Part 2: Augmented Reality with Planar Homographies

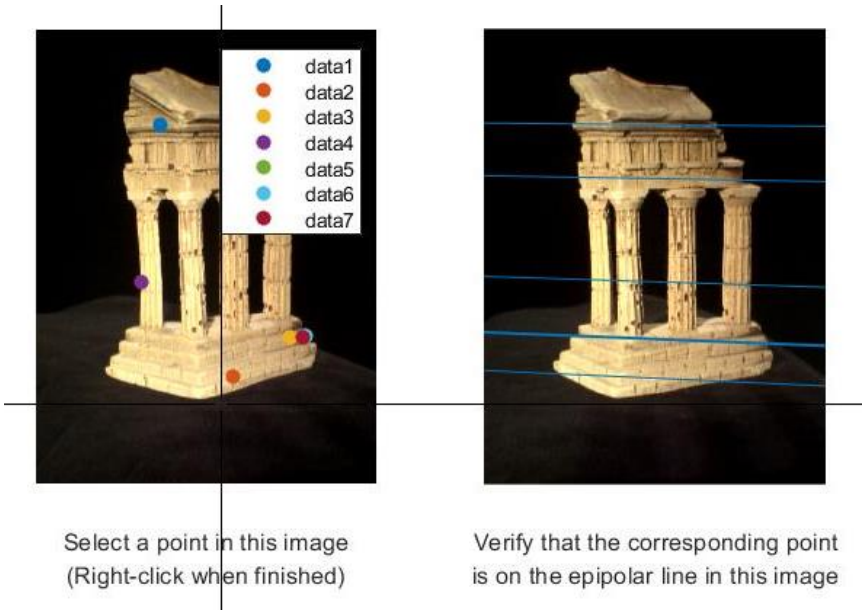
Date: 2019-5-20

- ## 2.1.1 Implement the eight point algorithm

The output of the fundamental matrix:

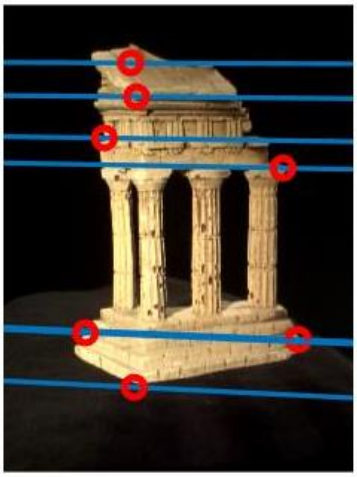
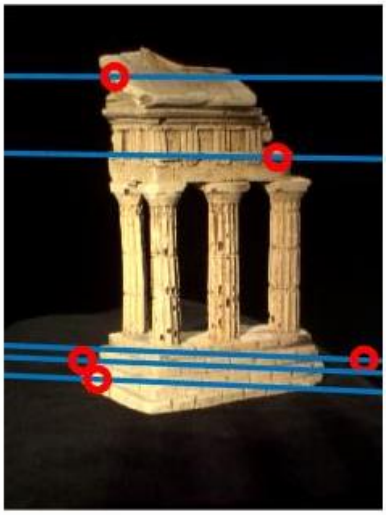
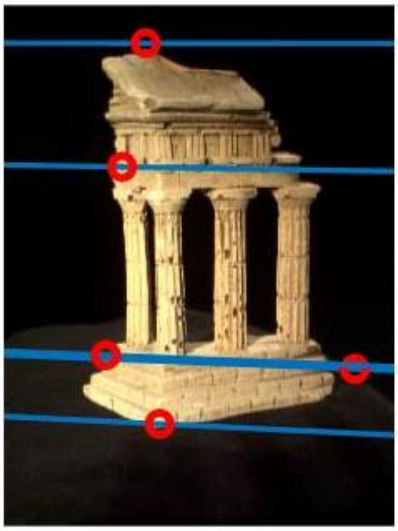
```
F =

    0.0000   -0.0000    0.0000
   -0.0000   -0.0000    0.0011
    0.0000   -0.0011   -0.0046
```

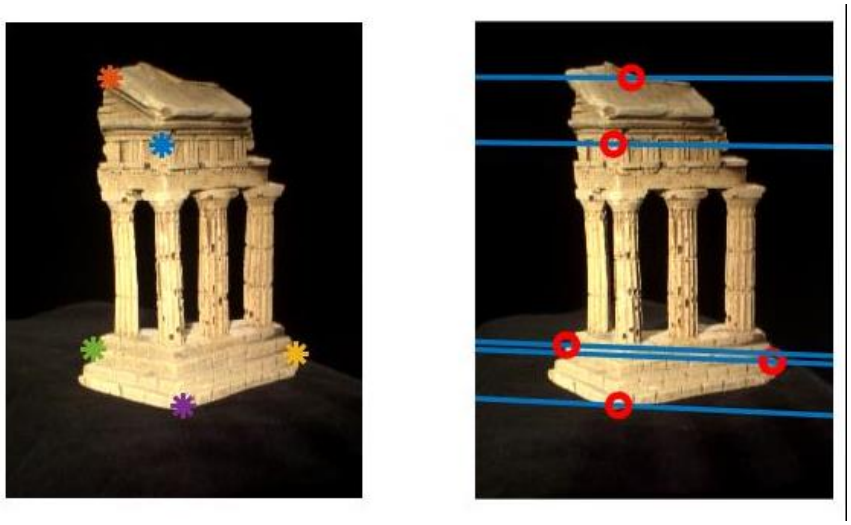
the visualization of some epipolar lines:



- ## 2.1.2 Find epipolar correspondences

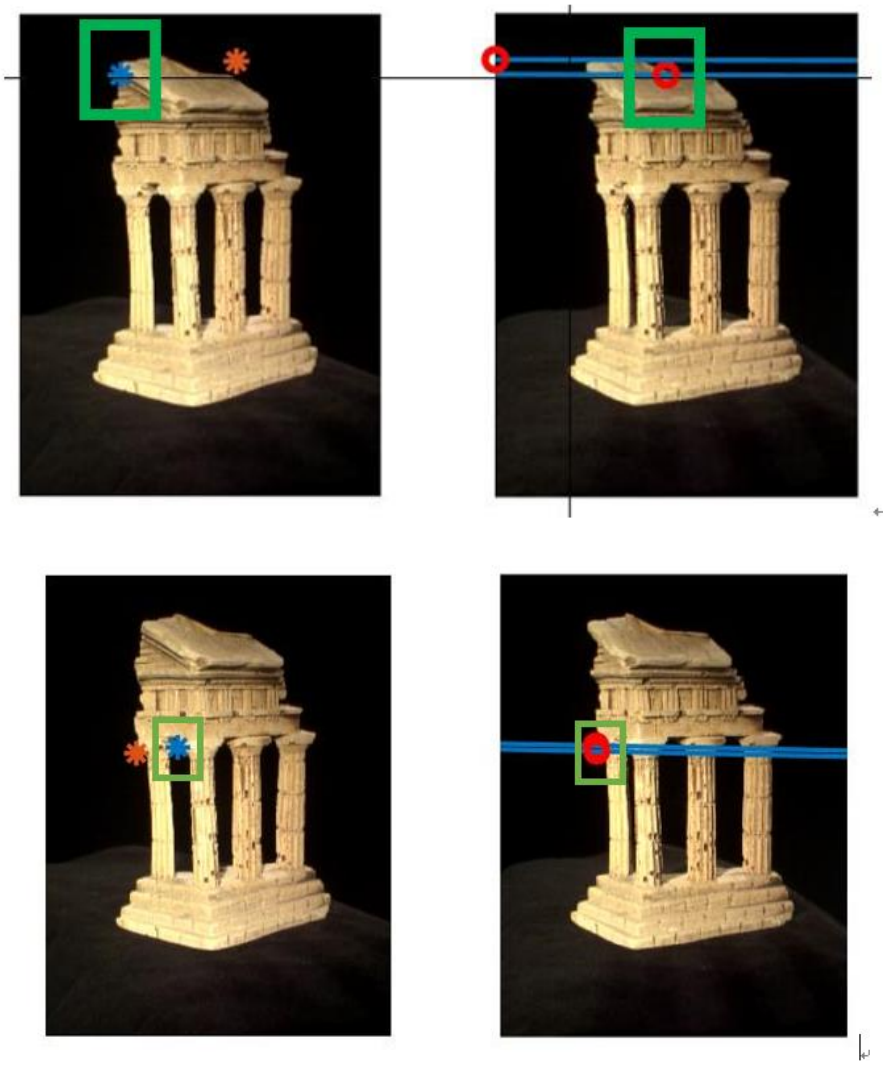
In this section, we need to find the corresponding points in the second image, which have highest similarity score, so that the ways to define the similarity between two points are really important.

Euclidean distance	 
Chebychev distance [∑(a[i] - b[i])^p]^(1/p) (i = 1, 2, …, n) p→∞	 
Cityblock distance (Manhattan)	 

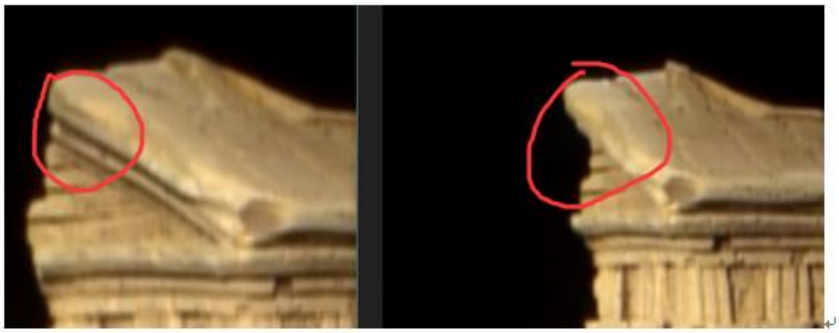
Squaredeuclidean distance	

- And for each algorithm, there are also some fail cases, take the following result that uses Euclidean distance as an example.

  The mistake may be caused by the fact that we can't see some parts of the first image in the second image, so that the algorithm failed.

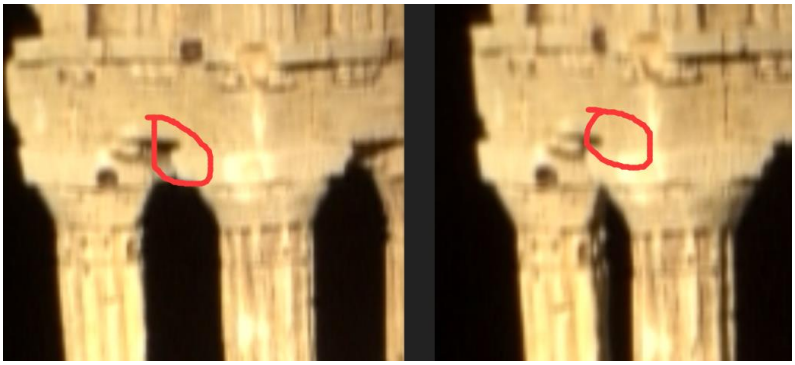


The possible cause of the problem:



The area in red circle is visible

The area in red circle is covered , so that it is invisible



- 2.1.4 Implement triangulation

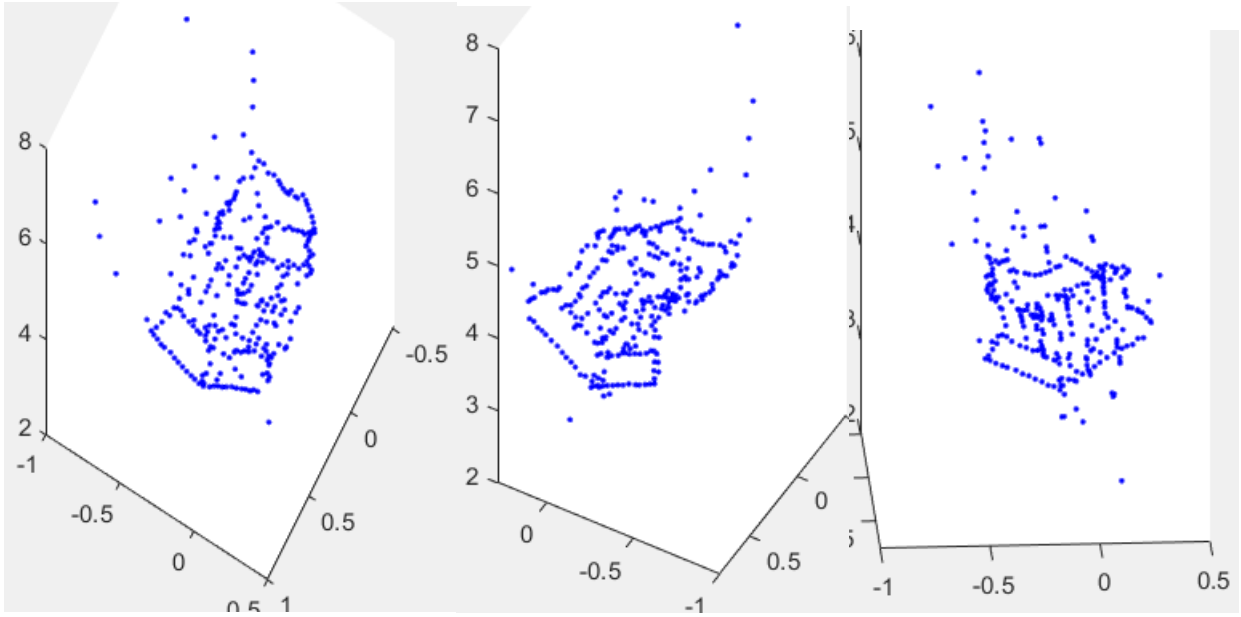
To determine which extrinsic matrices is correct, I choose the matrix that has the most correct configuration (3D points are in front of both cameras), if 2 matrix have the same correct configuration, I compare their re-projection error, choose the one that has less re-projection error. From the output of the script, we can see that those points that generated by the correct extrinsic matrix are all in front of both cameras, and the average re-projection error for per pair of points is less than 1 pixel.

```
if(correct_cfg<correct_cfg_temp)
    ERROR=ERROR_temp;
     pts3d=pts3d_temp;
      correct_cfg=correct_cfg_temp;
     E_final=E2;
     continue;
end
 if (correct_cfg==correct_cfg_temp&&ERROR>ERROR_temp)
        ERROR=ERROR_temp;
     pts3d=pts3d_temp;
      correct_cfg=correct_cfg_temp;
     E_final=E2;
     continue;
end
```

```
correct_cfg =

   288


ERROR =

  219.3970
```

- 2.1.5 a test script that uses templeCoords

final reconstruction of the templeCoordspoints, from different angles :



- 2.2.1 Estimate camera matrix P

In this section, I used the same strategy as homography estimation by Direct Linear Transform (DLT), what's more, I also used the ransac algorithm . And the error of reprojected is relatively small.

the output of the script 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 1.8674
Pose Error with noisy 2D points is 0.1430
```

```
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 2.4336
Pose Error with noisy 2D points is 0.0964
```

```
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 2.5753
Pose Error with noisy 2D points is 0.2342
```

	Clean 2D points		Noisy 2D points	
	Reprojected error	Pose error	Reprojected error	Pose error
1	0.0000	0.0000	1.8674	0.1430
2	0.0000	0.0000	2.4336	0.0964
3	0.0000	0.0000	2.5753	0.2342

- 2.2.2 Estimate intrinsic/extrinsic parameters

the output of the script testKRt.m

```
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 0.9739
Rotation Error with clean 2D points is 0.5671
Translation Error with clean 2D points is 0.5508
```

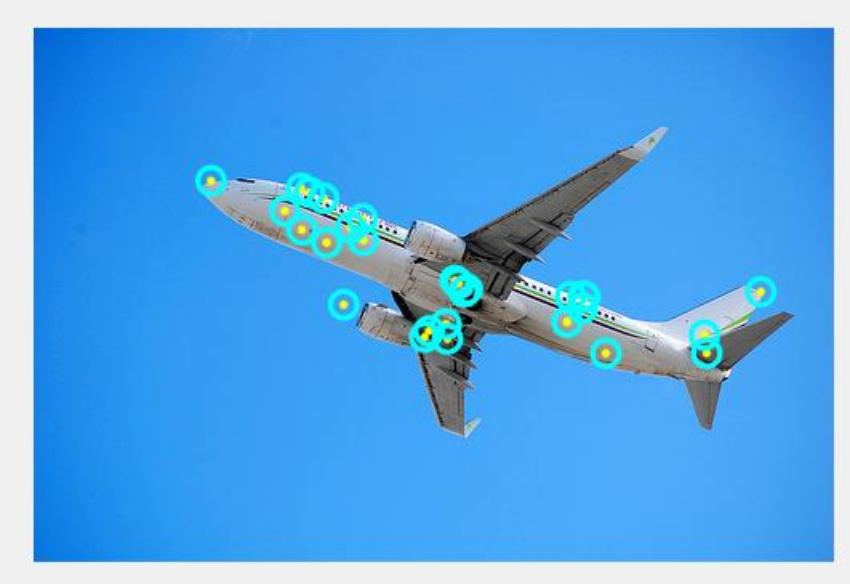
```
Intrinsic Error with clean 2D points is 0.0000
Rotation Error with clean 2D points is 0.0000
Translation Error with clean 2D points is 0.0000
----------------------------
Intrinsic Error with clean 2D points is 0.9352
Rotation Error with clean 2D points is 0.9889
Translation Error with clean 2D points is 2.1816
```

```
Intrinsic Error with clean 2D points is 0.0000
Rotation Error with clean 2D points is 2.0000
Translation Error with clean 2D points is 2.2774
----------------------------
Intrinsic Error with clean 2D points is 0.9306
Rotation Error with clean 2D points is 2.0000
Translation Error with clean 2D points is 2.3644
```

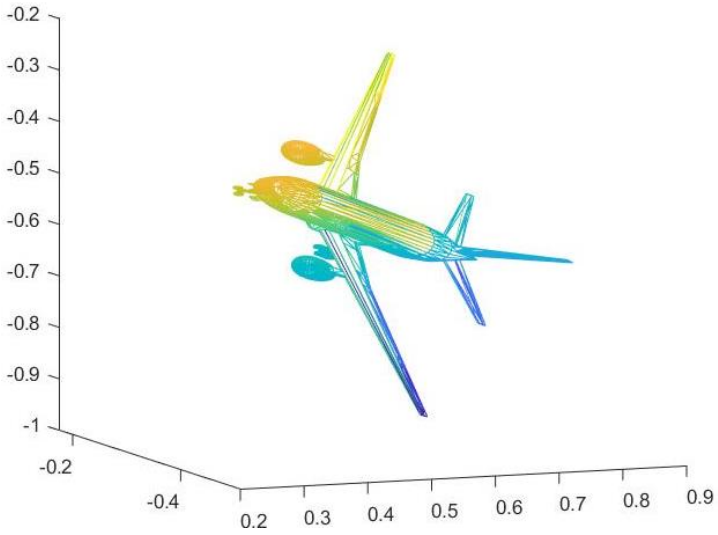
	Clean 2D points			Noisy 2D points		
	intrinsic error	Rotation error	Translation error	intrinsic error	Rotation error	Translation error
1	0.0000	0.0000	0.0000	0.9739	0.5671	0.5508
2	0.0000	0.0000	0.0000	0.9352	0.9889	2.1816
3	0.0000	2.0000	2.2774	0.9306	2.0000	2.3644

- 2.2.3 Project a CAD model to the image

Plot the given 2D points x and the projected 3D points on screen



Draw the CAD model rotated by estimated rotation R

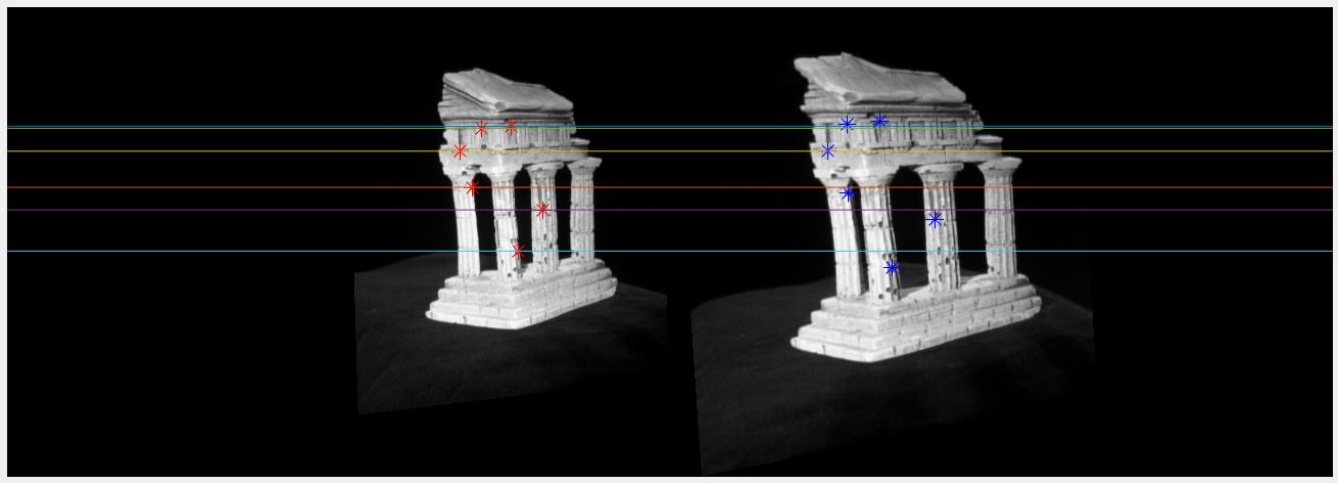


Project the CAD's all vertices onto the image and draw the projected CAD model overlapping with the

2D image:

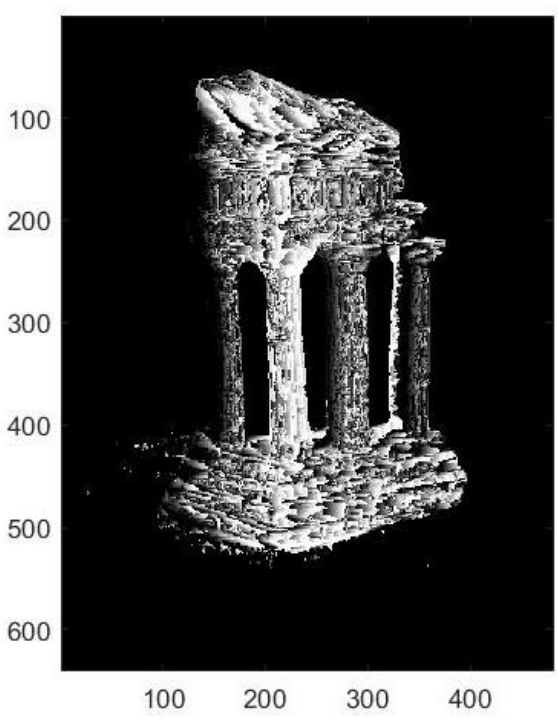
- 2.3.1 Image rectification

a screenshot of the result of running testRectify.m

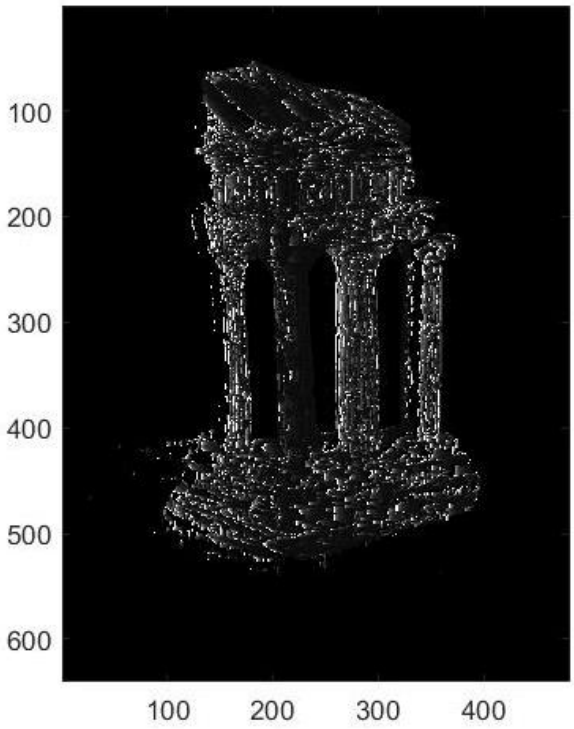


We can see some epipolar lines that are perfectly horizontal, with corresponding points in both images lying on the same line.

- 2.3.2&2.3.3 Dense window matching to find per pixel density & Depth map



disparity map



depth map