Computer Vision Home Work 1 - AutoCalib

Butchi Venkatesh Adari
Robotics Engineering
Worcester Polytechnic Institute
Worcester, Massachusetts 01609
Email: badari@wpi.edu

***Abstract*—This paper details the software implementation of a camera calibration algorithm. The utilized method is described in "A flexible new technique for camera calibration" by Z. Zhang.**

I. Introduction

We follow the method detailed by Zhengyou Zhang to estimate the parameters of camera calibration matrix. This method is robust and quite efficient. The data utilized are images taken using a Google Pixel XL camera with focus lock enabled. The subject of the images is the checkerboard pattern of a chessboard. The vertices of the pattern serve as our calibration points basis.

The localization of objects in 3D space viewed by a camera has many important applications in computer vision and robotics engineering. This requires two reference frame transformations, first to transform between the 2D pixel coordinates on the image sensor plane to the camera's 3D frame, then from the camera frame to the 3D world coordinates. The matrices representing these transforms are known as the camera intrinsic and extrinsic matrices.

A. Data Preparation and Solving for Calibration Matrix

Using cv2.findChessboardCorners, the vertices of the chessboard are found for all greyscaled images in our dataset. Using the method discussed in the paper by Zhangyou Zhang, initial rotation (R) and translation (t) matrices calculated. We can also initially assume that the camera has minimal distortion and we can assume that

$$\text{k} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

After calculation, the initial K matrix is shown in Fig 1.

This is an approximate solution which can be further refined/fine-tuned.

B. Non-linear Geometric Error Minimization

Now that we have the initial estimates of K, R, t, and k matrices, we can try to minimize the geometric error. We can use scipy.optimize to minimize using the loss function as given below:

After optimizing, the optimized K matrix and the optimized distortion matrix k are shown in Fig 1.

The reprojection error before optimizing is 0.697577 and the reprojection error after optimizing is 0.6833688.

$$\sum_{i=1}^{N}\sum_{j=1}^{M} ||x_{i,j} - \hat{x}_{i,j}\left(K, R_i, t_i, X_j, k\right)||$$

The outputs generated from the code are as follows. The A matrix from the analytical computation is:

```
(AI) abven@abven-ubuntu:~/CV_Course/badari_hw1$ python Wrapper.py
[[ 2.46608379e+03 -1.14440612e+00  7.62413505e+02]
 [ 0.00000000e+00  2.44647026e+03  1.34707151e+03]
 [ 0.00000000e+00  0.00000000e+00  1.00000000e+00]]
Optimizing ...
Updated K matrix
[[ 2.46607881e+03 -1.14709130e+00  7.62421729e+02]
 [ 0.00000000e+00  2.44645822e+03  1.34708901e+03]
 [ 0.00000000e+00  0.00000000e+00  1.00000000e+00]]
Updated Distortion matrix
[[ 0.01439521]
 [-0.14644599]]
Reprojection Error Before Optimization:  0.6975773337712492
Reprojection Error After Optimization:  0.6833688424426175
```

Fig. 1: The output of the reprojection

II. Output

The checkerboard image after rectification using the matrices calculated above is shown in Fig. 1

$$\sum_{i=1}^{N}\sum_{j=1}^{M} ||x_{i,j} - \hat{x}_{i,j}\left(K, R_i, t_i, X_j, k\right)||$$

Fig. 2: Loss function for optimization

III. Results

In the Images the blue circles indicates the corners before reprojected and the red circles indiates after reprojection.

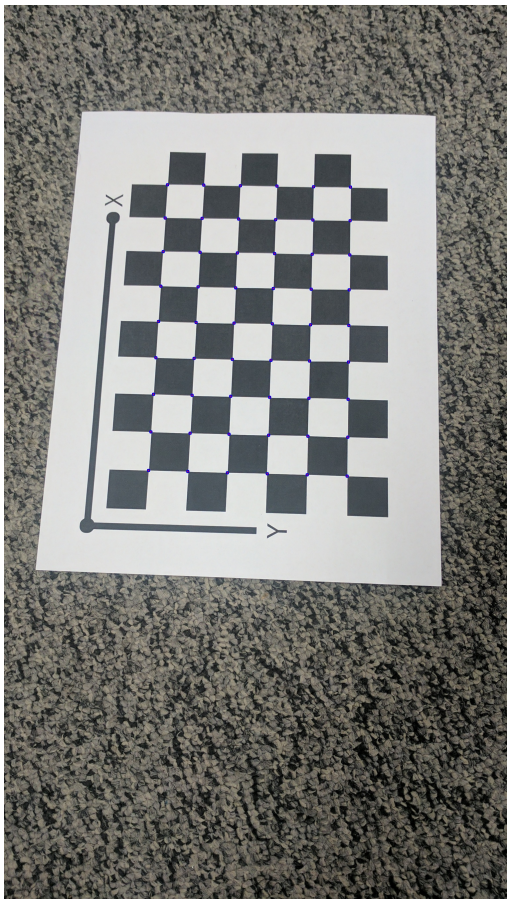

Fig. 3: Image 1 After calibration

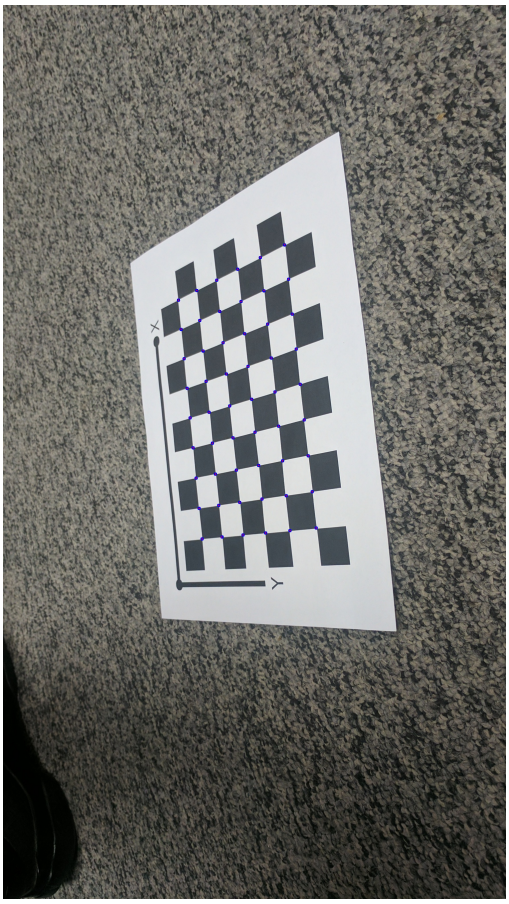

Fig. 4: Image 2 After calibration

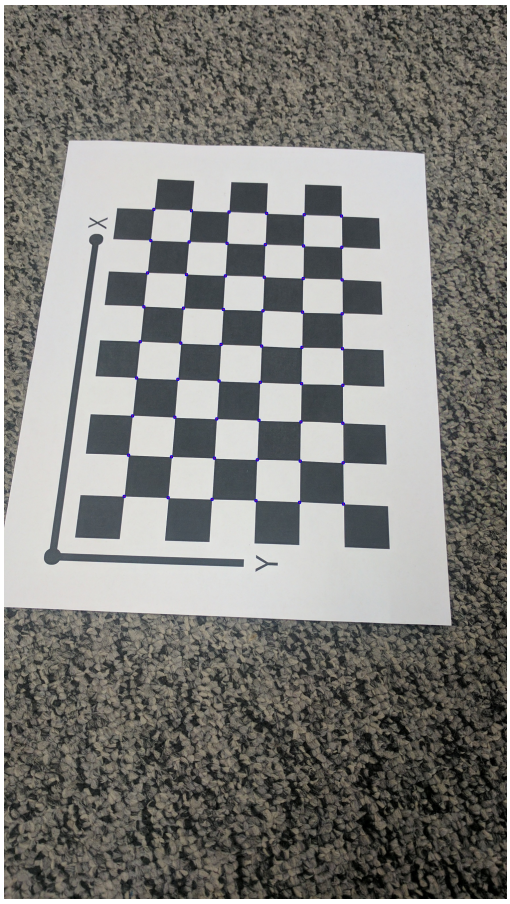

Fig. 5: Image 3 After calibration

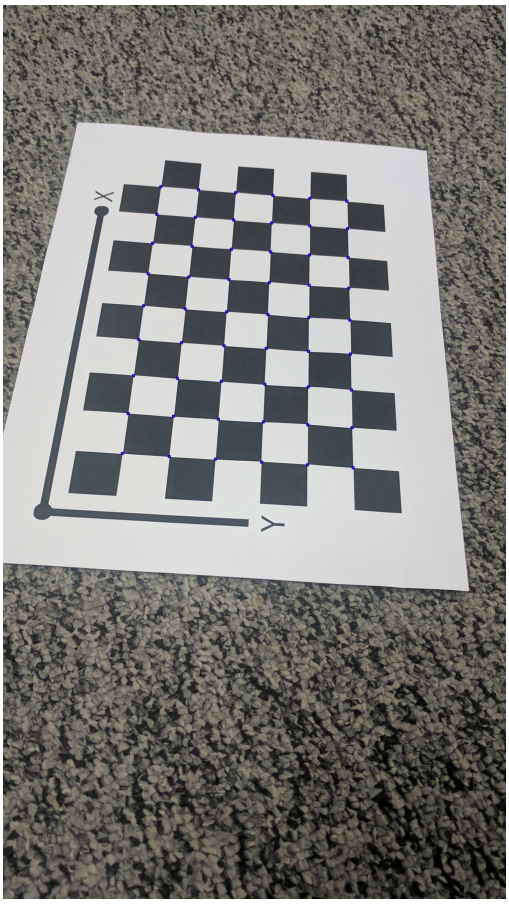

Fig. 6: Image 4 After calibration

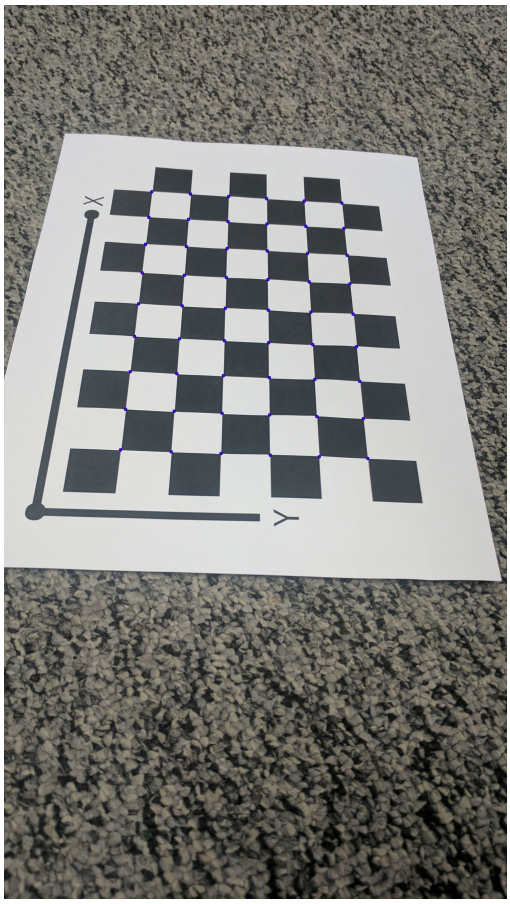

Fig. 7: Image 5 After calibration

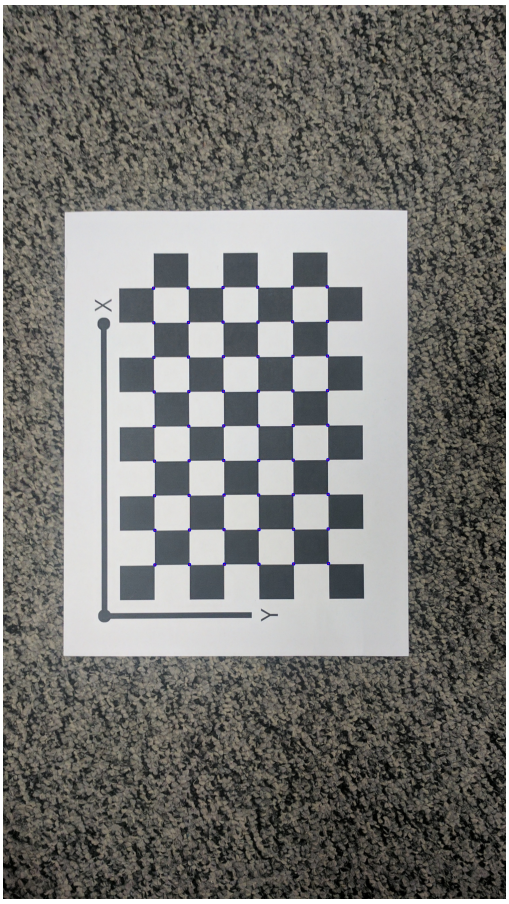

Fig. 8: Image 6 After calibration

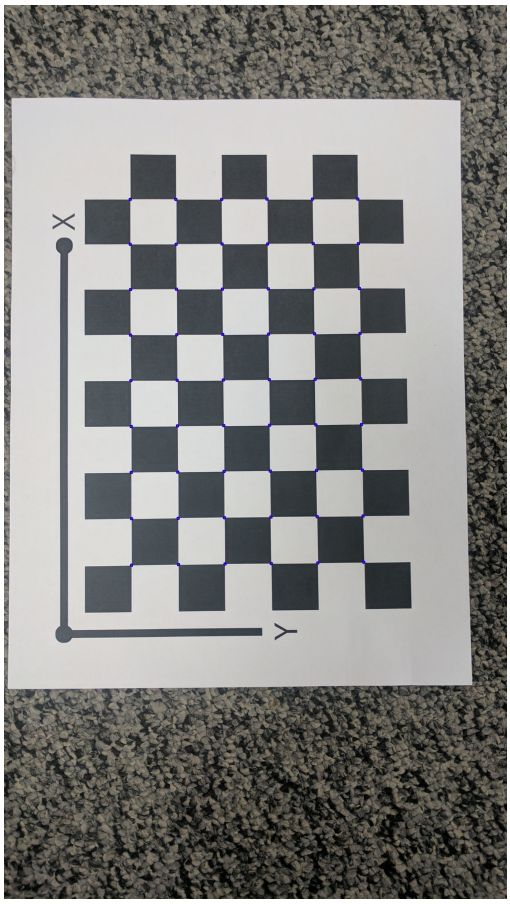

Fig. 9: Image 7 After calibration

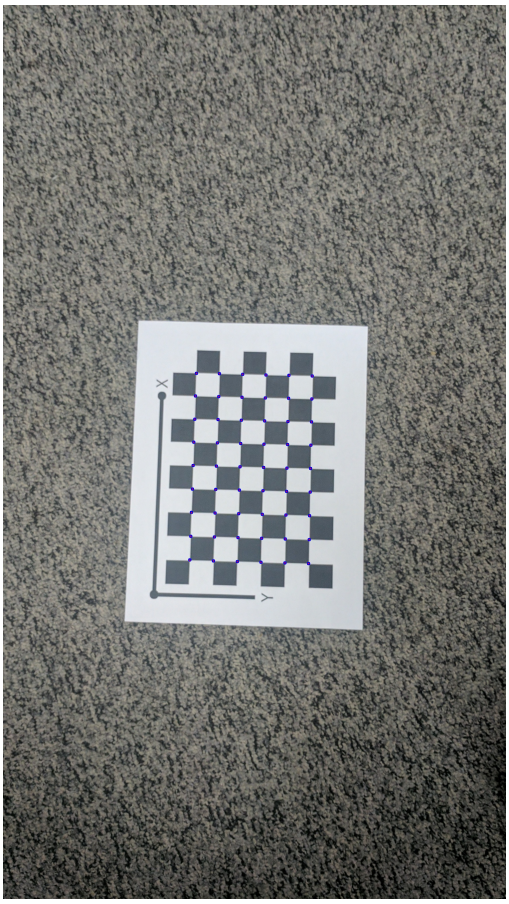

Fig. 10: Image 8 After calibration

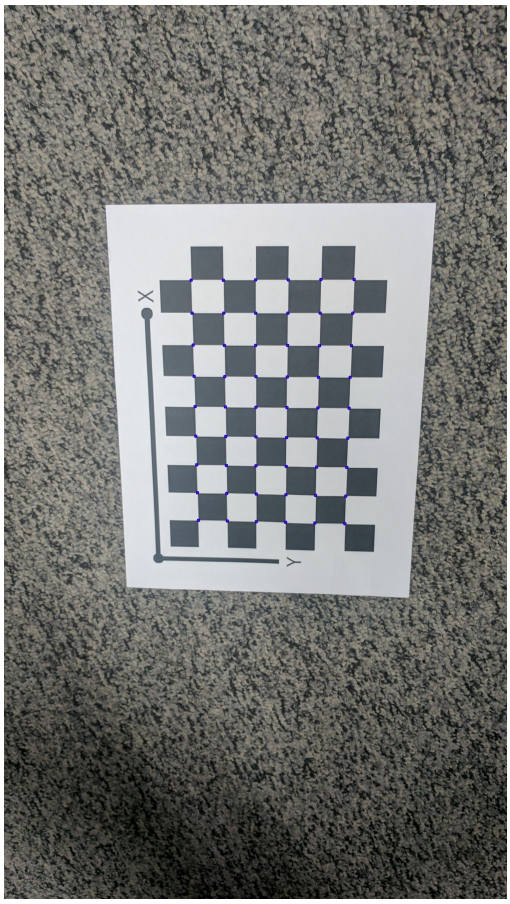

Fig. 11: Image 9 After calibration

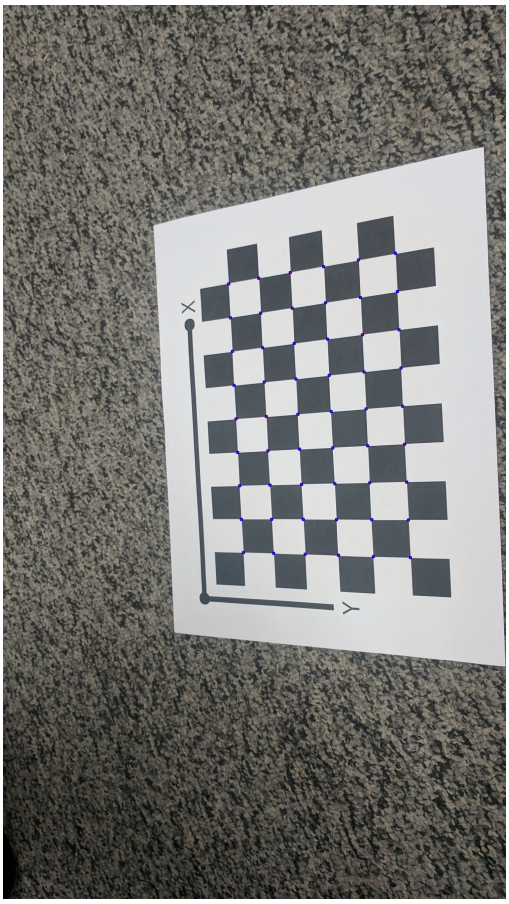

Fig. 12: Image 10 After calibration

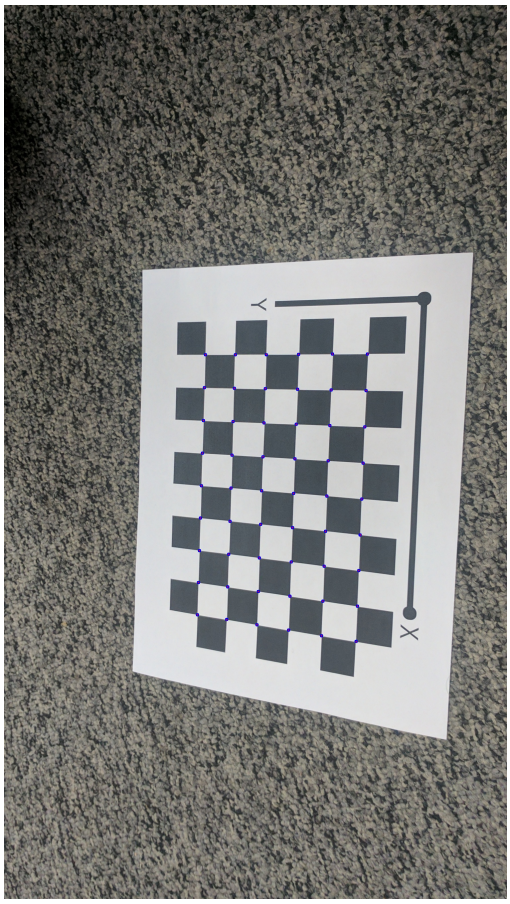

Fig. 13: Image 11 After calibration

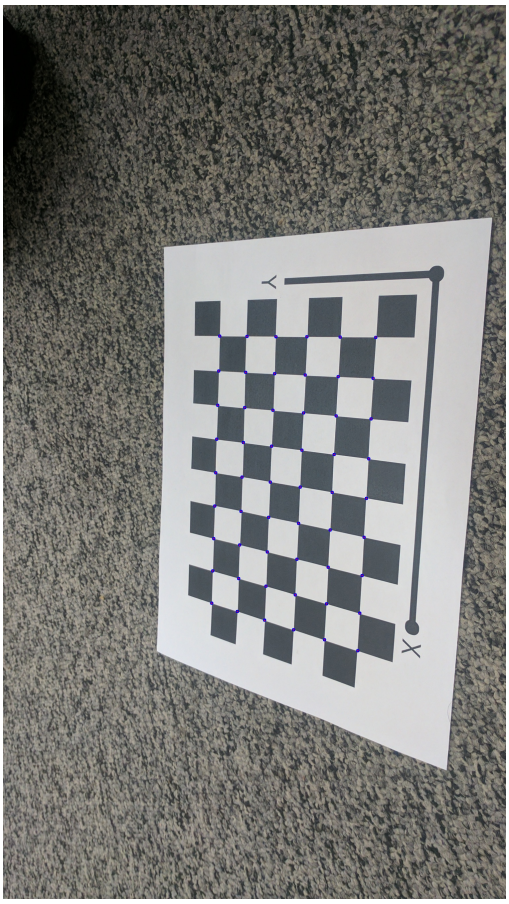

Fig. 14: Image 12 After calibration

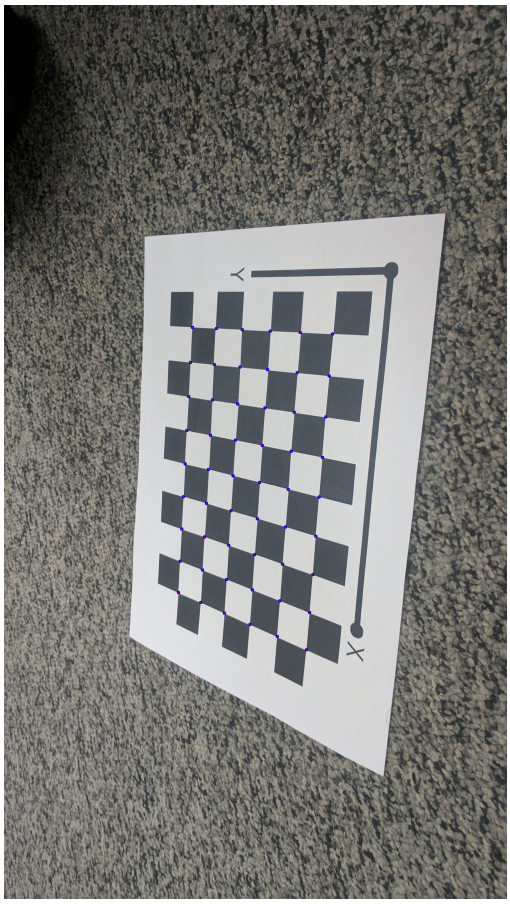

Fig. 15: Image 13 After calibration