Computer Vision Assignment 1

by 0856609王心瑩, 0856167許振揚, 0856170連家彤

Introduction

Camera calibration is a necessary step in 3D computer vision in order to extract metric information from 2D images. The goal of calibration is to find the effective projection transform hence yielding significant information regarding the vision system such as focal lengths, camera pose, camera center, etc.

The aim of this project is to implement the camera calibration function from scratch without using the function provided by OpenCV. In this project, there are two files “camera_calibration.py” and “feature.py”. The former is the main program and the latter contain a few functions being used to solve essential metric information like homography matrix, intrinsic matrix, and extrinsic matrix. To solve those information, we need to apply some mathematical approaches, and the details will be explained in following sections.

Implementation Procedure

There are three steps in our implementation,

1. Calculate Homography of each images
2. Calculate intrinsic matrix K with homography
3. Calculate extrinsic matrix for each images

1. Calculate Homography of each images

We chose to estimate Homography *H* with the Direct Linear Transformation. Since by definition,

$$\dot{\underline{\boldsymbol{u}}}_j = \mathbf{H} \cdot \underline{\boldsymbol{X}}_j \quad \text{or} \quad \begin{pmatrix} \dot{\underline{u}}_j \\ \dot{\underline{v}}_j \\ \dot{\underline{w}}_j \end{pmatrix} = \begin{pmatrix} H_{0,0} & H_{0,1} & H_{0,2} \\ H_{1,0} & H_{1,1} & H_{1,2} \\ H_{2,0} & H_{2,1} & H_{2,2} \end{pmatrix} \cdot \begin{pmatrix} X_j \\ Y_j \\ 1 \end{pmatrix}.$$

it could be written into a pair of equation,

$$u_j = \frac{\dot{\underline{u}}_j}{\dot{\underline{w}}_j} = \frac{H_{0,0} \cdot X_j + H_{0,1} \cdot Y_j + H_{0,2}}{H_{2,0} \cdot X_j + H_{2,1} \cdot Y_j + H_{2,2}},$$
$$v_j = \frac{\dot{\underline{v}}_j}{\dot{\underline{w}}_j} = \frac{H_{1,0} \cdot X_j + H_{1,1} \cdot Y_j + H_{1,2}}{H_{2,0} \cdot X_j + H_{2,1} \cdot Y_j + H_{2,2}},$$

and by rearrangement,

$$\dot{u}_j \cdot X_j \cdot H_{2,0} + \dot{u}_j \cdot Y_j \cdot H_{2,1} + \dot{u}_j \cdot H_{2,2} - H_{0,0} \cdot X_j - H_{0,1} \cdot Y_j - H_{0,2} = 0,$$
$$\dot{v}_j \cdot X_j \cdot H_{2,0} + \dot{v}_j \cdot Y_j \cdot H_{2,1} + \dot{v}_j \cdot H_{2,2} - H_{1,0} \cdot X_j - H_{1,1} \cdot Y_j - H_{1,2} = 0.$$

also it could be written into form for each point *i*,

$$\begin{pmatrix} -X_j & -Y_j & -1 & 0 & 0 & 0 & \dot{u}_j X_j & \dot{u}_j Y_j & \dot{u}_j \\ 0 & 0 & 0 & -X_j & -Y_j & -1 & \dot{v}_j X_j & \dot{v}_j Y_j & \dot{v}_j \end{pmatrix} \cdot \boldsymbol{h} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

where $h = \left(H_{0,0}, H_{0,1}, H_{0,2}, H_{1,0}, H_{1,1}, H_{1,2}, H_{2,0}, H_{2,1}, H_{2,2}\right)^T$

Since there are N points, and assume they are related by the same homography H, we could yield the equation into a form A (a $2N$x9 matrix). In our case, N=49. This equation could be solve by singular-value decomposition (SVD), where $Ah = 0$. By SVD, A will be separated into 3 product metrices U, S, V, where S is the singular value and indicate the "goodness of fit" of the corresponding homography metric. As the result, h will be pick from the row of V with the smallest s_k.

$$\begin{pmatrix} -X_0 & -Y_0 & -1 & 0 & 0 & 0 & \dot{u}_0 X_0 & \dot{u}_0 Y_0 & \dot{u}_0 \\ 0 & 0 & 0 & -X_0 & -Y_0 & -1 & \dot{v}_0 X_0 & \dot{v}_0 Y_0 & \dot{v}_0 \\ \hline -X_1 & -Y_1 & -1 & 0 & 0 & 0 & \dot{u}_1 X_1 & \dot{u}_1 Y_1 & \dot{u}_1 \\ 0 & 0 & 0 & -X_1 & -Y_1 & -1 & \dot{v}_1 X_1 & \dot{v}_1 Y_1 & \dot{v}_1 \\ \hline -X_2 & -Y_2 & -1 & 0 & 0 & 0 & \dot{u}_2 X_2 & \dot{u}_2 Y_2 & \dot{u}_2 \\ 0 & 0 & 0 & -X_2 & -Y_2 & -1 & \dot{v}_2 X_2 & \dot{v}_2 Y_2 & \dot{v}_2 \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline -X_{N-1} & -Y_{N-1} & -1 & 0 & 0 & 0 & \dot{u}_{N-1} X_{N-1} & \dot{u}_{N-1} Y_{N-1} & \dot{u}_{N-1} \\ 0 & 0 & 0 & -X_{N-1} & -Y_{N-1} & -1 & \dot{v}_{N-1} X_{N-1} & \dot{v}_{N-1} Y_{N-1} & \dot{v}_{N-1} \end{pmatrix} \cdot \begin{pmatrix} H_{0,0} \\ H_{0,1} \\ H_{0,2} \\ H_{1,0} \\ H_{1,1} \\ H_{1,2} \\ H_{2,0} \\ H_{2,1} \\ H_{2,2} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \hline 0 \\ 0 \\ \hline 0 \\ 0 \\ \hline \vdots \\ \hline 0 \\ 0 \end{pmatrix}$$

2. Calculate intrinsic matrix K with homography

With

$$H = \left(h_0, h_1, h_2\right) = K\left(r_0, r_1, t\right), \text{ and } B = K^{-T}K^{-1} = \begin{bmatrix} B_0 & B_1 & B_3 \\ B_1 & B_2 & B_4 \\ B_3 & B_4 & B_5 \end{bmatrix}$$

by definition B could be solve by

$$Vb = 0 = \begin{pmatrix} v_{0,1}(H) \\ v_{0,0}(H) - v_{1,1}(H) \end{pmatrix} \cdot b = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

where

$$b = \left(B_0, B_1, B_2, B_3, B_4, B_5\right) \text{ and } V_{p,q}(H) = \begin{pmatrix} H_{0,p} \cdot H_{0,q} \\ H_{0,p} \cdot H_{1,q} + H_{1,p} \cdot H_{0,q} \\ H_{1,p} \cdot H_{1,q} \\ H_{2,p} \cdot H_{0,q} + H_{0,p} \cdot H_{2,q} \\ H_{2,p} \cdot H_{1,q} + H_{1,p} \cdot H_{2,q} \\ H_{2,p} \cdot H_{2,q} \end{pmatrix}^T$$

We could solve equation $Vb = 0$ by SVD as we do so when solving homography.

After we get B by SVD, we could solve K by Cholesky Decomposition. Since B will needed to be positive definite according to Cholesky Decomposition, we check if B_0, B_2, B_5 are bigger than zero, if not then let $B = -B$. After solving K by Cholesky Decomposition, we do normalization and get the final K.

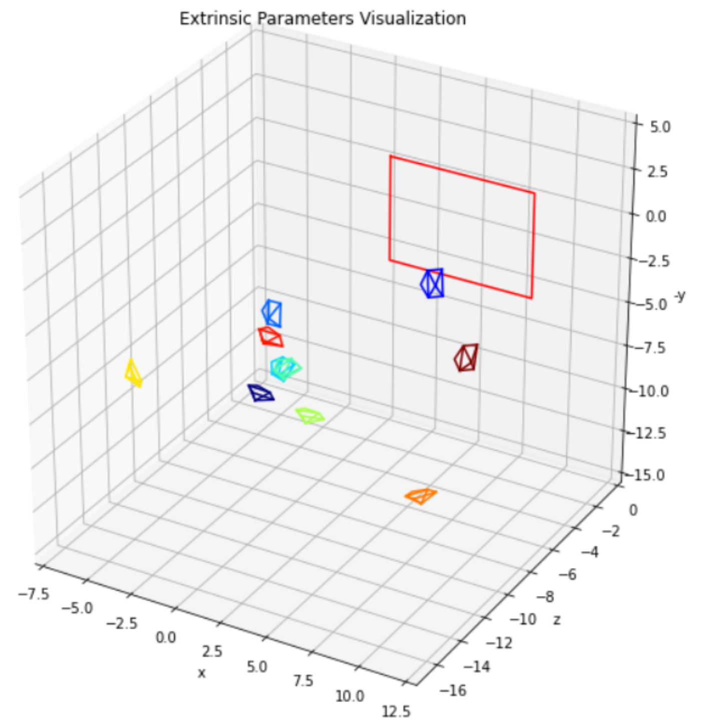
3. Calculate extrinsic matrix for each images

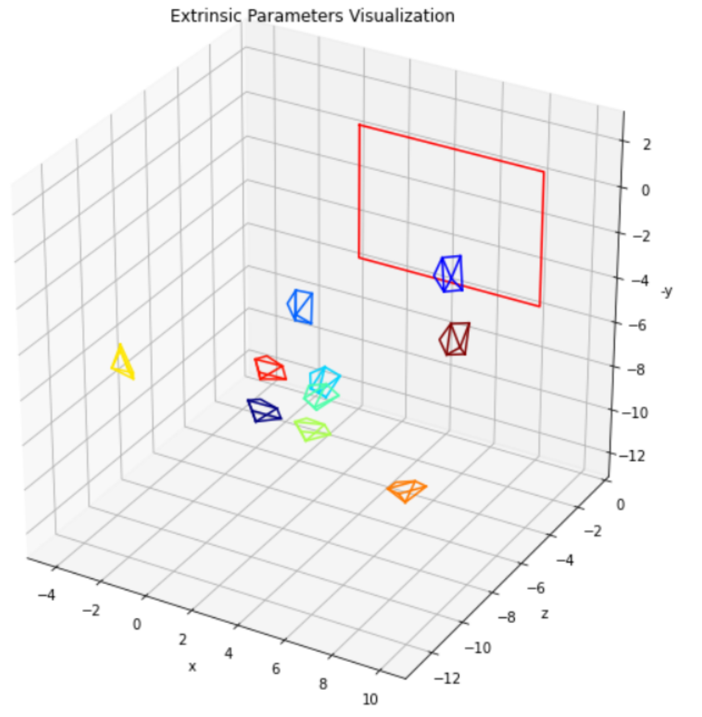
After getting the intrinsic matrix, we could calculate the entrinsic matrix of each image.By the definition,

$$\lambda = 1/||\mathbf{K}^{-1}\mathbf{h}_1|| \quad \text{and} \quad \begin{aligned} \mathbf{r}_1 &= \lambda\mathbf{K}^{-1}\mathbf{h}_1 \\ \mathbf{r}_2 &= \lambda\mathbf{K}^{-1}\mathbf{h}_2 \\ \mathbf{r}_3 &= \mathbf{r}_1 \times \mathbf{r}_2 \\ \mathbf{t} &= \lambda\mathbf{K}^{-1}\mathbf{h}_3 \end{aligned}$$

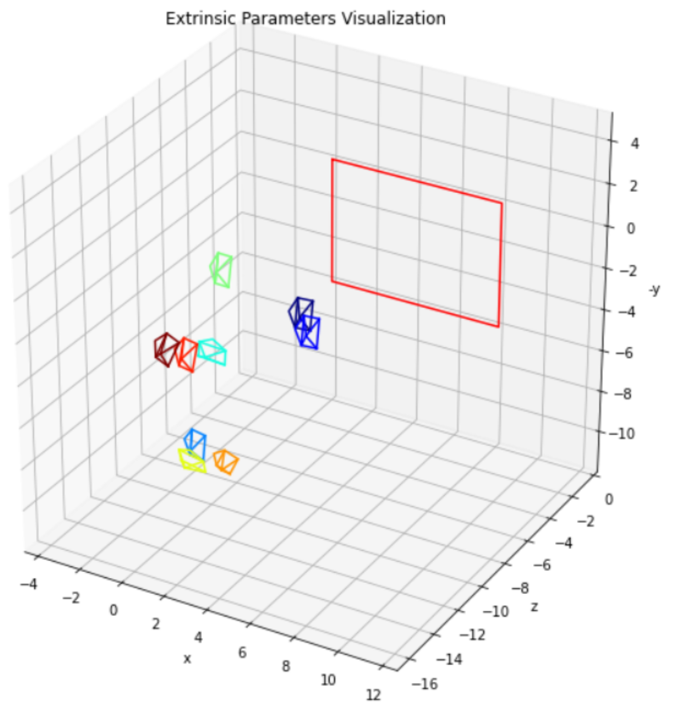
Experimental Result

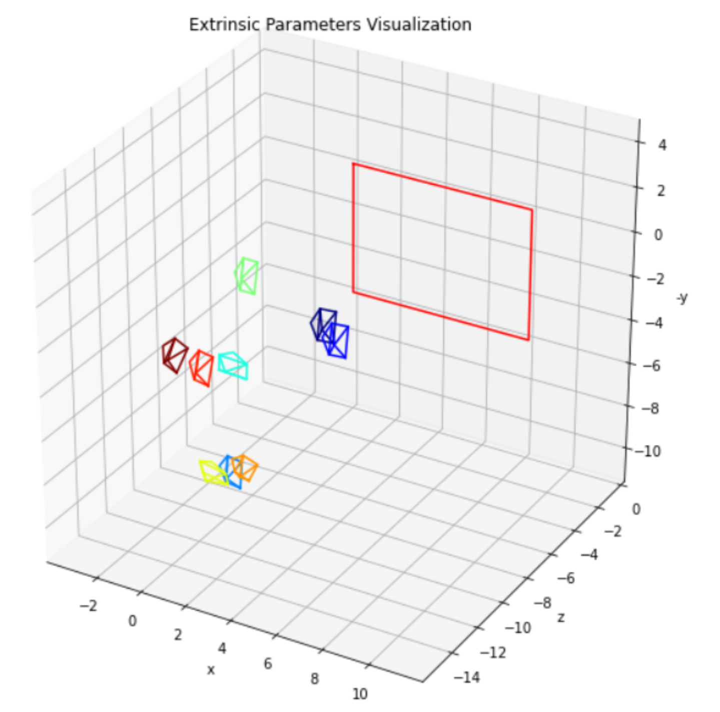
1. with images provided

by our method	

by cv2.calibrateCamera	

2. with our own images

by our method	

by cv2.calibrateCamera	

Discussion

During implementing the camera calibration from scratch, we come across a few mathematical problems that need resolving. The paper and online resources we've surveyed to solve these problems will be listed in references section.

1. How to solve the Homography matrix by world points and image points for each view?
   An essential part of the estimating homographies[1] is to obtain a solution using Direct Linear transformation and then get the result by SVD.
2. How to build the matrix **V** where **Vb = 0** from Homogrphies matrix?
   We can obtain the matrix **V** from some mathematical derivation[2][p.18].
3. When we find out **b** from **Vb = 0**, and build the matrix **B** from **b**, in some situation, Cholesky Decomposition can't be apply to **B** in order to get the intrinsic matrix K.
   The reason why **B** can't be factorized by Cholesky Decomposition is that **B** is probably not positive definite in some case. So we need to do some tricks to adjust **B** to be positive definite[2][Algorithm 4.4 or Algorithm 4.5].

Conclusion

The difference in the intrinsic matrix between our solution and the one solved by cv2.calibrateCamera is probably the main reason that make our final result different, which leads to different value in extrinsic matrixes. We have tried different strategies to make the value of intrinsic matrix as near as possible to the intrinsic matrix that receive by cv2.calibrateCamera. We have tried, such as, picking 3 homographies from 10 homographies to construct the matrix V, instead of using all of the homographies to solve for the intrinsic matrix, and go through every possible combination of homographies to find out the most suitable one. However, the result of picking only 3 from 10, are not as good as using all of the 10 homographies. As the result, we chose to use the intrinsic matrix that solved with the V which contains all 10 homographies. The following are the result of our intrinsic matrix compared with the one received from cv2.calibrateCamera.

intrinsic - NumPy array

	0	1	2
0	3436.97	0	1512.24
1	-0	3383.4	1381.36
2	-0	0	1

Intrinsic Matrix solved by our method

mtx - NumPy array

	0	1	2
0	2701.93	0	1538.21
1	0	2738.09	1960.13
2	0	0	1

Intrinsic Matrix solved by cv2.calibrationCamera

There might still be some space for us to improve. By using other methods, such as distortion or other methods that we don't know yet, we might get a better result. We hope that during this course and the assignments, we could have the opportunities to learn more and improve our result.

Work assignment plan between team members

We work for this homework together. First, we discussed the details of the assignment together. Then we developed our own programs individually. When encountering problems during the coding process, we posted out the question and found out the solution together. Finally, we integrated everyone's content.

Reference

[1] David Kriegman, *Homography Estimation*, Computer Vision CSE 252A, Winter 2007, https://cseweb.ucsd.edu/classes/wi07/cse252a/homography_estimation/homography_estimation.pdf

[2] Wilhelm Burger, *Zhang's Camera Calibration Algorithm: In-Depth Tutorial and Implementation*, University of Applied Sciences Upper Austria, School of Informatics, Communications and Media, Hagenberg, Austria, 2016, http://staff.fh-hagenberg.at/burger/publications/reports/2016Calibration/Burger-CameraCalibration-20160516.pdf