

# CMSC733 HW1 : AutoCalib

Param Dave  
Masters in Robotics  
University of Maryland , College Park  
Email: pdave1@umd.edu

## I. INTRODUCTION

Camera calibration is an important process of any computer vision research involving 3D geometry. Here we use zhang's calibration method to solve for the same. We estimate intrinsic and extrinsic parameters and distortion parameters by using 13 images checkerboard clicker at different angles and distances. We estimate the values by solving equations given in the paper. We then use a non linear optimizer to minimize the geometric error.

The following steps are there in the homework.

- 1) Corner detection of checkerboard
- 2) Calculating initial intrinsic parameters
- 3) Calculating initial extrinsic parameters
- 4) Approximating distortion coefficient
- 5) Minimizing Non-Linear Geometric error

## II. CORNER DETECTION OF CHECKERBOARD

The checkerboard pattern used for the camera calibration technique is 9x6 excluding the borders. Each square is 21.5 x 21.5 mm in dimension. The corners are detected using opencv's inbuilt function cv2.findChessboardCorners. The output is shown in figure 1.

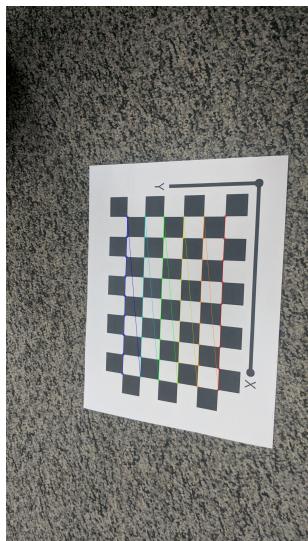


Fig. 1. Corners Detected in a Checkerboard Pattern

## III. ESTIMATING INTRINSIC PARAMETERS

The world coordinates for each corner are calculated by numbering the corners from 0 to 8 in x coordinates and 0 to 5 in y coordinates. Thus the top left corner will have a coordinate of (0,0) and bottom left will have a coordinate of (9,6). As each tile has length and width of 21.5cm , we multiply each coordinate with 21.5 to get all corner coordinates. I used this coordinates and image coordinates to compute homography for each image. After that I followed the section 3.1 to find the camera intrinsic parameters. The initial parameters after calculating is shown in equation 1.

$$\begin{bmatrix} 2197.7 & -4.7875 & 755.35 \\ 0 & 2184.93 & 1333.87 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

## IV. ESTIMATING CAMERA EXTRINSIC PARAMETERS

For each image we calculate the extrinsic parameters. We calculate the rotation and translation with respect to model coordinates. I followed the method as shown in section 3.1 .We get a 3x4 matrix which contains 3 rotation column vectors and 1 translation column vector. After calculating the camera extrinsic parameters I calculated mean re-projection error which is equal to 0.9295.

## V. OPTIMIZATION AND COMPUTING DISTORTION COEFFICIENT $K_c$

The next step after computing the extrinsic and intrinsic parameters is to do optimization. Initially, value of radial distortion is taken as (0,0). To minimize the reprojection error, scipy.optimize.leastsquare is used.

After the optimization, the distortion coefficients come out to be:  $k_1 = 0.024$  and  $k_2 = -0.1413$

## VI. RESULTS

After optimization,The final camera matrix and radial distortion coefficients are given below in equation.

$$\begin{bmatrix} 2197.67 & -4.7878 & 754.85 \\ 0 & 2184.75 & 1335.15 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$k_1 = 0.024$  and  $k_2 = -0.1413$  The mean projection error after optimization comes out to be : 0.891

## VII. OUTPUT

Using the above extrinsic and intrinsic parameters, the corner points are reprojected , both the corner points and reprojected point are shown in figures below.

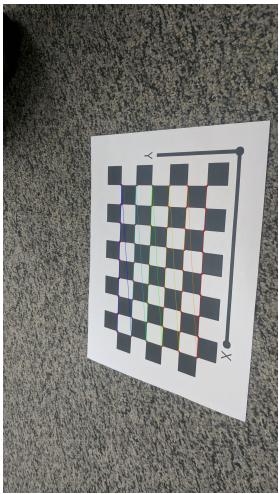


Fig. 2. Re projected points of Image 1

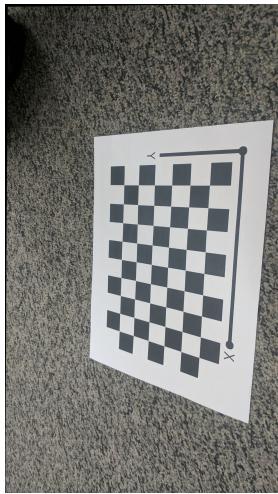


Fig. 3. Rectified image 1

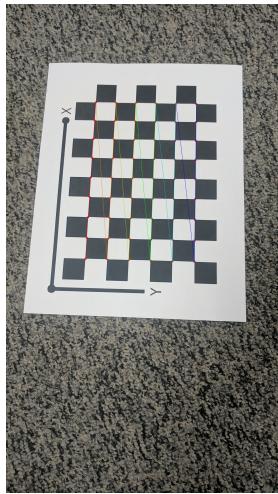


Fig. 8. Re projected points of Image 4

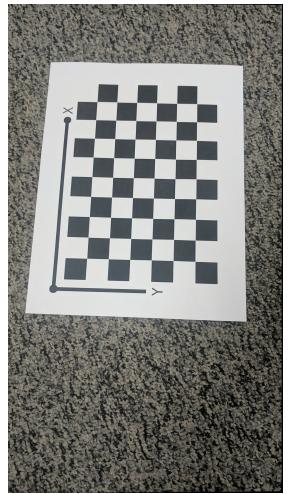


Fig. 9. Rectified image 4.

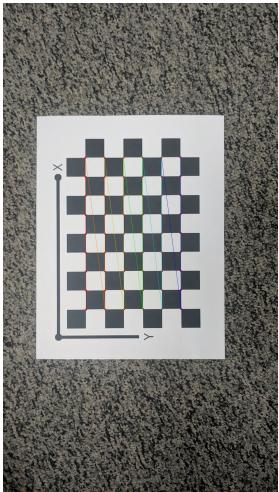


Fig. 4. Re projected points of Image 2

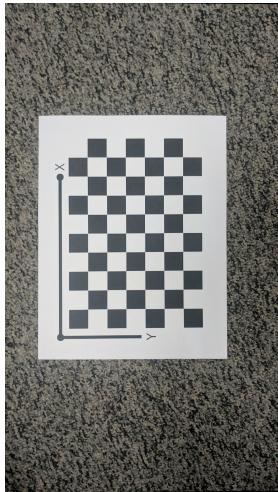


Fig. 5. Rectified image 2.



Fig. 10. Re projected points of Image 5



Fig. 11. Rectified image 5.

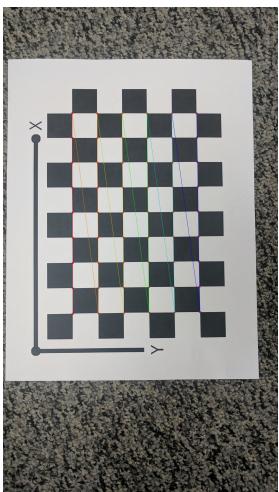


Fig. 6. Re projected points of Image 3

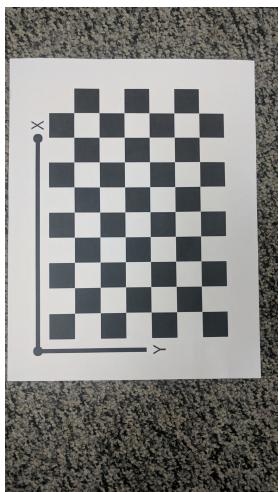


Fig. 7. Rectified image 3.



Fig. 12. Re projected points of Image 6

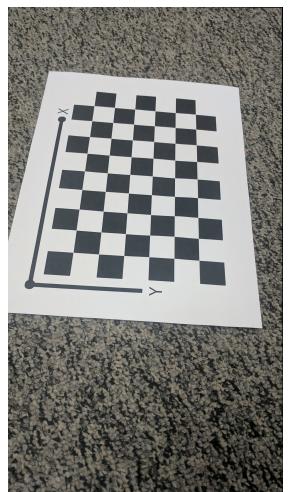


Fig. 13. Rectified image 6.

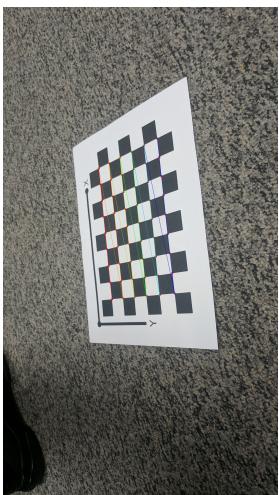


Fig. 14. Re projected points of Image 7

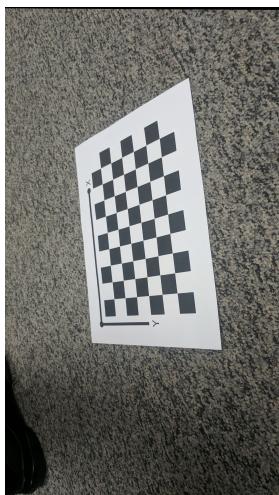


Fig. 15. Rectified image 7.

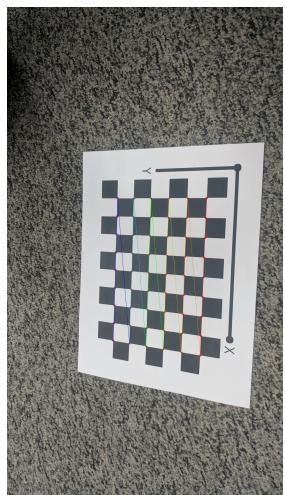


Fig. 20. Re projected points of Image 10

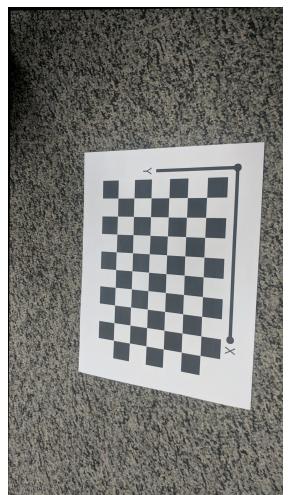


Fig. 21. Rectified image 10.

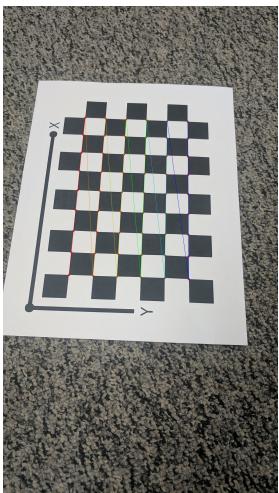


Fig. 16. Re projected points of Image 8

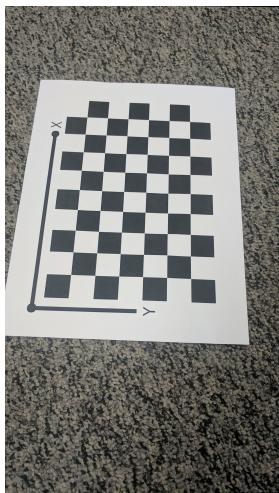


Fig. 17. Rectified image 8.

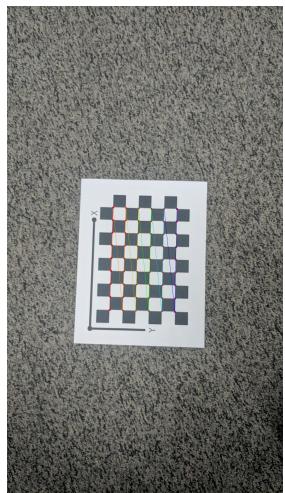


Fig. 22. Re projected points of Image 11

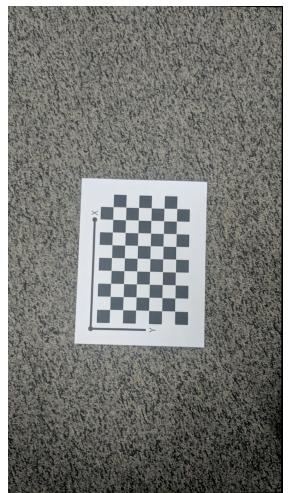


Fig. 23. Rectified image 11.

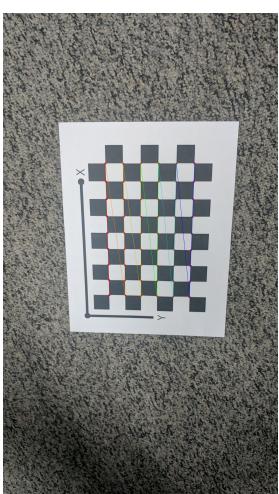


Fig. 18. Re projected points of Image 9

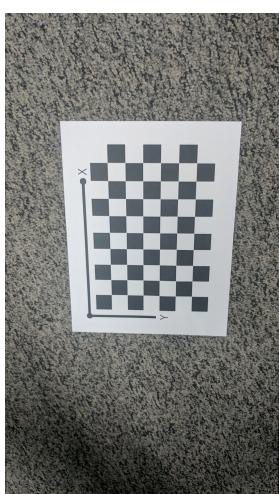


Fig. 19. Rectified image 9.

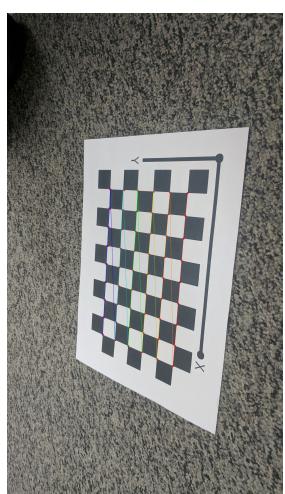


Fig. 24. Re projected points of Image 12

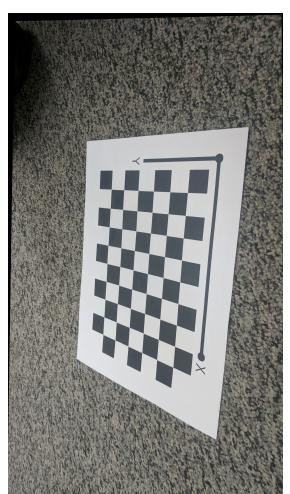


Fig. 25. Rectified image 12.

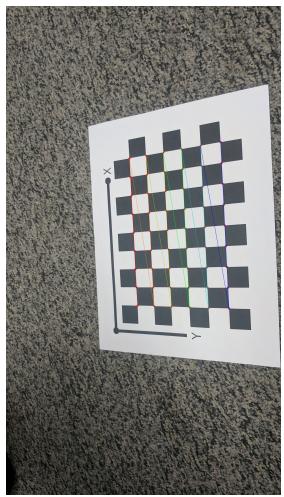


Fig. 26. Re projected points of  
Image 13

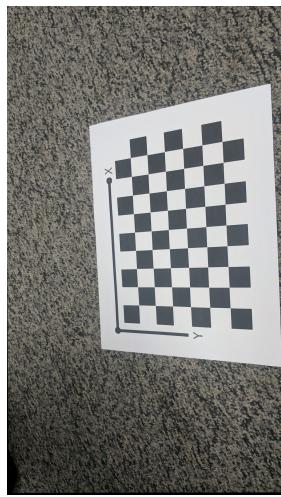


Fig. 27. Rectified image 13.