Homework 1 - AutoCalib
RBE/CS549 Computer Vision

Shambhuraj Mane
MS Robotics Engineering
samane@wpi.edu

***Abstract*—Camera calibration is an essential step for 3D computer vision. In this project, the intrinsic parameters (focal length, principal point, distortions) and extrinsic parameters (rotation and translation) of a camera using images of a checkerboard pattern are estimated. The technique outlined in Zhengyou Zhang's paper is used for this purpose and is divided in two stages, Initial Parameter Estimation and Non-linear Geometric Error Minimization. After calibration, accurate intrinsic parameters and a reprojection error are obtained, enabling the undistortion and rectification of images from the camera. This report presents the calibration procedure and results.**

- **Focal coordinates:** f_x, f_y
- **Principal point:** c_x, c_y
- **Radial distortion:** k_1, k_2
- **Rotation and translation:** R, t

Index Terms* — *Camera calibration, calibration from planes, 2D pattern, absolute conic, projective mapping, lens distortion, closed-form solution, maximum likelihood estimation

I. Introduction

Estimating parameters of the camera like the focal length, distortion coefficients and principle point is called Camera Calibration. It is one of the most time consuming and important part of any computer vision research involving 3D geometry.

We start with projecting 3D point from world coordinate to 2D point pixel coordinate.

$$x = P.X \tag{1}$$

Here, x denotes the image points and X denotes the world points (points on the checkerboard). P consists of intrinsic parameters- the coordinates of the principal point (u_0, v_0), the scale factors in image u and v axes (α and β), and the parameter describing the skewness of the two image axes (γ). and the camera matrix is expressed as a 3x3 matrix:

$$A = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

and extrinsic parameters- Rotation ($\mathbf{R}$) and Translation ($\mathbf{t}$)

$$[R|t] = \begin{bmatrix} r_11 & r_12 & r_13 & t_1 \\ r_21 & r_22 & r_23 & t_2 \\ r_31 & r_32 & r_33 & t_3 \end{bmatrix} \tag{3}$$

So, total $5 + 6 = 11$ DoF. Hence we usually need 6 points to find the projection matrix. In the zhang's method a known 3D object ie. a checkerboard pattern is used to calibrate the camera. The assumption here is that we have known 3D structure (size and structure) and the surface is flat (Z = 0). Initially we were having,

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & \gamma & u \\ 0 & \beta & v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_11 & r_12 & r_13 & t_1 \\ r_21 & r_22 & r_23 & t_2 \\ r_31 & r_32 & r_33 & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{4}$$

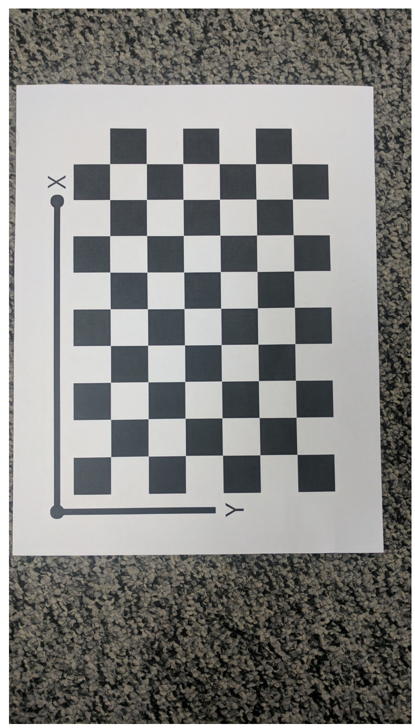

Fig. 1. Known Checkerboard structure

after considering the assumptions mentioned above we have,

$$\begin{aligned} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} &= \begin{bmatrix} \alpha & \gamma & u \\ 0 & \beta & v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_11 & r_12 & t_1 \\ r_21 & r_22 & t_2 \\ r_31 & r_32 & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \\ &= A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \end{aligned} \tag{5}$$

So now we have a homogeneous matrix (H) with 8 unknowns as follows,

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{6}$$

where,

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} = A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \tag{7}$$

To calculate the homogeneous matrix we will need 4 matching points from pixel coordinates(from checkerboard images) and world coordinate (known checkerboard structure). We can use this homogeneous matrix to calculate the intrinsic matrix K. Here, we are exploiting the fact that coordinate frame changes from frame to frame but within a frame we have same coordinate system.

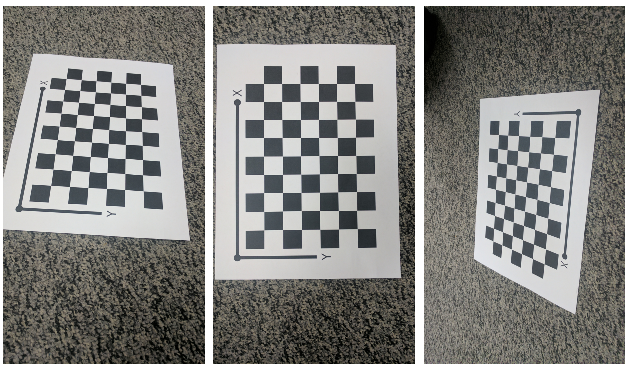

Fig. 2. Sample images for camera calibration

II. Initial Parameter Estimation

In this stage, we are trying to get a good initial estimate of the projection parameters so that we can feed it into the non-linear optimizer. We have known checkerboard structure of following dimensions:

- Checker board size, CHESS_BOARD_DIM = (9, 6)
- The size of Square, SQUARE_SIZE = 21.5 mm

This gives us the 3D point coordinates of world frame. Considering our assumption $Z = 0$ and using $\tilde{x}$ to denote the augmented vector by adding 1 as the last element we get,

$$\tilde{M} = \begin{bmatrix} X & Y & 1 \end{bmatrix}^T \tag{8}$$

Similarly from images, the corners of the Checker board are found using cv2.findChessboardCorners function in OpenCV. These corners are the 2D point coordinates of image frame denoted by

$$\tilde{m} = \begin{bmatrix} u & v \end{bmatrix}^T \tag{9}$$

Hence, we have -

$$\begin{aligned} s\tilde{m} &= H\tilde{M} \\ &= K \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \end{aligned} \tag{10}$$

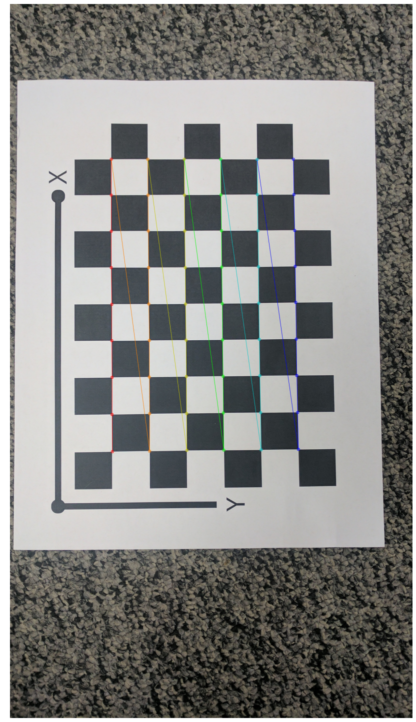

Fig. 3. Corners detected in Checkerboard pattern

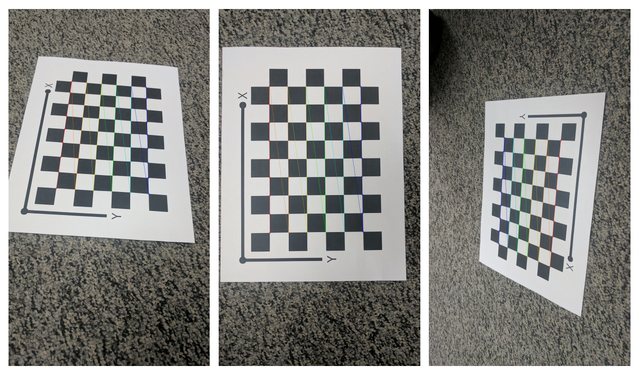

Fig. 4. Corners detected in Checkerboard sample images

where s is an arbitrary scale factor.

We know that r_1 and r_2 are from rotational matrix and r_1, r_2 and r_3 form an orthonormal basis. These facts add two constraints

- $r_1^T r_2 = 0$
- $\|r_1\| = \|r_2\| = 1$

Using this knowledge, we have

$$h_1^T A^- T h_2 = 0 \tag{11}$$

$$h_1^T A^- T A^- 1 h_1 = h_2^T A^- T A^- 1 h_2 \tag{12}$$

Let

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

If we can calculate B matrix we can compute the intrinsic matrix parameters as follows,

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

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

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

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

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

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

For computing B matrix, we reformulate equation 11 and 12 as,

$$v_{12}^T b = 0 \tag{14}$$

$$v_{11}^T b - v_{22}^T b = 0 \tag{15}$$

where, v_{11}, v_{12}, v_{22} are vectors resulted from

$$\mathbf{v}_{ij} = \begin{bmatrix} h_{i1}h_{j1} \\ h_{i1}h_{j2} + h_{i2}h_{j1} \\ h_{i2}h_{j2} \\ h_{i3}h_{j1} + h_{i1}h_{j3} \\ h_{i3}h_{j2} + h_{i2}h_{j3} \\ h_{i3}h_{j3} \end{bmatrix}^T \tag{16}$$

Therefore, the two fundamental constraints 11 and 12, from a given homography, can be rewritten as 2 homogeneous equations in b as follows

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} \mathbf{b} = 0 \tag{17}$$

If n images of the model plane are observed, by stacking n such equations as (8) we have

$Vb = 0$

where V is a 2n × 6 matrix. This is similar to solving homogeneous matrix by Singular Value Decomposition (SVD). In calculation of homogeneous matrix we were having 8 unknowns and we need 4 matching points while in calculation b matrix we have 6 unknowns hence we need minimum 3 different views ie. minimum 3 images taken from different views to get a unique solution b.

Once A is known, the extrinsic parameters for each image is readily computed.

$r_1 = \lambda \mathbf{A}^{-1}\mathbf{h}_1$

$r_2 = \lambda \mathbf{A}^{-1}\mathbf{h}_2$

$r_3 = r_1 \times r_2$

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

The initial distortion is assumed to be zero and therefore

$$k_c = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{18}$$

```
● PS C:\sam\WPI\COURSEWORK\CV\Assignemnts\HW1> python3 .\Wrapper.py
  intrinsic matrix
  [[ 2.06525652e+03 -2.93974703e+00  7.64676160e+02]
   [ 0.00000000e+00  2.05348352e+03  1.36276925e+03]
   [ 0.00000000e+00  0.00000000e+00  1.00000000e+00]]
  distortion coefficients
  [0, 0]
  initial guess
  [2065.2565244    -2.93974703 2053.48351884  764.67616031 1362.76924937
      0.            0.        ]
```

Fig. 5. Intrinsic matrix and initial guess of distortion coefficients

Now we have our initial parameter estimation of intrinsic parameters and distortion coefficients and we can feed it into the non-linear optimizer.

Sample image	1	2	3	4	5	6
Projection error	0.6458	1.0333	0.7027	0.6073	0.6846	1.7538

TABLE I
PROJECTION ERROR BEFORE OPTIMIZATION.

III. NON-LINEAR GEOMETRIC ERROR MINIMIZATION

We have the initial estimates of A, R, t, k_c now we want to minimize the geometric error defined as given below

$$\sum_{i=1}^{N}\sum_{j=1}^{M} \| x_{i,j} - \hat{x}_{i,j}(A, R_i, t_i, X_{j,k}) \| \tag{19}$$

Here $x_{i,j}$ and $\hat{x}_{i,j}$ are an inhomogeneous representation. $x_{i,j}$ is the 2D point coordinates of image frame and $\hat{x}_{i,j}(A, R_i, t_i, X_{j,k})$ is the projection of 3D point coordinates of world frame on image frame considering our initial estimates of A, R, t, k_c. Formally, the optimization problem is as follows:

$$\arg \min_{f_x, f_y, c_x, c_y, k_1, k_2} \sum_{i=1}^{N}\sum_{j=1}^{M} \| x_{i,j} - \hat{x}_{i,j}(K, R_i, t_i, X_{j,k}) \| \tag{20}$$

For optimization we are calculating the projection error considering the radial distortion represented as follows,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x(1 + k_1 r^2 + k_2 r^4) \\ y(1 + k_1 r^2 + k_2 r^4) \end{bmatrix} \tag{21}$$

with,

$$r^2 = x^2 + y^2 \tag{22}$$

Then the projection of 3D point coordinates of world frame on image frame $\hat{x}_{i,j}$ is can be represented as,

$$\hat{x}_{i,j} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} x' f_x + u_0 \\ y' f_y + v_0 \end{bmatrix} \tag{23}$$

The problem defined in (21) is a nonlinear minimization problem,which is solved with the Levenberg-Marquardt Algorithm as implemented in scipy.optimize.least_squares.

After rectification of error caused due to radial distortion, the image is undistorted using cv.undistort() function and the 3D points are re-projected on the images we used for calibration. The results are undistorted image and precisely projected corner points on the image.

IV. Results

Intrinsic matrix calculated for given images is

$$A = \begin{bmatrix} 2.0652 \times 10^3 & -2.93974 & 7.64676 \times 10^2 \\ 0.00000 & 2.05348 \times 10^3 & 1.36276 \times 10^3 \\ 0.00000 & 0.00000 & 1.00000 \end{bmatrix}$$

Distortion coefficients are

$$K = \begin{bmatrix} 2.06522 \times 10^{-}2 \\ -1.40592 \times 10^3 \end{bmatrix}$$

Sample image	1	2	3	4	5	6
Projection error	0.5214	0.9850	0.5537	0.6073	0.5137	1.6341

TABLE II
PROJECTION ERROR AFTER OPTIMIZATION.

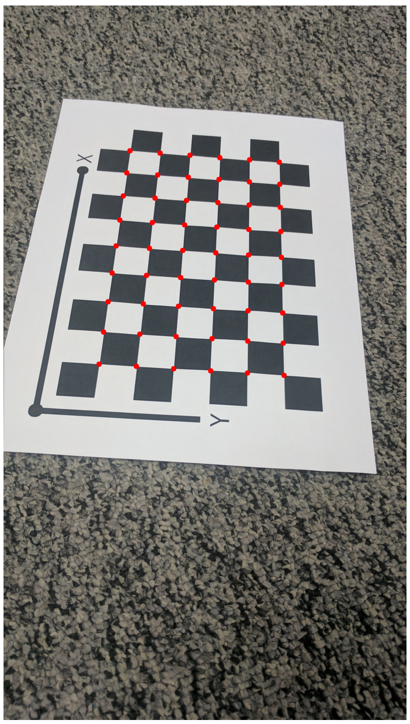

Fig. 6. Re-projected points on image

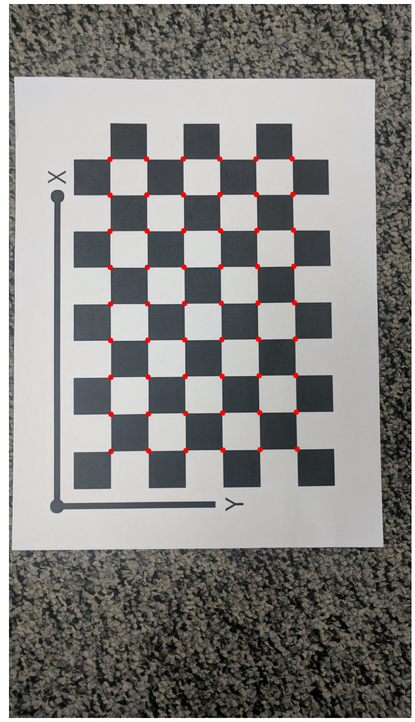

Fig. 7. Re-projected points on image

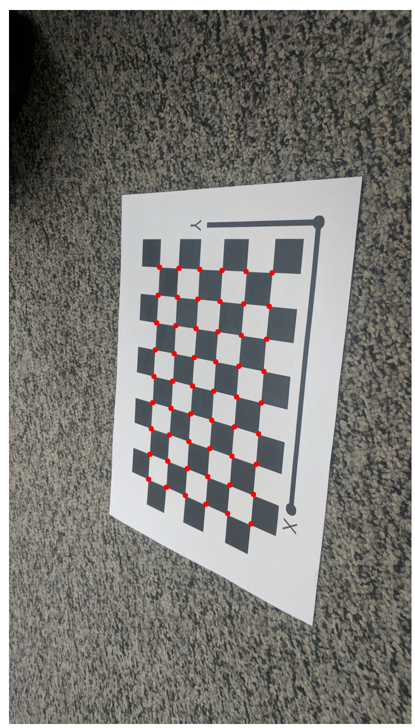

Fig. 8. Re-projected points on image

References

[1] Z. Zhang, "A flexible new technique for camera calibration," IEEE Trans. Pattern Anal. Machine Intell., vol. 22, no. 11, pp. 1330–1334, Nov. 2000, doi: 10.1109/34.888718.
[2] OpenAI. (2022). ChatGPT. https://www.openai.com/chatgpt