RBE 549 - HW 1: Auto Calib

Simran Chauhan
MS Robotics Engineering
Worcester Polytechnic Institute
schauhan@wpi.edu

***Abstract*—This report implements the robust camera calibration technique proposed by Z. Zhang in ”A Flexible New Technique for Camera Calibration” [1]. The method, which requires at least two images at different orientations, is tested on a dataset of 13 images captured using a Google Pixel XL phone. The camera parameters are estimated in two stages: first, computing homographies for each image to obtain initial camera intrinsics through a closed-form solution, then calculating extrinsic parameters for each image. Finally, these parameters, along with radial distortion coefficients, are refined using the Levenberg-Marquardt optimization algorithm to minimize reprojection error.**

I. Introduction

A. Pinhole Camera Projection Model

The camera calibration process requires understanding of the projection of 3D world points onto a 2D image plane. This projection establishes a mapping between a 2D image point $\mathbf{x} = (x, y)^\top$, with homogeneous representation $\bar{\mathbf{x}} = (x, y, 1)^\top$, and a 3D world point $\mathbf{X} = (X, Y, Z)^\top$, with homogeneous form $\bar{\mathbf{X}} = (X, Y, Z, 1)^\top$.

Using the pinhole camera model, this projection is mathematically expressed as:

$$s\bar{\mathbf{x}} = \mathbf{K}[\mathbf{R}\ \ \mathbf{t}]\bar{\mathbf{X}} \tag{1}$$

Here, s represents an arbitrary scale factor, $\mathbf{K}$ denotes the camera's intrinsic parameter matrix, while $\mathbf{R}$ and $\mathbf{t}$ are the rotation and translation matrices (known as extrinsic parameters) that transform the world frame to the camera frame.

For our application, we can simplify this model by assuming the model plane lies on $Z = 0$ in world coordinates. This reduces Equation (1) to:

$$s\bar{\mathbf{x}} = \mathbf{K}[\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{r}_3\ \mathbf{t}] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \mathbf{K}[\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{t}] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{2}$$

where $\mathbf{r}_i$ represents the ith column of $\mathbf{R}$. In this context, $\mathbf{X}$ now refers to the simplified 3D world point with $Z = 0$.

The relationship between the model plane and its image is defined by a homography $\mathbf{H}$:

$$\mathbf{H} = \mathbf{K}[\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{t}] \tag{3}$$

This homography can be computed directly from an image of a known planar pattern, using the model points of the planar pattern and the image points detected from the image.

B. Initial Homography Computation

The homography computation begins with initializing a set of world points representing a 9×6 checkerboard grid with 21.7mm square size. For each input image, checkerboard corners are detected using OpenCV's `findChessboardCorners` function, followed by `cornerSubPix` refinement for sub-pixel accuracy. The homography matrix $\mathbf{H}$ is then computed using `cv2.findHomography`, which establishes the geometric mapping between the planar world points and their corresponding detected image points.

II. CAMERA CALIBRATION METHOD

This section presents a method for estimating camera intrinsic parameters and distortion coefficients by utilizing a dataset consisting of N model points and a collection of M images, each containing corresponding N image points.

A. Camera Intrinsics

The camera intrinsic parameters are represented in matrix $\mathbf{K}$, which contains five independent variables: the principal point coordinates (u_0, v_0), the scaling factors α and β along the u and v axes, and a skew parameter γ:

$$\mathbf{K} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

By leveraging the orthonormality property of rotation vectors $\mathbf{r}_1$ and $\mathbf{r}_2$, we obtain two constraints:

$$\mathbf{h}_1^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_2 = 0 \tag{5}$$

$$\mathbf{h}_1^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_1 = \mathbf{h}_2^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathbf{h}_2 \tag{6}$$

where $\mathbf{h}_i$ represents the ith column of the homography matrix $\mathbf{H}$.

To simplify our analysis, we introduce matrix $\mathbf{B} = \mathbf{K}^{-\top}\mathbf{K}^{-1}$:

$$\mathbf{B} \equiv \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} \tag{7}$$

Given the symmetry of $\mathbf{B}$, we can represent it using a six-dimensional vector:

$$\mathbf{b} = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^\top \tag{8}$$

This allows us to express the constraints as:

$$\mathbf{h}_i^\top \mathbf{B} \mathbf{h}_j = \mathbf{v}_{ij}^\top \mathbf{b} \tag{9}$$

where $\mathbf{v}_{ij} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]$.

These constraints can be consolidated into a single equation:

$$\left[\mathbf{v}12^\top; (\mathbf{v}11 - \mathbf{v}_{22})^\top\right] \mathbf{b} = \mathbf{0} \tag{10}$$

By combining equations from all M images, we form a linear system:

$$\mathbf{V}\mathbf{b} = 0 \tag{11}$$

where $\mathbf{V}$ is a $2M \times 6$ matrix. By applying singular value decomposition, we extract the right singular vector associated with the smallest singular value to determine $\mathbf{b}$. Using this solution, we then derive the intrinsic parameters:

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

B. Camera Extrinsics

With the intrinsic parameters previously estimated, we can now calculate the extrinsic parameters (rotation and translation). Using the homography relationship from Equation (3), we have:

$$\mathbf{r}_1 = \lambda \mathbf{K}^{-1}\mathbf{h}_1 \tag{12}$$

$$\mathbf{r}_2 = \lambda \mathbf{K}^{-1}\mathbf{h}_2 \tag{13}$$

$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2 \tag{14}$$

$$\mathbf{t} = \lambda \mathbf{K}^{-1}\mathbf{h}_3 \tag{15}$$

where λ is a scaling factor determined by the norm of either transformed column:

$$\lambda = \frac{1}{\|\mathbf{K}^{-1}\mathbf{h}_1\|} = \frac{1}{\|\mathbf{K}^{-1}\mathbf{h}_2\|} \tag{16}$$

In practice, due to measurement noise, the computed rotation matrix $\mathbf{R}$ may not perfectly satisfy rotation matrix properties. To address this, we use singular value decomposition (SVD) to find the closest valid rotation matrix. If SVD decomposes $\mathbf{R}$ into matrices $\mathbf{U}$, $\mathbf{S}$, and $\mathbf{V}$, then the best rotation matrix is obtained as:

$$\mathbf{R}_{\text{best}} = \mathbf{U}\mathbf{V}^\top \tag{17}$$

C. Radial Distortion

The previous analysis assumes an ideal pinhole camera model without distortion. However, real cameras typically exhibit significant distortion, particularly radial distortion. For practical purposes, we consider only the first two terms of the distortion model.

Let $\mathbf{x} = (x, y)$ represent the ideal (undistorted) normalized image coordinates, and $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y})$ denote the distorted image coordinates. The ideal points are obtained through pinhole model projection. The distortion model is given by:

$$r^2 = x^2 + y^2 \tag{18}$$

$$D = k_1 r^2 + k_2 r^4 \tag{19}$$

$$\tilde{\mathbf{x}} = \mathbf{x}(1 + D) \tag{20}$$

where k_1 and k_2 are the radial distortion coefficients. This model assumes the center of radial distortion coincides with the principal point (u_0, v_0).

To simplify distortion calculations, we apply these equations to the model points in normalized coordinate space before mapping them to pixel coordinates. The complete projection from model points $\mathbf{X}_j$ to pixel coordinates $\mathbf{u}_j$ proceeds as follows.

Let $\mathbf{R} = [\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}]$. The normalized coordinates are computed as:

$$\mathbf{x}_j = \frac{1}{\bar{\mathbf{R}}_3^\top \mathbf{X}_j} \begin{bmatrix} \bar{\mathbf{R}}_1^\top \mathbf{X}_j \\ \bar{\mathbf{R}}_2^\top \mathbf{X}_j \end{bmatrix} \tag{21}$$

After applying distortion to obtain $\tilde{\mathbf{x}}_j$, the final pixel coordinates are given by:

$$\mathbf{u}_j = \begin{bmatrix} \bar{\mathbf{K}}_1 \tilde{\mathbf{x}}_j \\ \bar{\mathbf{K}}_2 \tilde{\mathbf{x}}_j \end{bmatrix} \tag{22}$$

where $\bar{\mathbf{K}}_i$ denotes the ith row of $\mathbf{K}$.

While closed-form methods exist for approximating k_1 and k_2, we initially assume $k_1 = k_2 = 0$. This assumption provides sufficient accuracy to enable convergence in the subsequent optimization process.

D. Optimization

To optimize the intrinsic and extrinsic parameters of our camera, we define the geometric error as follows:

$$\text{Error} = \sum_{i=1}^{M} \sum_{j=1}^{N} \|\mathbf{x}_{ij} - \mathcal{X}(\mathbf{K}, k_1, k_2, \mathbf{R}_i, \mathbf{t}_i, \mathbf{X}_j)\|^2 \tag{23}$$

where $\mathcal{X}$ represents the projection function that maps 3D points to 2D image points using Equations (21) and (22). The parameters involved are: - Camera intrinsic matrix $\mathbf{K}$, Distortion coefficients k_1 and k_2, Rotation matrix $\mathbf{R}_i$, and translation vector $\mathbf{t}_i$ for each image i, and Model points $\mathbf{X}_j$.

Here, M denotes the number of images and N the number of points in each image.

The camera calibration optimization process refines both intrinsic and extrinsic parameters by minimizing the total projection error in (23) using the Levenberg-Marquardt algorithm. This optimization is implemented by using the `optimize.least_squares` function from SciPy, with `method='lm'`.

III. Results

We evaluated our calibration method on a dataset of 13 distinct checkerboard images. This section details the implementation process and presents the resulting camera parameters and error metrics.

A. Input Data

The calibration process requires two key inputs: a set of model points and their corresponding image points. We used a 9×6 checkerboard pattern with 21.7mm square size. The corner points in each image were detected using OpenCV's `findChessboardCorners` function, as illustrated in Fig. 1. The complete set of detected corners for all images is provided in Appendix A.

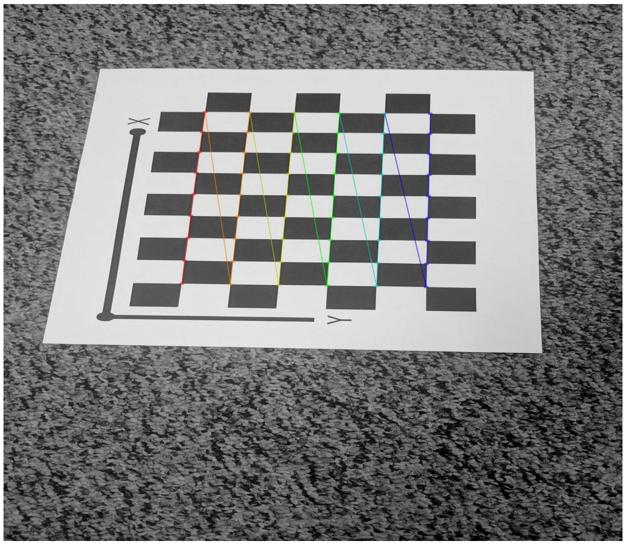

Fig. 1: CheckerboardCorners: Image 1

B. Calibration Results

The camera calibration yielded two sets of parameters: initial estimates and optimized values. The camera intrinsic matrices before and after optimization are:

$$\mathbf{K} = \begin{bmatrix} 2058.359 & -0.87 & 762.677 \\ 0 & 2041.726 & 1348.166 \\ 0 & 0 & 1 \end{bmatrix} \tag{24}$$

$$\mathbf{K}_{\text{opt}} = \begin{bmatrix} 2043.765 & 0 & 761.429 \\ 0 & 2035.951 & 1346.775 \\ 0 & 0 & 1 \end{bmatrix} \tag{25}$$

The radial distortion parameters changed from initial values of $\mathbf{k} = [0, 0]$ to optimized values of $\mathbf{k}_{\text{opt}} = [0.168, -0.721]$.

Table I presents the geometric reprojection errors (RMS) for each image, both before and after optimization. The root mean square error decreased significantly from 2.15 pixels to 0.511 pixels after optimization. Fig. 2 shows the reprojected corners on the rectified image. The complete set of reprojected corners on the calibration images can be found in Appendix B.

TABLE I: Geometric Reprojection Error in pixels (RMS)

Image	Unoptimized	Optimized
1	0.704	0.426
2	1.024	0.501
3	1.16	0.648
4	2.00	0.755
5	1.22	0.349
6	1.81	0.437
7	1.40	0.181
8	2.14	0.348
9	2.07	0.415
10	1.55	0.379
11	1.55	0.559
12	2.05	0.657
13	5.37	0.666
RMS error	2.15	0.511

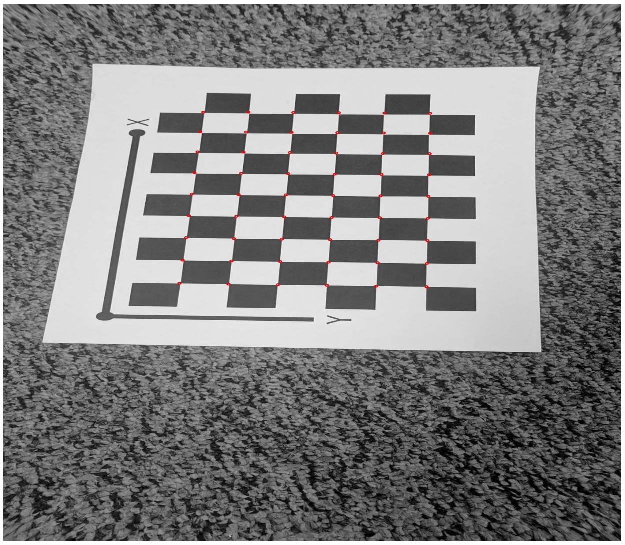

Fig. 2: Rectified Image with Re-projected Corners: Image 1

IV. Conclusion

The experimental results presented in this work demonstrate the robust performance and practical application of computer vision-based camera calibration algorithms for modern imaging devices.The quantitative improvement in reprojection error from 2.15 to 0.511 pixels through nonlinear optimization underscores the reliability of the implemented calibration pipeline.

References

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