IC5303 Computer Vision

Student : Nguyen Van Chuong. ID: 20161199

October 30, 2017

Program Assignment 1: Camera calibration from 3D calibration rig

1 Compute projection matrix P

Use DLT method to compute P. The points to be measured are shown with white dot. All world frame coordinates are measured in centimeters, and the points are measured in the image coordinate frame, with the origin at top left corner of the image. The image of rig has resolution 1600×1200. To guarantee unique solution of the equation (2), the matrix $\mathbf{A}$, which is given below, satisfying $\mathbf{A}^\top\mathbf{A}$ be non-singular. It means that rank($\mathbf{A}^\top\mathbf{A}$) or rank$\mathbf{A}$ must be 11. Therefore, we need to choose at least 6 points so that the having number of rows larger than number of columns.

We denote (X,Y,Z) as a position of a point in world coordinates and (u,v) as the corresponding point in image coordinate. I select 8 points in 3 planes by hand with position as follows

Table 1: Gather 8 points by hand.

Point	1	2	3	4	5	6	7	8
X	36	12	0	0	24	36	48	0
Y	0	0	0	48	36	12	0	48
Z	36	36	0	48	0	0	0	24
u	545	703	760	1136	853	621	469	1101
v	405	321	531	370	772	717	715	555

The calibration matrix $\mathbf{P} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & 1 \end{bmatrix}$, where the value p_i is the solution of the following equation

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 \\ .... \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_NX_n & -u_1Y_N & -u_1Z_n \\ 0 & 0 & 0 & 0 & X_N & Y_N & Z_N & 1 & -v_NX_N & -v_NY_N & -v_NZ_N \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \\ p_8 \\ p_9 \\ p_{10} \\ p_{11} \end{bmatrix} = \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ ... \\ u_n \\ v_n \end{bmatrix} \tag{1}$$

or

$$\mathbf{Ap} = \mathbf{D} \tag{2}$$

In order to find $\mathbf{p}$, we apply the Direct Linear Transformation (DLT) method

$$\mathbf{p} = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{D} \tag{3}$$

The written code by Matlab generates the result:

$$\mathbf{P} = \begin{bmatrix} -7.0864 & 4.5968 & -1.2777 & 59.7768 \\ 2.2809 & 2.7054 & -7.5451 & 530.9689 \\ -0.0022 & -0.0017 & -0.0022 & 1 \end{bmatrix} \tag{4}$$

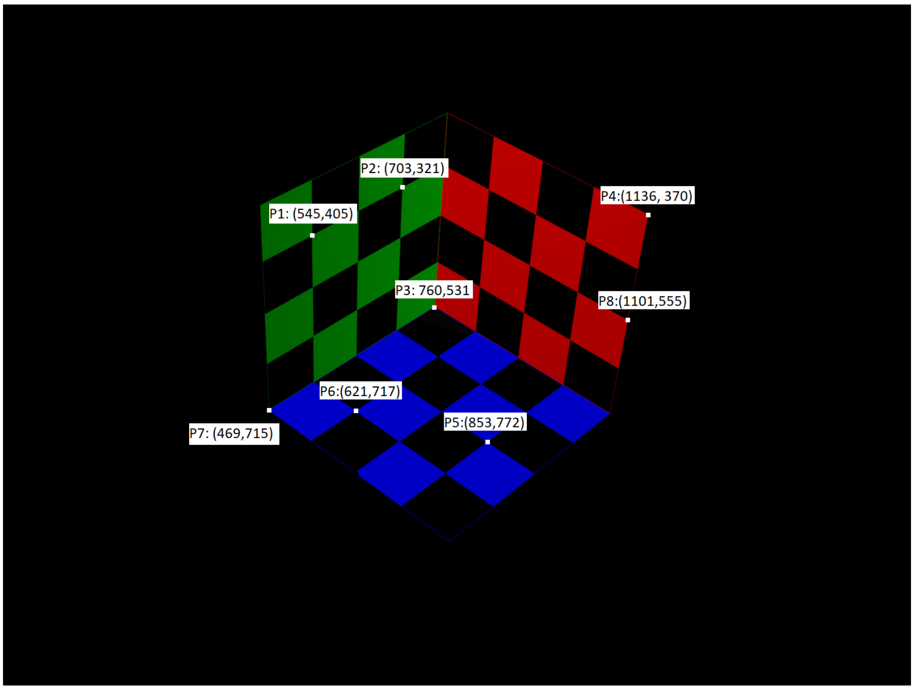

Figure 1: Gather 8 points by hand.

2 Intrinsic matrix, rotation matrix and translation matrix

$$\mathbf{P} = \mathbf{KR}[\mathbf{I}_3 - \tilde{\mathbf{C}}] = \mathbf{M}[\mathbf{I}_3 - \tilde{\mathbf{C}}] \tag{5}$$

Step 1:
It is well noted that $\mathbf{C} = [\tilde{\mathbf{C}}\ 1]^\top$ is the null vector of matrix $\mathbf{P}$. In order to find $\mathbf{C}$, we use Singular Value Decomposition (SVD) to write $\mathbf{P} = \mathbf{USV}^\top$. As a result, $\mathbf{C}$ is the unit singular vector of $\mathbf{P}$ corresponding to the smallest singular value, or the last column of $\mathbf{V}$.
By dividing C for the last term C(4), we obtain the matrix C in the form of $\mathbf{C} = [\tilde{\mathbf{C}}\ 1]^\top$. The result is

$$\mathbf{C} = \begin{bmatrix}168.7196 & 142.7812 & 172.5732 & 1\end{bmatrix}^\top \tag{6}$$

Step 2:
The non-singular square matrix $\mathbf{M}$ can be decomposed into product of upper-triangular matrix $\mathbf{K}$ and an orthogonal matrix $\mathbf{R}$ using the RQ factorization. It is similar to QR factorization, we have

$$[\mathbf{Q}_{de}, \quad \mathbf{R}_{de}] = qr(\mathbf{M}(3:-1:1, 3:-1:1)^\top, 0) \tag{7}$$

then

$$\mathbf{K} = \mathbf{R}_{de}(end:-1:1, end:-1:1)^\top \tag{8a}$$
$$\mathbf{R} = \mathbf{R}_{de}(end:-1:1, end:-1:1)^\top \tag{8b}$$

The translation matrix $\mathbf{t} = -\mathbf{R}\tilde{\mathbf{C}}$
The Matlab code gives

$$\mathbf{M} = \begin{bmatrix}-7.0864 & 4.5968 & -1.2777\\ 2.2809 & 2.7054 & -7.5451\\ -0.0022 & -0.0017 & -0.0022\end{bmatrix} \tag{9}$$

$$\mathbf{K} = \begin{bmatrix}-8.0501 & -0.0253 & 2.8592\\ 0 & -8.0910 & 1.9964\\ 0 & 0 & 0.0036\end{bmatrix} \tag{10}$$

$$\mathbf{R} = \begin{bmatrix} 0.6655 & -0.7436 & -0.0654 \\ -0.4321 & -0.4552 & 0.7785 \\ -0.6086 & -0.4898 & -0.6242 \end{bmatrix} \tag{11}$$

$$\mathbf{t} = \begin{bmatrix} 5.1799 & 3.5501 & 280.3477 \end{bmatrix}^\top \tag{12}$$

The estimated position of a camera center in the world frame is

$$\tilde{\mathbf{C}} = \begin{bmatrix} 168.7196 & 142.7812 & 172.5732 \end{bmatrix}^\top \tag{13}$$

It is closed to the ground truth $X = 166.20, Y = 141.46, Z = 170.08$. The error is small but likely to exist because of the manually selection points using mouse click.

```matlab
% Written by Nguyen Van Chuong, ID 20161199

N=8;
X(1:N)=0;Y(1:N)=0;Z(1:N)=0;
X(1)=36; X(2)=12; X(3)=0;X(5)=24;X(6)=36; X(7)=48;
Y(4)=48;Y(5)=36;Y(6)=12; Y(3)=0;Y(8)=48;
Z(1)=36; Z(2)=36; Z(4)=48;Z(3)=0;Z(8)=24;

u=[545 703 760 1136 853 621 469 1101];
v=[405 321 531 370 772 717 715 555];
A(1:2*N,1:11)=0;
% creat matrix A based on 8 points in world coordinates and image
% coordinates
i=1;
% Construct matrix A
for k=1:2:2*N
        A(k,1)=X(i);A(k,2)=Y(i);A(k,3)=Z(i);A(k,4)=1;
        A(k,9)=-u(i)*X(i);
        A(k,10)=-u(i)*Y(i); A(k,11)=-u(i)*Z(i);

        A(k+1,5)=X(i);A(k+1,6)=Y(i);A(k+1,7)=Z(i);
        A(k+1,8)=1;A(k+1,9)=-v(i)*X(i);
        A(k+1,10)=-v(i)*Y(i);A(k+1,11)=-v(i)*Z(i);
i=i+1;
end

D=[u(1) v(1) u(2) v(2) u(3) v(3) u(4) v(4) u(5) v(5) u(6) v(6) u(7) v(7) u(8)
v(8)];

p=inv(A'*A)*A'*D';
P(1:3,1:4)=0;
P=[p(1) p(2) p(3) p(4)
   p(5) p(6) p(7) p(8)
   p(9) p(10) p(11) 1];

[U,S,V]=svd(P);
%C is the unit singular vector of $\mathbf{P}$ corresponding to the smallest
singular value,
% or the last column of V
C = V(1:4, 4);
C=C/C(4);
C_tidle=[C(1) C(2) C(3)];
C_tidle=C_tidle';

M=P(1:3,1:3);
[Q_de, R_de] = qr (M (3:-1:1, 3:-1:1)', 0) ;
K = R_de (end:-1:1, end:-1:1)' ;
R = Q_de (end:-1:1, end:-1:1)' ;
t=-R*C_tidle;
```