HW4 Write-Up

16-720: Computer Vision

Spring 2016

Cole Gulino

March 24, 2016

I. Theory

Q1.1: (5 points) Suppose two cameras fixate on a point P (see figure 1) in space such that their principal axes intersect at that point. Show that if the image coordinates are normalized so that the coordinate origin (0, 0) coincides with the principal point, the F_{33} element of the fundamental matrix is zero.

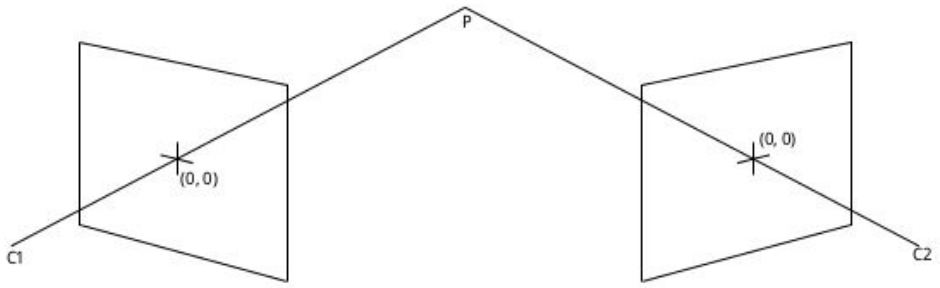

Figure 1: Figure for Q1.1. $C1$ and $C2$ are the optical centers. The principal axes intersect at point P.

The fundamental matrix is defined as:

$$F = K_2^{-T} E K_1^{-1}$$

Where the relation for K_1 and K_2 are given by:

$$\lambda_1 \mathbf{x}_1 = K_1 \mathbf{X}$$
$$\lambda_2 \mathbf{x}_2 = K_2 \mathbf{X}$$

K_1 and K_2 are the intrinsic camera matrix for C_1 and C_2.

For the fundamental matrix, we also have this property:

$$\mathbf{x}_2^T F \mathbf{x}_1 = 0$$

We can expand the matrix form of this to get a better understanding:

$$\mathbf{x}_2^T F \mathbf{x}_1 = \begin{bmatrix} x_2 & y_2 & 1 \end{bmatrix} \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

If we assume that the coordinate origin (0, 0) coincides with the principal point as stated in the question:

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

Carrying out the matrix multiplication:

$$\mathbf{x}_2^T F \mathbf{x}_1 = \begin{bmatrix} F_{31} & F_{32} & F_{33} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 0$$

$$\mathbf{x}_2^T F \mathbf{x}_1 = F_{31}(0) + F_{32}(0) + F_{33}(1) = 0$$

So this shows that in order for this to be true:

$$F_{33} = 0$$

Q1.2:(10 points) Consider the case of two cameras viewing an object such that the second camera differs from the first by a pure translation that is parallel to the x-axis. Show that the epipolar lines in the two cameras are also parallel to the x-axis.

The relation of the points in space for the two cameras can be shown as:

$$\lambda_1 \mathbf{x}_1 = K_1 \mathbf{X}$$
$$\lambda_2 \mathbf{x}_2 = K_2 \mathbf{X}$$

If we assume that C_2 undergoes a translation x'. Then we can assume that:

$$K_1 = K_2$$

We know that the epipolar lines have an equation governed by this relation:

$$\mathbf{l}_1 = F \mathbf{x}_2$$
$$\mathbf{l}_2 = F \mathbf{x}_1$$

The fundamental matrix can be expressed now as:

$$F = K^{-T} E K^{-1}$$
$$F = K^{-T} \hat{T} R K^{-1}$$

Because the K matrices are the same and the rotation matrix is unity, the fundamental matrix between two cameras can be expressed as:

$$F = \hat{T} K R K^{-1}$$

Because the rotation matrix is unity and the K matrices are equal:

$$F = \hat{T}$$

The epipole for the second camera is the right nullspace of $\hat{T}$ and thus:

$$\mathbf{e}_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

The fundamental matrix is then a skew symmetric matrix which is based on the epipole parameters:

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = E \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

$a, b, \text{ and } c$ are the coefficients of the epipolar line $\mathbf{l}_2$. The fundamental matrix can then be specified as:

$$F = \begin{bmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{bmatrix}$$

When the camera translation is parallel to the x-axis as is specified in the problem, the epipolar line has the direction:

Then the fundamental matrix becomes:

$$F = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

We can then look at the equation that the fundamental matrix must satisfy:

$$\mathbf{x}_2^T F \mathbf{x}_1 = 0$$

$$\mathbf{x}_2^T F \mathbf{x}_1 = \begin{bmatrix} x_2 & y_2 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = 0$$

This shows that the equation becomes:

$$y_2 = y_1$$

This shows that the epipolar lines are corresponding rasters.

Let us look at the equation for converting the two-dimensional camera coordinates into the three-dimensional space:

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = K \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

There is no rotation and we assume that there is no translation for camera C_1:

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = K \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = zK^{-1} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

Where z is the depth of $\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$ in the camera plane.

Now we can look at the second camera:

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

The intrinsics are the same and the extrinsics are just a translation in the x direction as specified in the problem. And thus:

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

We can replace the projected point with what we've obtained from the the previous equation:

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} zK^{-1} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

Whenever we expand this, we get:

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} + \frac{K}{z} \begin{bmatrix} t_x \\ 0 \\ 0 \end{bmatrix}$$

This shows that our points in the second camera are a translation in the x-axis which is dependent on the camera intrinsics and the depth of the object from the camera. And thus the epipolar lines are shown to be parallel to the x-axis.

Because this translation is inversely proportional to the depth of the object from the camera, objects closer to the camera appear to move faster than those further away from the camera.

Some of this work was found from reading this chapter from this chapter [1].

Q1.3: (10 points) Suppose that a camera views an object and its reflection in a plane mirror. Show that this situation is equivalent to having two images of the object which are related by a skew-symmetric fundamental matrix. (Hint: draw the relevant vectors)

The matrix of a reflection for a plane with normal n passing through the origin is $I - 2nn^T$ [2].

The drawing of the lines is as shown in Figure 2 [3]:

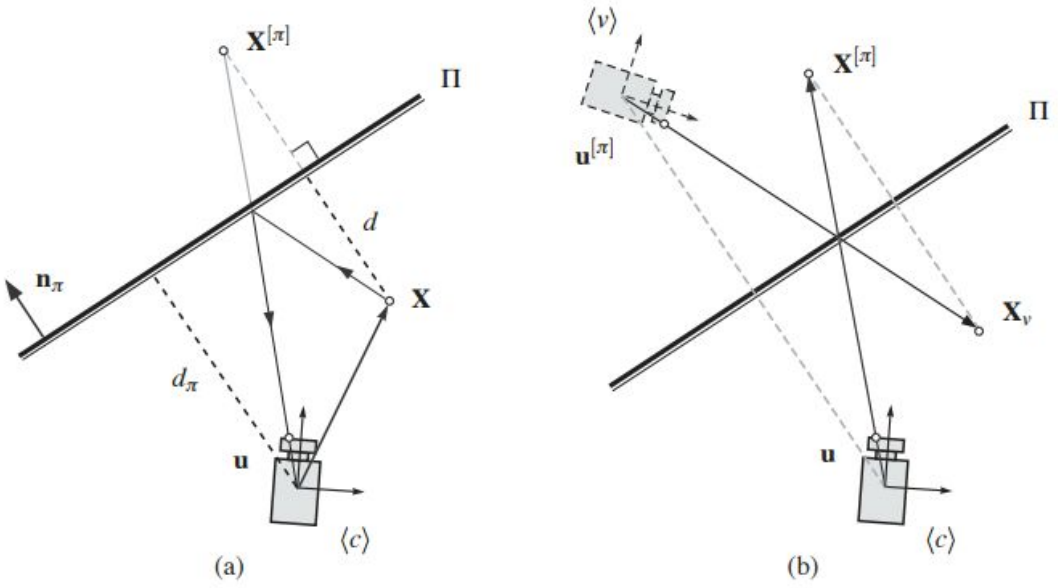

Figure 2) Camera with a Plane Mirror Setup

Let the distance between the camera and the mirror be d_m .

Now we can specify the projection of the image reflected from the mirror with a rotation, translation, and reflective matrix [3]:

$$\lambda \begin{bmatrix} x_m \\ y_m \\ 1 \end{bmatrix} = K \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{I} - 2\mathbf{n}\mathbf{n}^T & 2d\mathbf{n} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

If we realize that a mirrored reflection specifies a “virtual point” in space which is part of the reflection:

$$\mathbf{X}_v = \begin{bmatrix} X_v \\ Y_v \\ Z_v \\ 1 \end{bmatrix}$$

Now we can specify this with the distance between the actual object and the virtual point as d.

So then:

$$\mathbf{X}_v = \mathbf{X} + 2d\mathbf{n}$$

The distance $d = d_m + \mathbf{X}^T\mathbf{n}$. So then we can show that:

$$\mathbf{X}_v = (\mathbf{I} - 2\mathbf{n}\mathbf{n}^T)\mathbf{X} + 2d_m\mathbf{n}$$

From this we can see that:

$$\mathbf{X}_v = \begin{bmatrix} \mathbf{I} - 2\mathbf{n}\mathbf{n}^T & 2d\mathbf{n} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

So then:

$$\lambda \begin{bmatrix} x_v \\ y_v \\ 1 \end{bmatrix} = \lambda \left(\mathbf{I} - 2\mathbf{n}\mathbf{n}^T\right) \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} + 2d_m\mathbf{n}$$

Doing some manipulation we find:

$$\begin{bmatrix} x_v & y_v & 1 \end{bmatrix} \left(2d_m \hat{n} \left(\mathbf{I} - 2\mathbf{n}\mathbf{n}^T \right) \right) \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

This shows that the Essential matrix is given by:

$$E = 2d_m \hat{\mathbf{n}}$$

Because the Essential matrix depends on two constants and a skew-symmetric matrix $\hat{\mathbf{n}}$ based on the mirror normal, the Essential matrix is skew-symmetric.

From this, we can get the Fundamental matrix as :

$$F = K_2^{-T} E K_1^{-1}$$
$$F = K_2^{-T} 2d_m \hat{\mathbf{n}} K_1^{-1}$$

If we assume that the camera is calibrated, we can assume that $K_1 = K_2 = \mathbf{I}$.

So then the Fundamental matrix depends on two constants, two identity matrices and a skew-symmetric matrix $\hat{\mathbf{n}}$ based on the mirror normal. This proves that the **Fundamental matrix of an object and its reflection in a plane mirror is skew-symmetric**.

Content and work sourced from [3].

II. Practice

2. Fundamental matrix estimation

The Eight Point Algorithm: Q2.1 (10 points) Submit a function with the following signature for this portion of the assignment:

```
Function F = eightpoint(pts1, pts2, M)
```

For `M` I took the the maximum of the number of rows and columns. For me this was `640`. My T matrix took the form:

$$T = \begin{bmatrix} 2/M & 0 & -1 \\ 0 & 2/M & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

I did this to solve the equation:

$$\begin{bmatrix} x_1x_1' & x_1y_1' & x_1 & y_1x_1' & y_1y_1' & y_1 & x_1' & y_1' & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_mx_m' & x_my_m' & x_m & y_mx_m' & y_my_m' & y_m & x_m' & y_m' & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0$$

$$A\mathbf{F} = 0$$

This shows that the fundamental matrix is a right singular vector of the A matrix. So I calculated the SVD of A:

$$A = USV$$

And then the fundamental matrix was the rightmost column vector of V. I then reshaped F and took the SVD in order to ensure that it has a rank of 2 by setting the smallest eigenvalue to 0 and then reconstructing F.

I finally rescaled F using:

$$F_{unscaled} = T^T FT$$

My F matrix took on the form:

$$F = \begin{bmatrix} -1.5582e-8 & 2.5130e-7 & -0.0017 \\ 2.7177e-8 & -6.7474e-9 & 3.1037e-5 \\ 0.0016 & -1.4303e-4 & 0.0586 \end{bmatrix}$$

To make this matrix, I used the point correspondences found in the provided `some_corresp.mat`.

Figure 3 shows an example of my output using the eight point algorithm:

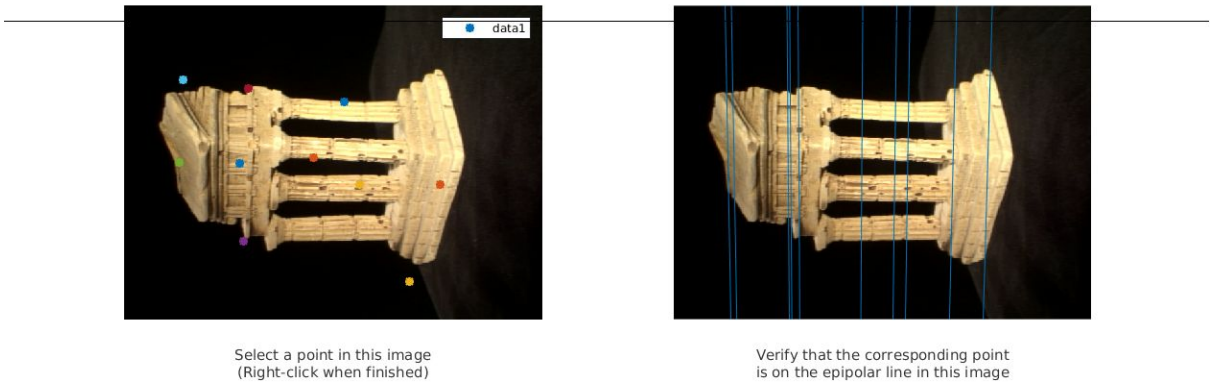

Figure 3) Example Output for the Eight Point Algorithm

The Eight Point Algorithm: Q2.2 (10 points) Submit a function with the following signature:

```
Function F = sevenpoint(pts1, pts2, M)
```

The seven point algorithm looks to solve for F using the same equation as the eight point algorithm shown below:

$$\begin{bmatrix} x_1x'_1 & x_1y'_1 & x_1 & y_1x'_1 & y_1y'_1 & y_1 & x'_1 & y'_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_mx'_m & x_my'_m & x_m & y_mx'_m & y_my'_m & y_m & x'_m & y'_m & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0$$

$$A\mathbf{F} = 0$$

In this case only 7 point correspondences are needed to solve this equation.

The first part of the algorithm is the same. You take the equation of the matrix A:

$$A = USV$$

In the seven point algorithm we take the two vectors that span the nullspace of A. This corresponds to the last two column vectors of the matrix V: F_1 and F_2.

The true fundamental matrix then corresponds to the equation:

$$F = \alpha F_1 + (1 - \alpha) F_2$$

We can find the relation α by solving the equation:

$$\det(\alpha F_1 + (1 - \alpha) F_2) = 0$$

To do this, I solved this equation using the matlab symbolic function:

```
syms alpha
alphas = vpa(solve(det(alpha * F1 + (1-alpha) * F2)));
```

I then found the solution that was real and calculated the true F using the equation above with the new alpha.

My F matrix took on the form:

$$F = \begin{bmatrix} 1.0505e-9 & -1.8922e-7 & 0.0027 \\ -2.6033e-7 & 3.1013e-9 & 8.7691e-6 \\ -0.0026 & 3.7454e-5 & -0.0107 \end{bmatrix}$$

Figure 4 shows an example output:

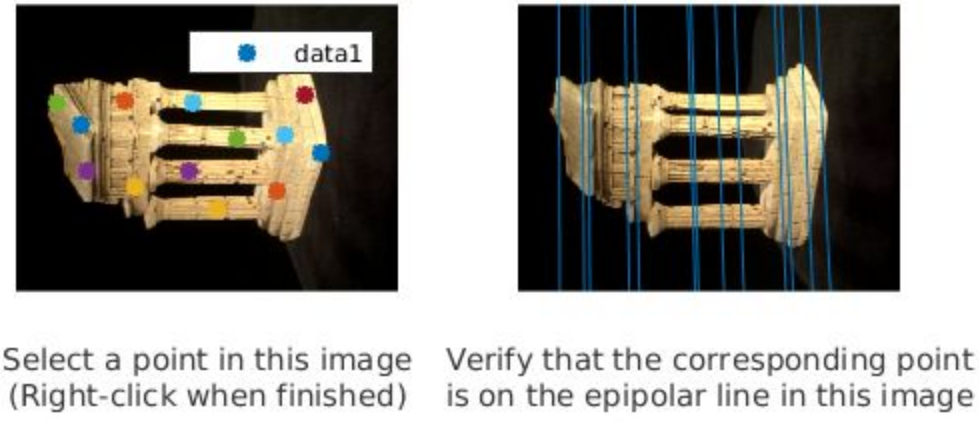

Figure 4) Example Output for the Seven Point Algorithm

Extra Credit: Automatic Computation of F: Q2.X Implement an algorithm with the signature:

```
Function [F] = ransacF(pts1, pts2, M)
```

The RANSAC algorithm I used was very similar to the `ransacH` algorithm that we wrote for HW2. The error metric that I used was:

$$\text{error} = (\mathbf{x}_2^T F \mathbf{x}_1)^2 + (\mathbf{x}_1^T F^T \mathbf{x}_2)^2$$

This is the combined reprojection error for both.

The F is calculated by using the seven point algorithm by randomly selecting 7 random points from the point correspondences. I gather a score for each of the F matrices returned by the seven point algorithm and

The algorithm then checks all of the other point correspondence guesses and calculates the error on them.

If `error < tol`, then the point is considered an inlier. The tolerance I set was `0.00001`.

My F matrix for this took the form:

$$F = \begin{bmatrix} -3.5602e4 & -3.8229e5 & 9.0479e8 \\ 2.1091e5 & 8.4983e3 & -6.794e7 \\ -8.4454e8 & 8.5026e7 & -7.2067e9 \end{bmatrix}$$

An example of the output is given in Figure 5:

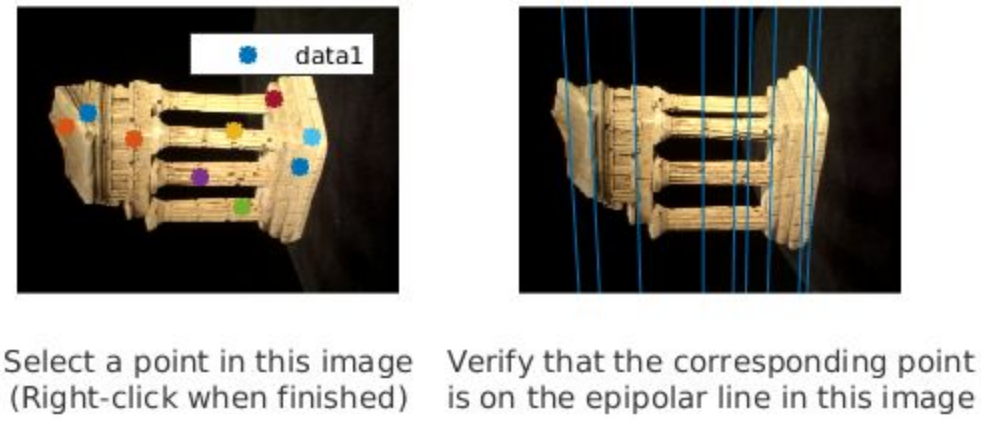

Figure 5) Example Output Using RANSAC

Running the eight point algorithm on the noisy values had similar results with the RANSAC version seeming to have slightly better results.

3. Metric Reconstruction

Q2.3: (5 points) Write a function to compute the essential matrix E using F and K_1 and K_2 with the signature:

```
Function M2s = essentialMatrix(F, K1, K2)
```

This function takes in the two K values and returns the M matrix for the second camera.

I first computed the essential matrix using the equation:

$$E = K_1^T F K_2$$

I then used the function `M2s = camera2(E)` to get the M2s that correspond to the four views.

The Fundamental and Essential Matrix Gathered:

$$F = \begin{bmatrix} -1.5582e-8 & 2.5130e-7 & -0.0017 \\ 2.7177e-8 & -6.7474e-9 & 3.1037e-5 \\ 0.0016 & -1.4303e-4 & 0.0586 \end{bmatrix}$$

$$E = \begin{bmatrix} -0.0360 & 0.5830 & -2.5549 \\ 0.0631 & -0.0157 & 0.0574 \\ 2.4398 & -0.1049 & 0.0091 \end{bmatrix}$$

The M2s that I found using the F from Q2.1 after normalization are shown in Figure 5:

```
val(:,:,1) =

   1.0e+03 *

    1.5219    0.0205    0.2938    0.0436
    0.0589    1.5365   -0.1585    1.5710
    0.0000    0.0003    0.0010    0.0002


val(:,:,2) =

   1.0e+03 *

    1.5219    0.0205    0.2938   -0.0436
    0.0589    1.5365   -0.1585   -1.5710
    0.0000    0.0003    0.0010   -0.0002


val(:,:,3) =

   1.0e+03 *

   -1.5193    0.0649   -0.3009    0.0436
    0.0358    1.5424   -0.0957    1.5710
    0.0000    0.0001   -0.0010    0.0002


val(:,:,4) =

   1.0e+03 *

   -1.5193    0.0649   -0.3009   -0.0436
    0.0358    1.5424   -0.0957   -1.5710
    0.0000    0.0001   -0.0010   -0.0002
```

Figure 5) Shows the M Matrix Options for Camera 2

Q2.4: (10 points) Using the above, write a function to triangulate a set of 2D coordinates in an image to a set of 3D points with the signature:

```
Function P = triangulate(M1, p1, M2, p2)
```

For this question, I utilized the resource found at Chapter 7.1 of [4].

I also used the resource found in the slides of [5].

The equation for a projection of points is:

$$\mathbf{x}_1 = M_1 X$$
$$\mathbf{x}_2 = M_2 X$$

Where:

$$M = K \left[R|t\right]$$

From these we can get the equation:

$$\mathbf{x}_1 = M_1 M_2^{-1} \mathbf{x}_2$$

I then expressed this equation in the form:

$$A\mathbf{X} = 0$$

Where:

$$A = \begin{bmatrix} x_1 * M_1(3,:) - M_1(1,:) \\ y_1 * M_1(3,:) - M_1(2,:) \\ x_2 * M_2(3,:) - M_2(1,:) \\ y_2 * M_2(3,:) - M_2(2,:) \end{bmatrix}$$

And:

$$M_1 = \begin{bmatrix} M_1(1,:) \\ M_1(2,:) \\ M_1(3,:) \end{bmatrix}$$

$$M_2 = \begin{bmatrix} M_2(1,:) \\ M_2(2,:) \\ M_2(3,:) \end{bmatrix}$$

Figure 6 shows an example of the reprojection using the 110 points in `some_correspondences.mat`.

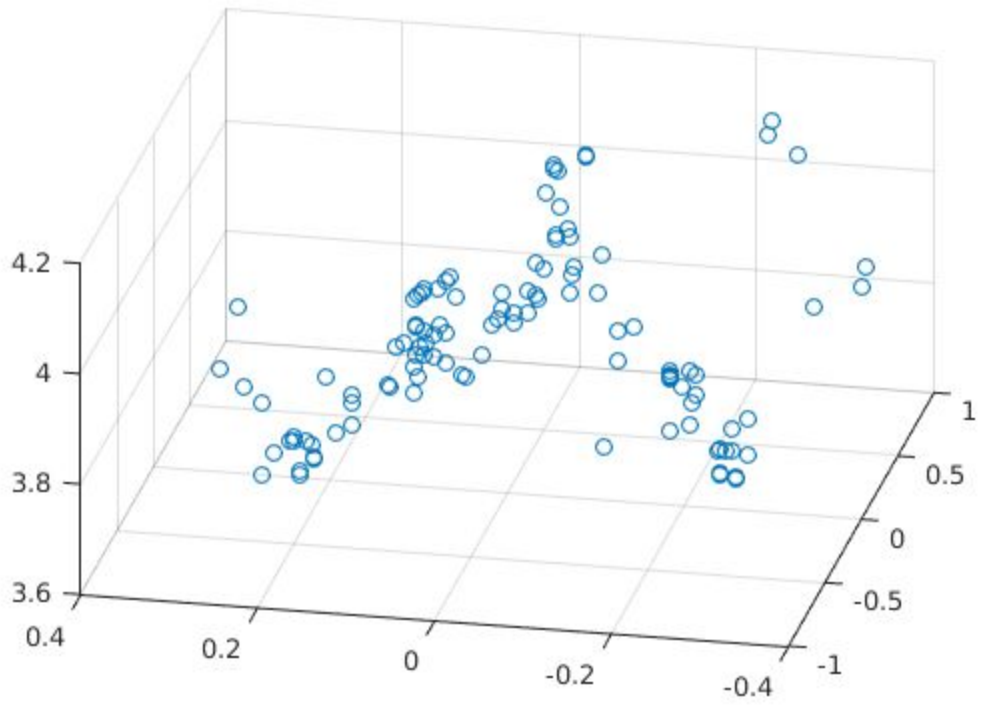

Figure 6) 3D Reconstruction Using 110 Sparse Points

Q2.5: (10 points) Write a script `findM2.m` to obtain the correct M2s by testing them.

In order to find the correct M2, I got the P from the M2s I get from triangulate and determine the points from triangulate that generate a positive value for every Z, which means that every point is projected in front of the camera, which is what we want.

I saved my values in `q2_5.mat`.

4. 3D Visualization

Q2.6: Implement a function with the following signature:

```
function [x2, y2] = epipolarCorrespondance(im1, im2, F, x1, x2)
```

To get the epipolar correspondences, I first find the epipolar line through the point by finding:

$$\mathbf{l} = F \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

I then use this line to interpolate x and y values on the line around a 5x5 box around the image. I only move along the line within 5 pixels of the point on the first image, because I assume that the image has not moved too far.

I then use the sum of squared differences of the boxed images in order to find the best point:

$$SSD = \sum_{x,y} (I_{box1}(x,y) - I_{box2}(x,y))^2$$

Figure 7 shows an example of the point correspondences:

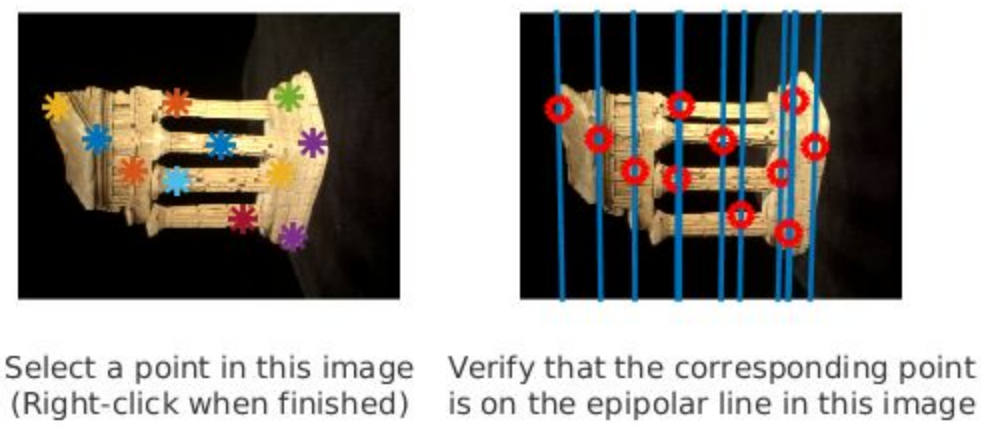

Figure 7) Point Correspondences Shown on epipolarMatchGUI

I saved my values in `q2_6.mat`.

Q2.7: Show data

Here I write a script called `q2_7.m` which loads in all of the data, finds the point correspondences for all 288 values, finds the correct M for the second camera, gets the 3D coordinates using triangulation and then plots the points.

Figure 8 shows the output of the point clouds:

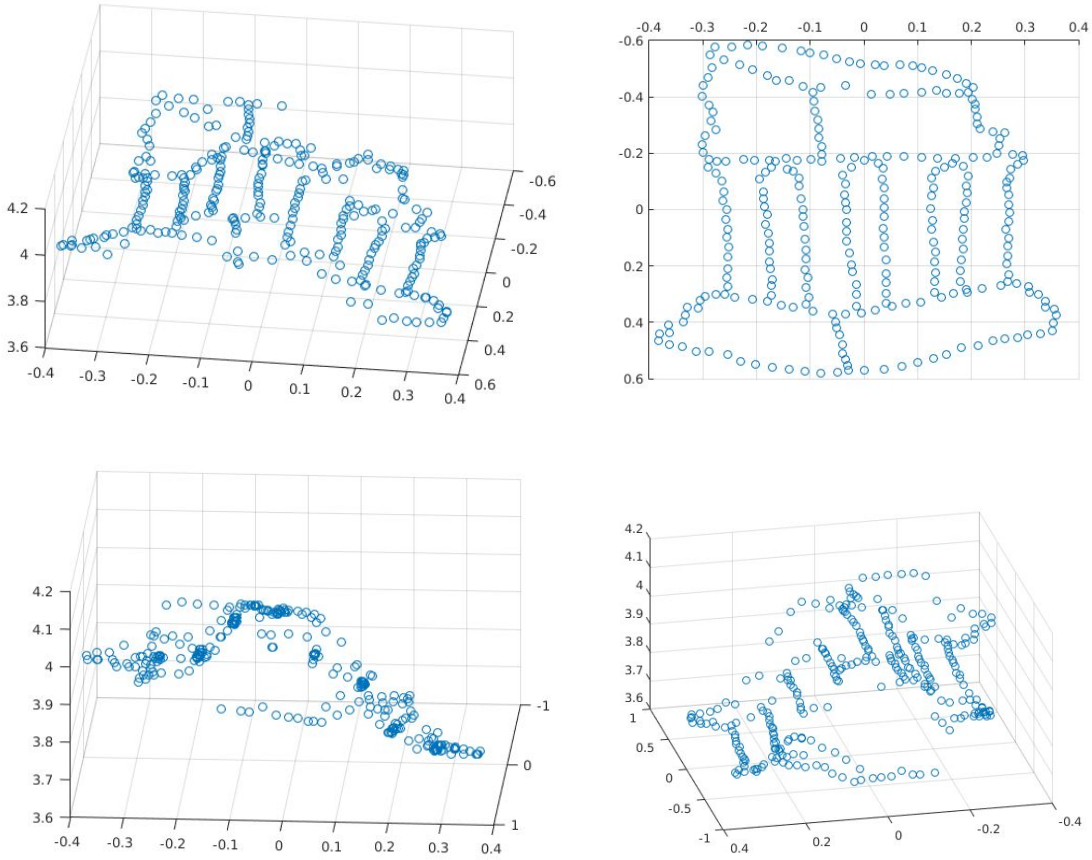

Figure 8) Point Clouds of the Temple Data

I saved the data in `q2_7.mat`.

Resources

[1] Epipolar Geometry and the Fundamental Matrix

[2] Previous Semester Homework Prompt

[3] Catadioptric Stereo with Planar Mirrors: Multiple-View Geometry and Camera Localization

[4] Computer Vision: Algorithms and Applications by Szeliski

[5] ICS Reconstruction Slides