AutoCalib, HW-1

Shreyas Kanjalkar
Robotics Engineering Department
Worcester Polytechnic Institute
Worcester, USA
skanjalkar@wpi.edu

I. Introduction

Estimating parameters of the camera like the focal length, distortion coefficients and principle point is called Camera Calibration. It is quite time consuming and a very important part of any computer vision application involving 3D geometry, estimating depth, etc. An automatic way to perform efficient and robust camera calibration was presented by Zhengyou Zhang from Microsoft in this paper[1] and is considered one of the most groundbreaking papers in camera calibration.

II. Data

The data used for calibrating the camera is of a Chess Board which can be seen in Figure 1. Thirteen images of the model were taken from different view have been used in this homework. The reason to use chessboard is because Chessboard is extremely helpful since the dimensions of each square is known and thus we can find perfect corners.

III. Approach

The pipeline used for this homework follows the one given in Zhang's paper. It involves the following steps:

1) Corner detection using Chessboard
2) Estimating Intrinsic Camera parameters
3) Estimating Extrinsic Camera parameters
4) Approximate distortion coefficients
5) Non Linear Geometric error minimization

A. Chessboard corner detection

The chessboard pattern used for this camera calibration technique is a 9x6 excluding the borders. In order to find these corners we use the inbuilt function in opencv, *cv2.findChessboardCorners()*. The output is shown in figure 1.

The object coordinates in 3d are defined by taking the Z-coordinate as zero and the axis is assigned to one of the corner at the edge. The points are asssigned using meshgrid of 9x6 and multiplying the points with 21.5, although the length of the box can be chosen as anything arbitarily.

B. Estimating Camera Intrinsic parameters

The main purpose of the project is to ensure that there is a seamless mapping of the real world coordinates on to the image coordinates. To that we want to find the the most optimal value of the Intrinsic matrix (A) and the Extrinsic matrix (Rt). The relation between the image and the world coordinates is given by the transformation shown in the equation 1. Here, (u, v) are the image coordinates, s is the scaling factor, (X, Y) are the world coordinates. R$_i$, t$_i$ are the rotation and translation values of the image i.

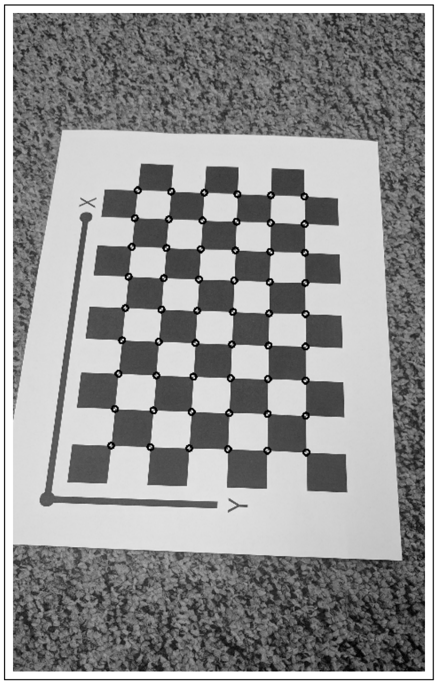

Fig. 1: Corners in the chessboard

$$s\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}A(R_i|t_i)\end{bmatrix}\begin{bmatrix}X\\Y\\0\\1\end{bmatrix} \tag{1}$$

where A is the intrinsic matrix consisting of the intrinsic parameters given by the equation

$$A = \begin{bmatrix}\alpha & \gamma & u_0\\0 & \beta & v_0\\0 & 0 & 1\end{bmatrix} \tag{2}$$

1) Finding homographies: Homography between image coordinates which are obtained from *cv2.findChessboardCorners()* and their respective world coordinates is obtained from *cv2.findHomography()* which finds the homography matrix required to transform the chessboard from world coordinate to image coordinate.

2) Finding b matrix: The b matrix in Zhang's paper is given as

$$b = \begin{bmatrix} B_{11} & B_{12} & B_{22} & B_{13} & B_{23} & B_{33} \end{bmatrix}^T \tag{3}$$

and the V matrix is a stack of

$$\begin{bmatrix} v_{12}^T (v_{11} - v_{22})^T) \end{bmatrix} \tag{4}$$

where v_{ij} is found using

$$\begin{aligned} v_{ij} = [&h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, \\ &h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]^T \end{aligned} \tag{5}$$

and solved using $Vb = 0$. After we get the b matrix we can solve the parameters inside the intrinsic matrix given by

$$v_0 = (B_{12}B_{13} - B_{11}B_{23})/(B_{11}B_{22} - B_{12}^2) \tag{6}$$

$$\lambda = B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})]/B_{11} \tag{7}$$

$$\alpha = \sqrt{\lambda/B_{11}} \tag{8}$$

$$\beta = \sqrt{\lambda B_{11}/(B_{11}B_{22} - B_{12}^2)} \tag{9}$$

$$\gamma = -B_{12}\alpha^2\beta/\lambda \tag{10}$$

$$u_0 = \gamma v_0/\beta - B_{13}\alpha^2/\lambda \tag{11}$$

From 6 — 10, the initial matrix that was estimated was:

$$A = \begin{bmatrix} 2.056e+03 & -1.017e+00 & 7.616e+02 \\ 0 & 2.040e+03 & 1.351e+03 \\ 0 & 0 & 1 \end{bmatrix} \tag{12}$$

C. Camera Extrinsic

After computing the homographies and the intrinsic camera parameters, we can find the extrinsic parameters using the equations below -

$$r_1 = \lambda A^{-1} h_1 \tag{13}$$

$$r_2 = \lambda A^{-1} h_2 \tag{14}$$

$$r_3 = r_1 \times r_2 \tag{15}$$

$$t = \lambda A^{-1} h_3 \tag{16}$$

where A is the intrinsic matrix, $\mathrm{h}_1, \mathrm{h}_2, \mathrm{h}_3$ are the column vectors of the Homography matrix for that image.

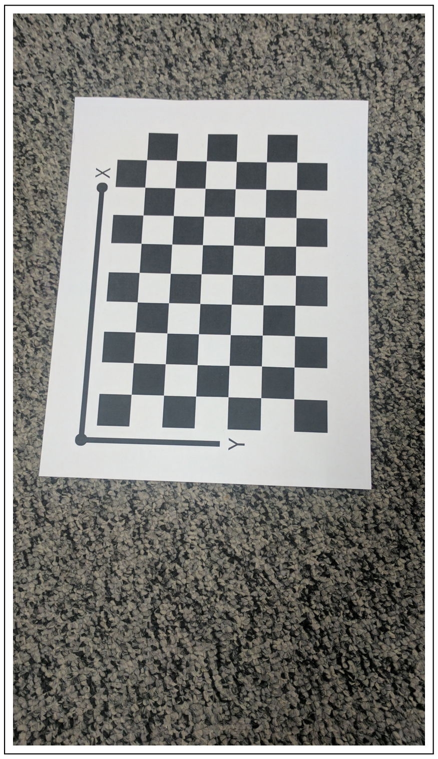

Fig. 2: Recitified Image example

D. Non-Linear Optimization

The optimization of the initial estimates is done using the function *scipy.optimize.leastsquares()*. It is the implementation of the non-linear algorithm called **Levenberg-Marquardt Algorithm**. The algorithm minimizes the following cost:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} ||\mathrm{m}_{ij} - \hat{\mathrm{m}}(\mathrm{A}, \mathrm{R}_i, \mathrm{t}_i, \mathrm{M}_j)||^2 \tag{17}$$

where m represents the points inside the image.

E. New Intrinsic Matrix

Now that we have new values for $\alpha, \beta, \gamma, u_0, v_0$ we get the new A matrix, which is

$$A = \begin{bmatrix} 2.056e+03 & -1.5113e+00 & 7.621e+02 \\ 0 & 2.040e+03 & 1.351e+03 \\ 0 & 0 & 1 \end{bmatrix} \tag{18}$$

and the distortion coefficients are:

$$\begin{aligned} k_1 &= 0.039 \\ k_2 &= -0.293 \end{aligned} \tag{19}$$

IV. Rectified images with corners

This section shows the rectified 13 images.

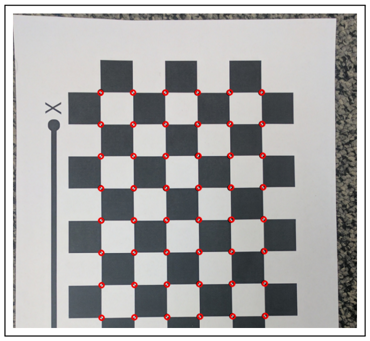
Fig. 3: Cropped Image to show distortion

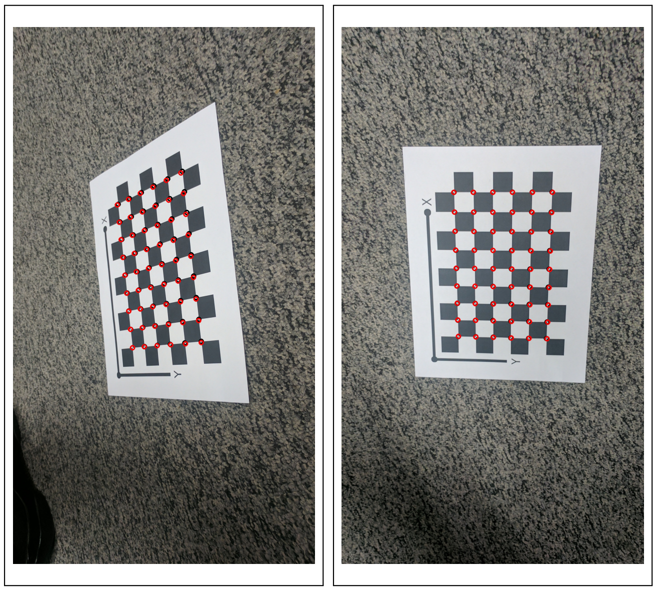
Fig. 4: Reprojected image 1 and 2

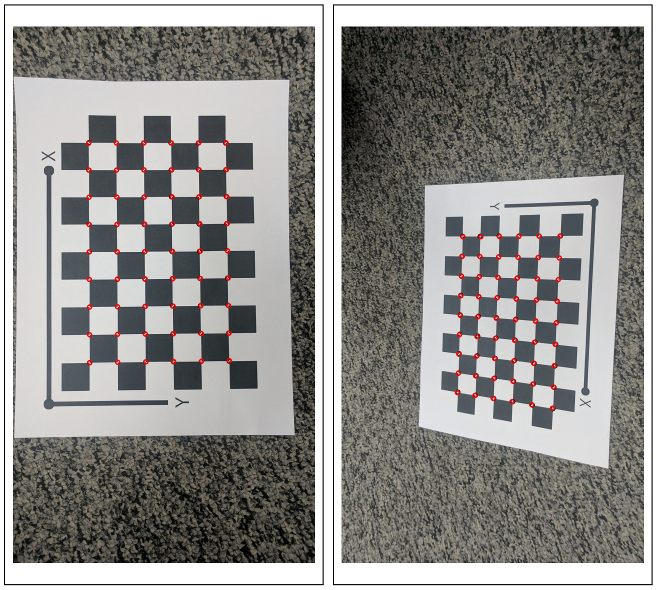
Fig. 5: Reprojected image 3 and 4

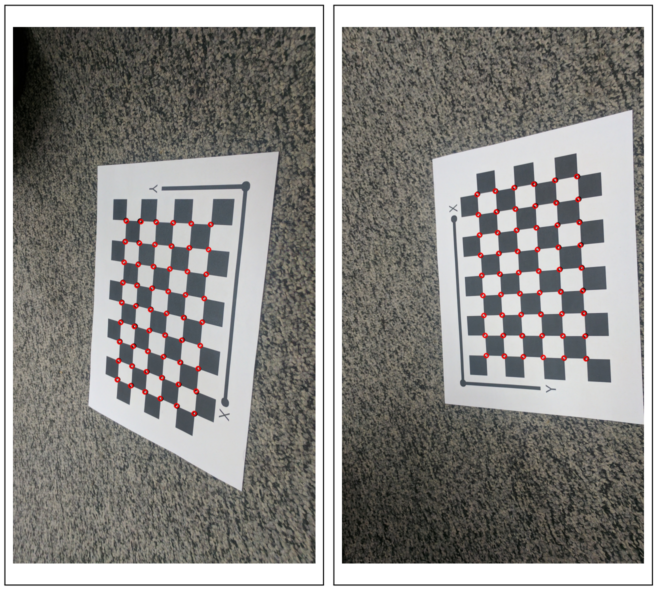
Fig. 6: Reprojected image 5 and 6

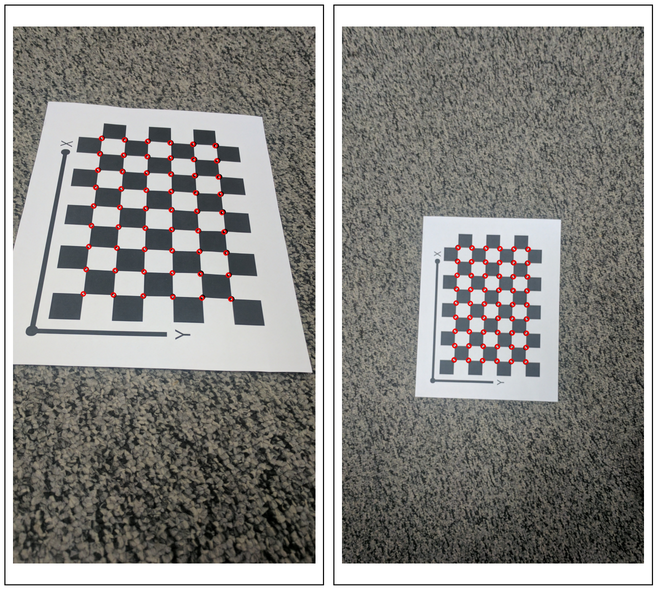
Fig. 7: Reprojected image 7 and 8

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.

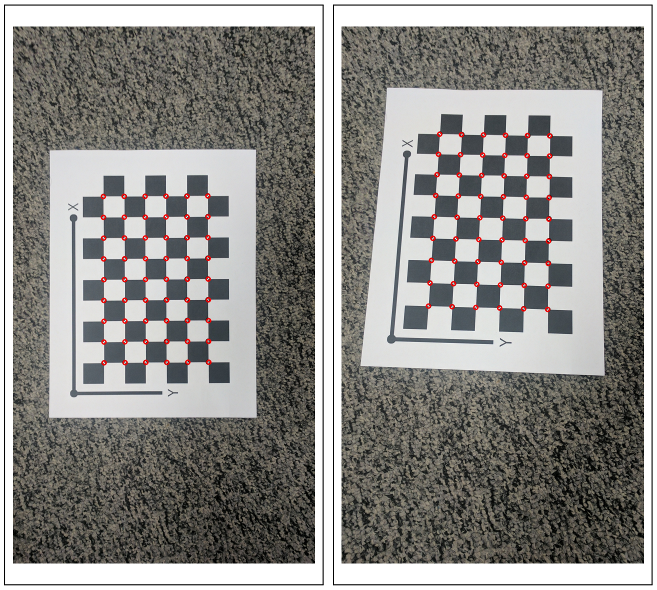

Fig. 8: Reprojected image 9 and 10

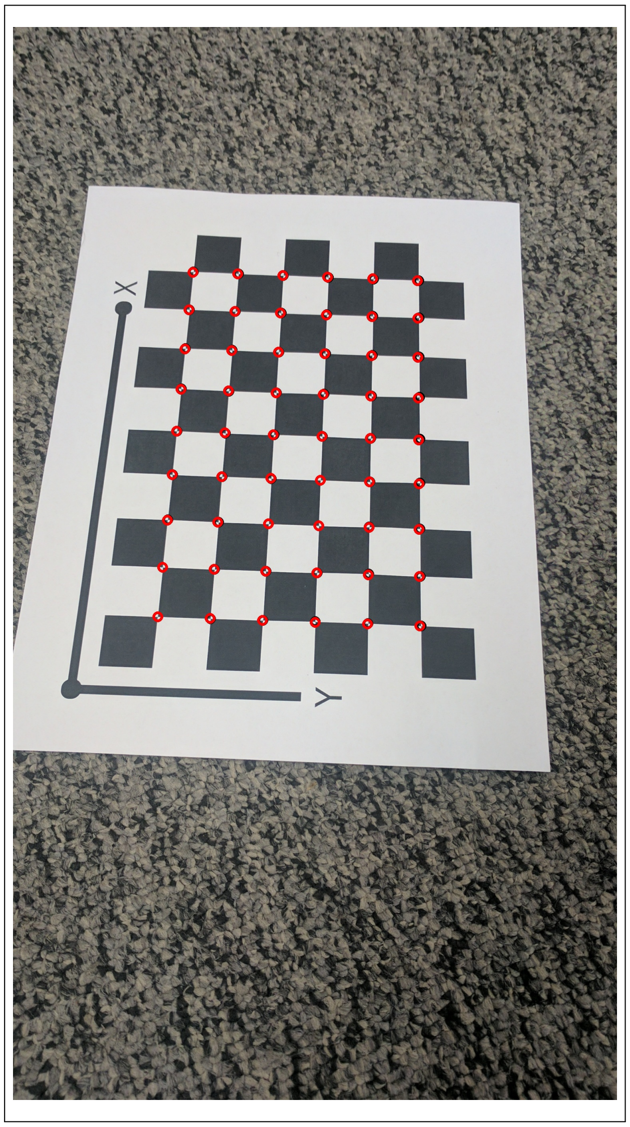

Fig. 10: Reprojected image 13

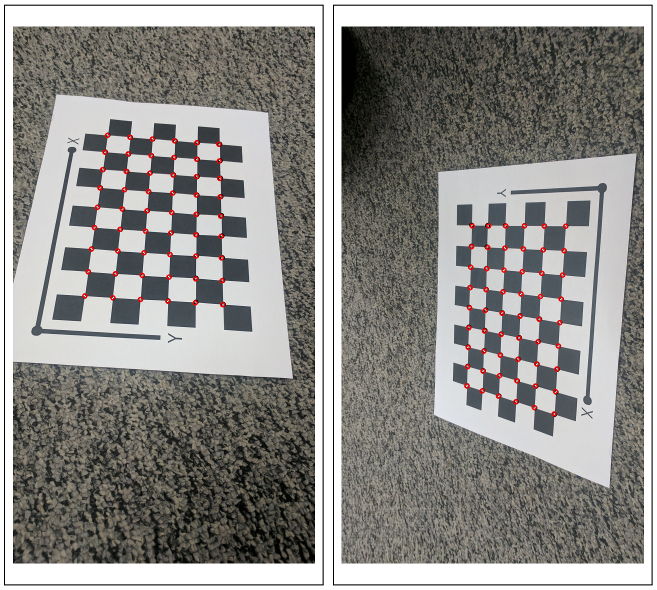

Fig. 9: Reprojected image 11 and 12