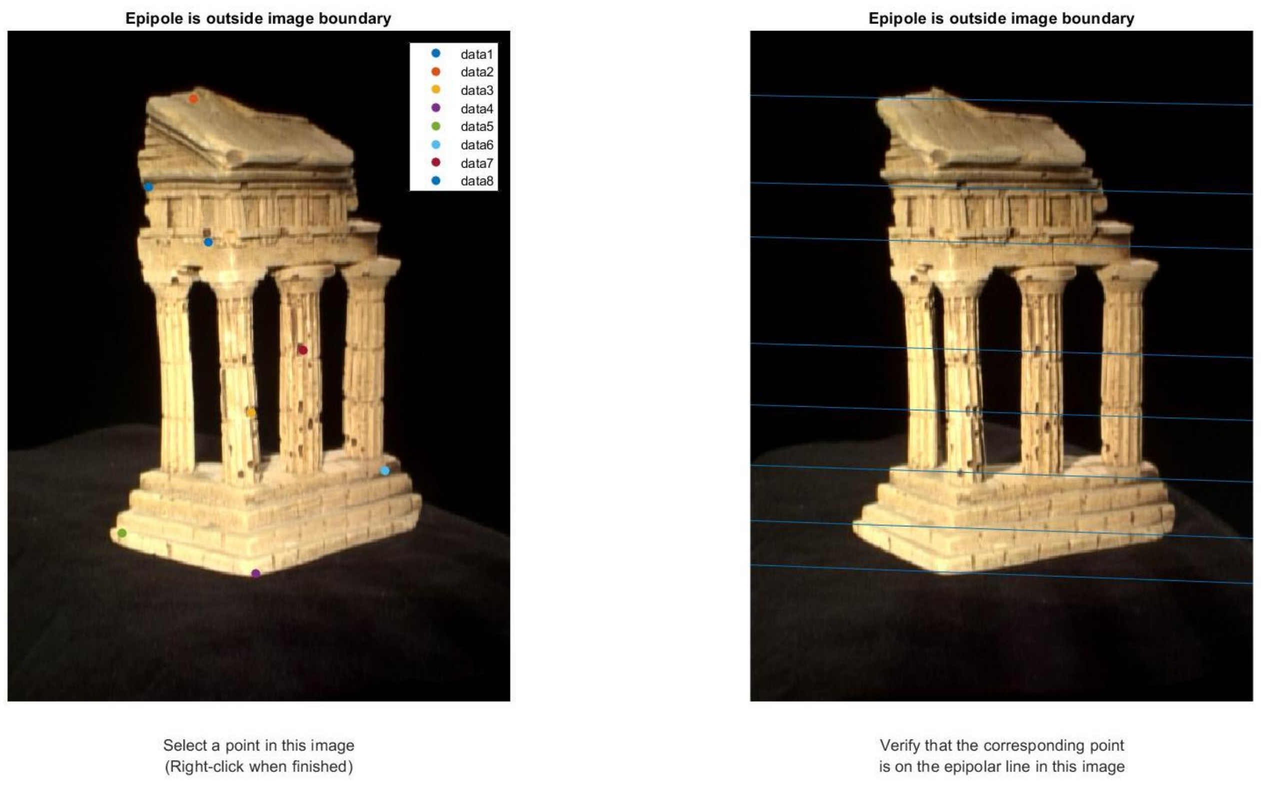
Mahdie Ghane (301478672) Project 5

I want to use one free late day that is left (I have used 3 for three first assignment, 1 for 4 and 4 in total)

Q 3.1.1) in this part, we are supposed to implement the 8-point algorithm. Due to the course materials and following the instruction, first points have been normalized and F has been un-normalized by dividing both coordinates by M. Next, the rank 2 constraint has been enforced on F by using SVD and the also F has been refined using the F function. The function has been tested using the part 3.1.5 in testTemplecoords.m. The results have been shown below:


Epipole is outside image boundary
data1
data2
data3
data4
data5
data6
data7
data8
Select a point in this image
(Right-click when finished)
Epipole is outside image boundary
Verify that the corresponding point
is on the epipolar line in this image

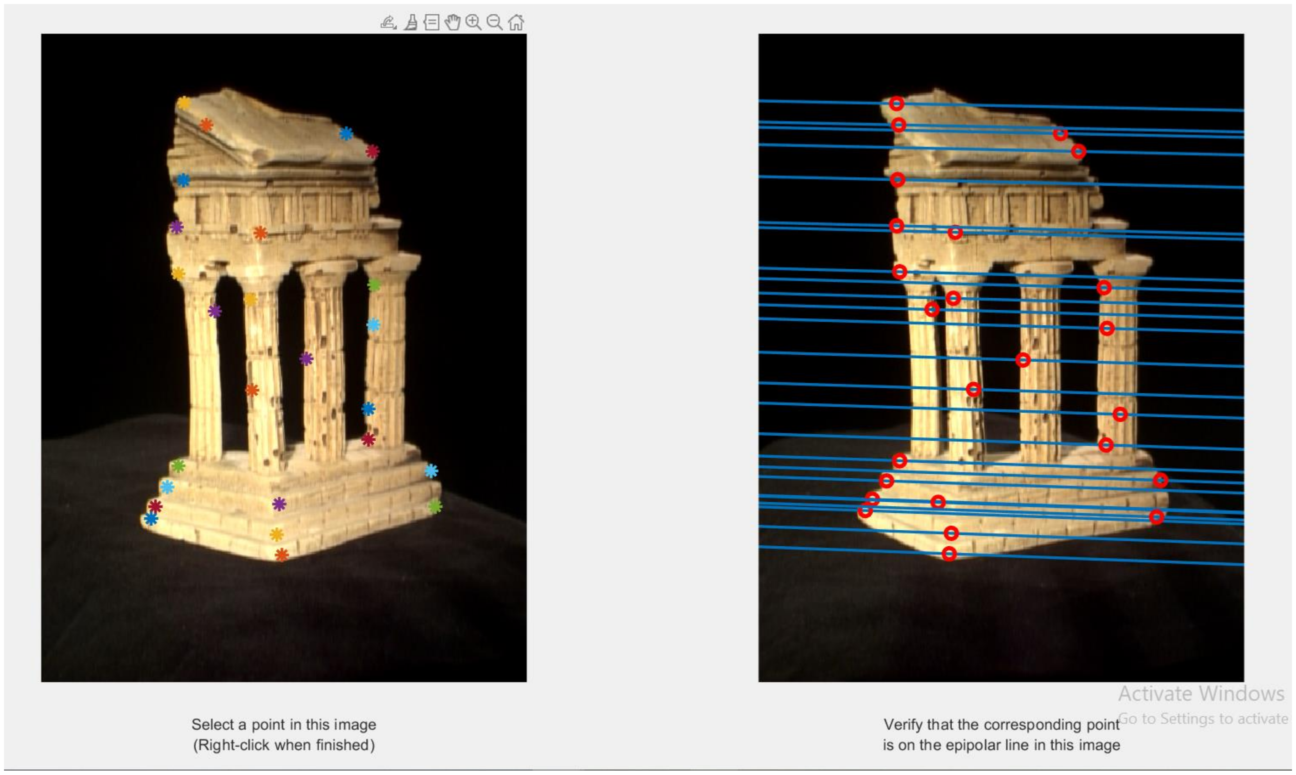
Recovered F has saved in the output folder as a F.mat which is equal to:

F= [1.7518316887523763E-9 -1.866743156896314E-8 -8.520163811603416E-6

-6.456713956318325E-8 -4.021378670372595E-10 4.956769072120689E-4

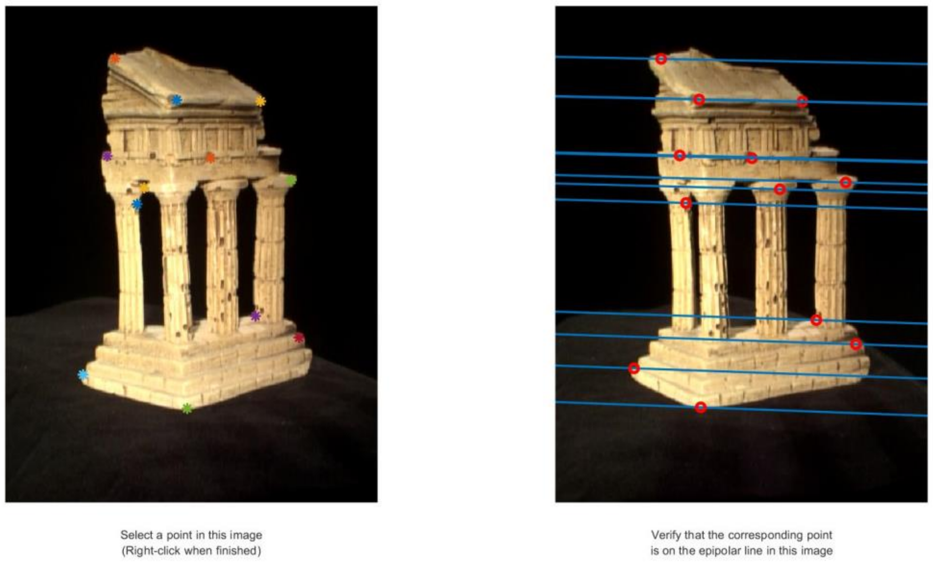
1.6635390742474427E-5 -4.760979270421088E-4 -0.0020569323090249697]

Q3.1.2) The Correspondence function has implemented and the result using the pipolarMatchGui has been shown below:

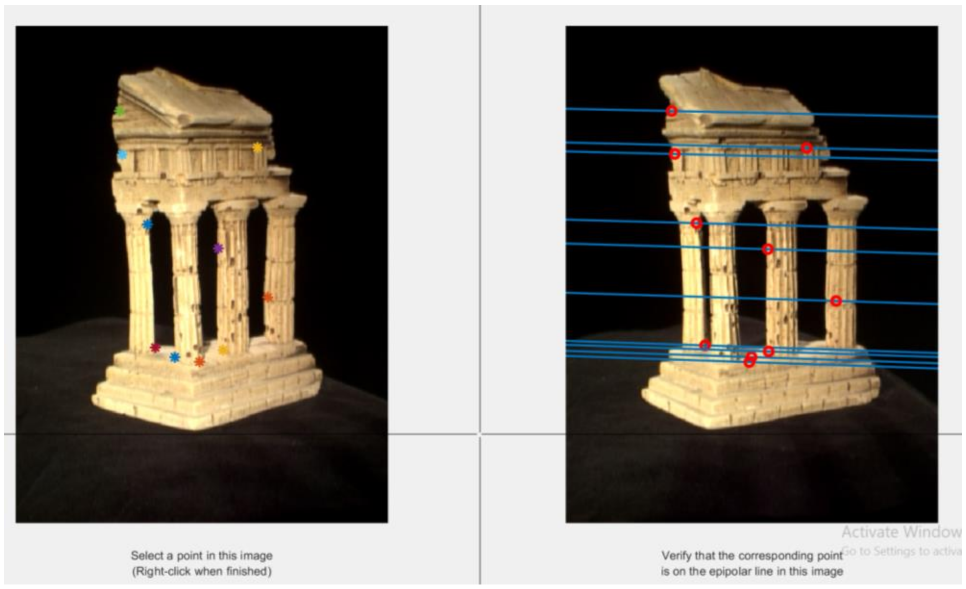

Select a point in this image
(Right-click when finished)

Verify that the corresponding point
is on the epipolar line in this image

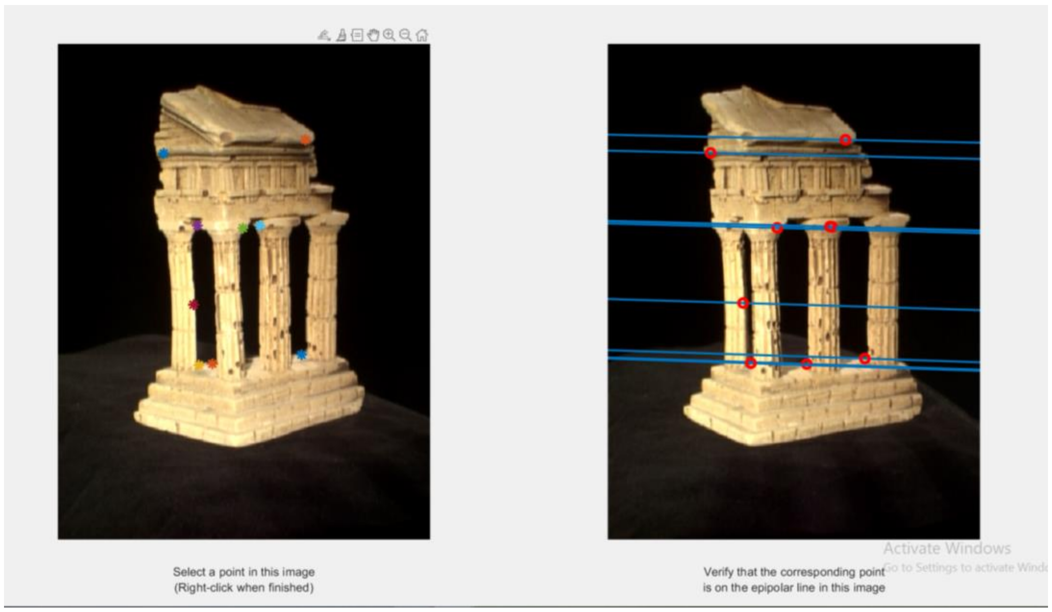
For the similarity part, I have considered a window with length and width 3 around the point using the written function and calculated Euclidean/Manhattan distance between these two window considering all point. This function fails to get correct corresponding points in second image. The figure below shows this case:


Select a point in this image
(Right-click when finished)

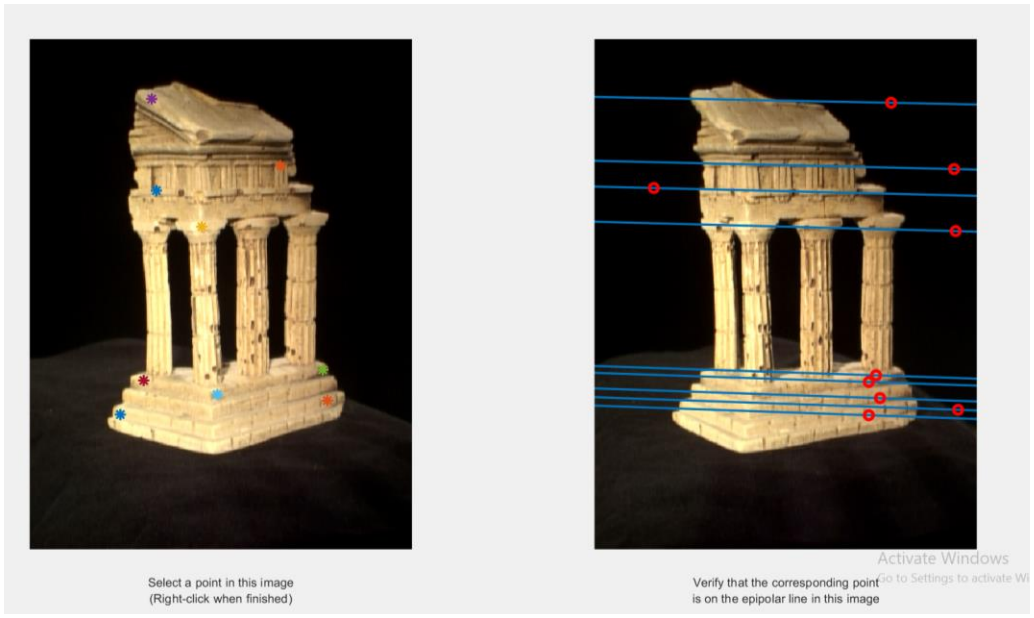
Verify that the corresponding point
is on the epipolar line in this image


Select a point in this image
(Right-click when finished)
Verify that the corresponding point
is on the epipolar line in this image
Activate Windows
Go to Settings to activa

Also, for the Manhatan distanc:


Select a point in this image
(Right-click when finished)
Verify that the corresponding point
is on the epipolar line in this image
Activate Windows
Go to Settings to activate Wind

In these 3 pictures, we can see there are some points that has been incorrectly corresponded in the second image. for example, in the first image, the purple point in the top left is this case. Also, in the last image, the yellow point at the bottom left is the same scenario. We can conclude that the points that would be in the backside in the second image or have more depth are more likely to incorrectly be matched in the second image.


Select a point in this image
(Right-click when finished)
Verify that the corresponding point
is on the epipolar line in this image
Activate Windows
Go to Settings to activate Wi

Also, this case is for the using Euclidean function without considering any window around the point for the similarity. As it could be seen, it has consistently fails, which comes from the poor similarity function and failure in finding the correspondence image properly.

Q3.1.3) For this part, the essentialMatrix function has been implemented using K1 and K2. The estimated E Matrix in this part is:

E=[0.00404956244132008 -0.0433080372767753 -0.0191554874996271

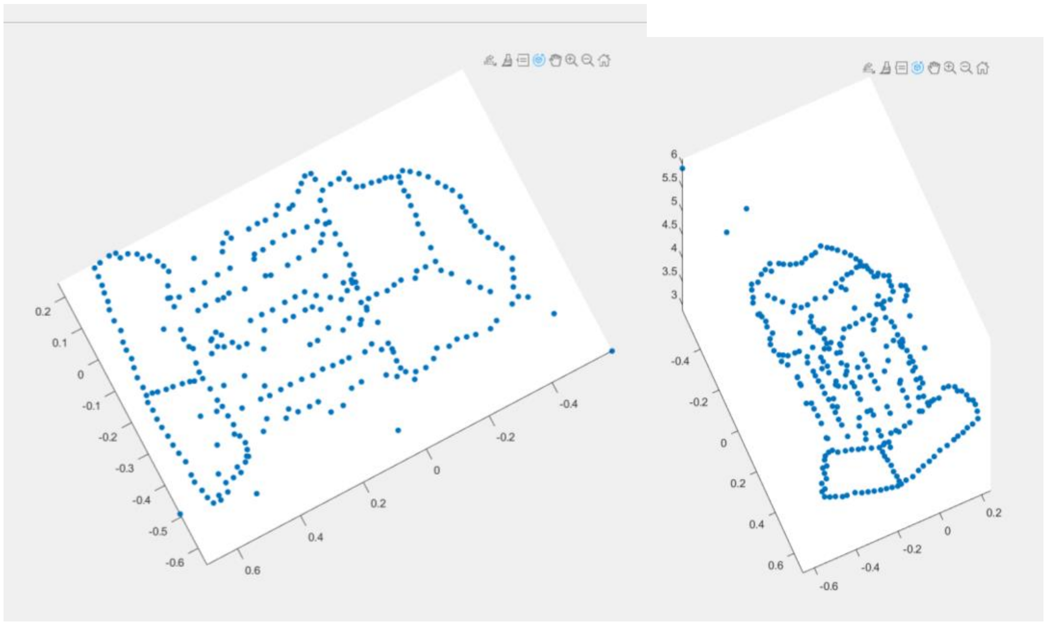
-0.149794366553685 -0.000936326071205216 0.726416434975664

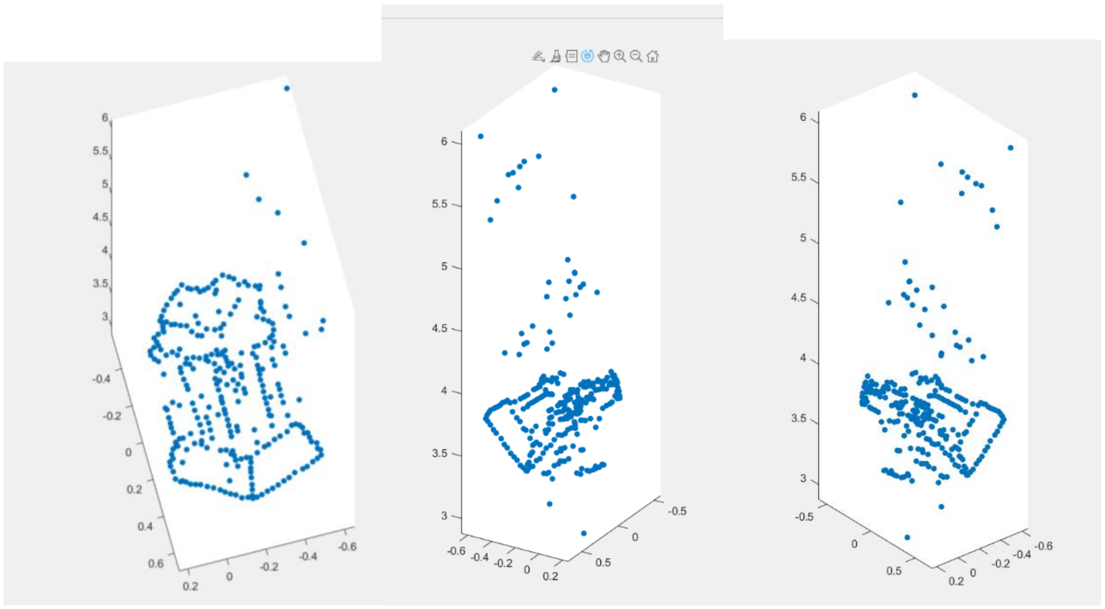
0.00186296855297746 -0.735240786278798 -0.000846576656313824].

It has been saved in the output directory as well. Also, as it has been mentioned in the question, the R1, t1 in this question are simply I3*3 and [0 0 0].

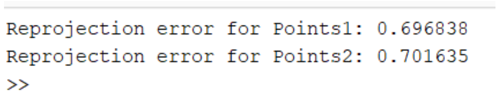
Q3.1.4) For this part, Triangulate function has been implemented and P2, R2 and t2 computed. I have used positive depth test chose the extrinsic matrix in which there is no negative element in the third element (or depth) and is positive.

The reprojection error for this part is 0.558602 and 0.563204 for point 1 and corresponding point 2 from templeCoords.mat respectively and for the points in someCorresp.mat these values are 0.69 and 0.7 for points1 and points2 respectively. The concatenation of the R1, t1, R2, t2 Matrix has been saved as a extrinsic.mat file in the output folder. The five image of the final reconstruction of the templeCoords are:





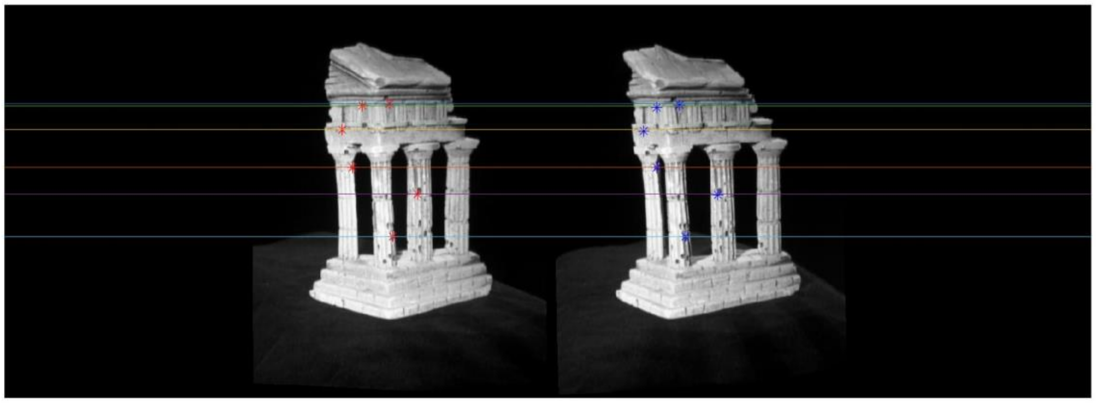
Q3.1.5) The results has been shown in previous parts.



```
Reprojection error for Points1: 0.696838
Reprojection error for Points2: 0.701635
>>
```

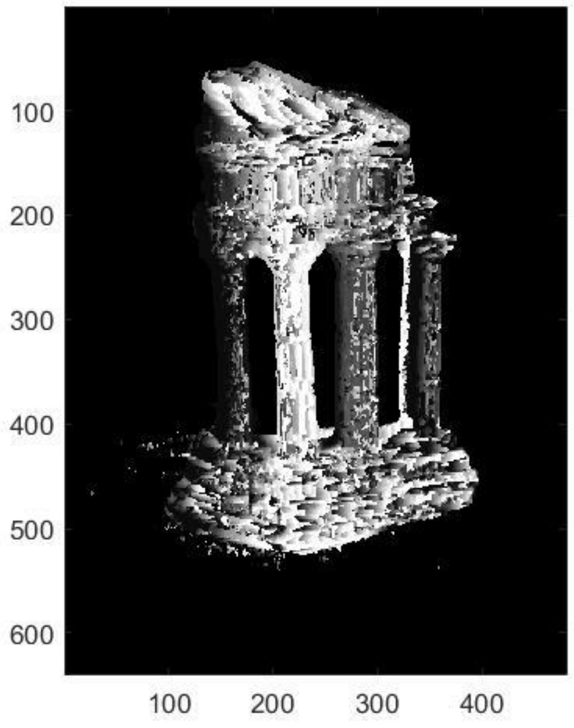
Q3.2.1)

The result from the testRectify.m has been shown in below:

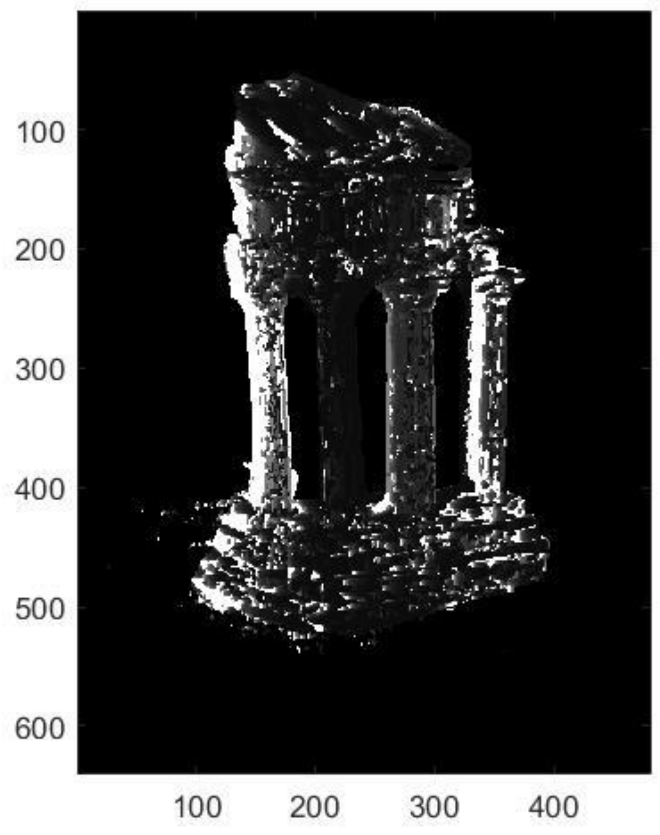


As we can see, there are some epipolar lines that are perfectly horizontal with corresponding points in both images lying on the same line..

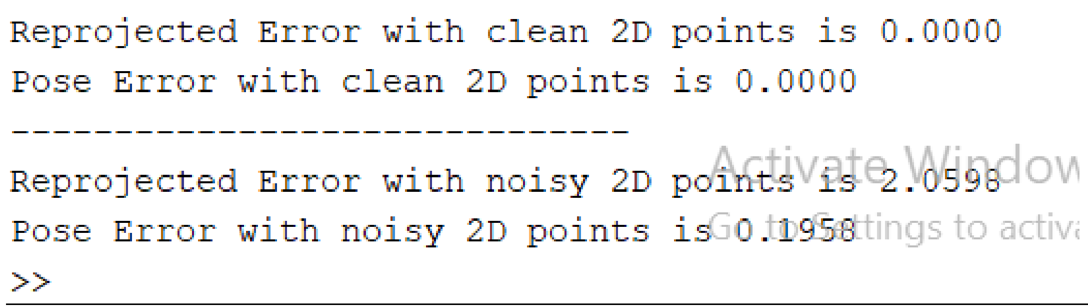
Q3.2.2) The get_disparsity has been implemented and the output is:



Q3.2.3) The get_deph function has been implemented and the output is:

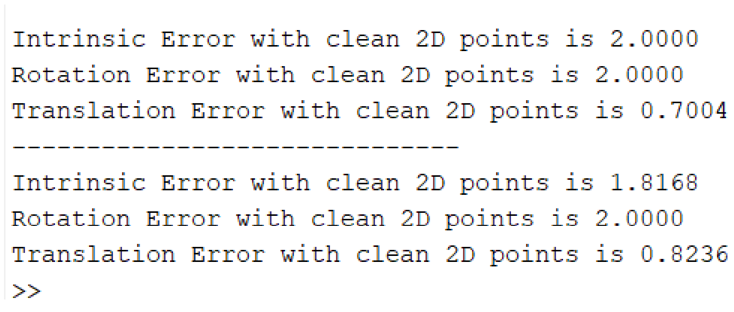


Q3.3.1) The estimate_pose functio has been implemented and its output through testPose is :

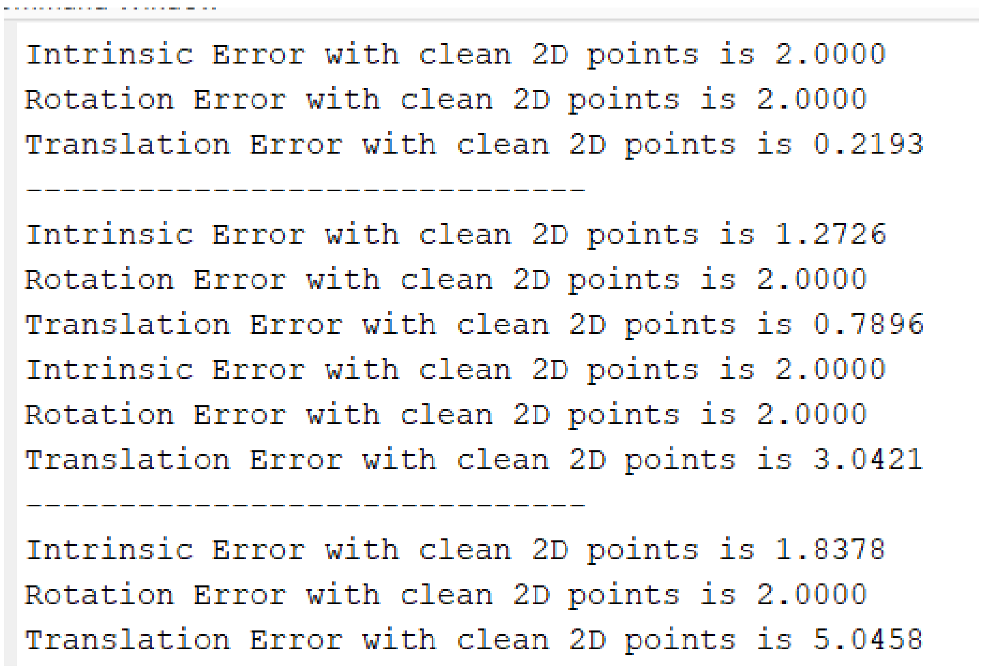


```
Reprojected Error with clean 2D points is 0.0000
Pose Error with clean 2D points is 0.0000
--------------------------------
Reprojected Error with noisy 2D points is 2.0598
Pose Error with noisy 2D points is 0.1958
>>
```

Q3.3.2) The estimate_params function has been implemented and here is its output:



```
Intrinsic Error with clean 2D points is 2.0000
Rotation Error with clean 2D points is 2.0000
Translation Error with clean 2D points is 0.7004
------------------------------
Intrinsic Error with clean 2D points is 1.8168
Rotation Error with clean 2D points is 2.0000
Translation Error with clean 2D points is 0.8236
>>
```



```
Intrinsic Error with clean 2D points is 2.0000
Rotation Error with clean 2D points is 2.0000
Translation Error with clean 2D points is 0.2193
------------------------------
Intrinsic Error with clean 2D points is 1.2726
Rotation Error with clean 2D points is 2.0000
Translation Error with clean 2D points is 0.7896
Intrinsic Error with clean 2D points is 2.0000
Rotation Error with clean 2D points is 2.0000
Translation Error with clean 2D points is 3.0421
------------------------------
Intrinsic Error with clean 2D points is 1.8378
Rotation Error with clean 2D points is 2.0000
Translation Error with clean 2D points is 5.0458
```

Here is the two consecutive run (Note that 2D and 3D input variables are random).

Multi-view stereo: