CMPT 412 Project 5 Report

Ritika Goyal (301401516)

Project 5: 3D Reconstruction

3.1 Sparse reconstruction

Part 3.1.1 Eight point algorithm

First, I loaded pts1 and pts2 from someCorresp.m and applied eight point algorithm which gave me following F:

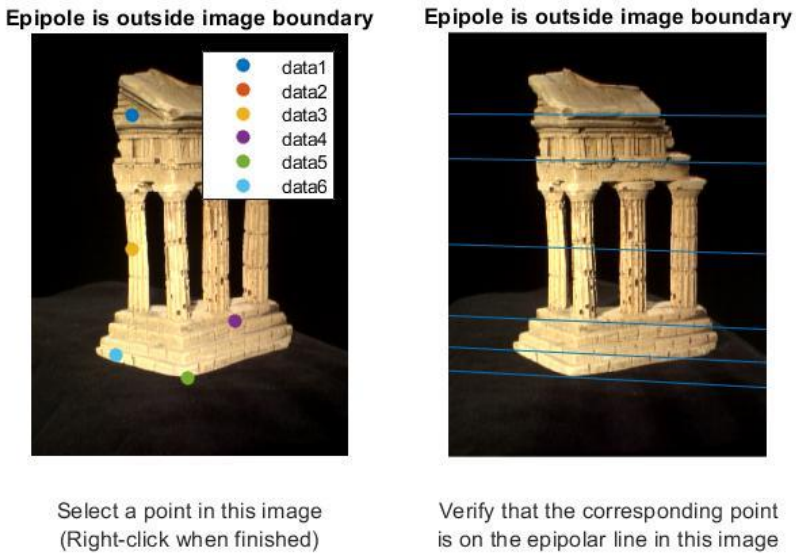
```
Command Window
>> F = eightpoint(pts1, pts2, M)

F =

   -0.0000    0.0000   -0.0000
    0.0000    0.0000   -0.0015
   -0.0000    0.0015    0.0064

fx >>
```

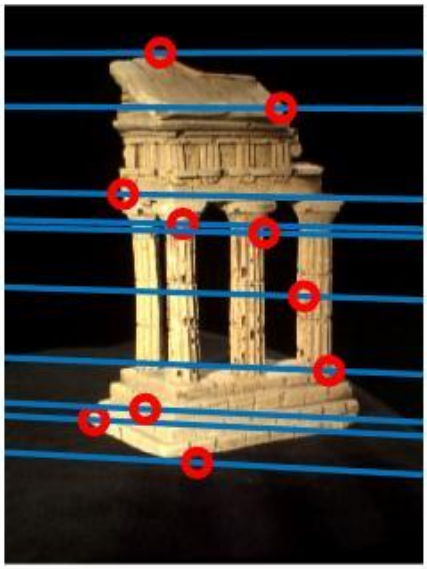
The result after running displayEpipolarF.m is given below:



Part 3.1.2 Epipolar correspondence

To calculate the y point, I converted my epipolar line into slope-intercept form. I took the window size of 5 and thus made a window around original point and searched 10 pixels to left and 10 pixels to right. I then compared the pixel values of a window 5X5 for each candidate of original point and calculated distance between them. The candidate point with the smallest distance was selected. The similarity metric is calculated using Euclidean distance based on difference values of different windows. I tried different window size like 50, 40 and 5, and got the best results with 10 as a window size. The results are provided below:

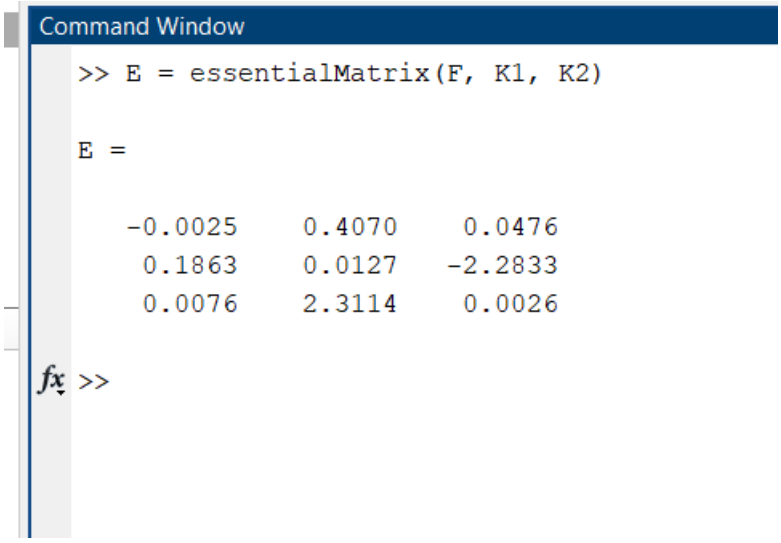
Select a point in this image
(Right-click when finished)



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

Part 3.1.3 Essential Matrix

Obtained Essential Matrix is given below:



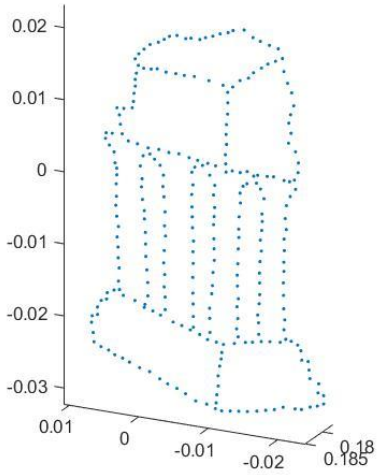
Part 3.1.4 Triangulation

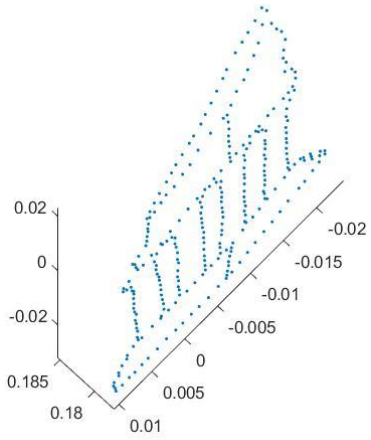
There were 4 extrinsic metrices that were given after calling camera2(E) function. The matrix that was chosen on the basis that have the maximum number of 3D point in front of both the cameras. The correct extrinsic matrix chosen was 4 as it had maximum points in front of both the cameras. It had in total of 288 points correspondences.

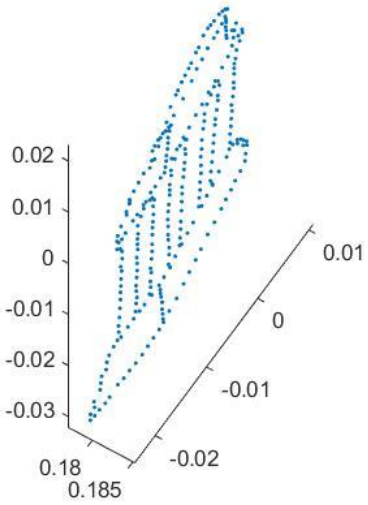
The reprojections error using pts1 was 0.5672 and the reprojection error using pts2 was 0.5643.

Part 3.1.5 TempleCoords

Final reconstruction of templeCoords points from different angles:



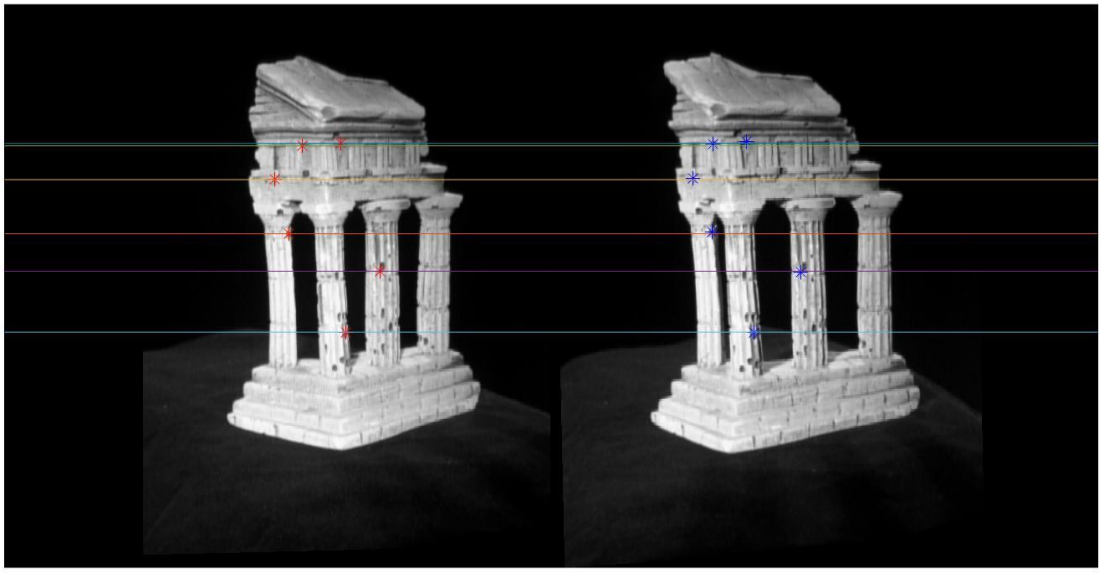




3.2 Dense reconstruction

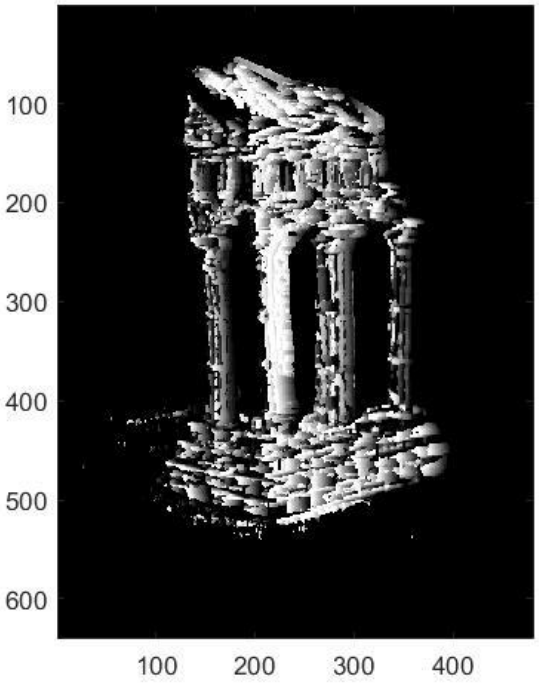
3.2.1 Image Rectification

The results of image rectification are as follows:



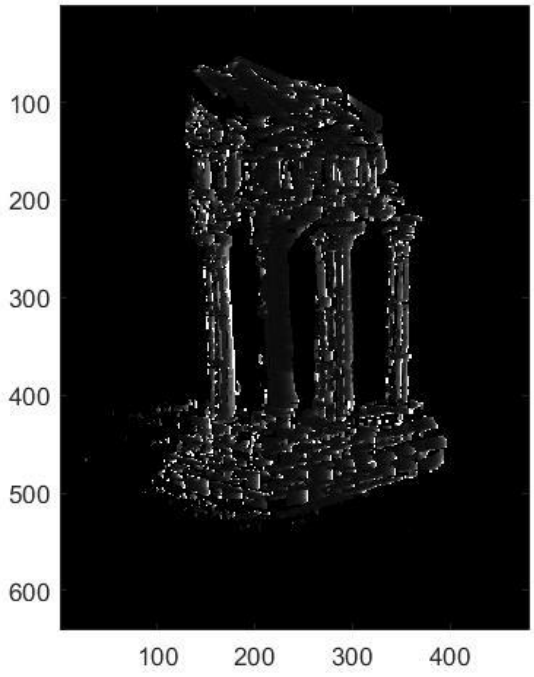
3.2.2 Dense window matching to find per pixel density

Image of disparity map:



3.2.3 Depth map

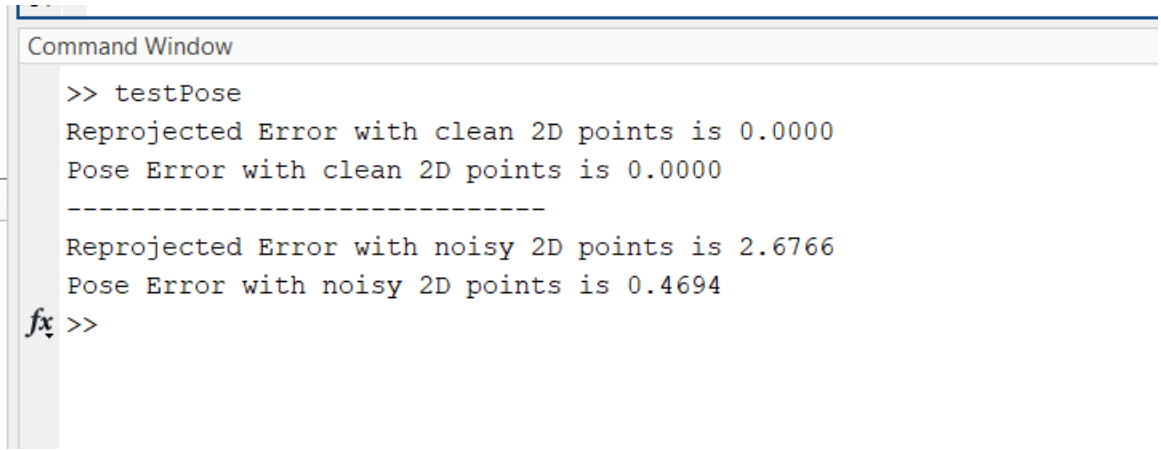
Image of depth map:



3.3 Pose estimation

3.3.1 Estimate camera matrix

Output of script testPose:



3.3.2 Estimate intrinsic/extrinsic parameters

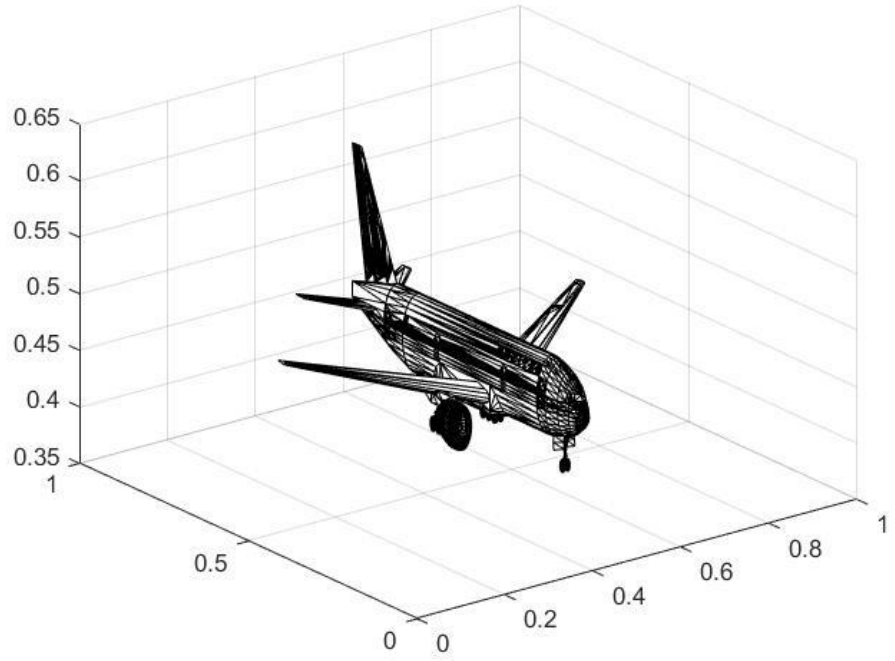
Output of script testKRt:

```
Command Window
>> 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 0.6040
Rotation Error with clean 2D points is 0.1280
Translation Error with clean 2D points is 0.0910
fx >>
```

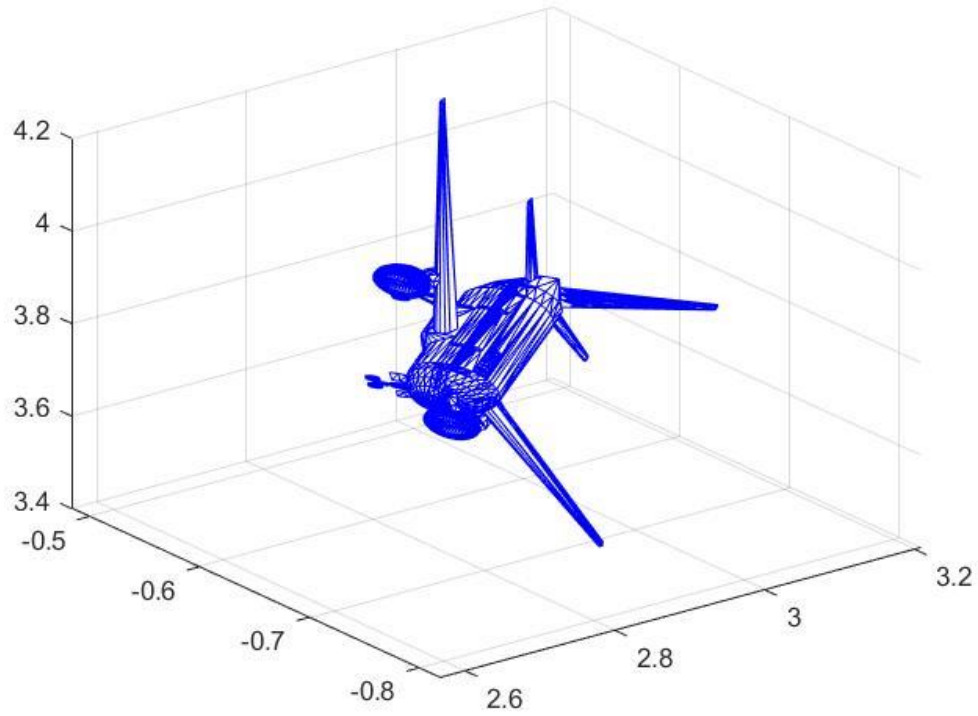
3.3.3 Project a CAD model to the image

Projection of 2d points (Red) and 3d points (green) on the image:

3D graph of vertices:



3D graph after rotation after estimated R:



CAD model overlapping with the image: