RBE 549 - Computer Vision HW1 - AutoCalib

Ashwin Disa
M.S Robotics Engineering
Worcester Polytechnic Institute (WPI)
Worcester, MA 01609
Email: amdisa@wpi.edu

***Abstract*—Camera calibration is one of the most time consuming and important part of any computer vision research involving 3D geometry. In this Homework, we estimate the camera parameters like focal length, distortion coefficients and principle point, also known as camera calibration. An efficient and robust camera calibration technique was presented by Zhengyou Zhang of Microsoft which is implemented here.**

I. Methodology

Camera calibration is the process of estimating any camera model's intrinsic, extrinsic and distortion parameters. The focal length and principal point position are the intrinsic parameters and distortion coefficients are the distortion parameters. The intrinsic parameter matrix is given by A as shown below -

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

Fig. 1: Camera intrinsic matrix.

Here, α and β are the scale factors in u and v axes, γ is the skew parameter and (u_0, v_0) are the coordinates of the principal point.

Zhang's paper relies on a calibration target (checkerboard in our case) to estimate camera intrinsic parameters. The checkerboard is printed on an A4 paper and the size of each square is 21.5mm. We have a 9×6 grid for computation. Thirteen images taken from a Google Pixel XL phone with focus locked are used to calibrate.

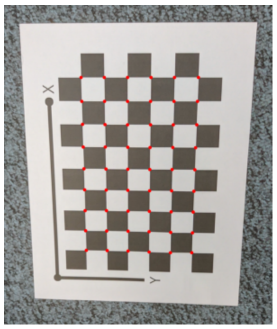

Fig. 2: corners using $cv2.findChessboardCorners()$

First, we estimate the initial parameters so that we can feed it into the nonlinear optimizer. We read the given images and find the checkerboard corners using the $cv2.findChessboardCorners()$. We find a total of 54 corners in each of 13 images. One of the images is shown in Fig. 2.

Each square on the checkerboard is of side 21.5mm. Using this information, we define the world corners i.e corners in the frame shown in Fig. 2. Same as the corners in the image we find 54 corners in the world frame. Without loss of generality, we assume the model plane is on $Z = 0$ of the world coordinate system. The transformation between the world plane and image plane is shown below.

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

Fig. 3: Transformation between model and image plane.

We calculate the homogeneous transformation matrix between the model and image plane using the $cv2.findHomography()$ function. The H matrix is the matrix multiplication of the matrix A and rotation with translation matrix which is denoted as $\begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix}$.

To find the B matrix we use the homography. The i^{th} column vector of H be $h_i = \begin{bmatrix} h_{i1} & h_{i2} & h_{i3} \end{bmatrix}$, We use the two equations below to get 2 homogeneous equations in b. The shape of this matrix is (26×6) as we have 13 images.

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

Fig. 4: B matrix.

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

Fig. 5: Homogeneous equations.

Once we have the B matrix, we can easily estimate the elements of the K matrix using the equations given in Appendix B of Zhang's paper. The initial estimate of the K matrix is shown below.

$$K_{initial} = \begin{bmatrix} 2068.8 & -3.111 & 765.22 \\ 0.0 & 2056.6 & 1359.9 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

We use the K matrix and the Homography obtained earlier to estimate the extrinsic parameters such as the R and t matrices as shown in the equations below. We column stack the individual vectors to form a matrix of shape (3×4) in the form $\begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix}$.

$$\begin{aligned} \mathbf{r}_1 &= \lambda \mathbf{A}^{-1}\mathbf{h}_1 \\ \mathbf{r}_2 &= \lambda \mathbf{A}^{-1}\mathbf{h}_2 \\ \mathbf{r}_3 &= \mathbf{r}_1 \times \mathbf{r}_2 \\ \mathbf{t} &= \lambda \mathbf{A}^{-1}\mathbf{h}_3 \end{aligned}$$

Fig. 6: Matrix R formation.

We assume the distortion to be $(0, 0)$ for the first iteration. The above solution is obtained through minimizing an algebraic distance which is not physically meaningful. The maximum likelihood estimate can be obtained by minimizing the following functional.

$$\sum_{i=1}^{n}\sum_{j=1}^{m} \|\mathbf{m}_{ij} - \breve{\mathbf{m}}(\mathbf{A}, k_1, k_2, \mathbf{R}_i, \mathbf{t}_i, \mathtt{M}_j)\|^2$$

Fig. 7: Equation for nonlinear minimization problem.

where $m(A, k_1, k_2, R_i, t_i, M_j)$ is the projection of point M_j in image i. This is a nonlinear minimization problem we solve using the $scipy.optimize.leastsquares()$ function. Basically, we minimize the pixel distance between the corner pixel value from the $cv2.findChessboardCorners()$ and the projected co-ordinates. The final K matrix is shown below.

$$K_{final} = \begin{bmatrix} 2068.8 & -3.112 & 765.23 \\ 0.0 & 2056.6 & 1360.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

And the distortion parameters $k = \begin{bmatrix} 0.0167 \\ -0.1154 \end{bmatrix}$

And the mean re-projection error = 0.7885.

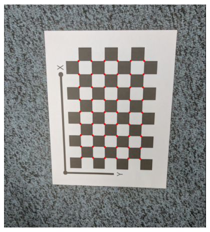

Fig. 8: Rectified Image 1

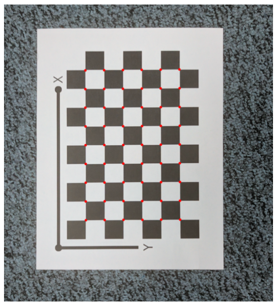

Fig. 9: Rectified Image 2

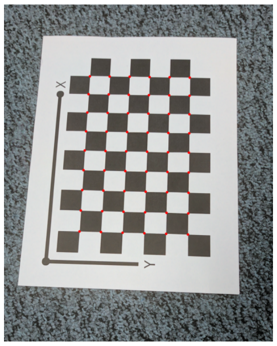

Fig. 10: Rectified Image 3

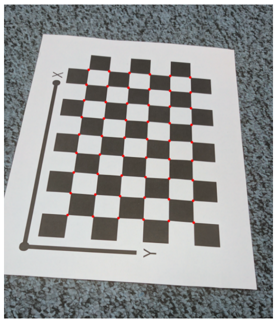

Fig. 11: Rectified Image 4

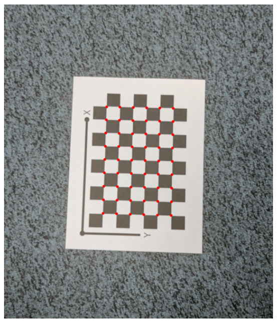

Fig. 12: Rectified Image 5

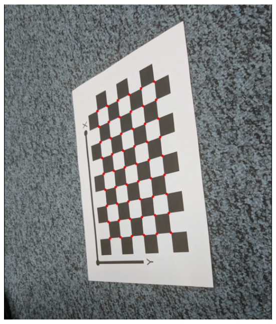

Fig. 13: Rectified Image 6

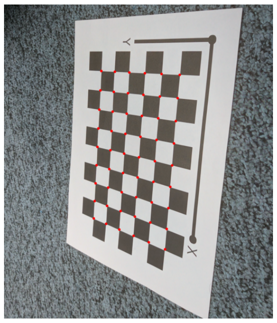

Fig. 14: Rectified Image 7

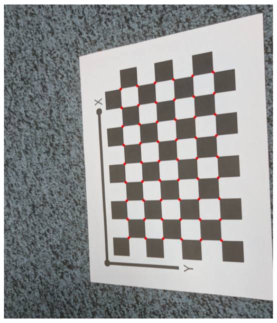

Fig. 15: Rectified Image 8

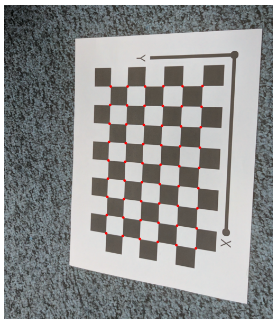

Fig. 16: Rectified Image 9

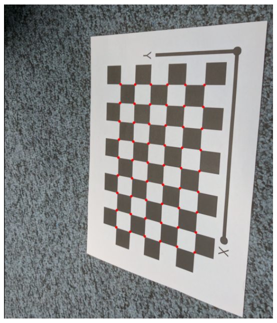

Fig. 17: Rectified Image 10

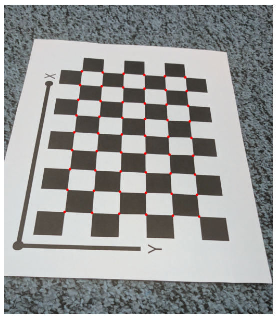

Fig. 18: Rectified Image 11

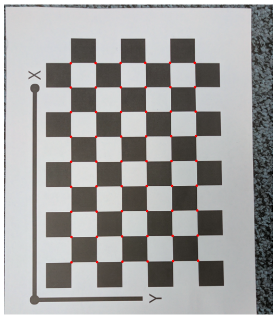

Fig. 19: Rectified Image 12

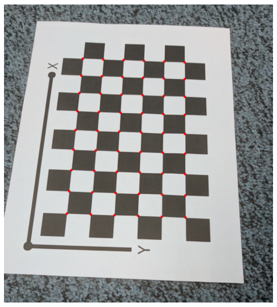

Fig. 20: Rectified Image 13