AutoCalib

Venkatesh Mullur
Robotics Engineering Department,
Worcester Polytechnic Institute,
Worcester, MA, USA.
vmullur@wpi.edu

I. Introduction

The report explains Camera calibration in the Zhengyou Zhang's paper by calculating the intrinsics and the extrinsics of the camera. To calibrate the camera, we used checkerboard patterns. Specifically we used 13 checkerboard patterns with different perspectives having locked focus. Main steps to solve this problem is given below.

1) Corner detection on the checkerboards
2) Find Instrinsics
3) Find Extrinsics
4) Approximate distortion coefficients
5) Non linear geometric error minimization.

II. Corner detection on the checkerboards

The checkerboard corner detection is done using pre-written $cv2$ function called $chessboardcorners$. The checkerboard pattern in the physical world was known that is we were known that each square is 21.5 mms apart. Basically we knew the checkerboard pattern in the world, so that is an assumption we are making here.

The checkerboard has 9 rows and 6 columns and I found the corners using the above mentioned function. I found out the world coordinates since we know they are 21.5 mm apart.

III. Estimating Intrinsic Matrix

Before finding the intrinsic matrix, I need to find out the homography between the world coordinates which is calculating from the physical information and the corners in all the 13 images. Initially since, we were used to finding 4 point homography, I tried finding the homography using just 4 points, but it was always better to make use of all the 54 points on the checkerboard (6*9). Thus I got 13 homography matrices of 3 by 3. Now, to find the intrinsic matrix, I calculated a matrix which contains information between the homographies.

$$Hi = \begin{bmatrix} h1 & h2 & h3 \end{bmatrix}$$

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

By using these two identities, and the below given identity and using SVD, I could find the B matrix through which all the initial parameters are estimated.

$$V.B = 0$$

Fig. 1. Initial Parameters

SVD is used after this point to get the B matrix. From the B matrix, I could find the all the initial parameters. They are shown in the figure 1

IV. Estimate Extrinsics

Using the estimated intrinsic matrix and all the homography matrices and the initial parameters, I computed the Rotation and the translation matrix for each image. $\lambda = 1/(\|A^{-1}h1\|)$ where h1 is given above as the first column of the homography matrix. The rotation matrix found by this method has some errors since it does not satisfy the properties of the rotation matrix, hence SVD of the transpose of the rotation matrix is found and then the errorless rotation matrix is the dot product of the U and V^T of the SVD's. Initial estimates of the A and the paramters are shown in the figure 1

$$A = \begin{bmatrix} 2.05278817e+03 & -3.69529711e-01 & 7.63061089e+02 \\ 0.00000000e+00 & 2.03138417e+03 & 1.35261456e+03 \\ 0.00000000e+00 & 0.00000000e+00 & 1.00000000e+00 \end{bmatrix}$$

$$fx = 2052.788166884138$$

$$lambda = -0.6331722760245666$$

$$vo = 1352.61456117404$$

$$fy = 2036.6341494517958$$

$$uo = 763.0610889577671$$

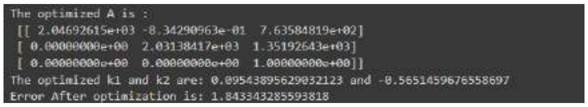
$$gamma = -0.36952971148561514$$

V. Distortion Coefficients

Initially $k1$ and $k2$ = 0.

A. Non Linear Geometric error Minimization

The initial error I got was 1.9675292755920457. Using the estimates of both the extrinsics and the intrinsics, I want to optimize them. These parameters are optimized on the projection error such that distance between the projected point and the ground truth should be as minimum as possible. To do this, *scipy optimize* is used. The optimized parameters are given below.



```
The optimized A is :
 [[ 2.04692615e+03 -8.34290963e-01  7.63584819e+02]
 [ 0.00000000e+00  2.03138417e+03  1.35192643e+03]
 [ 0.00000000e+00  0.00000000e+00  1.00000000e+00]]
The optimized k1 and k2 are: 0.09543895629032123 and -0.5651459676558697
Error After optimization is: 1.843343285593818
```

Fig. 2.

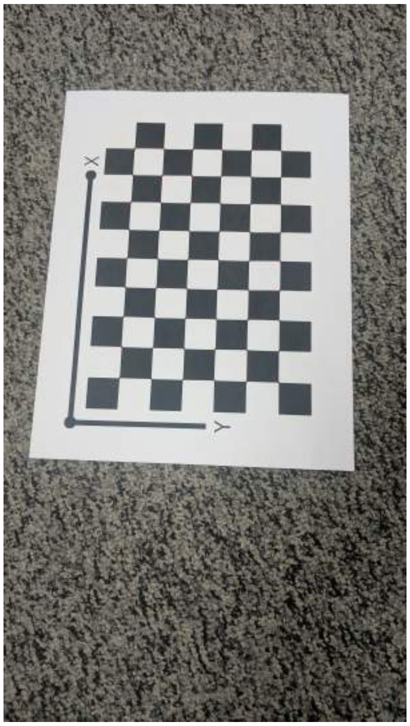


Fig. 3.

$$A = \begin{bmatrix} 2.04692615e+03 & -8.34290963e-01 & 7.63584819e+02 \\ 0.00000000e+00 & 2.03138417e+03 & 1.35192643e+03 \\ 0.00000000e+00 & 0.00000000e+00 & 1.00000000e+00 \end{bmatrix}$$

$$k1 = 0.09543895629032123$$

$$k2 = -0.5651459676558697$$

The error I got after optimization was 1.843343285593818.

VI. Conclusion

Camera Calibration is done in this project. Initially we found the homographies between the real world points and the images corners, then I found the intrinsic matrix and the extrinsic matrix which is the transformation matrix and then found the reprojection error. After finding the error, we need to take the radial distortion constants $(k1andk2)$ to 0 and then find the optimized parameters and recalculate the radial distortion constants. The green dots in the images are the corners which are found by $cv2$ and the red are the reporjected.

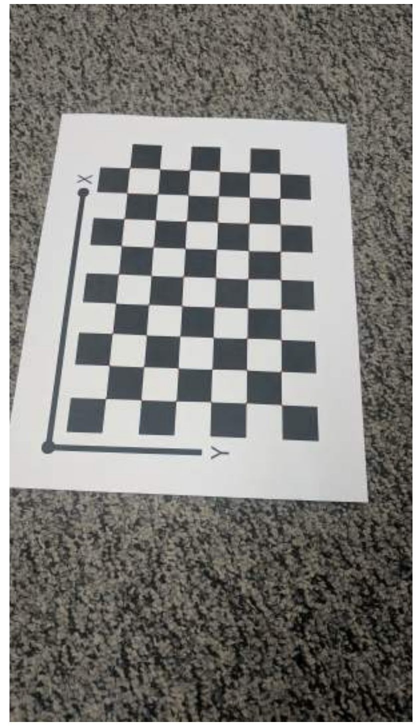


Fig. 4.

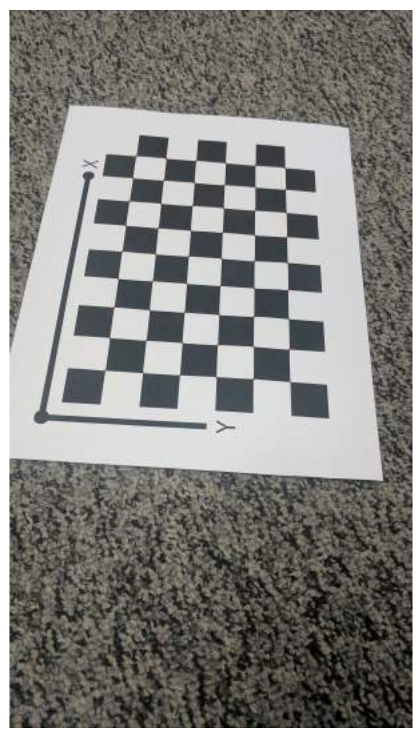


Fig. 5.

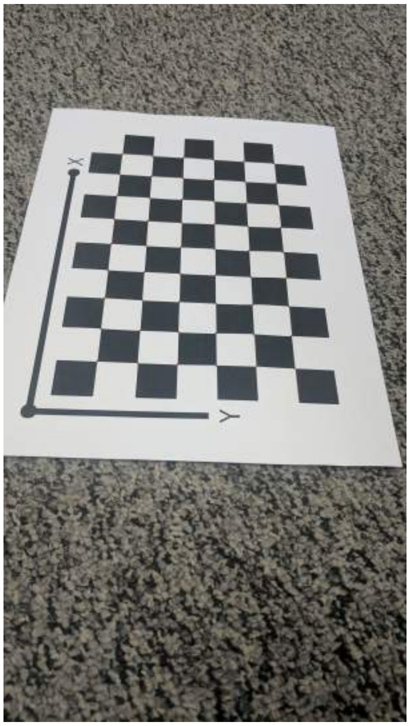


Fig. 6.

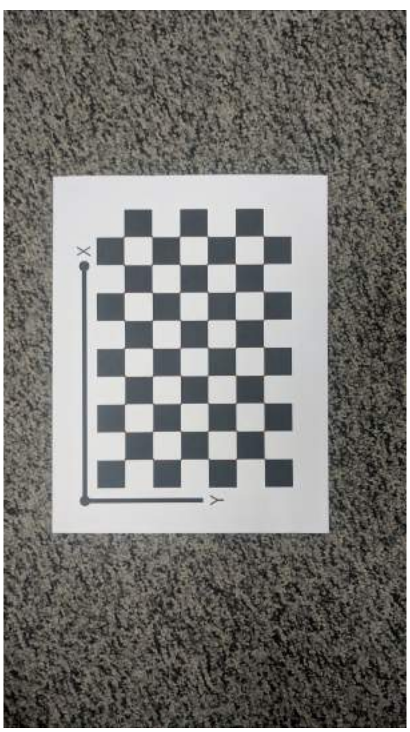


Fig. 7.

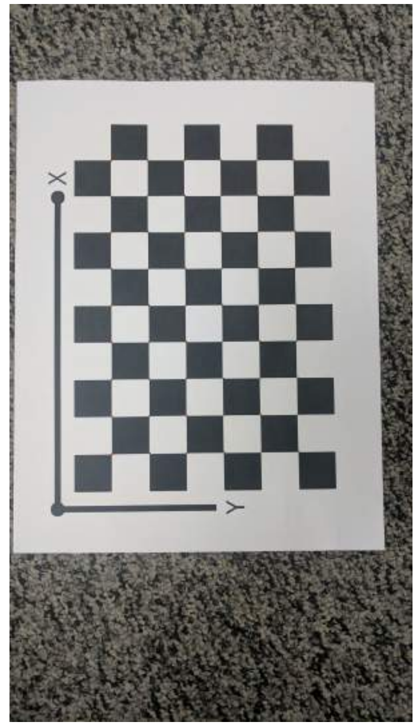


Fig. 8.

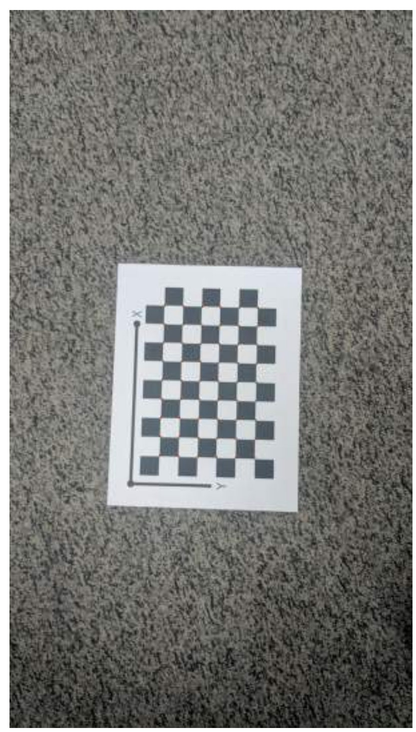


Fig. 9.

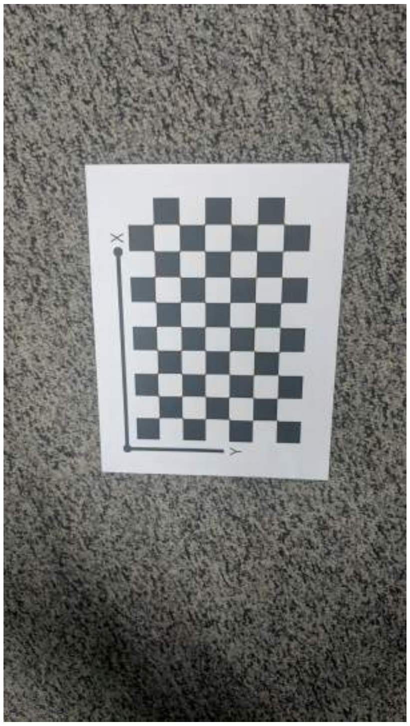


Fig. 10.

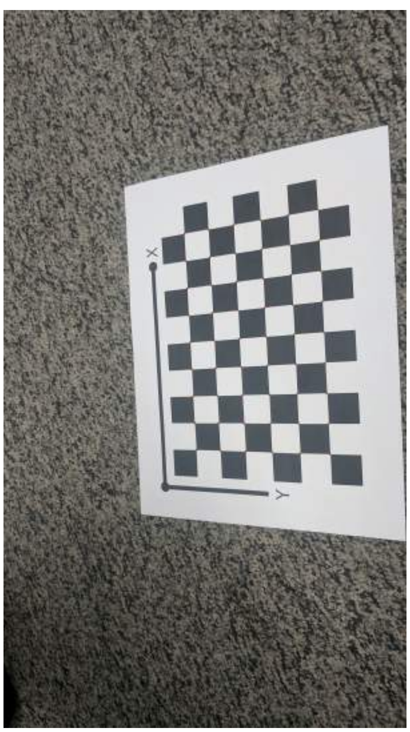


Fig. 11.

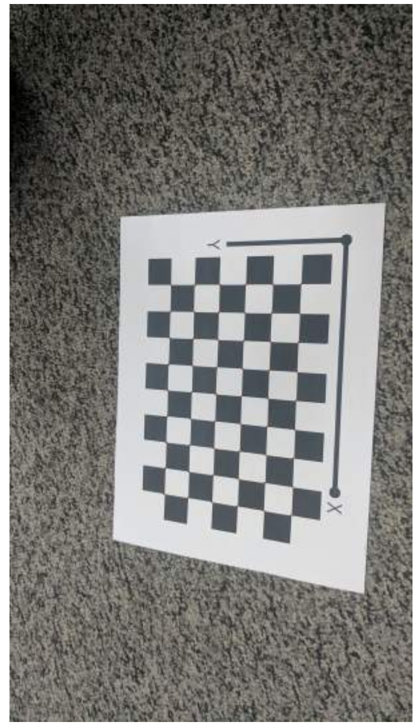


Fig. 12.

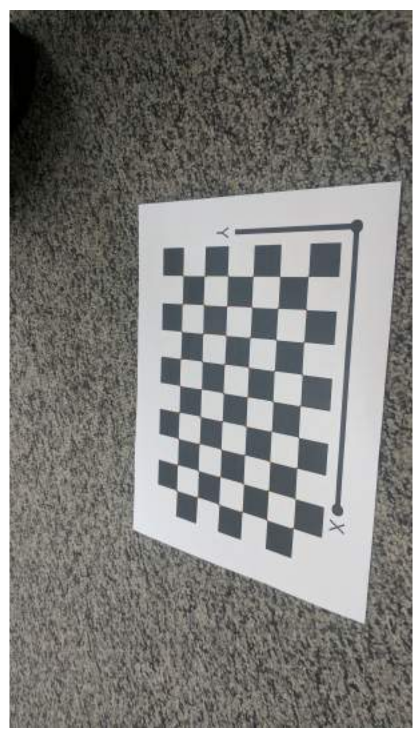


Fig. 13.

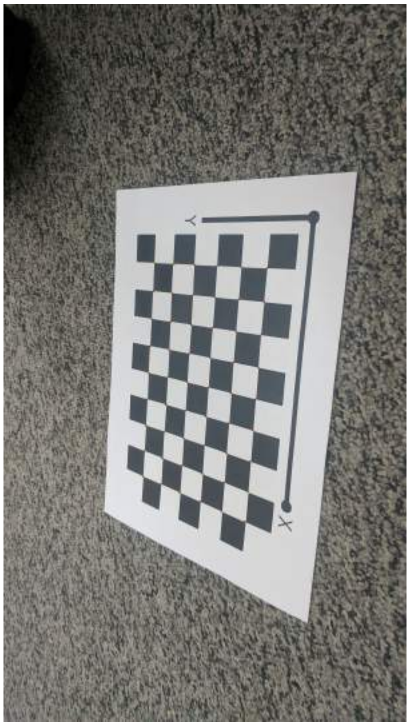


Fig. 14.

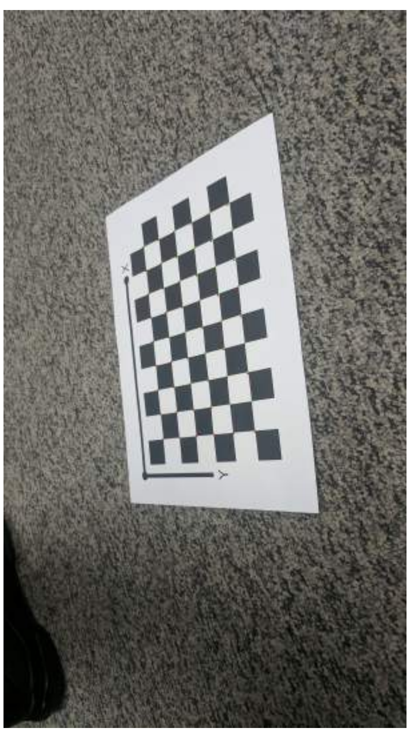


Fig. 15.

REFERENCES

[1] https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr98-71.pdf

[2] https://github.com/SrinidhiSreenath/AutoCalib-Caibration-of-Camera/blob/master/Wrapper.py

[3] https://cmsc733.github.io/assets/2019/hw1/results/pdf/patilameya$_{h}w1.pdf$