RBE 549: Homework 1 AutoCalib

Deepak Harshal Nagle
Robotics Engineering
Worcester Polytechnic Institute
Worcester, Massachusetts 01609
Email: dnagle@wpi.edu
Telephone: (774) 519-8335
Using one late day

***Abstract*—Abstract—The aim of this study is to estimate the intrinsic and extrinsic camera parameters as well as the distortion coefficients using the method described by Zhengyou Zhang in his paper titled ”A Flexible New Technique for Camera Calibration”. To achieve this, a checkerboard pattern with square sizes of 21.5 mm was used as the calibration target. Firstly, the intrinsic matrix (A) is approximated, followed by the estimation of rotation (R) and translation (t) using the initial estimates. Then, non-linear optimization is performed to refine the intrinsic matrix parameters and the distortion coefficients, minimizing the geometric error. The proposed approach was evaluated using real-world images of the calibration target, and the results demonstrate its effectiveness in accurately estimating the camera parameters.**

Introduction

When working with computer vision research that involves 3D geometry, it is essential to estimate the camera intrinsic parameters, which include the focal lengths, principle points, skew, and distortion coefficients. In 1998, Zhengyou Zhang presented an automatic and efficient way to perform camera calibration using a camera calibration matrix denoted by K. This report presents a summary of the implementation of Zhang's technique for calibration. The intrinsic matrix A is given by α, β, γ, u0, v0, where α and β are the scale factors, and describes skewness. The principal points are given by (u0, v0). Due to non-ideal camera lenses, there can be radial distortion, which we assume in our model. The distortion parameters k1 and k2 are estimated to correct the distortion in the image.

Data Generation

Estimating camera intrinsic parameters is an important task for computer vision research involving 3D geometry. Zhang's method for automatic camera calibration using a checkerboard pattern as a calibration target is widely used. In this study, a checkerboard pattern of 9 inner rows and 6 inner columns, with each square size of 21.5mm, was printed on an A4 paper and used as the calibration target for a Google Pixel XL phone camera. Thirteen images of the checkerboard with locked focus were taken to estimate the camera intrinsics using Zhang's method, which involves computing a homography matrix using at least four points per image and refining the intrinsic matrix and distortion coefficients through non-linear optimization.

I. Calibration Process

At first, corners of the checkerboard pattern image are detected using *cv2.findChessboardCorners*. Following the corner detection, the Homography matrix (H) between input image and an ideal 9x6 grid is calculated. Using the elements of Homography matrix, the array vij is obtained:

$$\mathbf{v}_{ij} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, \\ h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]^T$$

Fig. 1. vij calculation

Using v11, v12, v22, SVD is performed for calculating ”b”, based on this equation:

$$\begin{bmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{bmatrix} \mathbf{b} = \mathbf{0}$$

Fig. 2. Singular matrix relation

The elements of ”b” are as follows:

$$b = [B11, B12, B22, B13, B23, B33]^T$$

Fig. 3. ”b” expansion

Finally, the intrinsic parameters are calculated as follows:

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

Fig. 4. Intrinsic parameters calculations

Finally, we get intrinsic parameter matrix:

$$
\mathbf{A} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}
$$

Fig. 5. Matrix A

The corresponding values for our model inputs are:

```
alpha 2052.7881668840882
beta 2036.6341494517599
gamma -0.3695297113051416
lamba 0.633172276024536
u0 763.0610889578858
v0 1352.6145611741238
A =  [[ 2.05278817e+03 -3.69529711e-01  7.63061089e+02]
 [ 0.00000000e+00  2.03663415e+03  1.35261456e+03]
 [ 0.00000000e+00  0.00000000e+00  1.00000000e+00]]
```

Fig. 6. Matrix A

Thus, (u0, v0) = (763.06, 1352.61), (fx, fy) = (2052, 2036), skew-factor = -0.3695. Using ”A” matrix and the columns of ”H”, ”R” and ”t” matrices are computed as follows:

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

$$
\text{with } \lambda = 1/\left\|A^{-1}h_1\right\| = 1/\left\|A^{-1}h_2\right\|
$$

$$
R = [r_1, r_2, r_3]
$$

Fig. 7. R, t calculations

k1 and k2, both are initialized to be equal to zero.

II. Optimization:

Using A, R, and t matrices, the world point image (mesh grid) is projected to the image (pixel coordinates). The L2 norm between the image corner location and the projected point are calculated as:

$$
\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} - \mathbf{A}\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}
$$

Fig. 8. Loss calculations

The calculation was modified to fix the radial distortion by using the formula given below:

$$
\begin{aligned}
\breve{u} &= u + (u - u_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \\
\breve{v} &= v + (v - v_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]
\end{aligned}
$$

Fig. 9. Radial Distortion correction

here, u0 and v0 are calculated in the previous section through the matrix ”A”, (x, y) were obtained by multiplying transformation matrix (Rt) to the 3D world point coordinates.

The loss is calculated over all the corners of all the images, which gives us this loss summation:

$$
\sum_{i=1}^{n} \sum_{j=1}^{m} \|\mathbf{m}_{ij} - \hat{\mathbf{m}}(\mathbf{A}, \mathbf{R}_i, \mathbf{t}_i, \mathtt{M}_j)\|^2 \, ,
$$

Fig. 10. Loss Summation

This is a type of MLE (maximum likelihood estimation) where noise is considered to be equally distributed.

I used scipy.optimize.least_squares to optimize the loss function. I even took k1, k2 into consideration to optimize the loss. After a lot of efforts and changes in the loss calculation inside the code, I still cound not get good loss values. Also, the loss values hardly converged. However, the loss calculation method that I have attempted to implement is correct. The values that I obtained for ”A” matrix, intrinsic parameters, and k1, k2 are given below:

```
Final A:  [[ 2.05278817e+03 -3.69529711e-01  7.63061089e+02]
 [ 0.00000000e+00  2.03663415e+03  1.35261456e+03]
 [ 0.00000000e+00  0.00000000e+00  1.00000000e+00]]
Final k1, k2:  -4.34207e-06 1.3572462e-12
alpha:  2052.7881668840882
beta:  2036.6341494517599
gamma:  -0.3695297113051416
u0:  763.0610889578858
v0:  1352.6145611741238
```

Fig. 11. Final A, k1, k2, alpha, beta, gamma, u0, v0

Using k1 and k2 values, I fixed the distortion of the images. Below are the sets of images, original followed by images after fixing the distortion:

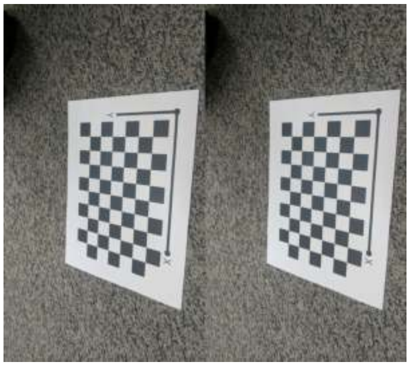

Fig. 12. An image followed by another image where distortion is fixed

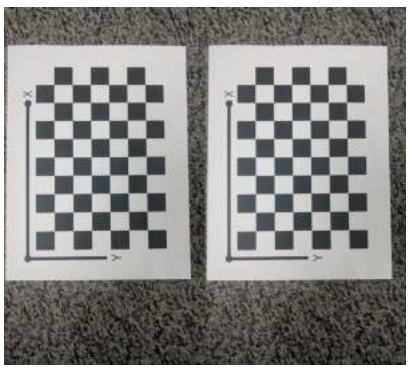

Fig. 13. An image followed by another image where distortion is fixed

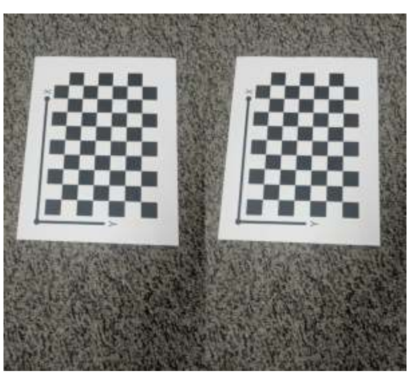

Fig. 14. An image followed by another image where distortion is fixed

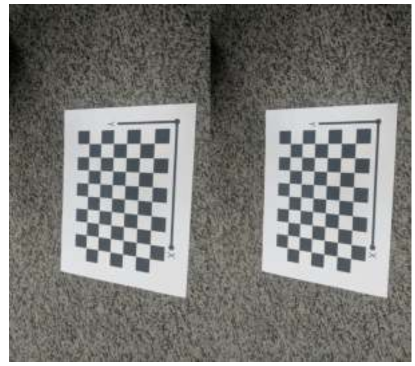

Fig. 15. An image followed by another image where distortion is fixed

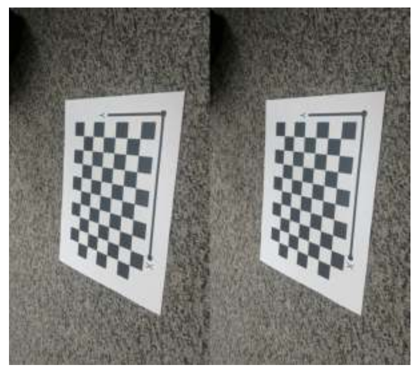

Fig. 16. An image followed by another image where distortion is fixed

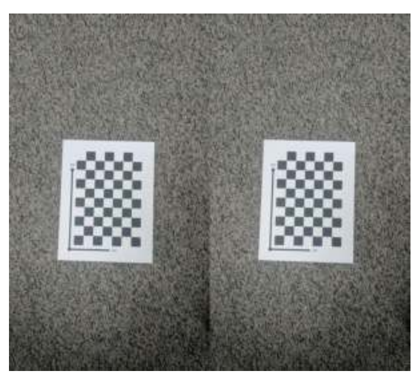

Fig. 17. An image followed by another image where distortion is fixed

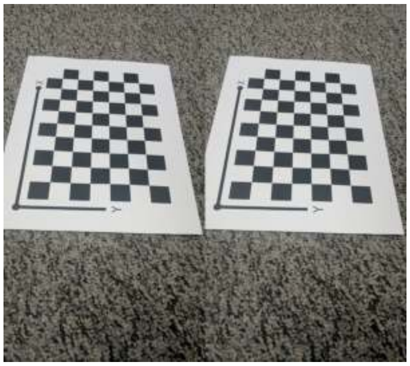

Fig. 18. An image followed by another image where distortion is fixed

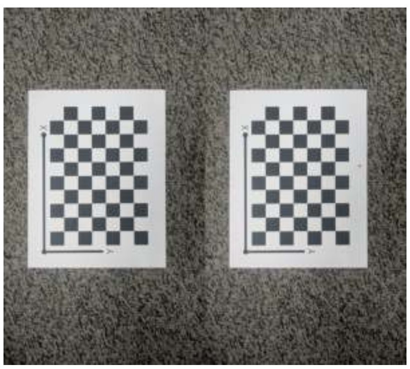

Fig. 19. An image followed by another image where distortion is fixed

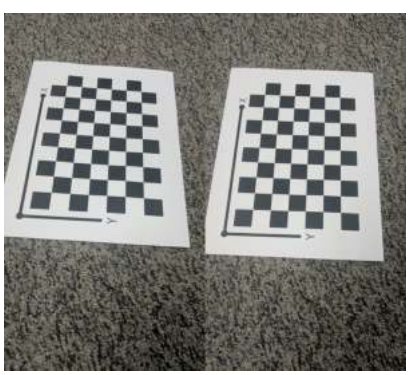

Fig. 20. An image followed by another image where distortion is fixed

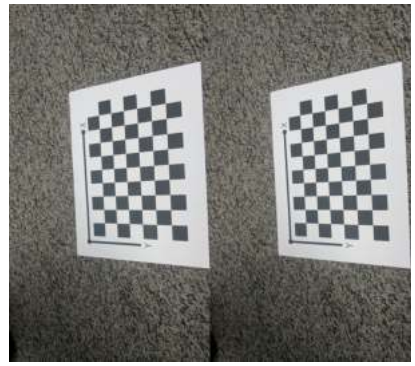

Fig. 21. An image followed by another image where distortion is fixed

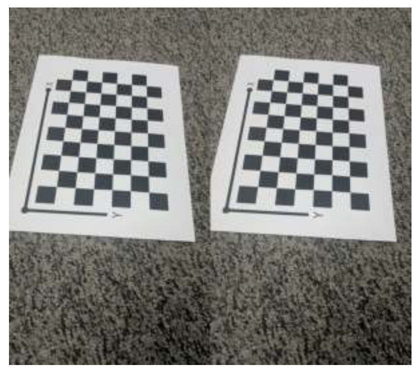

Fig. 22. An image followed by another image where distortion is fixed

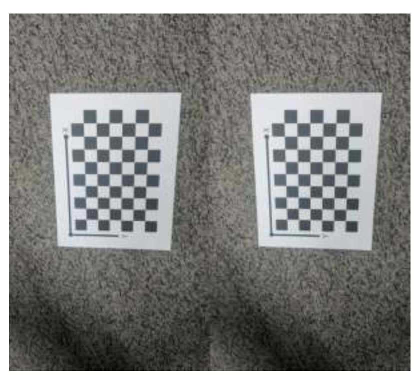

Fig. 23. An image followed by another image where distortion is fixed

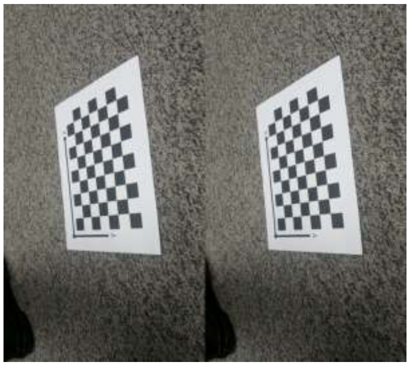

Fig. 24. An image followed by another image where distortion is fixed

References

[1] Zhengyou Zhang. A Flexible New Technique for Camera Calibration. Microsoft Research (1998)