AI3604 Computer Vision: Homework 2

Xiangyuan Xue (521030910387)

Written Assignment

1. (a) For clearness, vectors are denoted by lowercase letters and matrices are denoted by uppercase letters. Then the error function can be written as

$$\mathcal{L}(\boldsymbol{A},\boldsymbol{t})=\sum_{i=1}^{n}\left(\boldsymbol{y}_i-\boldsymbol{A}\boldsymbol{x}_i-\boldsymbol{t}\right)^T\left(\boldsymbol{y}_i-\boldsymbol{A}\boldsymbol{x}_i-\boldsymbol{t}\right)$$

Let the partial derivatives be zero, we have

$$\begin{cases}\dfrac{\partial\mathcal{L}}{\partial\boldsymbol{A}}=\displaystyle\sum_{i=1}^{n}2\left(\boldsymbol{A}\boldsymbol{x}_i+\boldsymbol{t}-\boldsymbol{y}_i\right)\boldsymbol{x}_i^T=\boldsymbol{O}\\ \dfrac{\partial\mathcal{L}}{\partial\boldsymbol{t}}=\displaystyle\sum_{i=1}^{n}2\left(\boldsymbol{A}\boldsymbol{x}_i+\boldsymbol{t}-\boldsymbol{y}_i\right)=\boldsymbol{o}\end{cases}$$

The latter equation indicates that

$$\boldsymbol{t}=\frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{y}_i-\boldsymbol{A}\boldsymbol{x}_i\right)=\bar{\boldsymbol{y}}-\boldsymbol{A}\bar{\boldsymbol{x}}$$

Substitute it into the former equation, we have

$$\sum_{i=1}^{n}\left(\boldsymbol{A}\boldsymbol{x}_i-\boldsymbol{A}\bar{\boldsymbol{x}}+\bar{\boldsymbol{y}}-\boldsymbol{y}_i\right)\boldsymbol{x}_i^T=\boldsymbol{O}$$

which is equivalent to

$$\boldsymbol{A}\left(\sum_{i=1}^{n}\left(\boldsymbol{x}_i-\bar{\boldsymbol{x}}\right)\boldsymbol{x}_i^T\right)=\sum_{i=1}^{n}\left(\boldsymbol{y}_i-\bar{\boldsymbol{y}}\right)\boldsymbol{x}_i^T$$

Note that

$$\sum_{i=1}^{n}\left(\boldsymbol{x}_i-\bar{\boldsymbol{x}}\right)\boldsymbol{x}_i^T=\sum_{i=1}^{n}\left(\boldsymbol{x}_i-\bar{\boldsymbol{x}}\right)\left(\boldsymbol{x}_i-\bar{\boldsymbol{x}}\right)^T=\boldsymbol{X}\boldsymbol{X}^T$$
$$\sum_{i=1}^{n}\left(\boldsymbol{y}_i-\bar{\boldsymbol{y}}\right)\boldsymbol{x}_i^T=\sum_{i=1}^{n}\left(\boldsymbol{y}_i-\bar{\boldsymbol{y}}\right)\left(\boldsymbol{x}_i-\bar{\boldsymbol{x}}\right)^T=\boldsymbol{Y}\boldsymbol{X}^T$$

Then the equation can be written as

$$\boldsymbol{A}\boldsymbol{X}\boldsymbol{X}^T=\boldsymbol{Y}\boldsymbol{X}^T$$

which yields

$$\begin{cases}\boldsymbol{A}^*=\left(\boldsymbol{Y}\boldsymbol{X}^T\right)\left(\boldsymbol{X}\boldsymbol{X}^T\right)^{-1}\\ \boldsymbol{t}^*=\bar{\boldsymbol{y}}-\boldsymbol{A}^*\bar{\boldsymbol{x}}\end{cases}$$

(b) Since $\boldsymbol{A} \in \mathbb{R}^{3\times 3}$, $\boldsymbol{t} \in \mathbb{R}^3$, there are $9 + 3 = 12$ unknown parameters in total. For each transormation pair $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, we have

$$\boldsymbol{y}_i = \boldsymbol{A}\boldsymbol{x}_i + \boldsymbol{t}$$

which provides 3 constraints. To make the solution unique, at least 4 correspondences are required. For more than 4 correspondences, least square method can be applied to eliminate the negative influence of noise $\boldsymbol{e}_i$.

Programming Assignment

1. (a) First, we should read the chessboard images, where each chessboard contains 31×23 inner corners. Here we apply the function `cv2.findChessboardCorners` to detect the chessboard corners. It is difficult to find all the corners correctly, so we enable adaptive thresholding to obtain as many views as possible.

   (b) For each view, we need to solve the projection matrix $\boldsymbol{H}$ by the relationship that

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}}_{\boldsymbol{H}} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

   Rearrange and stack all the equations, we have

$$\underbrace{\begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -u_1x_1 & -u_1y_1 & -u_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -v_1x_1 & -v_1y_1 & -v_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -u_2x_2 & -u_2y_2 & -u_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -v_2x_2 & -v_2y_2 & -v_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & 1 & 0 & 0 & 0 & -u_nx_n & -u_ny_n & -u_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -v_nx_n & -v_ny_n & -v_n \end{pmatrix}}_{\boldsymbol{A}} \underbrace{\begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{pmatrix}}_{\boldsymbol{h}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}$$

   Then we can solve the optimization problem that

$$\min_{\boldsymbol{h}} \|\boldsymbol{A}\boldsymbol{h}\|^2 \quad \text{s.t.} \quad \|\boldsymbol{h}\| = 1$$

   The solution $\boldsymbol{h}^*$ is the eigenvector corresponding to the smallest eigenvalue of $\boldsymbol{A}^T\boldsymbol{A}$. To minimize the impact of distortion parameters, we only use 300 image points in the central region of the image.

   (c) Since $\boldsymbol{B} = \boldsymbol{K}^{-T}\boldsymbol{K}^{-1}$ is symmetric, we might as well assume that

$$\boldsymbol{B} = \begin{pmatrix} b_1 & b_2 & b_3 \\ b_2 & b_4 & b_5 \\ b_3 & b_5 & b_6 \end{pmatrix}$$

Notice it holds that

$$\begin{cases} \boldsymbol{h}_1^T \boldsymbol{B} \boldsymbol{h}_1 - \boldsymbol{h}_2^T \boldsymbol{B} \boldsymbol{h}_2 = 0 \\ \boldsymbol{h}_1^T \boldsymbol{B} \boldsymbol{h}_2 = 0 \end{cases}$$

Rearrange and stack all the equations, we have

$$\underbrace{\begin{pmatrix} g_{11}^{(1)} & g_{12}^{(1)} & g_{13}^{(1)} & g_{14}^{(1)} & g_{15}^{(1)} & g_{16}^{(1)} \\ g_{11}^{(1)} & g_{22}^{(1)} & g_{23}^{(1)} & g_{24}^{(1)} & g_{25}^{(1)} & g_{26}^{(1)} \\ g_{11}^{(2)} & g_{12}^{(2)} & g_{13}^{(2)} & g_{14}^{(2)} & g_{15}^{(2)} & g_{16}^{(2)} \\ g_{11}^{(2)} & g_{22}^{(2)} & g_{23}^{(2)} & g_{24}^{(2)} & g_{25}^{(2)} & g_{26}^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ g_{11}^{(m)} & g_{12}^{(m)} & g_{13}^{(m)} & g_{14}^{(m)} & g_{15}^{(m)} & g_{16}^{(m)} \\ g_{11}^{(m)} & g_{22}^{(m)} & g_{23}^{(m)} & g_{24}^{(m)} & g_{25}^{(m)} & g_{26}^{(m)} \end{pmatrix}}_{\boldsymbol{G}} \underbrace{\begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{pmatrix}}_{\boldsymbol{b}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}$$

The detailed values are described as

$$\begin{cases} g_{11}^{(i)} = h_{11}^{(i)} h_{11}^{(i)} - h_{12}^{(i)} h_{12}^{(i)} \\ g_{12}^{(i)} = 2(h_{11}^{(i)} h_{21}^{(i)} - h_{12}^{(i)} h_{22}^{(i)}) \\ g_{13}^{(i)} = 2(h_{11}^{(i)} h_{31}^{(i)} - h_{12}^{(i)} h_{32}^{(i)}) \\ g_{14}^{(i)} = h_{21}^{(i)} h_{21}^{(i)} - h_{22}^{(i)} h_{22}^{(i)} \\ g_{15}^{(i)} = 2(h_{21}^{(i)} h_{31}^{(i)} - h_{22}^{(i)} h_{32}^{(i)}) \\ g_{16}^{(i)} = h_{31}^{(i)} h_{31}^{(i)} - h_{32}^{(i)} h_{32}^{(i)} \end{cases} \qquad \begin{cases} g_{21}^{(i)} = h_{11}^{(i)} h_{12}^{(i)} \\ g_{22}^{(i)} = h_{11}^{(i)} h_{22}^{(i)} + h_{12}^{(i)} h_{21}^{(i)} \\ g_{23}^{(i)} = h_{11}^{(i)} h_{32}^{(i)} + h_{12}^{(i)} h_{31}^{(i)} \\ g_{24}^{(i)} = h_{21}^{(i)} h_{12}^{(i)} \\ g_{25}^{(i)} = h_{21}^{(i)} h_{32}^{(i)} + h_{22}^{(i)} h_{31}^{(i)} \\ g_{26}^{(i)} = h_{31}^{(i)} h_{32}^{(i)} \end{cases}$$

Then we can solve the optimization problem that

$$\min_{\boldsymbol{b}} \|\boldsymbol{G}\boldsymbol{b}\|^2 \quad \text{s.t.} \quad \|\boldsymbol{b}\| = 1$$

The solution $\boldsymbol{b}^*$ is the eigenvector corresponding to the smallest eigenvalue of $\boldsymbol{G}^T\boldsymbol{G}$.

(d) Since $\boldsymbol{B} = \boldsymbol{K}^{-T}\boldsymbol{K}^{-1}$, $\boldsymbol{K}^{-1}$ can be solved by Cholesky decomposition. Then the intrinsic matrix $\boldsymbol{K}$ can be obtained by matrix inversion. According to homogeneous representation, $\boldsymbol{K}$ should be normalized by dividing k_{33}, namely

$$\boldsymbol{K} = \begin{pmatrix} f_x & s & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{pmatrix}$$

where (f_x, f_y) describes the focal length and (o_x, o_y) describes the principal point.

(e) Since we have figured out the projection matrices $\left\{\boldsymbol{H}^{(i)}\right\}_{i=1}^{m}$, the projected points can be easily obtained by the relationship that

$$\begin{pmatrix} u_j^{(i)} \\ v_j^{(i)} \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} h_{11}^{(i)} & h_{12}^{(i)} & h_{13}^{(i)} \\ h_{21}^{(i)} & h_{22}^{(i)} & h_{23}^{(i)} \\ h_{31}^{(i)} & h_{32}^{(i)} & h_{33}^{(i)} \end{pmatrix}}_{\boldsymbol{H}^{(i)}} \begin{pmatrix} x_j^{(i)} \\ y_j^{(i)} \\ 1 \end{pmatrix}$$

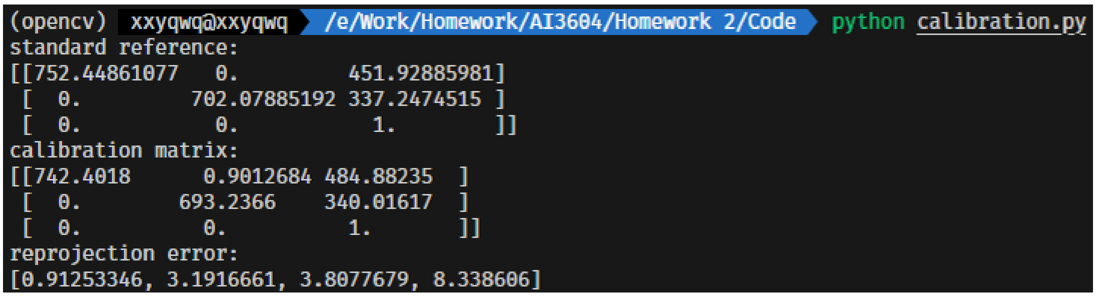
For simplicity, we rewrite the equation in matrix form that

$$\boldsymbol{u}_j^{(i)} = \boldsymbol{H}^{(i)} \boldsymbol{x}_j^{(i)}$$

Then the reprojection error corresponding to the i-th view should be

$$\mathcal{E}_i = \frac{1}{n} \sum_{j=1}^{n} \left\| \boldsymbol{u}_j^{(i)} - \boldsymbol{H}^{(i)} \boldsymbol{x}_j^{(i)} \right\|^2$$

(f) All the implementation details are described in the previous parts. We apply the function `cv2.calibrateCamera` to obtain the standard reference. The calibration matrix and the reprojection error are shown in the figure below.



```
(opencv) xxyqwq@xxyqwq /e/Work/Homework/AI3604/Homework 2/Code python calibration.py
standard reference:
[[752.44861077   0.         451.92885981]
 [  0.         702.07885192 337.2474515 ]
 [  0.           0.           1.        ]]
calibration matrix:
[[742.4018      0.9012684 484.88235  ]
 [  0.        693.2366    340.01617  ]
 [  0.          0.          1.       ]]
reprojection error:
[0.91253346, 3.1916661, 3.8077679, 8.338606]
```

Figure 1: calibration result

We can conclude that the focal length $(f_x, f_y) \approx (742, 693)$ and the principal point $(o_x, o_y) \approx (485, 340)$, which are quite close to the standard reference. 4 different views are used for calibration. Their reprojection errors are around 0.9, 3.2, 3.8 and 8.3 respectively, indicating that the estimation is relatively accurate.

Note that the results are scaled by the size of the chessboard square because we use $(0, 0), (1, 0), (2, 0), \ldots, (w, h)$ as the world coordinates for simplification.

Appendix

To reproduce the results, run the following command in terminal.

```
$ python calibration.py
```

The structure of the source code is slightly modified for the sake of elegance.