RBE 549 Homework 1 - AutoCalib

UdayGirish Maradana
MS Robotics Engineering
Worcester Polytechnic Institute
Worcester, Massachusetts - 01609
Email: umaradana@wpi.edu

***Abstract*—This paper explains the work that I did as a part of RBE 549 Homework 1. In this homework, I have implemented a paper on Camera Calibration named "A Flexible New technique for Camera Calibration" proposed by Zhengyou Zhang.**

Index Terms - Camera Calibration, Homography, Intrinsic Parameters, Extrinsic parameters, Radial Distortion

I. INTRODUCTION

Camera Calibration is one of the primary requirements in Computer Vision as this helps to understand the world to Image coordinates. These transformations are primary even for monocular cameras and Stereo cameras. Especially for applications like Scene Reconstruction, 3D Understanding, and 3D Vision, this is important to correct for any camera-related issues and to correct for it. To convert a real-world point to an image point we need to understand two main parameters, one of which is intrinsic which is related to the camera internals which are the focal length and principal point etc. And the other one is extrinsic which is the Rotation and Translation matrix associated. This rotation and translation relates the world coordinate frame to the Image coordinate frame. Once we get an initial estimate of the Intrinsic parameters we further try to get the Distortion coefficients. The initial estimate is a form of Closed Form Solution and further we use nonlinear optimization to estimate and correct both intrinsic parameters and the distortion coefficient.

II. ESTIMATION - CAMERA INTRINSIC PARAMETERS

The camera intrinsic matrix consists of (u_0, v_0), the coordinates of the principal point, α and β the scale factors in image u and v axes, and γ the parameter describing the skewness of the two image axes.

$$A = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

For estimating homography between the model plane and its image, first, the pixel coordinates of the chessboard corners are detected using the `cv2.findChessboardCorners` function. Then, the real-world coordinates of the chessboard are calculated with the help of the size of the square. Using these coordinates, the homography matrix is computed using the singular Value Decomposition(SVD). This homography matrix consists of both the intrinsic matrix and the extrinsic matrix. Assuming the model plane is on $Z = 0$ of the world coordinate system, we get the following relation.

As we know, r_1 and r_2 are orthonormal; the following two equations can be obtained. Further using Cholesky Decomposition, the following closed-form solution is computed:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

Given an image of the model plane, an homography can be estimated. Using the knowledge that $\mathbf{r_1}$ and $\mathbf{r_2}$ are orthonormal, we get,

$$\begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} = \lambda A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$$

$$h_1^T A^{-T} A^{-1} h_1 = h_2^T A^{-T} A^{-1} h_2$$

For final symmetric solution, we define a vector B which is symmetric, defined by a 6D Vector,

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

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

Let the ith column vector of H be $\mathbf{h}_i = [h_{i1}, h_{i2}, h_{i3}]^T$. Then, we have

$$\mathbf{h}_i^T \mathbf{B} \mathbf{h}_j = \mathbf{v}_{ij}^T \mathbf{b}$$

$$\begin{aligned} v_{ij} = [& hi_1 hj_1, \\ & hi_1 hj_2 + hi_2 hj_1, \\ & hi_2 hj_2, \\ & hi_3 hj_1 + hi_1 hj_3, \\ & hi_3 hj_2 + hi_2 hj_3, \\ & hi_3 hj_3]^T \end{aligned}$$

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

From the above, we get b as

$$
\begin{aligned}
V\mathbf{b} &= 0 \\
v_0 &= \frac{B_{12}B_{13} - B_{11}B_{23}}{B_{11}B_{22} - B_{12}^2} \\
\lambda &= B_{33} - \frac{B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})}{B_{11}} \\
\alpha &= \sqrt{\frac{\lambda}{B_{11}}} \\
\beta &= \frac{\lambda B_{11}}{B_{11}B_{22} - B_{12}^2} \\
\gamma &= -\frac{B_{12}\alpha^2\beta}{\lambda} \\
u_0 &= \frac{\gamma v_0}{\beta} - \frac{B_{13}\alpha^2}{\lambda}
\end{aligned}
$$

III. Estimation- Camera Extrinsic Parameters

Once A is known, the extrinsic parameters for each image are readily computed.

$$
\begin{aligned}
r_1 &= \lambda A^{-1}\mathbf{h}_1 \\
r_2 &= \lambda A^{-1}\mathbf{h}_2 \\
r_3 &= r_1 \times r_2 \\
t &= \lambda A^{-1}\mathbf{h}_3
\end{aligned}
$$

IV. Estimation- Estimation of Distortion (Radial)

To reduce the camera distortion, we compute the Radial Distortion, in this we consider the simple case of estimating Radial distortion to the fourth order and neglecting the translational distortion.

We, initially assume our camera is accurate and start with the radial distortion parameter $k = [0, 0]^T$ and then we use maximum likelihood estimation to minimize the projection error.

$$\sum_{i=1}^{n}\sum_{j=1}^{m} \|m_{ij} - \hat{m}(A, R_i, t_i, M_j)\|^2$$

where, $\hat{m}(A, R_i, t_i, M_j)$ is the projection of point M_j in image i.

Adding Radial Distortion to the estimates,

$$
\begin{aligned}
\tilde{x} &= x + x\left[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2\right] \\
\tilde{y} &= y + y\left[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2\right]
\end{aligned}
$$

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

V. Results and Comparison

To understand the accuracy of the calculations, I have added one more library to calculate the calibration matrix using `cv2.calibrateCamera` and then the results are compared with our estimates. Open CV has 5 values for distortion coefficients, out of which 3 are radial distortion where the distortion is estimated to 6th power and 2 translational variances.

Intrinsic matrix Distortion - Initial esitmate:

$$A = \begin{bmatrix} 2058.80 & -0.96 & 762.41 \\ 0.00 & 2042.42 & 1347.07 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$$

$$k = [0, 0]$$

Intrinsic matrix Distortion - Optimized Estimate:

$$A_opt = \begin{bmatrix} 2058.79 & -0.96 & 762.42 \\ 0.00 & 2042.41 & 1347.08 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$$

$$k = [0.0132, -0.0971]$$

The value of mean projection error after optimization is 0.681

CV2 Baseline - Intrinsic matrix Distortion:

$$A = \begin{bmatrix} 2040.39 & 0.00 & 764.59 \\ 0.00 & 2032.17 & 1359.29 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$$

$$k = [0.2891, -2.399]$$

TABLE I
Reprojection Errors for 5 Random Images

Image	Reprojection Error	Reprojection Error - Optimized
Image 1	0.5677	0.5650
Image 2	0.67099	0.66182
Image 6	0.48776	0.48266
Image 9	0.67540	0.67066
Image 10	0.58541	0.58476

Outputs:

1. Output for a Slanted Image

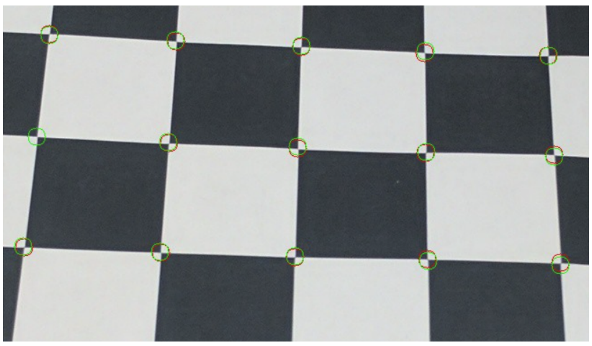

Fig. 1. Original Corners (red) and the Reprojected Corner (Green)

2. Outputs of Reprojected Images - Samples Numbering is based on the order in Image folder after sorting.

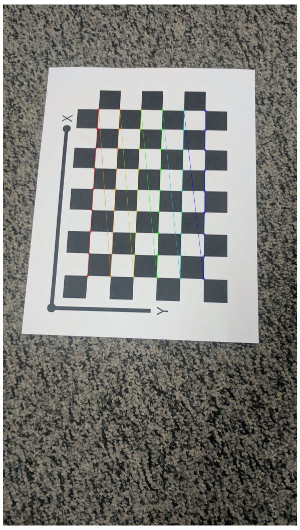

Fig. 2. Image 1

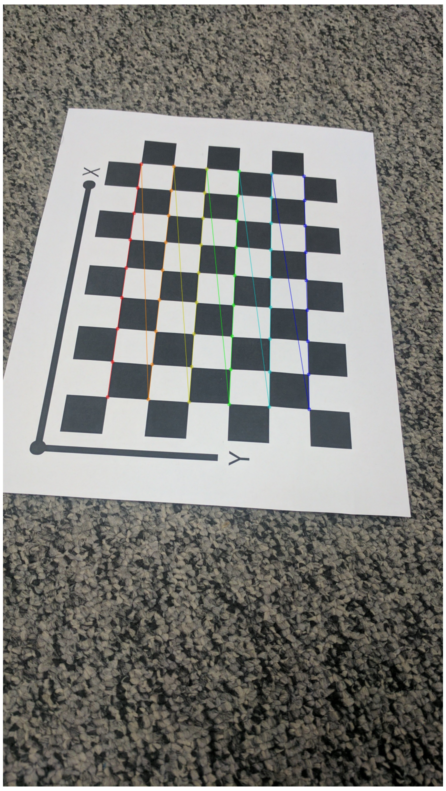

Fig. 3. Image 3

VI. Conclusion

The implementation is almost proper with the closer values to the CV2 baseline. Our distortion coefficients are a bit far from the baseline estimate, as we haven't estimated them with higher order and the uncertainties of homography implementation. This has to be studied more and understand more about how OpenCV does internal optimization and also how the translational distortion coefficients are considered. Overall, the implementation results are correct as shown in figures.

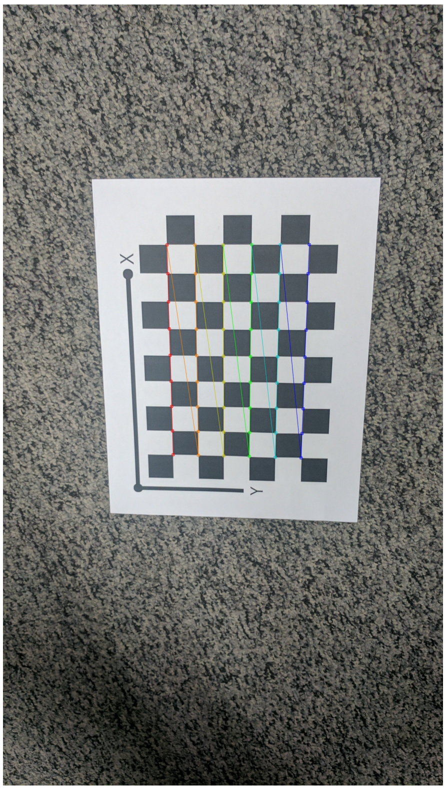

Fig. 4. Image 8

References

[1] Homework1 -Course Website(https://rbe549.github.io/spring2024/hw/hw1/)

[2] Zhang, Z. A flexible new technique for camera calibration. *IEEE Transactions On Pattern Analysis And Machine Intelligence*. **22**, 1330-1334 (2000)

[3] J. J. More, "The levenberg-marquardt algorithm: implementation and theory," in Numerical analysis. Springer, 1978, pp. 105–116.