EE6222 Assignment-2

Name: FENG QIYUAN	Matriculation Number: G2001613G
Name: HUANG SONG	Matriculation Number: G2001577C
Name: LU YE	Matriculation Number: G2101459F
Name: TIAN WENQIANG	Matriculation Number: G2001799H
Name: WEI ZHIFENG	Matriculation Number: G2002825F
Name: XUE ZIXIAN	Matriculation Number: G2002551F

(Names are sorted in alphabetical order.)

1. **Find the focal length f of your hand phone (in pixels). You may use real person or printed figure and include one figure of the settings in your report. Make sure you turn the camera's "zooming/auto-focusing" off.**

Solution:

Focal length is the distance from the center of projection to the image plane. In an image system, the real object will be projected to the image plane and become the image we saw in the picture. The progress is shown as Fig. 1.

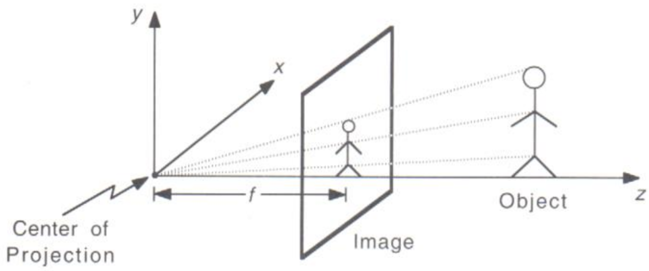


Fig. 1

The focal length in Figure 1 is denoted as f. and after projection, the real object will be scaled proportionally to the image plane. If we have the size of the image object and the distance of the center of projection to the real object, then we can calculate the focal length of this system. As shown in Fig. 2.

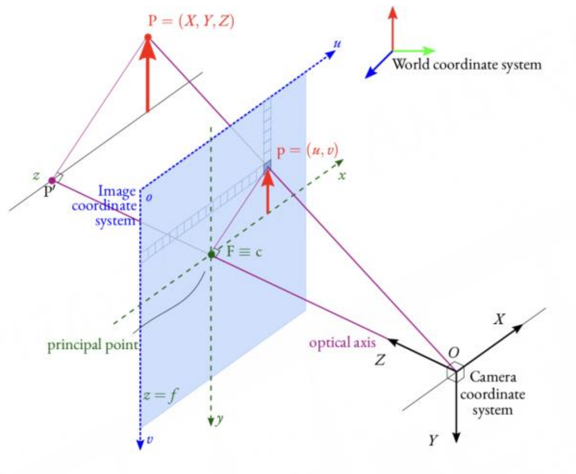


Fig. 2

In Fig. 2, the coordinate of the real object is (X, Y, Z), and the coordinates of the image object in image plane is (u, v). We assume that the focal length is f, so that we could easily deduce the relation as following:

$$\frac{u}{f} = \frac{X}{Z}$$

$$\frac{v}{f} = \frac{Y}{Z}$$

Therefore, we could get the focal length using following formula:

$$f = \frac{v * Z}{Y} = \frac{u * Z}{X}$$

In order to get the focal length of my hand phone, we first took a picture of a figure, the distance from my hand phone to the real figure is **29cm**, and the picture is shown as Fig. 3.

Fig. 3

The real line in the paper is **10cm**. Using the IrfanView application, we could get the coordinate of A and B in pixels.

$$A(525,996)$$
$$B(1645,996)$$

Therefore, the length of the image line is **1120 pixels**.

So

$$f = \frac{1120}{10} * 29 = 3248\, pixels$$

The focal length of my hand phone is **3248 pixels.**

The calculation process is shown as below:

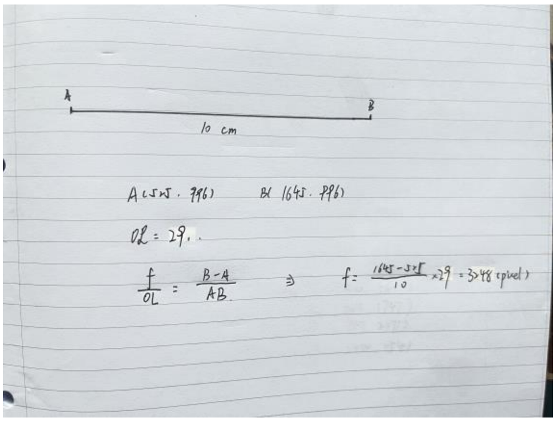


Fig. 4

2. **Take two snaps of an outdoor scene, with 5 to 10 degrees angle difference. You need to keep the angle as ground truth.**

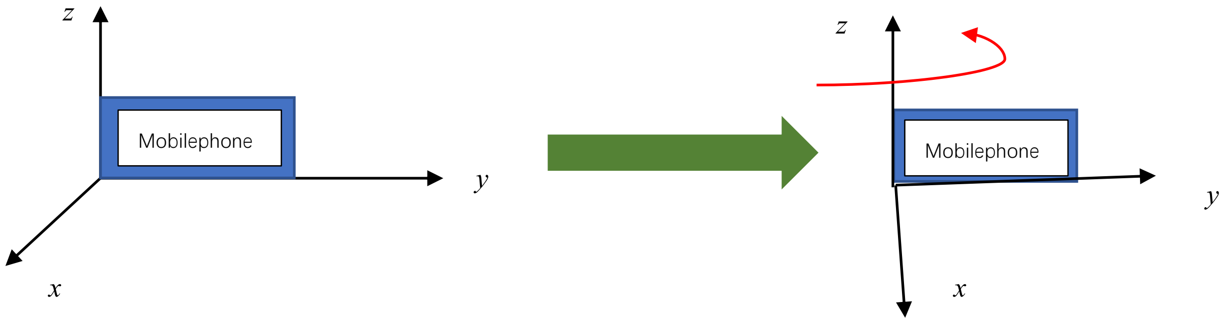
Solution:

Take the floor as the x-y plane and perpendicular to the ground as the z axis. Rotate 10 degrees counterclockwise around the z axis to get Fig.6.

Fig. 5

Fig. 6

The schematic diagram of shooting the mobile phone rotation is shown in Fig. 7.



(a) (b)

Fig. 7 (a) Position of mobile phone and coordinate axis before rotation;

(b) Position of mobile phone and coordinate axis after rotation.

3. **Hand pick 8 points or more from one image, and find the matching points on the other image. These points should not be co-planar. You need to turn these points into N-vector, and submit them into the equation for calculation.**

Solution:

We selected 8 non-coplanar points in the image taken. To ensure that the pixels of the two images match exactly, we selected easily recognizable points in the images. For example, we picked the corner of the packaging bottle and the corner of the floor tile. Next, we marked these selected points in the graph.

Fig. 8

Fig. 9

We recorded the coordinates of the points. P1, P2, P3, P4, P5, P6, P7 and P8 represent the eight points in the pictures.

In Fig. 8, we obtained 8 points' coordinates.

P1:[747,583,3284]

P2:[1900,694,3248]

P3:[2371,1841,3248]

P4:[2680,1846,3248]

P5:[907,2443,3248]

P6:[1416,2422,3248]

P7:[1992,294,3248]

P8:[3010,314,3248]

In Fig 9, we also obtained 8 points' coordinates, corresponding to Fig. 8.

P1:[21,370,3248]

P2:[1330,560,3248]
P3:[1841,1746,3248]
P4:[2113,1747,3248]
P5:[216,2412,3248]
P6:[797,2371,3248]
P7:[1433,156,3248]
P8:[2440,254,3248]

For further calculation, we converted these points into N-vectors.
In Fig. 8, we obtained 8 points' N-vectors.

P1:[0.2208,0.1723,0.9600]
P2:[0.4966,0.1814,0.8488]
P3:[0.5361,0.4163,0.7344]
P4:[0.5829,0.4015,0.7064]
P5:[0.2178,0.5867,0.7800]
P6:[0.3299,0.5643,0.7568]
P7:[0.5213,0.0768,0.8499]
P8:[0.6780,0.0707,0.7316]

In Fig. 9, we obtained 8 points' N-vectors, corresponding to Fig. 8.

P1:[0.0064,0.1132,0.9936]
P2:[0.3742,0.1576,0.9139]
P3:[0.4467,0.4236,0.7881]
P4:[0.4971,0.4110,0.7642]
P5:[0.0533,0.5953,0.8017]
P6:[0.1944,0.5784,0.7923]
P7:[0.4033,0.0439,0.9140]
P8:[0.5995,0.0625,0.7980]

4. **Calculate the rotation angle from the matched points using the quaternion approach (pp 14 in [4]), or the SVD(in [3]).**

Solution:

A. Quaternion Approach

Definition of quaternion:

In mathematics, the quaternion number system extends the complex numbers. Quaternions are generally represented in the form below:

$$q = q_0 + q_1 i + q_2 j + q_3 k$$
$$\|q\|^2 = q_0^2 + q_1^2 + q_2^2 + q_3^2$$

Where q_0, q_1, q_2, q_3 are real numbers, and i, j, k are symbols that can be interpreted as unit-vectors pointing along the three spatial axes.

Experiments on quaternion method:

According to quaternion matrix:

$$K = \begin{bmatrix} K11+K22+K33 & K32-K23 & K13-K31 & K21-K12 \\ K32-K23 & K11-K22-K33 & K12+K21 & K31+K13 \\ K13-K31 & K12+K21 & -K11+K22-K33 & K23+K32 \\ K21-K12 & K31+K13 & K23+K32 & -K11-K22+K33 \end{bmatrix}$$

where k_{ij} represents the corresponding rotation matrix R_{ij} calculated by $A*B'$.

The equation of calculating correlation matrix K is shown below:

$$\mathrm{K} = \sum_{\mathrm{i}}^{\mathrm{N}} \mathrm{W_i m_i m_i'^T}$$

The m_i and m_i' corresponding to the matching points before rotation and matching points after rotation, then centralize each point.The coefficients W_i is set to be 1 in this experiment.

We first do the eigen decomposition of K matrix and sort the eigen values and corresponding eigen vectors. The close form solution can be maximized if we choose the largest eigenvector **q** of K for the largest eigenvalue:

$$\max : trace(R^T K)$$

$$q = (s, v)$$

In above formula, q and v represent vectors and s is defined as:

$$s = \cos(\frac{\theta}{2})$$

So the rotation angle can be solved by:

$$\theta = 2 * \arccos(s)$$

In the assignment, the K matrix is:

$$\begin{bmatrix} 0.6402 & 0.0031 & 0.0406 & -0.0132 \\ 0.0031 & -0.1291 & -0.2390 & -0.1124 \\ 0.0406 & -0.2390 & 0.0326 & -0.1429 \\ -0.0132 & -0.1124 & -0.1429 & -0.5436 \end{bmatrix}$$

The largest eigen vector for largest eigen value of K is:

$$\begin{bmatrix} 0.9967 & -0.0172 & 0.0774 & -0.0188 \end{bmatrix}^T$$

The rotation angle is:

$$\theta_{quaternion} = \mathbf{9.349627569055862(degree)}$$

B. *SVD Approach*

For the 8 pairs of matched points, we can represent them by matrices X and P.

$$X = [x_1, x_2, \dots, x_8]$$
$$P = [p_1, p_2, \dots, p_8]$$

where x_i and p_i are N-vectors of matched points.

To ignore the translation, we shift points to their centers of mass, which are μ_x and μ_p.

$$\mu_x = \frac{1}{N_x}\sum_{i=1}^{N_x} x_i$$
$$\mu_p = \frac{1}{N_p}\sum_{i=1}^{N_p} p_i$$

And

$$X' = [x_1 - \mu_x, x_2 - \mu_x, \dots, x_8 - \mu_x]$$
$$P' = [p_1 - \mu_p, p_2 - \mu_p, \dots, p_8 - \mu_p]$$

The results are:

$$X' = \begin{bmatrix} -0.22713 & 0.048637 & 0.088179 & 0.13497 & -0.23011 & -0.118 & 0.073342 & 0.2301 \\ -0.13645 & -0.12739 & 0.1075 & 0.092736 & 0.27791 & 0.25555 & -0.23183 & -0.23803 \\ 0.16399 & 0.052851 & -0.061604 & -0.089567 & -0.016009 & -0.039225 & 0.05393 & -0.064363 \end{bmatrix}$$

$$P' = \begin{bmatrix} -0.31544 & 0.05235 & 0.12481 & 0.17526 & -0.26855 & -0.12745 & 0.081404 & 0.2776 \\ -0.18499 & -0.14061 & 0.12545 & 0.11284 & 0.29717 & 0.28018 & -0.25427 & -0.23577 \\ 0.14785 & 0.068161 & -0.057649 & -0.081544 & -0.044003 & -0.053418 & 0.06833 & -0.047731 \end{bmatrix}$$

The calculation of matrix W is denoted by:

$$W = \sum_{i=1}^{N_p} x_i' \, p_i'^T$$

where N_p is the number of pairs.

We picked 8 pairs of points from the two images, so $N_p = 8$. And W is calculated as

$$W = X' * P'^T = \begin{bmatrix} 0.25553 & -0.11287 & -0.0359 \\ -0.12611 & 0.33636 & -0.072976 \\ -0.076526 & -0.069888 & 0.04826 \end{bmatrix}$$

The singular value decomposition (SVD) of W is denoted by:

$$W = U \begin{bmatrix} 0.42438 & 0 & 0 \\ 0 & 0.21599 & 0 \\ 0 & 0 & 0.0031314 \end{bmatrix} V^T$$

And the matrices U and V are denoted as:

$$U = \begin{bmatrix} -0.55252 & -0.72234 & 0.41586 \\ 0.83242 & -0.45291 & 0.31928 \\ -0.042289 & 0.52258 & 0.85154 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.57242 & -0.7753 & 0.26692 \\ 0.81369 & -0.49692 & 0.30163 \\ -0.10121 & 0.38985 & 0.9153 \end{bmatrix}$$

We can see that rank(W)=3, then, the rotation matrix R is unique and is given by:

$$R = UV^T = \begin{bmatrix} 0.98731 & 0.034802 & 0.15495 \\ -0.040135 & 0.9987 & 0.031426 \\ -0.15366 & -0.037246 & 0.98742 \end{bmatrix}$$

The rotation angle can be computed by:

$$\Omega = \arccos \frac{\mathrm{trace}\mathbf{R} - 1}{2}$$

$$\mathrm{trace}\mathbf{R} = 0.98731 + 0.9987 + 0.98742 = 2.9734$$

$$\Omega = \arccos\left(\frac{2.9734 - 1}{2}\right) = 0.16318 = 9.3496°$$

Therefore, the rotation angle calculated by SVD method is 9.3496°, which is close to the groundtruth of 10°.

From above data, we can conclude:

1. Quaternion method has no singularity, which can represent any rotation relationship. The calculation is simple, but not straight forward.
2. SVD result has little difference from quaternion result, but the expression of SVD is more intuitive. The rotation matrix R must be nonsingular when SVD decomposition is performed.

References:

[1] Wang Han, “Machine Vision,” Course Slides of EE6222, School of EEE, Nanyang Technological University, Singapore, 2021.

Codes:

```
clc
clear
% 10 Degree
% normalized matrix A
f = 112*29;
A1 = [747,583,f]; NA1 = A1./norm(A1);
A2 = [1900,694,f]; NA2 = A2./norm(A2);
A3 = [2371,1841,f]; NA3 = A3./norm(A3);
A4 = [2680,1846,f]; NA4 = A4./norm(A4);
A5 = [907,2443,f]; NA5 = A5./norm(A5);
A6 = [1416,2422,f]; NA6 = A6./norm(A6);
A7 = [1992,294,f]; NA7 = A7./norm(A7);
A8 = [3010,314,f]; NA8 = A8./norm(A8);
```

```
% normalized matrix B
B1 = [21,370,f]; NB1 = B1./norm(B1);
B2 = [1330,560,f]; NB2 = B2./norm(B2);
B3 = [1841,1746,f]; NB3 = B3./norm(B3);
B4 = [2113,1747,f]; NB4 = B4./norm(B4);
B5 = [216,2412,f]; NB5 = B5./norm(B5);
B6 = [797,2371,f]; NB6 = B6./norm(B6);
B7 = [1433,156,f]; NB7 = B7./norm(B7);
B8 = [2440,254,f]; NB8 = B8./norm(B8);

%calculate the mean of two image matrixs
A=[[NA1(1),NA2(1),NA3(1),NA4(1),NA5(1),NA6(1),NA7(1),NA8(1)];
[NA1(2),NA2(2),NA3(2),NA4(2),NA5(2),NA6(2),NA7(2),NA8(2)];
[NA1(3),NA2(3),NA3(3),NA4(3),NA5(3),NA6(3),NA7(3),NA8(3)]];

B=[[NB1(1),NB2(1),NB3(1),NB4(1),NB5(1),NB6(1),NB7(1),NB8(1)];
[NB1(2),NB2(2),NB3(2),NB4(2),NB5(2),NB6(2),NB7(2),NB8(2)];
[NB1(3),NB2(3),NB3(3),NB4(3),NB5(3),NB6(3),NB7(3),NB8(3)]];

% Shift to center
mean_A = mean(A');
mean_B = mean(B');

A(1,:) = A(1,:)-mean_A(1);
A(2,:) = A(2,:)-mean_A(2);
A(3,:) = A(3,:)-mean_A(3);
B(1,:) = B(1,:)-mean_B(1);
B(2,:) = B(2,:)-mean_B(2);
B(3,:) = B(3,:)-mean_B(3);

% calculate SVD
W = A*B';
[U,S,V] = svd(W);
R = U*V';
thi = acos((trace(R)-1)/2);
rad2deg(thi)

% quaternion
K=[[W(1,1)+W(2,2)+W(3,3),W(3,2)-W(2,3),W(1,3)-W(3,1),W(2,1)-W(1,2)];
    [W(3,2)-W(2,3),W(1,1)-W(2,2)-W(3,3),W(1,2)+W(2,1),W(3,1)+W(1,3)];
    [W(1,3)-W(3,1),W(1,2)+W(2,1),-W(1,1)+W(2,2)-W(3,3),W(2,3)+W(3,2)];
    [W(2,1)-W(1,2),W(3,1)+W(1,3),W(2,3)+W(3,2),-W(1,1)-
W(2,2)+W(3,3)]];
```

```
[m,v]=eig(K);
ev=diag(v);
[absv,abse]=sort(ev,'descend');
eigen_ve=m(:,abse);
thi_quternion=rad2deg(2*acos(eigen_ve(1,1)))
```