RBE 549 Homework 1: AutoCalib

Pradnya Sushil Shinde
Robotics Engineering (MS)
Worcester Polytechnic Institute
Worcester, MA 01609
Email: pshinde1@wpi.edu

***Abstract*—The following report consists of a detailed analysis of a classical approach to the implementation of camera calibration.**

I. Introduction

Estimating parameters of the camera like the focal length, distortion coefficients, and principle point is called Camera Calibration. It is one of the most time-consuming and important parts of any computer vision research involving 3D geometry. An automatic way to perform efficient and robust camera calibration would be wonderful. One such method was presented by Zhengyou Zhang of Microsoft in [1] and is regarded as one of the hallmark papers in camera calibration. The camera calibration intrinsic matrix K is given as follows:

$$K = \begin{bmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

II. Methodology

Camera calibration consists of the following specific steps.

A. Calibration Data

Figure 1 represents the checkboard image used for the context of this calibration approach. The image was printed on an A4 paper and 13 images were collectively taken to be used for calibration purpose. Each square of the checkerboard pattern measures 21.5 mm. There are in total 11 rows and 8 columns but we will consider 9 rows and 6 columns for computation as a common practice.

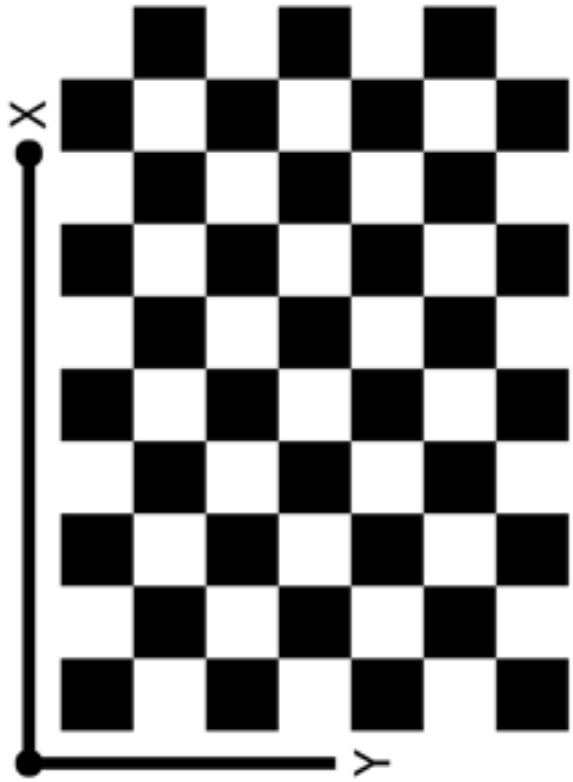

Fig. 1: Checkerboard Pattern

B. Initial Parameter Estimation

Parameters used in camera calibration are:
x, y - - image points
X, Y - - world points (points on the checkerboard)
k=[k1,k2] - - radial distortion parameters
K - - camera calibration matrix
R and t - - rotation matrix and the translation of the camera in the world frame.

1) Estimate Homography: First, we read a set of calibration images and generate image coordinates disguised as corners in the image. We can find the corners in each image by using *cv2.findChessboardcorners()* function of OpenCV. The output of detected corners for the first 9 images can be seen in Figure 2. Then we proceed to compute world coordinates with the help of the number of rows and columns of the checkboard and the pre-defined square size. We pass both the image coordinates and world coordinates to a function to compute the homography matrix.

2) Compute Intrinsic Camera Calibration Matrix K: To compute the intrinsic matrix K, we first need to define a vector B which can be calculated by computing a vector v defined as $\mathbf{v}_{ij} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]$ where h represents the homography matrix.
A matrix V then can be formed from the vector v definition: $\begin{bmatrix} (v_{12})^T \\ (v_{11} - v_{22})^T \end{bmatrix}^T$
We will then calculate the Singular Value Decomposition (SVD) of the matrix V using *np.linalg.svd()* function of NumPy library, transpose the resulting matrix Vh and

extract the last column of Vh as the vector B. The elements $\alpha(f_x), \gamma, \beta(f_y), u0(c_x), v0(c_y)$ of the intrinsic matrix K can be calculated from B in the following way:

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

3) Extrinsic Matrix: We will calculate the Extrinsic Matrix defined by the rotational and translational quantities. The extrinsic matrix Rt consists of the following values:

$$r_1 = \lambda K^{-1} h_1$$
$$r_2 = \lambda K^{-1} h_2$$
$$t = \lambda K^{-1} h_3$$
$$Rt = \begin{bmatrix} r_1 \\ r_2 \\ t \end{bmatrix}$$

where h represents the homography matrix and K represents the camera intrinsic matrix.

4) Initial Distortion: Let's define the initial distortion error as $k = [00]$

5) Projection Error & Optimization: The next step is to define an error function that calculates the projection error and estimates reprojected image points. The error function can be defined as below:

$$\sum_{i=1}^{N}\sum_{j=1}^{M} \|x_{i,j} - \hat{x}_{i,j}(K, R_i, t_i, X_j, k)\|$$

The optimization problem for error minimization is defined below:

$$\arg \min_{f_x, f_y, c_x, c_y, k_1, k_2} \sum_{i=1}^{N}\sum_{j=1}^{M} \|x_{i,j} - \hat{x}^{i,j}(K, R_i, t_i, X_j, k)\|$$

We will use *leastsquares()* function from the *scipy.optim* library to optimize the parameters using the error function defined above.

C. Results

1) Rectification of Images: The distorted images are corrected using the optimized distortion error value and intrinsic camera matrix. The last 6 undistorted images can be found in Figure 3.

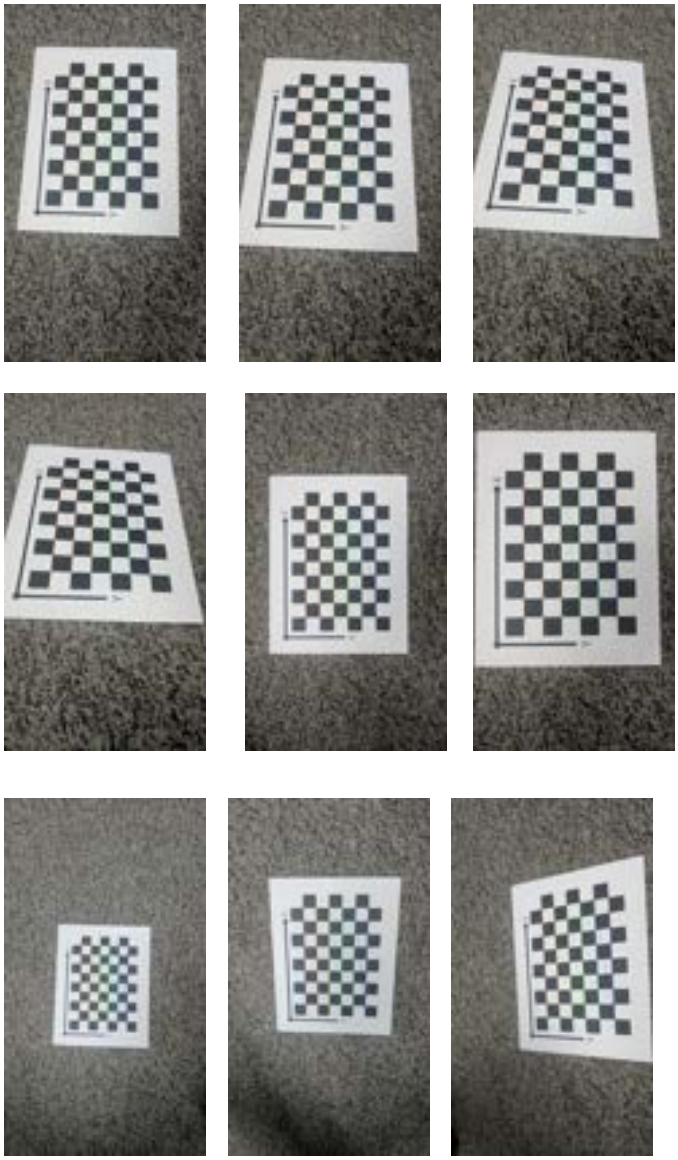

Fig. 2: Detected Corners Images 1-9

Parameter	Initial Values	Optimized Value
f_x	2053.04111	2053.03664
f_y	2037.10192	2037.09391
c_x	762.79857	762.80454
c_y	1351.64448	1351.65692
γ	-0.4682611	-0.4685900
k	[0 0]	[0.01218641 -0.08670301]

TABLE I: Initial and Optimized Camera Parameters

2) Camera Parameters: Find the values of k and $\alpha(f_x), \gamma, \beta(f_y), u0(c_x), v0(c_y)$ of the intrinsic matrix K, pre and post-optimization in the table below.

D. Errors

The projection error pre-optimization is: $[0.69\ 0.75\ 0.87\ 0.98\ 0.59\ 0.74\ 0.84\ 0.52\ 0.68\ 0.64\ 0.85\ 0.98\ 0.75]$

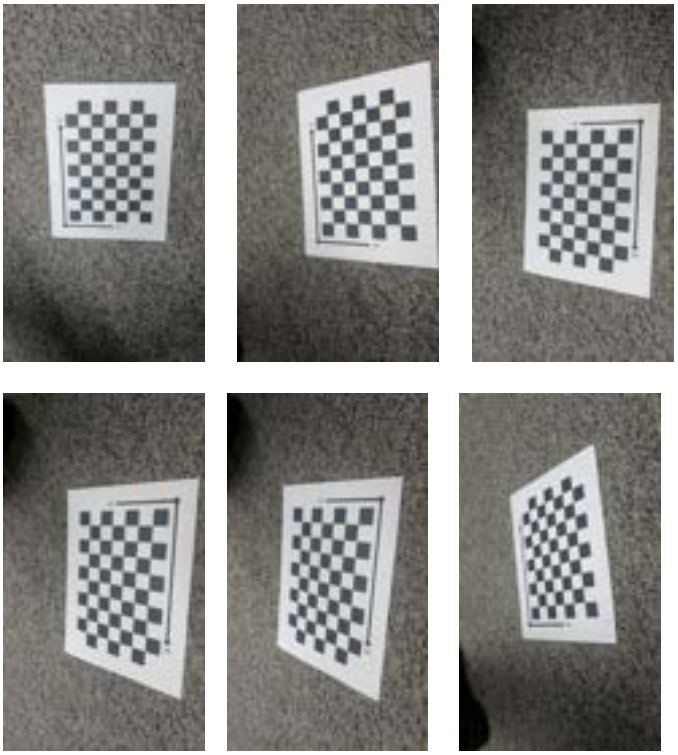

Fig. 3: Rectified Images 8-13

The projection error post-optimization is: [0.65 0.72 0.85 0.97 0.58 0.71 0.84 0.51 0.67 0.64 0.84 0.970.75]
The mean projection error after optimization is: 0.75

REFERENCES

[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, doi: 10.1109/34.888718. keywords: Cameras;Calibration;Computer vision;Layout;Lenses;Nonlinear distortion;Closed-form solution;Maximum likelihood estimation;Computer simulation;Testing,

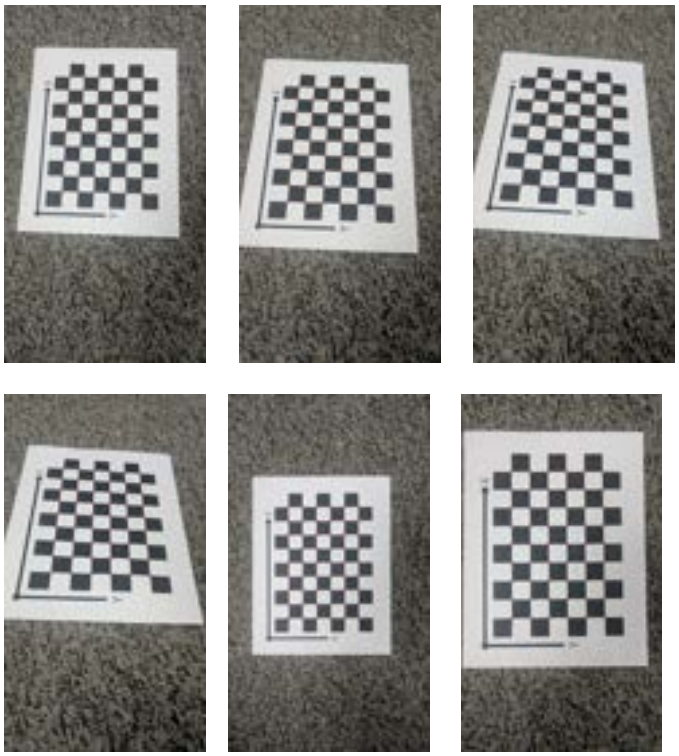

Fig. 4: Reprojected Points Images 1-6

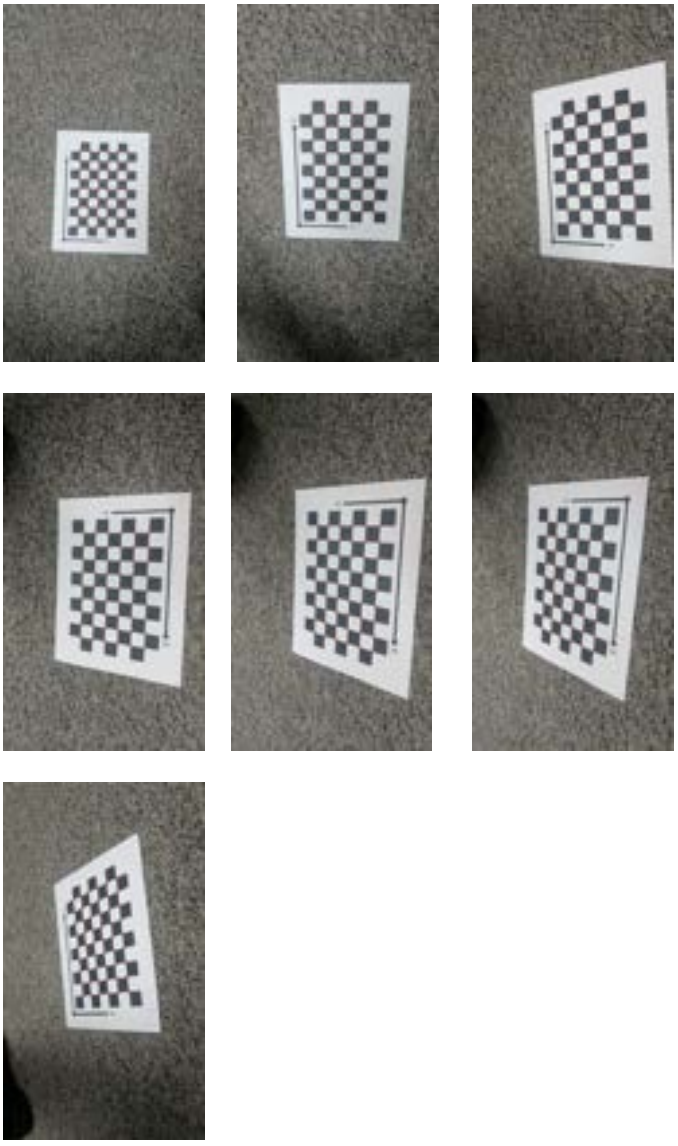

Fig. 5: Reprojected Points Image 7-13