RBE549 - HW1 AutoCalib

Harshal Bhat
Robotics Engineering Department
hbhat@wpi.edu
Using **1** Late day

I. Introduction

Camera calibration is an important step before taking any image data. In this report, the objective is to estimate camera parameters like Intrinsic parameters and extrinsic parameters, calculate distortion coefficients, and rectify the image mentioned in the method used in [1] for auto-camera calibration.

A. Camera Calibration Outline

- Print a pattern that has good feature points and attach it to a planar surface without any warp or distortion.
- Take images from different viewpoints by moving the image plane or camera viewpoint.
- Estimate the 5 intrinsic parameters and extrinsic parameters using a closed-form solution described in Section 3.1 of [1]
- Estimate the distortion coefficients by solving the least-squares described in Section 3.3 of [1]. Initially assume them to be 0.
- Refine the parameters by solving the nonlinear minimization problem defined in equation (14) of section 3.3

I first obtain the real-world picture points for a set of checkerboard photographs. Without the outer squares, I have 9 × 6 points placed at 21.5mm intervals. This generates a mesh grid of (x, y) world coordinates for each corner point, ranging from 0 to 5 rows and 0 to 8 columns, for a total of 54 points (n_p), designated as M. After computing the world points, I used OpenCV's built-in function, *cv2.findChessboardCorners*, to get the image coordinates of the checkerboard. All n_p point correspondences between m and M were ordered consistently across all n_i images in the given dataset.

B. Camera Intrinsic Parameters estimation

We know that the relation between world points and image correspondences is given by

$$m = K[Rt]M \tag{1}$$

where K is the camera matrix. In the paper[1], it is referred to A and given by

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

The Homography matrix indices are used to calculate **V** as highlighted in section 3.1 of [1]

$$H = A \begin{bmatrix} r1 & r2 & r3 & t \end{bmatrix} \tag{3}$$

$$h_1 A^{-T} A^{-1} h_2 = 0 \tag{4}$$

We estimate B matrix by solving the homogenous equation

$$Vb = 0 \tag{5}$$

where b = $\begin{bmatrix} B11, B12, B22, B13, B23, B33 \end{bmatrix}$ The intrinsic parameters are computed using the following equations

$$v_0 = \frac{B_{12}B_{13} - B_{11}B_{23}}{B_{11}B_{22} - B_{12}^2} \tag{6}$$

$$v_0 = \frac{B_{12}B_{13} - B_{11}B_{23}}{B_{11}B_{22} - B_{12}^2} \tag{7}$$

$$\lambda = B_{33} - \frac{B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})}{B_{11}} \tag{8}$$

$$\alpha = \sqrt{\frac{\lambda}{B_{11}}} \tag{9}$$

$$\beta = \sqrt{\frac{\lambda B_{11}}{B_{11}B_{22} - B_{12}^2}} \tag{10}$$

$$\gamma = -\frac{B_{12}\alpha^2\beta}{\lambda} \tag{11}$$

$$u_0 = \frac{\gamma v_0}{\beta} - \frac{B_{13}\alpha^2}{\lambda} \tag{12}$$

C. Computation of Camera Extrinsic Parameters

To establish the camera's extrinsic characteristics, which included rotation and translation, I followed the steps provided in [1], notably section 3.1. Using $\mathbf{K}_{\text{init}}^{-1}$ and $\mathbf{H}_i$, the rotation vectors $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$ and translation vector $\mathbf{t}$ were computed for each picture in the dataset of n_i images. This approach provided an initial estimate of intrinsic camera parameters $\mathbf{K}_{\text{init}}$ and extrinsic parameters $\mathbf{RT}_{\text{init}_i}$ for all n_i images. Using these parameters, I can reproject the world coordinates $\mathbf{M}_i$ onto the image plane, yielding'reprojected' image coordinates $\hat{\mathbf{m}}_i$.

D. Projecting world coordinates and Distortion Estimatinon

The coordinates $\hat{m}$ represent image coordinates reprojected using our camera calibration parameters. To assess the accuracy of our intrinsic and extrinsic parameters, we can compute the reprojection error ρ_{error} between m_i and $\hat{m}_i$. The reprojection error is computed as follows:

$$\rho_{\text{error}} = \frac{1}{n_i}\frac{1}{n_p}\sum_{i=1}^{n_i}\sum_{j=1}^{np}\|m_{ij} - \hat{m}_{ij}\|$$

Algorithm 1 Project Coordinates

function PROJECTCOORDINATES(M_i, RT, K, kC)
Extract camera parameters and distortion coefficients from K and kC
Extract rotation matrix R and translation vector t from RT
Initialize an empty list to store projected coordinates
  for each 3D point M in M_i **do**
Append a homogeneous coordinate $[M, 1]$ to M
Project M onto the image plane using RT
Normalize the projected coordinates
Extract normalized x and y coordinates
Apply radial distance correction based on distortion coefficients
Compute 2D coordinates u and v
Append $[u, v]$ to the list
  end for
return List of projected coordinates
end function=0

II. RESULTS

Distortion Coordinates (kC) before optimization: $(0, 0)$

Projection error before optimization: 0.7201492174873587

Intrinsic matrix (K) before optimization:

$$\begin{bmatrix} 2056.10659 & -1.01709784 & 761.655247 \\ 0 & 2040.50404 & 1351.30849 \\ 0 & 0 & 1 \end{bmatrix}$$

Distortion Coordinates after optimization: $(0.01272755043233208, -0.09773893429297463)$

Projection error after optimization: 0.7061400154376088 (without using `cv2` function)

Intrinsic matrix (K) after optimization:

$$\begin{bmatrix} 2056.10227 & -1.01709784 & 761.713151 \\ 0 & 2040.49566 & 1351.31937 \\ 0 & 0 & 1 \end{bmatrix}$$

Figures 1-12 show the corrected checkerboard pictures with reprojected points after calibration and optimization of extrinsic parameters RTi. Images before and after calibration are presented below.

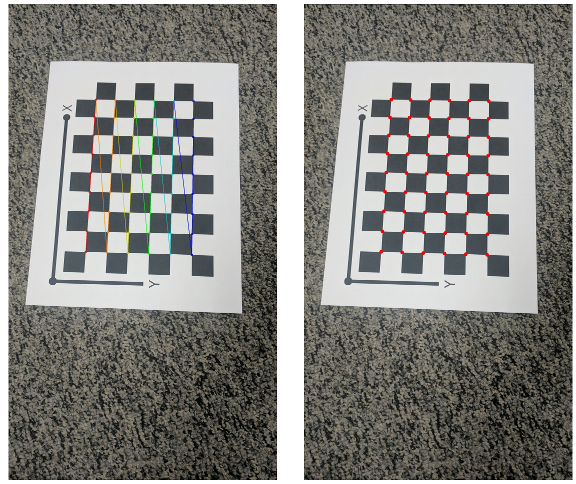

Fig. 1. Corner detected Image and Reprojected Corners Image 1

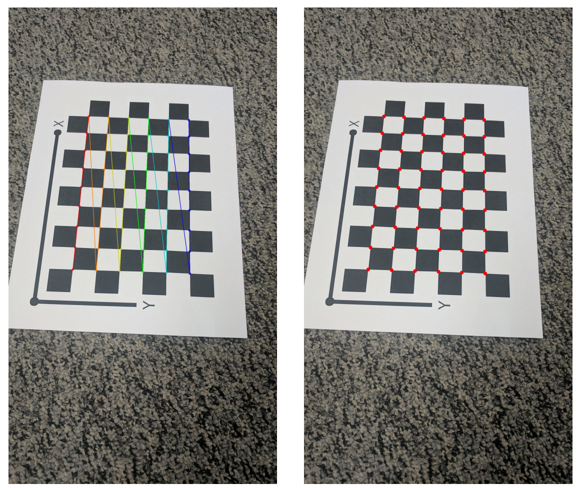

Fig. 2. Corner detected Image and Reprojected Corners Image 2

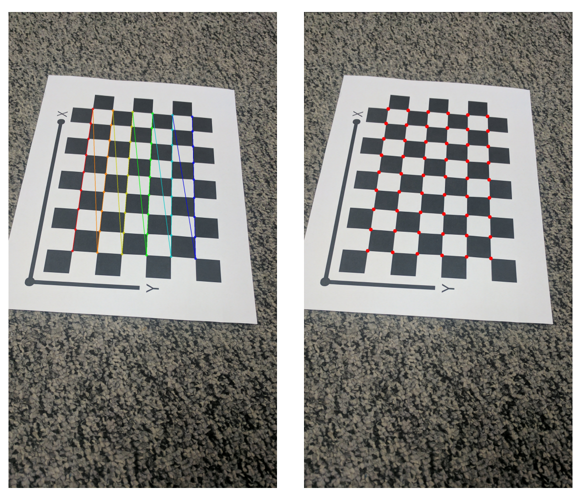

Fig. 3. Corner detected Image and Reprojected Corners Image 3

REFERENCES

[1] Z. Zhang, "A flexible new technique for camera calibration," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330-1334, Nov. 2000, doi: 10.1109/34.888718.

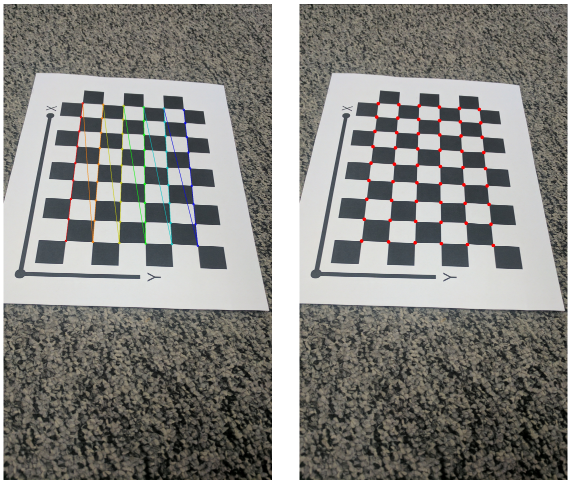

Fig. 4. Corner detected Image and Reprojected Corners Image 4

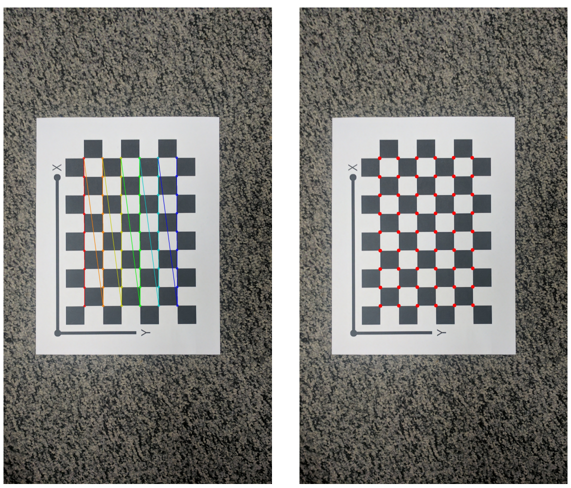

Fig. 5. Corner detected Image and Reprojected Corners Image 5

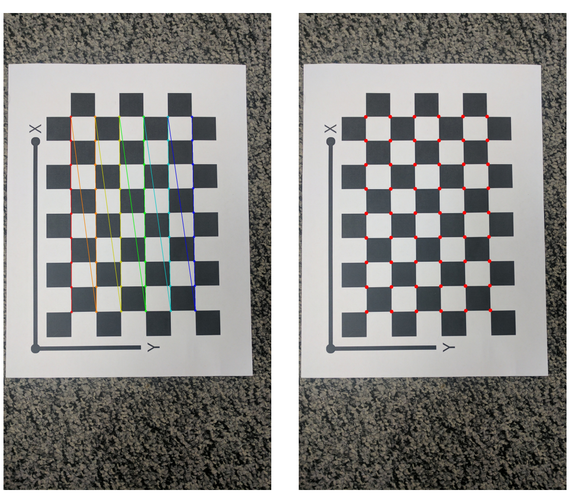

Fig. 6. Corner detected Image and Reprojected Corners Image 6

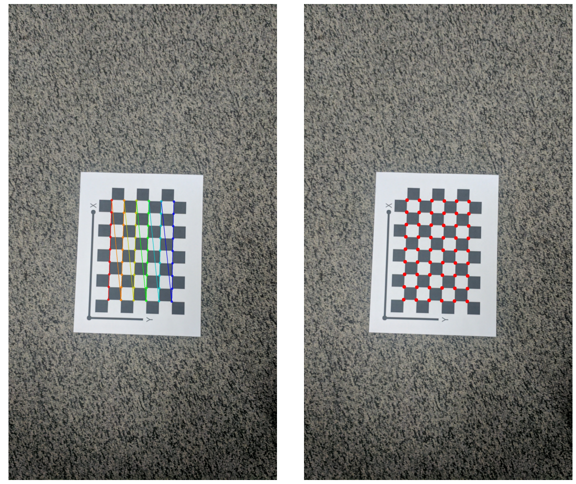

Fig. 7. Corner detected Image and Reprojected Corners Image 7

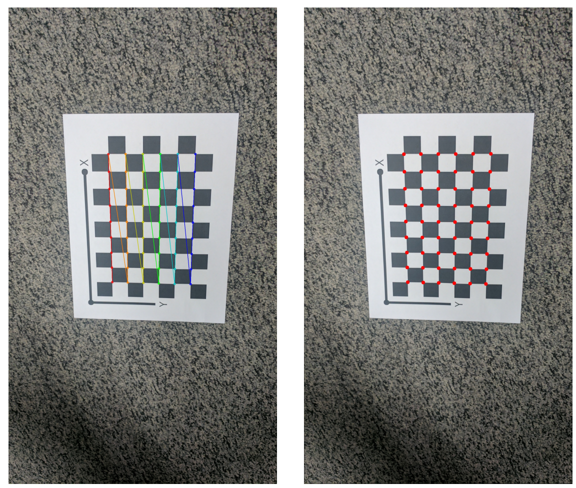

Fig. 8. Corner detected Image and Reprojected Corners Image 8

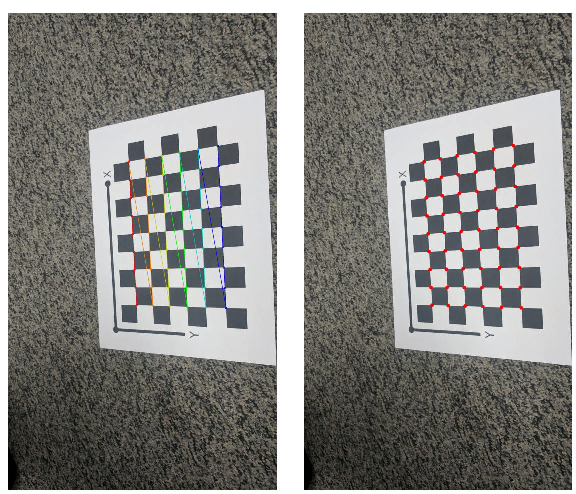

Fig. 9. Corner detected Image and Reprojected Corners Image 9

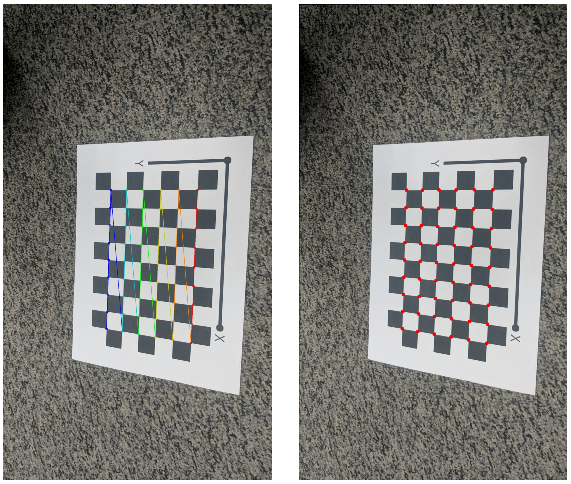

Fig. 10. Corner detected Image and Reprojected Corners Image 10

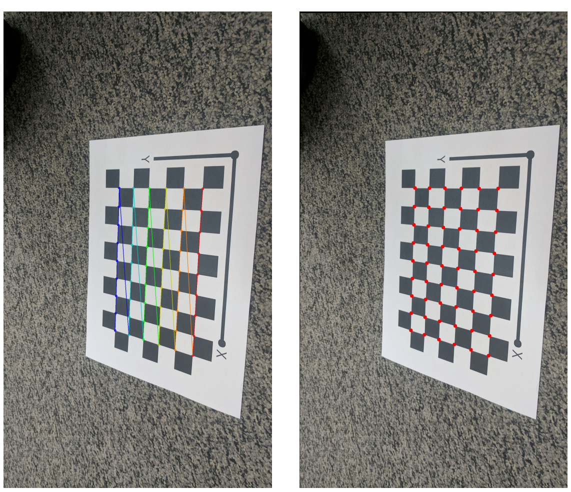

Fig. 11. Corner detected Image and Reprojected Corners Image 11

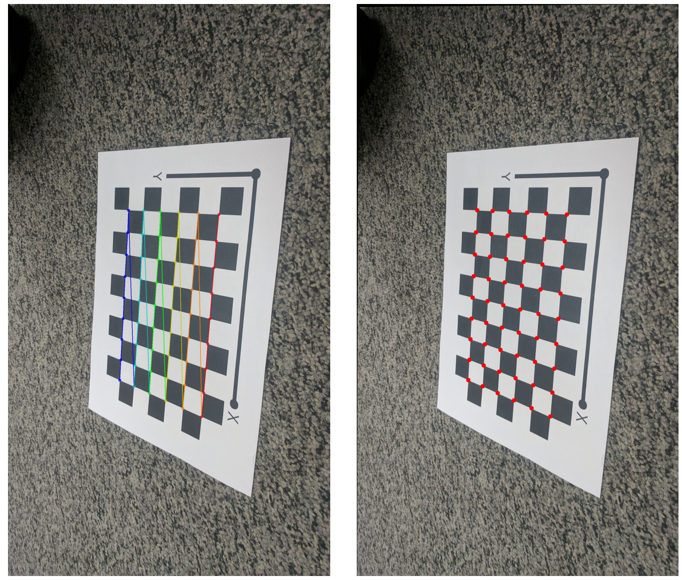

Fig. 12. Corner detected Image and Reprojected Corners Image 13

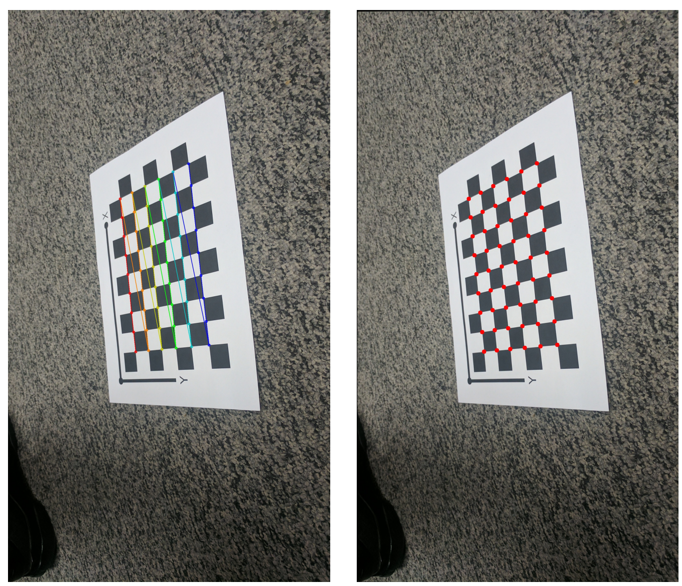

Fig. 13. Corner detected Image and Reprojected Corners Image 14