Assignment 5

Name: Hankun Wang

Student ID: 301416863

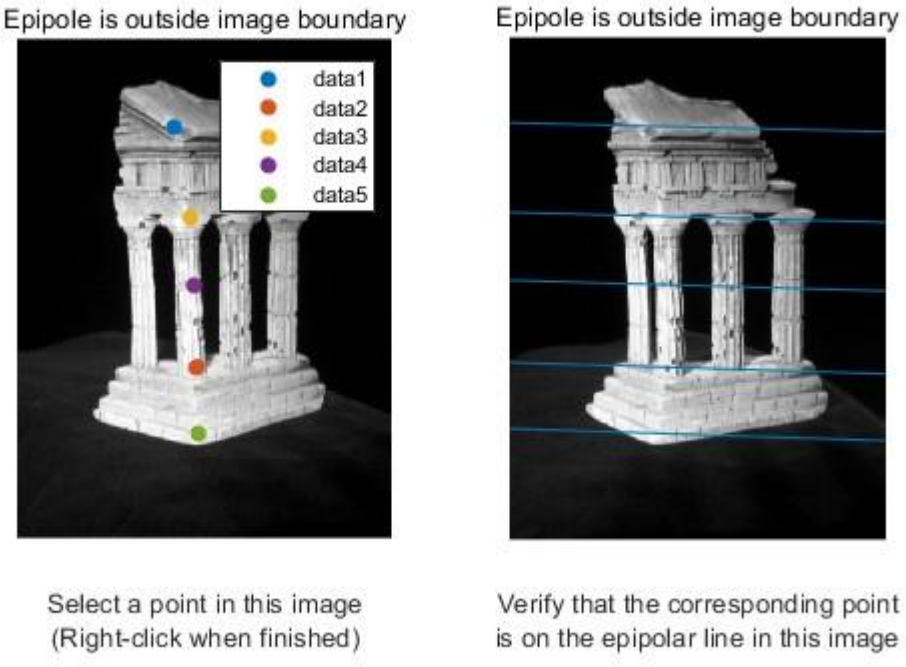
Late day: 1

3.1.1:

The recovered F is shown below:

1.96850760899073e-09	-1.04019931015479e-08	-1.08430195333271e-05
-7.24491404925752e-08	4.64029142279524e-10	0.000500073016894471
1.86868110322887e-05	-0.000481018259900046	-0.00195204075662263

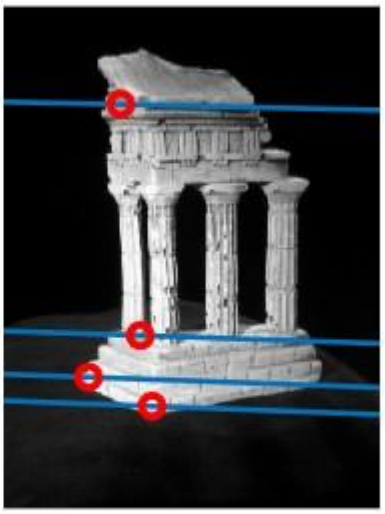
And after running displayEpipolarF(), I chose five points that are mostly on edges. The result figure is shown below.



3.1.2:

When I wrote function epipolarCorrespondence() function, I found that if I go over all points on im2 along with epipolar line l', there will be a slightly bigger error occurring on points matching. Below is the result when I wrote the loop to let it go over all points.

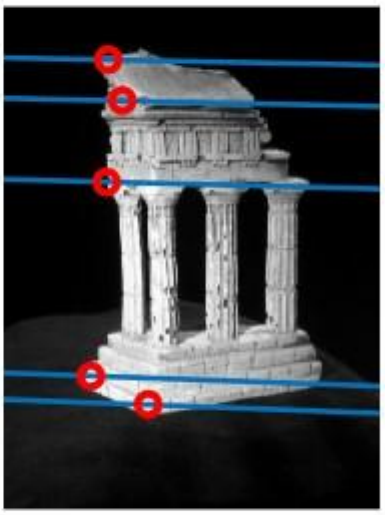
Select a point in this image
(Right-click when finished)



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

It can be seen in the resulting images, points that are on the edges, which can be referred as “points without similar points”, will most likely be matched correctly. But points that have “similar points” will most likely be wrongly recognized(like shown below).

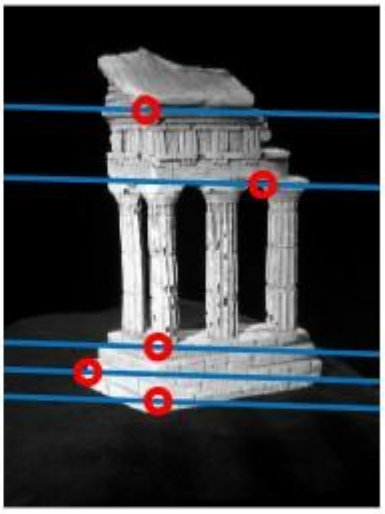
Select a point in this image
(Right-click when finished)



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

Then I tried to only search points that are around x1 in im2, I set the boundary as x1-30 to x1+30. The resulting image is 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

It can be seen that compared with the previous implementation, with this method the accuracy increased obviously.

Probable reason for this:

with going over all points along the l', if the correct point has a very tiny difference with the recorded min_distance points, due to machine error, the points will not be renewed to the correct one. I thought to make the condition statement become equal or less: if distance <= min_distance, but in this case, if the correct point is before the one that has the same distance, it will be renewed to that point, which also causes inaccuracy.

The similarity metric:

I used Euclidean distance as my similarity metric. Instead of only taking just the points, I take a window around the target point. Compute distance by subtracting, .^2 and sqrt(), then sum all elements in the result matrix, setting it to be the distance.

3.1.3

E: (after running testTempleCoords.m)

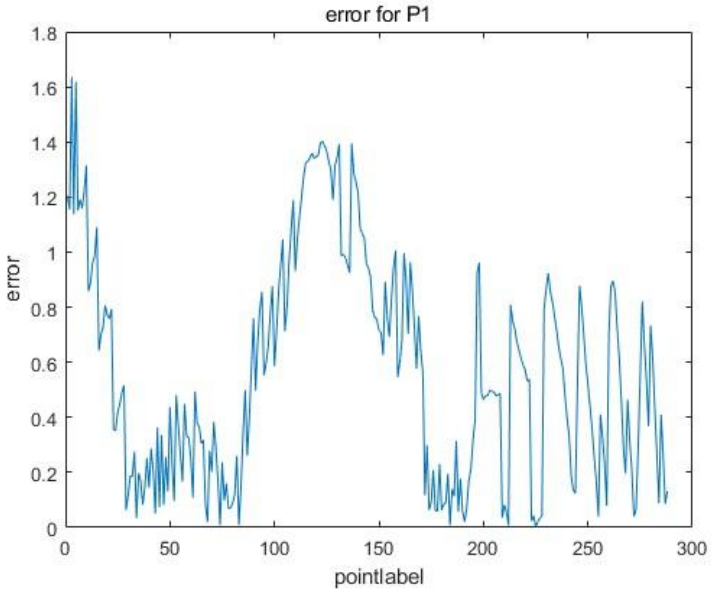
0.00404956244132063	-0.0433080372767723	-0.0191554874996293
-0.149794366553689	-0.000936326071206377	0.726416434975663
0.00186296855297908	-0.735240786278836	-0.000846576656319945

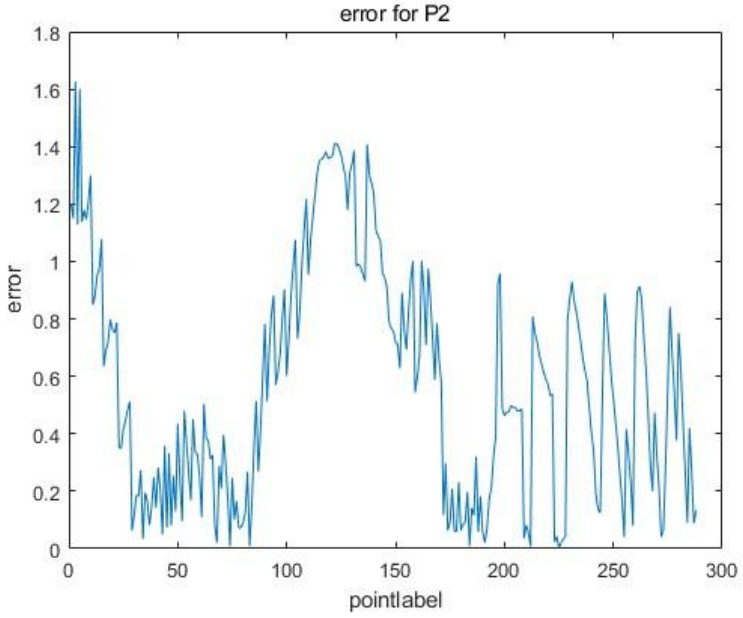
3.1.4

Method to determine extrinsic matrix:

As stated in the project description, I first used candidate extrinsic matrix to get a projection matrix(by multiplying intrinsic matrix with extrinsic matrix). Then use this projection matrix to get pts3d (the N*3 matrix), for each output, the number of positive depths is calculated (i.e., number of pts3d(:,3) > 0). The biggest one will be chosen as the correct pts3d and its extrinsic matrix is the correct one.

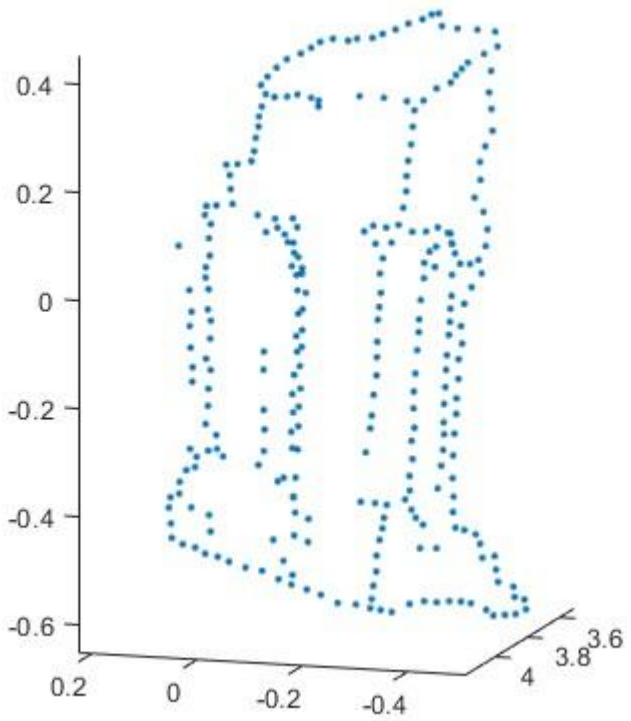
Below is the figure for points-distance

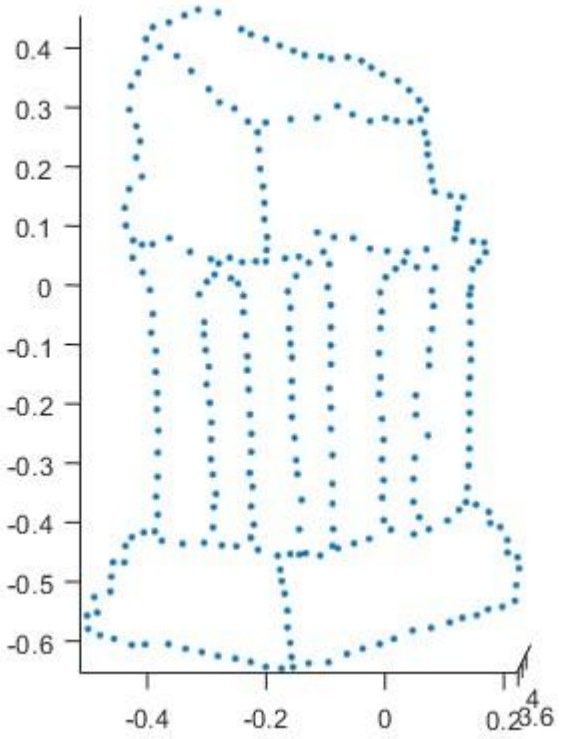


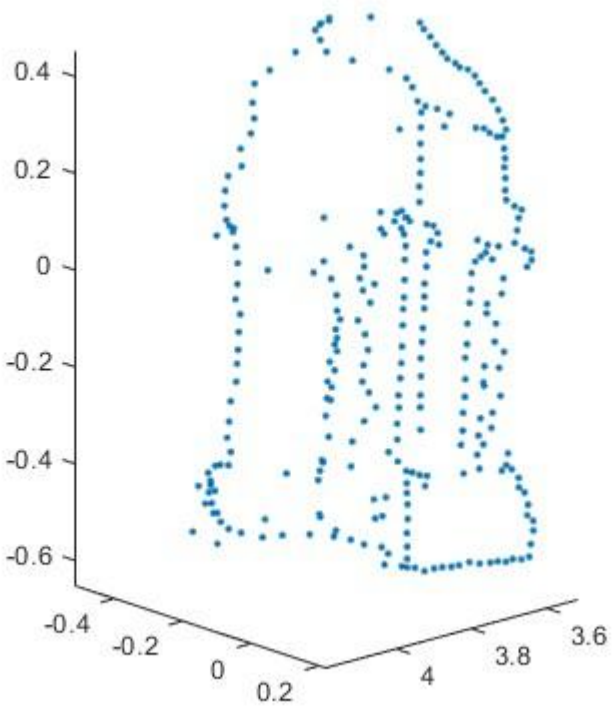


Although there are some points that have errors greater than 1, most points have reprojection errors less than 1.

3.1.5
Below is the figures through three different angles

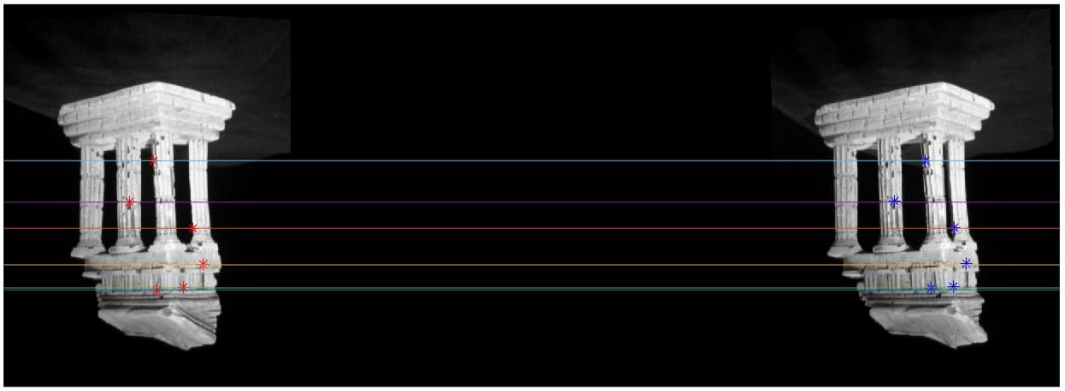






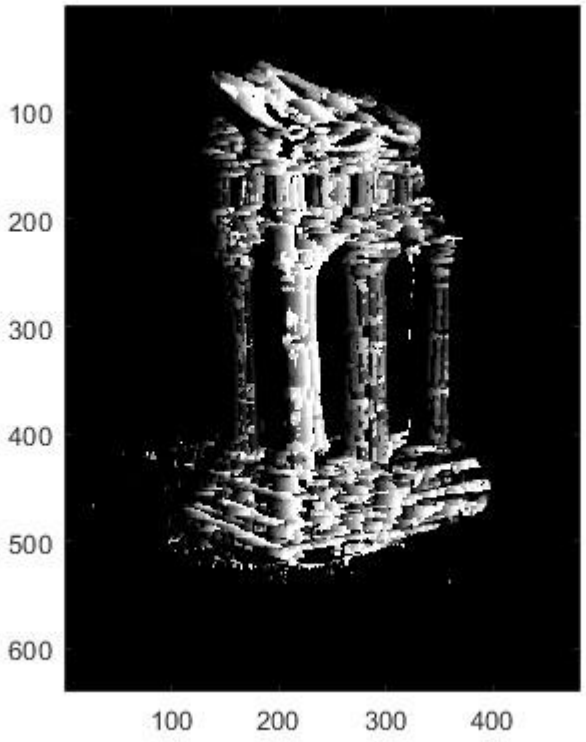
3.2.1

The result is shown below

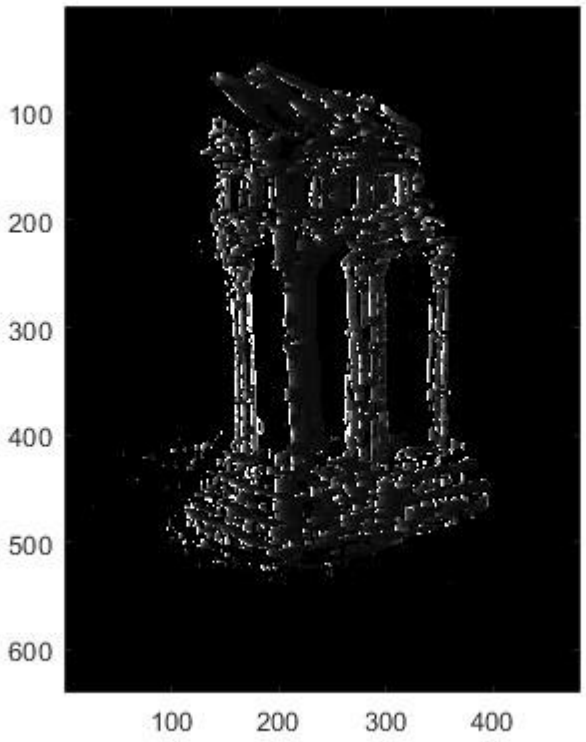


3.2.2

The result will be generated from testDepth.m in part 3.2.3, The result figure is shown below:



3.2.3
The result is shown below:



3.3.1
The result of running testPose.m is shown below:

```
>> testPose
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 2.3516
Pose Error with noisy 2D points is 0.1831
>>
```

3.3.2

Result is shown below.

```
>> testKRt
Intrinsic Error with clean 2D points is 141.4402
Rotation Error with clean 2D points is 1.4893
Translation Error with clean 2D points is 0.8475
------------------------------
Intrinsic Error with clean 2D points is 141.4402
Rotation Error with clean 2D points is 1.4428
Translation Error with clean 2D points is 0.8065
>>
```