HW 1: AutoCalib

Aadhya Puttur

I. Introduction

The goal of this project was to map 3D world coordinates to image coordinates and estimate parameters of the camera. Camera calibration is known as estimating parameters of the camera like the focal length, distortion coefficients and principle point.The camera calibration is given as follows in figure [1].

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Fig. 1. Camera Calibration Matrix K

The radial distortion parameters are denoted as k_1 and k_2. Our task is to estimate f_x, f_y, c_x, c_y, k_1, k_2.The method that is used is one from Zhengyou Zhang of Microsoft. In Zhang's paper, he relies on a checkerboard as a calibration target. We have a set of n images taken of the checkerboard from different angles and one reference of the image of the checkerboard shown in figure [2]. Following this method we

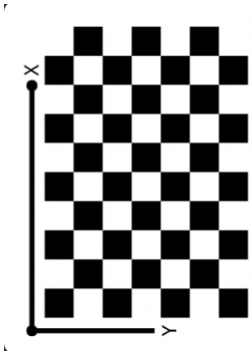


Fig. 2. Checkerboard reference used, each square 21.5mm

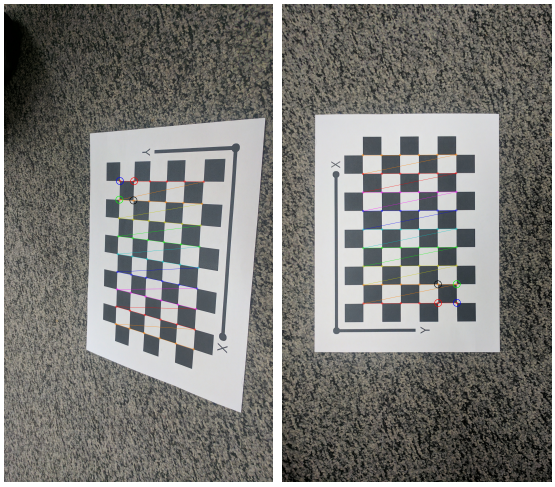
are able to properly reproject the corners on rectified image. This project consists of 3 steps.

1) Calculate the intrinsic parameters
2) Calculate the extrinsic parameters
3) Non-linear Geometric Error Minimization

II. HW1: AutoCalib

1) Estimate Intrinsic Parameters

The first thing we want to do is find the corresponding image points of each of the n images to the world points. We can do this because we know the scale of each of the squares on the checkerboard as each square is 21.5mm. With this scale we can make a grid of all the coordinates and have it represent our world coordinates. Our 2D image points will be taken from those set of n images of the different orientations of the checkerboard. We will use the function cv2.findChessboardCorners() to find the corners in the chessboard of the 2D image and map them to the corresponding 3D world coordinate to get the homography matrix.



(a) (b)

Fig. 3. Marked corners from corner detection and corners that will be used for homography

After we find the homography of the 2D image and the world coordinates we want to start solving for $\mathcal{B}$ shown in figure [4]. We can use the homographies to solve for v_ij shown in figure [4].

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

$$v_{ij} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, h_{i3}h_{j1} + h_{i1}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]$$

Fig. 4. Camera Calibration

Then we can find b from the equation $Vb = 0$ for n images. Using b we can get the intrinsic matrix with the equations below in figure [5]. From these equations we can get the values for the K intrinsic matrix from figure [1] to figure [6].

A. Puttur is with the Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA 01609, USA (e-mail: aputtur@wpi.edu)

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

Fig. 5. Equations

Then we get the intrinsic matrix from figure [1] below figure [5].

$$
K = \begin{bmatrix} 2064 & 0 & 794 \\ 0 & 2058 & 1346 \\ 0 & 0 & 1 \end{bmatrix}
$$

Fig. 6. Camera Calibration Matrix K

2) Estimate Extrinsic Parameters

Now that we found the intrinsic matrix, we can find the extrinsic matrix, using the intrinsic matrix and the homographies. We calculate r_1, r_2, r_3, and t as follows in the equations below figure [7].

$$
\begin{aligned}
r_1 &= \lambda A^{-1} h_1 \\
r_2 &= \lambda A^{-1} h_2 \\
r_3 &= r_1 x r_2 \\
t &= \lambda A^{-1} h_3
\end{aligned}
$$

Fig. 7.

Once I calculated the extrinsic parameters I wanted to get the x, y coordinates of just rotation and translation applied from the World coordinates and the distorted x, y coordinates of the intrinsic matrix applied from figure [8].

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

Fig. 8. Calculate the distorted u, v and x and y coordinates

3) Non-linear Geometric Error Minimization In order to estimate k we use Estimating Radial Distortion by Alternation which is the figure [9] We use the x, y, u, v and some of the values from the intrinsic matrix to to the equation. We then find k_1 and k_2 by first initializing them to 0. We iterate this about 100 times in order to get a

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

Fig. 9.

minimization loss of the projected distorted coordinates of the image and the ideal image pixels. We would then use $scipy.optimize.least_squares$ to get the the loss.

REFERENCES