16-720 Computer Vision
Spring 2020

Homework 4

Alex Wu, chichiaw@andrew.cmu.edu

1 Part I

1.1 Q1.1

$$x_r^T \cdot F x_l = 0$$

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

$$\therefore f_{33} = 0$$

1.2 Q1.2

$$
\begin{aligned}
&P_l \equiv K_l \begin{bmatrix} R|t \end{bmatrix} \begin{bmatrix} P \\ 1 \end{bmatrix}, (assume : K_l = I, R = I, t = 0) \\
&\therefore P_l \equiv P \\
&P_r \equiv K_r \begin{bmatrix} R|t \end{bmatrix} \begin{bmatrix} P \\ 1 \end{bmatrix}, (assume : K_r = I, R = I, t = t) \\
&\therefore P_r \equiv [I|t]P \\
&t \times P_r \equiv t \times [I|t] \begin{bmatrix} P_l \\ 1 \end{bmatrix} = t \times P_l \\
&t = \begin{bmatrix} t_x \\ 0 \\ 0 \end{bmatrix} \quad [t_x] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -t_x \\ 0 & t_x & 0 \end{bmatrix}
\end{aligned}
$$

$Epipolarlines : (K_l = K_r = I)$

$$
\begin{aligned}
&l' = Fx = Ex = [t_x]x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -t_x \\ 0 & t_x & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -t_x \\ t_x y \end{bmatrix} \\
&l = F^T x' = E^T x' = [t_x]x' = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & t_x \\ 0 & -t_x & 0 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ t_x \\ -t_x y' \end{bmatrix} \\
&l_1 : -t_x + t_x y = 0 \\
&l_2 : t)x - t_x y' = 0
\end{aligned}
$$

l_1 and l_2 are parallel to $x - axis$

1.3 Q1.3

$$\text{real point w(world)}$$
$$\text{at time 1: } w_1 = R_1 \cdot w + t_1$$
$$\text{at time 2: } w_2 = R_2 \cdot w + t_2$$
$$\therefore R_1^{-1}(w_1 - t_1) = w$$

$$\begin{aligned}
w2 &= R_2 \cdot R_1^{-1}(w_1 - t_1) + t_2 \\
&= R_2 R_1^{-1} w_1 - R_2 R_1^{-1} t_1 + t_2 \\
\therefore R_{rel} &= R_2 R_1^{-1} \\
t_{rel} &= t_2 - R_2 R_1^{-1} t_1 \\
E &= [t_x] R_{rel} \\
F &= K^{-T} E K^{-1} \\
&= K^{-T} [t_x] R_{rel} K^{-1}
\end{aligned}$$

1.4 Q1.4

Camera (c), the object(P_r), and its reflection(P_l) form a plane in the 3D space. A vector T is pointing from P_r to P_l.

$$P_r + T = P_l$$

T, $P_l - T$, P_r, and P_l these four vectors are coplanar.

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

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

$$\therefore E = T$$

$$T = \begin{bmatrix} 0 & t_1 & -t_2 \\ t_1\_1 & 0 & -t_3 \\ t_2 & -t_3 & 0 \end{bmatrix} = E = -E^T$$

$$F = K^{-T} E K^{-1}, F^T = K^{-T} E^T K^{-1}$$

$$F = -F^T \text{(skew symmetric)}$$

2 Part II - Fundamental Matrix Estimation

2.1 Eight Point Algorithm

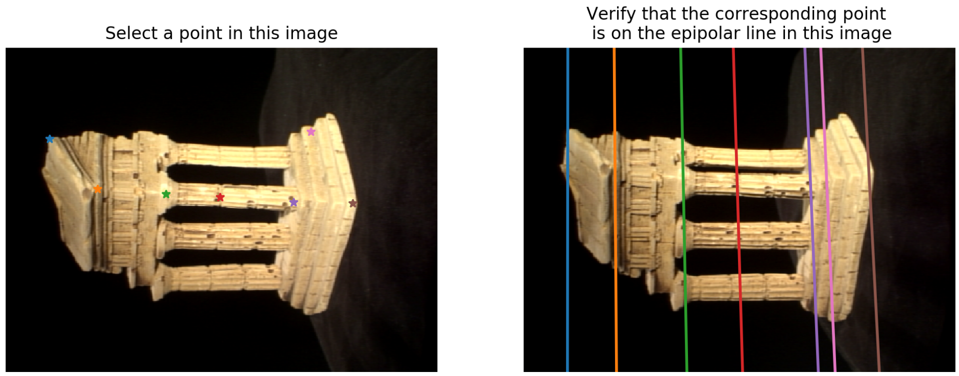


Figure 1: Eight-point algorithm result

$$F = \begin{bmatrix} 8.6 \times 10^{-8} & -1.5 \times 10^{-5} & 2.9 \times 10^{-1} \\ -3.4 \times 10^{-5} & 1.8 \times 10^{-7} & 2.5 \times 10^{-3} \\ -2.8 \times 10^{-1} & 2.6 \times 10^{-3} & -1.16 \end{bmatrix}$$

3 PartII - Metric Reconstruction

3.1 Essential Matrix

$$E = K_2^T F K_1$$

$$E = \begin{bmatrix} 1.9 \times 10^{-1} & -3.5 \times 10 & 4.4 \times 10^{2} \\ -8.0 \times 10 & 4.2 \times 10^{-1} & -1.2 \times 10 \\ -4.5 \times 10^{2} & -3.0 & -1.8 \times 10^{-1} \end{bmatrix}$$

3.2 Triangulation

$x = \alpha PX,$ (x: points projected on 2D image, P: camera matrix, X: points in 3D)
$x = \alpha PX, \quad x' = \alpha P' X$

$$P = \begin{bmatrix} P_1^T \\ P_2^T \\ P_3^T \end{bmatrix}, \quad P' = \begin{bmatrix} P_1'^T \\ P_2'^T \\ P_3'^T \end{bmatrix}$$

x, PX (x',P'X) are in the same direction, therefore the cross product of them are zero.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \times \begin{bmatrix} P_1^T X \\ P_2^T X \\ P_3^T X \end{bmatrix} = \begin{bmatrix} yP_3^T - P_2^T \\ P_1^T - xP_3^T \\ xP_2^T - yP_1^T \end{bmatrix} \begin{bmatrix} X \end{bmatrix} = 0$$

The third row is actually linear combination of the first and the second row, so we need at least two points to solve for three unknown.

$$A = \begin{bmatrix} yP_3^T - P_2^T \\ P_1^T - xP_3^T \\ y'P_3'^T - P_2'^T \\ P_1'^T - x'P_3'^T \end{bmatrix}$$

4 PartII - 3D Visualization

4.1 Epipolar Match

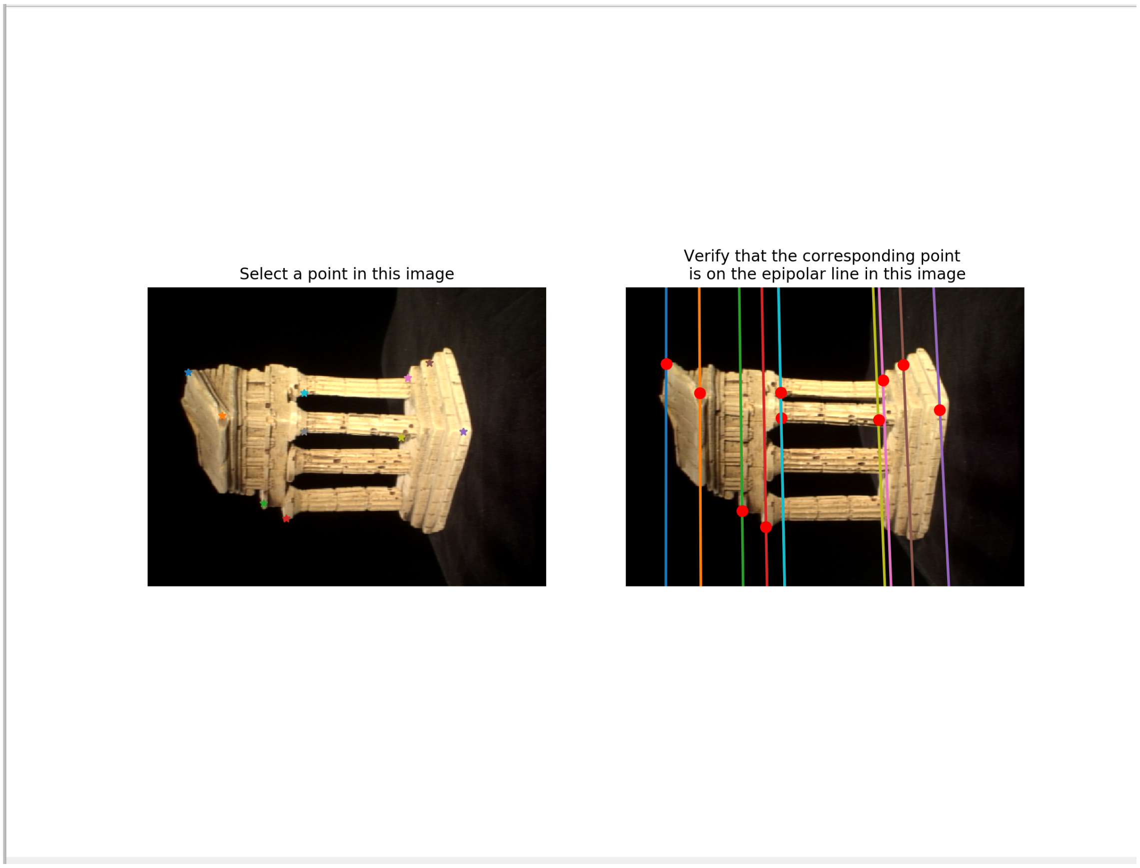


Figure 2: Eipipolar Match result

4.2 3D visualization

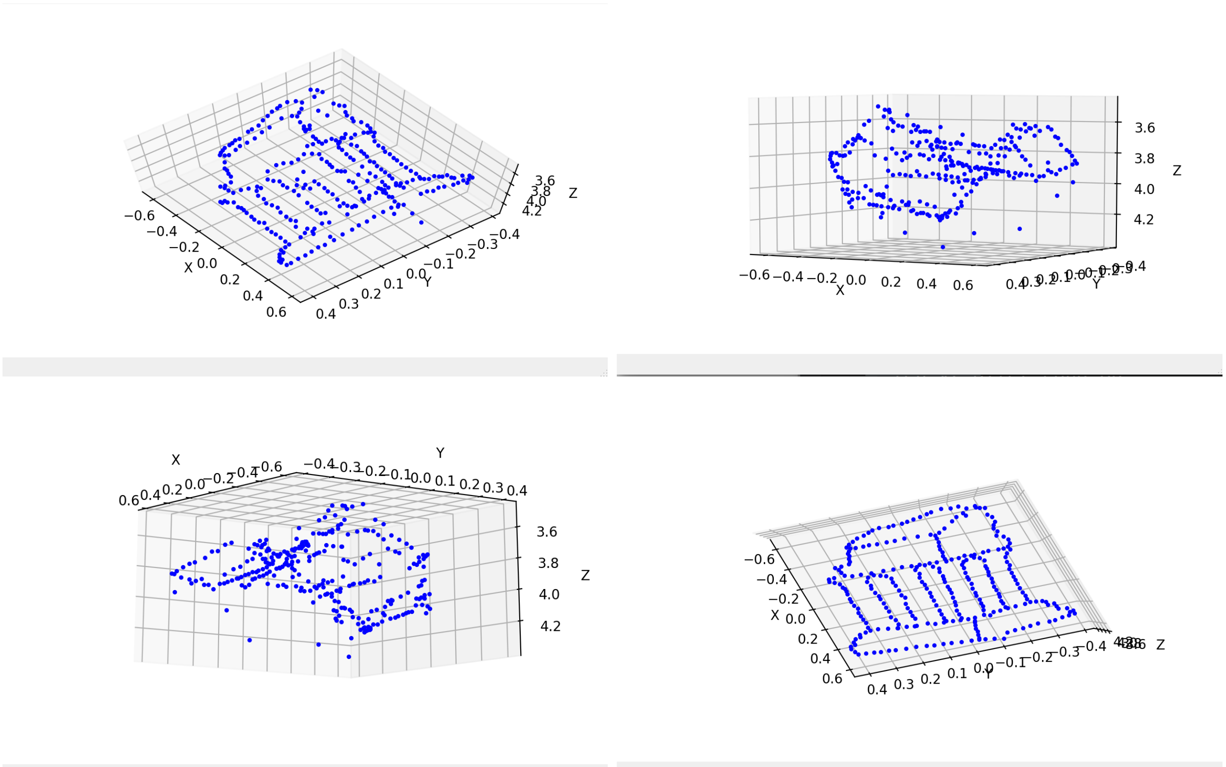
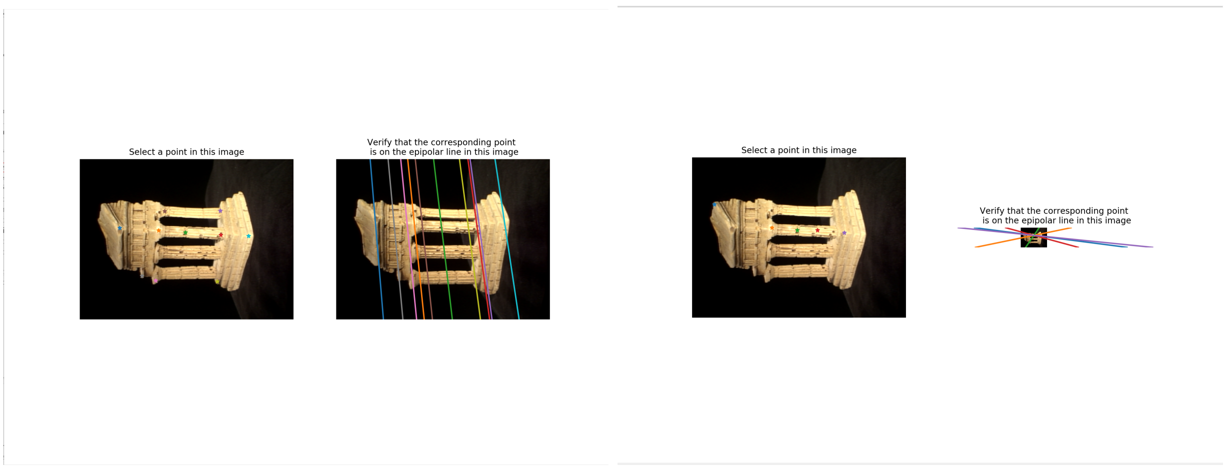


Figure 3: 3D Visualization

5 PartII - Bundle Adjustment

5.1 ransacF



(a) Ransac (b) eightpoint

Figure 4: Ransac and eightpoint methods comparison

Here I used seven-point method combining with RANSAC, since seven-point method was faster than eight-point method. At each iteration, RANSAC algorithm chooses seven random point pairs and calculate an estimate F. Then the distances between each point to its epipolar line are calculated. Those distances less than "tol" would be considered inliers. Therefore, increasing "tol" will increase the number of inlier points. Whichever F that gives out maximum number of inliers would be the result of the RANSAC algorithm. With noisy data, high tolerance will not be a good approach, because the noise in the data will influence the estimation of the Fundamental matrix. "nIters" determines the maximum number of iterations of RANSAC. Theoretically, increasing this number will result in calculating F with more combinations of random seven point pairs, which could possibly obtain a better F(not guarantee). However, higher iteration number would cost longer running time.

5.3 Optimization

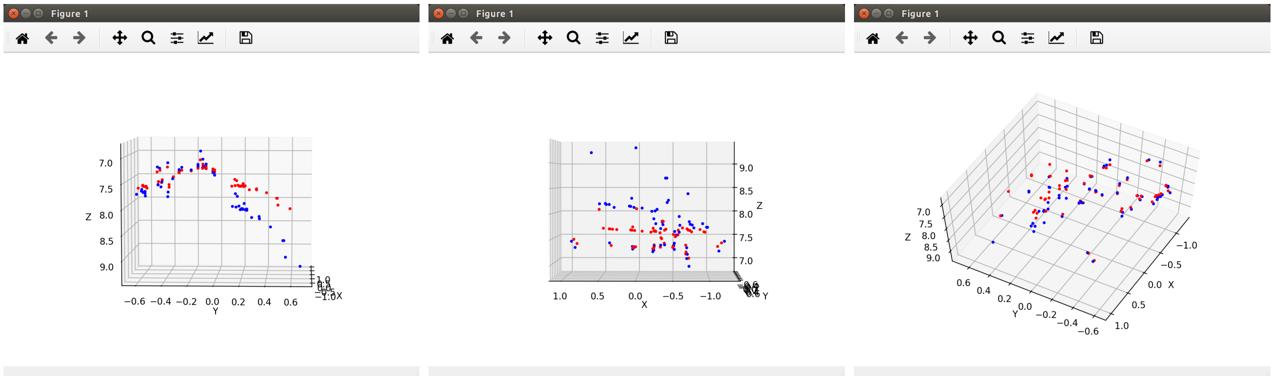


Figure 5: Bundle adjustment result, Blue: original (err = 8944.5), Red: optimize(err = 9.25)

6 PartII - Multi View Keypoint Reconstruction

6.1 Multiview Reconstruction

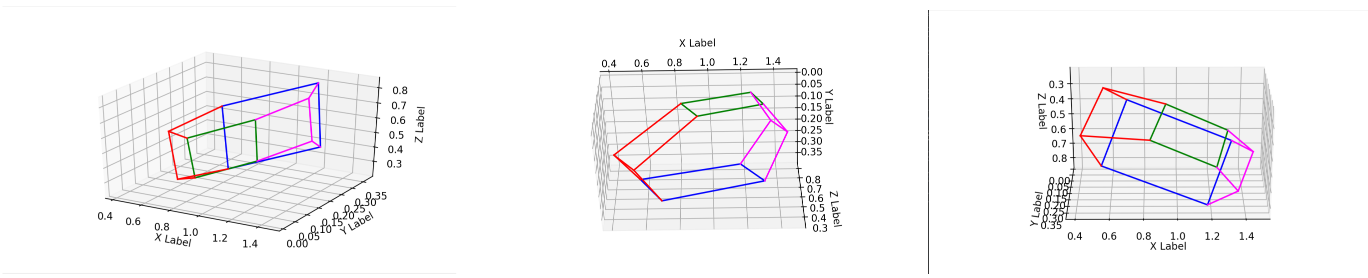
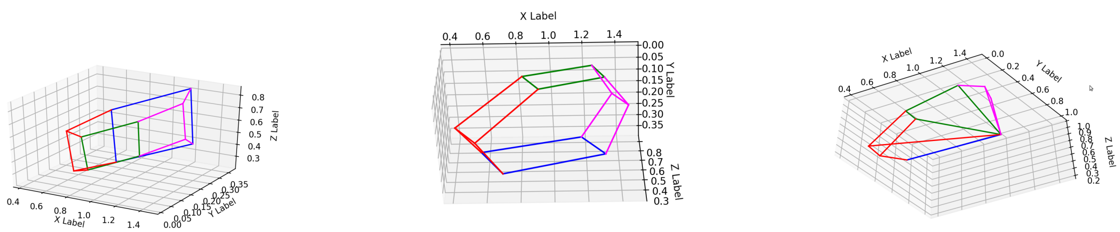


Figure 6: Multi View Keypoint Reconstruction, Err = 176.07

The method I computed the 3D locations were similar to that in the previous question using the triangulation function. The function read in points from 3 images and a threshold value, and the first thing I did was get rid of the points that were under the threshold confidence. There were four cases after this operation, all three 2D projections of a 3D point were above the confidence threshold, only two of them left, only one left, and none was left. In the first case, I chose the top two confidence images for triangulation; and for cases that were only one or no point left, it simply could not calculate the location in 3D space. With that saying, as the confidence threshold increases, less number of 3D points is possible to be calculated.



(a) thres = 100 (b) thres = 400 (c) thres = 600

Figure 7: Multi View Keypoint Reconstruction with various threshhold

In figure 7, from left to right the errors are 184.4, 176.1, 52.1. As we increases the threshold, the 3D triangulation results are improved. However, there might not be enough points(at least two) that are above the threshold confidence for triangulation. In figure 7(c), there is a point missing due to this reason.

6.2 3D trajectory

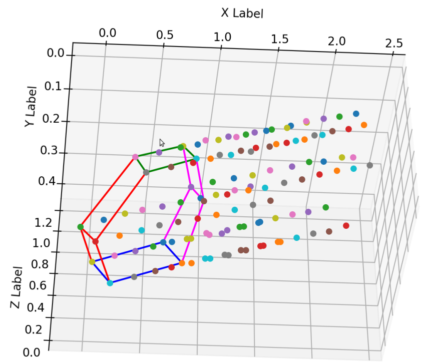


Figure 8: 3D trajectory

Here I simply plotted out the history position of all points and only connected the last 12 dots to show the shape of the object.