Camera Calibration

0856144 邱賢祐
0856095 黃柏豪
0856148 陳奕遠

I. Introduction

Capturing views and turning it into a digital image is a series of processes. Basically, it can be described with two parts, extrinsic and intrinsic parameters. Extrinsic parameters describe a transformation from 3D global coordinate to coordinate of perspective of a camera. Intrinsic parameters describe transformations from a 2D image to digital one. Ideally, we can purely describe the transformation with these two matrices, however, distortions may be involved such as lens distortion and tangential distortion. Zhengyou, Zhang proposed an approach that makes use of a planar pattern, chessboard, in an image to estimate values close to these two types of parameters. Generally, the method consists of 5 steps: 1. To take multiple pictures of a chessboard at different positions and angles. 2. To locate corners of chessboards in images respectively. 3. Estimating homography matrix by each correspondence of corners between a picture and its global coordinate. 4. To decompose the homography matrices into extrinsic and intrinsic parameters. 5. To calibrate the images. In this report, we reproduce Zhang's method and present evaluation of our work.

The rest of this report is organized as follows: Firstly, we sketch the background knowledge and illustrate the procedure of our implementation. Secondly, we present experimental results. Discussion and conclusion will be present in the last.

II. Implementation Procedure

In order to implement camera calibration, we follow the steps of the paper [1]. This paper proposed a more convenient way to compute the intrinsic and the extrinsic matrix of the camera from a chessboard in the image. Here are the steps:

1. Estimate homography of each image with Direct Linear Transformation (DLT).

$$\begin{pmatrix} -X_0 & -Y_0 & -1 & 0 & 0 & 0 & \dot{u}_0X_0 & \dot{u}_0Y_0 & \dot{u}_0 \\ 0 & 0 & 0 & -X_0 & -Y_0 & -1 & \dot{v}_0X_0 & \dot{v}_0Y_0 & \dot{v}_0 \\ \hdashline -X_1 & -Y_1 & -1 & 0 & 0 & 0 & \dot{u}_1X_1 & \dot{u}_1Y_1 & \dot{u}_1 \\ 0 & 0 & 0 & -X_1 & -Y_1 & -1 & \dot{v}_1X_1 & \dot{v}_1Y_1 & \dot{v}_1 \\ \hdashline -X_2 & -Y_2 & -1 & 0 & 0 & 0 & \dot{u}_2X_2 & \dot{u}_2Y_2 & \dot{u}_2 \\ 0 & 0 & 0 & -X_2 & -Y_2 & -1 & \dot{v}_2X_2 & \dot{v}_2Y_2 & \dot{v}_2 \\ \hdashline \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hdashline -X_{N-1} & -Y_{N-1} & -1 & 0 & 0 & 0 & \dot{u}_{N-1}X_{N-1} & \dot{u}_{N-1}Y_{N-1} & \dot{u}_{N-1} \\ 0 & 0 & 0 & -X_{N-1} & -Y_{N-1} & -1 & \dot{v}_{N-1}X_{N-1} & \dot{v}_{N-1}Y_{N-1} & \dot{v}_{N-1} \end{pmatrix} \cdot \begin{pmatrix} H_{0,0} \\ H_{0,1} \\ H_{0,2} \\ H_{1,0} \\ H_{1,1} \\ H_{1,2} \\ H_{2,0} \\ H_{2,1} \\ H_{2,2} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \hdashline 0 \\ 0 \\ \hdashline 0 \\ 0 \\ \hdashline \vdots \\ \hdashline 0 \\ 0 \end{pmatrix}$$

The equation can be solved by singular value decomposition. The optimal solution is the eigenvector (the last column of the V matrix in SVD) with respect to smallest eigenvalue (singular value).

2. Compute matrix B (B = K^-T * K^-1, K is the intrinsic matrix of the camera) by matrix V. The matrix V is:

$$\begin{bmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{bmatrix} \mathbf{b} = \mathbf{0} \,. \quad \mathbf{v}_{ij} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]^T \,.$$

and b is $\mathbf{b} = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T \,.$

And then use the same method to solve this homogeneous system of linear equations.

3. Use Cholesky decomposition to factorize intrinsic matrix K from B. But if B is not positive definite, we can use the close form to compute K.

$$\mathbf{B} = \mathbf{A}^{-T}\mathbf{A}^{-1} \equiv \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2\beta} & \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} \\ -\frac{\gamma}{\alpha^2\beta} & \frac{\gamma^2}{\alpha^2\beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0\gamma - u_0\beta)^2}{\alpha^2\beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix}$$

$$v_0 = (B_{12}B_{13} - B_{11}B_{23})/(B_{11}B_{22} - B_{12}^2)$$
$$\lambda = B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})]/B_{11}$$
$$\alpha = \sqrt{\lambda / B_{11}}$$
$$\beta = \sqrt{\lambda B_{11}/(B_{11}B_{22} - B_{12}^2)}$$
$$\gamma = -B_{12}\alpha^2\beta/\lambda$$
$$u_0 = \gamma v_0/\beta - B_{13}\alpha^2/\lambda \,.$$

4. Finally, we can compute extrinsic matrix by intrinsic matrix and homography.

$$\mathbf{r}_1 = \lambda \mathbf{A}^{-1}\mathbf{h}_1$$
$$\mathbf{r}_2 = \lambda \mathbf{A}^{-1}\mathbf{h}_2$$
$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$$
$$\mathbf{t} = \lambda \mathbf{A}^{-1}\mathbf{h}_3$$

III. Experiment Result

Firstly, we present sample/own data. Then, we will present the results by replacing the OpenCV function that returns intrinsic and extrinsic matrices with our own implementation mentioned above. To assure if the implementation is valid, we test the work with our own pictures. The result with our own pictures will be presented after the first experiment.

Data

In our experiment, two sets of data are used: sample set and test set. data in the sample set is provided by this course. To assure our work is valid in practice, we made another set of chessboard picture with our own cell phone. In a set, there are 10 pictures of a chessboard in different sizes and angles. On the right side of the following pictures, corners are marked with dots and connected with a line.

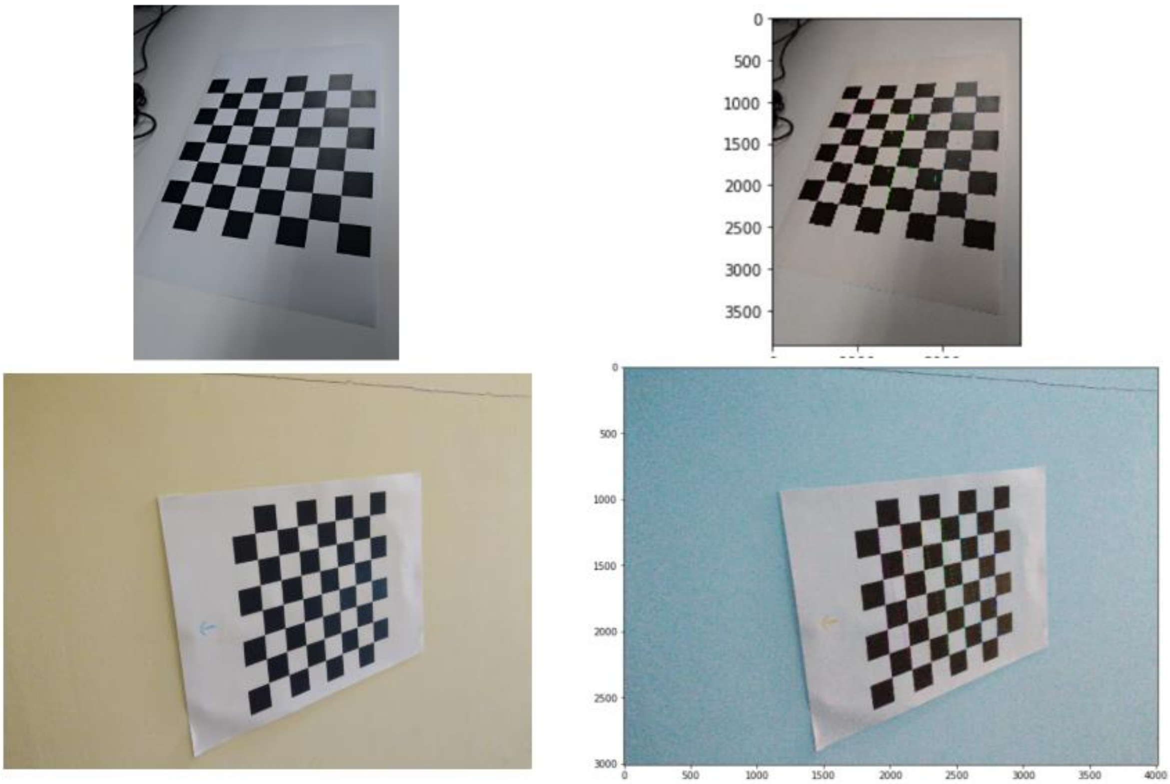

Fig.1 Data. Upper side shows the original and marked of a sample data. Lower side shows the original and marked of a test data.

Result

We present visualization of extrinsic matrices evaluated with sample and test data. The reference result evaluated via OpenCV library is on the left side. The result evaluated via our implementation is presented on the right side. Each cone presents rotation parameters and translation parameters.

Result of Sample Data

In general, our implementation locates each camera correctly in Fig. 2. The relative positions of orange and yellow cameras are slightly different from the reference result. In addition, the distances between zero point and the cameras in our result are slightly longer than the reference result. Finally, directions of yellow, orange, blue, light blue are wrong in both results.

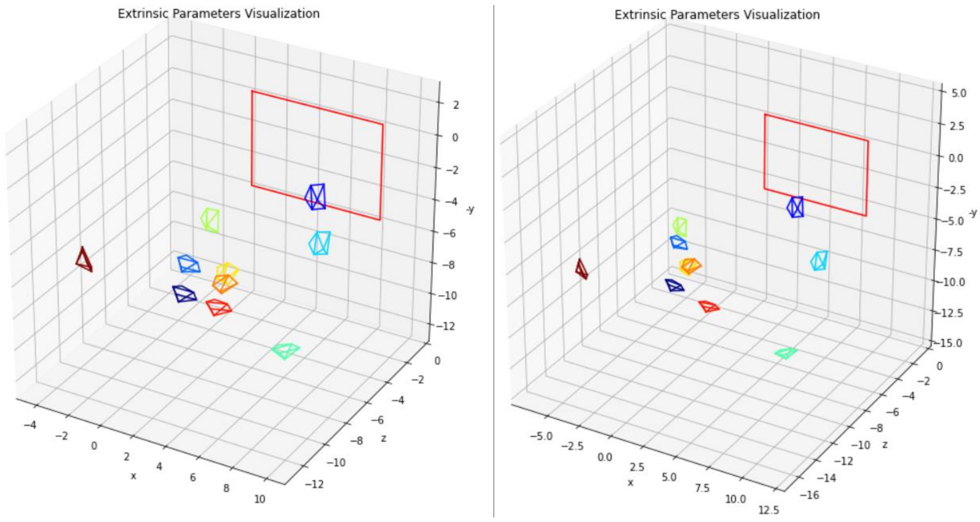

Fig. 2 Visualization of Extrinsic matrix evaluated via sample set. Right side is an evaluation via the OpenCV. Left side is an evaluation via our implementation.

Result of test data

The results of test set data are shown in Fig. 3. Surprisingly, locations and angles are the same in both results. Notably, four cameras closest to the chessboard show the wrong direction.

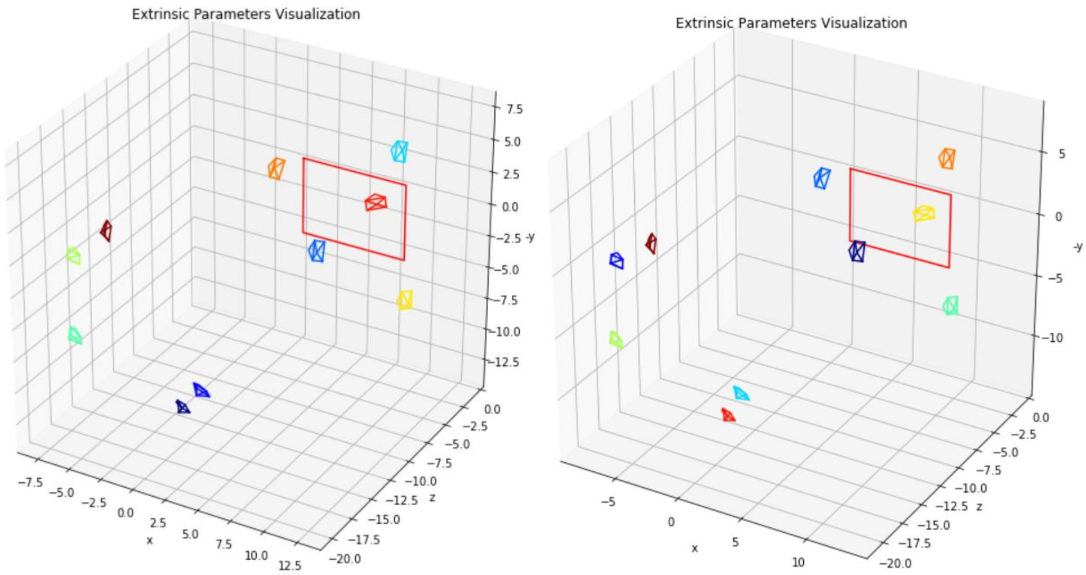

Fig. 3 Visualization of Extrinsic matrix evaluated via test set.
Right side is an evaluation via the OpenCV. Left side is an evaluation via our implementation.

IV. Discussion

When we tried the image data in the sample set, we found out some camera translation is slightly different from the reference result. We guess this is because Direct Linear Transformation is trying to find out the optimal solution homogeneous system of linear equations we constructed. But the point coordinates, homography and linear equation are "estimated" by our implementation so there should be some error.

In paper [2], the author normalized the obj and img coordinate before estimated the homography in order to improve numerical stability of the calculations and handle some extreme case (ie. some chess boards in images are very small, some are very large, or different image sizes). Therefore, we followed this paper and implemented coordinate normalization in our program. But we found out that the extrinsic visualization is slightly different from the unnormalized one (camera rotation and translation).

V. Conclusion

In this assignment, we study several papers and try to implement their algorithm for camera calibration from scratch. Our implementation contains: Normalize data points, Estimate Homography of each image, Compute intrinsic and extrinsic matrix of camera. Then, we compare our result with the calibration function of OpenCV by extrinsic visualization and found that they are almost the same. Moreover, we also try the images we take by smartphone and observe the results. We can see that our calculation is roughly correct.

VI. Group work assignment

0856095 黃柏豪: In charge of algorithm study and group work organization.

0856144 邱賢祐: In charge of shooting pictures and algorithm implementation.

0856148 陳奕遠: In charge of algorithm study and making report.

VII. Reference

1. Z. Zhang, "A flexible new technique for camera calibration," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330-1334, Nov. 2000.
2. Burger, W.: Zhang's camera calibration algorithm: in-depth tutorial and implementation. month (2016). Technical report HGB16-05, 16 May 2016, Department of Digital Media, University of Applied Sciences Upper Austria, School of Informatics, Communications and Media, Softwarepark 11, 4232 Hagenberg, Austria.