HW1 : AutoCalib

Thanikai Adhithiyan Shanmugam
Robotics Engineering Department
Worcester Polytechnic Insititute
(Using 1 Late Day)

***Abstract*—Camera Calibration is a method of estimating parameters of the camera like the focal length, distortion coefficients and principle point. Camera calibration plays a pivotal role in computer vision applications by establishing the intrinsic and extrinsic parameters of imaging devices. This paper performs the most widely used calibration technique from Zhang on checkerboard patterns and analyses the effect of distortion on projection error and the calibration matrix.**

I. Introduction

The paper focuses on implementing the camera calibration technique proposed by Zhang where it proposes a feasible solution to estimate the parameters of the camera calibration matrix and distortion parameters. The camera calibration matrix is generally denoted by A, which is a symmetric matrix.

To generate the data required for calibration, we use the target image ”Checkerboard.pdf” which has been used in the Zhang paper to calibrate and find the corners of the checkboard. These are our world frame coordinates or reference frames. In this paper, the corners are found in 2 different ways and are compared later to analyse changes in the distortion matrix.

1. First method is directly finding the coordinates by inputting the target image into cv2.findcorners() function.
2. The second method includes hardcoding the corners with the width(21.5mm) given in the problem statement and creating a world/reference frame.

A. Data

In this paper, we are utilising 13 images of checkerboard orientations, taken from a Google Pixel XL phone. The size of the board is 7x10 but following the common practice, we will exclude the corners and take 6x9 size for our calibration.

II. Inital Parameter Estimation

1) Homography: Once the corners of the images are found using the methods using findcorners() function, we need to calculate the Transformation/Homography matrix for each image.
For performing this operation, the paper has written function for 2 methods:

1. Use inbuilt function cv2.homography()
2. Implement Singular value Decomposition by arranging Ax =0.

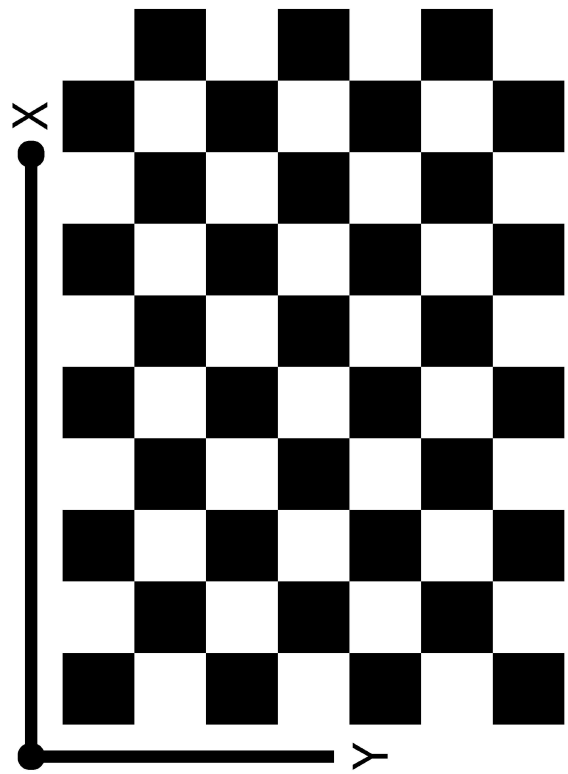

Fig. 1. Target Checkboard

After performing SVD, we normalize the elements so that the last element in H becomes 1 for every image.

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = H \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{1}$$

A. Decomposition of H

We decompose the H matrix as the intrinsic and extrinsic matrices. Since r1 and r2 in the Rotation matrix are orthonormal and the property based on circular point, we can write Eqn 3.

$$h_1^T A^{-T} A^{-1} h_2 = 0 \tag{2}$$

$$h_1^T A^{-T} A^{-1} h_1 = h_2^T A^{-T} A^{-1} h_2 \tag{3}$$

To simplify the equation the matrix above, we define B as,

$$B = A^{-T}A^{-1} \tag{4}$$

B. Calculate Intrinsic and Extrinsic Parameters

From the above equation, since B is a symmetric matrix, we have 6 unknown variables and therefore we need 3 tomography matrices to calculate them.
We define v in term of all the 3 tomography matrices and estimate the intrinsic parameters of A.

$$\mathrm{v}_{ij} = [h_{i1}h_{j1}, h_{i2}h_{j1} + h_{i1}h_{j2}, h_{i2}h_{j2},$$
$$h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i2}h_{j3} + h_{i3}h_{ij}2, h_{i3}h_{j3}]$$

$$v0 = (B_{12}B_{13} - B_{11}B_{23})/(B_{11}B_{22} - B_{12}^2) \tag{5}$$

$$\lambda = B_{33} - [B_{13}^2 + v0(B_{12}B_{13} - B_{11}B_{23}]]/B_{11} \tag{6}$$

$$\alpha = \sqrt{\lambda/B_{11}} \tag{7}$$

$$\beta = \sqrt{_{11}/(B_{11}B_{22} - B_{12}^2)} \tag{8}$$

$$\gamma = -B_{12}\alpha^2\beta/\lambda \tag{9}$$

$$u0 = 0/\beta - B_{13}\alpha^2/\lambda \tag{10}$$

Once we have calculated the intrinsic parameters, we define the extrinsic parameters as:

$$r1 = \lambda A^{-1}h_1 \tag{11}$$

$$r2 = \lambda A^{-1}h_2; \tag{12}$$

$$r3 = \lambda A^{-1}h_3 \tag{13}$$

From this we have calculated the R and t of the transformation matrix and also the A matrix.

III. Distortion and Projection error Estimation

The paper initially assumes the distortion to be minimal and assigned $k = [0, 0]^T$. With the initial parameters estimated , we try to minimize the projection error from:

$$\sum_{i=1} n \sum_{j=1} m ||m_{ij} - \dot{m}(A, R, t, M_j)||^2 \tag{14}$$

Also, it is important to note that (u,v) here are distortion-free image coordinates and (u, v) which is calculated by

$$u = u + (u - u0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \tag{15}$$

$$u = v + (v - v0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \tag{16}$$

To optimize the loss function we use the least squares method from the scipy library.

IV. Results and discussions

A. Calibration Matrix

The calibration Matrix A before optimization was:

$$\begin{bmatrix} 2.08171231e+03 & -2.13491393e+00 & 7.52916307e+02 \\ 0.00000000e+00 & 2.07356037e+03 & 1.36723461e+03 \\ 0.00000000e+00 & 0.00000000e+00 & 1.00000000e+00 \end{bmatrix} \tag{17}$$

The calibration Matrix A after Optimization is:

$$\begin{bmatrix} 2.08171232e+03 & -2.13490452e+00 & 7.52916290e+02 \\ 0.00000000e+00 & 2.07356033e+03 & 1.36723449e+03 \\ 0.00000000e+00 & 0.00000000e+00 & 1.00000000e+00 \end{bmatrix} \tag{18}$$

B. Mean reprojection error and distortion matrix

The mean reprojection error, before and after optimization is summarised in the table below.

Time	Mean Error
Before	0.7513312548525106
After	0.7480409832200003

The distortion matrix after optimization is :

$$k = \begin{bmatrix} 0.019596196066044572 & -0.2599113406540603 \end{bmatrix} \tag{19}$$

V. Conclusion

This paper helps us visualize and gain knowledge on the most widely used camera calibration technique and paves a new path into camera calibration and computer vision.

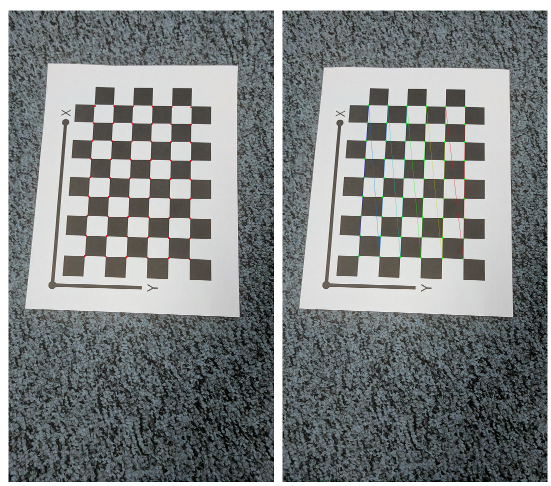

Fig. 2. Image1: Reprojected Corners and Corners

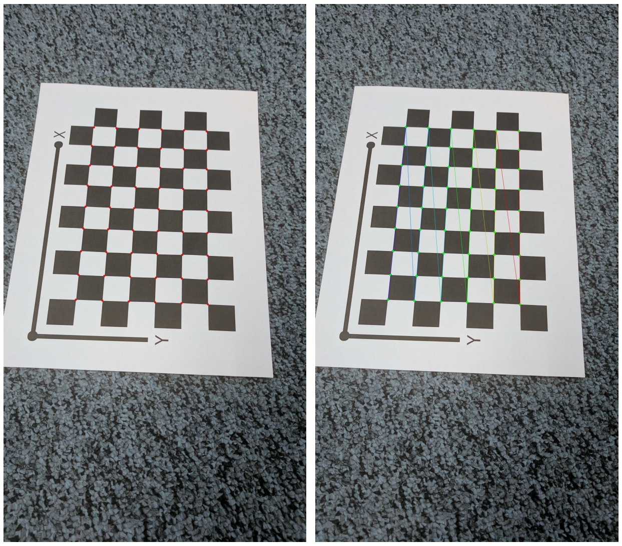

Fig. 3. Image2: Reprojected Corners and Corners

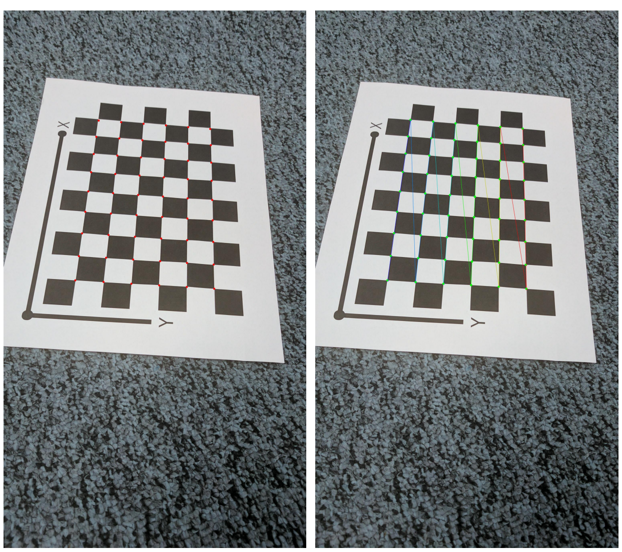

Fig. 4. Image3: Reprojected Corners and Corner

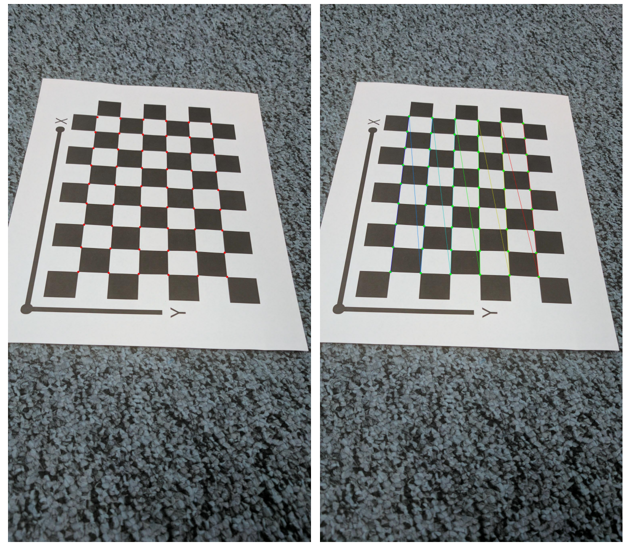

Fig. 5. Image4: Reprojected Corners and Corner

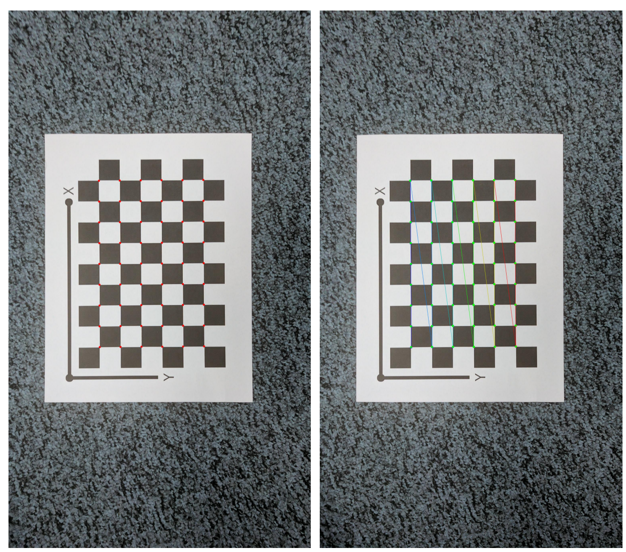

Fig. 6. Image5: Reprojected Corners and Corners

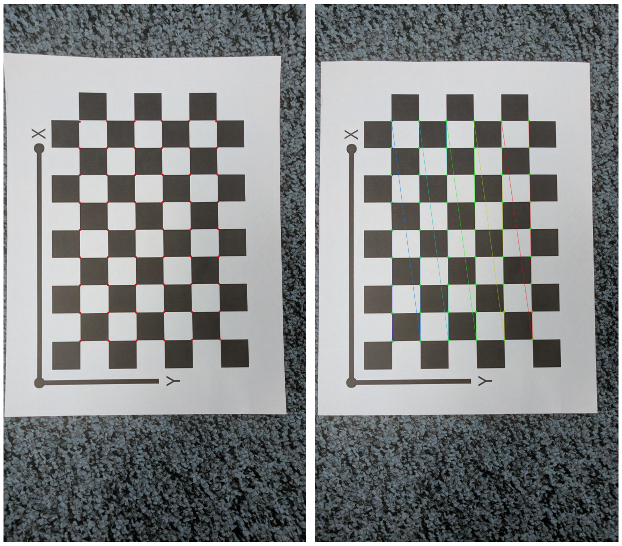

Fig. 7. Image6: Reprojected Corners and Corners

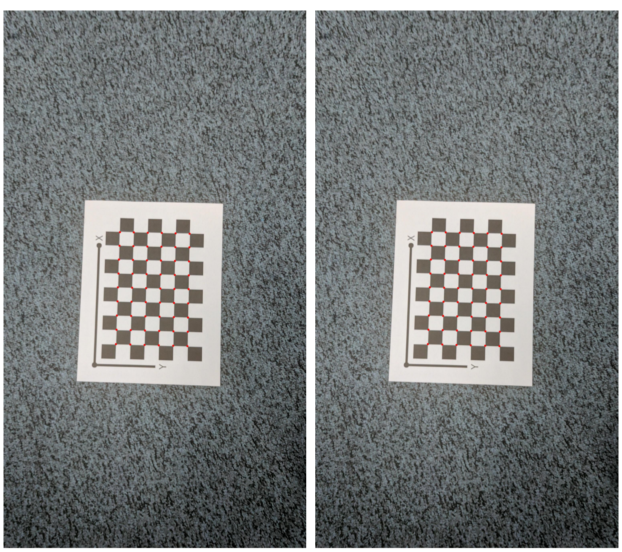

Fig. 8. Image7:Reprojected Corners and Corners

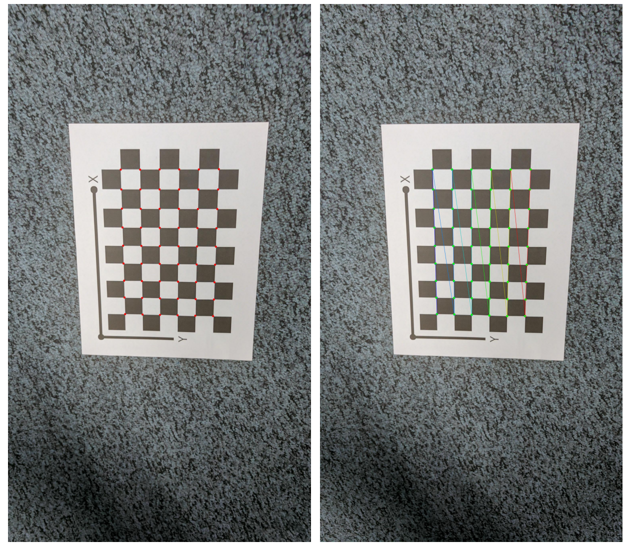

Fig. 9. Image8: Reprojected Corners and Corners

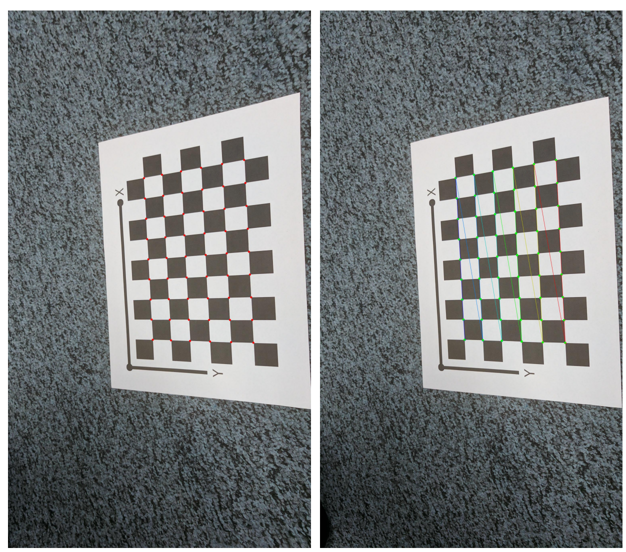

Fig. 10. Image9: Reprojected Corners and Corners

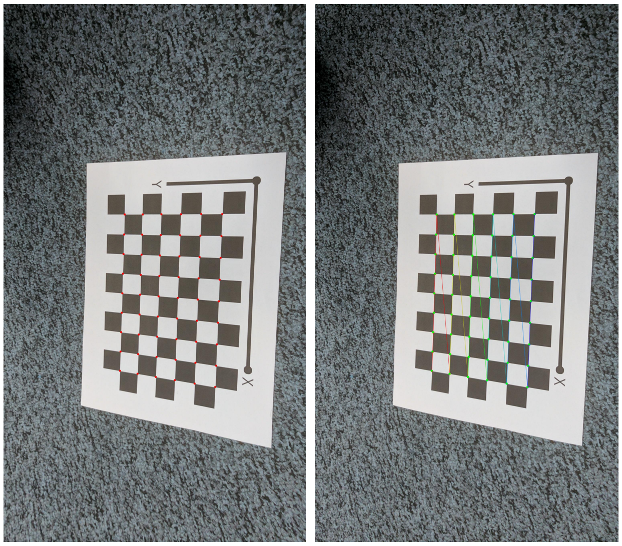

Fig. 11. Image10: Reprojected Corners and Corners

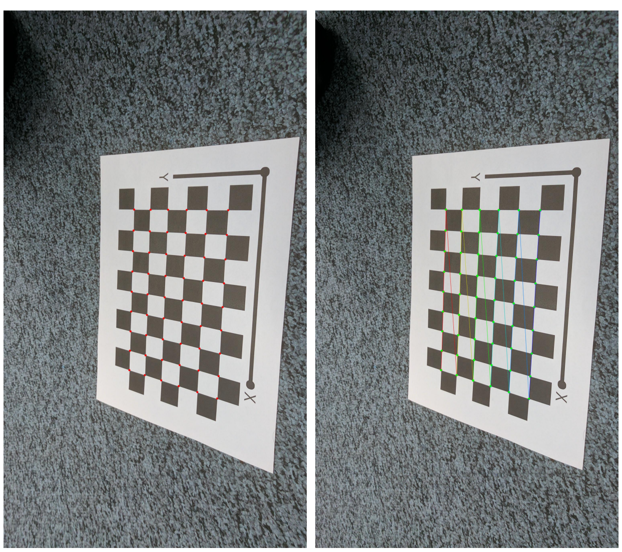

Fig. 12. Image11: Reprojected Corners and Corners

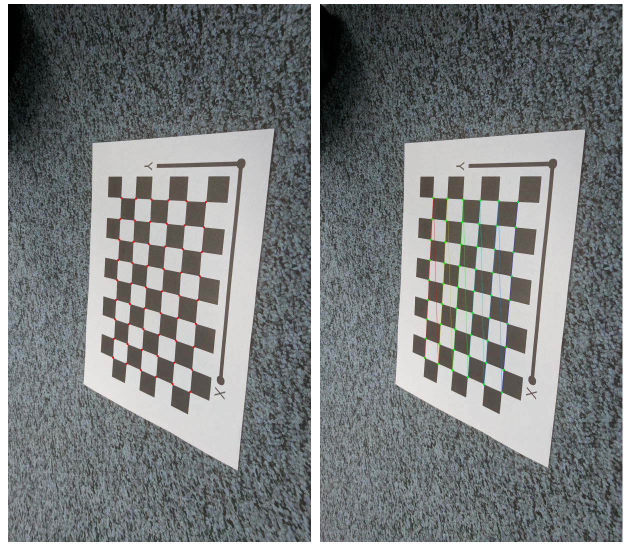

Fig. 13. Image12: Reprojected Corners and Corners

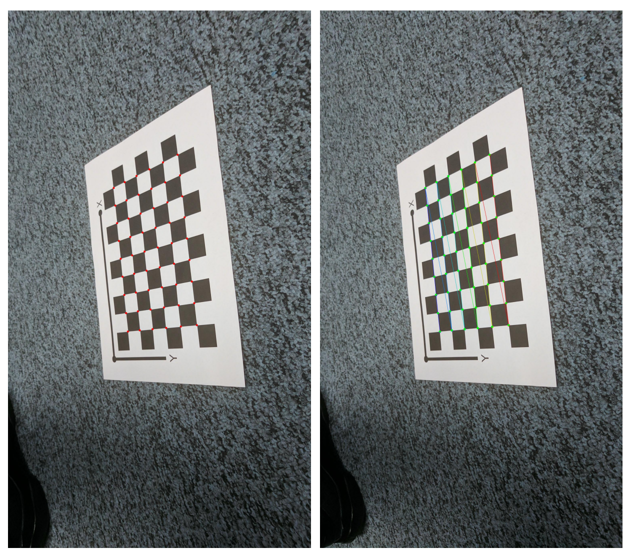

Fig. 14. Image13: Reprojected Corners and Corners