Q 1.1 Intersection of principal axes in point P

Coordinate origins coincide with the same point P (principal point), therefore:

$$\tilde{x}_2^T F \tilde{x}_1 = 0 \Rightarrow \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 0 \Rightarrow \begin{bmatrix} F_{31} & F_{32} & F_{33} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 0 \Rightarrow F_{33} = 0$$

Q 1.2 Epipolar lines in pure translation in x direction

If the cameras only differ by a pure translation $t = [t_x, 0, 0]^T$ parallel to x-axis, then for the camera matrices we have:

$M_1 = K[I \mid 0]$ and $M_2 = K[I \mid t]$

Then, for the fundamental matrix we will have:

$$F = [e]_\times KK^{-1} = [e]_\times$$

Due to the camera motion in the x direction, the epipole **e** is $[1, 0, 0]^T$ and the fundamental matrix becomes:

$$F = [e]_\times = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

Therefore, we will have:

$$l' = Fx_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ y_1 \end{bmatrix}$$

Which means that the epipolar lines have zero slope (first element is zero) and are parallel to x-axis.

Q 1.3 Reflection in a plane mirror

Suppose we have a mirror and a camera as shown in the Fig. 1. The camera sees the reflection of the point P in the mirror as P'.

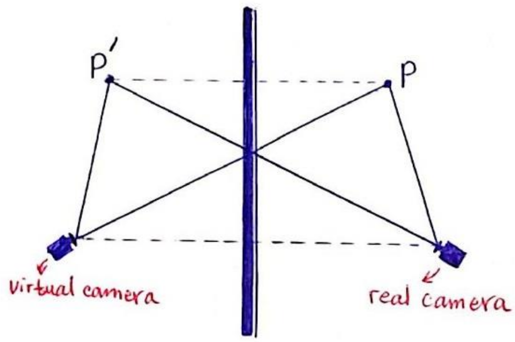

Figure 1. The camera looking at a point and its reflection in a planar mirror.

We can suppose that we have a virtual camera in the mirror (which captured a flipped version of the original image) as shown in the figure. Then we will have:

$$\tilde{P}' = T\tilde{P}$$

Where T is the transformation of the mirror that transforms point P to its image. Also, the position of the camera is transformed using the mirror transformation as :

$$M_1 = K[I\,|\,0] \Rightarrow M_2 = K[I\,|\,0]T = M_1T$$

Which can be incorporated in the $[I\,|\,0]$ matrix, but it is more useful in the above form. Obviously, the M_1 and M_2 matrices are the projective camera matrices of the cameras 1 (the real camera) and 2 (the virtual camera), respectively. I safely supposed that the pose of the real camera is $[I\,|\,0]$ for simplicity.

For the images of the points in the cameras 1 and 2 we will have (note that the point P is substituted as point P' in the flipped image and vice versa):

$$\begin{array}{lcl} \tilde{p}_1 = M_1\tilde{P} & & \tilde{p}_2F\tilde{p}_1{}' = 0 \\ \tilde{p}_1{}' = M_1\tilde{P}' & \text{and} & \tilde{p}_2{}'F\tilde{p}_1 = 0 \\ \tilde{p}_2 = M_2\tilde{P} & & \tilde{p}_1F^T\tilde{p}_2{}' = 0 \\ \tilde{p}_2{}' = M_2\tilde{P}' & & \tilde{p}_1{}'F^T\tilde{p}_2 = 0 \end{array}$$

Using the above equations, we have:

$$\tilde{p}_1{}'F^T\tilde{p}_2 + \tilde{p}_2F\tilde{p}_1{}' = 0 \Rightarrow M_1\tilde{P}'F^TM_2\tilde{P} + M_2\tilde{P}FM_2\tilde{P} = 0 \Rightarrow$$
$$M_1T\tilde{P}F^TM_1T\tilde{P} + M_1T\tilde{P}FM_1T\tilde{P} = 0 \Rightarrow \left(M_1T\tilde{P}\right)\left(F^T + F\right)\left(M_1T\tilde{P}\right) = 0$$

Since the matrix $M_1T\tilde{P}$ is not set to zero, we can conclude that $F^T = -F$. Therefore, the fundamental matrix is a skew symmetric matrix.

Q 2.1 Estimation of F using 8-Point Algorithm

The calculated fundamental matrix F using 8-point algorithm is:

```
-0.0000    0.0003   -6.3885
 0.0008   -0.0000   -0.0743
 6.1527   -0.0402   24.5403
```

The result of the 8-point algorithm visualized using the provided displayEpipolarF tool is shown in Fig. 2.

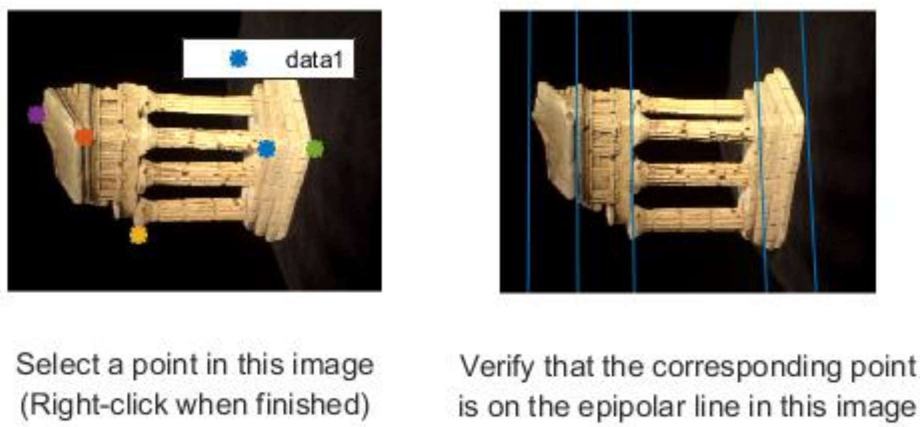

Figure 2. The result of using 8-point algorithm for estimating the fundamental matrix.

Q 2.2 Estimation of F using 7-Point Algorithm

There are 3 calculated fundamental matrices for each set of points. For the correspondences selected by me using the cpselect tool (points are shown in 'test_q2.m' file), the 3rd fundamental matrix (F{3}) gives the best results. This matrix is:

```
 0.0000    0.0000   -0.0008
-0.0000   -0.0000    0.0001
 0.0008   -0.0001   -0.0038
```

The result of the 7-point algorithm visualized using the provided displayEpipolarF tool on some points is shown in Fig. 3.

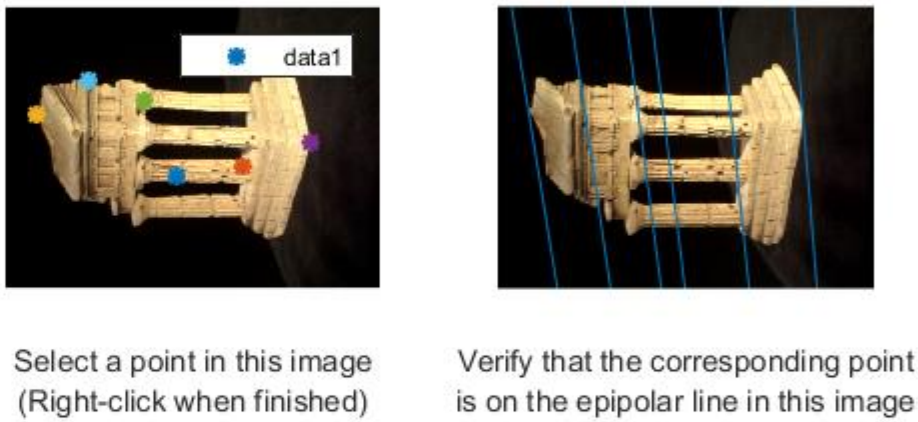

Figure 3. The result of using 7-point algorithm for estimating the fundamental matrix.

Q 2.X Estimation of F using RANSAC

The number of iterations k is calculated using the following equation discussed in the class:

$$(1-p)=(1-w^n)^k \Rightarrow k=\frac{\log(1-p)}{\log(1-w^n)}$$

Where n is the number of points used to estimate the model ($n = 7$ for 7-point algorithm and $n = 8$ for 8-point algorithm), p is the probability of finding a good set of points to estimate the model (0.999 in my code), and w is the fraction of inliers in the data (0.75 for the given noisy set of point pairs). This choice of parameters results in 49 iterations for 7-point algorithm and 66 iterations for 8-point algorithm. Just to be safe, I used 2 times the resulted number of iterations in the code.

As the error metric, I used $\left|\tilde{x}_2^T F \tilde{x}_1\right|$ and selected those with result less than a 10^{-3} threshold as inliers. The reason for this error measure is to minimize the distance of points on the second image from the estimated epipolar line of their corresponding point in the first image.

The estimated fundamental matrix F using RANSAC and 7-point algorithm is:

```
 0.0000    0.0000   -0.0008
-0.0000   -0.0000    0.0001
 0.0007   -0.0001    0.0021
```

The result of the 7-point algorithm with the RANSAC, visualized using the provided displayEpipolarF tool on some points is shown in Fig. 4.

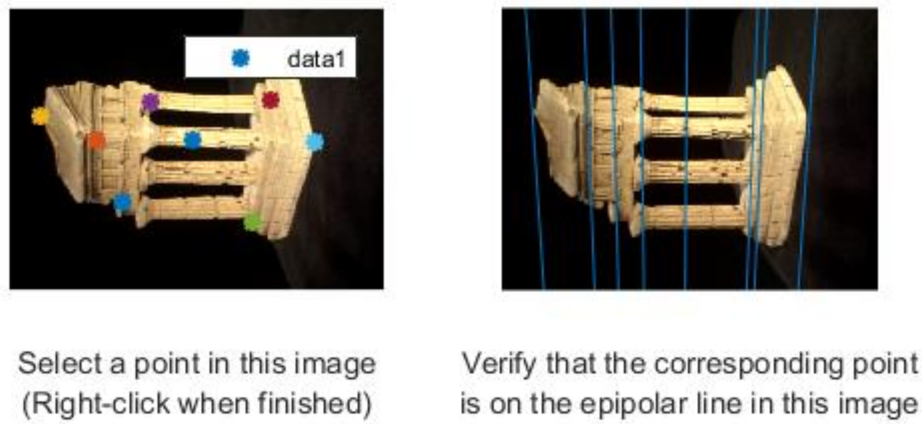

Figure 4. The result of using 7-point algorithm with RANSAC for estimating the fundamental matrix.

The estimated fundamental matrix F using RANSAC and 8-point algorithm is:

```
0.0000   -0.0000   -0.0011
0.0000    0.0000   -0.0008
0.0010    0.0009    0.0009
```

The result of the 8-point algorithm with the RANSAC, visualized using the provided displayEpipolarF tool on some points is shown in Fig. 5.

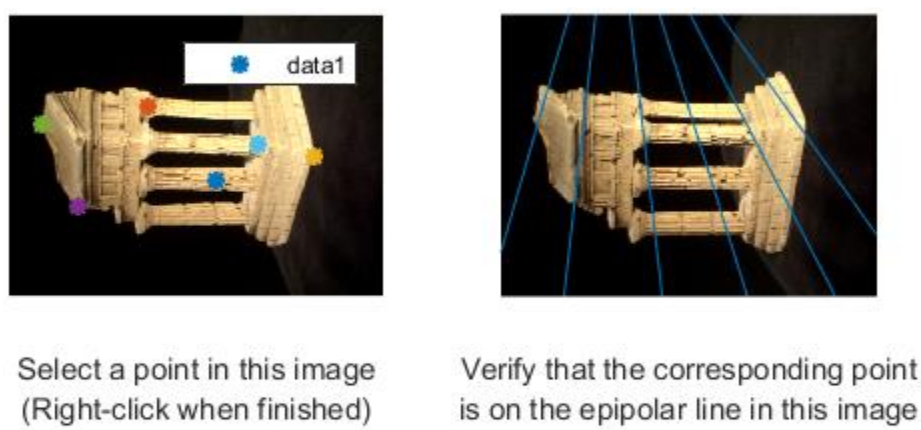

Figure 5. The result of using 8-point algorithm with RANSAC for estimating the fundamental matrix.

Q 2.3 Computing the essential matrix

The essential matrix computed from the fundamental matrix obtained from the 8-point algorithm (Q2.1) is:

```
1.0e+03 *
-0.0439    0.7193   -9.6055
 1.7882   -0.0030    0.2417
 9.6352    0.0811    0.0038
```

Q 2.4 Triangulation

The triangulation function is finding the 3D point using the homogeneous least-square triangulation method described in the section 12.2 of "Multiple View Geometry in Computer Vision (2nd Edition)" by Richard Hartley and Andrew Zisserman.

We have that $\tilde{p} = M\tilde{P}$ where $\tilde{P}$ is the homogeneous 3D point, $\tilde{p}$ is the homogeneous image point and M is the projective 3×4 camera matrix. Therefore:

$$\tilde{p} = M\tilde{P} \Rightarrow \tilde{p} \times M\tilde{P} = 0 \Rightarrow \begin{pmatrix} p_x M_3 - M_1 \\ p_y M_3 - M_2 \end{pmatrix} \tilde{P} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Since we have two such points (one on each image) with the same 3D point, we will have 4 total rows and can compute the 3D point.

The reprojection error is also computed and returned by the function.

Q 2.5 Finding the correct M2 matrix

Out of 4 M2s, only one of them produced all the 3D points in positive Z. The matrix is as follows:

```
1.0e+03 *

 1.5214   -0.0270    0.2959    0.0179
-0.0481    1.4115    0.6282   -1.4801
 0.0000   -0.0003    0.0010    0.0002
```

Q 2.6 Epipolar correspondence

After a few trials, I decided on using a window size of 11 (5 pixels on each side), weighted with Gaussian function with sigma of 2.5 (which do not ignore any of the pixels in the window, while giving reasonably small weights to the corners). Also, the maximum Euclidean distance for the point coordinated in the two images is set to 50. Finally, I used the image values to compare the patches in the two images.

A screenshot of epipolarMatchGUI with some detected correspondences is shown in Fig. 6.

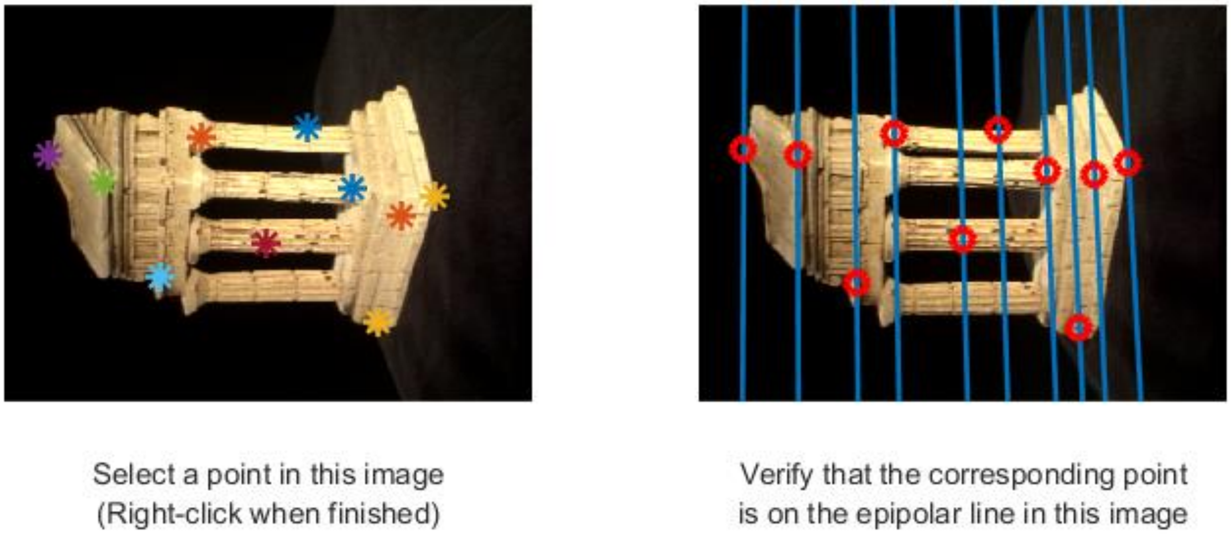

Figure 6. A screenshot of epipolarMatchGUI with some detected correspondences.

Q 2.7 3D Visualization

The screenshots for the point cloud of the temple are shown in Fig. 7.

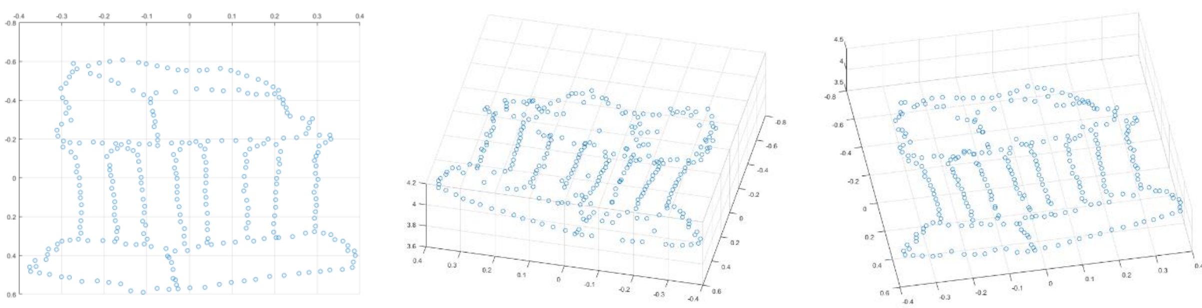

Figure 7. A few screenshots for the point cloud of the temple.

Q 2.8 Awesome visualization

I used a third-party open-source code called Fiji (http://fiji.sc) which is able to provide several types of 3D reconstruction. Fig. 7 shows the result of the dense reconstruction and visualization using meshes.

Figure 8. Result of the dense reconstruction and visualization by meshes using Fiji software.