16-720: Assignment 4
Abhinav Maurya
amaurya@andrew.cmu.edu

1.1:

Since both images are normalized, $x1 = [0, 0, 1]$ and $x2 = [0, 0, 1]$. By definition of fundamental matrix, $x2^T * F * x1 = 0$. By SVD, $F = U * \Sigma * V^T$.
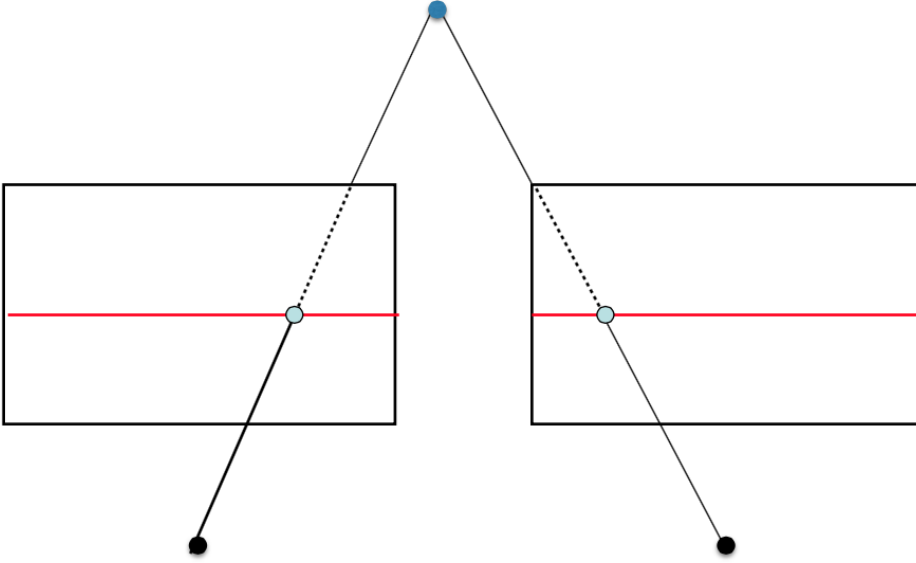
$$\therefore x2^T * U * \Sigma * V^T * x1 = 0$$

Considering $U = [u1; u2; u3]$ and $V = [v1; v2; v3]$, we can substitute $x1$, $x2$, U, and V in the equation to get

$$u3 * sigma3 * v3 = 0$$

If $sigma3 \neq 0$, then the relation does not hold. Hence, the smallest singular value of fundamental matrix has to be zero.

1.2:

The case of original camera C1 and camera translated along X-axis C2 looks like figure above.

Let's consider camera intrinsic matrix to be identity since we are investigating variations in camera extrinsics (trnaslation). Let's consider the camera extrinsic matrix for C1: $[I|0]$.

$$\therefore \lambda * [x; y; 1] = [I|0] * [X; Y; Z; 1]$$

For camera $C2$, the camera extrinsic matrix is $[I|t_x; 0; 0]$ i.e. t_x translation along X-axis.
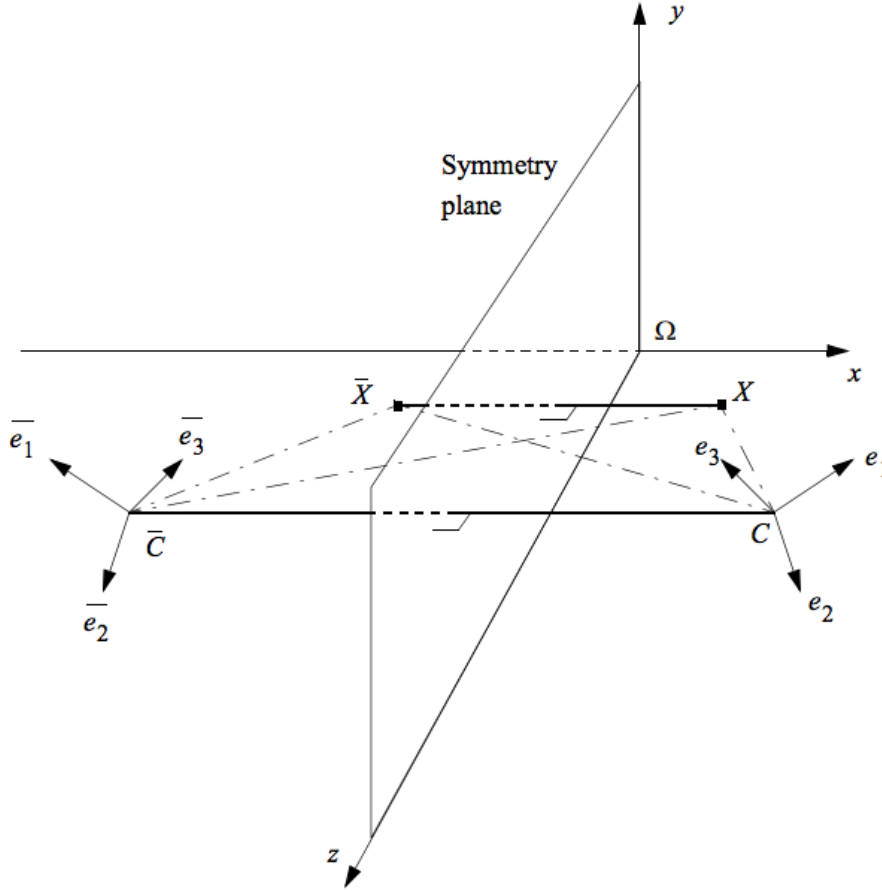
$$\therefore \lambda * [x; y; 1] = [I|t_x; 0; 0] * [X; Y; Z; 1]$$

1

Thus, we see that the point projected on the image surfaces of cameras C1 and C2 differ only in their x-coordinate. As we vary the 3D homogeneous point $[X; Y; Z; 1]$ in the epipolar plane, we trace out the epipolar line parallel to X-axis.

1.3:

From the figure above which appeared in `http://iris.usc.edu/outlines/papers/2003/alex-ivc2003.pdf`, we see that we can place the camera along a symmetry plane from which all corresponding points of the object and its mirror image are equidistant. If this is not the case, the same rotation and translation transformation can be applied to both points and our result will still hold. Also, the camera intrinsic matrix is assumed to be identity. If it is not, its effect is identical on projections of both the object and its mirror image.

Consider point P on the object and its corresponding point Q in the mirror image. In the plate of the camera positioned on the symmetry plane, these points are captured at positions $p = [p_x; p_y; 1]$ and $q = [q_x; q_y; 1]$. We know that $\lambda * p = P$ and $\lambda * q = Q$. Moreover, since P and Q are on either side of the symmetry plane, they differ only in one coordinate whose axis is perpendicular to the symmetry plane, say X coordinate.

$$\therefore P = [-1, 0, 0; 0, 1, 0; 0, 0, 1] * Q$$

Let $T = [-1, 0, 0; 0, 1, 0; 0, 0, 1]$

$$\therefore P = T * Q$$

2

.

$$\lambda * p = \lambda * T * Q$$

. Taking cross-product with p, we get

$$[0, -1, p_y; 1, 0, -p_x; -p_y, p_x, 0] * [-q_x; q_y; 1] = 0$$

Taking dot-product with $[1, 1, 1]$ to sum over, we get

$$[1, 1, 1] * [0, -1, p_y; 1, 0, -p_x; -p_y, p_x, 0] * [-q_x; q_y; 1] = 0$$

This can be reworked to

$$[p_x, p_y, 1] * [0, 1, -1; -1, 0, 1; 1, -1, 0] * [-q_x; q_y; 1]$$

If $F = [0, 1, -1; -1, 0, 1; 1, -1, 0]$,
$$p = F * q'$$

where $q' = [-q_x; q_y; 1]$

Thus, points p and q are related through a skew-symmetric fundamental matrix. The $-q_x$ in $[-q_x; q_y; 1]$ indicates that the mirror image is captured by the camera as a mirror image i.e. a flipped version of the image of the original object.
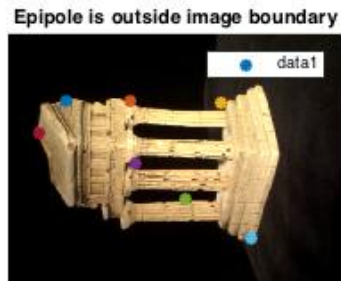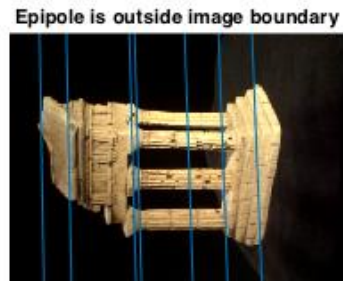

2.1: eightpoint

Included in `code/eightpoint.m` file. No refinement was used since it seemed to be causing instability in estimating F.

```
F =

    -0.0000    -0.0000     0.0011
    -0.0000     0.0000    -0.0000
    -0.0011     0.0000    -0.0042
```

Epipole is outside image boundary

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

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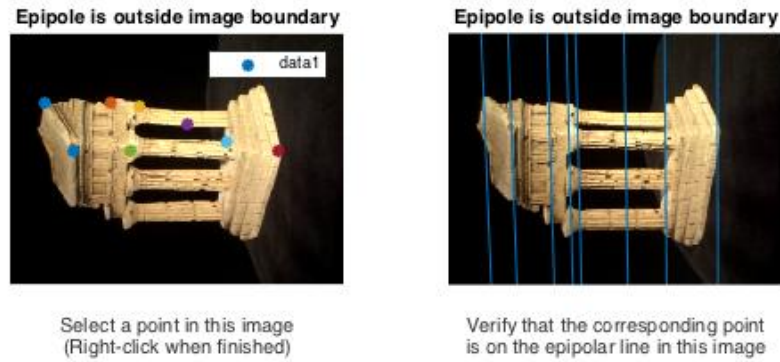is on the epipolar line in this image

2.2: sevenpoint

Included in `code/sevenpoint.m` file.

```
F =

    0.0000   -0.0000    0.0011
    0.0000    0.0000   -0.0001
   -0.0011    0.0001   -0.0046
```

2.3: essentialMatrix

Included in `code/essentialMatrix.m` file.

```
E =

   -0.0030   -0.3035    1.6603
   -0.1374    0.0083   -0.0512
   -1.6651   -0.0125   -0.0013
```

2.4: triangulate

Included in `code/triangulate.m` file.

2.5: findM2

Included in `code/findM2.m` file.

2.6: epipolarCorrespondence

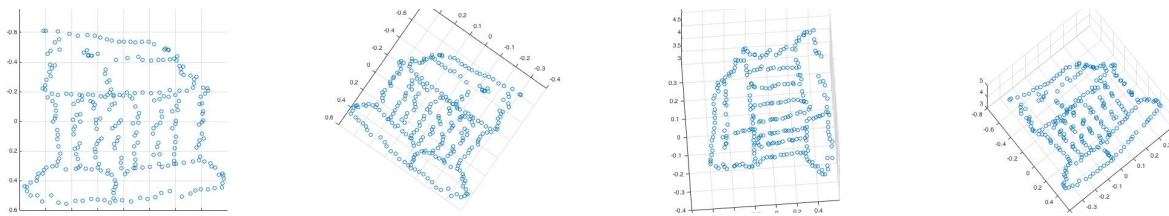Included in `code/epipolarCorrespondence.m` file. Result in figure 1.

2.7: q2_7

Included in `code/q2_7.m` file.

Figure 1: Output of 3D reconstruction