# RBE/CS549: P1 - AutoCalib

Shrishailya Chavan
*WPI Robotics Engineering*
*Worcester Polytechnic Institute*
schavan@wpi.edu
Using 1 late Day

***Abstract*—This assignment is motivated from paper A Flexible New Technique for Camera Calibration by Zhengyou Zhang where we are estimating the camera intrinsic, extrinsic parameters and distortion coefficients. The calibration target image used in this paper is a checkerboard pattern with each square size of 21.5 mm. First, we calculate the camera intrinsic matrix (K) and using that, the extrinsic parameters Rotation (R) and translation (t) are estimated. Then, using these as initial estimates, non linear optimization is done to minimize the geometric error and refine the intrinsic matrix parameters and the distortion coefficients.**

***Index Terms*—Calibration, Intrinsic, Optimization, Distortion, Geometric error, Matrix**

## I. Initial Parameter Estimation

Here we are provided with calibration target which is checkerboard, the target is printed on A4 paper and size of each square is 21.5mm. The Y axis on the checkerboard has odd number of squares and X axis has even number of squares. It is a general practice to neglect the outer squares (the row and columns of squares on the edges of the checkerboard). For example, if you have a 5×8 grid, you'll only consider the inner 3×6 grid for computation. Furthermore, we have been provided with a dataset of 13 images which are clicked from Google Pixel XL phone which we will use to calibrate.

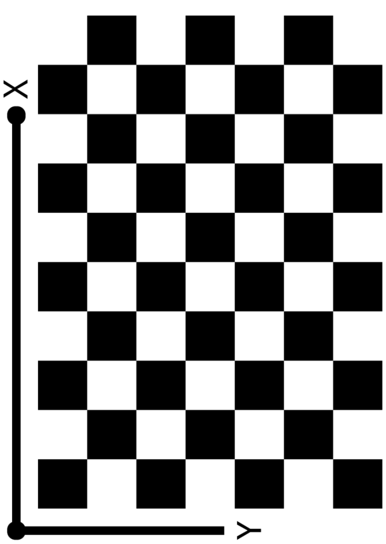



Fig. 1. Checkerboard Image

Here, the first step is to find the pixel coordinates of all checkerboard corners using the cv2.findChessboardCorners function. The patternsize parameter is (9,6) which is total number of inner corners that need to be detected. Totally we have 54 corner points for each image.

Furthermore, we have to get the world points of all the corners in checkerboard, which are 3D points. Here we have Z value as 0 for for the plane of chessboard. The values of X and Y may vary as the number of squares gets increased in chessboard. The size of each chessboard square is 21.5 mm. The value of world points is as follows:

[0, 0, 0], [21.5, 0, 0], [43, 0, 0], [64.5, 0, 0], ....[172, 0, 0],

[0, 21.5, 0], [21.5, 21.5, 0], [43, 21.5, 0], ......[172, 21.5, 0], . . . . [0, 107.5, 0], [21.5, 107.5, 0], [43, 107.5, 0], ..[172, 107.5, 0]

Here, we now have the relation between 2d points and 3d points, with this we can plot the homography matrix between two planes, this can be acheived using DLT(Direct Linear Transform). Here, we have a L matrix which is 2nx9 where n is number of points given.

to get H matrix we solve for LH = 0. Further by using singular value decomposition(SVD) on L which is L = US(V).T. The last singular vector of V is the solution to H.

Once we get the Homography matrix we can solve for V which is

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

Fig. 2. Equation to calculate V matrix

where we have the ith column of the homography matrix as

$$\mathbf{h}_i = [h_{i1}, h_{i2}, h_{i3}]^T .$$

Fig. 3. Equation to find ith column of Homography matrix

further the whole system can be represented as 2 Homogenous equations in b as follows:

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

To stack n images we have n equations we have the system Vb = 0, this solved for b taking the SVD of the V matrix and obtaining the right singular vector which is associated with smallest single value. The value of vector b is as follows:

$$\mathbf{b} = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T .$$

Fig. 4. Equation of Vector b

The use of this vector is used to estimate various parameters such as principal points, alpha, beta, gamma and lambda.

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

Fig. 5. Values of intrinsic Parameters

Then, further we have the Intrinsic Parameter Matrix which is denoted by A as follows:

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

Fig. 6. Equation of Intrinsic Parameters(A)

The results obtained for the above are as follows:

Below are the values of Parameters calculates using b,

```
u -  762.798540920507
v -  1351.6444606098087
lmda -  -0.6335494523874317
alpha -  2053.041145040028
beta -  2037.101962132167
gamma -  -0.46828815355567627
```

Fig. 7. Values of Parameters

In the above figure we have u, v, lamda, aplha, beta and gamma(skewness).

and we have the initial estimate of Calibration matrix as follows:

```
Initial estimation of Calibration matrix is

[[ 2.05304115e+03 -4.68288154e-01  7.62798541e+02]
 [ 0.00000000e+00  2.03710196e+03  1.35164446e+03]
 [ 0.00000000e+00  0.00000000e+00  1.00000000e+00]]
```

Fig. 8. initial Estimate of calibration matrix

In the above figure we have

f1 = 2053.04115 and f2 = 203710196

Further the rotation matrix R and the translation vector t for images can be estimated using the Intrinsic matrix(A) by the following equations:

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

In the above equation the values of Lambda are as follows:

$$\lambda = 1/\|A^{-1}h_1\| = 1/\|A^{-1}h_2\|.$$

and the value of r1, r2, r3 can be estimated using:

$$\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$$

As mentioned in the paper that k1 and k2 values must be zero initially so that camera has minimal distortion:

Kc = [0, 0].T

In the above Kc equation you can see that both the values of k1 and k2 are 0.

## II. Non-linear Geometric Error Minimization

Here, we firstly calculate the Reprojection error before performing the optimization of the Intrinsic Parameter matrix. To calculate error we have (u,v) as detected corner pixel coordinates, A is the intrinsic matrix , R is the rotation matrix and X, Y are the 3D coordinate points which correspond to the pixel coordinates. The error is obtained by taking the 12 norm of the resulting vector.

The above process is similar to the execution of following functional over all the points and all the images.

Further, we have Maximal Likelihood estimation which assumes that the images points are corrupted by the independent and identically disturbed noise. Following is the equation to the MLE:

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

Further, the error over all the points and images is summed up and the resultant error is divided by (594 = 54 * 11) to get the mean reprojection error.

The value of mean reprojection error before the optimization is 0.7645837759185026

Furthermore, to reduce the error, the function needs to be minimized for the error while dealing with radial distortion. Further, to obtain error due to the distortion, we have to calculate the following variables:

$$\breve{u} = u + (u - u_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]$$
$$\breve{v} = v + (v - v_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] .$$

In the above equations we have k1 and k2 as the radial distortion coefficients. (u0, v0) are denoted for the principal point on the image. (x, y) are estimated by multiplying the transformation matrix with the 3D coordinated which are (X, Y), the equation for above is

[x y 1] = [R—t][X Y 0 1].T

here above R, t are Rotation matrix and t is translation vector.

Further, we calculate (u, v) by

[u v 1].T = A[x y 1].T

here above A is intrinsic matrix.

Further the equation for the error calculation is as follows:

$$\begin{bmatrix} (u-u_0)(x^2+y^2) & (u-u_0)(x^2+y^2)^2 \\ (v-v_0)(x^2+y^2) & (v-v_0)(x^2+y^2)^2 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} \breve{u}-u \\ \breve{v}-v \end{bmatrix} .$$

Here above in the equation we have (u, v) as the ideal (non-observable distortion-free) pixel image coordinates, and (˘u, v˘) are the corresponding real observed image coordinates.

As we did earlier, the mean projection error which was the mean of summation of all the errors. Furthermore, we need to minimize the error which can be done by using scipy.optimize.leastsquares with the cost function returning the mean projection error taking the radial distortion coefficients into consideration.

One thing that needs to be keep in consideration is that intrinsic parameters and distortion coefficients are the only estimates that help in minimizing the error.

The following results shows the minimalization of error

Below is the result of Calibration Matrix after optimization

```
Calibration matrix after optimization is

[[1.33425935e+03 2.54865782e+03 8.00273178e+02]
 [0.00000000e+00 7.13201492e+02 1.34835486e+03]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]
```

Fig. 9. Result of Calibration Matrix after optimization

From the above results we get the principal point estimates after optimization as follows: (u0, v0) = (800.273178, 1348.35486)

and the value of Focal Lengths are as follows:

(fx, fy) = (1334.25935, 713.201492)

and the skewness is 2548.65782

The value of distortion coefficients after optimization are as follows:

```
Distortion coefficients after optimization:
0.0005697732270268874, -0.00012856266995392543
```

Fig. 10. Distortion coefficients after optimization

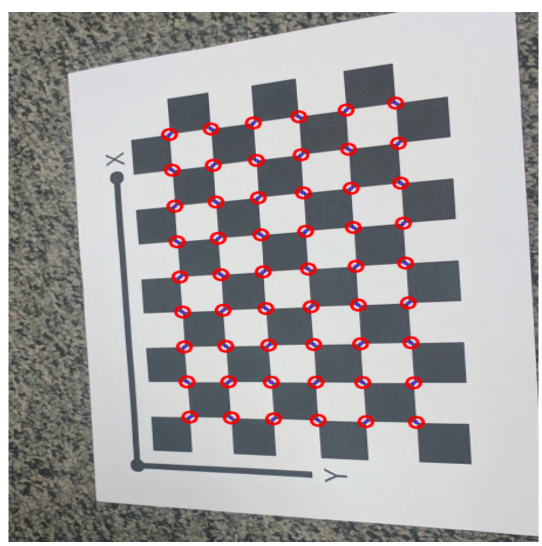
From above figure we get the distortion coefficients as follows: (k1, k2) = (0.0005697732270268874, -0.00012856266995392543)

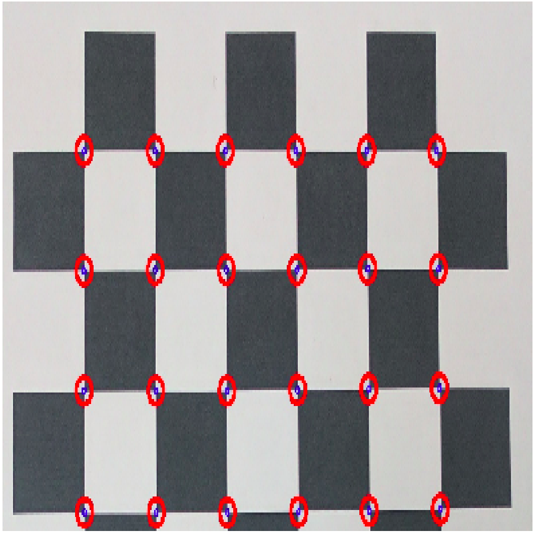
and finally the mean Reprojection error after optimization is 0.7375790546212971

```
Mean Reprojection error after optimization is 0.7375790546212971
```

Fig. 11. Mean Reprojection error after optimization

Below is the image of distorted image with Reprojected points and original detected corners that are plotted on the original images. The original detected corners are represented by Red circles and the Reprojected points are denoted by blue circles.





## REFERENCES

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

[2] https://towardsdatascience.com/estimating-a-homography-matrix-522c70ec4b2c

[3] https://kushalvyas.github.io/calib.html