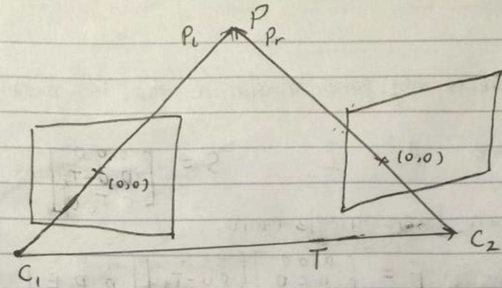
Writeup

Kai Li from MRSD

Q1.1

Q1.1



$$P_r = R(P_l - T)$$

$$P_l - T = R^{-1} P_r = R^T P_r$$

Since P_l, T and $P_l - T$ are coplanar.

$$(P_l - T)^T \cdot T \times P_l = 0$$

Therefore $(R^T P_r)^T \cdot T \times P_l = 0$

$$\because\ T \times P_l = S P_l = \underbrace{\begin{vmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{vmatrix}}_{S} \underbrace{\begin{bmatrix} P_{lx} \\ P_{ly} \\ P_{lz} \end{bmatrix}}_{P_l}$$

So, $P_r^T \underbrace{RS}_{E \to \text{essential matrix}} P_l = 0$

let $x_l = K_l P_l$

$x_r = K_r P_r$ Therefore $x_r^T \underbrace{(K_r^{-T} E K_l^{-1})}_{F} x_l = 0$

let $x_l = \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$, $x_r = \begin{pmatrix} x_1' \\ y_1' \\ 1 \end{pmatrix}$ $\Rightarrow \begin{bmatrix} x_1 x_1' & y_1 x_1' & x_1' & x_1 y_1' & y_1 y_1' & y_1' & x_1 & y_1 & 1 \\ & & & & \vdots & & & & \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ \vdots \\ f_{33} \end{bmatrix} = 0.$

Because for every pair (x_n, y_n) and (x_n', y_n'), the equation above is valid.

Use the points shown in the drawing: $(x_1, y_1) = (0,0)$, $(x_1', y_1') = (0,0)$

Therefore we get $f_{33} = 0$.

Q1.2 & Q1.3

Q 1.2 Because there is only pure translation that is parallel to the x-axis.

Therefore $R = I$, $S = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T_x \\ 0 & T_x & 0 \end{bmatrix}$, so $E = RS = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T_x \\ 0 & T_x & 0 \end{bmatrix}$

Since K_1^{-T} is an unknown lower triangle matrix, and K_2^{-1} is an upper triangle matrix.

We can assume $F = \begin{bmatrix} a & 0 & 0 \\ b & c & 0 \\ d & e & f \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T_x \\ 0 & T_x & 0 \end{bmatrix} \begin{bmatrix} A & B & C \\ 0 & D & E \\ 0 & 0 & F \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -cFT_x \\ 0 & fDT_x & fET_x - eFT_x \end{bmatrix}$

$\because$ Given a point $P_1 = \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$ in the left image, its corresponding epipolar line is parametrized by $\tilde{l} = \begin{pmatrix} 0 \\ -cFT_x \\ fDT_x \cdot y_1 + fET_x - eFT_x \end{pmatrix}$. The line is $(x, y, 1) \cdot \tilde{l} = -cFT_x y + fDT_x \cdot y_1 + fET_x - eFT_x$ which bespeaks the epipolar lines in two cameras are also parallel to the x-axis.

The similar reasoning also apply to given a point P_2 in the right image.

Q 1.3 let P_0 be the base frame.

$$P_l = R_1(P_0 - t_1) \; ; \quad P_r = R_2(P_0 - t_2)$$

Therefore $P_r = R_2 R_1^T [P_l - R_1(t_2 - t_1)]$

$$R_{rel} = R_2 R_1^T$$

$$t_{rel} = R_1(t_2 - t_1), \quad \text{let } t_{rel} = \begin{pmatrix} t_{xrel} \\ t_{yrel} \\ t_{zrel} \end{pmatrix}$$

$$E = RS = R_{rel} \begin{pmatrix} 0 & -t_{zrel} & t_{yrel} \\ t_{zrel} & 0 & -t_{xrel} \\ -t_{yrel} & t_{xrel} & 0 \end{pmatrix}$$

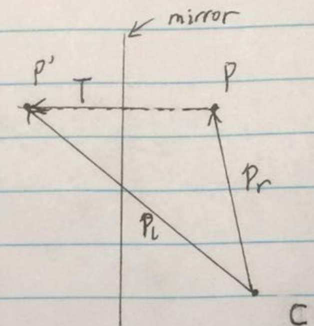
$$= R_2 R_1^T \begin{pmatrix} 0 & -t_{zrel} & t_{yrel} \\ t_{zrel} & 0 & -t_{xrel} \\ -t_{yrel} & t_{xrel} & 0 \end{pmatrix}$$

$$F = K'^{-T} E K^{-1}$$

$$= K'^{-T} R_2 R_1^T \begin{pmatrix} 0 & -t_{zrel} & t_{yrel} \\ t_{zrel} & 0 & -t_{xrel} \\ -t_{yrel} & t_{xrel} & 0 \end{pmatrix} K^{-1}$$

Q1.4

Q1.4



P is the real object, P' is P's mirror reflection. C is the camera.

$$\because \quad P_r + T = P_l$$

$$P_r = P_l - T$$

Because P_l, T and $P_l - T$ are coplanar.

$$(P_l - T)^T \cdot T \times P_l = 0$$

$$T \times P_l = \begin{vmatrix} i & j & k \\ T_x & T_y & T_z \\ P_{lx} & P_{ly} & P_{lz} \end{vmatrix} = \underbrace{\begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}}_{S} \underbrace{\begin{bmatrix} P_{lx} \\ P_{ly} \\ P_{lz} \end{bmatrix}}_{P_l}$$

Therefore $(P_l - T)^T \cdot S \cdot P_l = 0$

$$P_r^T \cdot S \cdot P_l = 0$$

Here S equals to E (essential matrix), $S^T = -S$.

$$\because \quad F = K^{-T} E K^{-1}$$

$$= K^{-T} S K^{-1}$$

$$F^T = K^{-T} S^T K^{-1} = -(K^{-T} S K^{-1}) = -F$$

Therefore F is skew-symmetric, and this situation is equivalent to having two images of the object which are related by a skew-symmetric fundamental matrix.

Q2.1

The F matrix is:

F =

-0.0000	-0.0000	0.0022
-0.0000	0.0000	0.0000
-0.0021	0.0000	-0.0092

And the output images are shown below:

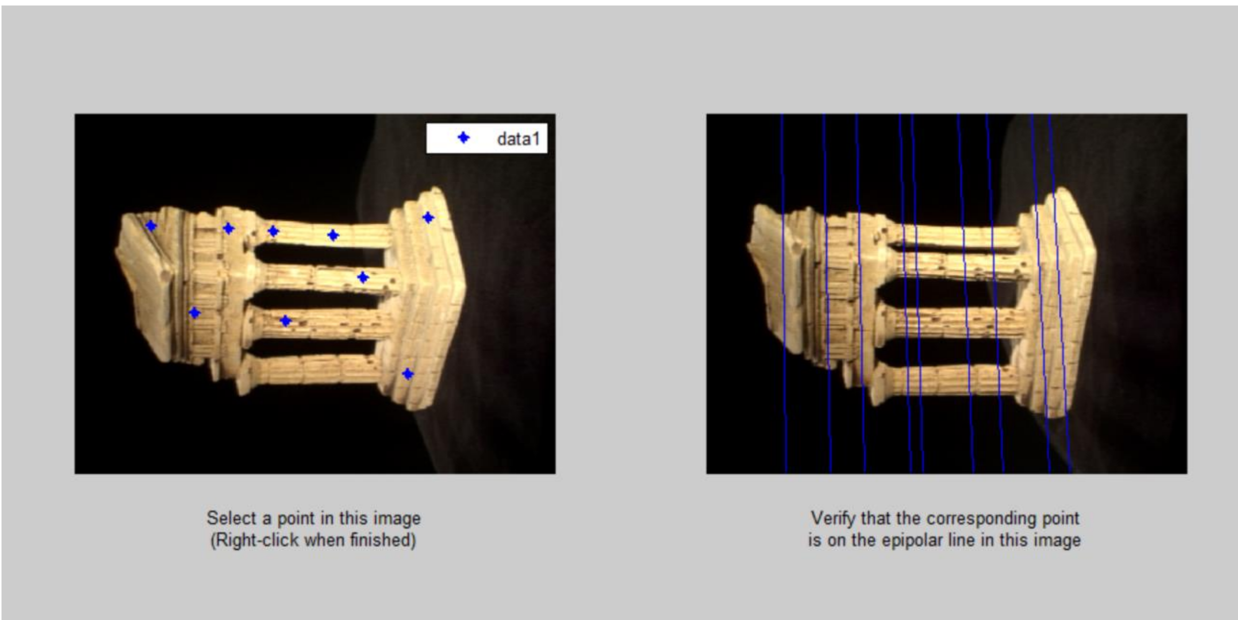


Figure1. The output sample images (eightpoint method)

The detailed code can be referred to eightpoint.m

Q2.2

Use manually selected points with cpselect, I picked:

pts1 =

```
   79.0000  135.5000
  228.0000  129.5000
  424.0000  224.5000
  373.0000  205.5000
  423.0000  126.5000
  230.0000  318.5000
  116.0000  331.5000
```

pts2 =

```
   78.0000  122.5000
  228.0000  133.5000
  424.0000  201.5000
  373.0000  189.5000
  418.0000  134.5000
  231.0000  330.5000
  118.0000  320.5000
```

Then using sevenpoint method, I got F:

F =

```
   -0.0000    0.0000    0.0020
   -0.0000   -0.0000    0.0004
   -0.0019   -0.0004   -0.0003
```

And the output sample image is:

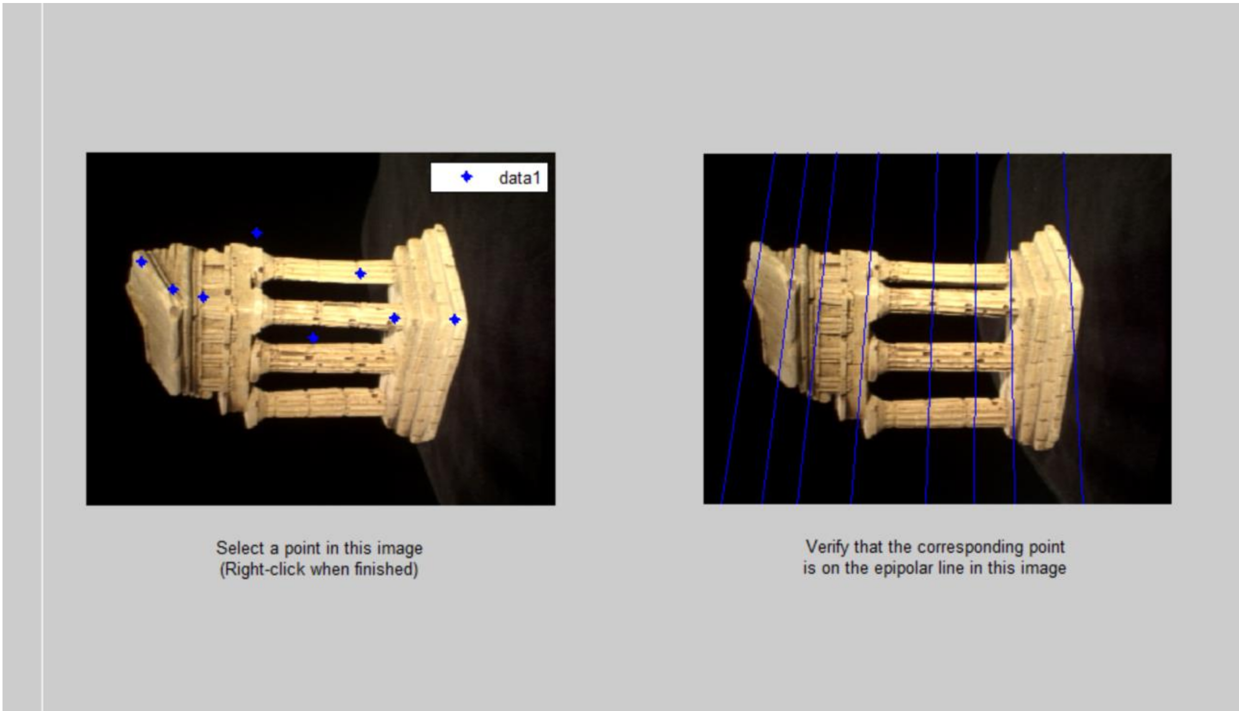


Figure 2. The output sample images (sevenpoint method)

The detailed code can be referred to sevenpoint.m

Q2X (Extra)

The error metrics basically follows the distance from point to epipolar line shown as figure 3:

$$d(p, Fp') = \frac{p^T Fp'}{\sqrt{(Fp')_1^2 + (Fp')_2^2}}$$

Figure 3. Error metrics for ransacF.m

I set a threshold t = 5e-2; the distance between the point and the mapped epipolar line that within this threshold will make that pair of points as inliers. I first ran the seven point methods with the minimal set of points to formulate F matrix. Then I used the error metrics logic to get the inliers pairs. Whenever the correct pairs' number reach 75% of the total pairs, I stopped the while loop. Finally, I use eightpoint method on the correct pairs to get the refined F using all the inliers.

F =

0.0000	0.0000	0.0019
-0.0000	0.0000	-0.0001
-0.0019	0.0001	-0.0052

And the output image is:

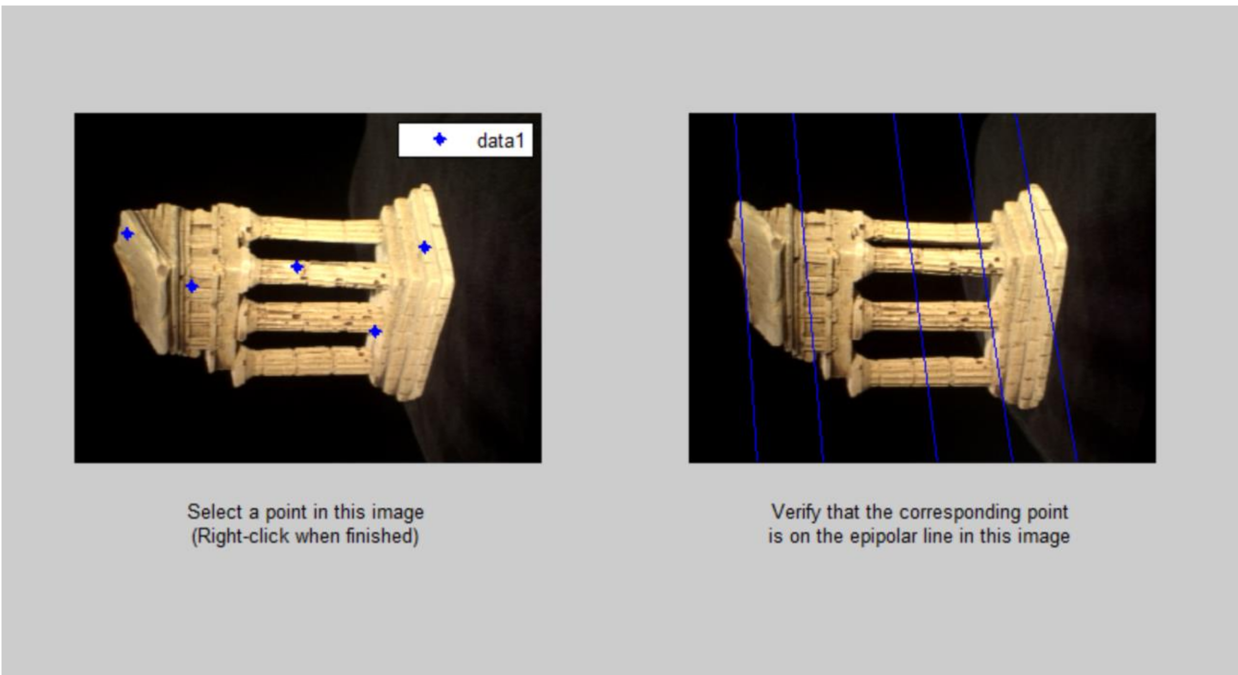


Figure4. The output sample images (ransacF method)

Compared to the output if we use eightpoint method on the noisy coorespondances, see figure 5:

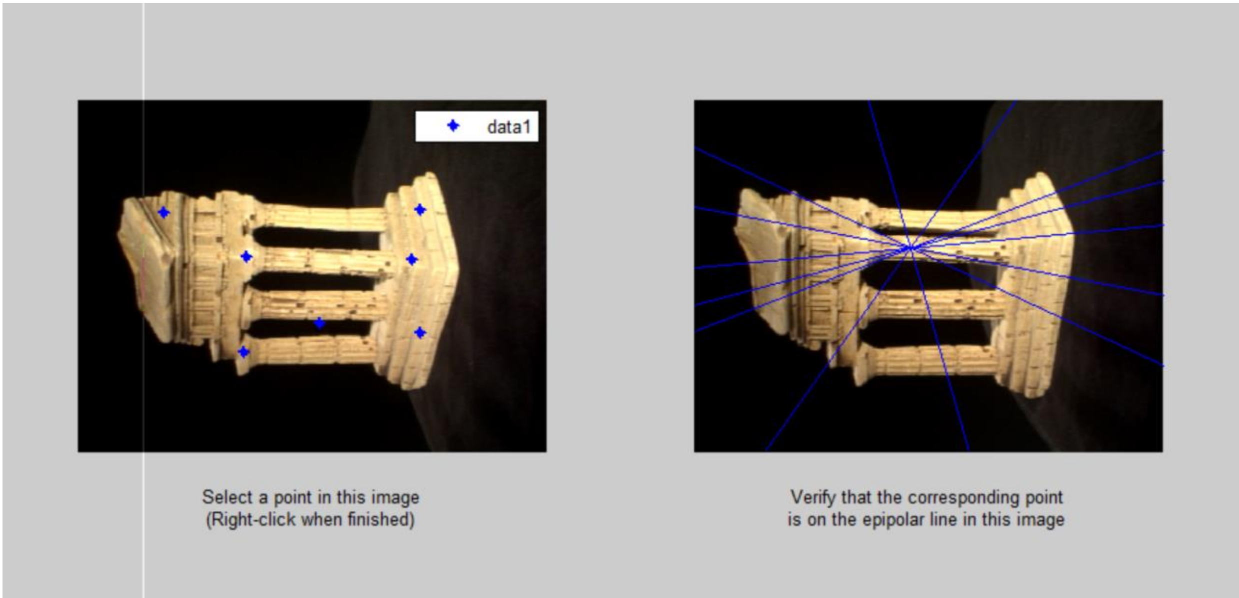


Figure5. The output sample images (eightpoint method on noisy coorespondances)

Therefore we can see after Ransac, the output image looks better! That's because we got rid of many wrong matching pairs and increased the correct matches possibility. Detailed code can be seen in ransacF.m.

Q2.3

E= K2'*F*K1;

Please see detailed code in essentialMatrix.m

Q2.4

Please refer to triangulate.m for detailed code. And please note that the error term I used here may have different scale as recommended.

Q2.5

Please refer to script findM2.m for details.

Q2.6

Please refer to epipolarCorrespondence.m for more details. And the screenshot of epipolarMatchGUI with some detected correspondences is shown below:

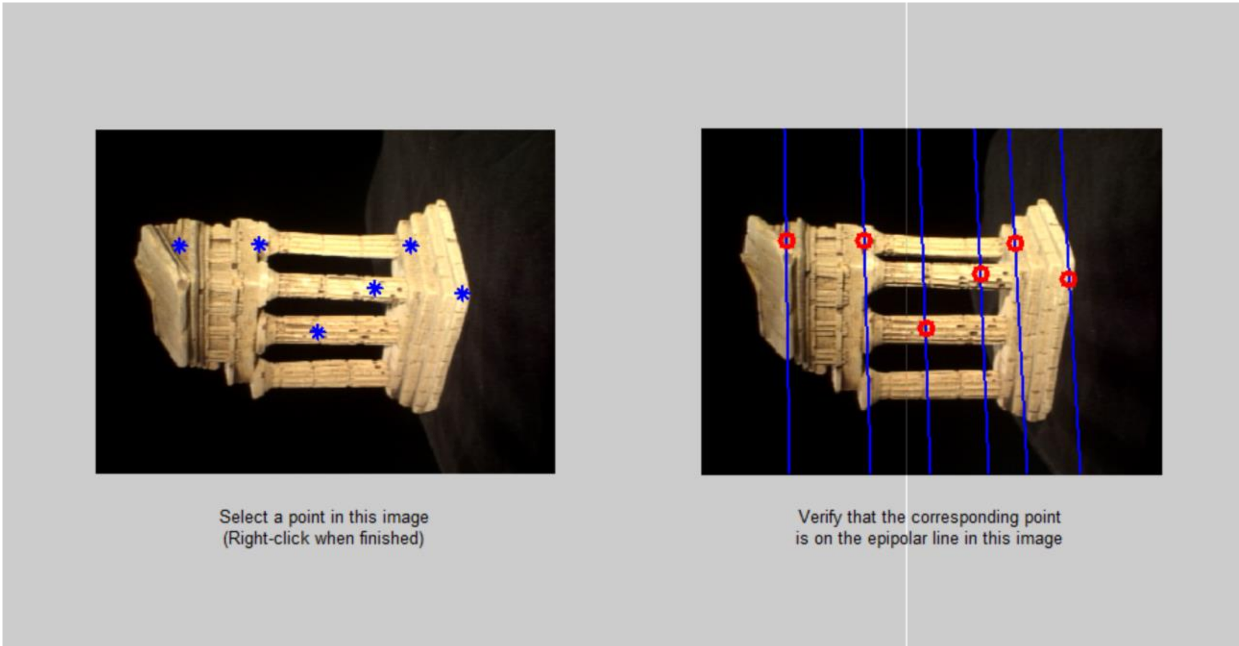


Figure 6. correspondences found by calling epipolarCorrespondence function

Q2.7

The following images are the point cloud I generated. It's amazing!

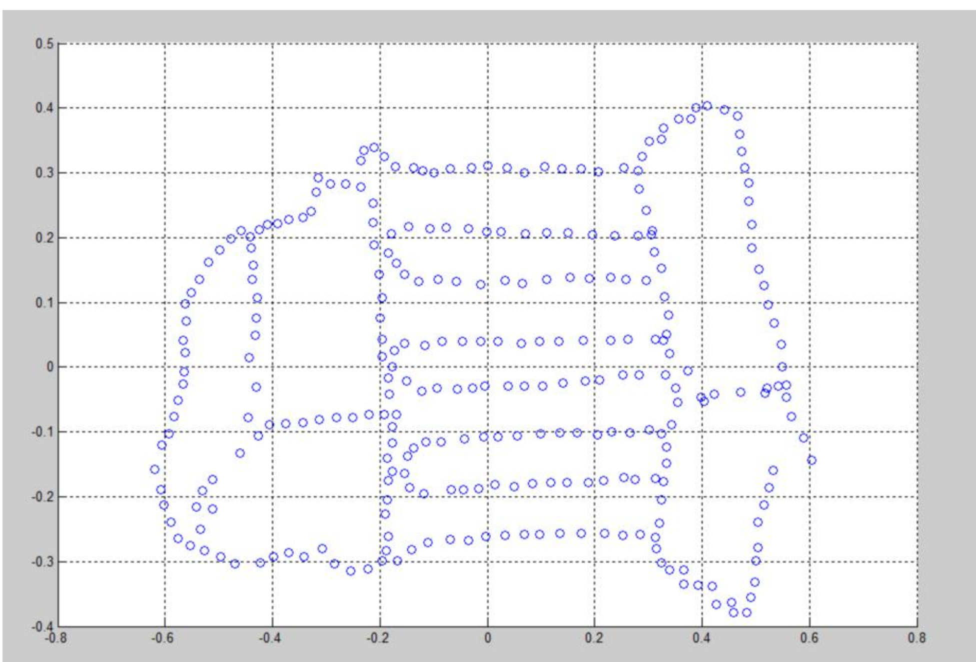


Figure 7

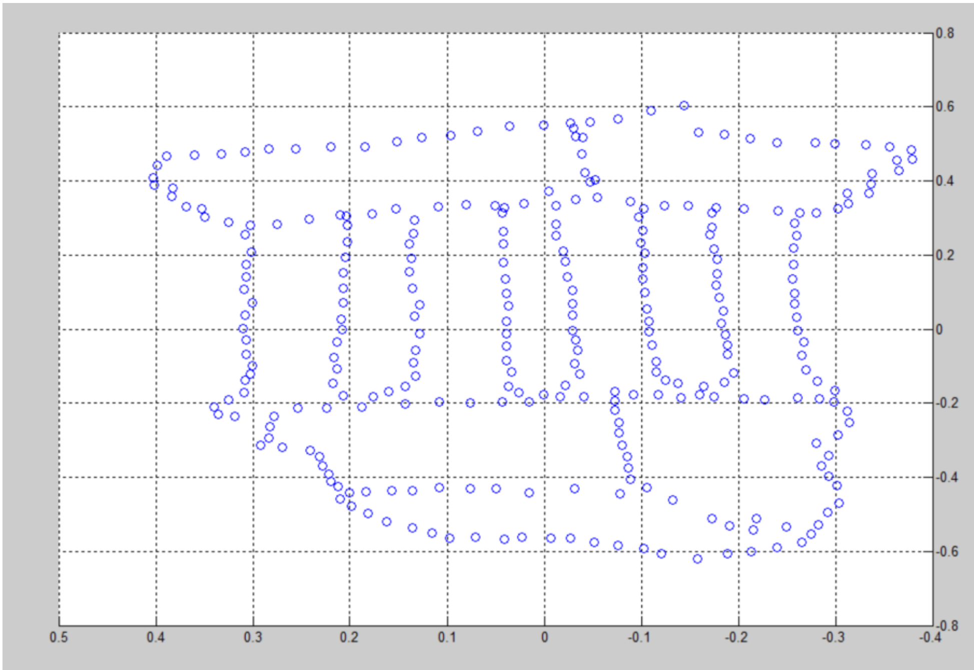


Figure 8

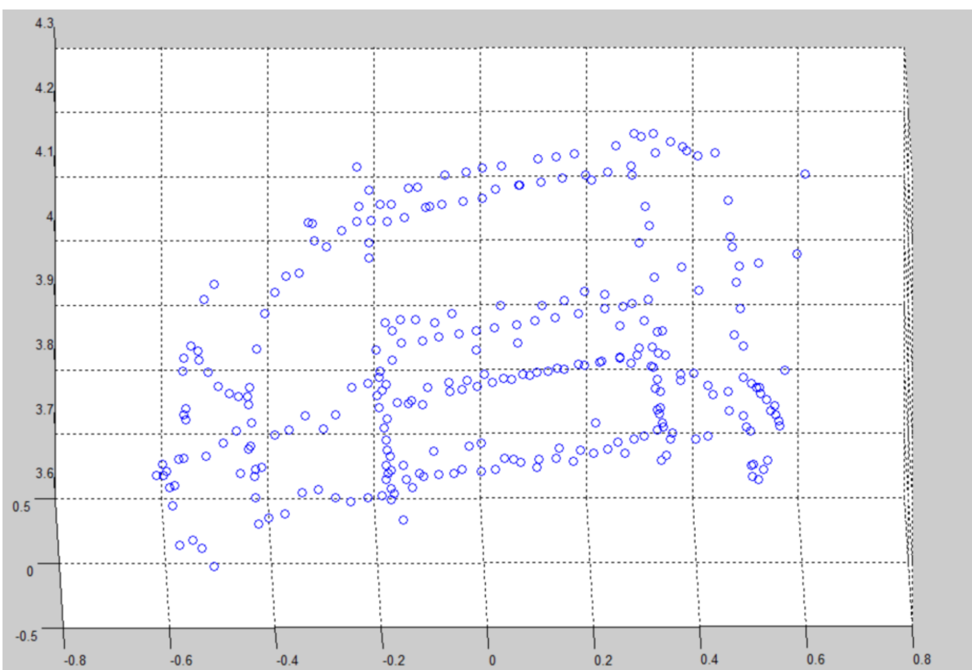


Figure 9

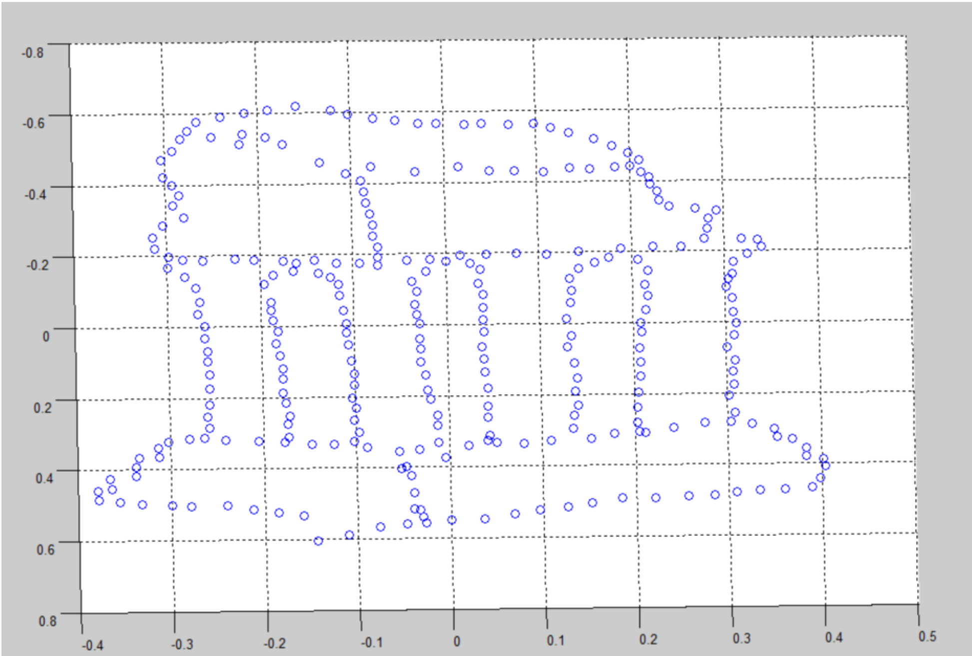


Figure 10

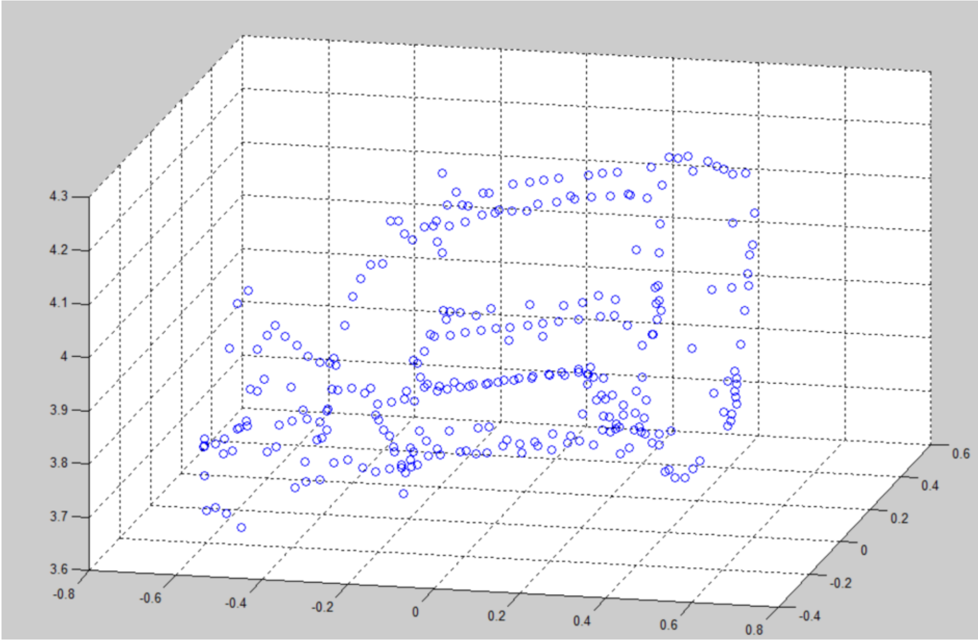


Figure 11

QX (Extra)

For this extra part, I found very useful tutorial on http://www.mathworks.com/help/vision/examples/structure-from-motion-from-two-views.html

And the basic idea is to get more points involved in this reconstruction process and record the pixel value of each points on image. And I used functions in 3-D Point Cloud Processing in **MATLAB 2016a Computer Vision System Toolbox** to generate much better visualization effects! Please see the following pictures:

Note: please refer to Q_X.m for detailed code. This may take long time to process and need to be tested on Matlab 2016a/2016b version. Thank you for this amazing homework!