AutoCalib

Computer Vision (RBE549) Homework 1

Hrishikesh Pawar
MS Robotics Engineering
Worcester Polytechnic Institute
Email: hpawar@wpi.edu

***Abstract*—The report outlines the implementation of Zhengyou Zhang's (1) seminal work on camera calibration technique.**

I. Introduction

Camera calibration aims to extract camera's parameters like focal length, distortion coefficients, and the principal point. The essence of camera calibration lies in constructing a forward imaging model. This model facilitates the transformation of points from a 3D world frame to a 2D image plane. This transformation process is executed through two primary transformations: **Coordinate Transformation** and **Perspective Projection**.

Coordinate Transformation is responsible for mapping points from the world frame to the camera frame. Following this, the **Perspective Projection**, converts the camera frame coordinates into pixel coordinates on the image plane. The parameters that dictate the specifics of these transformations are encoded within two sets of matrices: the Intrinsic and Extrinsic matrices. The Intrinsic matrix contains parameters that are inherent to the camera itself, such as its focal length and optical center, which are critical for the Perspective Projection. The Extrinsic matrix, details the camera's position and orientation in space, informing the Coordinate Transformation process.

This report delves into the methodology and steps involved in camera calibration, guided by Zhengyou Zhang's foundational work on the subject. The calibration process is outlined in four main stages:

1) Estimation of Initial Intrinsic Parameters.
2) Estimation of Initial Extrinsic Parameters.
3) Optimization for refinement of estimated parameters.
4) Results.

A. Estimation of Initial Intrinsic Parameters:

*1) **Data**:* As mentioned in Zhang's methodology, calibration procedure employs a calibration target (Fig. 1), specifically a checkerboard, to derive the camera's intrinsic parameters. This pattern was printed on an A4 sheet, with each square measuring 21.5mm. For the purpose of this calibration, thirteen images were captured using a Google Pixel XL smartphone, with the camera's focus fixed to ensure consistency. Within the calibration setup, it's important to note that the checkerboard's Y axis features an 7 squares, whereas the X axis comprises of 10. A common practice, though not mandatory, is to disregard the outermost squares, rows and columns at the checkerboard's periphery. For subsequent steps, the computation area is a central 6x9 grid.

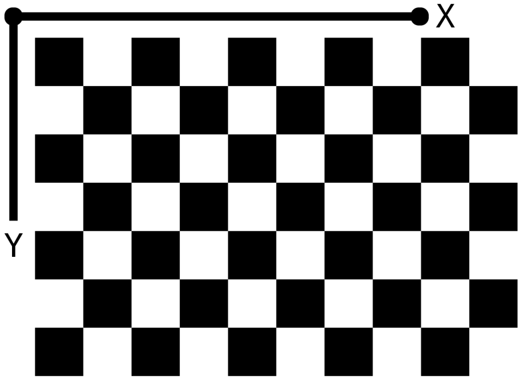

Fig. 1: Checkerboard Pattern

*2) **Finding the pixel and world coordinates**:* Next step involved identifying the checkerboard corners in both world coordinates and pixel coordinates. `cv2.findChessboardCorners()` function was used to detect the 54 (computation area consists of 6x9 grid) corners present in each image. Subsequently, corresponding real-world coordinates were also computed, taking into account the predefined size of the checkerboard squares (21.5mm). Fig 2 illustrated the output detected corners using `cv2.findChessboardCorners()`.

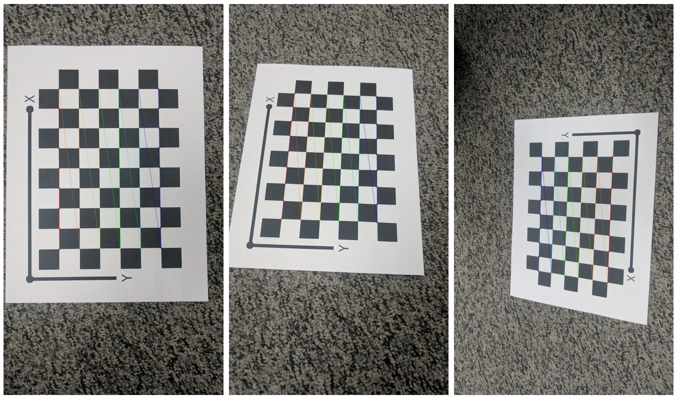

Fig. 2: Detected Corners

*3) **Computing Homography:*** Following are the basic notations that will be used in further formulations:

- A 2D point in image space: $m = [u, v]^T$.
- A 3D point in world space: $M = [X, Y, Z]^T$.

To facilitate the calculations, these vectors are represented as homogeneous coordinates resulting in the vectors $\tilde{\mathbf{m}} = [u, v, 1]^T$ for 2D points and $\tilde{\mathbf{M}} = [X, Y, Z, 1]^T$ for 3D points. The camera model adopted is the conventional pinhole model, which describes the relationship between a 3D point M and its image projection m. The relationship between a 3D point $\mathbf{M}$ and its image projection $\mathbf{m}$ is given by:

$$s\tilde{\mathbf{m}} = \mathbf{A}\left[\mathbf{R} \mid \mathbf{t}\right]\tilde{\mathbf{M}}, \tag{1}$$

where s is an arbitrary scale factor, $(\mathbf{R}, \mathbf{t})$, called the extrinsic parameters, is the rotation and translation which relates the world coordinate system to the camera coordinate system, and $\mathbf{A}$, called the camera intrinsic matrix, is given by

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

with (u_0, v_0) the coordinates of the principal point, α and β the scale factors in image u and v axes, and γ the parameter describing the skewness of the two image axes.

Without loss of generality, model plane is assumed to be on $Z = 0$ of the world coordinate system. The i^{th} column of the rotation matrix $\mathbf{R}$ is denoted by $\mathbf{r}_i$. Therefore from (1),

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{A}\left[\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{r}_3 \mid \mathbf{t}\right]\begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

$$= \mathbf{A}\left[\mathbf{r}_1\ \mathbf{r}_2 \mid \mathbf{t}\right]\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}.$$

Therefore, a model point $\mathbf{M}$ and its image $\mathbf{m}$ is related by a homography $\mathbf{H}$:

$$\tilde{\mathbf{m}} = \mathbf{H}\tilde{\mathbf{M}} \quad \text{with} \quad \mathbf{H} = \mathbf{A}\left[\mathbf{r}_1\ \mathbf{r}_2 \mid \mathbf{t}\right]. \tag{2}$$

$\mathbf{H}$ is a 3×3 matrix defined up to a scale factor.

Homography between the model plane and its image, is computed with the `find_homography()` function using the detected the real-world coordinates and the pixel coordinates. This homography matrix encodes both the intrinsic and extrinsic matrices, capturing the transformation from the 3D world to the 2D image plane.

*4) **Extracting the intrinsic matrix:*** After determining the H matrix, the goal is to estimate the A matrix which can be done in four steps:

1) Exploiting the constraints about $\mathbf{A}$, $\mathbf{r_1}$, $\mathbf{r_2}$
2) Defining matrix $\mathbf{B} = \mathbf{A^{-T}A^{-1}}$
3) Computing $\mathbf{B}$.
4) Decomposing $\mathbf{B}$.

Since A is on invertible matrix eq. (2) can be written as:

$$\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} = \mathbf{A}^{-1}\begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} \tag{3}$$

Since $\mathbf{r_1}$, $\mathbf{r_2}$ and $\mathbf{r_3}$ form an orthonormal basis,

$$\mathbf{r}_1^T\mathbf{r}_2 = 0 \quad \|\mathbf{r}_1\| = \|\mathbf{r}_2\| = 1 \tag{4}$$

Therefore, from eq. (3) and eq. (4)

$$\mathbf{h}_1^T\mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{h}_2 = 0 \tag{5}$$

$$\mathbf{h}_1^T\mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{h}_1 - \mathbf{h}_2^T\mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{h}_2 = 0 \tag{6}$$

Defining symmetric and positive matrix $\mathbf{B}$

$$\mathbf{B} := \mathbf{A}^{-T}\mathbf{A}^{-1}$$

$$\mathbf{B} = \mathbf{A}^{-T}\mathbf{A}^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} \tag{7}$$

From B, the intrinsic matrix can be extracted using Cholesky Decomposition. By definition B can be written as a 6D vector of unknowns.

$$\mathbf{b} = \begin{bmatrix} B_{11} & B_{12} & B_{22} & B_{13} & B_{23} & B_{33} \end{bmatrix}^T$$

Now constructing a system of linear equations $\mathbf{Vb} = 0$ and exploiting the constraints in eq. (4)

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

This is for single image. For n images

$$\begin{bmatrix} \mathbf{v}_{12}^T \\ \mathbf{v}_{11}^T - \mathbf{v}_{22}^T \\ \dots \\ \mathbf{v}_{12}^T \\ \mathbf{v}_{11}^T - \mathbf{v}_{22}^T \end{bmatrix}\mathbf{b} = 0 \tag{9}$$

where $\mathbf{V}$ is a $2n \times 6$ matrix. Now finding the solution that minimises the squares error.

$$\mathbf{b}^* = \underset{\mathbf{b}}{\text{argmin}}\ \|\mathbf{Vb}\| \quad \text{with} \quad \|\mathbf{b}\| = 1 \tag{10}$$

From eq. (10) estimating $\mathbf{B}$ upto an arbitrary scale λ the intrinsic parameters can be computed as:

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

B. Estimation of Initial Extrinsic Parameters:

Now as A is known, extrinsic parameters for each image can be computed from eq. (2) with $\lambda = \frac{1}{\|A^{-1}h_1\|} = \frac{1}{\|A^{-1}h_2\|}$

$$\begin{aligned} r_1 &= \lambda A^{-1} h_1 \\ r_2 &= \lambda A^{-1} h_2 \\ t &= \lambda A^{-1} h_3 \end{aligned}$$

C. Optimization for refinement of estimated parameters:

With the assumption that camera has minimal distortion, the initial estimate of distortion can be assumed as $\mathbf{k} = \begin{bmatrix} 0, & 0 \end{bmatrix}^T$ Now that the initial estimates of A, R, t and k are known, and the objective function to be minimised as the distance between ground truth corner coordinates and the reprojected coordinates, the optimization problem can be defined as,

$$\underset{\alpha,\beta,u_0,v_0,k_1,k_2}{\operatorname{argmin}} \sum_{i=1}^{N} \sum_{j=1}^{M} \left\| \mathbf{x}_{i,j} - \hat{\mathbf{x}}_{i,j} \left(\mathbf{A}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{X}_j, \mathbf{k} \right) \right\| \quad (11)$$

D. Results:

The intrinsic matrix before optimization:

$$A = \begin{bmatrix} 2053.04115 & -0.46828715 & 762.798539 \\ 0.00000000 & 2037.10197 & 1351.64446 \\ 0.00000000 & 0.00000000 & 1.00000000 \end{bmatrix}$$

After optimization:

$$A_{\text{opt}} = \begin{bmatrix} 2047.90769 & -0.47797346 & 762.789954 \\ 0.00000000 & 2031.46537 & 1351.66019 \\ 0.00000000 & 0.00000000 & 1.00000000 \end{bmatrix}$$

The optimized distortion coefficients:

$$k_{\text{opt}} = \begin{bmatrix} 0.082086245 & -0.45049271 \end{bmatrix}$$

The mean value of the re-projection error after optimization is $\mathbf{0.6883}$

The following table charts per image projection error before and after optimization.

Image	Before	After
1	0.6915	0.5494
2	0.7511	0.6811
3	0.8717	0.8173
4	0.9887	0.9447
5	0.5966	0.4782
6	0.7455	0.5863
7	0.8452	0.8361
8	0.5215	0.4948
9	0.6857	0.5919
10	0.6484	0.5854
11	0.8570	0.7530
12	0.9813	0.8940
13	0.75482	0.7348

TABLE I: Per-image projection error

REFERENCES

[1] https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr98-71.pdf

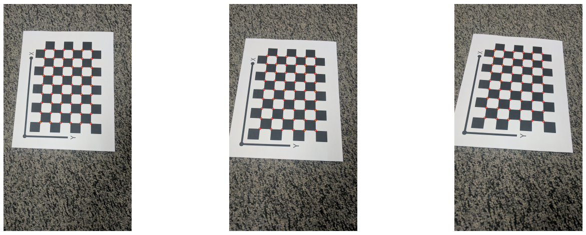

Fig. 3: Reprojected corners for images 1 to 3

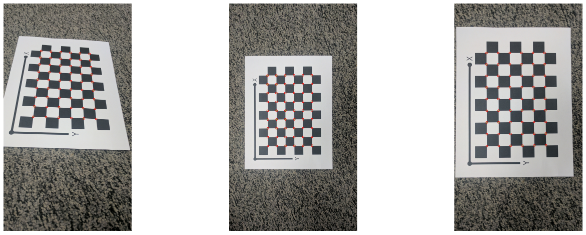

Fig. 4: Reprojected corners for images 4 to 6

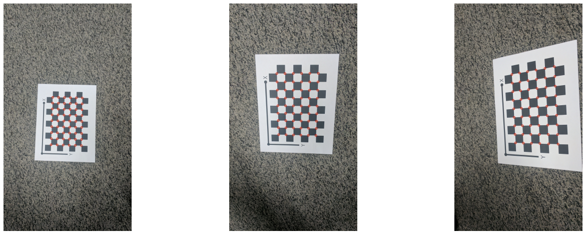

Fig. 5: Reprojected corners for images 7 to 9

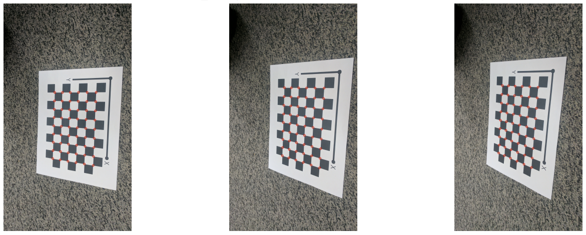

Fig. 6: Reprojected corners for images 10 to 12

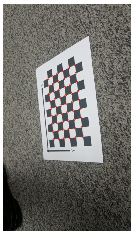

Fig. 7: Reprojected corners for image 13