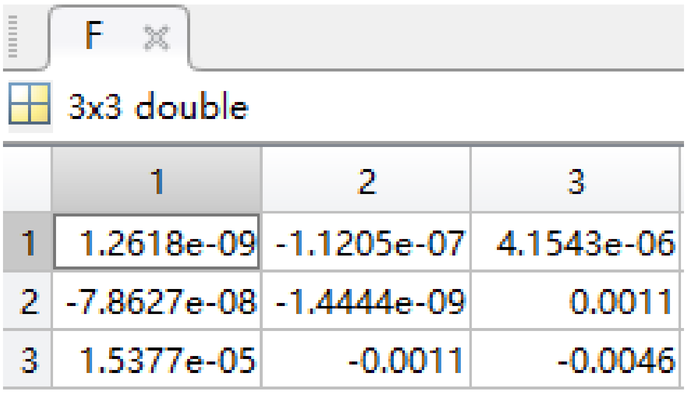
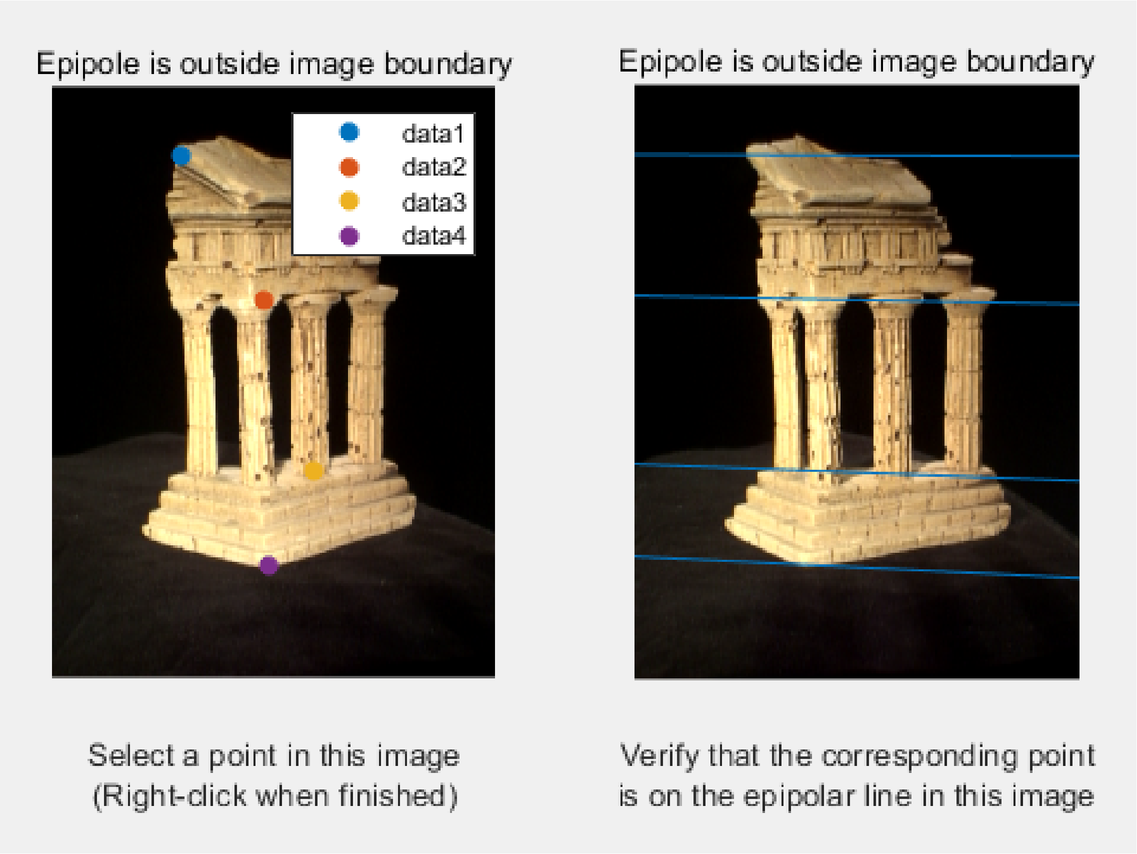
CMPT412
Project4
Xiaohong Xu
301417289

3.1.1 Implement the eight point algorithm (2 pts)



	1	2	3
1	1.2618e-09	-1.1205e-07	4.1543e-06
2	-7.8627e-08	-1.4444e-09	0.0011
3	1.5377e-05	-0.0011	-0.0046



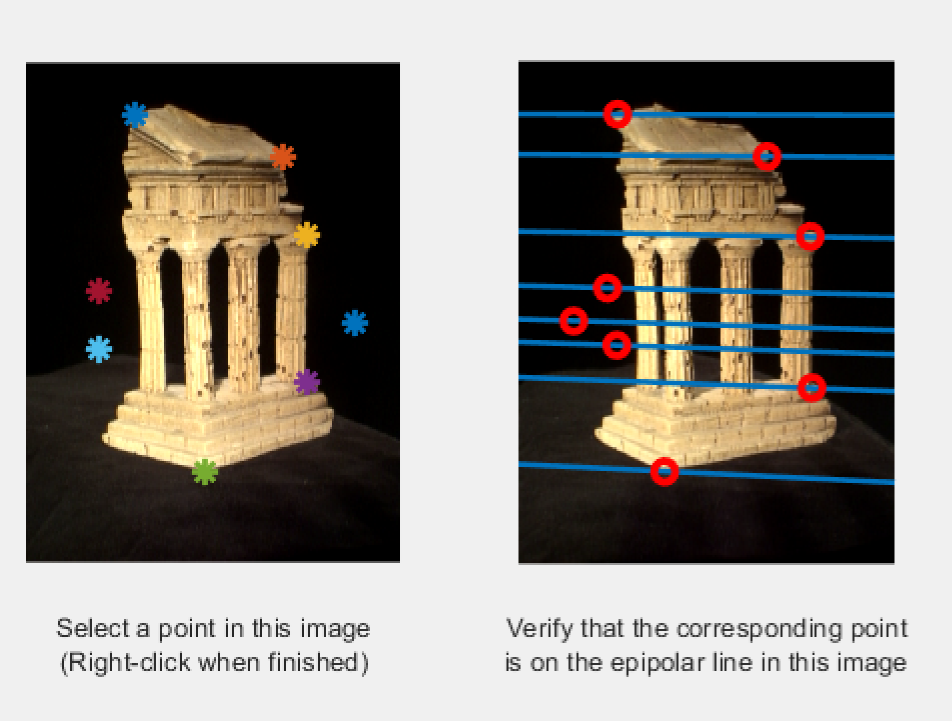
Epipole is outside image boundary

Epipole is outside image boundary

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

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

3.1.2 Find epipolar correspondences (2 pts)

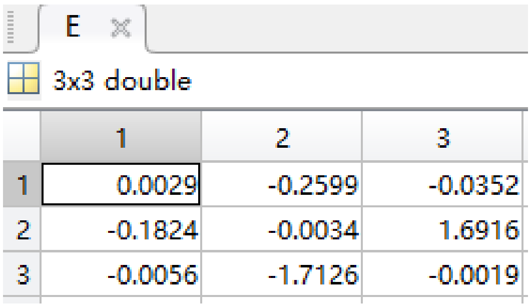


Select a point in this image
(Right-click when finished)

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

I used the Euclidean distance to determine the similarity by comparing the points within a specific window around the central point. Initially, I used a window size of 9, which yielded favorable results. However, I later switched to a window size of 4, which achieved the same goal with less computational effort.
Upon testing, my algorithm proved ineffective at identifying the deep blue points which are located in the right of the black area due to the close similarity values resulting from the corresponding points being on the same line and in the black area within the window size.

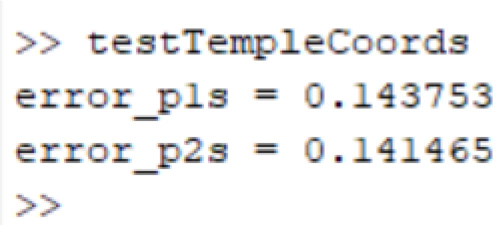
3.1.3 Write a function to compute the essential matrix (2 pts)


E
3x3 double

	1	2	3
1	0.0029	-0.2599	-0.0352
2	-0.1824	-0.0034	1.6916
3	-0.0056	-1.7126	-0.0019

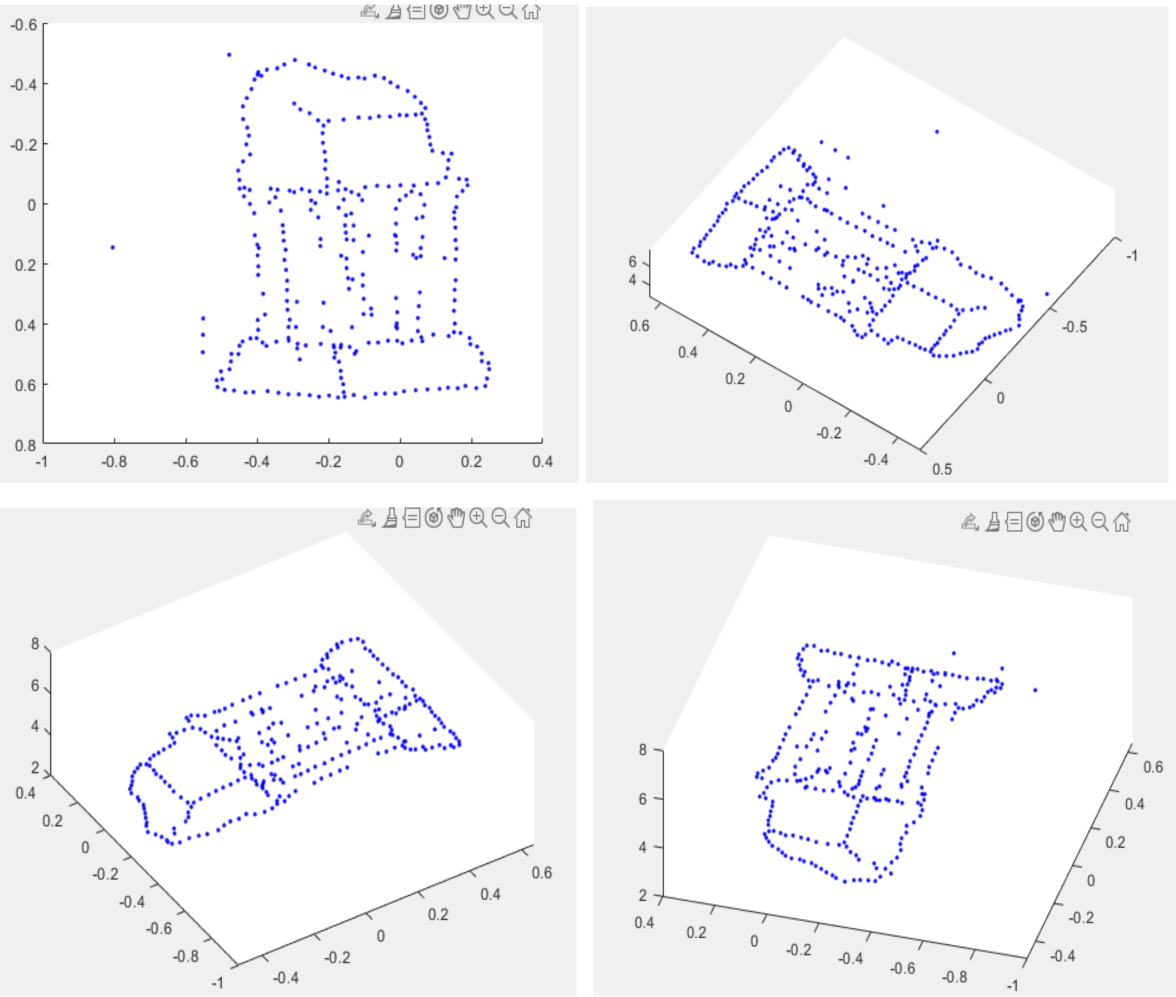
3.1.4 Implement triangulation (2 pts)

If the Z value of all the resulting 3D points is positive, it indicates that the extrinsic matrices are accurate, according to my verification method.

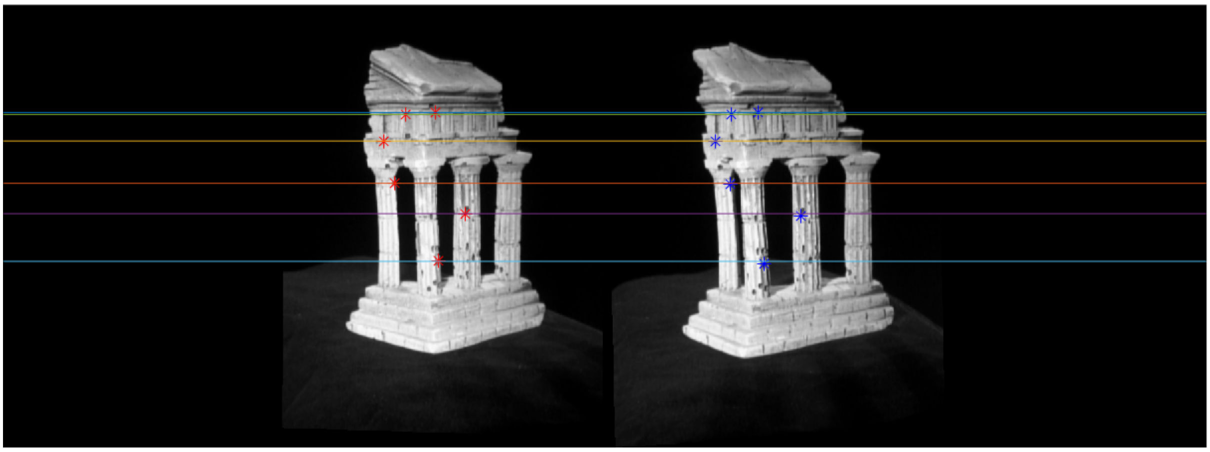


```
>> testTempleCoords
error_p1s = 0.143753
error_p2s = 0.141465
>>
```

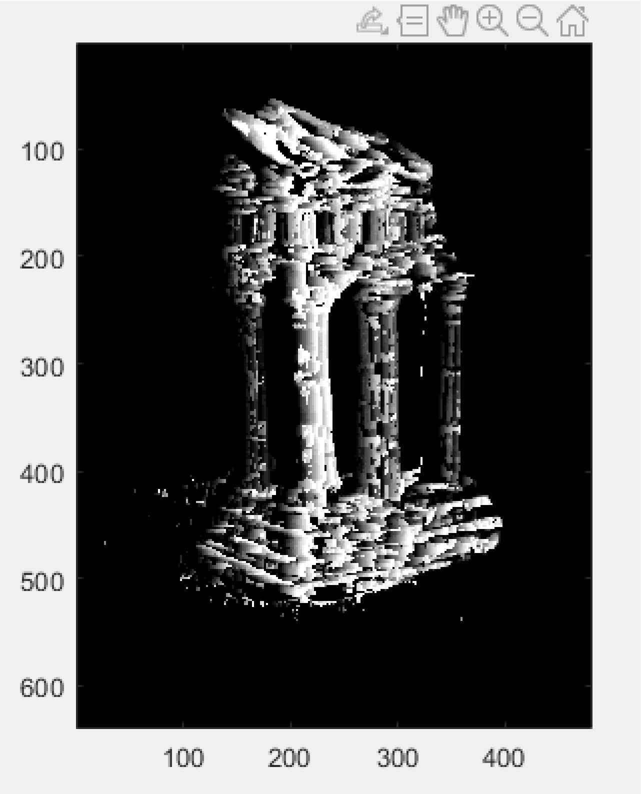
3.1.5 Write a test script that uses templeCoords (2 pts)



3.2.1 Image rectification (2 pts)



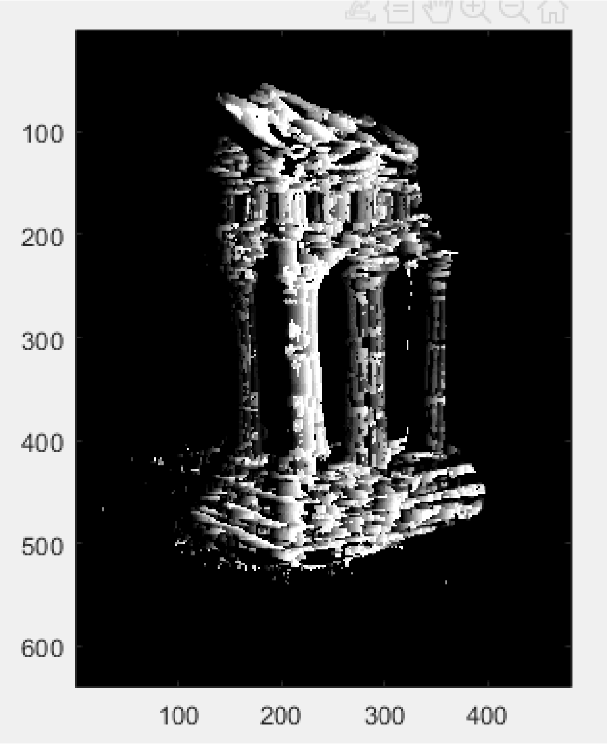
3.2.2 Dense window matching to find per pixel density (2 pts)

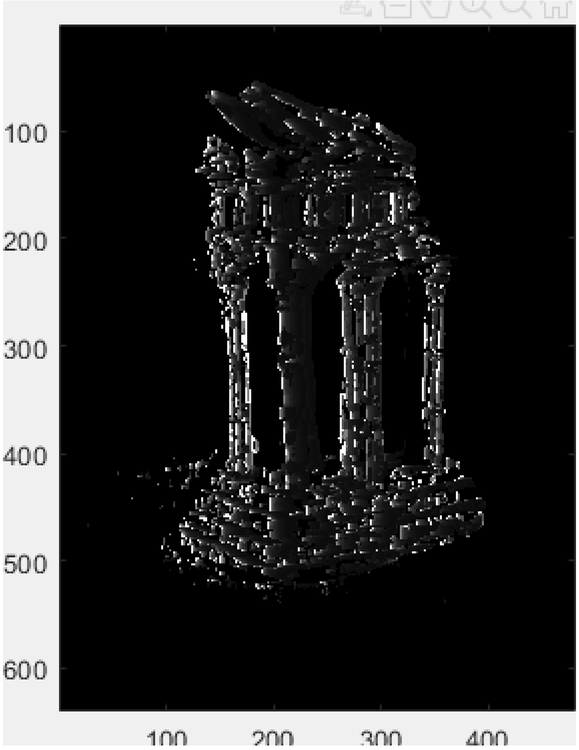


3.2.3 Depth map (2 pts)

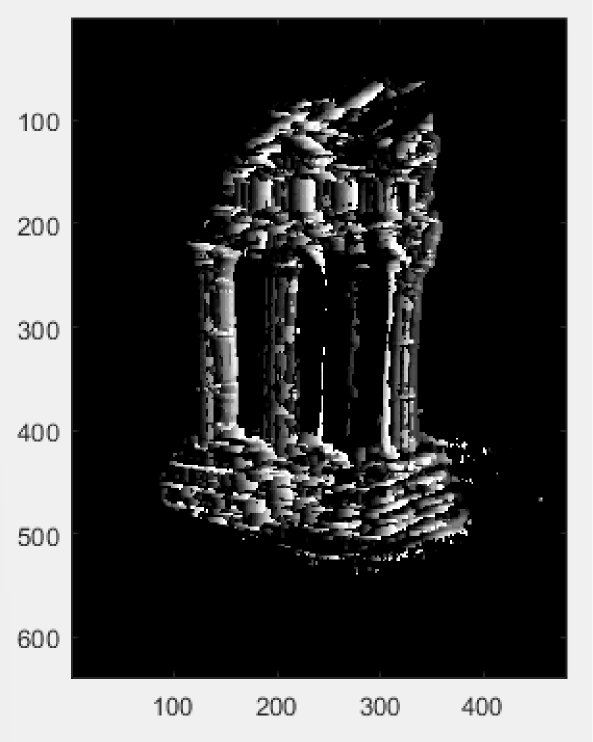
disparity **depth**

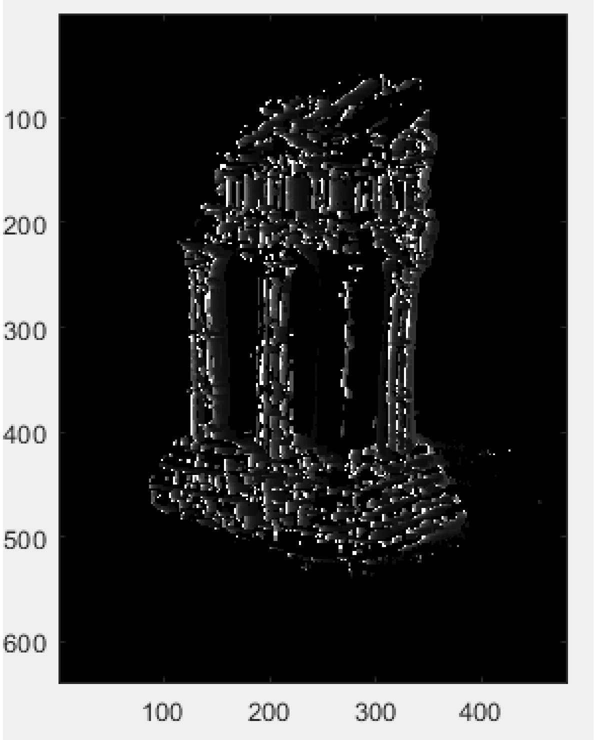
Before rectification:



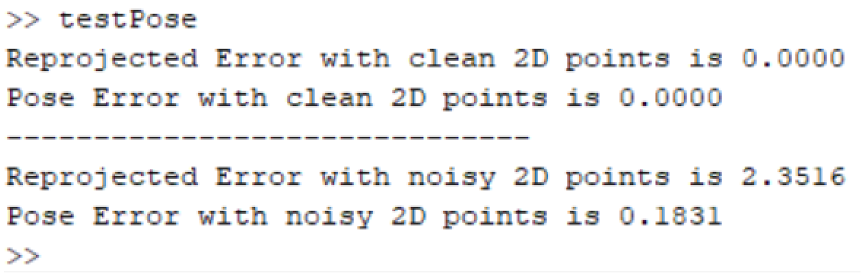


After rectification:



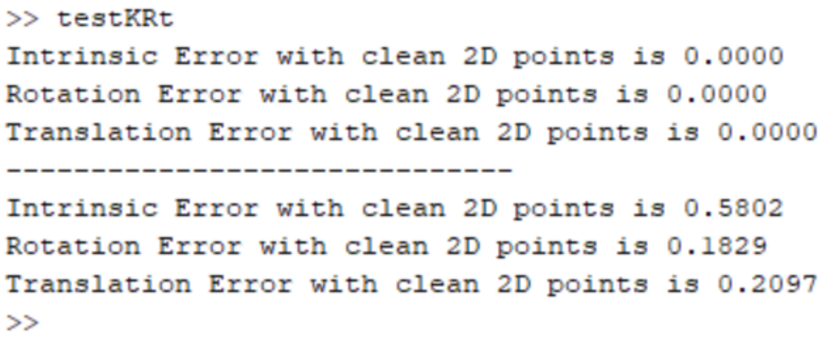


3.3.1 Estimate camera matrix P (2 pts)



```
>> 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 Estimate intrinsic/extrinsic parameters (1 pts)



```
>> testKRt
Intrinsic Error with clean 2D points is 0.0000
Rotation Error with clean 2D points is 0.0000
Translation Error with clean 2D points is 0.0000
------------------------------
Intrinsic Error with clean 2D points is 0.5802
Rotation Error with clean 2D points is 0.1829
Translation Error with clean 2D points is 0.2097
>>
```