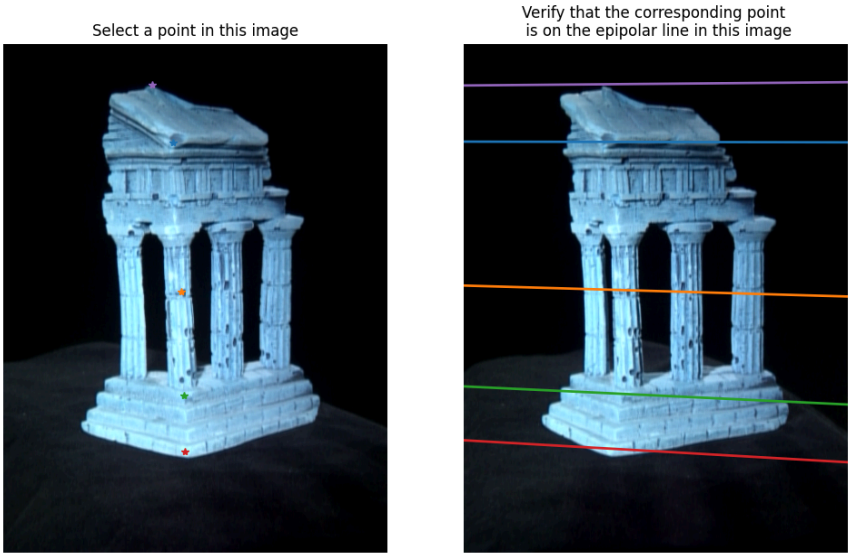
Assignment 2

Q1 Implement the 8-point algorithm

Here is how the recovered Fundamental Matrix (F) looks like:

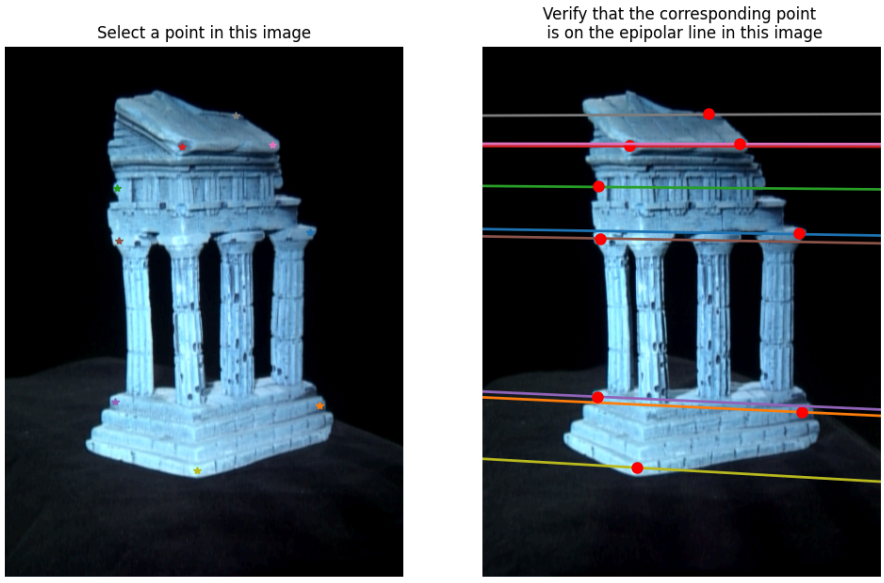
```
------------------------------------------------
Recovered Fundamental Matrix
[[ 1.31462331e-10  1.59558238e-07 -1.76090984e-05]
 [ 2.89498020e-08 -3.91464654e-11 -1.11779461e-03]
 [-2.30195107e-06  1.07499268e-03  4.44244471e-03]]
------------------------------------------------
```

Here is the visualization of some epipolar lines:

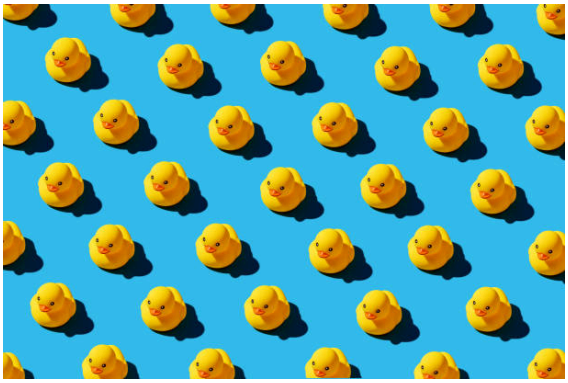


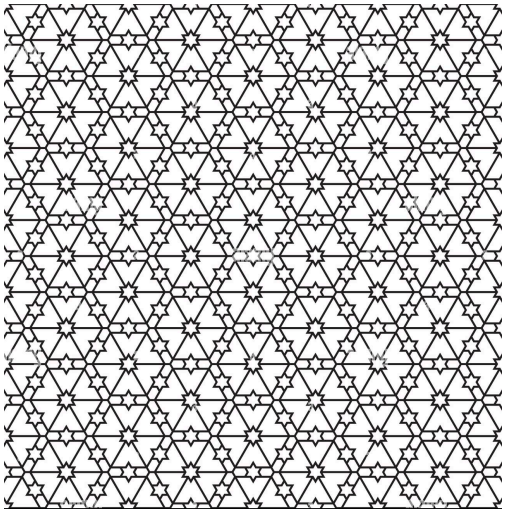
Q2 Find epipolar correspondences

Here is an example of matched points with their epipolar lines. We used the window size of 10 to construct a patch for candidate selection. The similarity metric was **Euclidean distance**.



The matching algorithm is highly likely to fail if the paired images depict multiple instances of the same object or the paired images portray some kind of abstraction with repetition. In those cases, the point on an image can have a lot of matching points but the algorithm has no ability to distinguish between them. Here are the examples:





Another fail case occurs when the image depicts an object such as white paper, which has no texture or structural changes on it. In this case, we cannot find a match to a point on an image as there is no difference between the candidate points in the other image.

Q3 Write a function to compute the essential matrix

Here is the computed Essential Matrix for the given temple image pair:

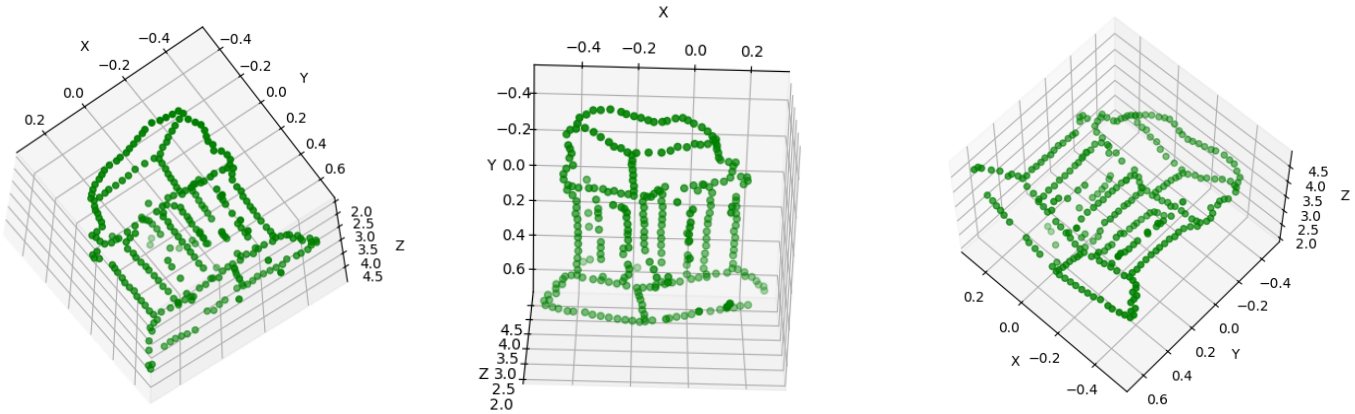
```
------------------------------
Essential Matrix:
[[ 3.03890450e-04  3.70171659e-01  3.31763253e-02]
 [ 6.71629143e-02 -9.11474873e-05 -1.69230271e+00]
 [ 7.42659183e-03  1.71392241e+00  1.93511318e-03]]
------------------------------
```

Q4 Implement triangulation

The index of the selected matrix for P2 was **3** (the indices of the matrix were 0,1,2,3). To determine which one is the best candidate, we performed the following steps. First, we predicted the 3D points using the triangulation function, which is from the perspective of the first camera (image). Then, to calculate the 3D points from the perspective of the second camera (image), we performed transformation on the 3D points using a candidate extrinsic matrix belonging to the second image. After that, we determined the candidate with the most number of points that lie in front of the both cameras (images).

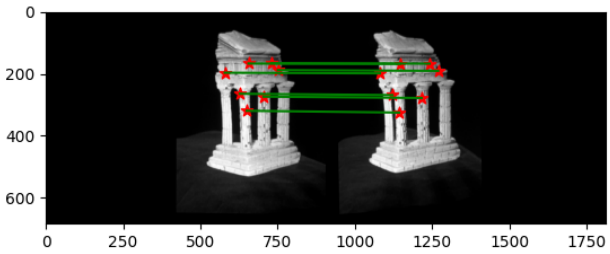
Q5 Write a result script

The reprojection error was around **2.2832**. Here we show some of the views of the sparsely reconstructed object.



Q6 Image Rectification

Here is the screenshot of the result on the temple images (the color of the epipolar lines changed to “green”).



The reason we compute the new matrix in Step 2 of the image rectification algorithm is that, as we learned in class, we want to project our images to the common space where those images are in the same plane and their epipoles are at infinity (in other words, the epipolar line are perfectly horizontal).

Q7 Dense window matching to find per pixel disparity

Here are the disparity maps that are constructed by using the **window size** of 3 (left figure), and 10 (right figure).

We can see that as we decrease the size of the window, the more fine-grained information we can capture, but more noise can also appear. With bigger window sizes, we will have smooth disparity maps that mitigate the noise to some extent.

Q8 Depth map

Here are the images of the constructed depth maps with window size equal to 3 (left figure), and 10 (right figure).

The relationship, which is described in Q7, between the window size used (for disparity calculation) holds for the constructed depth maps also.

When it comes to the baseline effect, it can improve depth estimation accuracy by increasing the disparity between corresponding points, providing more reliable depth information for distant objects. However, it can also limit the ability to estimate depth for nearby objects.

A longer focal length can result in better depth estimation for the distant objects in a scene as it has a narrow field of view, while a shorter focal length can increase the depth estimation performance in a scene which has a broad view that is close to the camera.

To improve depth estimation, one possible solution can be just incorporating additional viewpoints. Another solution, which was mentioned in the class, can be employing multiple window sizes to estimate the disparity in a more robust way, which can lead to better depth estimation.

Q9 Estimate camera matrix P

After we calibrate our camera, we will have an estimated matrix **P = K [R | t],** which can give us information about the intrinsic (**K**) and extrinsic (**R,t**) characteristics of our camera. The intrinsic characteristics include the focal length, skew factor (if exist), and image coordinate center translation information. The extrinsic characteristics include the camera's orientation and position in the world coordinate system, which come from the rotation matrix **R** and translation vector **t**.

Here is the output of reprojection and pose error with clean and noisy points.

```
Reprojection Error with clean 2D points: 4.2841115448993443e-10
Pose Error with clean 2D points: 4.435142333246908e-12
Reprojection Error with noisy 2D points: 6.915547013088516
Pose Error with noisy 2D points: 0.9505828754430184
```

Q10 Estimate intrinsic/extrinsic parameters

Here is the output of intrinsic and rotation errors for clean and noisy points.

```
Intrinsic Error with clean 2D points: 2.7479038247585192e-12
Rotation Error with clean 2D points: 3.4641016151377544
Translation Error with clean 2D points: 3.4597488394454263
Intrinsic Error with noisy 2D points: 0.6750041331242239
Rotation Error with noisy 2D points: 3.4637615799823496
Translation Error with noisy 2D points: 3.456107908240921
```

Q11 Project a CAD model to the image

Here are the images of a CAD model projected onto the image and some of the example 3D points projected to 2D.