16-720B Computer Vision: Homework 4

3D Reconstruction

Due: November 15 at 11:59pm

[redacted] (AndrewID: [redacted])

Part I Theory

Q1.1

According to the fundamental matrix relation we know that

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

Since $\mathbf{x}_1$ and $\mathbf{x}_2$ are at the coordinate origin, we can get

$$\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 0$$

Thus, we can derive from the equation above that $f_{33} = 0$

Q1.2

Since the translation is parallel to the x-axis, we can denote that the translation matrix is $\mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} t_x \\ 0 \\ 0 \end{bmatrix}$.

The cross product matrix can be written as

$$\mathbf{t}_\times = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -t_x \\ 0 & t_x & 0 \end{bmatrix}$$

Since it is a pure rotation, the rotation matrix would be

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

So we can get the essential matrix

$$\mathbf{E} = \mathbf{t}_\times \mathbf{R} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -t_x \\ 0 & t_x & 0 \end{bmatrix}$$

So the epipolar lines are

$$\mathbf{l}_1^T = \tilde{\mathbf{x}}_2^T \mathbf{E} = \begin{bmatrix} x_2 & y_2 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -t_x \\ 0 & t_x & 0 \end{bmatrix} = \begin{bmatrix} 0 & t_x & -t_x y_2 \end{bmatrix}$$

$$\mathbf{l}_2^T = \tilde{\mathbf{x}}_1^T \mathbf{E}^T = \begin{bmatrix} x_1 & y_1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & t_x \\ 0 & -t_x & 0 \end{bmatrix} = \begin{bmatrix} 0 & -t_x & t_x y_1 \end{bmatrix}$$

The equation of line 1 is $t_x y - t_x y_2 = 0$, and the equation of line 2 is $t_x y - t_x y_1 = 0$. These two lines are both parallel to the x-axis.

Q1.3

Suppose that the point in real world is $\mathbf{w} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$. Then at time 1 and time 2, the corresponding points in frame of reference of camera are:

$$\mathbf{w}_1 = \mathbf{R}_1 \mathbf{w} + \mathbf{t}_1$$

$$\mathbf{w}_2 = \mathbf{R}_2 \mathbf{w} + \mathbf{t}_2$$

Then we have

$$\mathbf{w} = \mathbf{R}_1^{-1}(\mathbf{w}_1 - \mathbf{t}_1)$$

So,

$$\begin{aligned} \mathbf{w}_2 &= \mathbf{R}_2 \mathbf{R}_1^{-1}(\mathbf{w}_1 - \mathbf{t}_1) + \mathbf{t}_2 \\ &= \mathbf{R}_2 \mathbf{R}_1^{-1} \mathbf{w}_1 - \mathbf{R}_2 \mathbf{R}_1^{-1} \mathbf{t}_1 + \mathbf{t}_2 \end{aligned}$$

So the effective rotation and translation between the two frames are:

$$\begin{aligned} \mathbf{R}_{rel} &= \mathbf{R}_2 \mathbf{R}_1^{-1} \\ \mathbf{t}_{rel} &= -\mathbf{R}_2 \mathbf{R}_1^{-1} \mathbf{t}_1 + \mathbf{t}_2 \end{aligned}$$

The essential matrix and the fundamental matrix are

$$\mathbf{E} = \mathbf{t}_{rel\times}\mathbf{R}_{rel}$$

$$\mathbf{F} = [\mathbf{K}^{-1}]^T\mathbf{E}\mathbf{K}^{-1} = [\mathbf{K}^{-1}]^T\mathbf{t}_{rel\times}\mathbf{R}_{rel}\mathbf{K}^{-1}$$

Q1.4

Assume that there is a point $\mathbf{p}$ of the object in 3D space and its reflection in the plane mirror is **p'**. Let the 2D coordinates on image1 and image2 corresponding to $\mathbf{p}$ are $\mathbf{x}_1$ and $\mathbf{x}_2$ respectively; 2D coordinates on image1 and image2 corresponding to **p'** are $\mathbf{x'}_1$ and $\mathbf{x'}_2$ respectively. Then we have that

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

Take the transpose of the above equation we get

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

Since the virtual camera (camera2) is a reflection of the true camera (camera1) in a plane mirror. Then $\mathbf{M}_2 = \mathbf{T}\mathbf{M}_1$, where $\mathbf{T}$ is a transition matrix of reflection and there is $\mathbf{T}^T\mathbf{T} = \mathbf{I}$. So we have

$$\tilde{\mathbf{x}}_1 = \mathbf{K}\mathbf{M}_1\tilde{\mathbf{p}}$$

$$\tilde{\mathbf{x}}_2 = \mathbf{K}\mathbf{T}\mathbf{M}_1\tilde{\mathbf{p}}$$

So,

$$\begin{aligned} & \tilde{\mathbf{x}}_2^T\mathbf{F}\tilde{\mathbf{x}}_1 + \tilde{\mathbf{x}}_1^T\mathbf{F}^T\tilde{\mathbf{x}}_2 = 0 \\ \rightarrow \quad & \tilde{\mathbf{p}}^T\mathbf{M}_1^T\mathbf{T}^T\mathbf{K}^T\mathbf{F}\mathbf{K}\mathbf{M}_1\tilde{\mathbf{p}} + \tilde{\mathbf{p}}^T\mathbf{M}_1^T\mathbf{K}^T\mathbf{F}^T\mathbf{K}\mathbf{T}\mathbf{M}_1\tilde{\mathbf{p}} = 0 \\ \rightarrow \quad & \tilde{\mathbf{p}}^T\mathbf{M}_1^T(\mathbf{T}^T\mathbf{K}^T\mathbf{F}\mathbf{K} + \mathbf{K}^T\mathbf{F}^T\mathbf{K}\mathbf{T})\mathbf{M}_1\tilde{\mathbf{p}} = 0 \end{aligned}$$

$$\begin{aligned} \rightarrow \quad & \mathbf{T}^T\mathbf{K}^T\mathbf{F}\mathbf{K} + \mathbf{K}^T\mathbf{F}^T\mathbf{K}\mathbf{T} = 0 \\ & \mathbf{K}^T\mathbf{F}\mathbf{K}\mathbf{T} + \mathbf{T}^T\mathbf{K}^T\mathbf{F}^T\mathbf{K} = 0 \end{aligned}$$

$$\begin{aligned} \rightarrow \quad & \mathbf{K}^T(\mathbf{F} + \mathbf{F}^T)\mathbf{K} = 0 \\ \rightarrow \quad & \mathbf{F} = -\mathbf{F}^T \end{aligned}$$

So the fundamental matrix is a skew-symmetric matrix.

Part II Theory

1. Overview

2. Fundamental matrix estimation

The Eight Point Algorithm

Q2.1: Function

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

is implemented to estimate the fundamental matrix. The matrix $\mathbf{F}$, scale `M` are saved to the file `q2_1.npz`.

The recovered $\mathbf{F}$ is as below:

$$\mathbf{F} = \begin{bmatrix} 9.78833288e-10 & -1.32135929e-07 & 1.12585666e-03 \\ -5.73843315e-08 & 2.96800276e-09 & -1.17611996e-05 \\ -1.08269003e-03 & 3.04846703e-05 & -4.47032655e-03 \end{bmatrix}$$

Fig.1 is an example of the output of `displayEpipolarF`

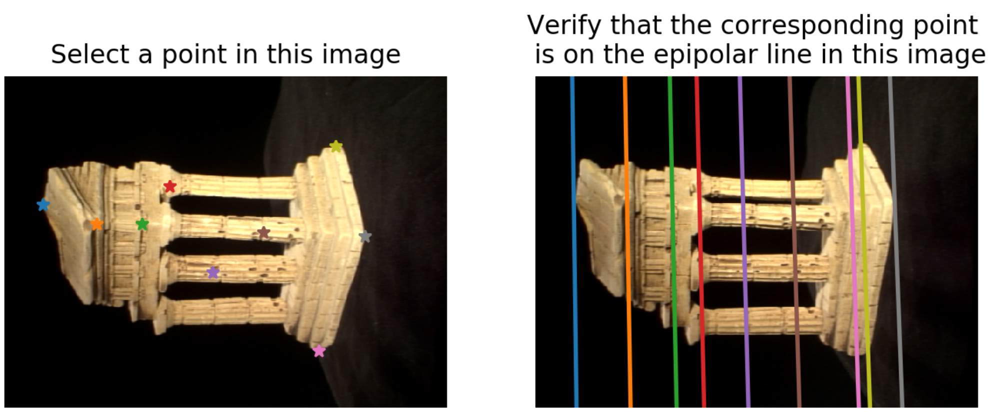

Figure 1: The visualization of the corresponding epipolar line using eight point algorithm.

The Seven Point Algorithm

Q2.2: Function

```
Farray = sevenpoint(pts1, pts2, M)
```

is implemented to calculate **F** using only seven point correspondences. The matrix **F**, scale `M`, 2D points `pts1` and `pts2` to the file `q2_2.npz`.

The recovered **F** is as below:

$$\mathbf{F} = \begin{bmatrix} 2.13988196e-08 & -8.17213112e-08 & 7.92729291e-04 \\ -6.92083811e-08 & -1.62680203e-09 & 1.58210573e-05 \\ -7.66381472e-04 & 3.28581738e-07 & -2.74246775e-03 \end{bmatrix}$$

Fig.2 is an example of the output of `displayEpipolarF` using the seven point algorithm.

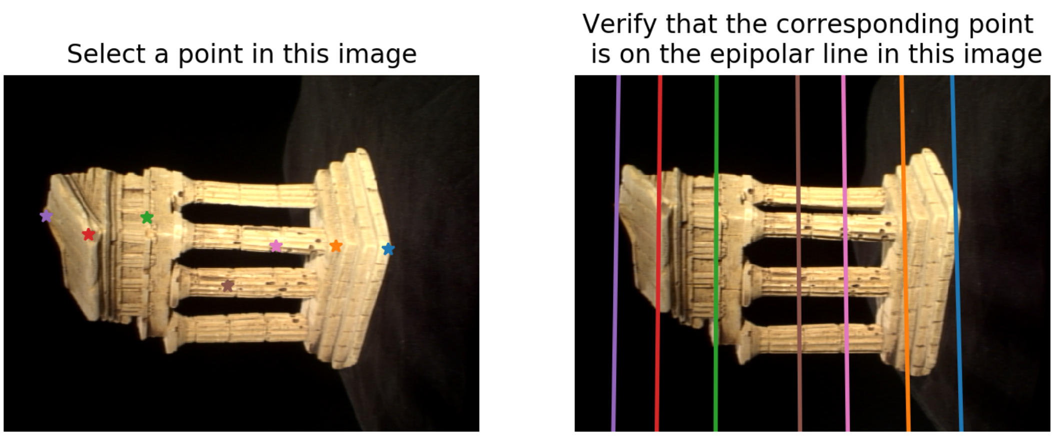

Figure 2: The visualization of the corresponding epipolar line using eight point algorithm.

3. Metric Reconstruction

Q3.1: Function

```
E = essentialMatrix(F, K1, K2)
```

is implemented to compute the essential matrix **E** given **F**, $\mathbf{K}_1$ and $\mathbf{K}_2$.

The estimated **E** using **F** from the eight-point algorithm is shown as below:

$$\mathbf{E} = \begin{bmatrix} 0.0023 & -0.3066 & 1.6626 \\ -0.1331 & 0.0069 & -0.0433 \\ -1.6672 & -0.0133 & -0.0007 \end{bmatrix}$$

Q3.2:

Let's denote that

$$\mathbf{C}_1 = \begin{bmatrix} c_{11}^{(1)} & c_{12}^{(1)} & c_{13}^{(1)} & c_{14}^{(1)} \\ c_{21}^{(1)} & c_{22}^{(1)} & c_{23}^{(1)} & c_{24}^{(1)} \\ c_{31}^{(1)} & c_{32}^{(1)} & c_{33}^{(1)} & c_{34}^{(1)} \end{bmatrix}, \quad \mathbf{C}_2 = \begin{bmatrix} c_{11}^{(2)} & c_{12}^{(2)} & c_{13}^{(2)} & c_{14}^{(2)} \\ c_{21}^{(2)} & c_{22}^{(2)} & c_{23}^{(2)} & c_{24}^{(2)} \\ c_{31}^{(2)} & c_{32}^{(2)} & c_{33}^{(2)} & c_{34}^{(2)} \end{bmatrix}$$

$$\tilde{\mathbf{w}}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}, \quad \tilde{\mathbf{x}}_{i1} = \begin{bmatrix} x_{i1} \\ y_{i1} \\ 1 \end{bmatrix}, \quad \tilde{\mathbf{x}}_{i2} = \begin{bmatrix} x_{i2} \\ y_{i2} \\ 1 \end{bmatrix}$$

Then we have

$$\mathbf{C}_1 \tilde{\mathbf{w}}_i = \lambda_{i1} \tilde{\mathbf{x}}_{i1}$$

$$\mathbf{C}_2 \tilde{\mathbf{w}}_i = \lambda_{i2} \tilde{\mathbf{x}}_{i2}$$

Further we can get that

$$\begin{bmatrix} c_{11}^{(1)} - c_{31}^{(1)} x_{i1} & c_{12}^{(1)} - c_{32}^{(1)} x_{i1} & c_{13}^{(1)} - c_{33}^{(1)} x_{i1} & c_{14}^{(1)} - c_{34}^{(1)} x_{i1} \\ c_{21}^{(1)} - c_{31}^{(1)} y_{i1} & c_{22}^{(1)} - c_{32}^{(1)} y_{i1} & c_{23}^{(1)} - c_{33}^{(1)} y_{i1} & c_{24}^{(1)} - c_{34}^{(1)} y_{i1} \\ c_{11}^{(2)} - c_{31}^{(2)} x_{i2} & c_{12}^{(2)} - c_{32}^{(2)} x_{i2} & c_{13}^{(2)} - c_{33}^{(2)} x_{i2} & c_{14}^{(2)} - c_{34}^{(2)} x_{i2} \\ c_{21}^{(2)} - c_{31}^{(2)} y_{i2} & c_{22}^{(2)} - c_{32}^{(2)} y_{i2} & c_{23}^{(2)} - c_{33}^{(2)} y_{i2} & c_{24}^{(2)} - c_{34}^{(2)} y_{i2} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = \mathbf{0}$$

So, this can be written as $\mathbf{A}\tilde{\mathbf{w}}_i = 0$, where

$$\mathbf{A} = \begin{bmatrix} c_{11}^{(1)} - c_{31}^{(1)} x_{i1} & c_{12}^{(1)} - c_{32}^{(1)} x_{i1} & c_{13}^{(1)} - c_{33}^{(1)} x_{i1} & c_{14}^{(1)} - c_{34}^{(1)} x_{i1} \\ c_{21}^{(1)} - c_{31}^{(1)} y_{i1} & c_{22}^{(1)} - c_{32}^{(1)} y_{i1} & c_{23}^{(1)} - c_{33}^{(1)} y_{i1} & c_{24}^{(1)} - c_{34}^{(1)} y_{i1} \\ c_{11}^{(2)} - c_{31}^{(2)} x_{i2} & c_{12}^{(2)} - c_{32}^{(2)} x_{i2} & c_{13}^{(2)} - c_{33}^{(2)} x_{i2} & c_{14}^{(2)} - c_{34}^{(2)} x_{i2} \\ c_{21}^{(2)} - c_{31}^{(2)} y_{i2} & c_{22}^{(2)} - c_{32}^{(2)} y_{i2} & c_{23}^{(2)} - c_{33}^{(2)} y_{i2} & c_{24}^{(2)} - c_{34}^{(2)} y_{i2} \end{bmatrix}$$

Function

```
[w, err] = triangulate(C1, pts1, C2, pts2)
```

is implemented to triangulate a set of 2D coordinates in the image to a set of 3D points

Q3.3: Script `findM2.py` is implemented to obtain the correct `M2` from `M2s` by testing the four solutions through triangulations. The correct `M2`, the corresponding `C2`, and 3D points `P` are saved to `q3_3.npz`.

4. 3D Visualization

Q4.1: Function

```
[x2, y2] = epipolarCorrespondence(im1, im2, F, x1, y1)
```

is implemented to estimate the coordinates of the pixel on `im2` which correspond to the input point on `im1`. The matrix **F**, points `pts1` and `pts2` that used to generate the screenshot are saved to the file `q4_1.npz`.

The screenshot of `epipolarMatchGUI` with some detected correspondences is shown as Fig.3.

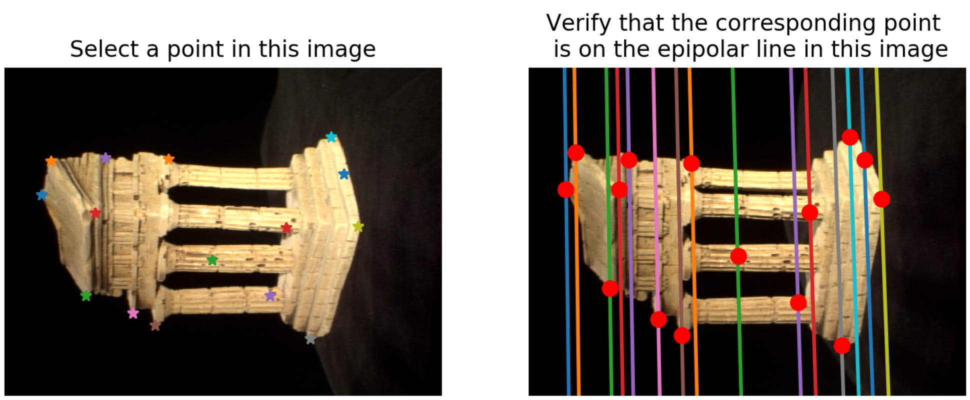

Figure 3: A screenshot of `epipolarMatchGUI` with some detected correspondences

Q4.2: Script `visualize.py` is implemented to generate the 3D reconstruction. The matrix **F**, matrices `M1`, `M2`, `C1`, `C2` that are used to generate the screenshots are saved to the file `q4_2.npz`.

Fig.4 shows some screenshots of the 3D visualization.

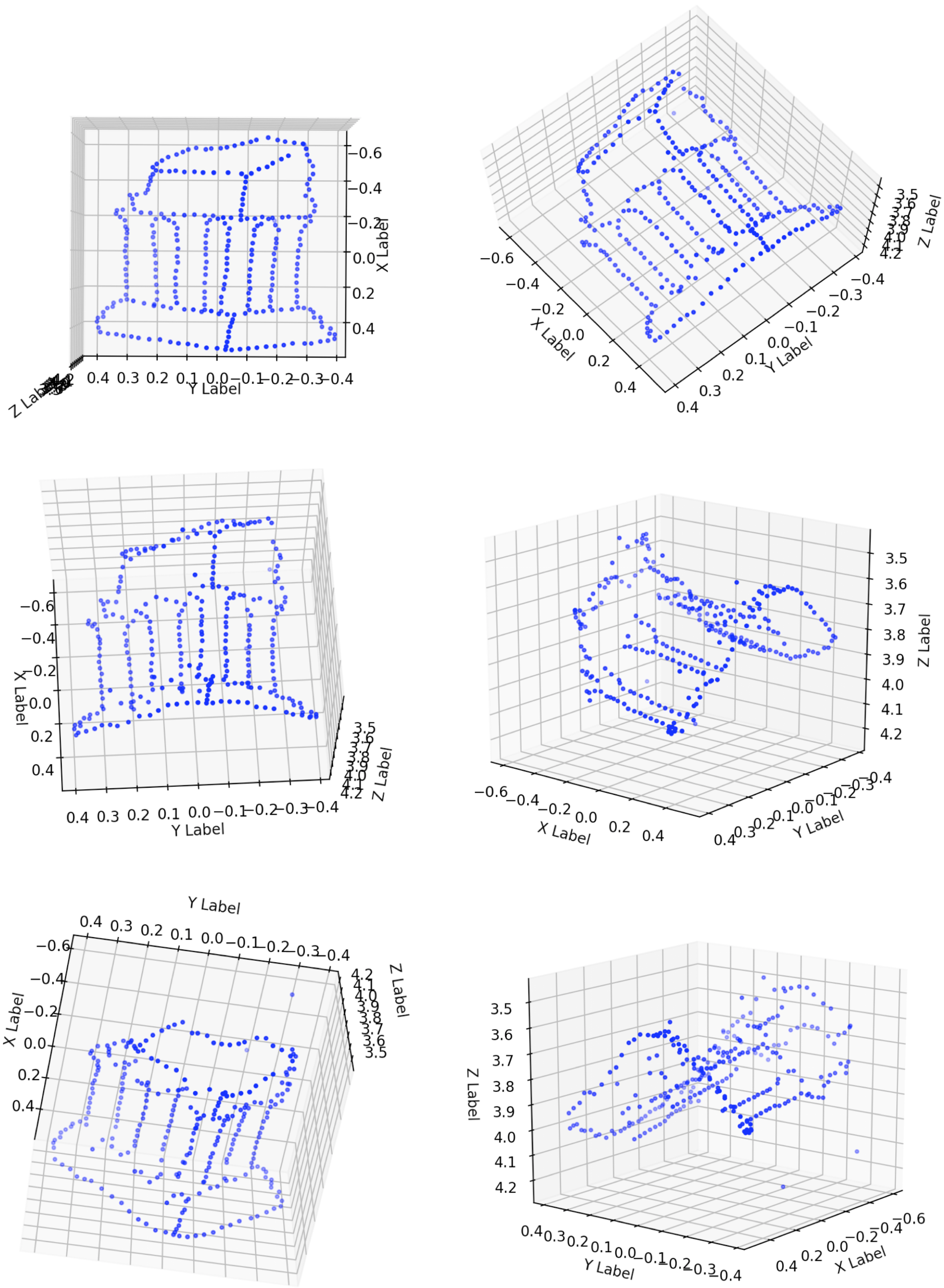

Figure 4: Examples of point cloud.

5. Bundle Adjustment

Q5.1: Function

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

is implemented to estimate the fundamental matrix **F** using the RANSAC method.

In each iteration of the RANSAC method (totally 1000 iterations used in the script), seven pairs of points are selected to calculate the fundamental matrix **F**. Then the fundamental matrix **F** is used to calculate the epipolar line on image2 for each point in `pts1`. Calculating the distance from each point in `pts2` to the corresponding epipolar line. When the distance is below the threshold, it is regarded as an inlier.

By using RANSAC, the result of `displayEpipolarF` is shown as Fig.5. The result of `displayEpipolarF` using eightpoint on the noisy coorespondance is shown as Fig.6.

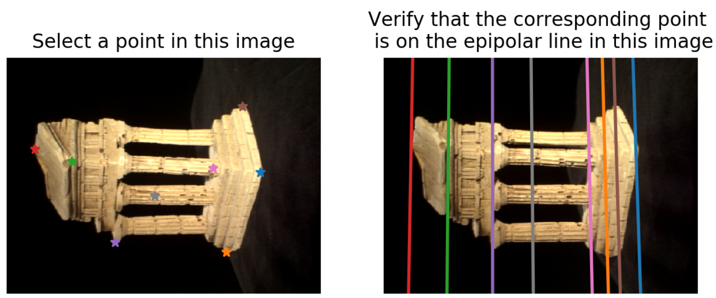

Figure 5: The result of `displayEpipolarF` using RANSAC.

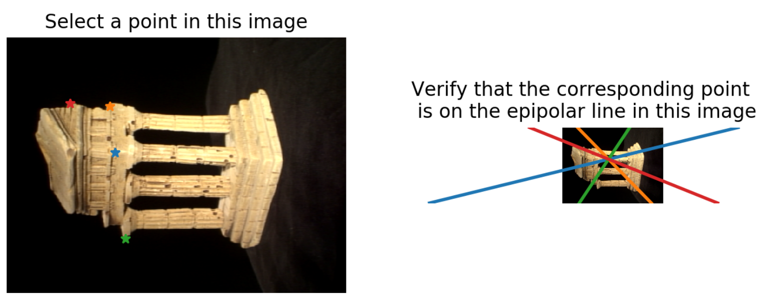

Figure 6: The result of `displayEpipolarF` using eightpoint.

Q5.2: Functions

```
R = rodrigues(r),   r = invRodrigues(R)
```

are implemented to make conversions between Rodrigues vector $\mathbf{r}$ and the rotation matrix $\mathbf{R}$.

Q5.3: Function

```
[M2, P] = bundleAdjustment(K1, M1, p1, K2, M2 init, p2, P init)
```

is implemented to perform the bundle adjustment.

The image of the original 3D points and the optimized points with initial $\mathbf{M}_2$ and $\mathbf{P}$, and with the optimized matrices is shown as Fig.7.

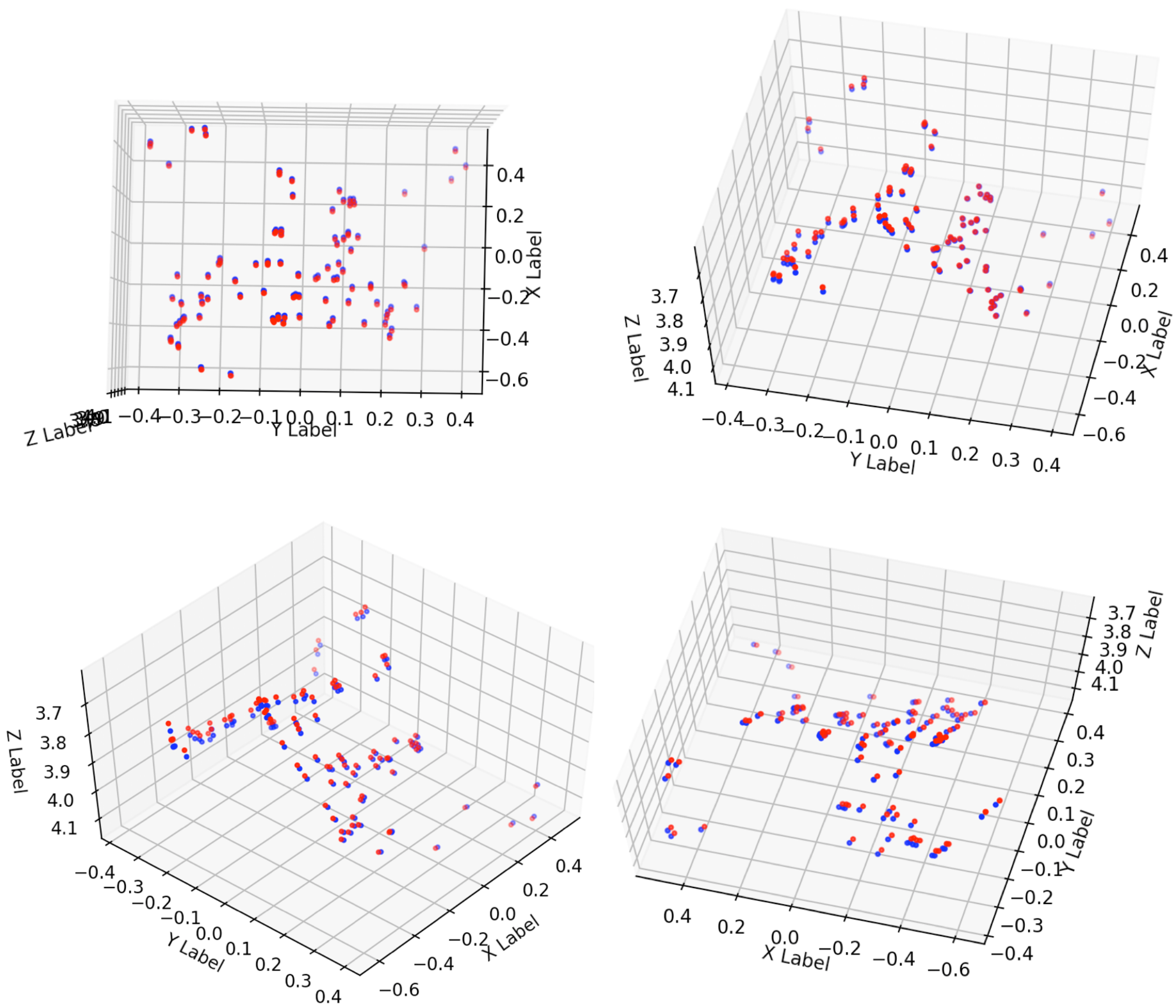

Figure 7: The original 3D points and the optimized points

The reprojection error with initial $\mathbf{M}_2$ and $\mathbf{P}$ is **4969.0288**; The reprojection error with the optimized matrices is **6.3609**.