CMSC733: Homework 1 - AutoCalib

Bhargav Kumar Soothram
Email : bsoothra@umd.edu

I. Introduction

Camera calibration can be understood as the process of obtaining the properties intrinsic (internal) to a camera, ie., its focal length, principle point, skewness and distortion coefficients. There are many calibration techniques and while these techniques have their merits, they bring with themselves their share of disadvantages. Zhang in his paper "**A Flexible New Technique for Camera Calibration**" proposed a technique for calibration that has been widely adopted since. Here in this project, we aim to implement Zhang's Calibration technique using a bunch of checkerboard images taken from different perspectives using a phone.

II. Implementation

The first step in the process of calibration is the determination of the intrinsic matrix A, given by

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

A. Homography Estimation

The closed-form solution can be computed by determining a matrix *B* expressed as $B = A^{-T}A^{-1}$, where B can in turn be determined by computing the homography between the world and image coordinate frames. So, we will first find homography matrices for all the images. To find the homography between a pair of images, we will need at least four point correspondences and these point correspondences can be obtained by detecting the checkerboard corners using $cv2.findChessboardCorners()$.

B. Finding B matrix

Since the B matrix is a symmetric one, it only has six DOF and let us call these B_{11}, B_{12}, B_{13}, B_{22}, B_{23}, B_{33}. We can find these six elements using the homography matrices obtained from the previous step.

C. Finding the Intrinsic Matrix A

Now that the six independent elements of the B matrix are known, we can compute the individual components of A can be determined using a series of equations described in Zhang's paper [1]. The initial estimate of the A matrix is given below:

$$\begin{bmatrix} 2.05631965e+03 & -1.10745234e+00 & 7.61492229e+02 \\ 0.00000000e+00 & 2.04093962e+03 & 1.35025616e+03 \\ 0.00000000e+00 & 0.00000000e+00 & 1.00000000e+00 \end{bmatrix}$$

D. Obtaining the Extrinsic Matrices Rt

The extrinsic parameters can be obtained once the calibration matrix is found, using the set of equations given in [1].

E. Distortion Coefficients

Distortion coefficients k_1 and k_2 were intialized to 0.

F. Optimizing the parameters

The last step in the pipeline is to optimize the camera parameters and this implies that the reprojection error has to be minimized. This was done using the $scipy.optimize$ using the least squares approach, but understanding and implementing this took a while, because of the way I had defined my cost function earlier (although it was trivial at the end).

III. Results

The obtained results for each of the aforementioned parameters are shown in the image below:

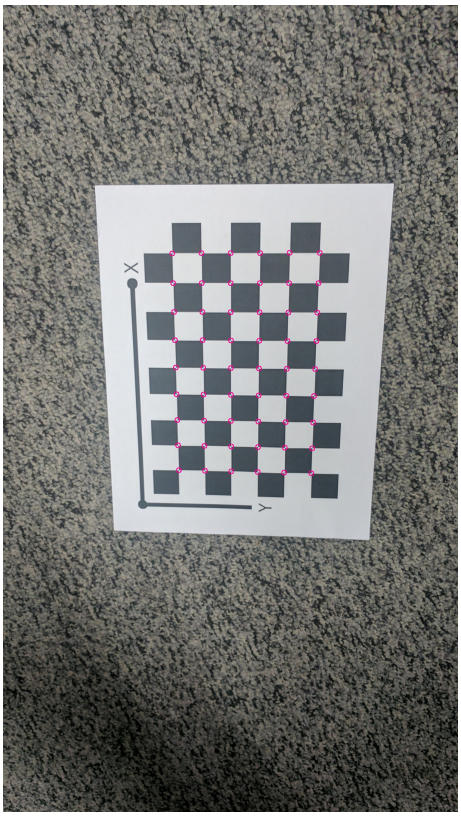
```
Initial estimate for the A matrix:
 [[ 2.05631965e+03 -1.10745234e+00  7.61492229e+02]
 [ 0.00000000e+00  2.04093962e+03  1.35025616e+03]
 [ 0.00000000e+00  0.00000000e+00  1.00000000e+00]]
Error before optimization:  0.6980139862313565
A matrix after optimization:
 [[ 2.05631599e+03 -1.10880325e+00  7.61497340e+02]
 [ 0.00000000e+00  2.04093070e+03  1.35026969e+03]
 [ 0.00000000e+00  0.00000000e+00  1.00000000e+00]]
Distortion coefficients obtained:  [ 0.01344836 -0.09806747]
Mean error after optimization:  0.6822532066912671
```

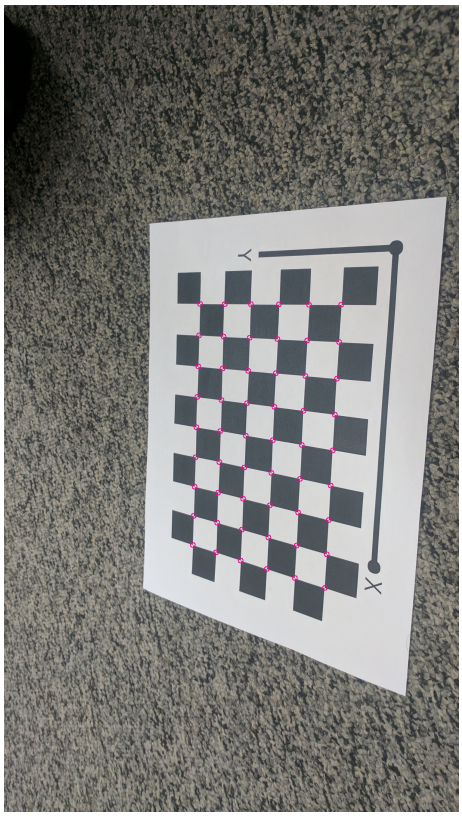
Fig. 1: Results obtained

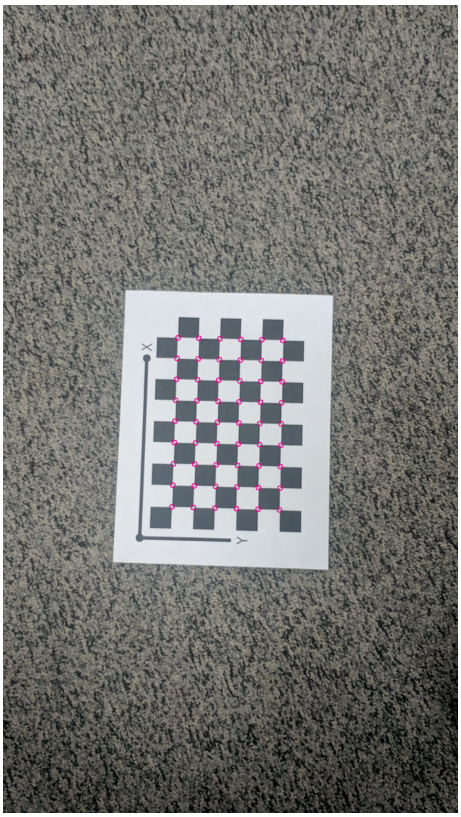
Now that all the parameters have been determined, including the distortion coefficients, we can use them to perform distortion correction using $cv2.undistort$ and then plot the re-projected points on them. The resulting images are shown in the figures below. Since there is not much distortion that has been detected, the resulting images are not that different from the original ones.

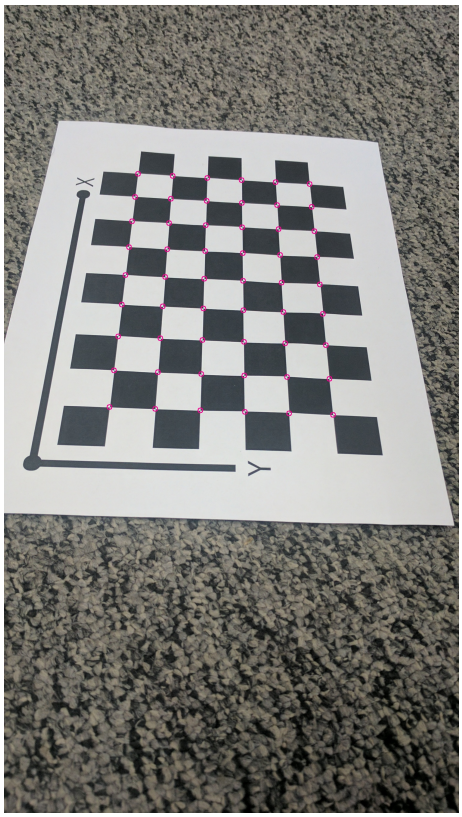
References

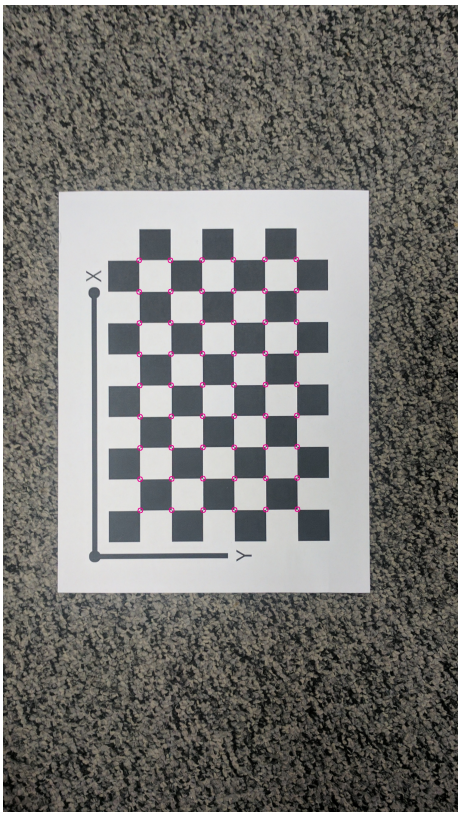
[1] https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr98-71.pdf
[2] https://opencv-python-tutroals.readthedocs.io/en/latest/pytutorials/pycalib3d/pycalibration/pycalibration.html
[3] https://www.youtube.com/watch?v=2XM2Rb2pfyQlist=PL2zRqk16wsdoCCLpoudGo7QQNks1Ppzoindex=4
[4] https://cmp.felk.cvut.cz/cmp/courses/dzo/resources/tutorial-pollefeys-eccv/node89.html
[5] https://www.youtube.com/watch?v=-9He7Nu3u8s

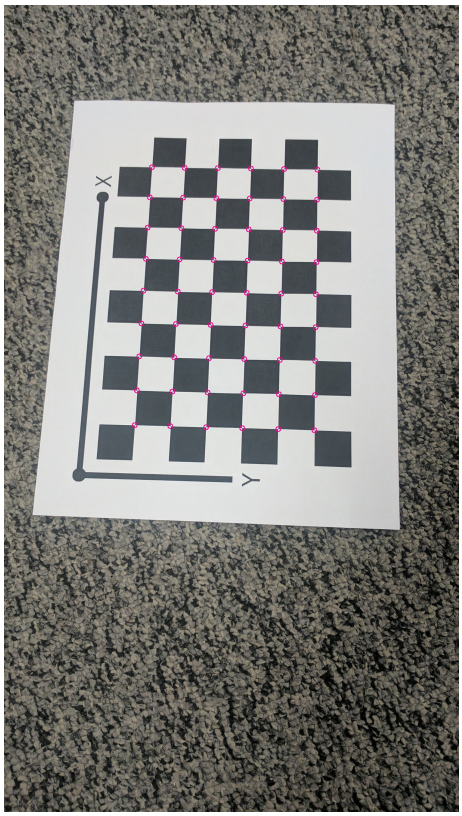

X
Y

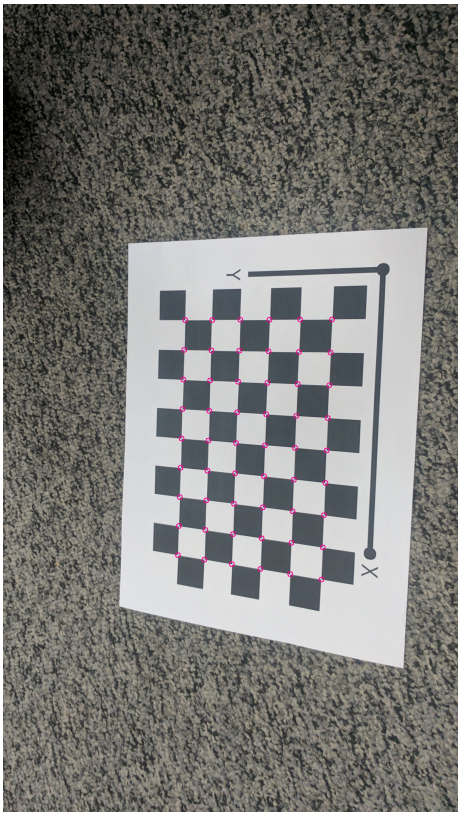

Y
X

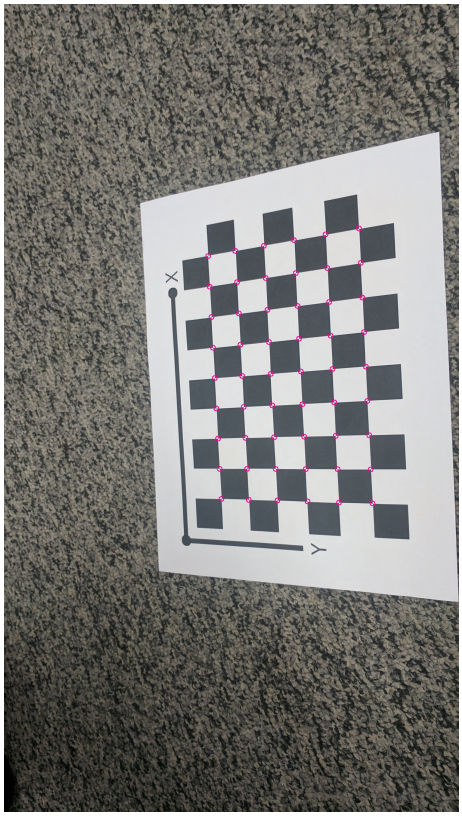

X
Y

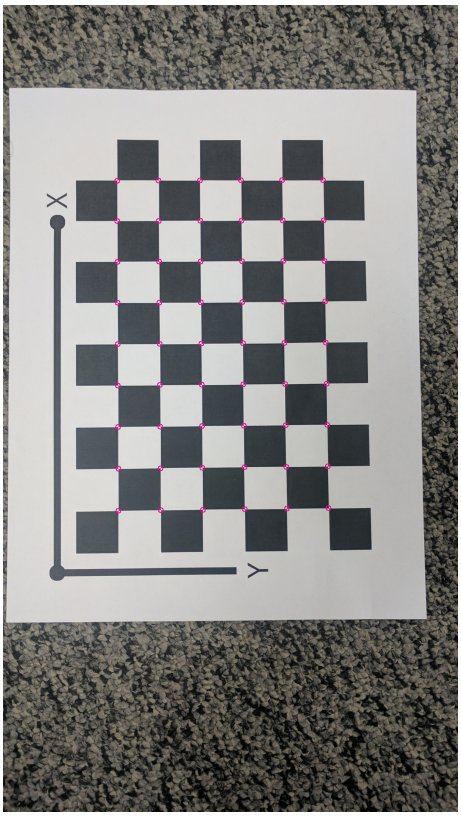

X
Y

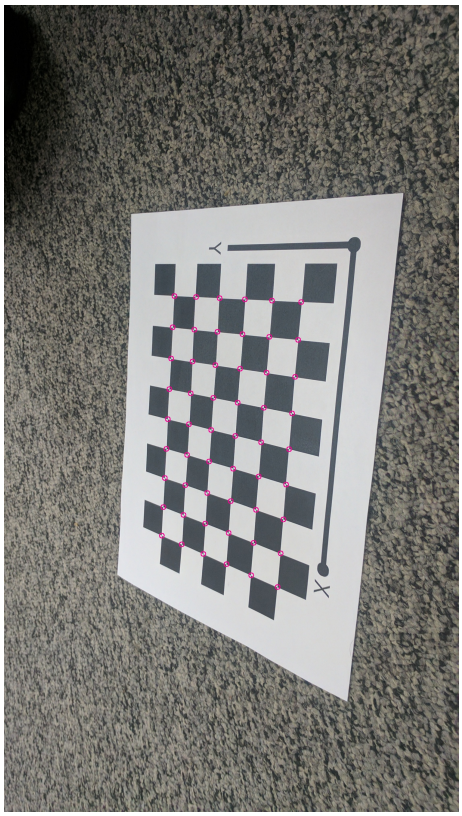

X
Y

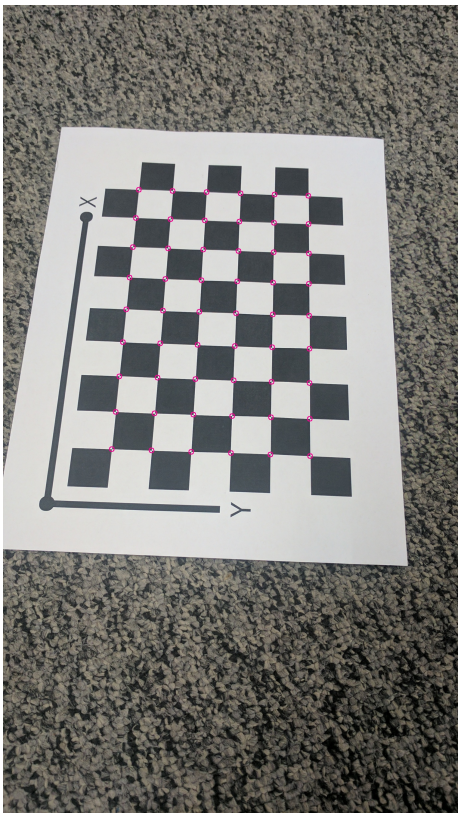

X
Y

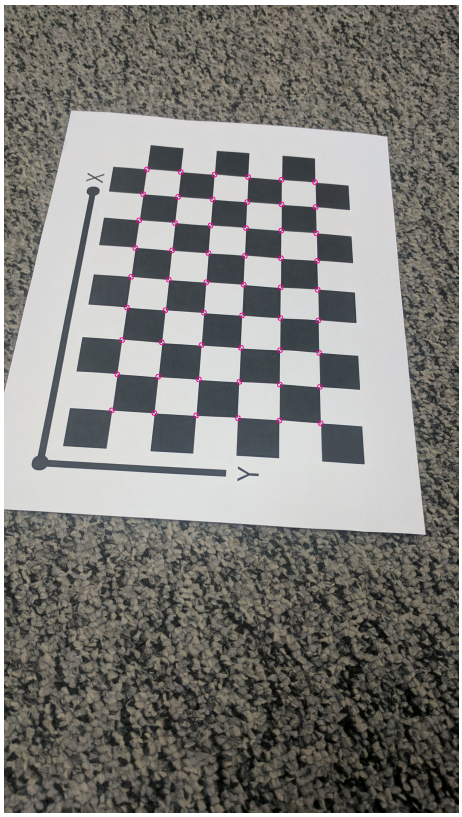

Y
X

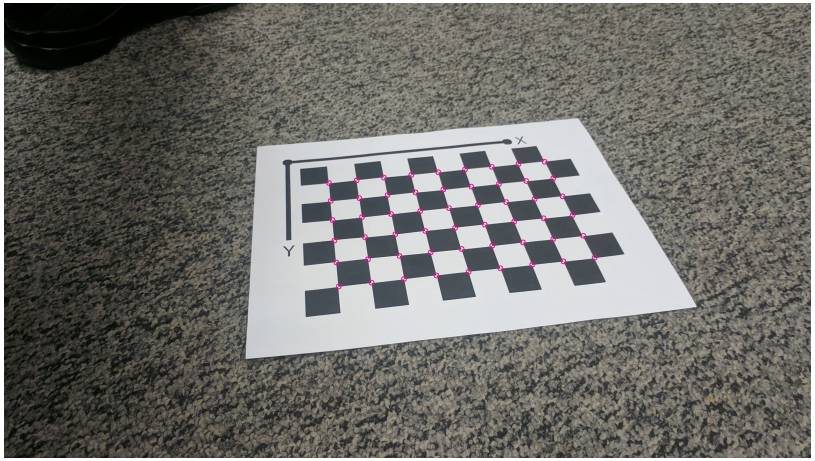

X
Y


X
Y


Y
X


X
Y


X
Y


X
Y