Computer Vision Homework 4

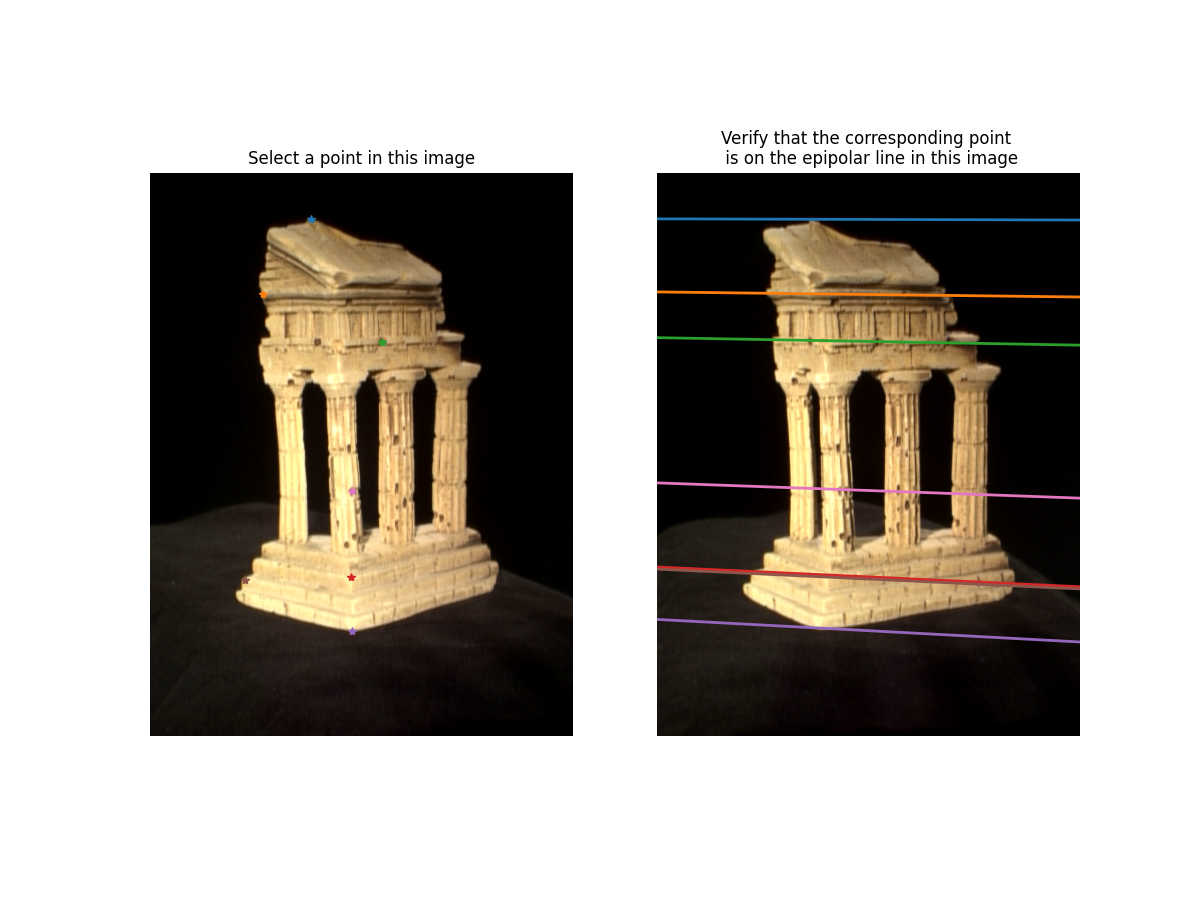
Oron Barazani 305295818 & Roee Hadar 206390098

Part 1:

1.1 Eight-points algorithm

We constructed the fundamental matrix F using the Eight-points algorithm. The algorithm gets a set of matching points of size (N,2) in 2 cameras worlds and find the F matrix

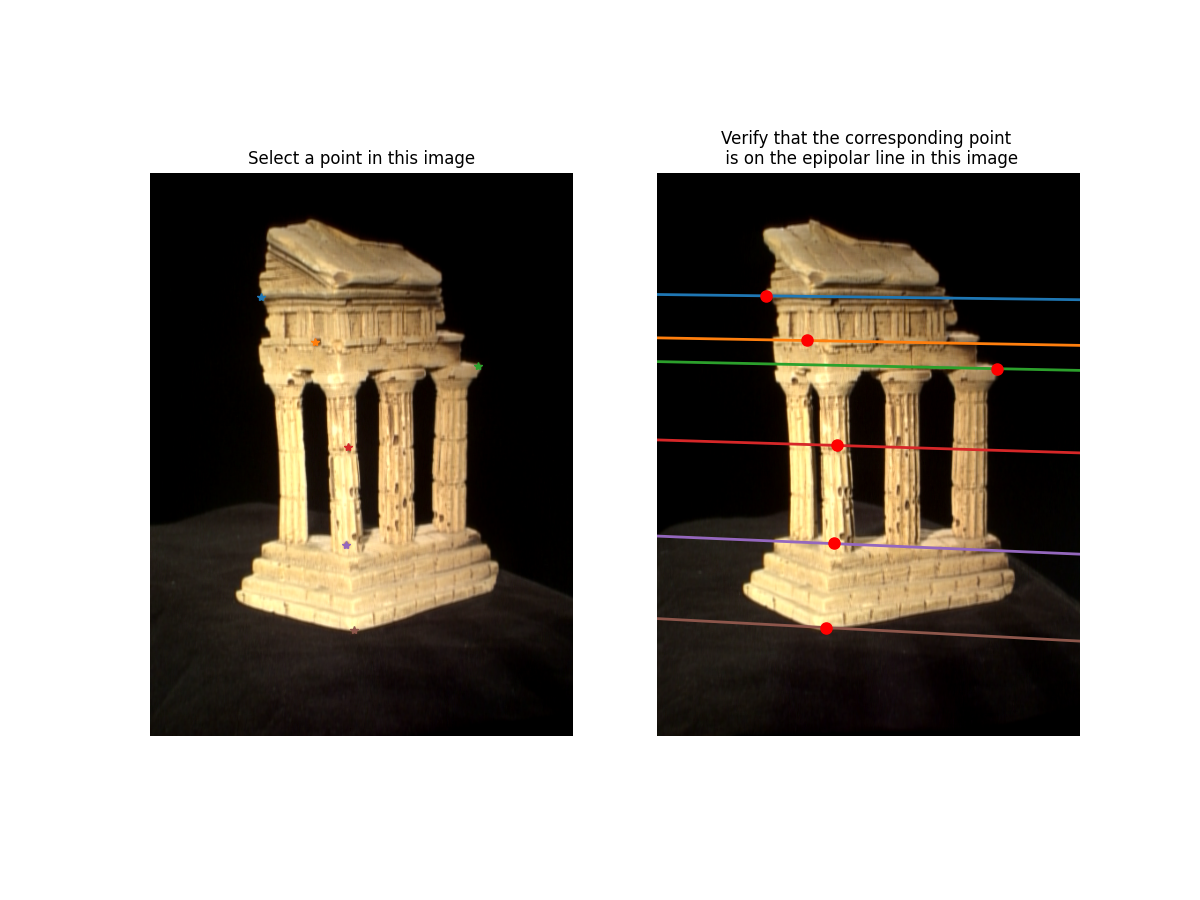
* Normalizing points
* Construct A matrix of shape (9,N) using points to find appropriate transformation (find parameters of the transformation)
* Solving the following problem:
* We used SVD to find the solution to this problem
* Since F matrix estimation is noisy, we enforced the rank 2 constraint and also use the supplied helper function to refine F
* The results:

And used the supplied GUI to view variety of epipolar lines in the second image corresponding to matching points in the first image:

1.2 Epipolar correspondence

We used F matrix that we have found in 1.1 and the supplied images to find the epipolar correspondence of each point in first image:

* For each point in the first image
* We find the epipolar line in the second image
* Along the epipolar line we look for the best matched point for
* The score determined by Euclidian distance between the 2 windows around each point – each window is in size
* The windows were normalized to improve results – more robust to illuminance difference

We used the supplied to plot the correspondence points in the second image:

1.3 Essential matrix

The Essential matrix relation to F:

So given we can calculate to essential matrix:

1.4 Triangulation

We used supplied points and and their respective matrix cameras and to construct matrix A to solve the following problem:

Where are 3d coordinates.

To do so, we need to first of all, find (assuming that is unit). Given E (and the supplied function) we can get 4 possible candidates for .

We can choose the right matrix by performing the following steps:

* Preform triangulation using supplied points and , matrix and the candidate matrix
* Project the 3D points back to each camera and find the point distance from the camera plane.
* We can assume that the point should have positive Z coordinate (distance of point P from the camera plane)
* Using the above assumption, we can narrow down to one matrix.

So, we can conclude our algorithm as follow:

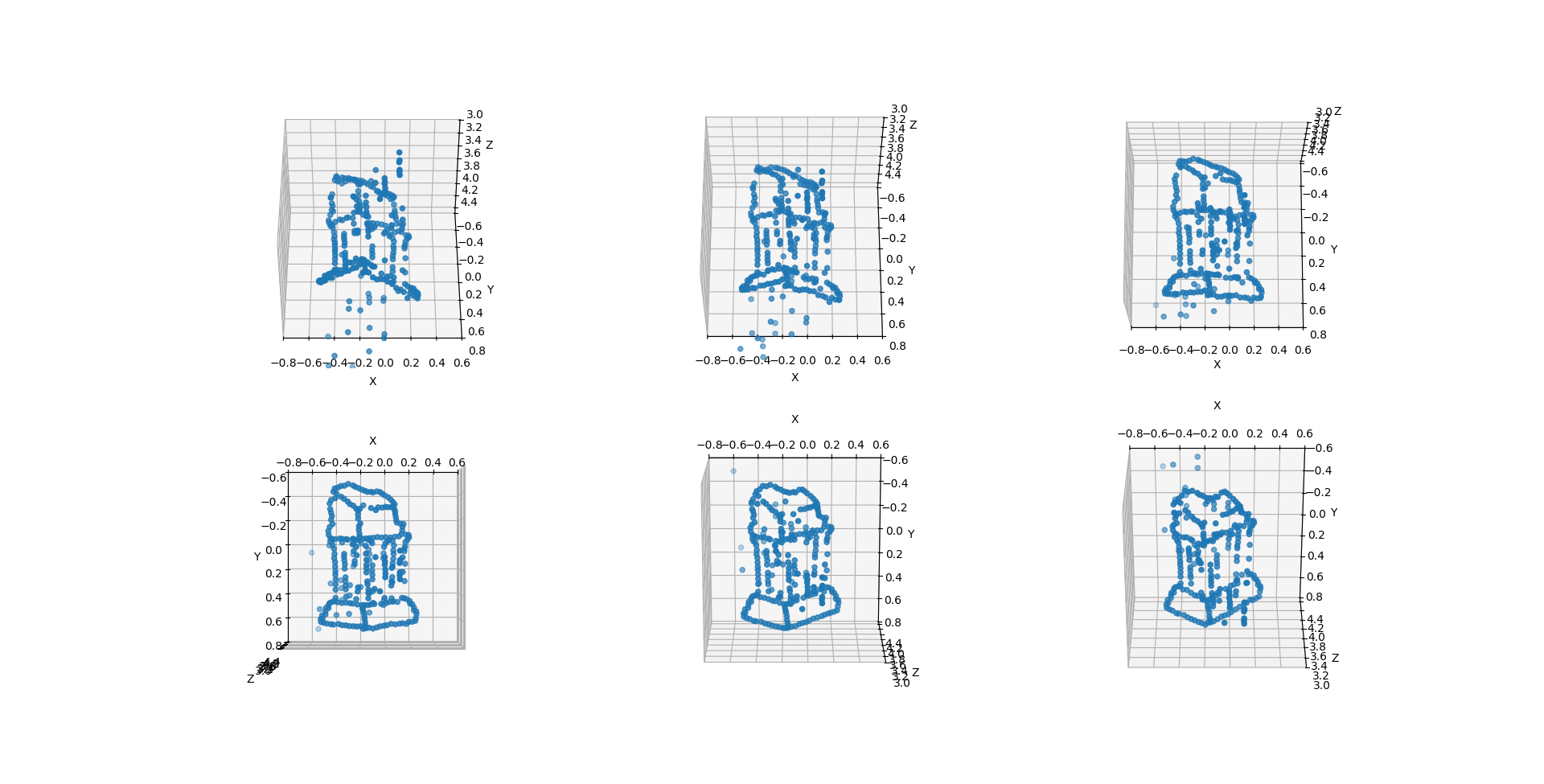
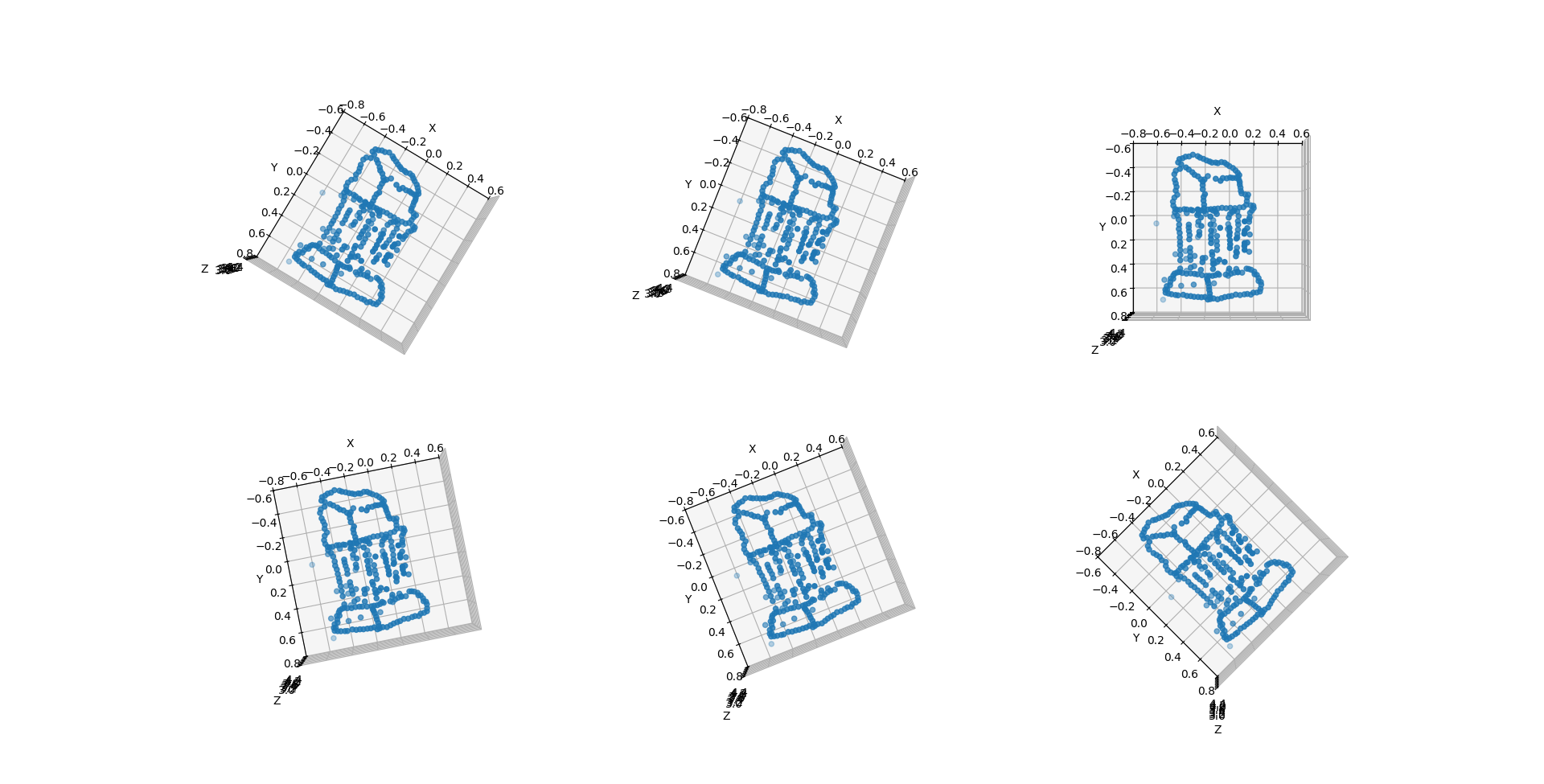
* Get candidates from matrix
* Find the best match using the method mentioned above
* Preform final triangulation using the chosen matrix

1.5 Putting it all together

We used all the steps above to find 3D points from 2 images and some points correspondence:

* Load the images the points
* Find F
* Use F to find epipolar correspondence points
* Load K matrices
* Find essential matrix E
* Get candidates from matrix
* Find the best match using the method mentioned above
* Preform final triangulation using the chosen matrix

The final 3D model:



We can see that most of the points are forming the temple shape, but there some outliers. The source of the error is in mismatch of the 2d points in the epipolar correspondences' algorithm. Some of the points are match to different region of the temple, leading to error in the 3D point reconstruction.