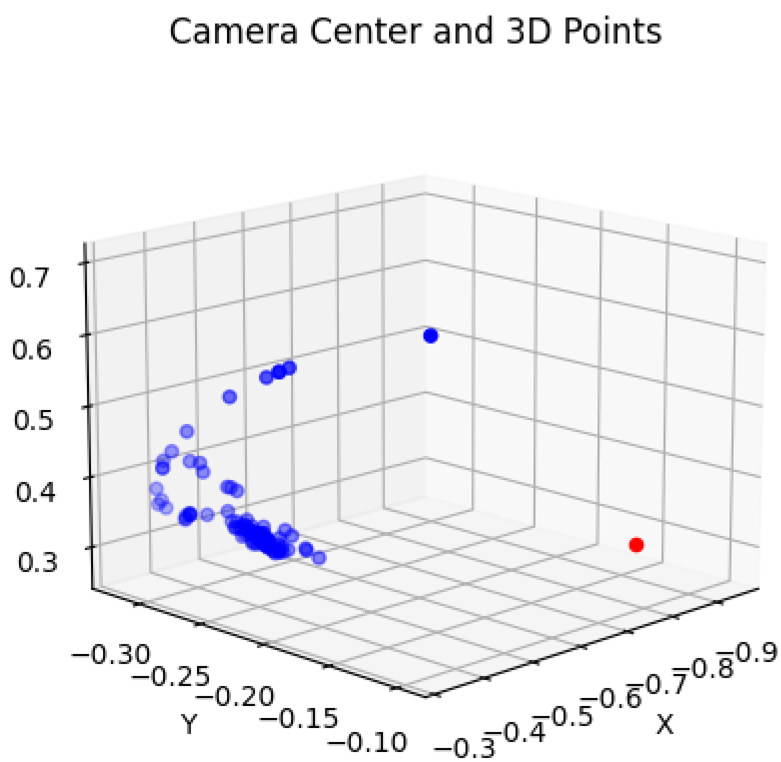
$$\begin{bmatrix}[p^1 1]\mathrm{x}M \\ [p^1 1]\mathrm{x}N\end{bmatrix}\mathrm{P}^{\hat{}} = 0 \Rightarrow \mathrm{AP}^{\hat{}} = 0.$$

- The equation is similar to Ae=0. We can find P by calculating the eigen vector of A^tA corresponding to the smallest eigen value .

Objective 5:

Approach/Implementation:

- After finding the 3D points we have plotted them using a 3D plot with the help of matplotlib .



The red point is the camera and blue are the 3D points we got from the triangulation method . There is one blue point which is separated from the cluster because in finding correspondence there was one outlier . There was one correspondence which was not present in the main image of the book . We can see that correspondence clearly in the correspondence image (on the blue background of the image)