Home Work 1- AutoCalib

Manoj Velmurugan
Robotics Engineering
Worcester Polytechnic Institute
Email: v.manoj1996@gmail.com

***Abstract*—(using one late day)**
Camera calibration is a crucial step in computer vision systems, ensuring accurate interpretation of images for various applications. In this work, we present an automated approach for calibrating camera intrinsics and distortion parameters. Leveraging advanced algorithms and techniques, our method achieves precise estimation of these parameters, leading to highly accurate results as demonstrated by the reprojection image. By automatically calibrating the camera, our approach streamlines the calibration process, reducing the need for manual intervention and ensuring consistent and reliable results across different imaging setups.

I. Algorithm

The flow of the program logic is provided in Fig 1

A. Homography Estimation

Homography estimation is a fundamental task in computer vision, essential for tasks such as image rectification, image stitching, and augmented reality. In this project, I employed OpenCV's findChessboardCornersSB function and the Direct Linear Transform (DLT) algorithm to estimate the homography matrix between world coordinate points and image points.

I utilized OpenCV's findChessboardCornersSB function to accurately detect the corners of a checkerboard pattern in the provided images. This function provides subpixel accuracy, ensuring precise localization of corner points even in images with noise or distortion.

To establish the world coordinate system, I selected the very first detected corner point near the black square as the origin in the world frame. Using the known length of the square (0.0215 m), I estimated the world points of the other corners. For simplicity, I assumed the Z-coordinate to be 0 without loss of generality.

I composed the Direct Linear Transform (DLT) algorithm, similar to previous course projects, to estimate the homography matrix. The DLT algorithm takes as input the world coordinate points and their corresponding image points. To solve for the homography matrix, I employed the null space trick, which involves computing the right singular vector corresponding to the least singular value (close to 0) of the resulting system. This vector, reshaped into the homography matrix H. H, represents the transformation between the world coordinate points and the image points. H was normalized with its last element.

B. Initial Guess - Intrinsics

Let the extrinsic matrix be given by,

$$RT = [r_1, r_2, r_3, t] \tag{1}$$

Homography matrix is related to intrinsic matrix K and homography matrix H as follows,

$$H = \lambda K[r_1, r_2, t] \tag{2}$$

where, λ is a scalar multiplier. Since r1 and r2 vectors form an orthonormal basis, we can write,

$$r_1.r_1 = 1 = r_2.r_2; r_1.r_2 = 0 \tag{3}$$

Alternatively,

$$(K^{-1}H_1)^T)(K^{-1}H_2) = 0 \tag{4}$$

$$(K^{-1}H_1)^T)(K^{-1}H_1) = (K^{-1}H_2)^T)(K^{-1}H_2) \tag{5}$$

This can be posed in the following form,

$$Ah = 0 \tag{6}$$

Where, h is 9x1 flattened vector formed from H matrix.
This equation was solved using nullspace trick. The right singular vector corresponding to the least eigen value (last row of vh of A) was computed to estimate a h. I reshape it into a matrix and call it H_0

C. Initial Guess - Extrinsics

I established the initial guess for the extrinsic parameters using known intrinsic parameters and the homography matrix H. First, I inverted the intrinsic matrix K and multiplied it with H to obtain the rotation and translation matrix $[r_1, r_2, t]$. I then scaled this matrix by a factor λ. Finally, I computed the third column of the rotation matrix r_3 by taking the cross product of r_1 and r_2, completing the extrinsics matrix.

D. Lens distortion

From the known intrinsic and extrinsic matrix, the world coorinates were projected onto ideal image coordinates. The image coordinates were converted to normalized image coordinates to avoid numerical issues with high pixel values and then distorted using the following equation,

$$x_{\text{distorted}} = x\left(1 + k_1 r^2 + k_2 r^4\right) \tag{7}$$

$$y_{\text{distorted}} = y\left(1 + k_1 r^2 + k_2 r^4\right) \tag{8}$$

Where, r is the radial distance from the center of the image.
Distorted normalized image coordinates were converted to the original image coordinates by offsetting with image center (from intrinsic).

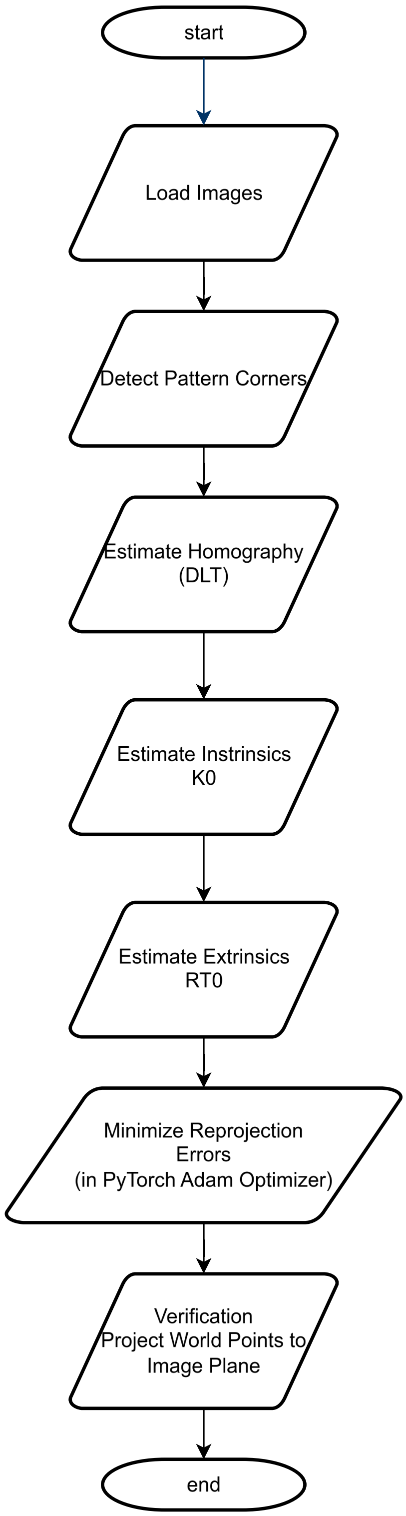

Fig. 1: Flow Chart

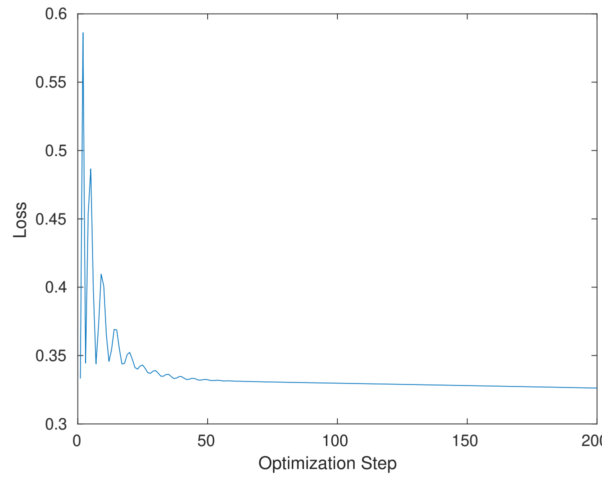

Fig. 2: Loss vs optimization iterations

E. Minimize Reprojection Errors

Error function was composed to perform the following,

- Project world coordinate into image coordinate using the provided intrinsics and extrinsics.
- Distort the image coordinates with the lends distortion model.
- Compute the norm of projection error over all the corner points across the entire image set.

The error function was written using PyTorch CPU Tensors in an efficient vectorized manner. The following error function was minimized using PyTorch Adam optimizer with a 1e-4 lr for 200 iterations.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(uv_i - \hat{uv}_i)^2 \tag{9}$$

The loss curve from optimization is given in Fig. 2. The K obtained from initial estimation were already good and the extra optimization refined it further as shown by the loss curve.

II. Results

Estimated intrinsic matrix is given by,

$$K = \begin{pmatrix} 2057.12702 & -1.272264 & 763.073467 \\ 0 & 2042.454596 & 1349.992139 \\ 0 & 0 & 1 \end{pmatrix} \tag{10}$$

Estimated distortion coefficients are,

$$k = (k1, k2) = \begin{pmatrix} 0.006403 & -0.01392 \end{pmatrix} \tag{11}$$

The images were undistorted using the estimated parameters and the world corners were reprojected onto the image Fig 3.

The mean squared error for all the corners (across all the images) is found to be 0.3263.

Alternatively, the norm of squared error for all the corners (across all the images) is found to be $\mathbf{0.5712}$ pixels. Error is under a pixel!

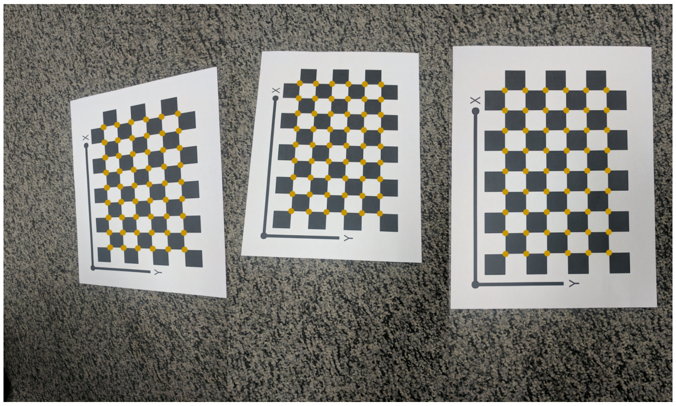

Fig. 3: Reprojection of world corners into undistorted images

Additionally, the obtained estimates show good agreement when cross verified against the MATLAB's camera calibration application's output.

The distortion coefficients were quite small as expected too. It is likely because the camera was far from the object compared to the dimension of the lens. Additionally, Pixel phone might be doing software corrections to undistort the images already.

III. Conclusion

In conclusion, this work has successfully achieved automatic calibration of camera intrinsics and distortion parameters. Through meticulous calibration procedures and advanced algorithms, we have attained highly accurate results, as evidenced by the quality of the reprojection image. The precision of our calibration process underscores its effectiveness in accurately modeling the behavior of the camera system. These calibrated parameters will serve as a crucial foundation for various computer vision tasks, enabling robust and reliable analysis of images captured by the camera.