Computer Vision
Camera Calibration

Student: Yuchang Jiang

Date: 29 Sep. 2020

1. Data Normalisation

To improve stability for Direct Linear Transform (DLT), input data points (both 2D and 3D points) are normalised to have zero mean and unit distance from origin.

1.1 Compute translation elements and shift points

At first, centroids for 2D and 3D points are computed separately:

$$x_{centroid} = \sum_{i=1}^{N} x_i \qquad y_{centroid} = \sum_{i=1}^{N} y_i \tag{1.1}$$

$$X_{centroid} = \sum_{i=1}^{N} X_i \qquad Y_{centroid} = \sum_{i=1}^{N} Y_i \qquad Z_{centroid} = \sum_{i=1}^{N} Z_i \tag{1.2}$$

where N is the number of data points. With centroids, T', U' can be formed:

$$T' = \begin{bmatrix} 1 & 0 & -x_{centroid} \\ 0 & 1 & -y_{centroid} \\ 0 & 0 & 1 \end{bmatrix} \qquad U' = \begin{bmatrix} 1 & 0 & 0 & -X_{centroid} \\ 0 & 1 & 0 & -Y_{centroid} \\ 0 & 0 & 1 & -Z_{centroid} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.3}$$

After adding '1' as the last coordinate for all data points, they can be shifted towards centroid: $xy_{shifted} = T' \cdot xy$, $XYZ_{shifted} = U' \cdot XYZ$. (inside xy or XYZ, each column contain a data point and the last coordinate equals to 1)

1.2 Compute scale elements

To ensure they have unit distance to the origin, scale factor is calculated based on average Euclidean distance of shifted data points. Also, $\sqrt{2}$ and $\sqrt{3}$ are used to force unit distance in space with different dimensions.

$$s_{xy} = \frac{\sqrt{2}}{xy_{meandist}} \qquad s_{XYZ} = \frac{\sqrt{3}}{XYZ_{meandist}} \tag{1.4}$$

1.3 Form transformation matrix and normalise points

Now both translation and scale elements are known, transformation matrix can be formed, $T = s_{xy} \cdot T'$ and $U = s_{XYZ} \cdot U'$. Thus, data points can be transformed: $xy_{normalised} = T \cdot xy$, $XYZ_{normalised} = U \cdot XYZ$. (xy and XYZ are homogeneous coordinates with last coordinate equals to 1)

2. Direct Linear Transform (DLT)

2.1 Compute P with normalised points

Firstly, normalize data points as discussed in section 1. The based on $x_i = PX_i$, the correspondence of 2D and 3D points can form equation $A_i \cdot p = 0$ to compute the elements of P, where A_i is composed by each pair of 2D and 3D data points:

$$A_i = \begin{bmatrix} X_i & Y_i & Z_i & 1 & 0_{1*4} & -x_iX_i & -x_iY_i & -x_iZ_i & -x_i \\ 0_{1*4} & -X_i & -Y_i & -Z_i & -1 & y_iX_i & y_iY_i & y_iZ_i & y_i \end{bmatrix} \tag{2.5}$$

$$p = \begin{bmatrix} P^1 & P^2 & P^3 \end{bmatrix}^T \tag{2.6}$$

If there are N data points, $A_{2n*12} \cdot p_{12*1} = 0$. (As the degree of freedom is 11 for p (up to scale), N should be at least 6.) To solve p, singular value decomposition of A is performed:

$$U, S, V^T = A \quad p = V(:, end) \tag{2.7}$$

where p is considered as the unit vector with smallest singular value. Then p_{12*1} can be reshaped into Pn_{3*4}.

2.2 Decompose P into K, R and t

Before decomposition, P should be denormalised:

$$P = T^{-1} \cdot Pn \cdot U \tag{2.8}$$

Then perform QR-decomposition on the inverse of the left 3x3 part of P and K and R are obtained by invert both resulted matrices. There are three things should be notices: ensure the last element of K is 1, make all diagonal elements of K non-nagative and det(R) is 1.

As for camera center, C, it is the right null vector of P so it can be computed by singular value decomposition. (only first 3 coordinates of C are needed) Then $t = -R \cdot C$

2.3 Computer reprojection error

To check the quality of result, 3D points are reprojected using resulted P and compared with true 2D points to compute average reprojection error.

2.4 Result and discussion

$$P = \begin{bmatrix} -2.7094 & 0.3385 & 0.4545 & 539.4155 \\ -0.8985 & -1.3098 & -1.8731 & 702.2368 \\ -0.0009 & -0.0012 & 0.0006 & 0.6215 \end{bmatrix} \tag{2.9}$$

$$K = \begin{bmatrix} 148.43 & 16.8 & 870.9 \\ 0 & 145.03 & 477.9 \\ 0 & 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} -0.8114 & 0.5842 & -0.0193 \\ -0.2048 & -0.3151 & -0.9267 \\ -0.5475 & -0.7480 & 0.3753 \end{bmatrix} \quad t = \begin{bmatrix} -2.7308 \\ 173.7424 \\ 386.4019 \end{bmatrix} \tag{2.10}$$

The average reprojection error is 0.9597 and as shown in Figure 1, the reprojection seems quite good. To check the function of normalisation, DLT without data normalisation is also performed. To my surprise, the reprojection error is 0.9556, even smaller than result with normalisation. This small error can be caused by my poor manual point selection. Besides error, the P matrix without normalisation is noticeable: all elements become smaller. As P matrix is up to scale in our case, the relationship between P matrix with and without normalisation is proportion change. Thus, there is only small difference for K, R, t matrix.

$$P = \begin{bmatrix} -0.0031 & 0.0004 & 0.0005 & 0.6092 \\ -0.0010 & -0.0015 & -0.0021 & 0.7930 \\ -0.0000 & -0.0000 & 0.0000 & 0.0007 \end{bmatrix} \tag{2.11}$$

Besides algorithm , the selection of data points is relevant for final results: choosing points covering all chessboard is better than points clustered at corner. As shown in Figure 2, the clustered data points lead to larger error, 5.1059. The error will even increase if all clustered points lie on one plane of chessboard, not on two planes.

From resulted intrinsic parameters (K), we can tell the pixel is almost square as $148.43 \approx 145.03$, which means focal length represented in unit of pixel in x, y direction are almost same. It is also noticeable skew coefficient here is non-zero so this camera belongs to finite projective camera. As discussed in section 6.2.4 in book 'Multiple View Geometry': a true CCD camera has a zero skew parameter. If it is not zero, the x, y axis of pixel in CCD are not perpendicular. This book also suggests this non-zero skew coefficient may appear when calibration photo is an image of an image.

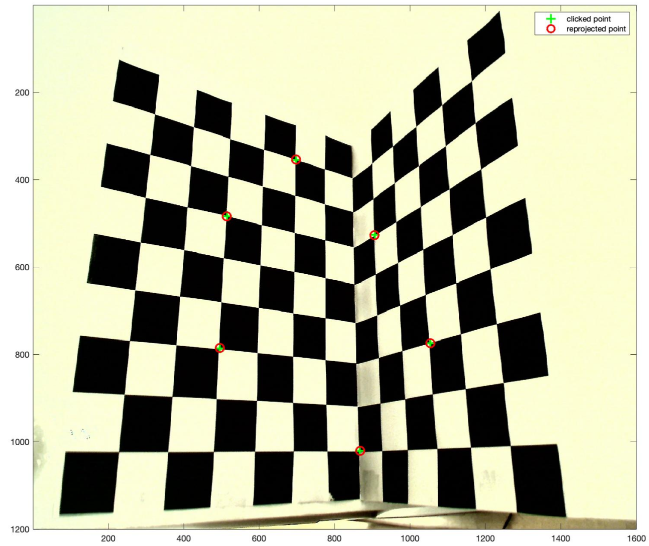

Figure 1: Clicked 2D points and reprojected point with DLT

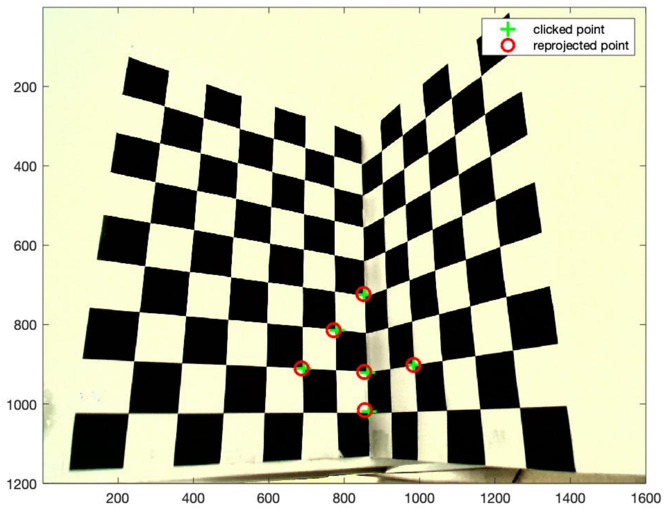

Figure 2: Clicked 2D points and reprojected point with DLT (clusted points)

3. Gold Standard Algorithm

3.1 Computation

This algorithm uses DLT for initial value of P and then try to minimize sum of sqaured reprojection errors $min_{P'} \sum_{i=1}^{N} d(x_i, P'X_i)^2$ iteratively. After that, denormalise P and decompose into K, R and t.

3.2 Result and discussion

The following result is based on the same 6 points used in DLT (evenly distributed).

$$P = \begin{bmatrix} -2.7155 & 0.3337 & 0.4543 & 540.1845 \\ -0.9041 & -1.3146 & -1.8760 & 703.2302 \\ -0.0009 & -0.0012 & 0.0006 & 0.6224 \end{bmatrix} \tag{3.12}$$

$$K = \begin{bmatrix} 147.66 & 17 & 873 \\ 0 & 144.50 & 482.1 \\ 0 & 0 & 1 \end{bmatrix} R = \begin{bmatrix} -0.8104 & 0.5855 & -0.0197 \\ -0.2038 & -0.3133 & -0.9275 \\ -0.5492 & -0.7477 & 0.3732 \end{bmatrix} t = \begin{bmatrix} -3.3113 \\ 172.5995 \\ 384.9575 \end{bmatrix} \tag{3.13}$$

The reprojection error is 0.9075 pixel, which is smaller than the error of DLT so it proves the usefulness of Gold Standard algorithm. As shown in Figure 3, the reprojection seems quite good. Usually more selected points can lead to more accurate results so 13 points are selected to run this algorithm again. Surprisingly, the reprojection error is 1.2299, which is larger than the error with 6 points. As shown in Figure 4, the selected points are evenly distributed. However, it is quite hard to click at the corner exactly, my manual click may add error to coordinates and increase final error.

The resulted camera intrinsic parameters are similar to the result of DLT so it also has non-zero skew coefficient and almost square pixel.

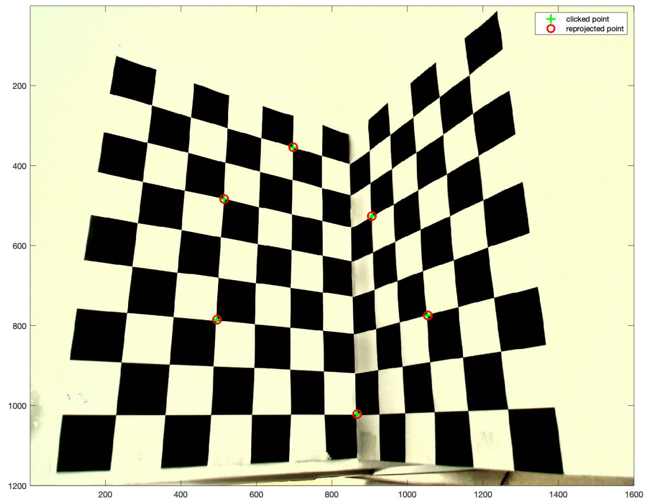

Figure 3: Clicked 2D points and reprojected point with Gold Standard algorithm (6 points)

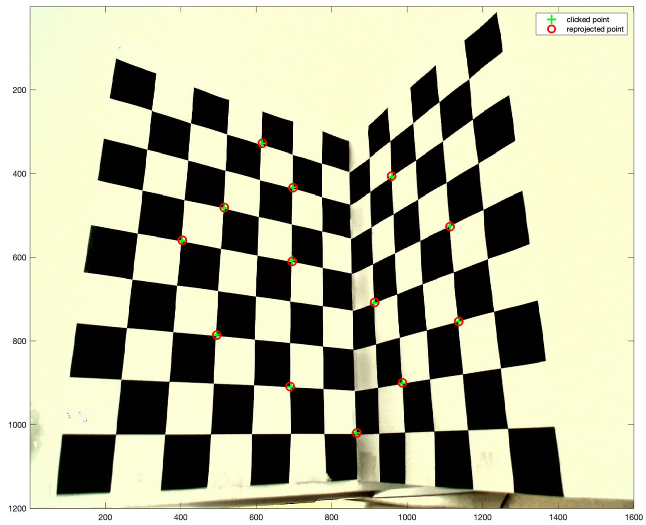

Figure 4: Clicked 2D points and reprojected point with Gold Standard algorithm (13 points)