

***Report Project 4***  
***3-D reconstruction***

**Name:** *Jashanraj Singh Gosain*

**Student Number:** *301435386*

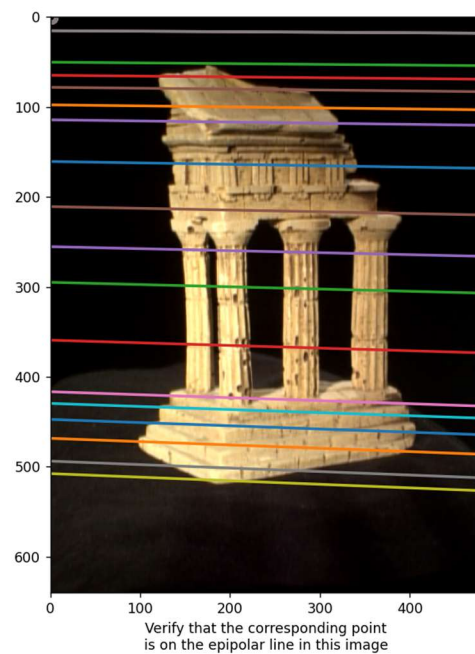
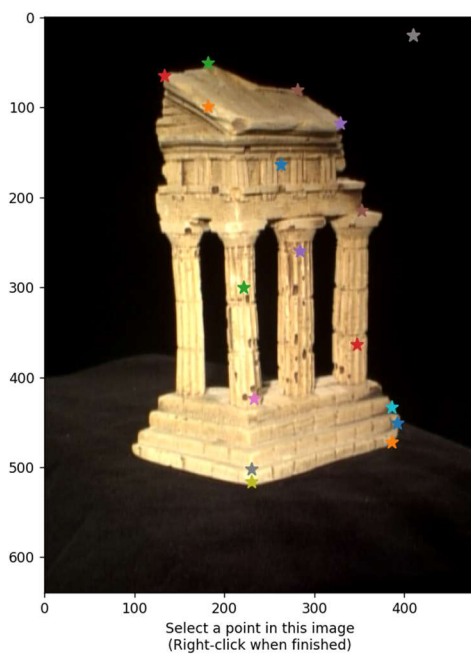
**Late Days:** *0 Used*

### Part 3.1.1:

The value of  $F$  recovered by eight-point algorithm is

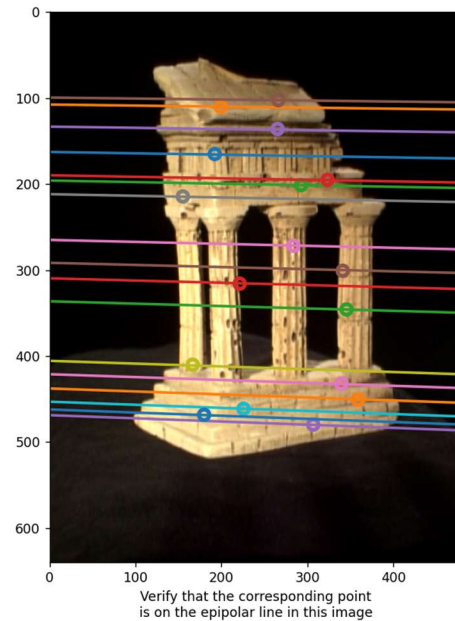
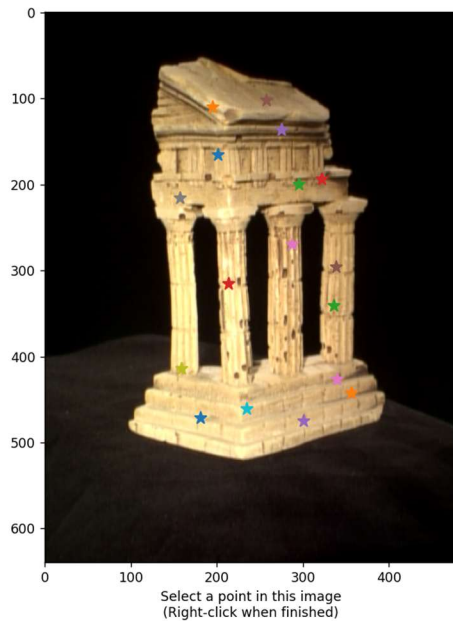
$$F = \begin{bmatrix} -3.40142695e-09 & 1.13381261e-07 & 8.43367433e-06 \\ 1.74890003e-07 & -9.50438978e-10 & -1.71085999e-03 \\ -3.75428408e-05 & 1.64479635e-03 & 6.88923072e-03 \end{bmatrix}$$

Here is the visualization of epipolar lines for each point in the left side image.



### Part 3.1.2 EpipolarCorrespondence

In the implementation of the epipolarCorrespondence function, I used local search strategy to determine corresponding points in the second image based on a provided set of points in the first image and  $F$ . The chosen similarity metric calculates Euclidean distance between image patches at specific points/regions.



### Part 3.1.13 Essential Matrix

My essential matrix calculated from the F, K1 and K2:

$$E = \begin{bmatrix} [-7.86279350e-03 & 2.63042072e-01 & 5.38157544e-02] \\ [4.05741023e-01 & -2.21297437e-03 & -2.53028076e+00] \\ [6.99982274e-03 & 2.56174064e+00 & 2.92577368e-03] \end{bmatrix}$$

## Part 3.1.4 Implement triangulation

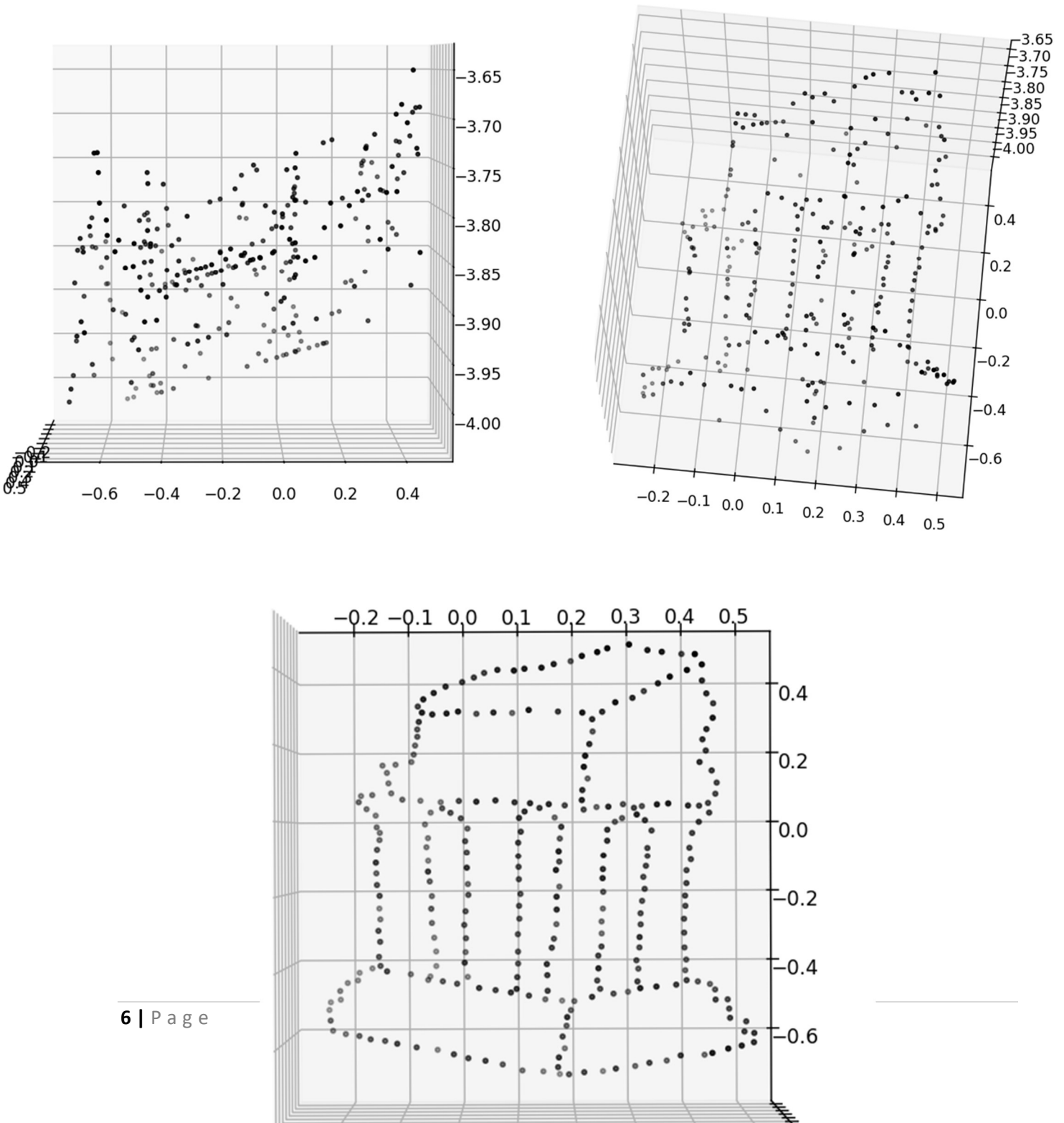
triangulate function calculates 3D points from 2D correspondences using P1 and P2. It minimize the projection error by minimizing Euclidean distance between projection and points.

Projection Errors for each point:

```
Min pts1 error: 0.07760512383122962  
Min pts2 error: 0.07738313207390468
```

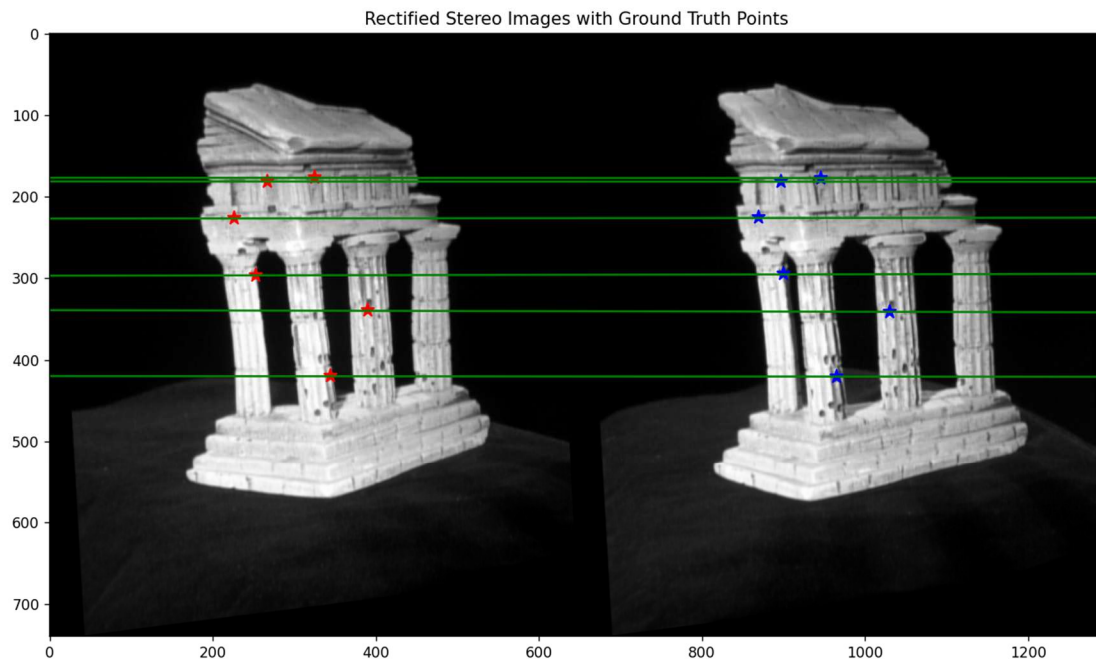
## Part 3.1.5 Script

Here are three images taken at different angles for the templecoords points in 3d



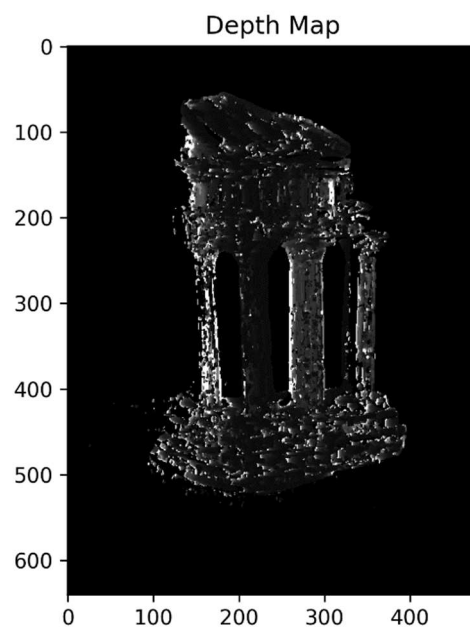
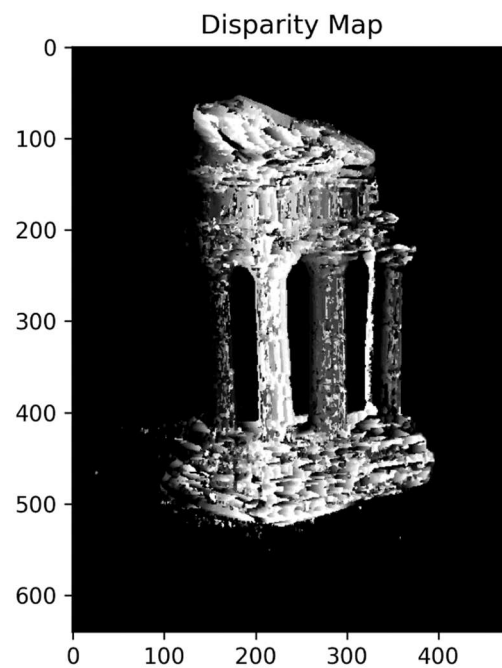
## Part 3.2.1 Rectification

The epipolar lines are perfectly horizontal line indicating that all the points are on the same horizontal line.



## Part 3.2.2 and 3.2.3 Depth and Disparity

My outputs are as follows for the disparity map and depth map:





## Part 3.3.1 and 3.3.2 - Estimate Pose and Params

Output for the testpose.py

```
PS C:\Users\jasha\SFU\CMPT412\Project4\python> python testpose.py
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 11.4180
Pose Error with noisy 2D points is 2.8947
```

Output for the testKRt.py

```
PS C:\Users\jasha\SFU\CMPT412\Project4\python> python testKRt.py
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 noisy 2D points is 1.1983
Rotation Error with noisy 2D points is 0.5068
Translation Error with noisy 2D points is 1.0171
```

## Part 3.4.1

Here is the image of corners displayed on each image.

