CS549: Homework 1 - AutoCalib

Yijia Wu
School of Robotics Engineering
Worcester Polytechnic Institute
Email: ywu21@wpi.edu

***Abstract*—This report introduces the implementation process and results of Zhang's camera calibration method[1].**

I. Introduction

Camera calibration is a necessary process for almost any computer vision task with images that are taken by a camera. Though we can know many parameters from the camera datasheet, such as focal length, image resolution, and pixel size, in practice, because of the mechanical and electrical errors, there is always offset which might influence the overall performance. Zhang's camera calibration method[1] is still the most well-known and popular camera calibration method.

II. Methodology

A. Image Points detection

A checkerboard can be used for calibration as shown in Fig. 1. The number of squares in rows and columns and the size of the squares should be known. The first step of the calibration is to take multiple images with the checkerboard fully inside the field of view. The next step is to detect the corners of the squares in the checkerboard. Only corners with four neighboring squares are considered to improve robustness.

The corner detection is implemented using *cv2.findChessboardCorners* function with the images and pattern parameter (corner column and row) as the input.

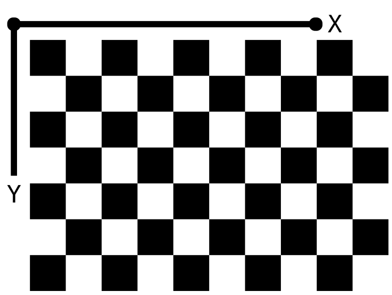

Fig. 1: Checkerboard

B. Initial Parameter Estimation

After getting the pattern corners in the images, the next step is to estimate the initial camera intrinsic matrix K, extrinsic matrices R and t, and distortion parameters $k = [k1, k2]$. They follow the relationship in the following equation. x denotes the detected corner position in the image and X denotes the corresponding 3D position in the world frame. The origin of the world frame is set to be the first corner of the checkerboard.

$$sx = K \begin{bmatrix} R & t \end{bmatrix} X \tag{1}$$

1) Estimate camera intrinsic matrix K: There are three steps that are necessary to estimate the intrinsic matrix.

- Estimate the homography matrix between the detected image corner and corresponding world points. Details can be found in Appendix A of [1].
- Estimate a matrix $B = (K^{-1})^T K^{-1}$. Details can be found in Section 3.1 of [1].
- Compute parameters $f_x, f_y, c_x, c_y, \lambda$ from matrix B to form intrinsic matrix K. Details can be found in Appendix B of [1].

$$K = \begin{bmatrix} f_x & \lambda & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

2) Estimate camera extrinsic R and t: After the initial intrinsic matrix K has been estimated, the extrinsic matrix R and t can be computed according to the equations in section 3.1 of [1].

C. Initialize distortion parameter

For a small distortion camera, we can assume that $k = [0, 0]$.

D. Non-linear Geometric Error Minimization

The last step is feeding the initial estimations into a non-linear minimization optimizer to refine camera parameter estimations. The *scipy.optimize.least_squares* function is used for optimization. Inside the function to be optimized, world points X are reprojected to the image plane using the intrinsic and extrinsic matrices and then distorted using the distortion parameters.

$$argmin_{f_x,f_y,c_x,c_y,k_1,k_2} \sum_{i=1}^{m} \sum_{j=1}^{n} = \left\| x_{ij} - \hat{x}(K, R_i, t_i, X_j) \right\|^2 \tag{3}$$

III. Results

To verify the auto-calibration implementation, 13 images that were taken by a Google Pixel XL phone with focus locked with a (9 x 6) corners checkerboard were used for calibration.

After the corner detection and the initial parameter estimation process, we can get the initial intrinsic matrix K_0.

Because of the small distortion, k_1 and k_2 are initialized as 0. With the method that is described in [1], the skew parameter can also be estimated as -0.46829, but we ignore it in K_0.

$$K_0 = \begin{bmatrix} 2053.04115 & 0 & 762.79854 \\ 0 & 2037.10197 & 1351.64446 \\ 0 & 0 & 1 \end{bmatrix}$$

With the non-linear optimization, we finally get the least square result for initial matrix K and distortion parameters k_1 and k_2. The reprojection error is 0.69426 pixel.

$$K = \begin{bmatrix} 2048.73500 & 0 & 762.830054 \\ 0 & 2032.33005 & 1351.66695 \\ 0 & 0 & 1 \end{bmatrix}$$

$$k_1 = 1.69346918 \times 10^{-8}$$

$$k_2 = -2.21194019 \times 10^{-14}$$

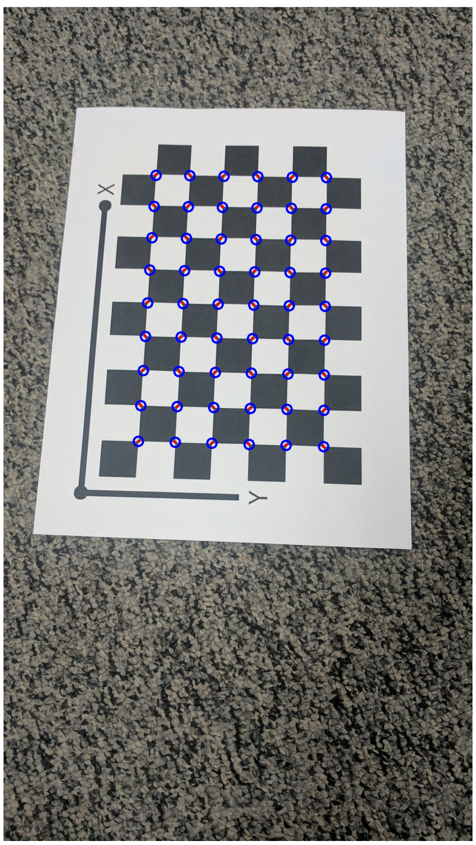

Fig. 2: Detected corners (red dots) and reprojected corners (center of blue circles) on undistorted image 1

In order to intuitively show the calibration performance, the undistorted images are shown in Fig. 2-4 with the initially detected corners (red dots) and the reprojected corners (center of blue circles) from the world points using the estimated camera intrinsic matrix.

IV. Discussion and Conclusion

This simplified implementation can already realize a less than 1-pixel reprojection error. It can be further improved. From the image collection size, more images with different scales, positions, and orientations can make the calibration result more robust. There are many other usable patterns, for example, a circular pattern board would be more robust to blur. From the parameter estimation side, there are many tricks in the paper [1] that are ignored in this implementation, such as initialization of distortion parameters, normalization of matrix for homography estimation, and refining extrinsic parameters during non-linear optimization.

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.

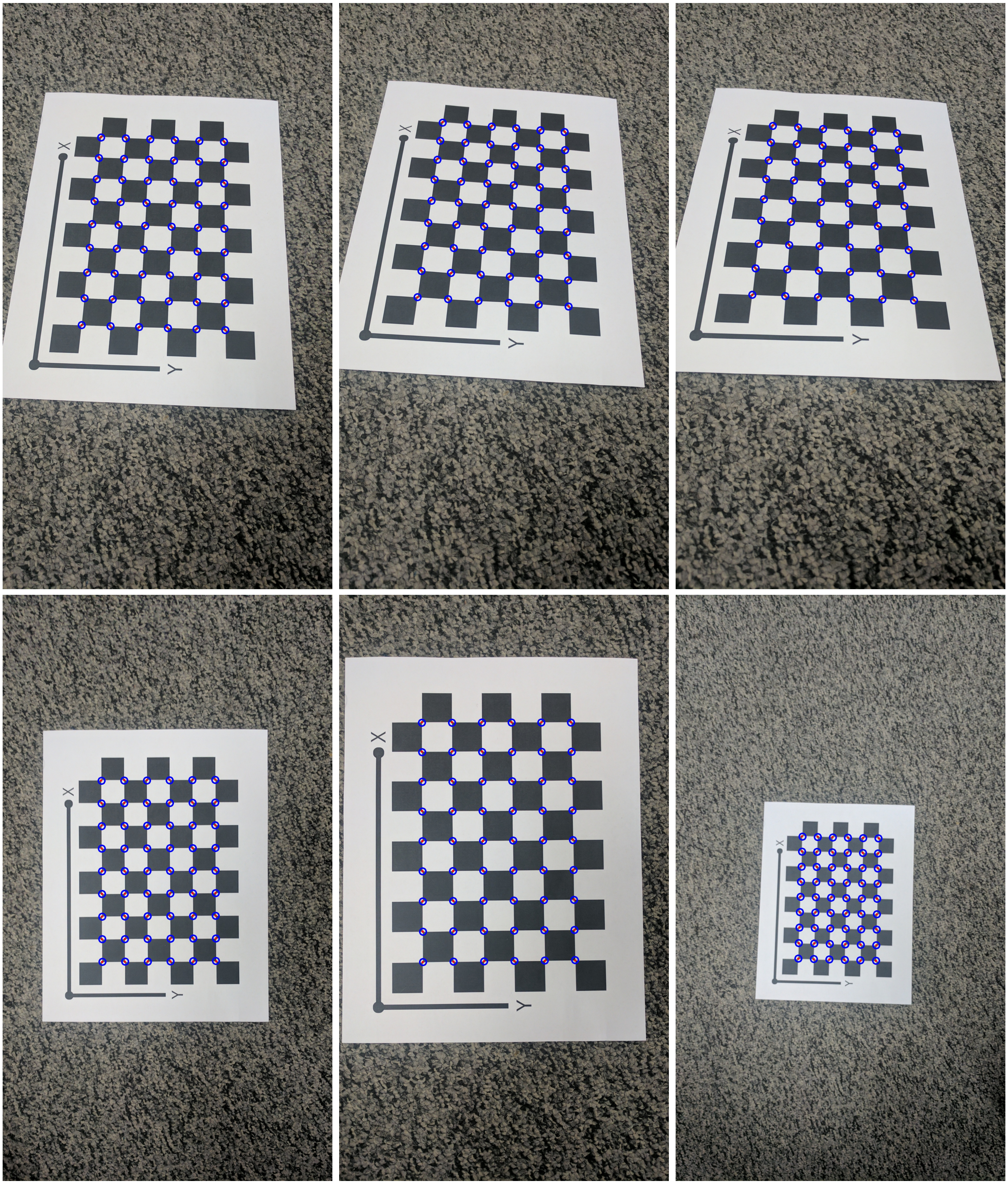

Fig. 3: Detected corners (red dots) and reprojected corners (center of blue circles) on undistorted image 2-7

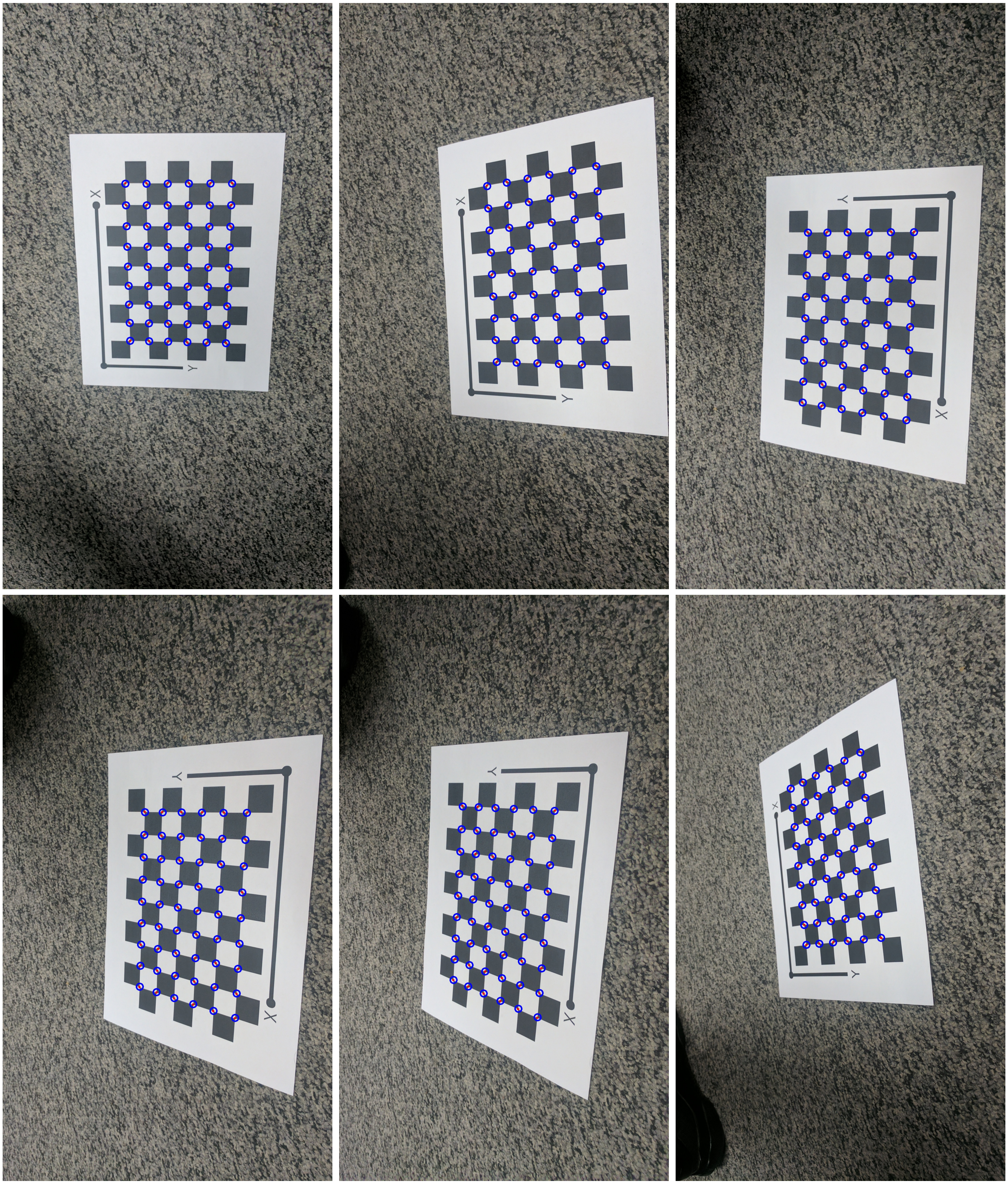

Fig. 4: Detected corners (red dots) and reprojected corners (center of blue circles) on undistorted image 8-13