RBE 549 HW1 : AutoCalib

Sai Hitesh Viswasam
Department of Robotics Engineering
Worcester Polytechnic Institute
Email: sviswasam@wpi.edu

***Abstract*—This report presents the implementation of the algorithmic technique of Camera Calibration proposed in "A Flexible New Technique for Camera Calibration" by Zhang from using multiple view images of a checkerboard. The approach uses multiple images of a flat pattern taken from different angles. First, a rough estimate of Homographies is found for each image and then improved. These transformations help calculate the camera's Intrinsic and Extrinsic parameters. Using these as initial estimates, finally the internal settings and distortion effects are optimised using the Levenberg-Marquardt algorithm.**

I. Introduction

Camera calibration is essential in 3D computer vision to extract real-world measurements from 2D images. It involves estimating a camera's internal parameters (like focal length and distortion) and external parameters (position and orientation). Calibration methods can be broadly classified into photogrammetric calibration, which uses a known 3D object, and self-calibration, which relies on image correspondences without a predefined object. These techniques help improve accuracy in applications like robotics, augmented reality, and 3D reconstruction.

This work mentions about calibrating a camera automatically just by using different views of the same scene. Homographies can be estimated through the multiple views and using a closed form equation capturing the relation between the points on the image and the 3D point in the world, the intrinsic and extrinsic matrices can be calculated, as mentioned in [1].

For our case, we can model the relation between the 2D image point and the 3D world point using the following equation

$$s\bar{x} = A\begin{bmatrix}R & t\end{bmatrix}\bar{X} \tag{1}$$

where s is an arbitrary scale factor, (R, t) are the extrinsic parameters which relate the world coordinate system to the camera coordinate system. A is the intrinsic matrix given by:

$$A = \begin{bmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

the vectors $\bar{x}$ and $\bar{X}$ are the homogenized 2D points in pixel coordinates and 3D points in the world respectively, and are given by

$$\bar{x} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \bar{X} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Without Loss of generality, it can be assumed that the image in the world lies on the plane Z = 0 of the world coordinate system. Since Z = 0, we can drop the term off entirely from $\bar{X}$, which can reduce the above equations to

$$\bar{x} = H\bar{X}$$

Where H is the Homogrpahy matrix given by

$$H = A\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t}\end{bmatrix}$$

Homography can be estimated from an image of a known planar pattern by matching model points to detected image points. Camera calibration then extracts the intrinsic matrix A from multiple homographies. Once the intrinsic matrix is known, we can calculate the extrinsics (Rotation and Translation) for each of the images using the above equation.

With these as the initial estimates, we can reproject the 3D points in the world frame into the image frame using the estimated matrices. We can further optimize the intrinsic and extrinsic values by minising the euclidian distance between the reprojected points and the original image points as a function of the parameters. This is done using using `optimize.least_squares` from `scipy`, with the `parameter method='lm'`.

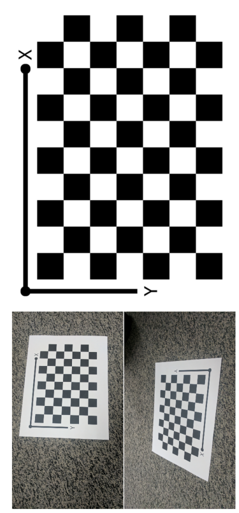

Fig. 1: Original Checkerboard pattern in the world frame (Above) and two of the different views of the pattern (Below) to estimate Homographies

II. Camera Calibration Method

This section describes the process of estimating the camera intrinsic matrix A and distortion parameters using a set of N model points across M images, each containing N image points.

A. Camera Intrinsics

The intrinsic matrix A has five free parameters: the principal point coordinates (c_x, c_y), the scaling factors along the u and v axes (f_x,f_y), and the skew factor (γ):

$$A = \begin{bmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Since the first two columns of the homography matrix H (h_1 and h_2) must be orthonormal, we obtain the following constraints:

$$h_1^\top A^{-\top} A^{-1} h_2 = 0$$

$$h_1^\top A^{-\top} A^{-1} h_1 = h_2^\top A^{-\top} A^{-1} h_2$$

where h_i represents the ith column of H.

To solve for A, we define the symmetric matrix B, which satisfies:

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

Since B is symmetric, it is characterized by a 6D vector:

$$b = \left[B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}\right]^\top$$

which allows rewriting the constraint equations as:

$$h_i^\top B h_j = v_{ij}^\top b$$

where:

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

Stacking these constraints for all M images results in the system:

$$Vb = 0$$

where V is a $2M \times 6$ matrix. The vector b is estimated using singular value decomposition (SVD) by selecting the right singular vector corresponding to the smallest singular value. Once b is determined, the intrinsic parameters can be recovered using the following expressions:

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

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

$$f_x = \sqrt{\frac{\lambda}{B_{11}}}$$

$$f_y = \sqrt{\frac{\lambda B_{11}}{B_{11}B_{22} - B_{12}^2}}$$

$$\gamma = -B_{12}\frac{{f_x}^2 f_y}{\lambda}$$

$$c_x = \frac{\gamma c_y}{f_y} - \frac{B_{13}{f_x}^2}{\lambda}$$

B. Camera Extrinsics

Once the camera intrinsics are determined, the extrinsic parameters can be directly obtained. From Equation of the Homography matrix from previous section, we derive:

$$\mathbf{r}_1 = \lambda A^{-1}\mathbf{h}_1$$

$$\mathbf{r}_2 = \lambda A^{-1}\mathbf{h}_2$$

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

$$\mathbf{t} = \lambda A^{-1}\mathbf{h}_3$$

Due to the noise in the measured values, the Rotation matrix constructed from these vectors does not satisfy the properties of a real Rotation matrix. To overcome this, SVD is used and the real roation matrix is obtained by the product of the left and right singular matrices of this rotation matrix.

C. Radial Distortion

So far, we have assumed an ideal camera without distortion, but real cameras exhibit significant radial distortion. We consider only the first two terms of distortion.

Let (u, v) be the distortion-free pixel coordinates and $(\hat{u}, \hat{v})$ the observed coordinates. Similarly, (x, y) and $(\hat{x}, \hat{y})$ are the ideal and distorted normalized coordinates. Radial distortion is modeled as:

$$\hat{x} = x + x\left[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2\right]$$

$$\hat{y} = y + y\left[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2\right]$$

Following the equations of the intrinsic matrix to convert image plane coordinates to pixel coordinates, we have

$$\hat{u} = u + (u - u_0)\left[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2\right]$$

$$\hat{v} = v + (v - v_0)\left[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2\right]$$

Here k_1 and k_2 are the parameters for radial distortion. Since we assumed that the camera has minimal distortion we can assume that both the values are 0 for a good initial estimate.

D. Non Linear Geometric Error minimization

Now that we have the initial estimates of the parameters, we can optimize them further by defining a pixel difference error over all corners and all images and optimizing it as follows:

$$\underset{f_x, f_y, \gamma, c_x, c_y, k_1, k_2}{\operatorname{argmin}} \sum_{i=1}^{N} \sum_{j=1}^{M} \| x_{i,j} - \hat{x}_{i,j}(A, R_i, t_i, X_j, k) \|$$

where $x_{i,j}$ and $\hat{x}_{i,j}$ are an inhomogeneous representation.

III. Results

Following the above process, the camera calibration was done by observing a checkerboard pattern in 13 different orientations as shown in fig.1.

A. Input Data

The method requires as input a set of world points obtained from the original corners of the chessboard in the world, and a set of image points which are obtained from the corners of the 13 images. To obtain this data, a grid size of $(9, 6)$ was selected from the chessboard. The function `cv2.findChessboardCorners` was then employed to detect the $(9, 6)$ grid of image points for each image, as illustrated in Fig.2

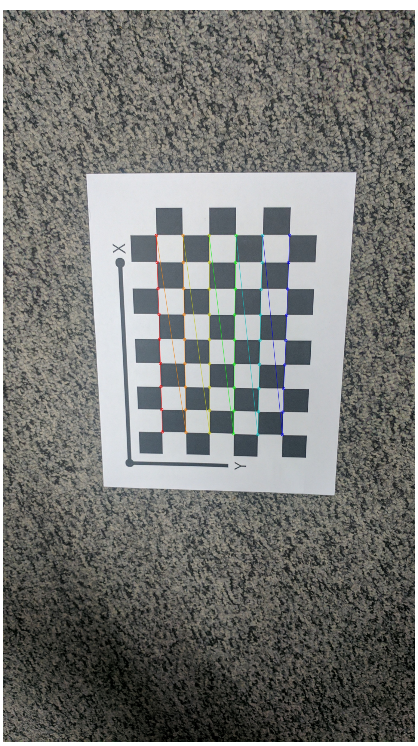

Fig. 2: Detected Corners in the Image using `cv2.findChessboardCorners`

Image	Mean Geometric Error	
	Unoptimized	Optimized
1	0.73	0.56
2	0.76	0.74
3	0.88	0.84
4	1.07	1.04
5	0.61	0.54
6	0.93	0.75
7	0.84	0.82
8	0.54	0.51
9	0.70	0.65
10	0.73	0.61
11	0.94	0.90
12	1.53	1.36
13	0.94	0.86
Mean	0.86	0.78

TABLE I: Reprojection Error

B. Reprojection Results and Optimized Parameters

The results of the Camera Calibration matrix before and after optimization are as follows:

$$A = \begin{bmatrix} 2083.69 & -2.56 & 745.93 \\ 0 & 2072.60 & 1359.88 \\ 0 & 0 & 1 \end{bmatrix}$$

$$A_{opt} = \begin{bmatrix} 2079.91 & -2.54 & 745.93 \\ 0 & 2069.45 & 1360.08 \\ 0 & 0 & 1 \end{bmatrix}$$

and the parameters for distortion are

$$k = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

$$k_{opt} = \begin{bmatrix} 0.071 & -0.406 \end{bmatrix}$$

The value of the reprojection error (the pixel difference error defined in the optimization function) is stored for each image when the image was reprojected using both the initial and optimized Camera Calibration matrices and the distortion coefficients and the values are tabluated as shown in I.

C. Reprojection Results - Images

The images of the reprojected corners of the original chessboard corners onto the 13 images are as follows.

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. doi: 10.1109/34.888718.

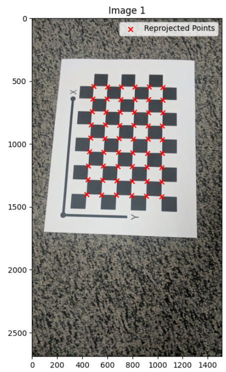

Fig. 3: Image 1: Rectified Image with Re-projected Corners

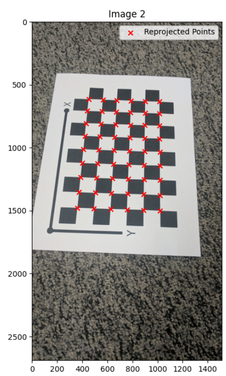

Fig. 4: Image 2: Rectified Image with Re-projected Corners

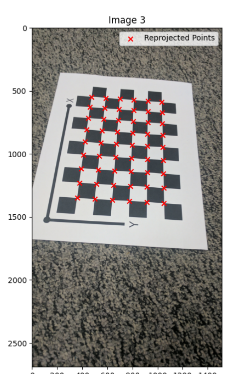

Fig. 5: Image 3: Rectified Image with Re-projected Corners

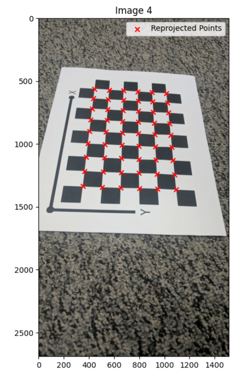

Fig. 6: Image 4: Rectified Image with Re-projected Corners

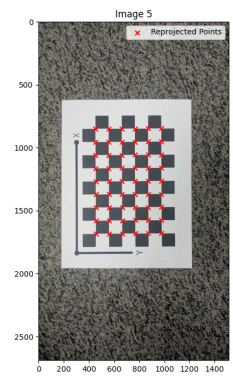

Fig. 7: Image 5: Rectified Image with Re-projected Corners

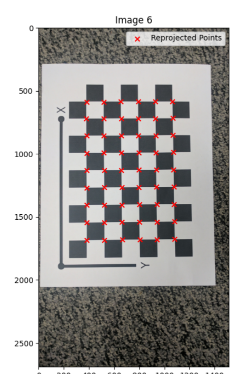

Fig. 8: Image 6: Rectified Image with Re-projected Corners

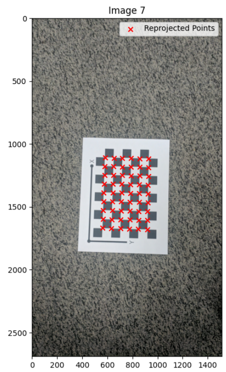

Fig. 9: Image 7: Rectified Image with Re-projected Corners

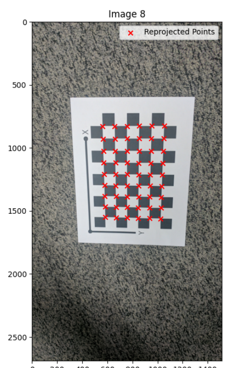

Fig. 10: Image 8: Rectified Image with Re-projected Corners

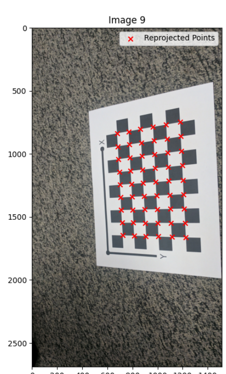

Fig. 11: Image 9: Rectified Image with Re-projected Corners

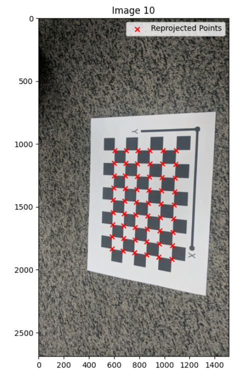

Fig. 12: Image 10: Rectified Image with Re-projected Corners

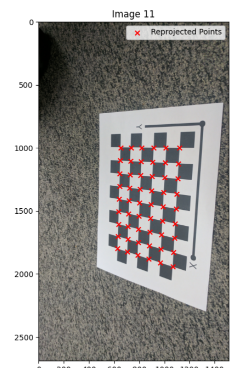

Fig. 13: Image 11: Rectified Image with Re-projected Corners

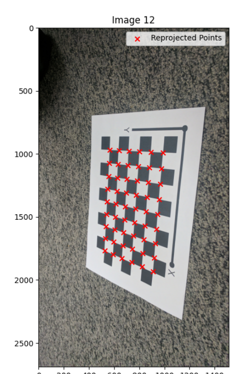

Fig. 14: Image 12: Rectified Image with Re-projected Corners

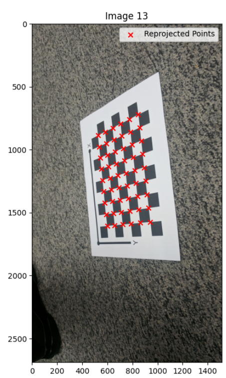

Fig. 15: Image 13: Rectified Image with Re-projected Corners