RBE549: Homework 1 - Autocalib Zhang's method

Chinmay Sunil Kate
Robotics Engineering Department
Worcester Polytechnic Institute
cskate@wpi.edu

I. Introduction

In Camera Calibration we estimate the parameters of camera by determining relationship between a 3D point in the real world and its corresponding 2D projection, in the image captured by Calibrated Camera. Parameters such as focal length, distortion coefficient, principle points are the intrinsic parameters, Rotational and Translation matrix accounts for extrinsic parameters.

$$P(X,Y,Z) \overset{\{\text{Rigid Transform}\}}{\longrightarrow} P(X,Y,Z)\, w.r.t.camera's frame \overset{\{\text{Projective Transform}\}}{\longrightarrow} p(u,v)$$

Fig. 1: Projection from 3D World to 2D Ima⁻⁻

$$\begin{bmatrix} \alpha & \gamma & u_c \\ 0 & \beta & v_c \\ 0 & 0 & 1 \end{bmatrix}$$

Fig. 2: Camera Matrix A having focal lengths, skews, camera center.

II. Data

Zhang's paper relies on calibration target to estimate camera intrinsic parameters. CheckerboardPattern was printed on A4 paper and size of each square was 21.5mm. 13 Images taken from Google Pixel XL phone with focus locked were used to calibrate.

III. Initial Parameters estimation

We first get the image points as well as the World points. Image points are taken using the openCV function "findChessboardCorners". We find the corresponding world points the the above points. While finding the initial Parameters, we also presents k1 and k2 as radial distortion parameters. Figure 3 corresponds to simplified equation of transformation from world to image point. We then further split M matrix to homogenous matrix containing A matrix wit Z=0. Here A Projective transform 3x3 matrix, [R—T] is 3x4 matrix, P is 4x1 matrix, p is 3x1 matrix.

$$p(u,v) = M.\,P(X,Y,Z)$$

Fig. 3: Simplified equation of Projection.

$$p(u,v,1) = M.\,P(X,Y,Z,1)$$

$$p = A.\,[R|t].\;P$$

Fig. 4: Homogeneous coordinate space with A matrix.

$$p_{3\times1} = A_{3\times3}.\,[R - R_{:,3}|t]_{3\times3}.\,[P_{P-Z}]_{3\times1}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} R_{00} & R_{01} & T_{03} \\ R_{10} & R_{11} & T_{13} \\ R_{20} & R_{21} & T_{23} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Fig. 5: Reduced equation after eliminating 3rd column of Rotation—Translation matrix.

A. Estimating Homography H matrix

Homography is calculated for all 13 Images using Direct linear transformation, here we also normalize the points around it's mean. Figure 6 shows the equation of Homography which converts points from one coordinate space to another. This computes matrix M.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Fig. 6: Homography matrix

B. Estimating B matrix

B matrix is calculated by dot product of A transpose into A inverse. Since A is triangular matrix, B is therefore symmetric matrix.

$$B = \begin{pmatrix} B_0 & B_1 & B_3 \\ B_1 & B_2 & B_4 \\ B_3 & B_4 & B_5 \end{pmatrix} \text{ or } \begin{pmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{pmatrix}$$

Fig. 7: B Matrix which is symmetric matrix.

C. Estimating A matrix

In A matrix we compute alpha, beta, gamma, uc, vc from the B matrix. Each has the equations and matrix A is formed which is 3x3 having alpha, beta, gamma, uc, vc terms in it.

$$A = \begin{bmatrix} \alpha & \gamma & u_c \\ 0 & \beta & v_c \\ 0 & 0 & 1 \end{bmatrix}$$

Fig. 8: A Matrix.

D. Estimating Rotational-Translation matrix

This matrix can be computed using the Homography matrix columns and A matrix. Figure 9 provides the equation to calculate the extrinsic parameters. Here h1, h2, h3 are columns of Homography matrix.

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

Fig. 9: To calculate Extrinsic Parameters.

E. Non-linear Geometric Error Minimization

This is optimization problem, Now we have initial estimates and now we want to minimize geometric error defined in fig 10. We have inhomogeneous representation. We optimize the equation using scipy.optimize.leatsquares function in openCV.

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

Fig. 10: Argument to calculate Non-linear error

F. Conclusion

The mean error after calibration was 0.6976324. The reprojected points are presented in the following figures.

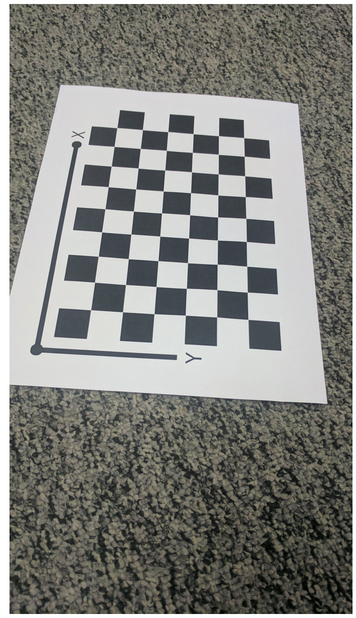

Fig. 11: Original Image.

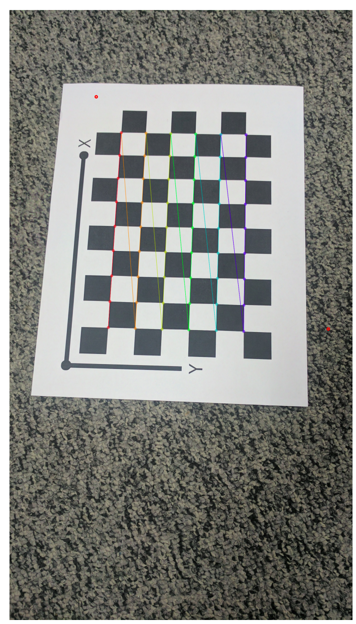

Fig. 12: Detected and Projected corners.