IC5303 Computer Vision

Fundamental matrix estimation

November 13, 2017

Student: Nguyen Van Chuong, ID: 20161199

1 Eight-point algorithm

1.1 Collecting and separating data

In this report, I use program to collect point correspondences and put it in two data file *fig1.txt* and *fig2.txt*. The data for each picture is a form of matrix of rank (696×2), which refer to 696 points of pixel (x, y). I divide it into two sets: the first set utilized to compute fundamental matrix F, and the other to validate the result.

- Corr1 and Corr2 are matrices of a form (72×2) containing 72 correspondence points for computing $\mathbf{F}$. These data points are shown in Fig.1
- Different 60 points is used to validate the result in Fig.2

Figure 1: 72 correspondence points for computing $\mathbf{F}$.

1.2 Computing fundamental matrix and drawing epipolar lines

For one pair of correspondences, $\mathbf{p} = (u \; v \; 1)^\top$ denotes for homogeneous coordinate of a point in left picture and $\mathbf{p}' = (u' \; v' \; 1)^\top$ refer to homogeneous coordinate of a point in right picture respectively. From:

$$\mathbf{Af} = \mathbf{0}, \tag{1}$$

Figure 2: 60 correspondence points for check validity of the result.

where $\mathbf{A} = \begin{bmatrix} u_1u_1' & u_1v_1' & u_1 & v_1u_1' & v_1v_1' & v_1 & u_1' & v_1' & 1 \\ u_2u_2' & u_2v_2' & u_2 & v_2u_2' & v_2v_2' & v_2 & u_2' & v_2' & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & 1 \\ u_nu_n' & u_nv_n' & u_n & v_nu_n' & v_nv_n' & v_n & u_n' & v_n' & 1 \end{bmatrix}$, $\mathbf{A} \in \mathbb{R}^{72\times 9}$ and $f = \begin{bmatrix} F_{11} & F_{12} & F_{13} & F_{21} & F_{22} & F_{23} & F_{31} & F_{32} & F_{33} \end{bmatrix}^\top \in \mathbb{R}^9$

We can use Singular Value Decomposition to compute $\mathbf{f}$, which is the last column of $\mathbf{V}$ satisfying $(\mathbf{U}, \mathbf{S}, \mathbf{V}) = \mathrm{svd}(\mathbf{A})$. Then, we take SVD to $\mathbf{F}$, and put the least eigenvalue of $\mathbf{S}$ to be 0. The fundamental matrix is finally obtained by

$$\mathbf{F} = \mathbf{U}\mathrm{diag}\{\sigma_1, \sigma_2, 0\}\mathbf{V}^\top. \tag{2}$$

The Matlab code shows result is $\mathbf{F} = \begin{bmatrix} 0 & 0 & 0.0098 \\ 0 & 0 & -0.0316 \\ -0.0119 & 0.0282 & 0.9990 \end{bmatrix}$

After scaling for $\mathbf{F}(3,3) = 1$, we can obtain $\mathbf{F} = \begin{bmatrix} 0 & 0 & 0.0098 \\ 0 & 0 & -0.0316 \\ -0.0119 & 0.0282 & 1 \end{bmatrix}$

Epipolar constraints:

$$\mathbf{p}^\top\mathbf{F}\mathbf{p}' = 0, (\mathbf{p}')^\top\mathbf{F}^\top\mathbf{p} = 0 \tag{3}$$

Therefore, the epipolar line corresponding to one point $\mathbf{p} = (u\ v\ 1)^\top$ in the left is $\mathbf{F}\mathbf{p}'$, and one point $\mathbf{p}' = (u'\ v'\ 1)^\top$ in the right is $\mathbf{F}^\top\mathbf{p}$. The result is shown in Fig.3

2 Normalized eight-point algorithm

2.1 Normalized input data

We normalize 72 input data, which is mentioned in the above section.

The mean of $rows$ = 72 input data of the left image.

$$\mu_u = \frac{1}{rows}\sum_{i\in rows} u_i,\ \mu_v = \frac{1}{rows}\sum_{i\in rows} v_i \tag{4}$$

The mean of $rows$ = 72 input data of the right image.

$$\mu_u' = \frac{1}{rows}\sum_{i\in rows} u_i',\ \mu_v' = \frac{1}{rows}\sum_{i\in rows} v_i' \tag{5}$$

Figure 3: 60 correspondence points and epipolar lines in the left and right picture for 8 points algorithm.

Standard deviation for the left image

$$\sigma_u = \sqrt{\frac{1}{rows}\sum_{i\in rows}(u_i-\mu_u)^2}, \qquad \sigma_v = \sqrt{\frac{1}{rows}\sum_{i\in rows}(v_i-\mu_v)^2} \tag{6}$$

Standard deviation for the right image

$$\sigma'_u = \sqrt{\frac{1}{rows}\sum_{i\in rows}(u'_i-\mu'_u)^2}, \qquad \sigma'_v = \sqrt{\frac{1}{rows}\sum_{i\in rows}(v'_i-\mu'_v)^2} \tag{7}$$

The transformation matrix:

$$\mathbf{T} = \begin{bmatrix} 1/\sigma_u & 0 & -\mu_u/\sigma_u \\ 0 & 1/\sigma_v & -\mu_v/\sigma_v \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

$$\mathbf{T}' = \begin{bmatrix} 1/\sigma'_u & 0 & -\mu'_u/\sigma'_u \\ 0 & 1/\sigma'_v & -\mu'_v/\sigma'_v \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

The ***normalized input*** now become:

$$u_i \rightarrow \frac{u_i-\mu_u}{\sigma_u}, \qquad v_i \rightarrow \frac{v_i-\mu_v}{\sigma_v}, \qquad u'_i \rightarrow \frac{u'_i-\mu'_u}{\sigma'_u}, \qquad v'_i \rightarrow \frac{v'_i-\mu_v}{\sigma_v} \tag{10}$$

2.2 Computing fundamental matrix and drawing epipolar lines

We follow similar 8-point algorithm steps to compute fundamental matrix, with normalized inputs. However, after obtaining $\mathbf{F}$, we transform it to original coordinates.

$$\mathbf{F} = \mathbf{T}^\top \mathbf{F}\mathbf{T}' \tag{11}$$

The Matlab code shows result is $\mathbf{F} = \begin{bmatrix} 0 & 0 & -0.0005 \\ 0 & 0 & -0.0150 \\ 0.0005 & 0.0152 & 0.0752 \end{bmatrix}$

After scaling for $\mathbf{F}(3,3) = 1$, we can obtain $\mathbf{F} = \begin{bmatrix} 0 & 0 & 0.0098 \\ 0 & 0 & -0.0316 \\ -0.0119 & 0.0282 & 1 \end{bmatrix}$

Similarly, we obtain the epipolar line corresponding to one point $\mathbf{p} = (u\ v\ 1)^\top$ in the left is $\mathbf{Fp'}$, and one point $\mathbf{p'} = (u'\ v'\ 1)^\top$ in the right is $\mathbf{F}^\top\mathbf{p}$. The result is shown in Fig.4

Figure 4: 60 correspondence points and epipolar lines in the left and right picture for normalized-8 points algorithm.

Conclusion: The result in normalized 8-point algorithm is much better. In 8 point algorithm, depending on the choice of correspondence point, the epipolar lines can be converge at one point, especially in the case if we don't choose the point locating uniformly in the picture. But the normalized 8-point provides more robustness to errors, which is more suitable in practice.