Project One: Camera Calibration

Zhiwei, Zhu

1 Introduction

The purpose of camera calibration is to determine the value of the extrinsic and intrinsic parameters of the camera. The problem of camera calibration is that how to estimate the the intrinsic and extrinsic parameters given one or more images of a calibration pattern.

A wide range of existing algorithms for 3D reconstruction and recognition reply on the the knowledge of the camera parameters. Therefore, in order to make these algorithms work well, accurate camera calibration is important to obtain these parameters accurately.

2 Camera Calibration Under Full Perspective Projection

2.1 Basic Equations

Let $X = (x, y, z)^t$ be a 3D point in object frame and $U = (u, v)^t$ the corresponding image point in the image frame. $p = (c, r)^t$ be the coordinates of U in the row-column frame. In general, the perspective projection equation for the full perspective camera model is

$$\lambda \begin{pmatrix} c \\ r \\ 1 \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{1}$$

where λ is the scalar factor and the projection matrix P is

$$P = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \tag{2}$$

The camera intrinsic parameter matrix W is written as follows:

$$W = \begin{pmatrix} s_x f & 0 & c_0 \\ 0 & s_y f & r_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3}$$

where f is the camera focus length and s_x and s_y is the horizontal and vertical effective pixel size, and c_0 and r_0 is the principal point of the image.

Also, the camera extrinsic parameter matrix M can be written as follows:

$$M = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix} \tag{4}$$

Further the projection matrix P can be written as functions of the intrinsic and extrinsic camera parameters:

$$P = WM = \begin{pmatrix} s_x f r_1 + c_0 r_3 & s_x f t_x + c_0 t_z \\ s_y f r_2 + r_0 r_3 & s_y f t_y + r_0 t_z \\ r_3 & t_z \end{pmatrix} \tag{5}$$

where $r_1 = (r_{11}, r_{12}, r_{13})$, $r_2 = (r_{21}, r_{22}, r_{23})$ and $r_3 = (r_{31}, r_{32}, r_{33})$ are the row vectors of the Rotation matrix R. The (t_x, t_y, t_z) is the translation vector.

2.2 Linear Method Using 2D/3D Points

From the full projection equation 1, we can see that the coordinates of a set of 3D points are linked with their projections in the image by the projection matrix. The projection matrix is decided by the camera intrinsic parameters and extrinsic parameters. Therefore, if we know the coordinates of a set of 3D points and their corresponding projection image points, we can obtain the intrinsic parameters and the extrinsic parameters inversely.

So, the camera calibration problem can be expressed as follows. Given image points $m_i = (c_i, r_i)$ and the corresponding 3D points $M_i = (x_i, y_i, z_i)$, where $i = 1, ..., N$, we want to compute the projection matrix P. After obtaining the projection matrix P, we recover the camera parameters from it. First, let's talk about how to compute the projection matrix P based on the linear method using 2D and 3D Points.

In here, we let the projection matrix P ne represented as

$$P = \begin{pmatrix} p_1 & p_{14} \\ p_2 & p_{24} \\ p_3 & p_{34} \end{pmatrix} \tag{6}$$

where $p_i = (p_{i1}, p_{i2}, p_{i3}), i = 1, 2, 3$ and p_i is the row vector the projection matrix P.

Therefore, for each pair of 2D-3D points, we have

$$\begin{cases} p_1 M_i^t + p_{14} - c_i p_3 M_i^t - c_i p_{34} = 0 \\ p_2 M_i^t + p_{24} - r_i p_3 M_i^t - r_i p_{34} = 0 \end{cases} \tag{7}$$

For N points, we can setup a system of linear equations

$$AX = 0 \tag{8}$$

where A is $2N \times 12$ matrix depending on the 3D and 2D coordinates of the reference points, and X is a 12×1 vector $(p_1, p_{14}, p_2, p_{24}, p_3, p_{34})^t$.

Further, we have to set $N \geq 6$ and the N points can not be coplanar points.

In general, the rank of A is 11 for 12 unknowns, which means that the solution is up to a scale factor. But due to the effect of noise and location error, A maybe is full rank. Then the solution is unique. So, we can choose the last column vector of the V matrix as the solution of X after we do the Singular Value Decomposition on matrix A which yields $AV = U\Sigma$.

Because the X is up to a scale factor, so, we have to choose the correct one for the projection matrix P. We know that in the matrix P, $r_3 = p_3$, then the norm of p_3 is equal to 1. Therefore, first, we should obtain the scale factor κ using the following equation:

$$\kappa = \sqrt{{X_9}^2 + {X_{10}}^2 + {X_{11}}^2} \tag{9}$$

Then the projection matrix P can be expressed as follows:

$$P = \frac{1}{\kappa} \begin{pmatrix} X_1 & X_2 & X_3 & X_4 \\ X_5 & X_6 & X_7 & X_8 \\ X_9 & X_{10} & X_{11} & X_{12} \end{pmatrix} \tag{10}$$

2.3 Recovering Camera Parameters From the Matrix P

We know that the projection matrix links the camera intrinsic and extrinsic parameters in the following equation:

$$P = WM = \begin{pmatrix} s_x f r_1 + c_0 r_3 & s_x f t_x + c_0 t_z \\ s_y f r_2 + r_0 r_3 & s_y f t_y + r_0 t_z \\ r_3 & t_z \end{pmatrix} \tag{11}$$

After obtaining the projection matrix, we can estimate the camera intrinsic and extrinsic parameters as follows:

$$r_3 = p_3 \tag{12}$$

$$t_z = p_{34} \tag{13}$$

$$c_0 = {p_1}^t p_3 \tag{14}$$

$$r_0 = {p_2}^t p_3 \tag{15}$$

$$s_x f = \sqrt{{p_1}^t p_1 - {c_0}^2} \tag{16}$$

$$s_y f = \sqrt{{p_2}^t p_2 - {r_0}^2} \tag{17}$$

$$t_x = \frac{p_{14} - c_0 t_z}{s_x f} \tag{18}$$

$$t_y = \frac{p_{24} - r_0 t_z}{s_y f} \tag{19}$$

$$r_1 = \frac{p_1 - c_0 r_3}{s_x f} \tag{20}$$

$$r_2 = \frac{p_2 - r_0 r_3}{s_y f} \tag{21}$$

2.4 Experiment Results

In the experiments, we choose total 72 2D and 3D points pairs. For the first camera, we get the following intrinsic and extrinsic camera parameter matrices:

$$W = \begin{pmatrix} 1642.4 & 0 & 389.8 \\ 0 & 1604.4 & 249.7 \\ 0 & 0 & 1 \end{pmatrix} \tag{22}$$

$$M = \begin{pmatrix} -0.5726 & 0.8192 & -0.0329 & -104.6048 \\ 0.1105 & 0.0344 & -0.9933 & 76.9204 \\ -0.8125 & -0.5724 & -0.1102 & 1889.4 \end{pmatrix} \tag{23}$$

For the second camera, we get the following intrinsic and extrinsic camera parameter matrices:

$$W = \begin{pmatrix} 837.9286 & 0 & 317.4372 \\ 0 & 804.8882 & 325.6387 \\ 0 & 0 & 1 \end{pmatrix} \tag{24}$$

$$M = \begin{pmatrix} 0.6181 & -0.7635 & -0.1873 & 117.6694 \\ 0.1423 & -0.1272 & 0.9816 & -64.3193 \\ 0.7733 & 0.6334 & -0.0300 & -1358.6 \end{pmatrix} \tag{25}$$

2.5 Verification of Camera Calibration Results

I proposed two methods to verify the camera calibration method.

1. Method one. First, choose a new 3D point (x_n, y_n, z_n) which is not in the calibration data. Then project the 3D pint using the obtained projection matrix, get a 2D project point $p_n = (c_n, r_n)$. Compare the projected point p_n with the real point p in the image to see how close they are. If they are close, then the camera calibration result is acceptable, otherwise, it's not acceptable.

   If the result is not acceptable, then there are two possible reasons, the first one is the data we used for calibration is not precise. The second is that our algorithm is not correct. If the reason is the second one, then we can test it using the following method.

2. Method Two We can create the synthetic data to test our camera calibration method. Give a projection matrix P_g and choose a set of 3D points which are not coplanar, and produce a set of corresponding 2D points. Use these 2D and 3D point pairs to estimate the projection matrix P_e in our method. If they are very close, then the method is correct, otherwise the method is wrong.

3 Camera Calibration Under Weak Perspective Projection

3.1 Theory of Camera Calibration

For the weak perspective projection, we have the following projection equation:

$$\lambda \begin{pmatrix} c - c_0 \\ r - r_0 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{pmatrix} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix} \tag{26}$$

where (c_0, r_0) is the 2D projection point of the 3D point (x_0, y_0, z_0) and they are used as the reference point pair. The projection matrix M_p is as follows:

$$M_p = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{pmatrix} \tag{27}$$

Therefore, if we are given the 2D point (c_i, r_i) and 3D point (x_i, y_i, z_i) pairs, we can solve the projection matrix M_p.

For N point pair, we can setup a system of linear equations

$$AX = 0 \tag{28}$$

where A is $2N \times 6$ matrix depending on the 3D and 2D coordinates of the reference points, and X is a 6×1 vector $(m_{11}m_{12}m_{13}m_{21}m_{22}m_{23})^t$.

Minimum 3 non-collinear point pairs are needed to solve the vector X uniquely and the centroid of these three points is chosen as the reference point pair.

The projection matrix M_p is linked with the camera parameters as follows:

$$M_p = \begin{pmatrix} \frac{f s_x r_1}{\hat{s}} \\ \frac{f s_y r_2}{\hat{s}} \end{pmatrix} \tag{29}$$

Therefore, we can obtain the camera parameters as follows:

$$\frac{s_x}{s_y} = \frac{|\frac{f s_x r_1}{\hat{s}}|}{|\frac{f s_y r_2}{\hat{s}}|} \tag{30}$$

$$r_1 = \frac{\frac{f s_x r_1}{\hat{s}}}{|\frac{f s_x r_1}{\hat{s}}|} \tag{31}$$

$$r_2 = \frac{\frac{f s_y r_2}{\hat{s}}}{|\frac{f s_y r_2}{\hat{s}}|} \tag{32}$$

$$r_3 = r_1 \times r_2 \tag{33}$$

3.2 Experiment Results

First, we choose the points on the left panel as the 2D and 3D pairs for the camera calibration. We found that the rank of A is equal to 4, then we can not solve the equation uniquely. We found that due to the object coordinate and the reference point we chosen will make the equation 26 look like the following format:

$$\lambda \begin{pmatrix} c - c_0 \\ r - r_0 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{pmatrix} \begin{pmatrix} x - x_0 \\ 0 \\ z - z_0 \end{pmatrix} \tag{34}$$

The above equation makes us impossible to obtain the parameter m_{12} and m_{22}.

So, to solve the above calibration problem, we can choose the point which is not located in the left panel as the reference point.

If we choose all the 3D points in the left and right panels, we can have the following results.

1. The First Camera.

$$\frac{s_x}{s_y} = 1.0372 \tag{35}$$
$$r1 = (-0.6486\ 0.7591\ \ -0.0551) \tag{36}$$
$$r2 = (0.0622\ 0.0105\ \ -0.9980) \tag{37}$$
$$r3 = (-0.7570\ \ -0.6508\ \ -0.0541) \tag{38}$$

2. The Second Camera

$$\frac{s_x}{s_y} = 1.1099 \tag{39}$$
$$r1 = (-0.6960\ 0.6983\ 0.1669) \tag{40}$$
$$r2 = (-0.1700\ 0.1070\ \ -0.9796) \tag{41}$$
$$r3 = (-0.7020\ \ -0.7102\ 0.0442) \tag{42}$$

3.3 Comparison Between the Method Full Perspective and the One Under Weak Perspective

For the first camera, we found that the estimated ratio of $\frac{s_x}{s_y}$ is equal to 1.0237 under the full perspective projection and the estimated ratio of $\frac{s_x}{s_y}$ is equal to 1.0372 under the weak perspective projection. They are very close. The estimated rotation matrix under full perspective projection is

$$R = \begin{pmatrix} -0.5726 & 0.8192 & -0.0329 \\ 0.1105 & 0.0344 & -0.9933 \\ -0.8125 & -0.5724 & -0.1102 \end{pmatrix} \tag{43}$$

And we can find that the rotation matrix estimated under the weak perspective projection is very close to the one under the weak perspective projection. But there are still deviations.

I think the reason to produce a deviation for the weak perspective method from the results coming from the full perspective method is as follows. First, the object is not sufficiently far away from the camera. Second, the object we use is a two-panel object and two panels are perpendicular with each other. The relative distance between any two points chosen from both panels respectively along the optical axis is too large compared with the average distance of the total points on this two-panel object along the optical axis.

4 Camera Calibration with Lens Distortion Estimation

We present an approach for simultaneous linear estimation of the camera parameters and the lens distortion.

4.1 Incorporating the Distortion Parameter

According to the divisional lens distortion model, we have

$$\begin{pmatrix} \hat{c} - c_0 \\ \hat{r} - r_0 \end{pmatrix} = (1 + ks^2) \begin{pmatrix} c - c_0 \\ r - r_0 \end{pmatrix} \tag{44}$$

where k is the lens distortion parameter and $s^2 = (\hat{c} - c_0)^2 + (\hat{r} - r_0)^2$. This is an approximation to the camera true distortion model.

Hence, we have

$$\frac{\lambda}{1 + ks^2} \begin{pmatrix} \hat{c} - c_0 \\ \hat{r} - r_0 \\ 0 \end{pmatrix} = P \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} - \lambda \begin{pmatrix} c_0 \\ r_0 \\ 1 \end{pmatrix} \tag{45}$$

Further, we can obtain the following equation for each pair of 2D and 3D points,

$$\begin{pmatrix} M_i & 1 & 0 & 0 & 0 & 0 & -\hat{c}_i M_i & -\hat{c}_i \\ 0 & 0 & 0 & 0 & M_i & 1 & -\hat{r}_i M_i & -\hat{r}_i \end{pmatrix} V + k \begin{pmatrix} s^2 M_i & s^2 & 0 & 0 & 0 & 0 & -s^2 \hat{c}_i M_i & -s^2 \hat{c}_i \\ 0 & 0 & 0 & 0 & s^2 M_i & s^2 & -s^2 \hat{r}_i M_i & -s^2 \hat{r}_i \end{pmatrix} = 0 \tag{46}$$

For N points, we can setup a linear equations

$$(D_1 + kD_2)X = 0 \tag{47}$$

where D_1 and D_2 are $2N \times 12$ matrices. Further, we have to set $N \geq 6$ and the N points can not be coplanar points.

The above equation is 1 degree polynomial eigenvalue problem and we can solve k and V by using the MATLAB function *polyeig*. Also, for the *polyeig* function, the matrices must be square matrices. To achieve this, we can multiply both sides of the above equation by ${D_1}^t$ yields the following equation:

$$({D_1}^t D_1 + k{D_1}^t D_2)X = 0 \tag{48}$$

4.2 Experiment Results

4.2.1 Experiments Results with Synthetic Data

In order to obtain the performance of the algorithm, the experiment with the synthetic data was conducted. First, 100 3D points were generated to represent the 3D object points. Then we give the specific camera intrinsic parameter matrix W and extrinsic camera parameter matrix M to obtain the full projection matrix P. The matrix M and W are shown as follows:

$$W = \begin{pmatrix} 104.5642 & 0 & 162.1163 \\ 0 & 103.4847 & 118.0244 \\ 0 & 0 & 1 \end{pmatrix} \tag{49}$$

$$M = \begin{pmatrix} 0.2500 & 0.4330 & -0.8660 & 100.0000 \\ -0.0580 & 0.8995 & 0.4330 & 100.0000 \\ 0.9665 & -0.0580 & 0.2500 & 150.0000 \end{pmatrix} \tag{50}$$

According to the full perspective camera projection, we can get one projection image point for each of 3D points. These 100 image points are perfect image points. Further, we will add the lens distortion parameter k on these image points to get the distorted image points.

The testing procedure is as follows:

1. Assume the principal point is at the image center $(160, 120)$.
2. Use MATLAB function *polyeig* to compute $[V, e] = polyeig({D_1}^t D_1, {D_1}^t D_2)$. V is the matrix of eigenvectors and e is the vector of corresponding (inverse) eigenvalues.
3. Discard the infinite eigenvalues from e and choose the smallest absolute real part of the eigenvalue e_k. Then choose the column vector V_k from V which corresponds to the eigenvalue e_k and store the real part of entries V_k into X.
4. normalize the vector X as the projection matrix P and calculate all the intrinsic parameters to get the new principal point (C_0, R_0). Compute the distance d between the new principal point and the old principal point.
5. compare the distance d between the new principal point and the old principal point with the predefined value $\varepsilon = 0.01$. If $d < \varepsilon$, stop; Else, Repeat step 2 and 3 using the new principal point.

1. when the lens distortion parameter $k = 0.00000005529$, the following are the estimated camera intrinsic parameter matrix W and the camera extrinsic parameter matrix M:

$$W = \begin{pmatrix} 104.5655 & 0 & 162.1181 \\ 0 & 103.4857 & 118.0256 \\ 0 & 0 & 1 \end{pmatrix} \tag{51}$$

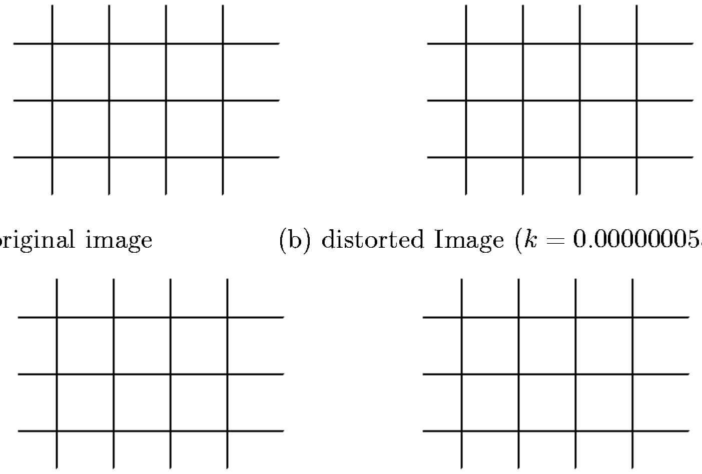
$$M = \begin{pmatrix} -0.2500 & -0.4330 & 0.8660 & -99.9972 \\ 0.0580 & -0.8995 & -0.4330 & -99.9983 \\ -0.9665 & 0.0580 & -0.2500 & -150.0028 \end{pmatrix} \tag{52}$$

The estimated lens distortion parameter $k_e = 0.000000056641$. In the following, we will show the resulted image after distortion and the undistorted images using the original lens distortion parameter k and the estimated lens distortion parameter k_e.

From Figure 1, we can see that due to the small lens distortion parameter $k = 0.00000005529$, which corresponds to around 0.02 pixel distortion in the 320×240 image, after un-distorting the image using the original lens distortion parameter k or the estimated lens distortion parameter k_e, the result images are close to the original image.

Also, the iteration number is 2.

If we use the conventional linear method to estimate the camera intrinsic and extrinsic parameters, The following shows the results:



(a) original image (b) distorted Image ($k = 0.00000005529$)

(c) undistorted Image ($k = 0.00000005529$) (d) undistorted Image ($k_e = 0.000000056641$)

Figure 1. The result images show the distortion effect and un-distortion effect.

$$W = \begin{pmatrix} 104.4847 & 0 & 162.0203 \\ 0 & 103.4266 & 117.9648 \\ 0 & 0 & 1 \end{pmatrix} \tag{53}$$

$$M = \begin{pmatrix} 0.2510 & 0.4330 & -0.8657 & 100.1488 \\ -0.0574 & 0.8994 & 0.4333 & 100.0785 \\ 0.9663 & -0.0590 & 0.2506 & 149.8216 \end{pmatrix} \tag{54}$$

Due to the small lens distortion involved in the 2D points, so the estimated camera parameters are pretty closer to the original camera parameters.

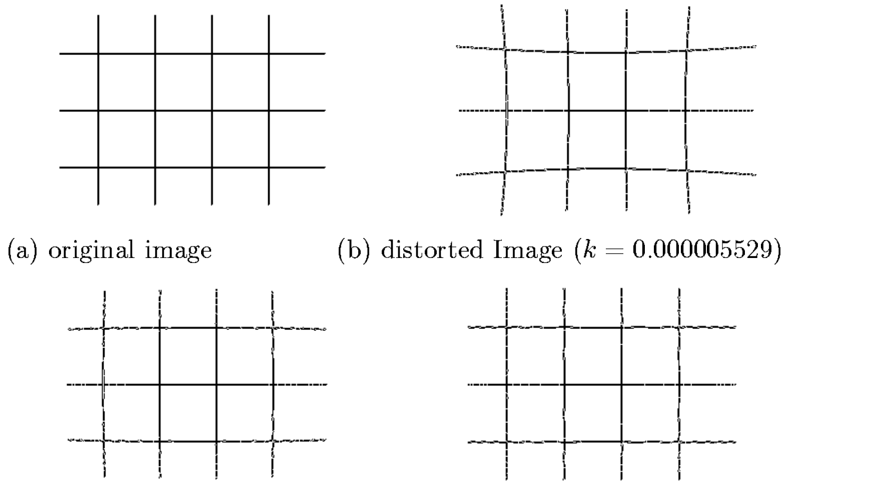
2. when the lens distortion parameter $k = 0.000005529$, the following are the estimated camera intrinsic parameter matrix W and the camera extrinsic parameter matrix M:

$$W = \begin{pmatrix} 105.9526 & 0 & 161.3977 \\ 0 & 105.1125 & 117.6026 \\ 0 & 0 & 1 \end{pmatrix} \tag{55}$$

$$M = \begin{pmatrix} -0.2505 & -0.4377 & 0.8635 & -100.6641 \\ 0.0593 & -0.9011 & -0.4296 & -100.0184 \\ -0.9662 & 0.0564 & -0.2516 & -150.1587 \end{pmatrix} \tag{56}$$

The estimated lens distortion parameter $k_e = 0.0000040394$. In the following, we will show the resulted image after distortion and the undistorted images using the original lens distortion parameter k and the estimated lens distortion parameter k_e.

The lens distortion parameter $k = 0.000005529$ corresponds to around 4 pixel distortion in the 320×240 image. From Figure 2, we can see that after un-distorting the image using the original lens distortion parameter k, the image can not be recovered very well and we can still see that the straight line is a little bended. But the estimated lens distortion parameter k_e, the result images are closer to the original image because the line is straight.



(a) original image (b) distorted Image ($k = 0.000005529$)

(c) undistorted Image ($k = 0.000005529$) (d) undistorted Image ($k_e = 0.0000040394$)

Figure 2. The result images show the distortion effect and un-distortion effect.

Also, the iteration number is 5.

If we use the conventional linear method to estimate the camera intrinsic and extrinsic parameters, The following shows the results:

$$W = \begin{pmatrix} 97.9720 & 0 & 154.3888 \\ 0 & 99.0524 & 113.2969 \\ 0 & 0 & 1 \end{pmatrix} \tag{57}$$

$$M = \begin{pmatrix} 0.3282 & 0.4327 & -0.8397 & 111.7856 \\ -0.0112 & 0.8896 & 0.4566 & 105.5228 \\ 0.9446 & -0.1405 & 0.2968 & 134.8610 \end{pmatrix} \tag{58}$$

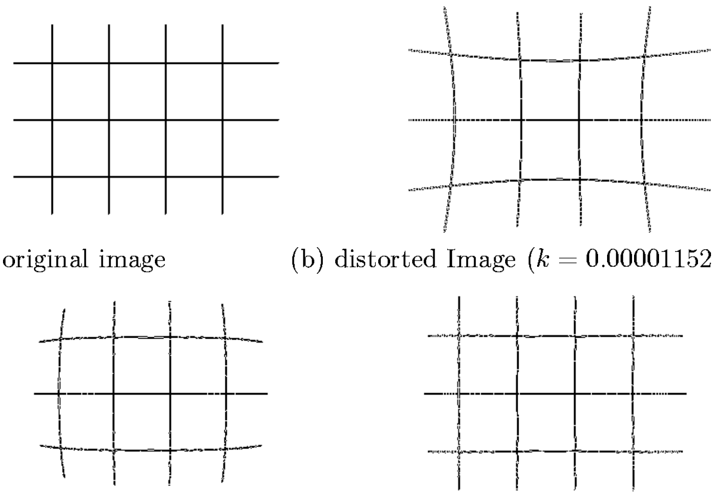
Due to the lens distortion involved in the 2D points is not small, so the estimated camera parameters are not that close to the original camera parameters. We can see that the method involved with lens distortion model is better than the conventional method.

3. when the lens distortion parameter $k = 0.000011529$, the following are the estimated camera intrinsic parameter matrix W and the camera extrinsic parameter matrix M:

$$W = \begin{pmatrix} 110.7458 & 0 & 160.4898 \\ 0 & 110.3598 & 117.1318 \\ 0 & 0 & 1 \end{pmatrix} \tag{59}$$

$$M = \begin{pmatrix} -0.2411 & -0.4501 & 0.8598 & -100.7740 \\ 0.0695 & -0.9062 & -0.4171 & -99.1306 \\ -0.9674 & 0.0408 & -0.2499 & -152.5077 \end{pmatrix} \tag{60}$$

The estimated lens distortion parameter $k_e = 0.0000065683$. In the following, we will show the resulted image after distortion and the undistorted images using the original lens distortion parameter k and the estimated lens distortion parameter k_e.



(a) original image (b) distorted Image ($k = 0.000011529$)

(c) undistorted Image ($k = 0.000011529$) (d) undistorted Image ($k_e = 0.0000065683$)

Figure 3. The result images show the distortion effect and un-distortion effect.

The lens distortion parameter $k = 0.000011529$, which corresponds to around 9 pixel distortion in the 320×240 image. From Figure 3, we can see that after un-distorting the image using the original lens distortion parameter k, the image can not be recovered very well and we can still see that the straight line is a little bended. But the estimated lens distortion parameter k_e, the result images are closer to the original image because the line is straight.

Also, the iteration number is 6.

If we use the conventional linear method to estimate the camera intrinsic and extrinsic parameters, The following shows the results:

$$W = \begin{pmatrix} 93.4878 & 0 & 148.5833 \\ 0 & 96.8073 & 109.7524 \\ 0 & 0 & 1 \end{pmatrix} \tag{61}$$

$$M = \begin{pmatrix} 0.3849 & 0.4328 & -0.8152 & 120.0696 \\ 0.0187 & 0.8800 & 0.4746 & 108.3672 \\ 0.9228 & -0.1979 & 0.3306 & 123.2147 \end{pmatrix} \tag{62}$$

Due to the lens distortion involved in the 2D points is not bigger, so the estimated camera parameters are not that close to the original camera parameters. We can see that the method involved with lens distortion model is much better than the conventional method.

4.2.2 Experiments Results With the Real Data

We applied the method on the real data, the following is the result we get.

1. The first camera.

$$W = \begin{pmatrix} 165.9383 & 0 & 324.8142 \\ 0 & 149.2775 & 221.6179 \\ 0 & 0 & 1 \end{pmatrix} \tag{63}$$

$$M = \begin{pmatrix} 0.6358 & -0.7706 & 0.0433 & 22.8757 \\ -0.0864 & -0.0097 & 0.9962 & -111.8368 \\ -0.7673 & -0.6372 & -0.0727 & -143.1665 \end{pmatrix} \quad (64)$$

The iteration number is 13.

2. The Second Camera.

   The second method can not converge if we set the predefined value $\epsilon = 0.01$. And the following shows the result which only iterates one times.

$$W = \begin{pmatrix} 140.3301 & 0 & 334.7524 \\ 0 & 157.4960 & 183.3424 \\ 0 & 0 & 1 \end{pmatrix} \quad (65)$$

$$M = \begin{pmatrix} -0.1631 & 0.9775 & 0.1336 & -163.5404 \\ -0.4365 & 0.0239 & -0.8994 & 268.6478 \\ -0.8827 & -0.2050 & 0.4229 & 161.7017 \end{pmatrix} \quad (66)$$

5 Conclusion

In these project, I learned how to do the camera calibration. Further, I learned that the conventional linear method to calibrate the camera works under the assumption that the lens distortion is negligible, and the 2D-3D point pairs are pretty precise. If we know that the camera has lens distortion, we can combine the lens distortion model into the calibration procedure and estimate the lens distortion parameter and the camera parameters together.