Computer Vision HW1: AutoCalib

Abhijeet Sanjay Rathi
MS Robotics Engineering
asrathi@wpi.edu
901015394

***Abstract*—Calibrating a camera involves determining parameters such as focal length, distortion coefficients, and the principal point. This process, known as Camera Calibration, stands as a crucial and time-consuming aspect in any computer vision research that incorporates 3D geometry. Here, I have implemented "A Flexible New Technique for Camera Calibration" proposed by Zhengyou Zhang.**

I. Introduction

In this homework, a checkerboard pattern of 10 x 7 (rows and columns) is given with each square of size 21.5 mm. This image will be used to calibrate the camera. To map a real-world point to its corresponding image representation, two types of matrices come into play. Firstly, the camera calibration matrix incorporates the coordinates of the principal point and the scale factors along the image axes. Secondly, the extrinsic parameters matrix involves rotation and translation, establishing the relationship between the world coordinate system and the camera coordinate system. The first matrix is also called Intrinsic Matrix which considers Intrinsic parameters coordinates of the principal point (u_0, v_0), the scale factors α and β in image u and v axes, and the parameter γ describing the skewness of the two image axes.

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

The second matrix which involves rotation and translation is given as:

$$R = \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix}$$

where r_1, r_2, r_3, t each is a 3 x 1 vector.
Nevertheless, owing to the non-ideal characteristics of the camera lens, image distortion may occur. In this context, we will make the assumption that the distortion is radial, exhibiting a symmetric nature where ideal image points undergo distortion along radial directions emanating from the distortion center. The distortion is quantified by parameters denoted as k_1 and k_2, and our objective is to accurately estimate these radial distortion parameters.

II. Intrinsic Parameters Estimation

To estimate the homography between the model plane and its image, initially, the pixel coordinates of the chessboard corners are identified using the cv2.findChessboardCorners function. Subsequently, the real-world coordinates of the chessboard are calculated based on the size of the squares. Using these coordinates, the Homography matrix is computed through the cv2.findHomography function. This matrix encompasses both the intrinsic and extrinsic matrices. Assuming the model plane lies on Z = 0 in the world coordinate system, the following relation is derived.

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

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

This gives the equation for finding Homography matrix (H):

$$H = A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$$

Using the Homography matrix H from above, we compute v_{ij}:

$$v_{ij} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, h_{i3}h_{j1} \\ + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]$$

As r_1 and r_2 are orthonormal,

$$h_1^T A^{-T} A^{-1} h_2 = 0$$

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

These above two fundamental constraints can we rewritten as:

$$s \begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0$$

If n images of model plane are observed, by stacking n such above equation,

$$Vb = 0$$

where V is a 2n x 6 matrix. From this, b a 6D vector is found.

$$b = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]$$

Finally, from this b vector, intrinsic parameters can be calculated as shown below:

$$v_0 = (B_{12}B_{13} - B_{11}B_{23})/(B_{11}B_{22} - B_{12}^2)$$

$$\lambda = B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23}]/B_{11}$$

$$\alpha = \sqrt{\lambda/B_{11}}$$

$$\beta = \sqrt{\lambda B_{11}/(B_{11}B_{22} - B_{12}^2)}$$

$$\gamma = -B_{12}\alpha^2\beta/\lambda$$

$$u_0 = \gamma v_0/\beta - B_{13}\alpha^2/\lambda$$

After computing these parameters, just enter the values in the A matrix to get the Intrinsic Matrix A.

III. Extrinsic Parameters Calculation

The extrinsic parameters for each image can be calculated using the below equations:

$$r_1 = \lambda A^{-1} h_1$$

$$r_2 = \lambda A^{-1} h_2$$

$$r_3 = r_1 \times r_2$$

$$t = \lambda A^{-1} h_3$$

where

$$\lambda = 1/||A^{-1}h_1|| = 1/||A^{-1}h_2||$$

IV. Estimation of Distortion and Non-linear Geometric Error Minimization

Assuming minimal distortion in the camera, the initial estimate of the distortion parameters can be made as $k = [0, 0]^T$. With this initial estimate, minimization of the reprojection error can be implemented using the equation:

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

where n are the number of images of model plane and m are the points on the model plane. Here, $\hat{m}(A, R_i, t_i, M_j)$ is the projection of point M_j in image i.

Let (u, v) be the ideal (nonobservable distortion-free) pixel image coordinates, and $(\breve{u}, \breve{v})$ the corresponding real observed image coordinates. The ideal points are the projection of the model points according to the pinhole model. Similarly, (x, y) and $(\bar{x}, \bar{y})$ are the ideal (distortion-free) and real (distorted) normalized image coordinates.

$$\breve{u} = u + (u - u_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]$$

$$\breve{v} = v + (v - v_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]$$

where k_1 and k_2 are the coefficients of radial distortion.

Furthermore, the Least Squares function is employed to minimize this error and optimize the parameters. Using the optimized parameters, the mean of the reprojected error for the given points is compute using:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} ||m_{i,j} - \hat{m}(A, k_1, k_2, R_i, t_i, M_j)||^2$$

V. Results

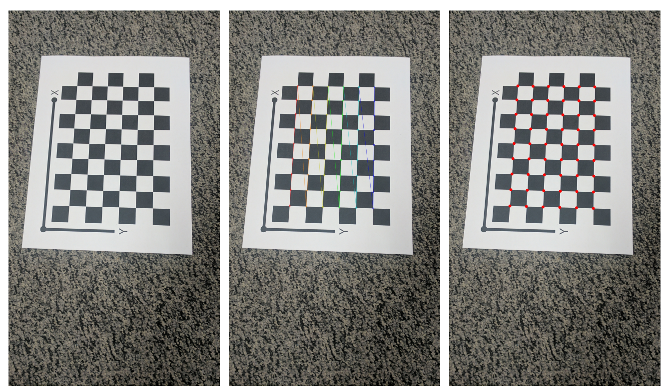
Executing the code, gives me following results:

$$A_{opt} = \begin{bmatrix} 2.0652566e+03 & -2.9397471e+00 & 7.6467639e+02 \\ 0.0000000e+00 & 2.0534836e+03 & 1.3627694e+03 \\ 0.0000000e+00 & 0.0000000e+00 & 1.0000000e+00 \end{bmatrix}$$

$$Distortion_{opt} = \begin{bmatrix} 0.00018355 & -0.00015911 \end{bmatrix}$$

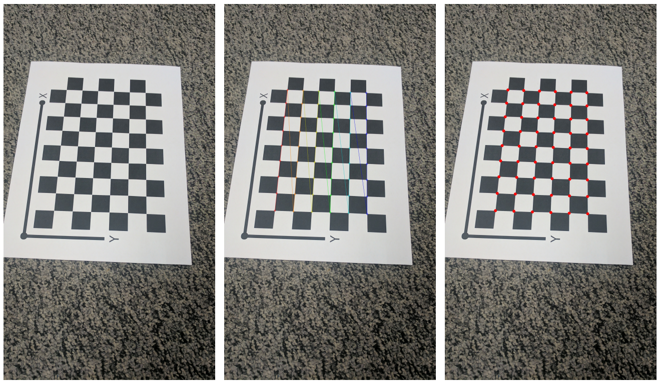
The mean reprojection error (before optimization) comes out to be 0.8110529542384748.
The mean reprojection error (after optimization) comes out to be 0.8105802901930453.

Image	**Error before Optimization**	**Error after Optimization**
1	0.05316171	0.05312406
2	0.05783641	0.05780631
3	0.06990129	0.06993464
4	0.07684916	0.07682356
5	0.05121874	0.05113317
6	0.05734893	0.057295
7	0.06645683	0.06644563
8	0.04001038	0.03998399
9	0.05281222	0.05277625
10	0.0501051	0.05006057
11	0.07910402	0.07902345
12	0.09793214	0.09786362
13	0.05831602	0.05831005



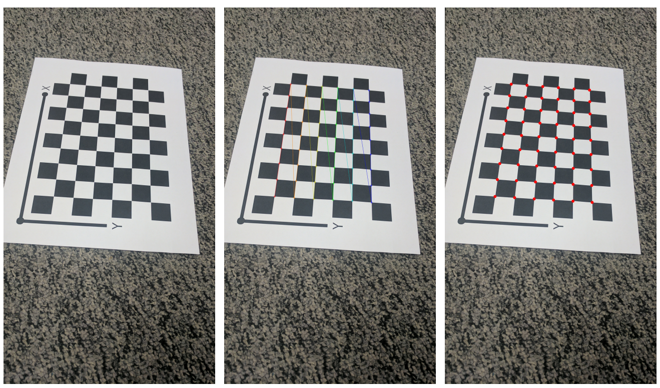
(a) Input (b) Chessboard (c) Rectified

Fig. 1: *Input, Chessboard, Rectified for Image 1*



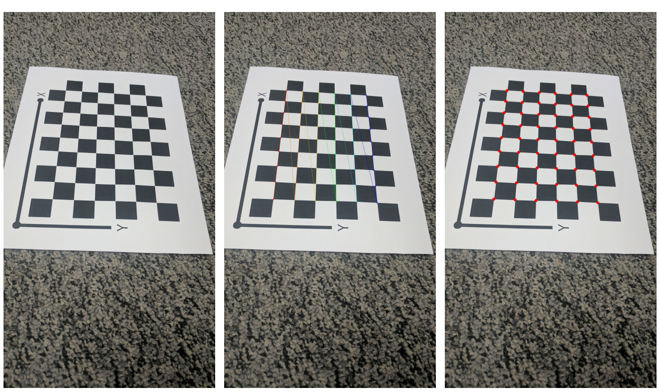
(a) Input (b) Chessboard (c) Rectified

Fig. 2: *Input, Chessboard, Rectified for Image 2*



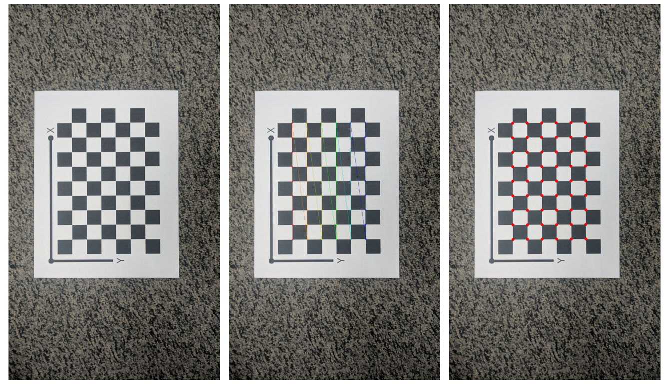
(a) Input (b) Chessboard (c) Rectified

Fig. 3: *Input, Chessboard, Rectified for Image 3*



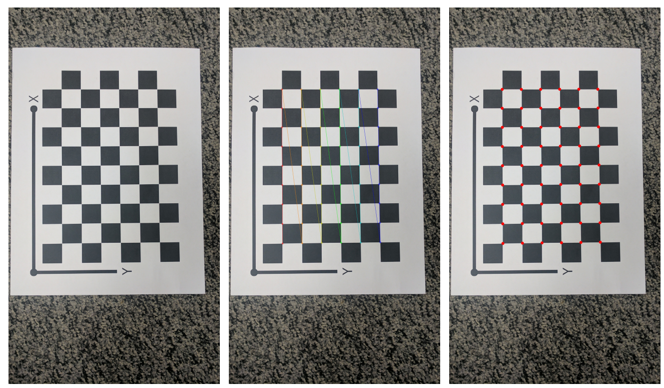
(a) Input (b) Chessboard (c) Rectified

Fig. 4: *Input, Chessboard, Rectified for Image 4*



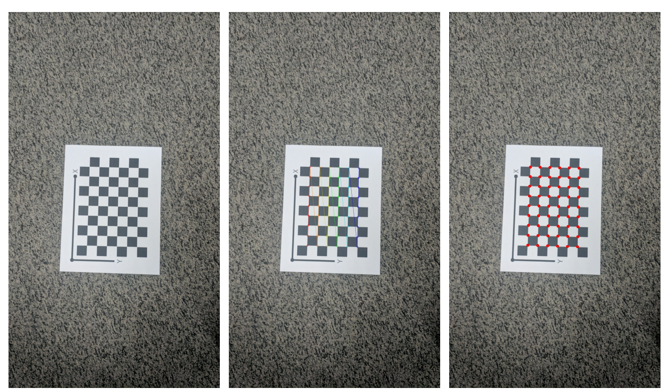
(a) Input (b) Chessboard (c) Rectified

Fig. 5: *Input, Chessboard, Rectified for Image 5*



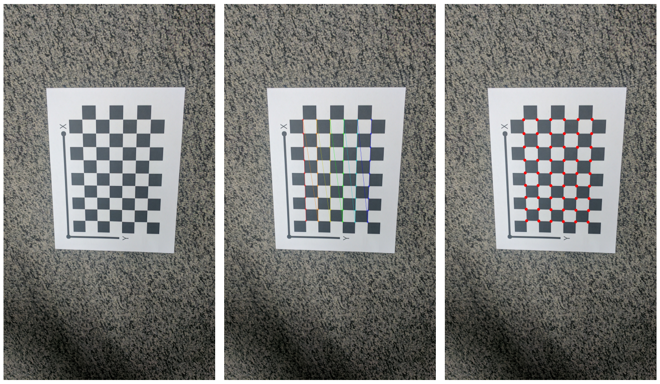
(a) Input (b) Chessboard (c) Rectified

Fig. 6: *Input, Chessboard, Rectified for Image 6*



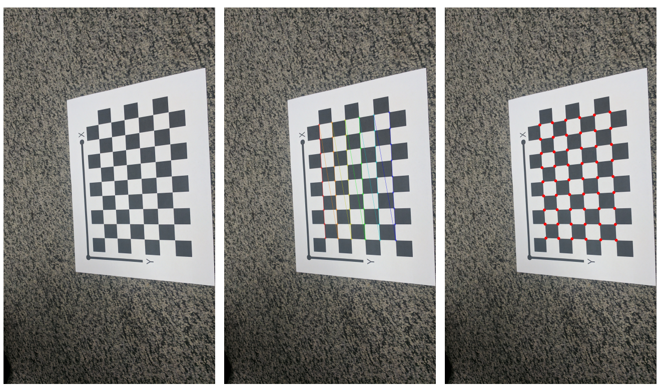
(a) Input (b) Chessboard (c) Rectified

Fig. 7: *Input, Chessboard, Rectified for Image 7*



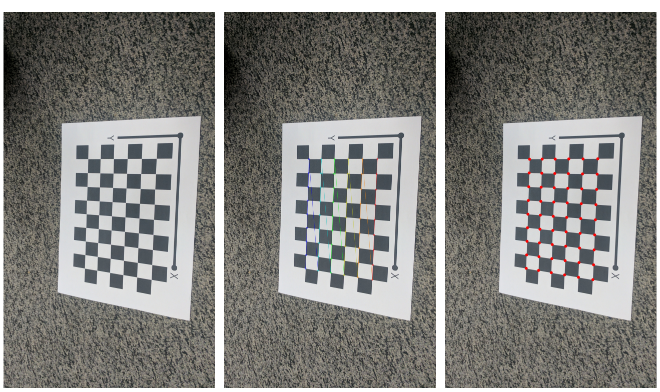
(a) Input (b) Chessboard (c) Rectified

Fig. 8: *Input, Chessboard, Rectified for Image 8*



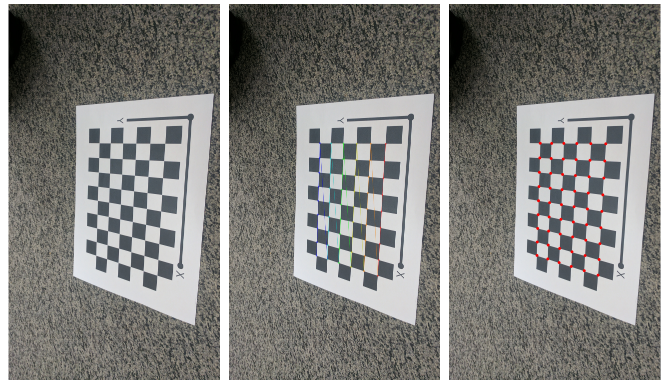
(a) Input (b) Chessboard (c) Rectified

Fig. 9: *Input, Chessboard, Rectified for Image 9*



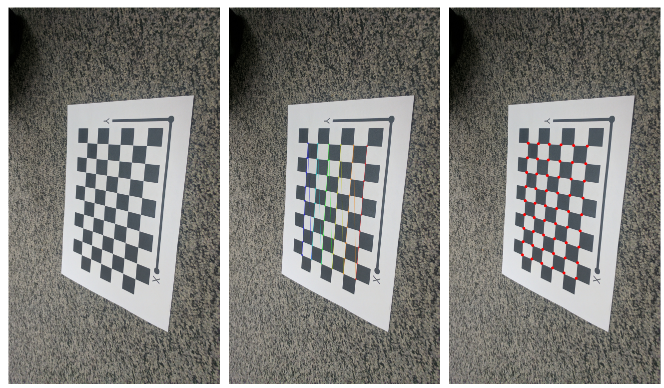
(a) Input (b) Chessboard (c) Rectified

Fig. 10: *Input, Chessboard, Rectified for Image 10*



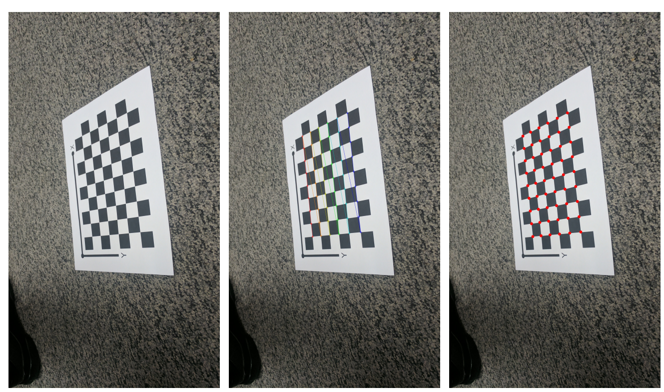
(a) Input (b) Chessboard (c) Rectified

Fig. 11: *Input, Chessboard, Rectified for Image 11*



(a) Input (b) Chessboard (c) Rectified

Fig. 12: *Input, Chessboard, Rectified for Image 12*



(a) Input (b) Chessboard (c) Rectified

Fig. 13: *Input, Chessboard, Rectified for Image 13*