Project 1 Report (Camera Calibration)

Nazmuzzaman Khan

1 Problem Statement

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

2 Theory and Algorithm

Checkerboard patterns are pasted on a wall corner at an angle to each other. The world frame axes are chosen as follows: The origin is at the lower end where two images meet on the corner. Z axis points upward from the origin. X axis is pointed along the left wall from the origin. Y axis is pointed along the right wall and goes through the origin. The origin and the three axes are shown in the following figure.

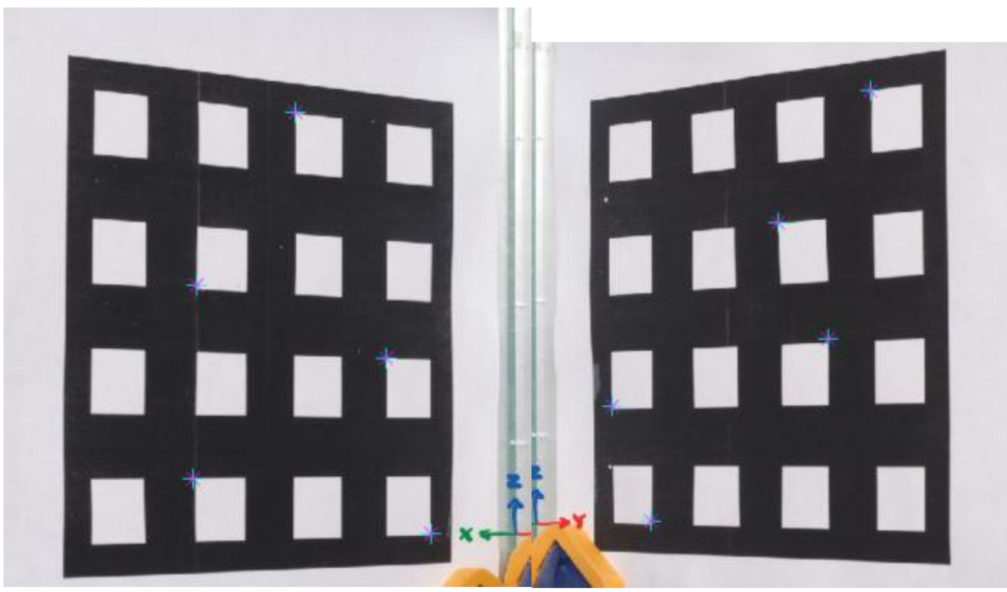

Figure 1: Checkerboard pattern on the wall corner and the world frame coordinate axes. Also showing the points taken for calibration.

Total 10 points are chosen in figure, with 5 on one checkerboard and 5 on the other checkerboard. The real-world coordinates for these 10 points can be measured from image. All world frame coordinates are measured in centimeters. The picture of the checkerboard pattern is captured using a camera and the coordinates of these 10 points are measured in the image coordinate frame using getpoint.m file. We need to estimate 11 free parameters for camera calibration, so this number of points is more than sufficient. World coordinates (xw,yw,zw) are known and image coordinates (xi,yi) are measured for each data point.

- Each data point $\mathbf{P}_w = (x_w, y_w, z_w)$ and its image $\mathbf{P}_I = (x_i, y_i)$ gives us two equations:

$$x_i(a_{31}x_w + a_{32}y_w + a_{33}z_w + a_{34})$$
$$= a_{11}x_w + a_{12}y_w + a_{13}z_w + a_{14}$$
$$y_i(a_{31}x_w + a_{32}y_w + a_{33}z_w + a_{34})$$
$$= a_{21}x_w + a_{22}y_w + a_{23}z_w + a_{24}$$

- We have 12 parameters to estimate in the projection matrix, but there are only 11 degrees of freedom.
- We need to collect at least 6 data points.

Fig 2: Slide 109

Rearranging equations of slide 109

-xi(a31*xwi+a32*ywi+a33*zwi) + (a11*xwi+a12*ywi+a13*zwi+a14) = xi*a34

-yi(a31*xwi+a32*ywi+a33*zwi) + (a21*xwi+a22*ywi+a23*zwi+a14) = yi*a34

setting a34=1

Set the scale factor to 1 arbitrarily and solve for the 11 parameters an equation of form **Ap**=**b** for 11 points. Solving for p, we get projection matrix. K contains intrinsic parameters, R is rotational matrix and t is translation matrix.

From projection matrix, **Π**=**K** [**R t**] = [**KR Kt**] = [**B b**], we have

B=**KR** and **b**=**Kt**

We create, **A**=**BB**^{T}. Following equations are solved to get Intrinsic and Extrinsic values.

Recovering the intrinsics

- If we do not assume $s = 0$,

$$\mathbf{A} = \begin{bmatrix} \underbrace{\alpha^2 + s^2 + u_0^2}_{k_u} & \underbrace{u_0 v_0 + s\beta}_{k_c} & u_0 \\ \underbrace{s\beta + u_0 v_0}_{k_c} & \underbrace{\beta^2 + v_0^2}_{k_v} & v_0 \\ u_0 & v_0 & 1 \end{bmatrix}$$

We can now estimate the intrinsic parameters as follows (in the given order):

$$u_0 = A_{13}$$
$$v_0 = A_{23}$$
$$\beta = \sqrt{k_v - v_0^2}$$
$$s = {(k_c - u_0 v_0)}/{\beta}$$
$$\alpha = \sqrt{k_u - u_0^2 - s^2}$$

Fig 3: Slide 120

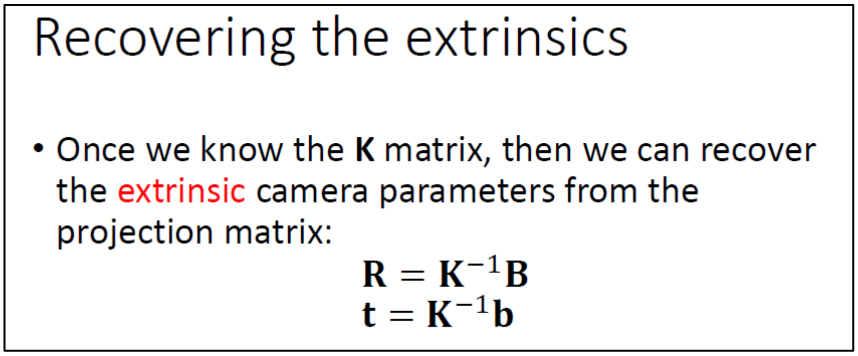

Fig 4: Slide 121

3 Camera Intrinsic and Extrinsic parameters

$u0$, $v0$: optical center coordinates in the image plane.
The estimated calibration matrix using least squares estimation

created_proj_mat =

1.0e+04 *

-0.257190816865057	0.112853313760400	0.004428923552767	7.759690577697220
-0.057800977466988	-0.061608474955022	-0.263347409160427	7.777754698087736
-0.000069109715381	-0.000071946664359	0.000006893818065	0.008253256339005

Intrinsic Matrix (assuming s = 0)

K =

K =

1.0e+03 *

2.636697121982136	0	0.968549685848159
0	2.685460941557550	0.661166424406166
0	0	0.001000000000000

Rotation Matrix

R =

-0.721564195437269	0.692295036094357	-0.008526128199874
-0.045087060732334	-0.052280619373326	-0.997614100638348
-0.691097153812613	-0.719466643591229	0.068938180652502

check for orthonormality of R matrix (ans = R*R^T)
ans =

```
   1.000000000000000   0.004845381141910   0.000000000000000
   0.004845381141910   1.000000000000000   0.000000000000000
   0.000000000000000   0.000000000000000   1.000000000000000
```

Translation Matrix

t =

```
  -0.887467334489680
   8.642757296534201
  82.532563390047997
```

4 Error Calculation

Once the calibration matrix is estimated, we pick 6 different points on the checkerboard pattern and record their coordinates in the world frame. These points are then transformed to image frame using the calibration matrix. These coordinates are then compared with measured coordinates of these points in the image frame. The error norm is calculated by subtracting corresponding coordinates and find 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. Following 6 points are chosen for error calculation.

```
%Points near (0,0,0)
   %point 1                     %2D points from getpoint code(true values)
   xw1_c=4; yw1_c=0;zw1_c=0;      x1_c=843.6;y1_c=946.94;
   %point 2
   xw2_c=6; yw2_c=0;zw2_c=0;      x2_c=792.56;y2_c=946.9;
   %point3
   xw3_c=0; yw3_c=4;zw3_c=0;      x3_c=1032.5;y3_c=943.86;
   %point4
   xw4_c=0; yw4_c=4;zw4_c=2;      x4_c=1034;y4_c=878.8;

   %Points far from (0,0,0)
   %point5
   xw5_c=0; yw5_c=4;zw5_c=10;      x5_c=1027.8;y5_c=607.86;
   %point6
   xw6_c=10; yw6_c=0;zw6_c=10;     x6_c=687.3;y6_c=600.25;
```

Following error values were obtained:

1. point 1 x-coordinate error 0.025026% and point 1 y-coordinate error 0.092893%. Normalized point 1 error 0.096205%
2. point 2 x-coordinate error 0.064167% and point 2 y-coordinate error 0.115601%. Normalized point 2 error 0.132216%
3. point 3 x-coordinate error 0.161029% and point 3 y-coordinate error 0.173345%. Normalized point 3 error 0.236599%
4. point 4 x-coordinate error 0.370768% and point 4 y-coordinate error 0.107604%. Normalized point 4 error 0.386066%
5. point 5 x-coordinate error 0.028716% and point 5 y-coordinate error 0.287718%. Normalized point 5 error 0.289147%

6. point 6 x-coordinate error 0.243697% and point 6 y-coordinate error 0.312087%. Normalized point 6 error 0.395963%

Following **error statistics** were calculated:
Error: Maximum 0.395963%, Minimum 0.096205%, Mean 0.256033, Median 0.262873, Std Dev 0.125570.
As all the points generate some amount of error, they come from manual handling (getting point location using ‘getpoint.m). Also, maximum error occurred at the point furthest from (0,0,0) and minimum error occurred at the point closest to (0,0,0). This indicated that, distortion may have some effects error. But overall, all errors are within reason.

To check if error is solely dependent on manual handling error, a **new set of points** were used for estimation.

```
%Points near (0,0,0)
%point 1                    %2D points from getpoint code(true values)
xw1_c=0; yw1_c=12;zw1_c=2;      x1_c=1235.2;y1_c=880.4;
%point 2
xw2_c=8; yw2_c=0;zw2_c=2;      x2_c=739.9;y2_c=881.9;

%Points far (0,0,0)
%point3
xw3_c=0; yw3_c=16;zw3_c=8;      x3_c=1340.5;y3_c=654.4;
%point4
xw4_c=18; yw4_c=0;zw4_c=14;      x4_c=456.7;y4_c=433.1;
```

```
point 1 x-coordinate error 0.243217% and point 1 y-coordinate error
0.098880%. Normalized point 1 error 0.262549%
point 2 x-coordinate error 0.057997% and point 2 y-coordinate error
0.212713%. Normalized point 2 error 0.220478%
point 3 x-coordinate error 0.067411% and point 3 y-coordinate error
0.032639%. Normalized point 3 error 0.074897%
point 4 x-coordinate error 1.631809% and point 4 y-coordinate error
0.878838%. Normalized point 4 error 1.853417%
```

Error stat: Maximum 1.853417%, Minimum 0.074897%, Mean 0.602835, Median 0.241514, Std Dev 0.837589
Maximum error occurred at furthest point, but minimum error occurred at 2nd furthest point from (0,0,0). So manual handling of point coordinates turned out to be the biggest source of error.

The camera coordinate system might also be skewed, due to some manufacturing error, so the angle θ between the two image axes is not equal to (but of course not very different from) 90 degrees. We have completed the same calculations **with skew s≠0.** Following K matrix is obtained.
K =

1.0e+03 *

2.636666169984183	0.012775802511780	0.968549685848159
0	2.685460941557550	0.661166424406166
0	0	0.001000000000000

But adding skew solved the orthonormality anomaly in R matrix.
R =

```
  -0.721354199368538   0.692556485492348  -0.003692350993980
  -0.045087060732334  -0.052280619373326  -0.997614100638348
  -0.691097153812613  -0.719466643591229   0.068938180652502
```

check orthonormality

```
 R*R^T
ans =

   1.000000000000000   0.000000000000000   0.000000000000000
   0.000000000000000   1.000000000000000   0.000000000000000
   0.000000000000000   0.000000000000000   1.000000000000000
```

But it has trivial effect on projection. Following errors were calculated for $s \neq 0$ case. They are same as s = 0 case.

1. point 1 x-coordinate error 0.025026% and point 1 y-coordinate error 0.092893%. Normalized point 1 error 0.096205%
2. point 2 x-coordinate error 0.064167% and point 2 y-coordinate error 0.115601%. Normalized point 2 error 0.132216%
3. point 3 x-coordinate error 0.161029% and point 3 y-coordinate error 0.173345%. Normalized point 3 error 0.236599%
4. point 4 x-coordinate error 0.370768% and point 4 y-coordinate error 0.107604%. Normalized point 4 error 0.386066%
5. point 5 x-coordinate error 0.028716% and point 5 y-coordinate error 0.287718%. Normalized point 5 error 0.289147%
6. point 6 x-coordinate error 0.243697% and point 6 y-coordinate error 0.312087%. Normalized point 6 error 0.395963%

4 Discussions

The image resolution is 1920 x 1280. This means that the center of the image lies at (960; 640). We obtain u0 and v0 equal to 968.5 and 661.1 respectively. The image center does not coincide with the principle point C0 and is offset by (8.5, 21.1).
$\alpha = su.f$ and $\beta = sv.f$ for scale factors su and sv and focal length f. The terms su and sv denote the number of pixels per centimeter, and f is the distance of physical image frame from the pinhole or equivalent lens. The magnitude of α and β are reasonable from this point of view.
For one set of test point, maximum error occurred at the point furthest from (0,0,0) and minimum error occurred at the point closest to (0,0,0). This indicated that, distortion may have some effects error. For 2nd set of test points, maximum error occurred at furthest point, but minimum error occurred at 2nd furthest point from (0,0,0). So manual handling of point coordinates turned out to be the biggest source of error.