Homework 1
Project Report
Camera Calibration

Fnu Karan
NetID: kx361

March 3, 2017

Date Performed: February 19, 2017
Instructor: Professor Gerig

1 Objective

The objective is to calibrate a camera for a fixed focal length using two orthogonal checkerboard planes, and to find intrinsic and extrinsic parameters.

2 Experimental Procedure

2.1 Data Capture

Two checkerboards are places on a wall corner which is at 90° as shown in the figure below. The world axes are chosen as: The origin is at the lower end where the two checkerboards meet each other. Y axis points upwards from the origin, X axis is towards the right checkboard from the origin and Z axis is towards the left checkerboard from the origin.

The points to be measured are shown with the labels, A to F in the Figure 1. Total six points are chosen, three on one checkerboard and three on the other. All the world frame coordinates are measured in millimeters(mm). Also the distance reorded from camera origin to the world origin is 677 mm.

The picture of the checkerboard pattern is captured using a camera and the coordiantes of these 6 points are measured in the image coordinate frame (with the origin taken at the top left corner of the image). The image resolution is 4032x3024. This step gives us the coordinates of 6 points with their corresponding points in the image. We need to estimate 11 free parameters for the camera calibration so this number of points is sufficient.

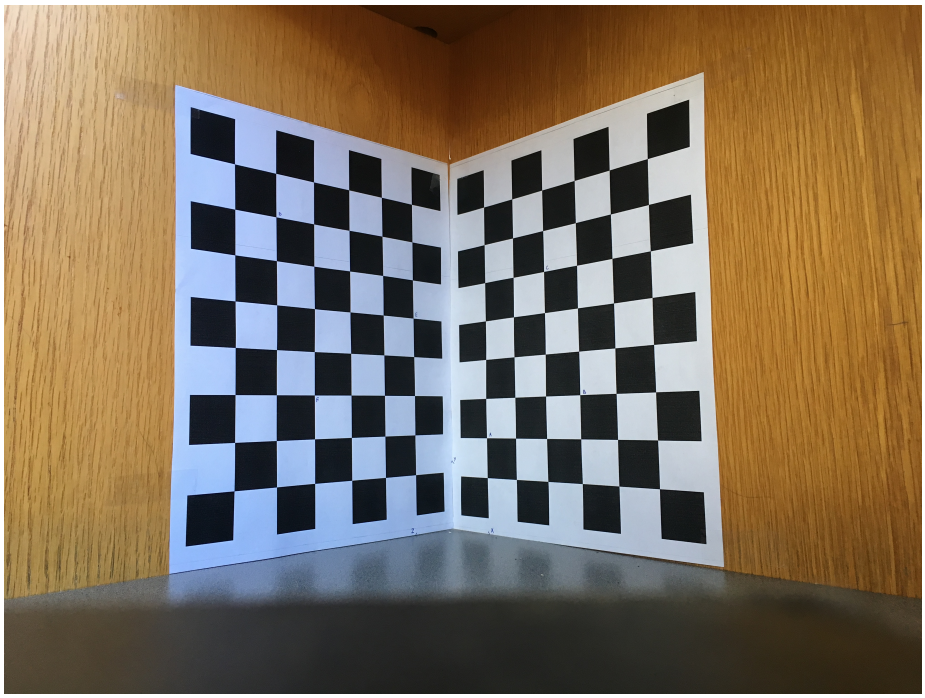

Figure 1: Checkerboard pattern on the corner of the wall and the world frame coordinate axes.

The camera used was of the phone iPhone 6S. It is 12MP and we can see it in the picture resolution as specified above, and it uses "Stacked back-illuminated CMOS image sensor", and its size is 4.80 x 3.60 mm (1/3"). The pixel size is 1.22 μm. The aperture is f/2.2. The focal length is 4.15 mm.

2.2 Estimation of the Calibration Matrix

Least squares method is used to estimate the calibration matrix. there are 12 homogeneous linear equations in 12 variables, which are the coefficients of the calibration matrix $\mathcal{M}$. Let's denote this system of linear equations as

$$\mathcal{P}m = 0 \tag{1}$$

$$m := \begin{bmatrix} m_1 & m_2 & m_3 \end{bmatrix}^T, \tag{2}$$

where m_1, m_2, m_3 are fist, second and third rows of the matrix $\mathcal{M}$ respectively. m is a 12 x 1 vector, and $\mathcal{P}$ is a 12 x 12 matrix. The problem of least square estimation of $\mathcal{P}$ is defined as

Parameters	Values
θ	89.93°
u_0	1903.371
v_0	1566.708
α	3567.708
β	3550.947

Table 1: Intrinsic Parameters

$$min||\mathcal{P}m||^2, \text{ subject to } ||m||^2 = 1. \tag{3}$$

As it turns out, the solution of above problem is given by the eigenvalue of matrix $\mathcal{P}^T\mathcal{P}$ having the least eigenvalue. The eigenvectors of the matrix $\mathcal{P}^T\mathcal{P}$ can also be computed by performing the singular value decomposition (SVD) of $\mathcal{P}$. The 12 right singluar vectors of $\mathcal{P}$ are also the eigenvectors of $\mathcal{P}^T\mathcal{P}$. The SVD method is used here to get the eigenvector corresponding to the least eigenvalue. This eigenvector is the solution to the above problem. Reorganizing the 12 x 1 vector m in a matrix of 3 x 4 gives us the matrix $\mathcal{M}$. The first three elements of the third row of this matrixx denote one of the three rotation vectors.

2.3 Computation of Intrinsic and Extrinsic Parameters

The intrinsic and extrinsic parameters are computed from the calibration matrix $\mathcal{M}$ by using the method given in the book. These methods use the various properties fo the rotation vectors, namely the norm of them being one and the dot product of the two rotation matrices being zero.

Once the calibration matrix is estimated, we pick these points again from the checkerboard pattern, and record their coordinates in the world frame. We then transform these points to the image coordinates using the matrix $\mathcal{M}$ that we obtained using the above described method. These image coordinates are then compared to the observed coordinates and finally we give the errors at last.

3 Results

The estimated calibtation matrix using the above described method obtained is,

$$\mathcal{M} = \begin{bmatrix} 0.00289166 & -0.00016395 & -0.00094185 & -0.94420018 \\ 0.00056440 & -0.00280669 & 0.00065426 & -0.32934464 \\ 0.00000048 & -0.00000006 & 0.00000050 & -0.00045809 \end{bmatrix} \tag{4}$$

The intrinsic parameters are shown in Table 1.
The extrinsic parameters consist of the rotation and translation matrix. The rotation matrix is shown below.

$$\mathcal{R} = \begin{bmatrix} -0.07314050 & -0.99600794 & -0.05117265 \\ 0.71758403 & -0.01692221 & -0.69626632 \\ 0.69262083 & -0.08764595 & 0.71595709 \end{bmatrix} \tag{5}$$

The translation matrix is estimated as $\rho\mathcal{K}^{-1}b$, where $\mathcal{K}$ is the intrinsic parameter matrix and b is the last column of the matrix $\mathcal{M}$. The obtained translation vector is,

$$\mathcal{T} = \begin{bmatrix} -89.1718 & 142.7421 & -649.3070 \end{bmatrix}^T. \tag{6}$$

4 Discussion

4.1 Intrinsic Parameters

We get θ almost equal to 90°, which means that the camera has very little skew, meaning that the X and Y axes in the image frame are almost at 90° to each other.

The image center lies at (2016, 1512). We had (u_0, v_0) as (1903.37, 1566.70). The image center does not coincide with C_0 and its position is shifted by (112.63, 54.7).

α and β are equal to kf and lf repectively. The terms k and l denote the number of pixels per mm, and f is the focal length of the lens. In our case, focal length is 4.15 mm. The pixel density (pixels per mm) is obtained as,

$$k = \frac{4032}{4.80} = 840 \tag{7}$$

and

$$l = \frac{3024}{3.60} = 840 \tag{8}$$

Now, we have pixel densities, k and l. We can find α and β as,

$$\alpha = k * f = 3486 \tag{9}$$

and

$$\beta = l * f = 3486 \tag{10}$$

These values of α and β are comparable to the ones that we estimated in Table 1.

World Coordinates	Image Coordinates (calculated)	Image Coordinates (measured)	Error norm
(37, 68, 0)	(1909.75, 1124.02)	(1910.64, 1123.23)	1.192
(121.5, 96, 0)	(1503.97,1310.25)	(1503.53, 1310.88)	0.763
(93.5, 181, 0)	(1660.77, 1852.17)	(1660.95, 1851.72)	0.479
(0, 212, 150)	(2832.89, 2089.38)	(2833.45, 2089.04)	0.653
(0, 154.7, 37)	(2237.12,1646.71)	(2236.34, 1647.52)	1.130
(0, 98.3, 122)	(2671.01, 1305.22)	(2670.60, 1305.35)	0.432

Table 2: Test of Calibration Parameters: Error between measured and estimated image coordinates. All the world coordinates are in mm.

4.2 Extrinsic Parameters

Three rows of the rotation matrix have their norm equal to 1 and their dot product is almost equal to zero. This is justified very well considering the human error, which cannot be avoided.

The translation vector also approximates the real values. We measured the difference between the world origin and the camera origin, which is approximately equal to the $||\mathcal{T}||$. The straight distance was measured as 667 mm and $||\mathcal{T}|| = 670.76$, and these both values are really close.

4.3 Image Coordinates Reconstruction

To revalidate the estimated parameters, we take those 6 points again, and corresponding to these points we also calculate the image coordinates. Along with that we also measure the true image coordinates using MATLAB. The values are shown in Table 2.

The error norm is calculated by subtracting corresponding coordinates and errors along X and Y axes. These errors are then squared and added, and then square root of the resultant is taken to get the error norm. We note that the reconstruction errors are not so big and can be justified given the amount of inherent errors in the measuring process itself.

5 MATLAB Code

5.1 Important Information about the Code

The code is done using MATLAB. The real world coordinates and image coordinates were saved in an excel file and then using some features of excel, values of matrix $\mathcal{P}$ were calculated. Once the values of matrix $\mathcal{P}$ are obtained, then we copied it to another excel file from where the matrix $\mathcal{P}$ is read into the MATLAB code.

The intrinsic parameters are stored in the variable named *Intrinsic*, rotation matrix is stored in the variable named *Rotation* and translation matrix is stored in the variable named *Translation*.

5.2 The Code

```
%Read the image
image = imread('C:\Users\K-Chak\Google Drive\NYU\Spring 2017\Computer Vision\Assignment\1\Problem 4\check_1.jpg');
imshow(image);

%Measure the pixel coordinates of the 6 points on the image.
[imageX,imageY] = ginput(6);

%Use excel to caculate the P Matrix and then read as well.
P = xlsread('C:\Users\K-Chak\Google Drive\NYU\Spring 2017\Computer Vision\Assignment\1\Problem 4\pmatrix.xlsx');

%Find the values of m1,m2,m3 or what we say the M matrix using the following three statements.
[U,S,V] = svd(P);
[min_value, min_index] = min(diag(S(1:12,1:12)));
m = V(1:12, min_index);

M = [ m(1:4)'; m(5:8)'; m(9:12)'];

%Extract the a1, a2 and a3 from the M matrix
a1 = M(1,1:3)';
a2 = M(2,1:3)';
a3 = M(3,1:3)';

%Calculate rho
rho = 1/norm(a3);
r3 = rho.*a3;

%To find the parameters we need the following values, so we calculate them here.
dot_a1a3 = dot(a1,a3);
dot_a2a3 = dot(a2,a3);
cross_a1a3 = cross(a1,a3);
cross_a2a3 = cross(a2,a3);
```

```
32
33 %Calculate the intrinsic parameters
34 u0 = rho*rho*dot_a1a3;
35 v0 = rho*rho*dot_a2a3;
36 theta =acosd( -(dot(cross_a1a3,cross_a2a3))/((norm(
       cross_a1a3))*norm(cross_a2a3)));
37 alpha = rho*rho*norm(cross_a1a3)*sin(theta);
38 beta = rho*rho*norm(cross_a2a3)*sin(theta);
39
40 %Combine all these intrinsic parameters into 1 single
       vector
41 Intrinsic = [u0;v0;theta;alpha;beta];
42
43 %Calculate the rotation matrix components, r1 and r2. r3
       was caculated earlier
44 r1 = cross_a1a3./norm(cross_a1a3);
45 r2 = cross(r3,r1);
46
47 %Form the rotation matrix using r1,r2 and r3
48 Rotation = [r1';r2';r3'];
49
50 %Extract the translational component from the M matrix,
       that is the last column
51 b = M(1:3,4);
52
53 %Calculate the K matrix to further calculate the
       Translation matrix
54 K =     [
55                 alpha   -alpha*cot(theta)       u0;
56                 0       beta/sin(theta)         v0;
57                 0       0                       1
58         ];
59
60 %Using the K matrix, get the translational matrix
61 Translation = rho*inv(K)*b;
62
63 %These are the world coordinates of the 6 points
64 world = [
65                 37      68      0       1;
66                 121.5   96      0       1;
67                 93.5    181     0       1;
68                 0       212     150     1;
69                 0       154.7   37      1;
70                 0       98.3    122     1;
71         ];
72
```

```
73 %Calculate the pixels using the M matrix for each point
       of the world
74 for(i=1:6)
75         image_reconstructed = M*world(i,:)';
76         imageCalculatedX(i)=image_reconstructed(1)/
               image_reconstructed(3);
77         imageCalculatedY(i)=image_reconstructed(2)/
               image_reconstructed(3);
78 end;
79
80 imageCalculatedX = imageCalculatedX';
81 errorX = imageCalculatedX - imageX;
82
83 imageCalculatedY = imageCalculatedY';
84 errorY = imageCalculatedY - imageY;
85
86 %Calculating the final norm error for each point
87 for i =1:6
88         finalError(i) = norm([errorX(i) errorY(i)]);
89 end;
```