Assignment 1: Camera Calibration Project Report

Zebin Xu

zebinxu@nyu.edu

February 22, 2017

1. Goal

The goal is to calibrate a (digital) camera with two orthogonal checkerboard patterns and find the intrinsic and extrinsic parameters so as to be able to capture images of objects from known locations and with a known camera model.

2. Environment

Coded in MATLAB (R2016a).
The picture is taken using the rear camera of iPhone 6s with a focal length of 4.15 mm. Its Sensor has 12 megapixels, with the size of 1/3 inches, and the pixel size is 1.22μm. The image taken has a resolution of 4032×3024 $(width \times height)$.

3. Data Capturing

3.1 World Coordinate System Setup

Glue two orthogonal checkerboard patterns onto the corner of the wall. Define a world coordinate system with its origin at the bottom (see Figure 1), x axis rightward, z axis leftward, and y axis upward (i.e. a right-handed orthonormal coordinate system). The circles are the points to be measured.

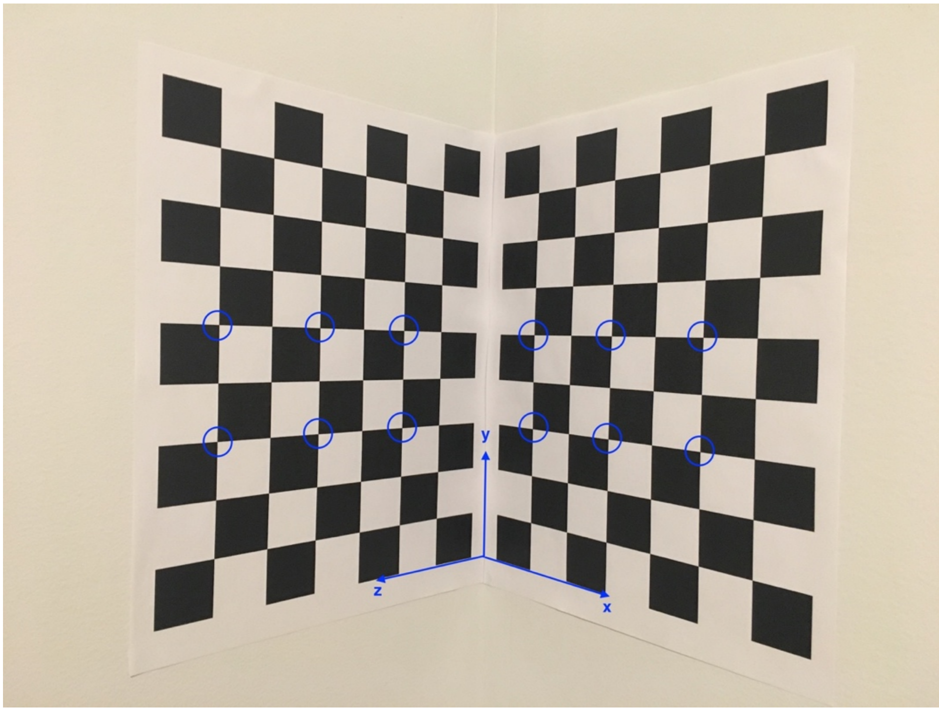


Figure 1. World coordinate system setup

3.2 Picture Capturing

Take a picture of the two orthogonal checkerboard planes that can almost fill the whole screen using the rear camera of iPhone 6s.

4. Data Measurement

4.1 Distance measurement

Measure the horizontal distance between the optical center of the camera and the checkerboard pattern. We got roughly 50 cm.

4.2 Image Coordinates Measurement

Write a script (collect_image_points.m) to measure 12 points on the image with mouse clicking and save these points into a file (image_coordinates_12.txt). The image space points are $\boldsymbol{p_i} = (\boldsymbol{u_i}, \boldsymbol{v_i})^{\boldsymbol{T}}$, where i = 1, 1, … 12. The unit is pixel. The saved file contains a 12x2 matrix, where each row represent a point $p_i = (u_i, v_i)^T$ in image space.

4.3 World Coordinates Measurement

To better measure world coordinates, we need to first measure the size of a square and the starting offset in x and z direction. We get square_size = 2.8 cm, offset = 1.1 cm. Figure 2 shows how this is measured in green lines.

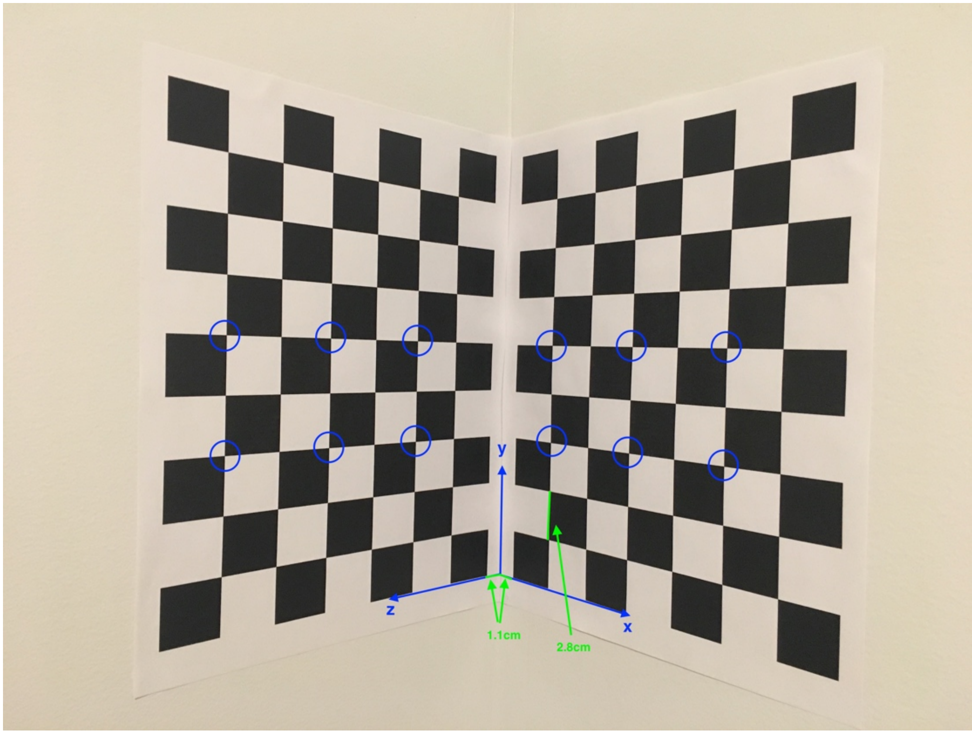


Figure 2. Measurement of square size and offset for world coordinates calculation

According to the 12 picked points, calculate their world space coordinates by counting the squares from the origin, multiplying square size, and adding the offset for x and z coordinates. Save these world coordinates into a file (world_coordinates_12.txt). These world space points are $\boldsymbol{P_i} = (\boldsymbol{x_i}, \boldsymbol{y_i}, \boldsymbol{z_i})^{\boldsymbol{T}}$, where i = 1, 1, … 12. Similarly, the saved file contains a 12x3 matrix.

5. Calibration Matrix Estimation

Our goal is to compute the calibration matrix $\boldsymbol{M}$ that relates the world space points and to the corresponding image space points. The equation $\vec{\boldsymbol{p}} = \frac{\mathbf{1}}{\boldsymbol{z}}\boldsymbol{M}\vec{\boldsymbol{P}}$ can be solved equivalently with

$$\begin{cases}(m_1 - u_i m_3) \cdot \vec{P}_i = 0 \\ (m_2 - v_i m_3) \cdot \vec{P}_i = 0\end{cases}$$

which in turn can be equivalently expressed in a matrix vector multiplication form:

$$Qm = 0$$

$$\begin{bmatrix} P_{1x} & P_{1y} & P_{1z} & 1 & 0 & 0 & 0 & 0 & -u_1P_{1x} & -u_1P_{1y} & -u_1P_{1z} & -u_1 \\ 0 & 0 & 0 & 0 & P_{1x} & P_{1y} & P_{1z} & 1 & -v_1P_{1x} & -v_1P_{1y} & -v_1P_{1z} & -v_1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ P_{nx} & P_{ny} & P_{nz} & 1 & 0 & 0 & 0 & 0 & -u_nP_{nx} & -u_nP_{ny} & -u_nP_{nz} & -u_n \\ 0 & 0 & 0 & 0 & P_{nx} & P_{ny} & P_{nz} & 1 & -v_nP_{nx} & -v_nP_{ny} & -v_nP_{nz} & -v_n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where the 3x4 matrix M is flattened into a 12-dimensional unit vector **m**.
Due to the over-constrained issue, we cannot solve the equation directly but need to minimize $|\boldsymbol{Qm}|^{\mathbf{2}}$. *Least square estimation (LSE)* is to compute $\boldsymbol{min}|\boldsymbol{Qm}|^{\mathbf{2}}$, subject to $|\boldsymbol{m}|^{\mathbf{2}} = \mathbf{1}$.

When the number of dataset (collected points) n >= 6, homogenous *least-squares* can be used to compute **m** that minimizes $|\boldsymbol{Qm}|^{\mathbf{2}}$as the eigenvector of matrix $\boldsymbol{Q^TQ}$ having the smallest eigenvalue. The eigenvector of matrix $\boldsymbol{Q^TQ}$ can also be computed using *Singular Value Decomposition (SVD)* of **Q**.

Thus, the Q matrix with a set of collected points in world space and image space should be constructed first. Then the SVD of Q is performed. Finally, **m** is given by extracting the right singular vector of the decomposition result with the minimum eigenvalue. Vector **m** is then converted back to the 3x4 matrix form **M**. **M** is the calibration matrix estimated.

6. Intrinsic and Extrinsic Parameters Computation

The calibration matrix **M** has the form:

$$\begin{bmatrix} \alpha r_1^T - \alpha \cot\theta \, r_2^T + u_0 r_3^T & \alpha t_x - \alpha \cot\theta \, t_y + u_0 t_z \\ \dfrac{\beta}{\sin\theta} r_2^T + v_0 r_3^T & \dfrac{\beta}{\sin\theta} t_y + v_0 t_z \\ r_3^T & t_z \end{bmatrix}$$

This is shown as a 3x3 and a 3x1 components. This can be written as $\boldsymbol{M} = \boldsymbol{\rho}(\boldsymbol{A}\ \ \boldsymbol{b})$, with $\boldsymbol{a_1^T}, \boldsymbol{a_2^T}, \boldsymbol{a_3^T}$ denoting the three rows of **A**. $\boldsymbol{\rho}$ is a scale factor, which is used to scale the recovered calibration matrix **M** to be a canonical form.

Thus we have:

$$\rho(A\ \ b) = K(R\ \ t) \Leftrightarrow \rho \begin{pmatrix} a_1^T \\ a_2^T \\ a_3^T \end{pmatrix} = \begin{pmatrix} \alpha r_1^T - \alpha \cot\theta \, r_2^T + u_0 r_3^T \\ \dfrac{\beta}{\sin\theta} r_2^T + v_0 r_3^T \\ r_3^T \end{pmatrix}$$

Given the fact that the the rows (or columns) of a rotation matrix forms a right-handed orthonormal coordinate system, it can be inferred that its rows must have unit length and are perpendicular to each other. Thus we have:

$$\begin{cases} \rho = \varepsilon / |a_3| \\ r_3 = \rho a_3 \\ u_0 = \rho^2 (a_1 \cdot a_3) \\ v_0 = \rho^2 (a_2 \cdot a_3) \end{cases}$$

where $\varepsilon = \mp 1$.

The method in the book uses the properties of rotation matrix and the sign of θ to determine all the other of the 11 parameters.

$$\begin{cases} \cos\theta = -\dfrac{(a_1 \times a_3) \cdot (a_2 \times a_3)}{|a_1 \times a_3||a_2 \times a_3|} \\ \alpha = \rho^2 |a_1 \times a_3| \sin\theta \\ \beta = \rho^2 |a_2 \times a_3| \sin\theta \\ r_1 = \dfrac{(a_2 \times a_3)}{|a_2 \times a_3|} \\ r_2 = r_3 \times r_1 \end{cases}$$

Now the K matrix containing the intrinsic parameters can be recovered.

The translation parameters $t = (t_1, t_2, t_3)^T$ can be computed using the equation:

$$Kt = \rho b$$

7. Results

The calibration matrix using LSE is as follows (already multiplied by ρ):

$$M = \begin{pmatrix} -791.1008 & -5.1204 & 3826.3883 & -103076.9951 \\ 1012.4161 & 3401.9912 & 1035.9977 & -117690.1704 \\ 0.7411 & 0.0330 & 0.6706 & -49.8807 \end{pmatrix}$$

The intrinsic parameters are given in the table 1.

Table 1. Intrinsic parameters

Parameter	Value
θ	1.5764 $\approx$ 90.32°
u_0	1979.31
v_0	1557.15
α	3368.84
β	3353.62

The extrinsic parameters are given below:

$$R = \begin{pmatrix} 0.6700 & 0.0265 & -0.7419 \\ -0.0422 & 0.9991 & -0.0024 \\ 0.7411 & 0.0330 & 0.6706 \end{pmatrix}$$

$$R_x = 2.8139^{\circ}, R_y = -47.8278^{\circ}, R_z = -3.6066^{\circ}$$

$$t = (-1.2232 \quad -11.9328 \quad -49.8807)^T$$

8. Discussion

8.1 Intrinsic Parameters

From table 1, it can be known that the x and y axis in the image space are nearly 90° to each other, so the image frame in the camera is hardly skewed; From section 2 we have known the image's resolution is 4032×3024, whose center is at $(2016, 1512)$, which is close to the offset ($(u_0, v_0) = (1979.31, 1557.15)$) to the origin of the image plane. Since our camera's focal length is 4.15mm, we can compute the pixel density using $k = \frac{\alpha}{f}, l = \frac{\beta}{f}$, thus $k = 811.77$ pixel/mm, $l = 808.10$ pixel/mm. The sensor size is $\frac{1}{k}$ or $\frac{1}{l}$, which is approximately 1.22μm, conforming with the specification of the sensor mentioned in section 2.

8.2 Extrinsic parameters

From the extrinsic parameters, we can estimate rotation and translation from the world origin to the camera frame.

Our camera has a significant rotation on y axis and minor rotations on x and z axes, which is reasonable based on the world frame we set. We also verify the rotation matrix R by checking the norms of the rows or columns that equal 1.

We also know that the world frame is translated into the camera frame by the translation vector $t = (-1.2232 \quad -11.9328 \quad -49.8807)^T$, which can be interpreted as the position of the world origin in camera frame. Since z axis of the camera coordinate faces inward, the world origin is on the negative z axis, thus $t_z = -49.8807$ accords with our measurement of the distance between world origin and camera origin as 50 cm in section 3. The world origin is slightly lower than the camera, so $t_y = -11.9328$ also conforms with our expectation.

However, sometimes it is more natural to specify the camera pose rather than how world points are transformed to camera frame. Inversely, we can view in the world frame. By taking the inverse of $\begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$, we extract the translation vector $t_c = (37.2838 \quad 13.5986 \quad 32.5111)^T$ which is the camera's position in the world frame. Because the world frame we set has its x and z axis horizontal, we can roughly calculate the distance between camera and checkerboard with $\sqrt{t_{cx}{}^2 + t_{cz}{}^2}$, which is almost the same as $|t_z| = 49.8807$ that we computed in the camera frame.

9. Image Coordinates Reconstruction

A script (world_coordinates_generator.m) is written to compute 96 points on the checkerboard in the world frame and save these world coordinates in a file (world_coordinates_96.txt).
Then 96 points are manually measured on the checkerboard in the image space and saved in a file (measured_image_coordinates_96.txt).
The matrix M estimated is used to calculate the image coordinates. The 96 reconstructed image points are plotted as red dots, while the measured image coordinates are plotted as blue circles (The bigger blue circles specify the points that we have used for camera calibration). It can be seen from figure 3 that the reconstructed points are mostly inside the measured circles.

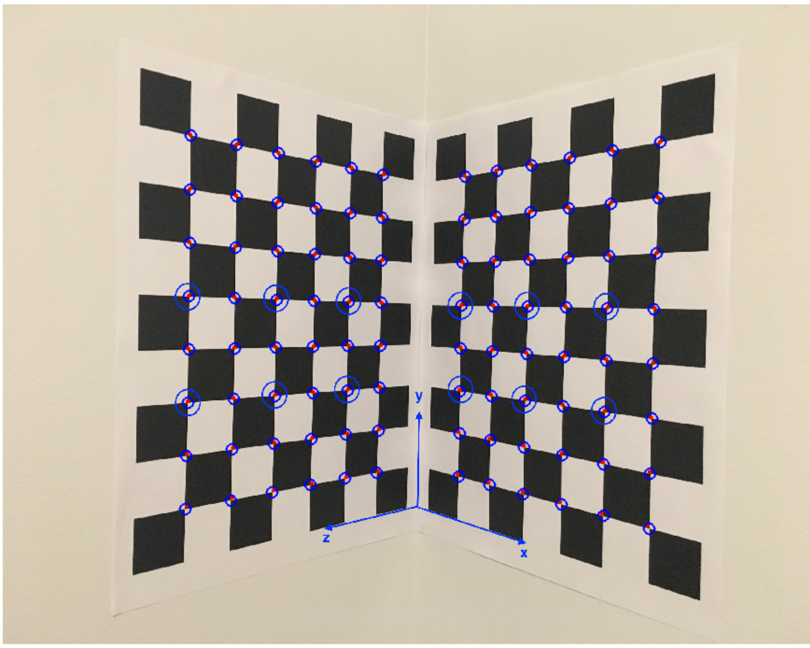


Figure 3. Measure points (circles) and reconstructed points(dots) on checkerboards

Figure 4 shows a better view without background image. The original data points (circles) are overlaid with the re-projected 3D points (dots). The root-mean-squared error is 2.71 pixel for this 4032×3024 image (root_mean_squared_error.m).

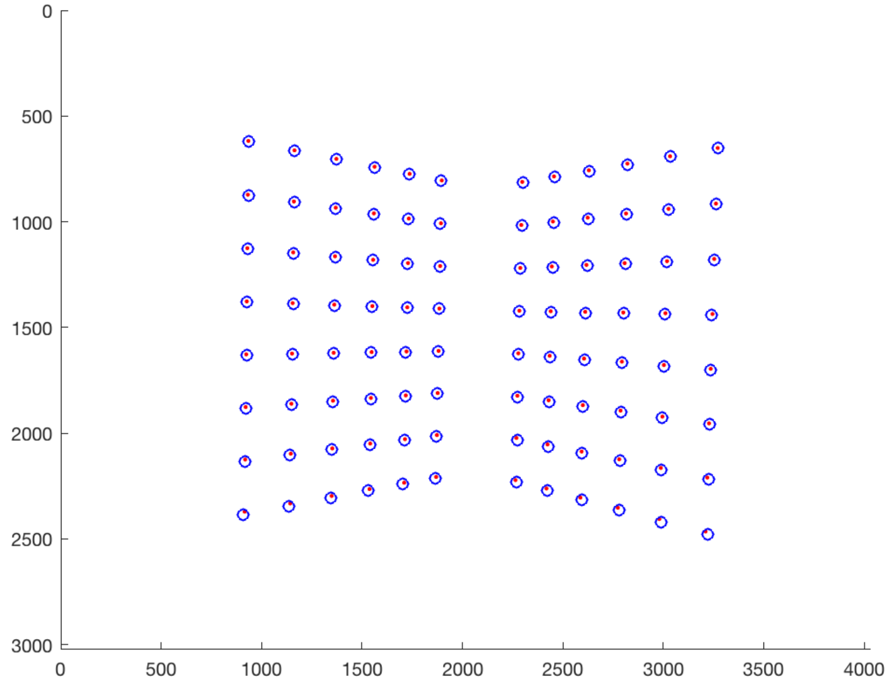
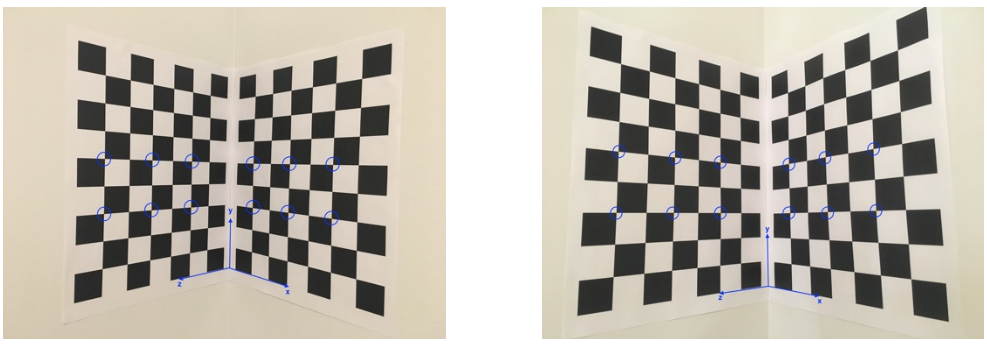


Figure 4. Measure points (circles) and reconstructed points(dots)

10. Intrinsic Parameters Comparison of Two Pictures

After taking the first picture, take another picture (Figure 5 (b)) by slightly changing the camera location. Calibrate the camera again using the second picture. Table 2 is a comparison of the intrinsic parameters of the two.



(a) (b)

Figure 5. Two pictures of the pattern with different camera location.
(a) The first picture. (b) The second picture.

Table 2. Intrinsic Parameters of two pictures taken

Parameter	First Picture	Second Picture	Percentage Error
θ	1.5764 $\approx 90.32°$	1.5788$\approx 90.46°$	0.15%
u_0	1979.31	1981.54	0.13%
v_0	1557.15	1524.63	2.13%
α	3368.84	3303.39	1.98%
β	3353.62	3315.94	1.14%

We can see from table 2 that there is tiny difference between the two, which means that their intrinsic parameters are almost the same.