Computer Vision and Scene Analysis(Prof. Guido Gerig)

Assignment 1

Camera Calibration

By:

Monil Shah
NYU
NetID: mds747

: MONIL SHAH
mds747

COMPUTER VISION AND SCENE ANALYSIS

Homework-1

a) "A straight line in the world space is projected onto a straight line at the image plane". (By Geometric consideration)

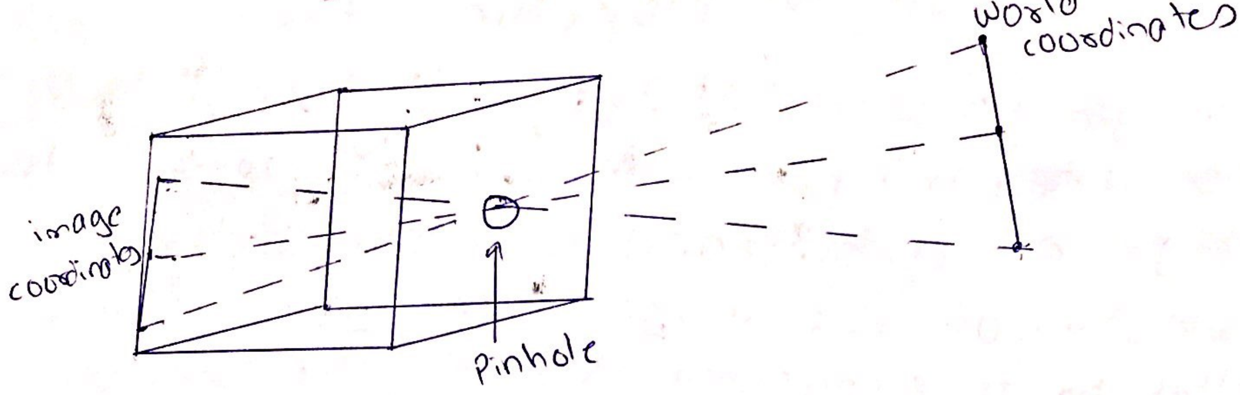

Fig. 1

b) Show that, in the pinhole camera model, three collinear points in 3-D space are imaged into three collinear points on the image plane.

> In a 3-D space as shown in Fig. 1, we can say that equation of line is a linear equation. Parametric equation of line for X and Y can be given as

$X(t) = X_0 + tU$, where U is $(X_2 - X_1)$ — ①

$Y(t) = Y_0 + tV$, when V is $(Y_2 - Y_1)$ — ②

Now, equation for Pinhole camera.

$X = -f\left(\frac{X}{Z}\right)$ $Y = -f\left(\frac{Y}{Z}\right)$, substituting in ① & ② respectively

Image coordinates:- $x = -f(X_0 + tU)/(Z_0 + tW)$ — ③

$y = -f(Y_0 + tV)/(Z_0 + tW)$ — ④

From equation (3) & (4), we can know that the equation for line in the image plane is also linear. Hence proved.

Problem 2 :- Perspective Projection.

a) Prove geometrically that the projections of two parallel lines lying in some plane π appear to converge on a horizon line H formed by the intersection of the image plane with the plane parallel to π & passing through the pinhole.

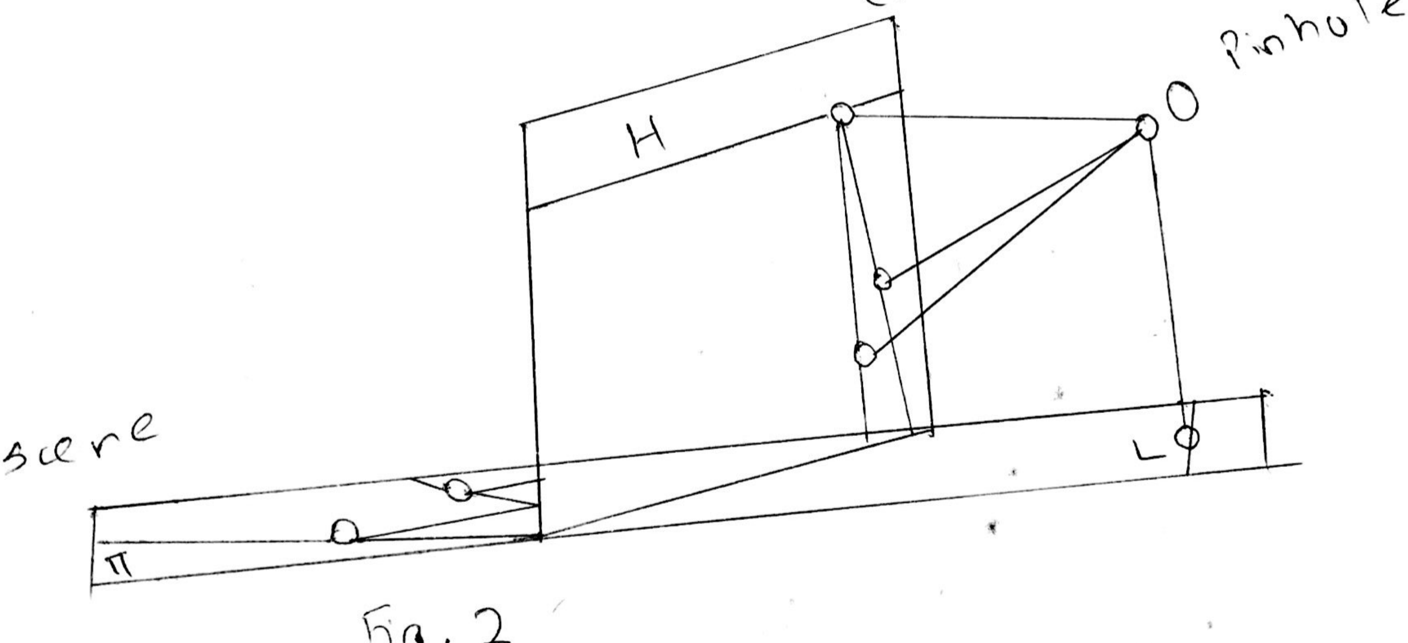

Fig. 2

Source :- Computer Vision Textbook, Forsyth & Ponce.

~~optical centre of~~

b) Prove that same result algebraically using perspective projection equation. You can assume for simplicity that the plane Π is orthogonal to image plane.

⇒ Equation of line in world space can be given as,

$$ax + bz = m \quad - (1)$$

Now, assuming the plane Π is orthogonal to image plane, the equation of plane can be given as

$$y = c \quad - (2)$$

According to equation of image coordinates we know that

$$x' = f'\frac{x}{z} \qquad y' = f'\frac{y}{z}$$

From equation (1) & (2)

$$x' = f'\frac{m - bz}{az} \quad - (3) \qquad y' = f'\frac{c}{z} \quad - (4)$$

Equation (3) & (4) are parametric representation of image δ of line Δ with z as parameter.

When $z \to -\infty$, $(x', y') = (-f'b/a, 0)$ on x' axis of image plane, hence half line.

This is vanishing point associated with all parallel lines with slope $-b/a$ in plane Π.

Q-3 ~~Let~~ Coordinates of Optical Center.

Let O denote the homogeneous coordinates vector of optical center of a camera in some reference frame, & let M denote the corresponding perspective projection matrix. show that MO = 0.

=> We know that, perspective projection matrix

$M = K(R t)$ from extrinsic parameters.

$M = K\left({}^{C}_{W}R \;\; {}^{C}O_{W} \right)$

vector O of camera optical center in world coordinate system

$O = \begin{pmatrix} {}^{W}O_{C} \\ 1 \end{pmatrix}$

$\therefore MO = K\left({}^{C}_{W}R \;\; {}^{C}O_{W} \right) \begin{pmatrix} {}^{W}O_{C} \\ 1 \end{pmatrix}$

$= K\left({}^{C}_{W}R \, {}^{W}O_{C} + {}^{C}O_{W} \right)$

Rotate camera to world coordinates, so ${}^{C}_{W}R$ vanishes

$\therefore MO = K(-{}^{C}O_{W} + {}^{C}O_{W})$

$\therefore \boxed{MO = 0}$

Scanned by CamScanner

Practical Assignment Report

Aim: To calibrate a camera for a fixed focal length using two orthogonal checkerboard planes, and to find intrinsic and extrinsic parameters.

Experimental Procedure

Data Capture Two checkerboard patterns are pasted on a wall corner at an angle of 90 degree to each other, as shown in the following figure. 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.

Total 60 points are chosen , with 30 on one checkerboard and 30 on the other checkerboard. The real world coordinates for these 60 points are measured using a ruler. All world frame coordinates are measured in centimeters. The picture of the checkerboard pattern is captured using a camera and the coordinates of these 60 points are measured in the image coordinate frame (with origin taken at top left corner of the image). This step gives us the coordinates of 60 pairs of points with each pair having one point in world frame and the corresponding point in image frame. We need to estimate 11 free parameters for camera calibration so this number of points is more than sufficient. We choose a large number of points so that human measurement errors (in marking points in image and world frame) are averaged out and we get a robust estimate of the camera parameters.

2) Estimation of the Calibration Matrix Least squares method is used to estimate the calibration matrix. There are 120 homogeneous linear equations in twelve variables, which are the coefficients of the calibration matrix M. Lets denote this system of linear equations as Pm = 0, m := [m1 m2 m3] T , (1) where, m1, m2, m3 are first, second and third rows of the matrix M respectively. m is a 12 ×1 vector, and P is a 120 × 12 matrix. The problem of least square estimation of P is defined as min||Pm|| ^{2} , subject to ||m||^2 = 1. (2) As it turns out, the solution of above problem is given by the eigenvector of matrix P TP having the least eigenvalue. The eigenvectors of matrix P TP can also be computed by performing the singular value decomposition (SVD) of P. The 12 right singular vectors of P are also the eigenvectors of P TP. The SVD method is used here to get the eigenvector corresponding to the least eigenvalue. This eigenvector is the solution to the above problem. Reorganizing the 12 ×1 vector m in a matrix of 4×3 gives us the matrix M. The first three elements of the third row of this matrix denote one of the three rotation vectors. As we know that the norm of rotation vectors is unity, this matrix is scaled by the norm of the third rotation vector to get the calibration matrix in canonical form. We take this matrix as the final calibration matrix M.

3) Computation of Intrinsic and Extrinsic Parameters The intrinsic and extrinsic parameters are computed from the calibration matrix M by using the method given in the book. These methods use the various properties of the rotation vectors, namely the norm of them being one and the dot product of two rotation matrix being zero. Once the calibration matrix is estimates, we pick 4 different points on the checkerboard pattern (other than the points picked before) and record their coordinates in the world frame. These points are then transformed to image frame using the calibration matrix obtained using least squares estimation. These coordinates are then compared with measured coordinates of these points in the image frame. We give the errors obtained in the following section.

Results

The estimated calibration matrix using least squares estimation is given below.

M =

-830.575	-483.765	-1859	32864
-673.057	-594.663	-2183	37741
-0.2905	-0.2449	-0.9250	16.0668

Parameter Value

- theta = 0.8528
- u0 = 2079.7
- v0 = 2360.5
- Alpha = 178.5279
- Beta = 15.6605

The intrinsic parameters are tabulated below. The extrinsic parameters consist of rotation matrix and the translation vector. The rotation matrix is given below:

R =

0.7421	0.5526	-0.3793
0.6041	-0.7967	0.0212
-0.2905	-0.2449	-0.9250

The translation vector t is estimated as K−1b, where K is the intrinsic parameter matrix and b is the last column of the calibration matrix M. The t obtained in our experiments is $[0.1519\ 7.3976\ 75.6330]^T$.

Discussion

Intrinsic Parameters

We get θ almost equal to $\pi/2$ (Table 1), which means that the camera coordinates are not skewed much and the X and Y axes in the image frame are almost at 90° to each other. As mentioned in the Sec. 2, the image resolution is 2048×1536. This means that the center of the image lies at (1024, 768). We obtain u_0 and v_0 equal to 2079.7 and 2360.5 respectively. The image center does not coincide with the principle point C_0 and is offset by (8.5, 166.4), as estimated in the experiments. α and β are magnifications equal to kf and lf respectively. The terms k and l 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.

Extrinsic Parameters

Three rows of rotation matrix have norm equal to 1 and their dot product with each other is a very low number (of the order of 10^{-18}). This is justified because of possible errors in the measurement process which is purely manual. The translation vector also approximately matches the real values. We measured the difference in the world origin and the camera along the three coordinates and the values match approximately. A plain distance measure from the world origin to the camera was measured almost equal to 75 centimeters which is approximately equal to the norm of the translation vector (76 centimeters). The experiments using total 60 points give a very good estimates of the camera parameters.

Image Coordinates Reconstruction

To reconfirm the validity of the estimated parameters, we identify 4 new test points marked as red dots in the following figure.

Figure 2: Checkerboard pattern on the wall corner

4 new test points The coordinates of these points are measured in the world coordinate system, and the camera calibration matrix estimated in the previous sections is used to get the corresponding coordinates in the image frame. We also measure the true coordinate values of these points using MATLAB. These true measurements are also prone to error as they are done manually using the mouse click. Results of this experiment are tabulated below:

World Coordinates	Image Coordinates(estimated)	Image Coordinates(measured)	Error Norm
0 0.9 0 1	321.17 317.1	207.50 242.30	2.78
0 3.7 0 1	297.91 352.51	223.70 245.90	2.74
0.9 0 25.2 1	-144.27 -198.0	182.30 563	1.82
3.7 0 25.2 1	-157.01 -194.7	195.50 599	1.79

Table 1: Test of the Calibration Parameters

Error between measured and estimated image coordinates 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. We note that the reconstruction errors are not so big and can be justified given the amount of inherent errors in the measuring process itself.

BONUS HOMEWORK :

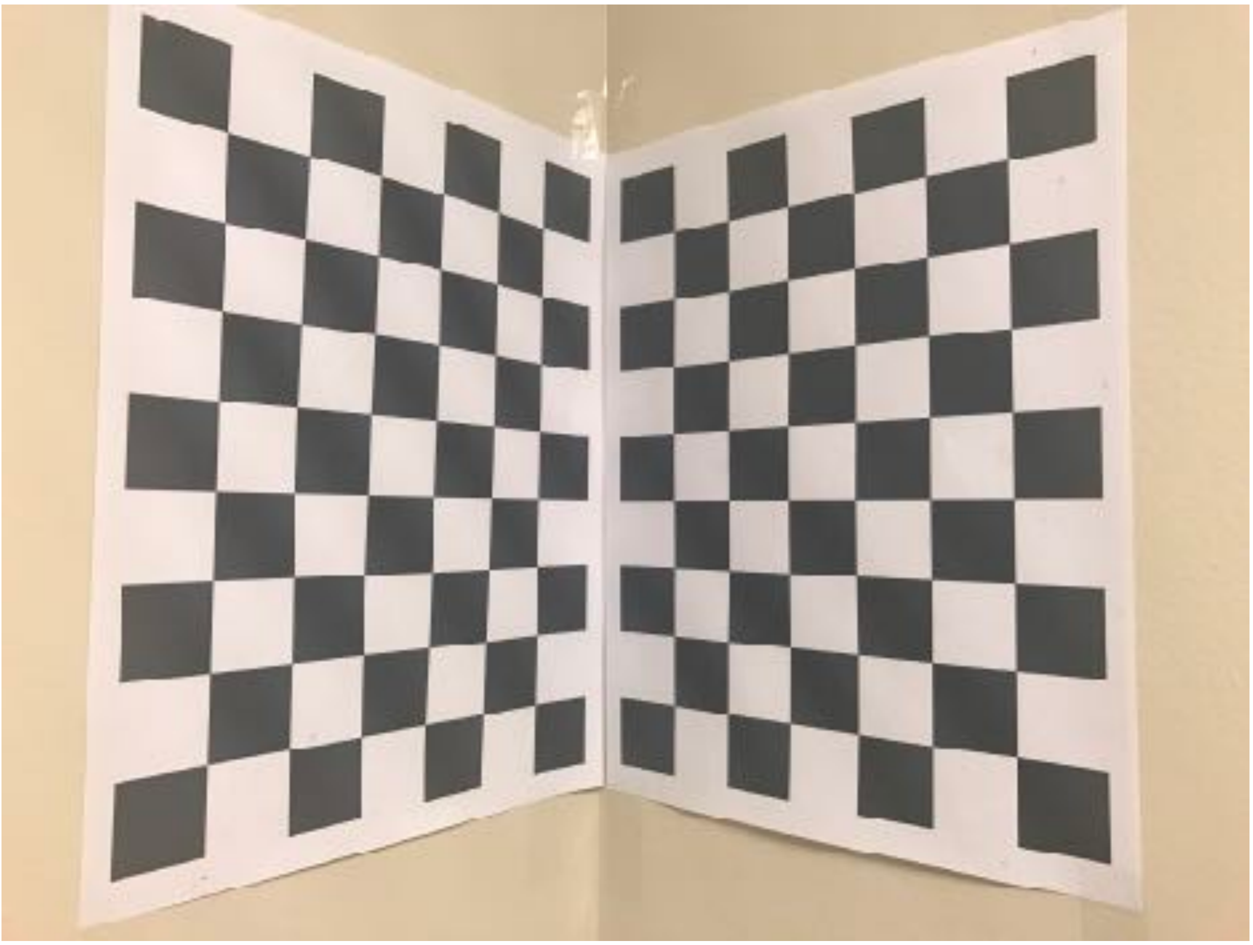

Figure 3: Image with different distance

As shown in the above experiment , the coordinate axis are taken as before. The 60 points are selected in the world coordinates and 60 points are selected in the image coordinates using ginput. Large number of points are selected to avoid human error.

The entire experiment is performed as mentioned earlier , Calibration Matrix and intrinsic parameters are calculated.

Results and comparison:

Calibration Matrix -
M=

271.4065	1004.2	1918.1	-31352
467.2973	1359.7	1691.5	-29378
0.1705	0.6071	0.7761	-13.2751

Comparison between the intrinsic parameters of two images. The intrinsic parameters are nearly the same, the theta angle is near to each other.

Intrinsic Parameters(Image 1)	**Intrinsic Parameters (Image 2)**
u_0 = 2079.7	u_0 = 2144.565
v_0 = 2360.5	v_0 = 2217.957
theta = 0.8528	theta = 1.0234
alpha = 178.5279	alpha = 343.6348
beta = 15.6605	beta = 80.995

Table 3: Comparison between the intrinsic parameters of both the images

Matlab code:

```
%worldcoordinates=w
wz(1:6) = 7*2.8;
wz(7:12) = 6*2.8;
wz(13:18) = 5*2.8;
wz(19:24) = 4*2.8;
wz(25:30) = 3*2.8;
wz(31:36) = 7*2.8;
wz(37:42) = 6*2.8;
wz(43:48) = 5*2.8;
wz(49:55) = 4*2.8;
wz(55:60) = 3*2.8;
wx(1:60)=0
wx(1:6:30)=0.9+6*2.8;
wx(2:6:30)=0.9+5*2.8;
wx(3:6:30)=0.9+4*2.8;
wx(4:6:30)=0.9+3*2.8;
wx(5:6:30)=0.9+2*2.8;
```

```
wx(6:6:30)=0.9+1*2.8;
wy(1:60)=0
wy(31:6:60)= 0.9+1*2.8;
wy(32:6:60)= 0.9+2*2.8;
wy(33:6:60)= 0.9+3*2.8;
wy(34:6:60)= 0.9+4*2.8;
wy(35:6:60)= 0.9+5*2.8;
wy(36:6:60)= 0.9+6*2.8;

A = imread('IMG_0918.JPG');
A = rgb2gray(A);
imshow(A);

[x,y]= ginput(60)
%imagecoordinates collected by ginput
ix = x;
iy =y;

n=60;
 P(1:2*n,1:12)=0;
 j=1;
 for i=1:2:120
     P(i,1)=wx(j);
     P(i,2)=wy(j);
     P(i,3)=wz(j);
     P(i,4)=1;
     P(i+1,5)=wx(j);
     P(i+1,6)=wy(j);
     P(i+1,7)=wz(j);
     P(i+1,8)=1;
     P(i,9:12)=P(i,1:4)*-1*ix(j);
     P(i+1,9:12)=P(i,1:4)*-1*iy(j);
     j=j+1;
 end
 [U,S,V]= svd(P);
 [min_val,min_index]=min(diag(S(1:12,1:12)));
 m=V(1:12,min_index);
 norm_31 = norm(m(9:11));
 m_canonical = m / norm_31;
 M(1,1:4) = m_canonical(1:4);
 M(2,1:4) = m_canonical(5:8);
 M(3,1:4) = m_canonical(9:12);
 a1 = M(1,1:3);
 a2 = M(2,1:3);
 a3 = M(3,1:3);
 b = M(1:3,4);
```

```
 r3 = a3;

 %intrinsic
 u_0 = a1*a3';
 v_0 = a2*a3';
 cross_a1a3 = cross(a1,a3);
 cross_a2a3 = cross(a2,a3);
 theta = acos (-1*cross_a1a3*cross_a2a3'/
(norm(cross_a1a3)*norm(cross_a2a3)));
 alpha = norm(cross_a1a3) * sin(theta);
 beta = norm(cross_a2a3) * sin(theta);

 %extrinsic
 r1 = cross_a2a3/norm(cross_a2a3);
 r2 = cross(r3, r1);
 K = [alpha -1*alpha*cot(theta) u_0
      0 beta/sin(theta) v_0
      0 0 1];

 t = inv(K) * b; %translation vector

 %rotation matrix
 R(1,1:3) = r1;
 R(2,1:3) = r2;
 R(3,1:3) = r3;
 %%Test of calibration estimates: reconstruction error of 4 new
points
 %image coordinates (measured)
 ix_test_measured = 1.0e+003 *[0.20750 0.22370 1.8230 1.9550];
 iy_test_measured = 1.0e+003 *[0.24230 0.24590 0.563 0.599];
 %world coordinates (measured)
 wx_test_measured = [0.9 0.9+2.8 0 0];
 wy_test_measured = [0 0 0.9 0.9+1*2.8];
 wz_test_measured = [0 0 9*2.8 9*2.8];
 %reconstruct image coordinates and calculate estimation error
 for i=1:4
    temp(1:4) = [wx_test_measured(i) wy_test_measured(i)
wz_test_measured(i) 1];
    image_reconstructed = M * temp';
    image_reconstructed_x(i) = (image_reconstructed(1)/
image_reconstructed(3));
    image_reconstructed_y(i) = (image_reconstructed(2)/
image_reconstructed(3));
    error(i) = norm([image_reconstructed_x(i)-
ix_test_measured(i) image_reconstructed_y(i)-
iy_test_measured(i)]);
```

```
 end
```

BONUS HOMEWORK :

%Matlab Code :

```
%worldcoordinates=w
wz(1:6) = 7*2.8;
wz(7:12) = 6*2.8;
wz(13:18) = 5*2.8;
wz(19:24) = 4*2.8;
wz(25:30) = 3*2.8;
wz(31:36) = 7*2.8;
wz(37:42) = 6*2.8;
wz(43:48) = 5*2.8;
wz(49:55) = 4*2.8;
wz(55:60) = 3*2.8;
wx(1:60)=0
wx(1:6:30)=0.9+6*2.8;
wx(2:6:30)=0.9+5*2.8;
wx(3:6:30)=0.9+4*2.8;
wx(4:6:30)=0.9+3*2.8;
wx(5:6:30)=0.9+2*2.8;
wx(6:6:30)=0.9+1*2.8;
wy(1:60)=0
wy(31:6:60)= 0.9+1*2.8;
wy(32:6:60)= 0.9+2*2.8;
wy(33:6:60)= 0.9+3*2.8;
wy(34:6:60)= 0.9+4*2.8;
wy(35:6:60)= 0.9+5*2.8;
wy(36:6:60)= 0.9+6*2.8;

%image coordinates
A = imread('IMG_0919.JPG');
A = rgb2gray(A);
imshow(A);
[x,y] = ginput(60);
ix = x;
iy = y;

n=60;
 P(1:2*n,1:12)=0;
 j=1;
```

```
for i=1:2:120
    P(i,1)=wx(j);
    P(i,2)=wy(j);
    P(i,3)=wz(j);
    P(i,4)=1;
    P(i+1,5)=wx(j);
    P(i+1,6)=wy(j);
    P(i+1,7)=wz(j);
    P(i+1,8)=1;
    P(i,9:12)=P(i,1:4)*-1*ix(j);
    P(i+1,9:12)=P(i,1:4)*-1*iy(j);
    j=j+1;
end
[U,S,V]= svd(P);
[min_val,min_index]=min(diag(S(1:12,1:12)));
m=V(1:12,min_index);
norm_31 = norm(m(9:11));
m_canonical = m / norm_31;
M(1,1:4) = m_canonical(1:4);
M(2,1:4) = m_canonical(5:8);
M(3,1:4) = m_canonical(9:12);
a1 = M(1,1:3);
a2 = M(2,1:3);
a3 = M(3,1:3);
b = M(1:3,4);
r3 = a3;

%intrinsic
u_0 = a1*a3';
v_0 = a2*a3';
cross_a1a3 = cross(a1,a3);
cross_a2a3 = cross(a2,a3);
theta = acos (-1*cross_a1a3*cross_a2a3'/
(norm(cross_a1a3)*norm(cross_a2a3)));
alpha = norm(cross_a1a3) * sin(theta);
beta = norm(cross_a2a3) * sin(theta);
```