Auto Calibration, Homework -1 Using 5 late days

Khizar Mohammed Amjed Mohamed
Robotics Engineering Department
Worcester Polytechnic Institute
Worcester, USA
kamjedmohamed@wpi.edu

***Abstract*—This paper implements the work of (1). We automatically estimate the parameters of the camera like the focal length, distortion coefficients and principle points, thereby calibrating our camera. We use a checker board pattern as our calibration target**

I. Introduction

Camera calibration is an important step in 3D computer vision. It enables us to extract real world coordinates from 2D images. The work done this in field can be broadly classified into two into two categories: Photogrammetric calibration and self calibration. Photogrammetric calibration involves using an object whose 3D geometry is well known. The object should also consist of orthogonal planes, like a cube. This technique usually requires an elaborate setup and can be very expensive. Self-calibration involves using multiple images in a static setting and using the correspondence between theses images to calibrate cameras. This technique is more flexible than the former one.

Authors in (1) propose a technique that is between Photogrammetric and self-calibration. Their approach involves using images of a known planar pattern like a checkerboard at different orientations. We are going to implement this proposed method in this paper. In order to understand the process, we will first go over some basic about the pin hole camera model in the next section.

II. Pinhole camera model

Figure 1 is a pin hole camera model where a 3-D object is seen as a 2D image on the image plane of a camera.. This image plane is nothing but the sensor of our camera. For ease of understanding we always refer to the virtual image plane as our sensor instead of image plane.This is done because the image is no longer inverted.

Figure 2 shows the three coordinate frames in the Pin hole camera model. The world frame, camera frame, and the image frame respectively. As shown in *Figure 3* the extrinsic parameters help in converting the world coordinates to camera coordinates. Similarly the intrinsic parameters convert the camera coordinates to the pixel coordinates.

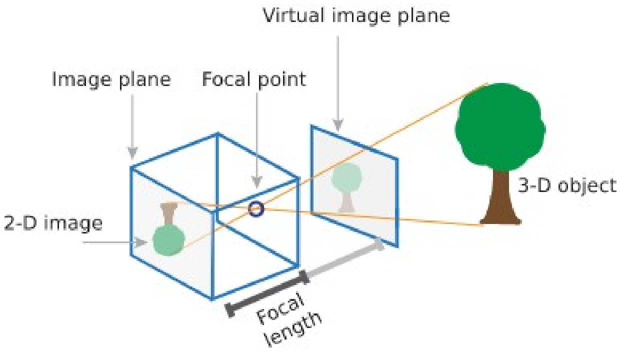

Fig. 1: Pin Hole camera model

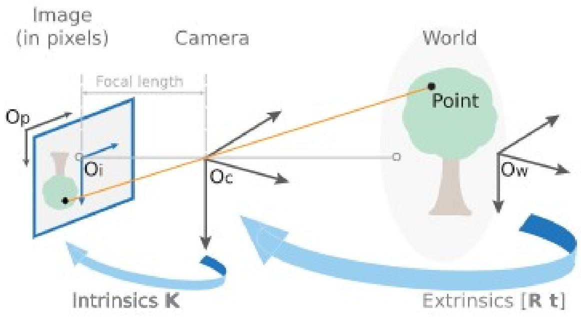

Fig. 2: Various coordinate systems

III. Camera matrix

The camera matrix is made up intrinsic and extrinsic camera matrices. It helps converting from the world coordinates to

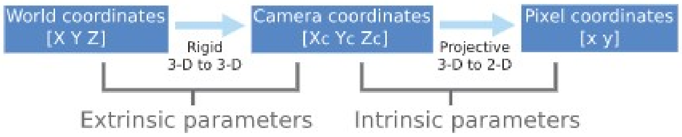

Fig. 3: Various coordinate systems

pixel coordinates. Consider the following equation.

$$s\begin{bmatrix}u\\v\\1\end{bmatrix} = H\begin{bmatrix}X\\Y\\1\end{bmatrix}$$

Here s is an arbitrary scale factor.The terms u and v represent the pixel coordinates. X,Y represents the world coordinates and H is the camera matrix. H is broken down as

$$H = A\begin{bmatrix}r_1 & r_2 & t\end{bmatrix}$$

Here A represents the intrinsic camera matrix. The terms r_1 and r_2 are the first two columns of the rotation matrix between the world frame and the camera frame. And t is the translation column. The A matrix is written as follows

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

α, β, γ,u_0 and v_0 represent the focal length in x, focal length in y, the distortion coefficient, and the principal points respectively.

As discussed above finding the Camera matrix H involves finding the Intrinsic and extrinsic camera parameters.

A. Intrinsinc camera matrix

The first step here is to calculate the Homography between the World coordinates(X,Y,Z) and the camera coordinates X_c,Y_c (refer to Figure3). This can be achieved by selecting common features in the world coordinates and the camera coordinates. In the assignment we have chosen the following world coordinates(green) shown in Figure 4. In order to find the corresponding selected points we uses cv2.checkerboard function. This gives us the corner points(green) in the image as shown in Figure 5.This process is repeated for all 13 pictures provided to us.

We refer to the process used by (1) in order to calculate the camera matrix. We will not go into detail about the logic of the process, but will only outline the steps. Using the previously calculated homography matrix we compute three vectors v_{ij} of the following form.

$$v_{ij} = \begin{bmatrix}h_{i1}h_{j1}\\h_{i1}h_{j2}+h_{i2}h_{j1}\\h_{i2}h_{j2}\\h_{i3}h_{j1}+h_{i1}h_{j3}\\h_{i3}h_{j2}+h_{i2}h_{j3}\\h_{i3}h_{j3}\end{bmatrix}$$

Here h_{ij} represents the i^{th} row and j^{th} column of the homography matrix. We then calculate the following.

$$V = \begin{bmatrix}v_{12}\\(v_{11}-v_{22})^T\end{bmatrix}$$

We then solve for b in

$$V * b = 0$$

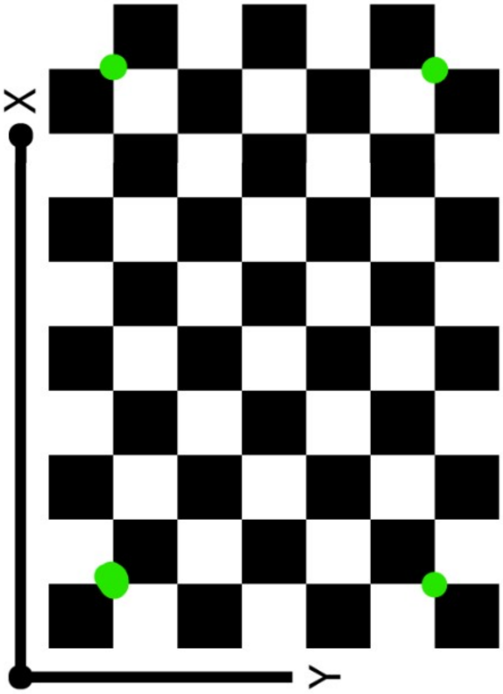

Fig. 4: World frame of checkerboard

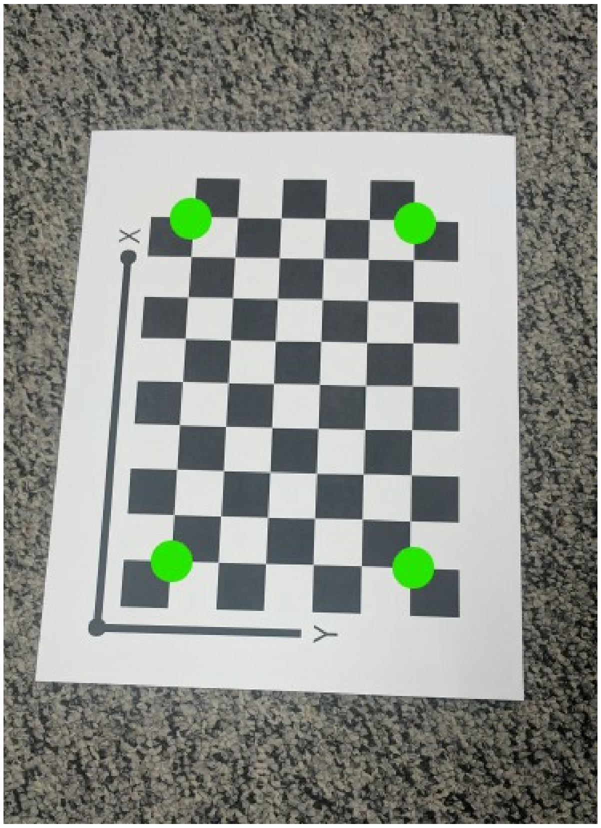

Fig. 5: Image frame of checkerboard

b is of the form

$$b = \begin{bmatrix} B_0 & B_1 & B_2 & B_3 & B_4 & B_5 \end{bmatrix}$$

We the use the following equations to calculate the various parameters

$$w = B_0B_2B_5 - B_1^2B_5 - B_0B_4^2 + 2B_1B_3B_4 - B_2B_3^2$$

$$d = B_0B_2 - B_1^2$$

$$\alpha = \sqrt{w/(d * B_0}$$

$$\beta = \sqrt{w/(d^2 * B_0}$$

$$\gamma = \sqrt{w/(d^2 * B_0}.B_0$$

$$u_0 = (B_1B_4 - B_2B_3)/d$$

$$u_0 = (B_1B_3 - B_0B_4)/d$$

B. Extrinsic parameters

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

$$r_1 = \lambda A^{-1}h_1$$

$$r_2 = \lambda A^{-1}h_2$$

$$r_3 = r_1 \times r_2$$

$$t = \lambda A^{-1}h_3$$

C. Maximum Likelihood estimation

Using the above mentioned equations we get a rough estimate about our intrinsic and extrinsic parameters. We then use maximum likelihood estimation to optimize our parameters. For this we use scipy.optimize.least-squares method.The following equation is used to optimize our parameters.

$$\sum_{i=1}^{n}\sum_{j=1}^{m} ||m_{ij} - \hat{m}(A, R_i, gt_i, M_j)||^2$$

Here m_{ij} represents the actual pixel coordinates in the image. And $\hat{m}$ represents the estimated pixel coordinates of the corners using the above mentioned optimization.M represents the world coordinates of a given corner. It is also important to consider the effects of radial distortion in our camera. Therefore we also consider the distortion coefficients k1 and k2 in our optimization. K1 and K2 are set to zero initially.The formula that takes the distortion coefficient is given as follows. Here u, v are the estimated pixel coordinates. $\hat{u}$ and $\hat{v}$ are the pixel coordinates adjusted for radial distortion.

$$\hat{u} = u + (u - u_0) * (k1 * (x^2 + y^2) + k2 * (x^2 + y^2)^2)$$

$$\hat{v} = v + (v - v_0) * (k1 * (x^2 + y^2) + k2 * (x^2 + y^2)^2)$$

IV. Results

A. Intrinsic matrix

Using the above mentioned method, the calculated Camera Intrinsic matrix is found to be the following. We have calculated the re-projection error obtained when just the intrinsic matrix is used. This is done by calculating the difference in actual pixel coordinates of corners and the estimated pixel coordinates of corners. The results are as follows. It is important to note that the distortion coefficients are not included here. Figure 6 shows the the plotted corners. The red ones are the actual corners obtained through cv2.checkerboard. The blue ones are the estimated corners. There is hardly any difference between both.

$$A = \begin{bmatrix} 2034.75 & 0.493 & 772.7 \\ 0.0 & 2017.90 & 1360.90 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

$$k_1 = k_2 = 0$$

$$Error = 1.08903$$

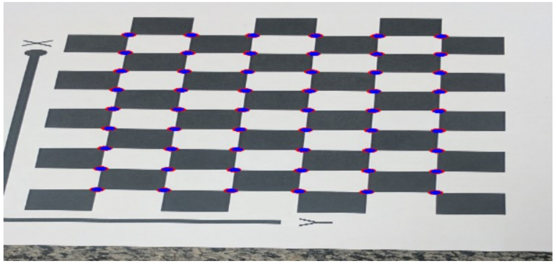

Fig. 6: Estimated corners using rough estimate of A

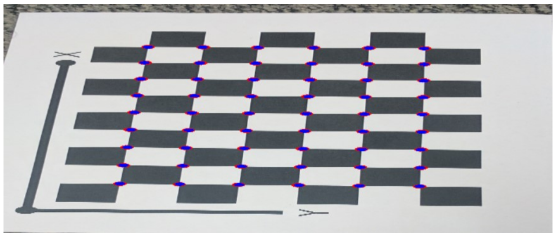

Fig. 7: Estimated corners using optimized A

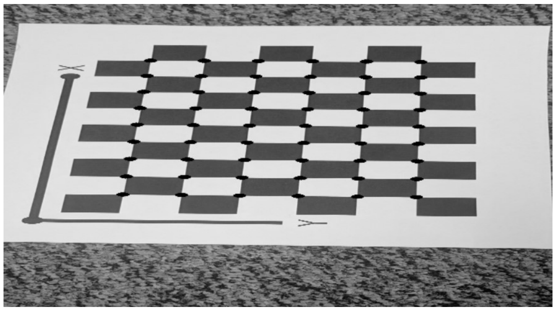

Fig. 8: Rectified image with re-projected corners

B. Optimized Intrinsic matrix

Similarly the error for the optimized A matrix and using the distortion coefficients is shown. Figure 7 shows the respective results as discussed in the previous subsection

$$A_{optimized} = \begin{bmatrix} 2026.95 & 1.4288 & 772.5 \\ 0.0 & 2009.83 & 1360.45 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

$$k_1 = 0.090$$

$$k_2 = -0.4275$$

$$Error = 0.9334$$

C. Rectification

Now that we have calculated the optimized intrinsic matrix and the distortion coefficients we can undistort our images based on the same. To make sure that our undistortion is correct we can plot our re-projected corners to see if they are still in the right spots. Figure 9 is an example of the an undistorted image with the re-projected corners. It is important to note that there is not really a big difference between the distorted and undistorted images.

References

[1] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.