CMSC 733 AutoCalib

Aditya Vaishampayan
116077354
Meng. Robotics
adityav@terpmail.umd.edu

***Abstract* - The goal of this homework is to implement Camera Calibration technique proposed by Zhang's paper. 13 images of Checkerboard pattern are used for this calibration. We use the equation given in the Zhang's paper for obtaining the intrinsic and extrinsic matrices The parameters are optimized using a non-linear geometric error minimization.**

I. CAMERA CALIBRATION

The pipeline used for this homework is given in Zhang's paper. It involves the following steps:
1. Corner Detection of Checkerboard pattern
2. Estimate Intrinsic Camera Matrix
3. Estimate Camera Extrinsic
4. Approximate distortion coefficient
5. Non-Linear Geometric error minimization

A. Checkerboard Corner Detection

The checkerboard pattern used for this camera calibration technique is 9x6 excluding the borders and is shown in Figure 1. We use an opencv built in function called cv2.findChessboardCorners for finding the corners for checkerboard pattern. The output is shown in Figure 2.

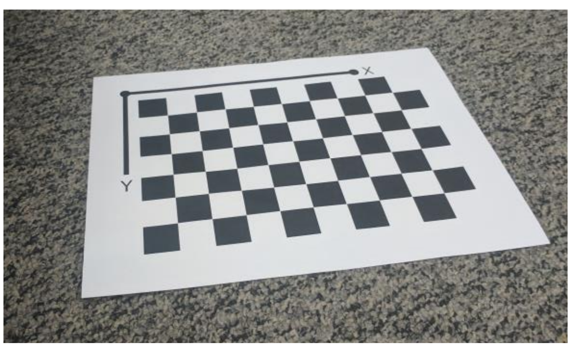

Fig. 1. Input checkerboard pattern

The 3D object coordinates are defined by taking Z coordinate as zero and the axis is assigned to one of the corners at edge. We assign the points using the mesh grid function having 9*6 grid size. We do not multiply the points with the square length as it does not matter.

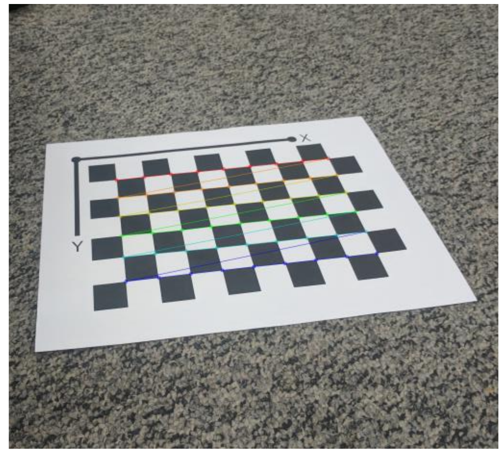

Fig. 2 Input checkerboard pattern

B. Estimating Camera Intrinsic Matrix

The relationship between a 3D point and its image projection is given by://

$$s\tilde{\mathbf{m}} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}]\widetilde{M} \tag{1}$$

In the above equation, K is camera matrix and it is given by://

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

Our goal is to find these parameters. The initial estimation of these parameters are computed using a closed form solution from section 3.1 of Zhang's paper.//

$$\mathrm{B} = \mathrm{K}^{-T}\mathrm{K}^{-1} \equiv \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} \tag{3}$$

$$\begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2\beta} & \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} \\ -\frac{\gamma}{\alpha^2\beta} & \frac{\gamma^2}{\alpha^2\beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0\gamma - u_0\beta)^2}{\alpha^2\beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix} \tag{4}$$

$$\mathbf{b} = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T \tag{5}$$

$$\mathbf{h}_i^T \mathbf{B} \mathbf{h}_j = \mathbf{v}_{ij}^T \mathbf{b} \tag{6}$$

$$\begin{aligned} \mathbf{v}_{ij} = [& h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2} \\ & h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]^T \end{aligned} \tag{7}$$

$$\begin{bmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{bmatrix} \mathbf{b} = \mathbf{0} \tag{8}$$

$$\begin{aligned} v_0 &= (B_{12}B_{13} - B_{11}B_{23}) / \left(B_{11}B_{22} - B_{12}^2\right) \\ \lambda &= B_{33} - \left[B_{13}^2 + v_0 (B_{12}B_{13} - B_{11}B_{23})\right] / B_{11} \\ \alpha &= \sqrt{\lambda / B_{11}} \\ \beta &= \sqrt{\lambda B_{11} / (B_{11}B_{22} - B_{12}^2)} \\ \gamma &= -B_{12}\alpha^2\beta / \lambda \\ u_0 &= \gamma v_0 / \beta - B_{13}\alpha^2 / \lambda \end{aligned} \tag{9}$$

The camera intrinsic calibration matrix obtained is:

$$\begin{pmatrix} \mathbf{2.05759e+03} & \mathbf{-6.68390e-01} & \mathbf{7.64017+02} \\ \mathbf{0} & \mathbf{2.04457e+03} & \mathbf{1.36076+03} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}$$

C. Estimate Camera extrinsic

Once we compute the homography H and the intrinsic parameters we use them to compute the extrinsic parameters i.e. Rotation and translation vectors. These are computed using the following equations with h1,h2,h3 are columns of homography matrix for respective image:

$$\begin{aligned} \mathbf{r}_1 &= \lambda \mathbf{A}^{-1} \mathbf{h}_1 \\ \mathbf{r}_2 &= \lambda \mathbf{A}^{-1} \mathbf{h}_2 \\ \mathbf{r}_3 &= \mathbf{r}_1 \times \mathbf{r}_2 \\ \mathbf{t} &= \lambda \mathbf{A}^{-1} \mathbf{h}_3 \end{aligned}$$

We obtain a mean re-projection of **0.93339.** The next step after computing the extrinsic and intrinsic parameters is to do optimization. Initially, value of radial distortion is (0,0). To minimize the reprojection error, scipy.optimize is used.

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\left\|\mathbf{m}_{ij} - \hat{\mathbf{m}}\left(\mathbf{K}, \mathbf{R}_i, \mathbf{t}_i, \mathrm{M}_j\right)\right\|^2$$

After the optimization, the distortion coefficients come out to be**: k1 = 0.04909 k2 = -0.3353**

D. DISCUSSION AND CONCLUSION

We obtained a much more refined intrinsic camera matrix once we perform non-linear minimization. The optimization is based on minimizing the Euclidean distance between corner points in the image and the points obtained after reprojection of the world coordinates. After optimization there is still some error left because this optimization gives a local minimum. This is why a good initial guess is necessary.

We obtain the norm between what we got with our transformation and the corner finding algorithm, thus obtaining the reprojection error which is important for evaluating the accuracy of calibration matrices.

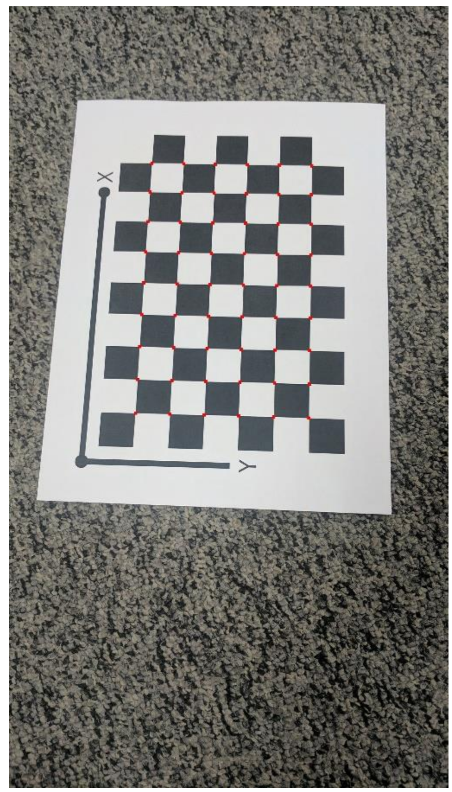

Figure 3: Rectified output of image 1

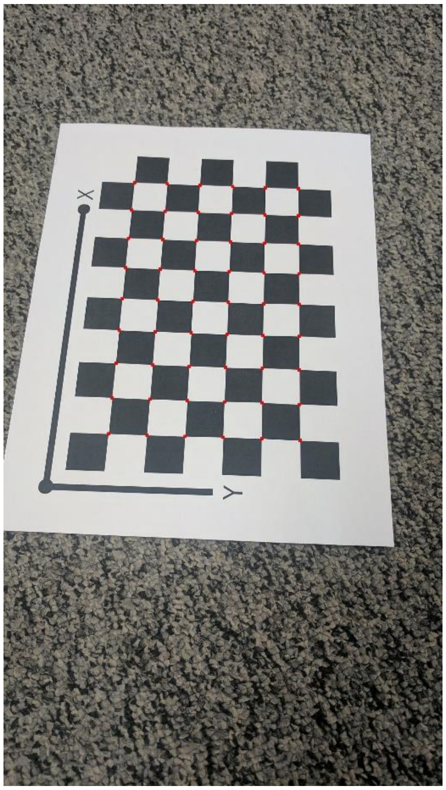

Figure 4: Rectified output of image 2

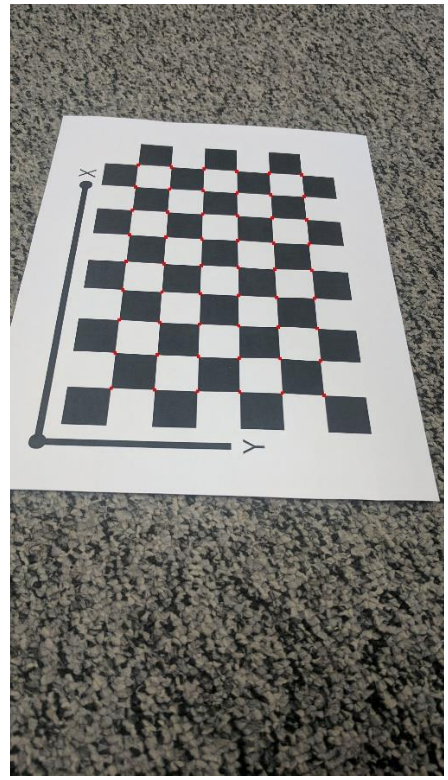

Figure 6: Rectified output of image 4

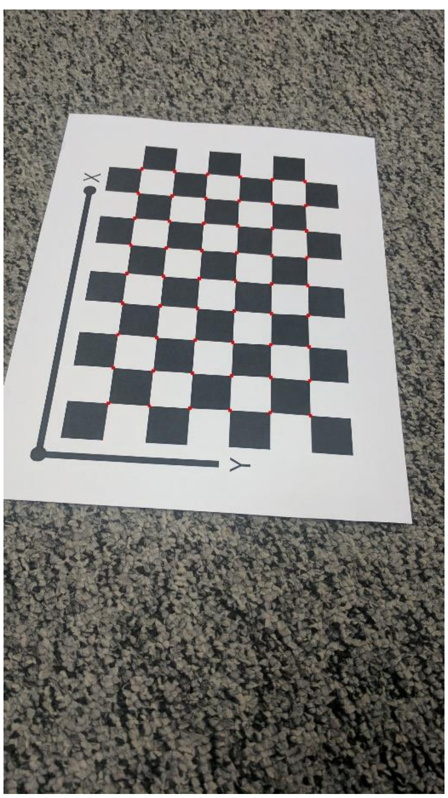

Figure 5: Rectified output of image 3

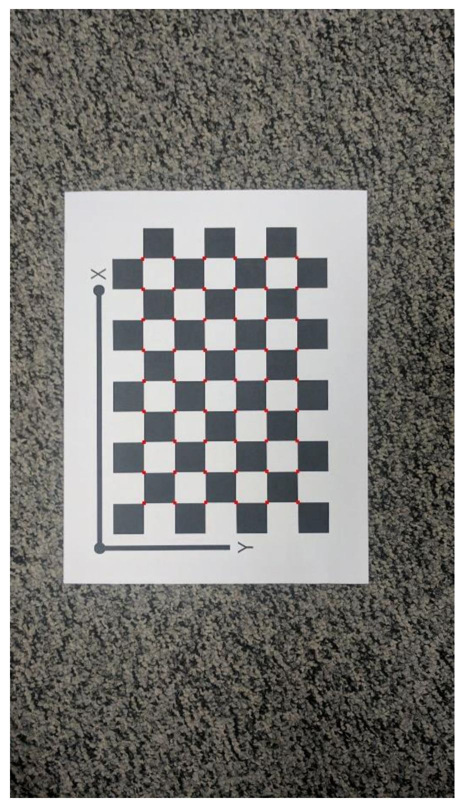

Figure 7: Rectified output of image 5

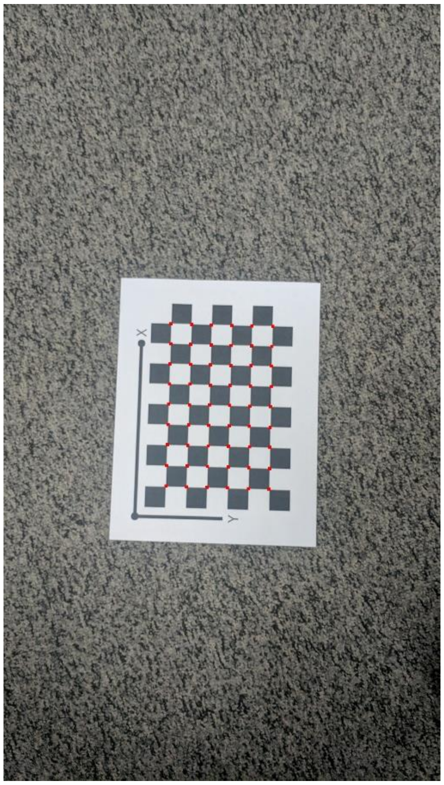

Figure 8: Rectified output of image 6

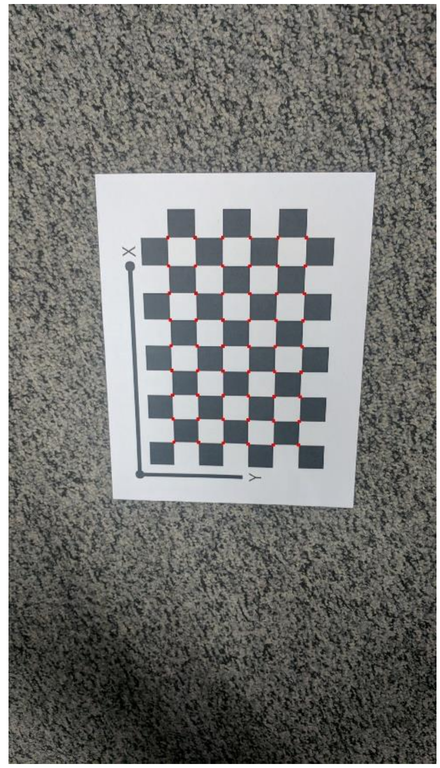

Figure 10: Rectified output of image 8

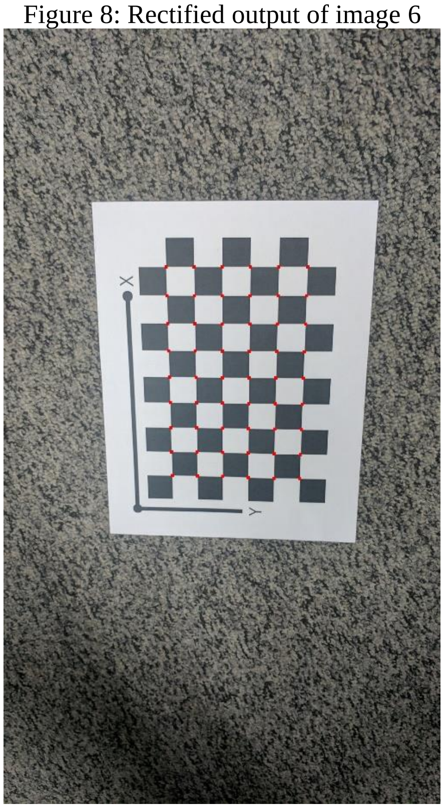

Figure 9: Rectified output of image 7

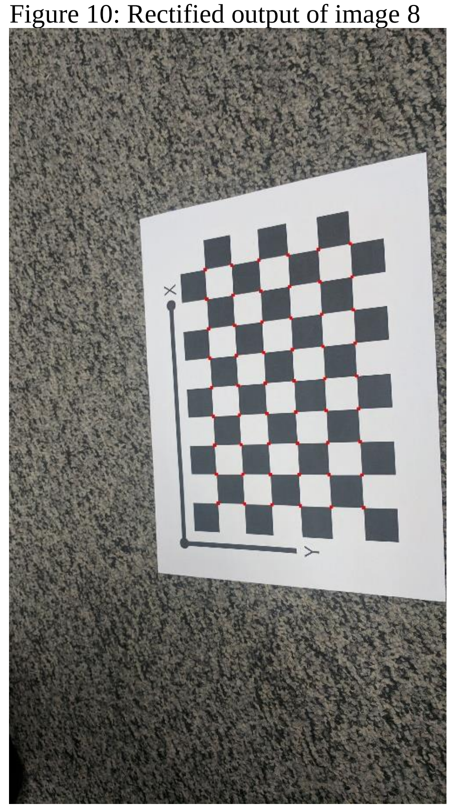

Figure 11: Rectified output of image 9

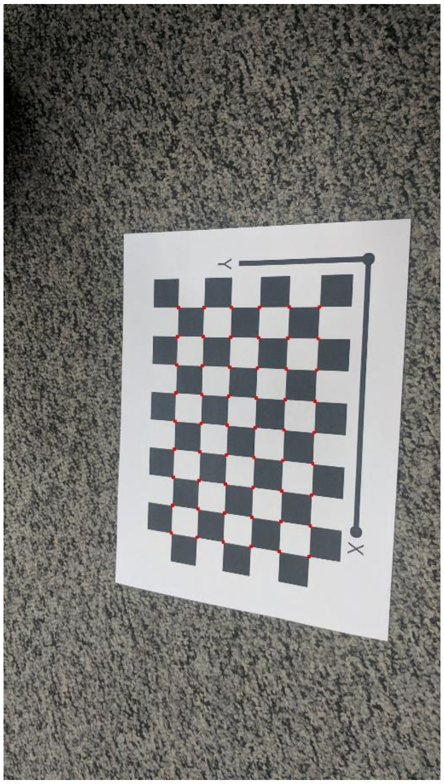

Figure 12: Rectified output of image 10

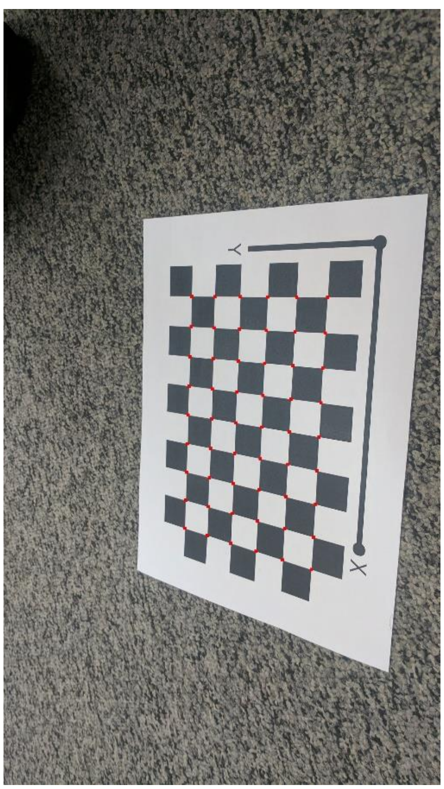

Figure 13: Rectified output of image 11

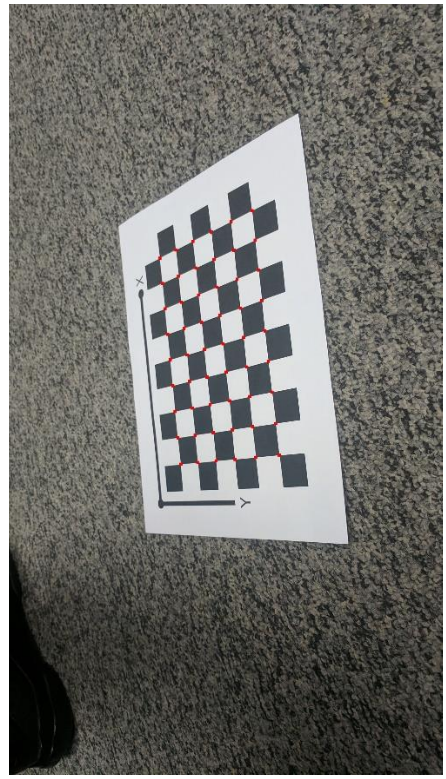

Figure 14: Rectified output of image 12

REFERENCES

[1] Project Description: https://cmsc733.github.io/2019/hw/hw1/
[2] Zhang, Zhengyou. ”A flexible new technique for camera calibration.” IEEE Transactions on pattern analysis and machine intelligence 22 (2000).
[3] https://kushalvyas.github.io/calib.html