

# Project 5

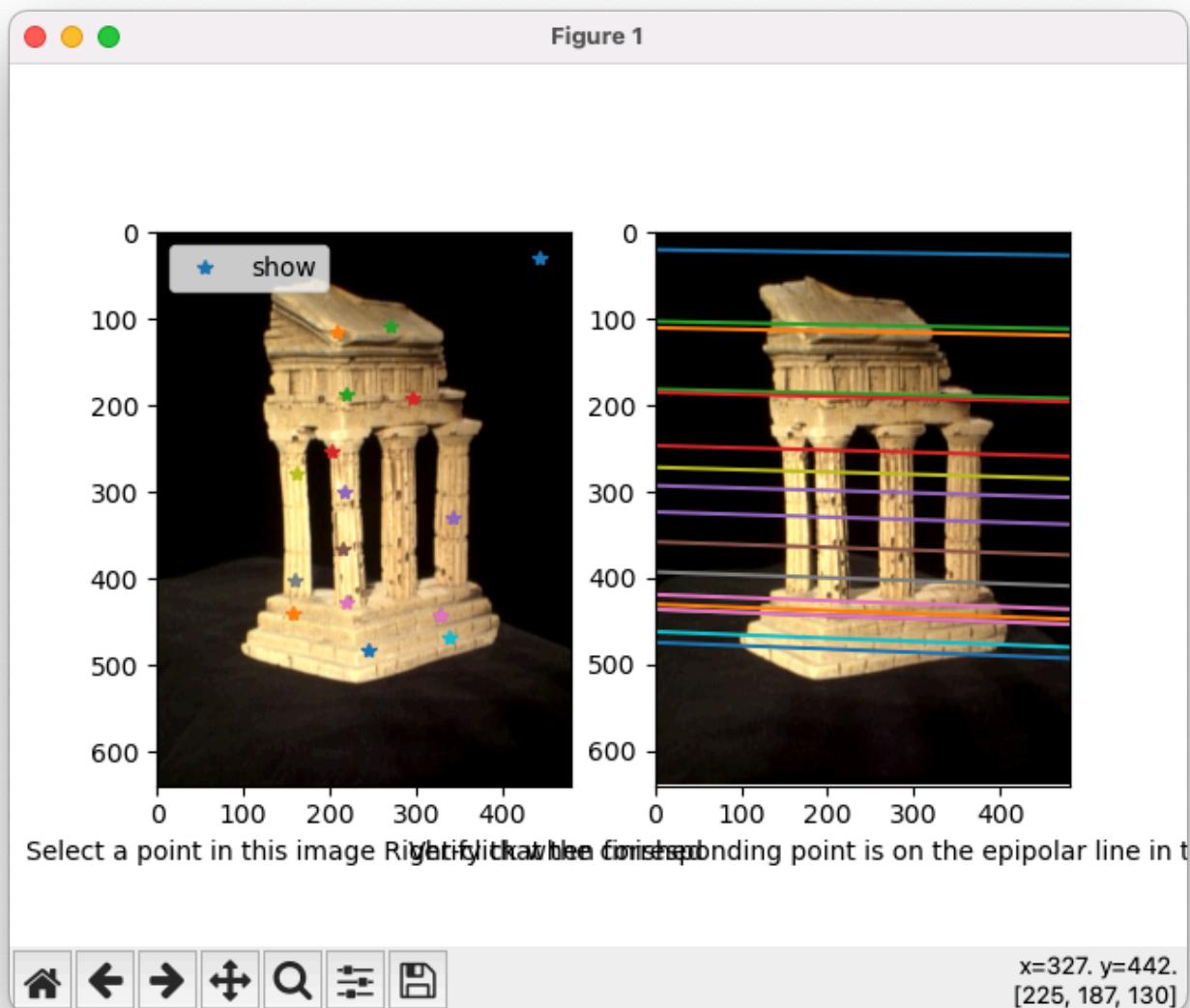
Rongsheng Qian 301449387

### 3.1 Sparse reconstruction

### 3.1.1 Implement the eight point algorithm (2 pts)

```

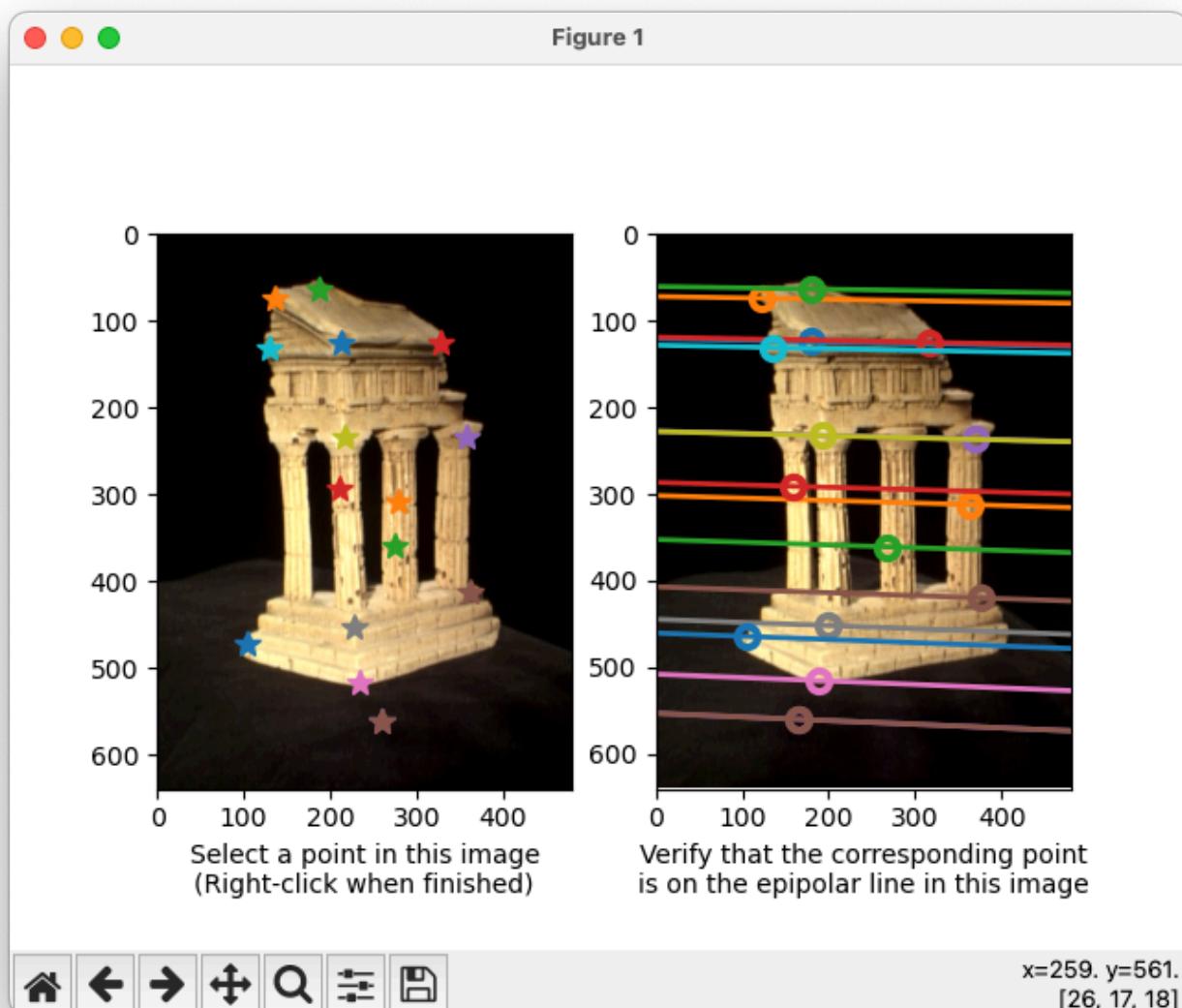
1 F = [[ 3.37554374e-09 -5.77488650e-08 -1.44640637e-05]
2 [-1.29390178e-07 -1.48285427e-09 1.12458135e-03]
3 [ 3.25298454e-05 -1.08034741e-03 -4.55969209e-03]]
```



### 3.1.2 Find epipolar correspondences (2 pts)

The similarity metric you decided to use is NCC. It is robust to Lighting and Shading Variations and can find instances of a smaller template within a larger image.

However it is sensitive to noise in the data. In the following instance, the orange/red point on the pillar in image 1 is matched on the other pillar in image 2. The details of the pillars are similar. In this case, the matching results are easily confused or affected by noise.



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

```
1 | E = [[ 0.00780296 -0.13397612 -0.04211513]
2 | [-0.30018241 -0.00345263  1.6557511 ]
3 | [ 0.00244448 -1.67570084 -0.00192676]]
```

### 3.1.4 Implement triangulation (2 pts)

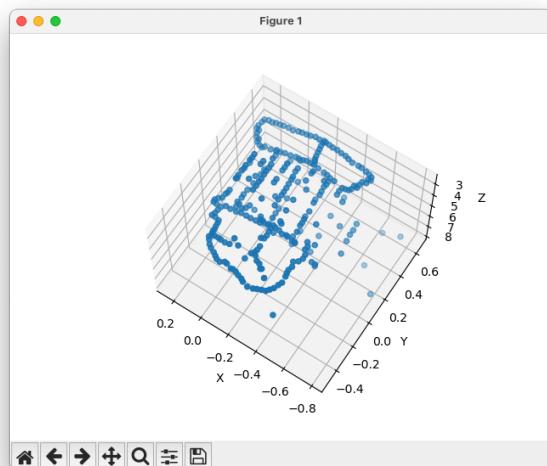
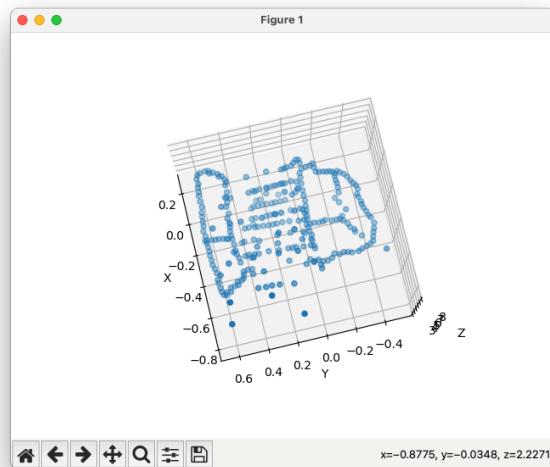
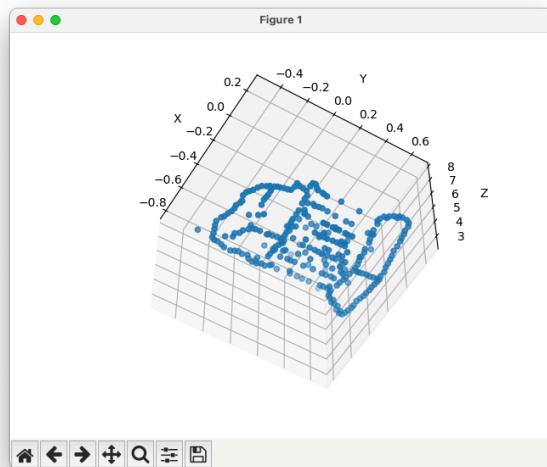
I determined which extrinsic matrices are correct by counting the 3D points which are in front of both cameras (positive depth) in each condition. I picked the one for which most of the 3D points are in front of both cameras (positive depth).

```
1 for i in range(P2_possible_e.shape[-1]):  
2     temp_p2_e = P2_possible_e[:, :, i]  
3     temp_P2 = np.dot(k2, temp_p2_e)  
4     points = triangulate(P1, pts1, temp_P2, pts2)  
5     front_count = 0  
6     for point in points:  
7         if point[-1] > 0:  
8             front_count += 1  
9     if front_count > best:  
10        best = front_count  
11        P2 = temp_P2  
12        R2 = temp_p2_e[:, :-1]  
13        t2 = temp_p2_e[:, -1].reshape(-1, 1)  
14        result_points = points
```

### re-projection error

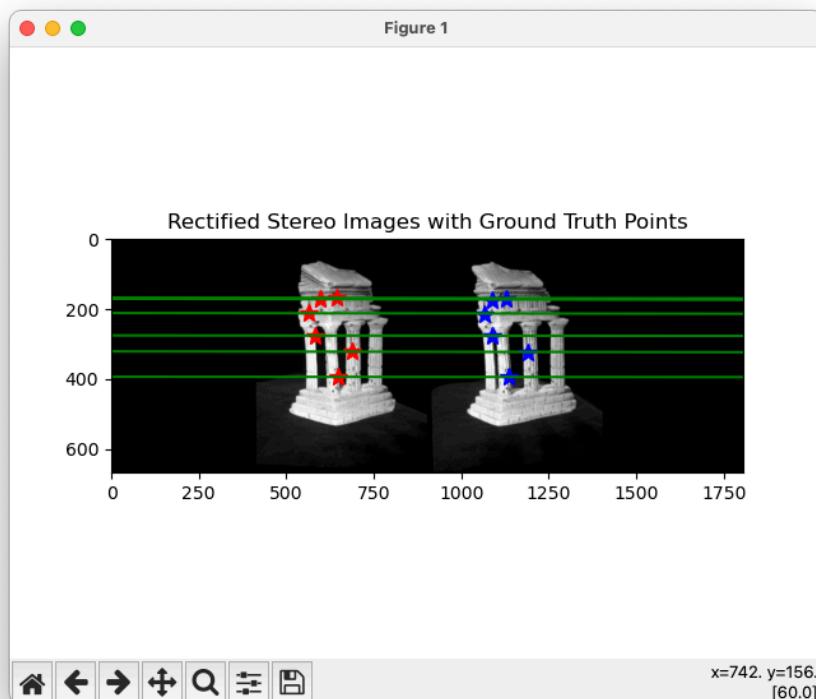
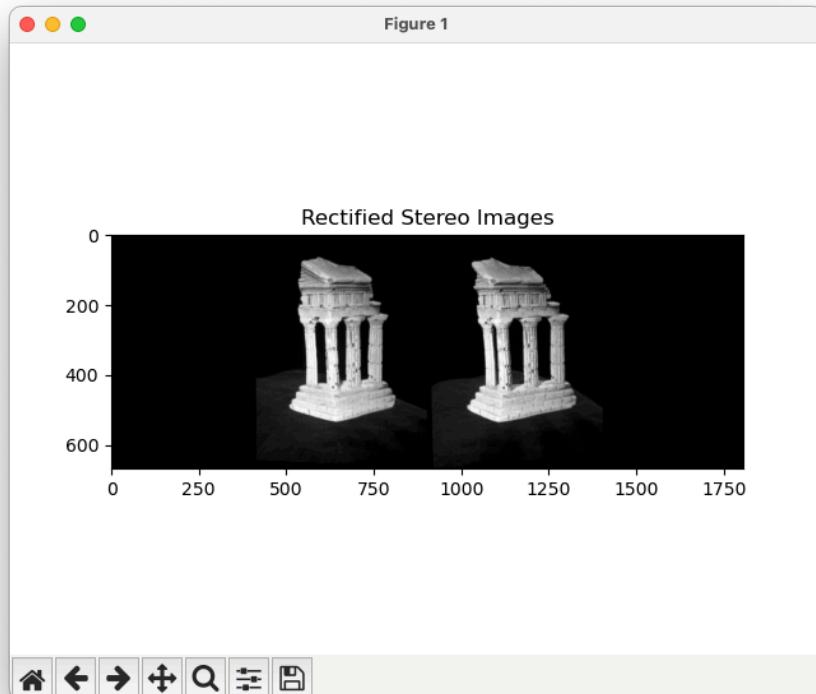
```
1 img1 re-projection error: 0.5527583965853167  
2 img2 re-projection error: 0.5536822966377606  
3 mean re-projection error: 0.5532203466115386
```

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

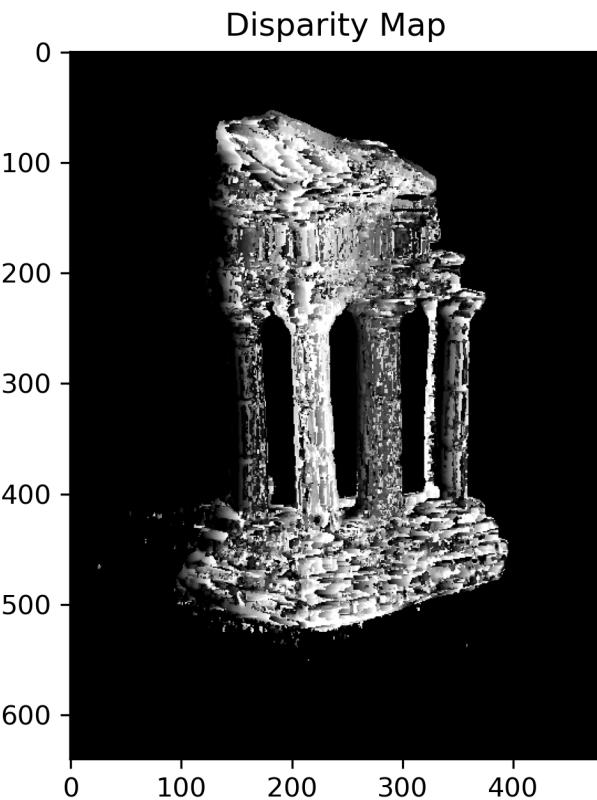


## 3.2 Dense reconstruction

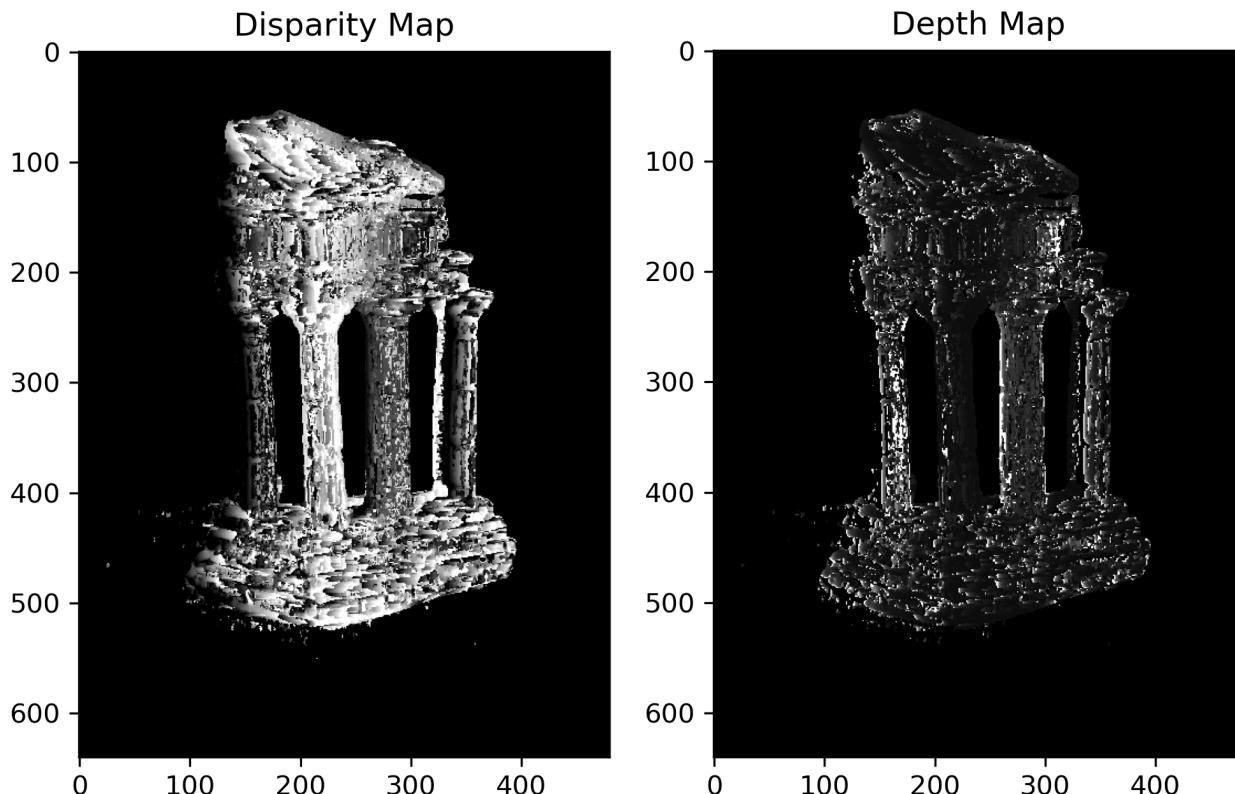
### 3.2.1 Image rectification (2 pts)



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



### 3.2.3 Depth map (2 pts)



## 3.3 Pose estimation

### 3.3.1 Estimate camera matrix P (2 pts)

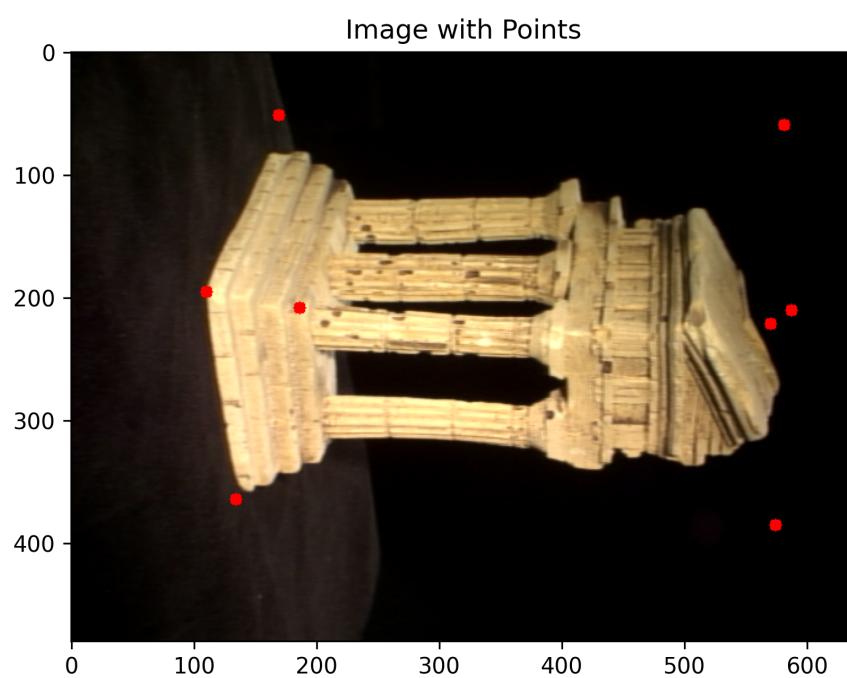
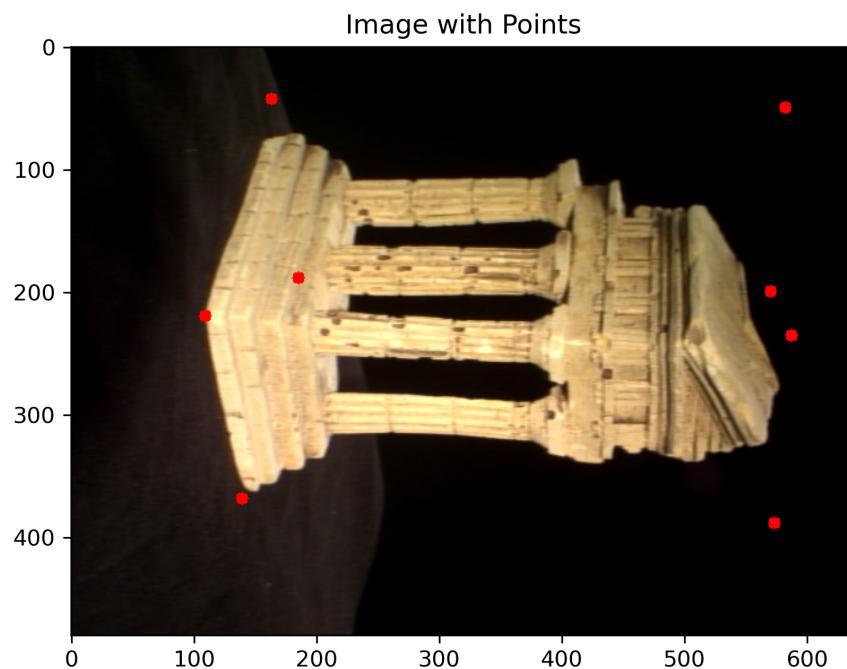
```
1 Reprojected Error with clean 2D points is 0.0000
2 Pose Error with clean 2D points is 0.0000
3 -----
4 Reprojected Error with noisy 2D points is 3.1531
5 Pose Error with noisy 2D points is 0.0135
6
7 Reprojected Error with clean 2D points is 0.0000
8 Pose Error with clean 2D points is 0.0000
9 -----
10 Reprojected Error with noisy 2D points is 2.2296
11 Pose Error with noisy 2D points is 0.0208
12
13 Reprojected Error with clean 2D points is 0.0000
14 Pose Error with clean 2D points is 0.0000
15 -----
16 Reprojected Error with noisy 2D points is 3.2137
17 Pose Error with noisy 2D points is 2.0240
```

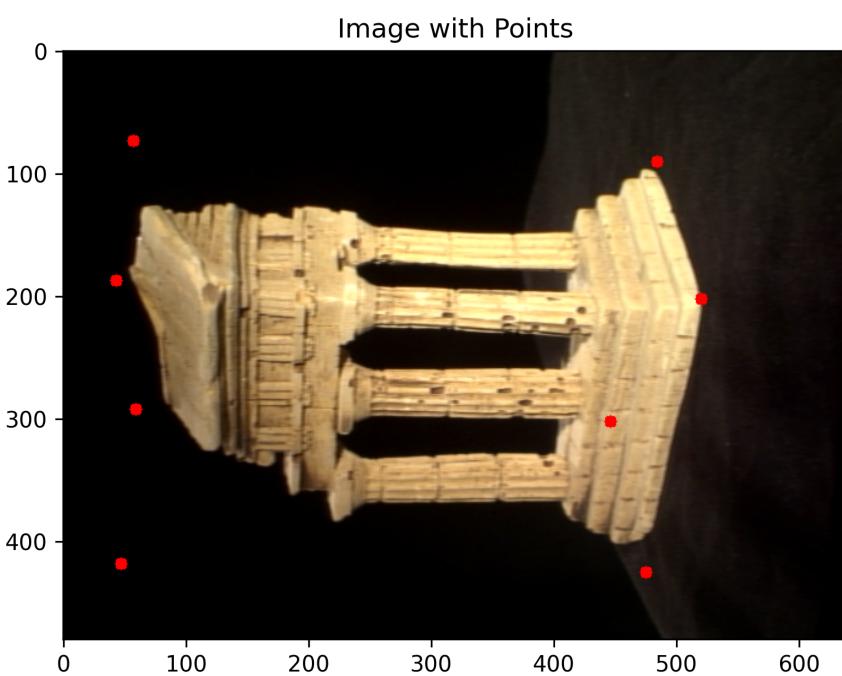
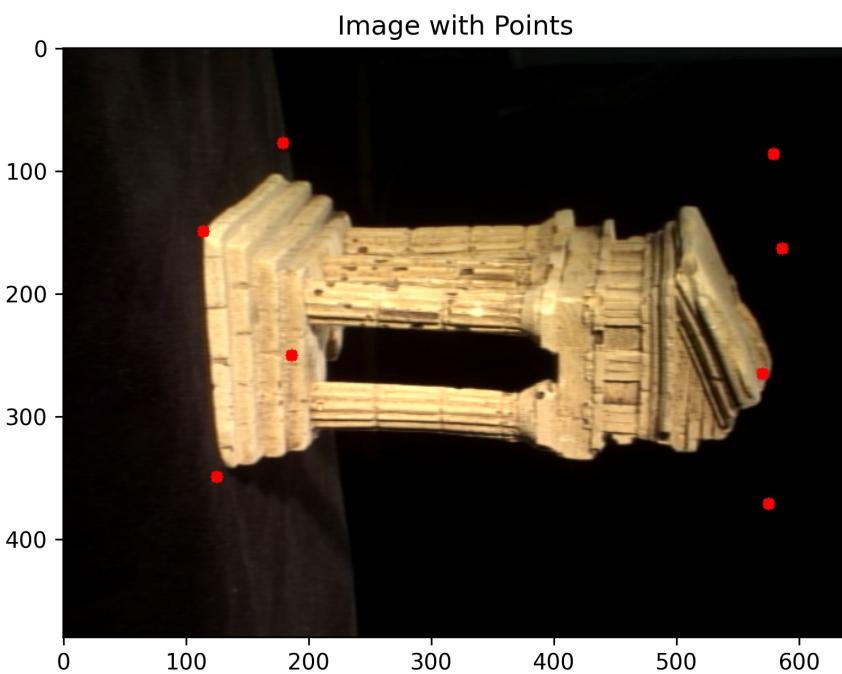
### 3.3.2 Estimate intrinsic/extrinsic parameters (1 pts)

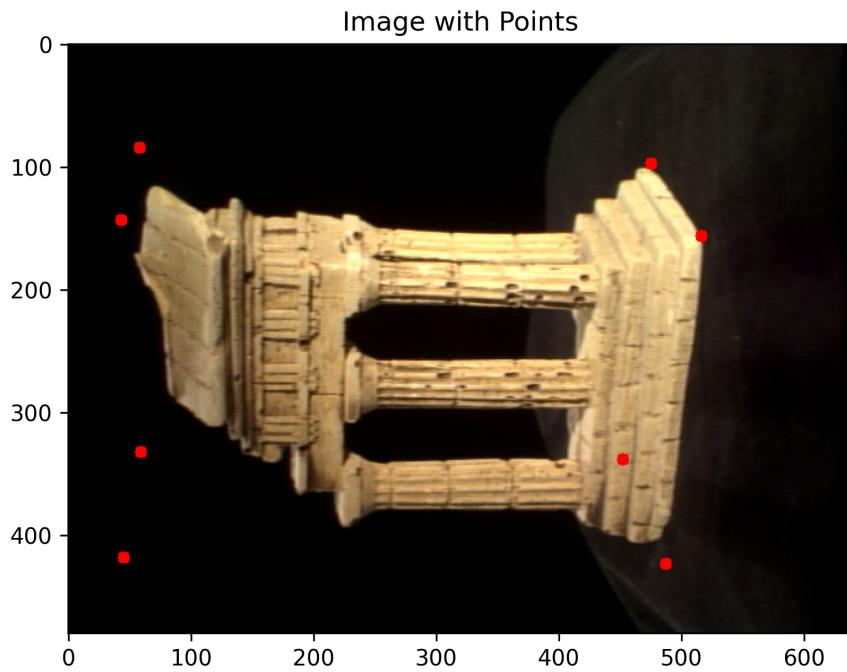
```
1 Intrinsic Error with clean 2D points is 0.0000
2 Rotation Error with clean 2D points is 0.0000
3 Translation Error with clean 2D points is 0.0000
4 -----
5 Intrinsic Error with noisy 2D points is 0.6822
6 Rotation Error with noisy 2D points is 0.0755
7 Translation Error with noisy 2D points is 0.1660
8
9 Intrinsic Error with clean 2D points is 0.0000
10 Rotation Error with clean 2D points is 0.0000
11 Translation Error with clean 2D points is 0.0000
12 -----
13 Intrinsic Error with noisy 2D points is 0.8108
14 Rotation Error with noisy 2D points is 0.0349
15 Translation Error with noisy 2D points is 0.3636
16
17 Intrinsic Error with clean 2D points is 0.0000
18 Rotation Error with clean 2D points is 0.0000
19 Translation Error with clean 2D points is 0.0000
20 -----
21 Intrinsic Error with noisy 2D points is 0.6813
22 Rotation Error with noisy 2D points is 0.0531
23 Translation Error with noisy 2D points is 0.1008
```

## 3.4 Multi-view stereo

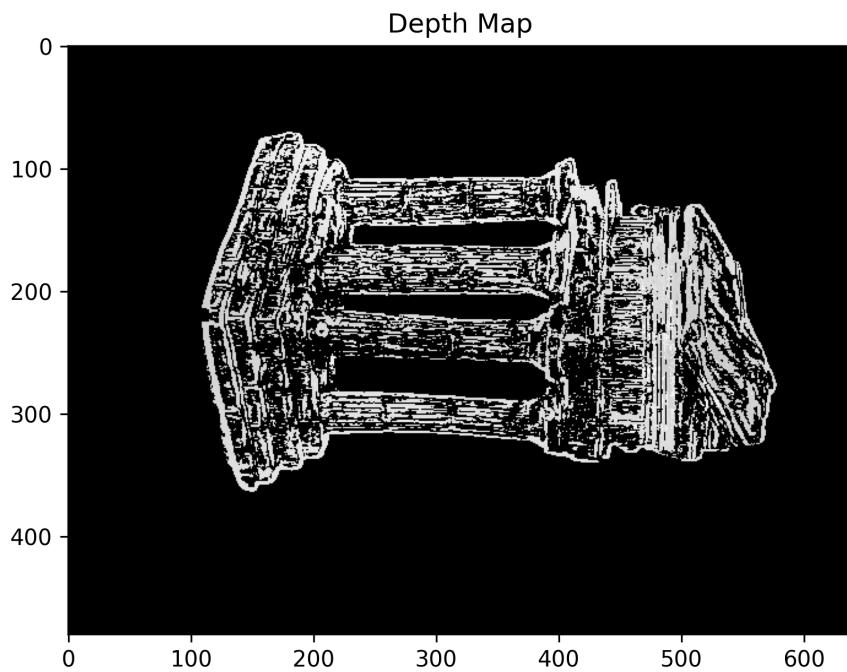
### 3.4.1 (1 pts)







**3.4.2 (1 pts) Here is my image[0] depthmap**



### 3.4.3 (1 pts)

