Computer vision (**CSC I6716**)

Assignment 3

Prof. Zhigang Zhu

Sonali Shintre

ID: 7602

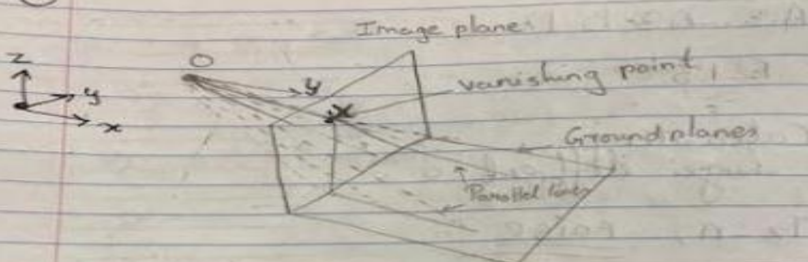
11/10/19

1. **(Camera Models- 30 points) Prove that the vector from the viewpoint of a pinhole camera to the vanishing point (in the image plane) of a set of 3D parallel lines is parallel to the direction of the parallel lines. Please show steps of your proof.**

In pinhole camera the focal length varies by given ratio of the image dimension to the distance of the pinhole from the image Due to perspective, all parallel lines in 3D space appear to meet in a point on the image - the vanishing point, which is the common intersection of all the image lines

.

1



Two parallel lines intersect the line is parallel to the parallel lines.
The vector x is also parallel to the parallel lines since vanishing point and O are on intersection line. so vector $\vec{ox}$ is Parrallel.

Prove:- Vanishing point v and the O (view point) should be on interpretation plane. So the vector $\vec{ox}$ should be also on interpretation plane. so the vector & parallel lines can't be a skew line.
If not parallel then it will have intersection point with the ground plane. Point should also parallel dotted lines. hence parallel lines form a intersection point.

If you select to use algebraic calculation, you may use the parametric representation of a 3D line: **P = P0 +tV, where P= $(X,Y,Z)^T$ is any point on the line (here ^{T} denote for transpose), P0 = $(X0,Y0,Z0)^T$ is a given fixed point on the line, vector V = $(a,b,c)^T$ represents the direction of the line, and t is the scalar parameter that controls the distance (with sign) between P and P0**.

Algebraic calculation:-

3D line : $P = P_0 + tV$

where $P = (x\ y\ z)^T$

$P_0 = (X_0\ Y_0\ Z_0)^T$

vector $V = (a, b, c)^T$

t = Scalor Parameter.

Using $x = f(X/Z)$ the vanishing point of its image.

so here $X = P$.

$$\lim_{t \to \pm\infty} P = f \frac{P_0 + tV}{P_z + tV_z}$$

$$= f \frac{P}{P_z} = f \begin{pmatrix} P_x/P_z \\ P_y/P_z \\ 1 \end{pmatrix}$$

So all the vanishing point meet ~~at~~ the parallel lines.

2. **(Camera Models- 20 points) Show that relation between any image point $(xim, yim)^T$ of a plane (in the form of $(x1,x2,x3)^T$ in projective space) and its corresponding point $(Xw, Yw, Zw)^T$ on the plane in 3D space can be represented by a 3×3 matrix. You should start from the general form of the camera model $(x1,x2,x3)^T = M_{int}M_{ext}(Xw, Yw, Zw, 1)^T$, where the image center (ox, oy), the focal length f, the scaling factors(sx and sy), the rotation matrix R and the translation vector T are all unknown. Note that in the course slides and the lecture notes, I used a simplified model of the perspective project by assuming ox and oy are known and sx = sy =1, and only discussed the special cases of planes. So you cannot directly copy those equations I used. Instead you should use the general form of the projective matrix, and the general form of a plane $n_x X_w + n_y Y_w + n_z Z_w = d$.**

(2)

Plane eqn:-

$$n_x X_\omega + n_y Y_\omega + n_z Z_\omega = d$$

$$= n^T P_\omega = d$$

P_ω is a 3D point on the plane = $(x_\omega \; y_\omega \; z_\omega)^T$
z_ω can be written as a function of x_ω & y_ω
$\therefore Z_\omega = 0$

$$\therefore P_\omega = (X_\omega \; Y_\omega \; 0 \; 1)^T$$

3D point $(x_\omega, y_\omega, z_\omega)$ becomes a 2D point (x_ω, y_ω)

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{bmatrix} -fr_{11} & -fr_{12} & -fT_x \\ -fr_{12} & -fr_{22} & -fT_y \\ r_{31} & r_{32} & T_z \end{bmatrix} \begin{pmatrix} X_\omega \\ Y_\omega \\ 1 \end{pmatrix}$$

$nz = 1$ so plane eqn will be

$$Z_\omega = d - n_x X_\omega - n_y Y_\omega$$

$$M = \begin{bmatrix} -fr_{11} & -fr_{12} & -fr_{13} & -fT_x \\ -fr_{21} & -fr_{22} & -fr_{23} & -fT_y \\ r_{31} & r_{32} & r_{33} & T_z \end{bmatrix}$$

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{bmatrix} -f(r_{11} - n_x r_{13}) & -f(r_{12} - n_y r_{13}) & -f(dr_{13} + T_x) \\ -f(r_{21} - n_x r_{23}) & -f(r_{22} - n_y r_{23}) & -f(dr_{23} + T_y) \\ (r_{31} - n_x r_{33}) & (r_{32} - n_y r_{33}) & (dr_{33} + T_z) \end{bmatrix} \begin{pmatrix} X_\omega \\ Y_\omega \\ 1 \end{pmatrix}$$

$$M_{ext} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \end{bmatrix} = \begin{bmatrix} R_1^T & T_x \\ R_2^T & T_y \\ R_3^T & T_z \end{bmatrix}$$

M_{ext} = includes all extrinsic parameters
M_{int} = all intrinsic parameters

$$M_{in} = \begin{bmatrix} -f_x & 0 & O_x \\ 0 & -f_y & O_y \\ 0 & 0 & 1 \end{bmatrix}$$

Combine M_{ext} and M_{int} in eq 1

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = M_{int} M_{ext} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$

Each side divide by x_3

$$\begin{pmatrix} x_{im} \\ y_{im} \end{pmatrix} = \begin{pmatrix} x_1/x_3 \\ x_2/x_3 \end{pmatrix}$$

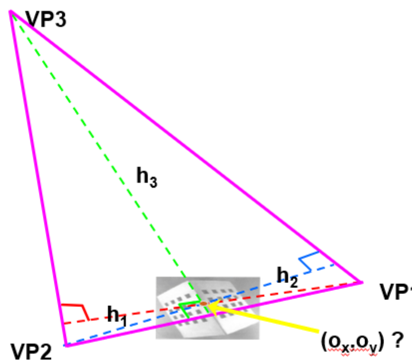
3. (Calibration- 20 points) Prove the Orthocenter Theorem by geometric arguments: Let T be the triangle on the image plane defined by the three vanishing points of three mutually orthogonal sets of parallel lines in space. Then the image center is the orthocenter of the triangle T (i.e., the common intersection of the three altitudes.

(1) Basic proof: use the result of Question 1, assuming the aspect ratio of the camera is 1. (10 points)

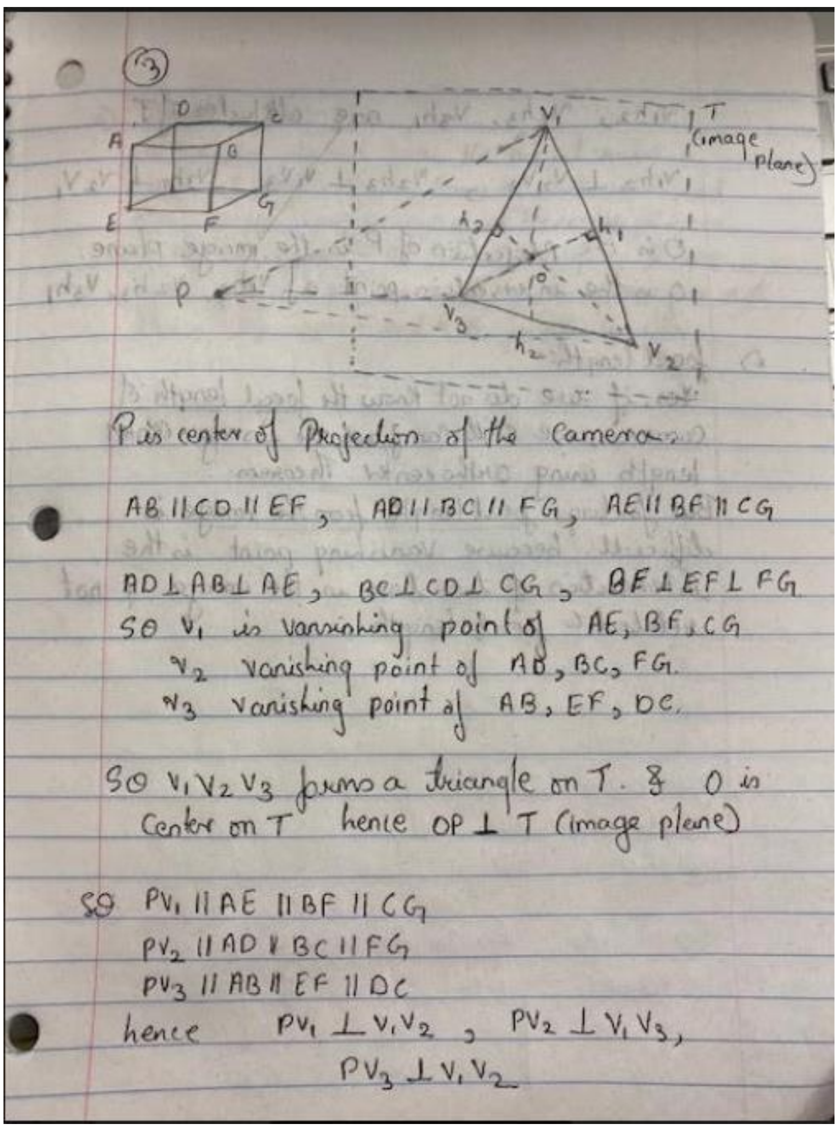
(2) If you do not know the focal length of the camera, can you still find the image center (together with the focal length) using the Orthocenter Theorem? Show why or why not. (5 points)

(3) If you do not know the aspect ratio and the focal length of the camera, can you still find the image center using the Orthocenter Theorem? Show why or why not. (5 points)

The center of projection of the camera in 3D space is O (ox, oy). Three mutually orthogonal sets of parallel lines: L1, L2, and L3. Assume there is a triangle formed by lines V1V2V3, and then they are also V1, V2, and V3 are the three vanishing points.



i. proof:



P is center of Projection of the Camera.

$AB \parallel CD \parallel EF$, $AD \parallel BC \parallel FG$, $AE \parallel BF \parallel CG$

$AD \perp AB \perp AE$, $BC \perp CD \perp CG$, $BF \perp EF \perp FG$

So V_1 is vanishing point of AE, BF, CG

V_2 vanishing point of AD, BC, FG.

V_3 vanishing point of AB, EF, DC.

So $V_1V_2V_3$ forms a triangle on T. & O is Center on T hence $OP \perp T$ (image plane)

So $PV_1 \parallel AE \parallel BF \parallel CG$

$PV_2 \parallel AD \parallel BC \parallel FG$

$PV_3 \parallel AB \parallel EF \parallel DC$

hence $PV_1 \perp V_1V_2$, $PV_2 \perp V_1V_3$,

$PV_3 \perp V_1V_2$

v_1h_2, v_2h_3, v_3h_1 are altitudes of T

$v_1h_2 \perp v_2v_3$, $v_2h_3 \perp v_1v_3$ $v_3h_1 \perp v_2v_1$

O is the projection of P in the image plane.
O in the intersection point of v_1h_2, v_2h_3, v_3h_1

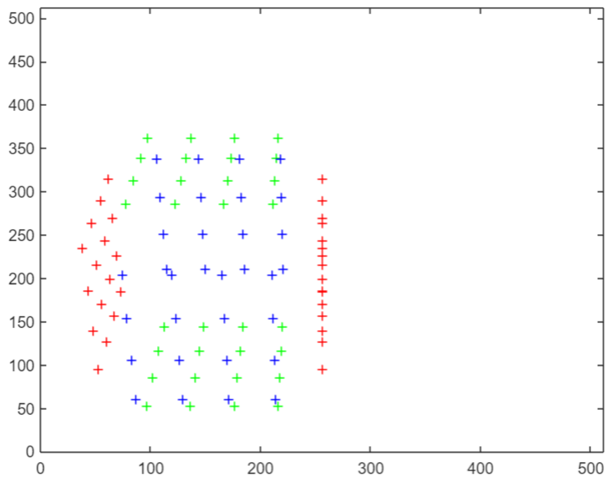
focal length:-
Yes, if we do not know the focal length of camera, we still can find the image center length using orthocenter. Theorem
But finding focal length from the image is difficult. because vanishing point is the intersection of two lines in the image & not related to focal length

ii. Focal Length:
Yes, if we do not know the focal length of the camera we can still find the image center length using orthocenter theorem, But finding focal length from the image is difficult, since vanishing point is intersection of the two lines in the image and not related to focal length

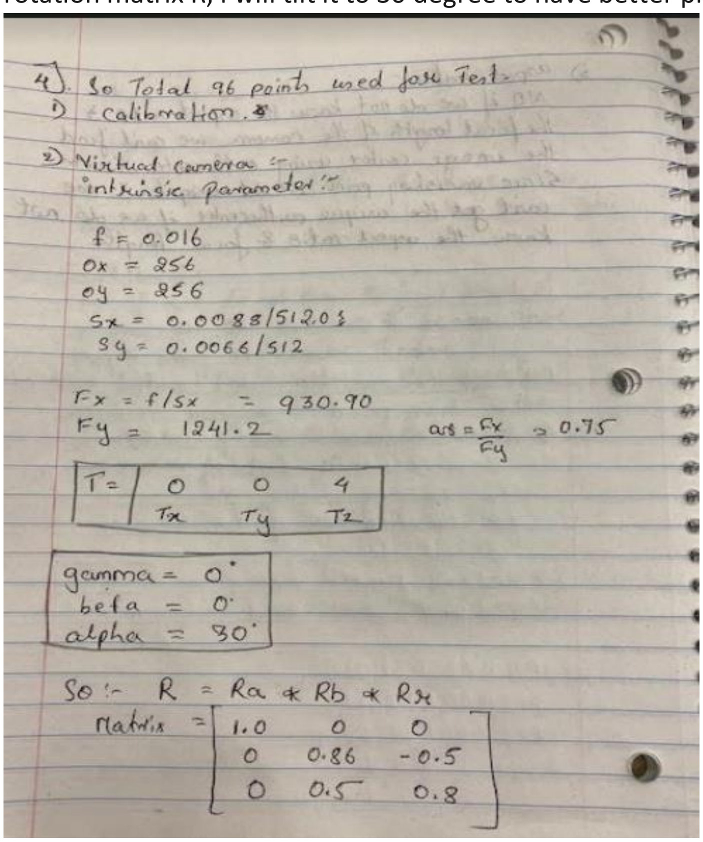
iii. Aspect Ratio:
No if we do not know the aspect ratio and the focal length of the camera, we cannot find the image center using orthocenter therom, since vanishing point is not uniqur and we can't get the unique orthocenter if we do not know the aspect ratio and focal length

4. **Calibration Programming Exercises (40 points): Implement the direct parameter calibration method in order to (1) learn how to use SVD to solve systems of linear equations; (2) understand the physical constraints of the camera parameters; and (3) understand important issues related to calibration, such as calibration pattern design, point localization accuracy and robustness of the algorithms. Since calibrating a real camera involves lots of work in calibration pattern design, image processing and error controls as well as solving the equations, we will mainly use simulated data to understand the algorithms. As a by-product we will also learn how to generate 2D images from 3D models using a "virtual" pinhole camera.**

1. Generated a virtual 3D cube. as a result, a graph is generated and, on the graph, at each surface there are 4x4 points. I need to use those points as variables for an equation.
   Xw=0, Yw=0, Xw=1, Yw=1, Zw=1, Zw=0. so here Pw is a 1x1x1 m3 cube, 96 points are generated and each surface have 16 points.
   Total 96 points for calibration



2. create virtual camera with known intrinsic parameters given focal length f=16mm, image center (o_x, o_y) = (256, 256), and pixel size (s_x, s_y). Using three rotation angles alpha, beta, gamma to generate the rotation matrix R, I will tilt it to 30 degree to have better projection.



4). So Total 96 points used for Test

1) Calibration.

2) Virtual Camera :-

intrinsic Parameter :-

f = 0.016

Ox = 256

oy = 256

Sx = 0.0088/512.03

Sy = 0.0066/512

Fx = f/sx = 930.90

Fy = 1241.2

ars = Fx / Fy = 0.75

T =	0	0	4
	Tx	Ty	Tz

gamma = 0°

beta = 0°

alpha = 30°

So :- R = Ra * Rb * Rr

Matrix =
$$\begin{bmatrix} 1.0 & 0 & 0 \\ 0 & 0.86 & -0.5 \\ 0 & 0.5 & 0.8 \end{bmatrix}$$

3. Estimating the intrinsic (f_x, f_y, aspect ratio a, image center (o_x,o_y)) and extrinsic (R, T and further alpha, beta, gamma)

   I. first, we know the image center Ox = 256; Oy = 256, every point in 3D have corresponding point image, first we find the matrix A where $A=UDV^T$ If the matrix A is a real matrix, then U and V are also real, Calculate Tz, Fx, Fy, forming 8 cols and 96 rows, which 8 unknows $v=(v1.....v8)^T$ 7 of them are independent parameters and 8th row is zero.

$$x_iX_iv_1 + x_iY_iv_2 + x_iZ_iv_3 + x_iv_4 - y_iX_iv_5 - y_iY_iv_6 - y_iZ_iv_7 - y_iv_8 = 0$$

$$(v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8) = (r_{21}, r_{22}, r_{23}, T_y, \alpha r_{11}, \alpha r_{12}, \alpha r_{13}, \alpha T_x)$$

Av=0

3) Ox = 256   Oy = 256

i) eq:- $x_iX_iv_1 + x_iY_iv_2 + \dots$
with 96 data points we have 8 unknows ~~row~~ cols and 96 rows.

eq:-

$$x_iX_iv_1 + x_iY_iv_2 + x_iZ_iv_3 + x_iv_4 - y_iX_iv_5 - y_iY_iv_6 - y_iZ_iv_7 - y_iv_8 = 0$$

Av = 0.

A matrix [96 x 8]

find v = [U, S, V] = svd(A);
(8 x 8)

v = [1 x 8]

$$\alpha = \sqrt{\bar{v}_5^2 + \bar{v}_6^2 + \bar{v}_7^2}$$

$$\alpha = \sqrt{\bar{v}_5^2 + \bar{v}_6^2 + \bar{v}_7^2} \Big/ \sqrt{\bar{v}_1^2 + \bar{v}_2^2 + \bar{v}_3^2}$$

α = 4.62 / 2.88 = 1.6

A matrix =

-217.8480	-43.5696	-43.5696	-217.8480	21.2634	4.2527	4.2527	21.2634
-209.3619	-41.8724	-83.7448	-209.3619	-7.4798	-1.4960	-2.9919	-7.4798
-201.5123	-40.3025	-120.9074	-201.5123	-34.0676	-6.8135	-20.4406	-34.0676
-194.2299	-38.8460	-155.3839	-194.2299	-58.7338	-11.7468	-46.9870	-58.7338
-212.8666	-85.1466	-42.5733	-212.8666	69.9366	27.9747	13.9873	69.9366
-204.7570	-81.9028	-81.9028	-204.7570	39.9713	15.9885	15.9885	39.9713
-197.2426	-78.8970	-118.3455	-197.2426	12.2054	4.8822	7.3232	12.2054
-190.2602	-76.1041	-152.2082	-190.2602	-13.5947	-5.4379	-10.8757	-13.5947
-208.1078	-124.8647	-41.6216	-208.1078	116.4336	69.8602	23.2867	116.4336

……….

SVD: [U,V,S]=svd(A)

V:

0.4618	-0.3469	0.3377	-0.0749	0.3104	-0.0129	0.6710	-0.0000
0.3374	0.0467	-0.0292	-0.4083	-0.4777	0.0892	-0.0163	0.6928
0.2443	-0.3095	-0.6967	-0.2771	-0.0977	0.3332	0.0432	-0.4000
0.5662	-0.3648	0.1026	0.3206	0.0738	-0.1695	-0.6316	0.0000
-0.2268	-0.2603	-0.4307	0.2868	0.4865	0.1192	0.0476	0.6000
-0.3162	-0.3905	0.4293	-0.4108	0.1704	0.5159	-0.3152	0.0000
-0.0990	-0.2486	0.1312	0.6121	-0.5509	0.4369	0.2052	0.0000
-0.3626	-0.6055	-0.0470	-0.1497	-0.3021	-0.6172	0.0714	-0.0000

v= [-0.0000   0.6928   -0.4000   0.0000   0.6000   0.0000   0.0000   -0.0000]

$$|\gamma| = \sqrt{\bar{v}_1^2 + \bar{v}_2^2 + \bar{v}_3^2}$$

$$\alpha = \sqrt{\bar{v}_5^2 + \bar{v}_6^2 + \bar{v}_7^2} \,/\, \sqrt{\bar{v}_1^2 + \bar{v}_2^2 + \bar{v}_3^2}$$

Calculating Fx, Fy

Calculating aspect

aspect ratio.	ground true	Estimated.	dif
ar	0.75	0.7544	0.00[illegible]

Tz, Fx, Fy -

Fx, Fy

	Ground True	Estimated	dif
Fx	930.9	930.9	0
Fy	1241.2	1241.2	0

Calculate R & T

$R_3^T = R_1^T \times R_2^T = sR_1^T \times sR_2^T$

Comparing ground True & Estimated R is identical to the virtual camera.

Estimated R_C

$$\begin{matrix} -1.0 & -0.0 & -0.0 \\ 0.0 & -0.8660 & 0.5000 \\ -0.0 & 0.50 & 0.8660 \end{matrix}$$

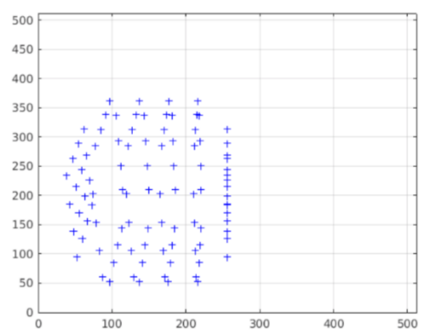
Calculate T = [0, 0 4]

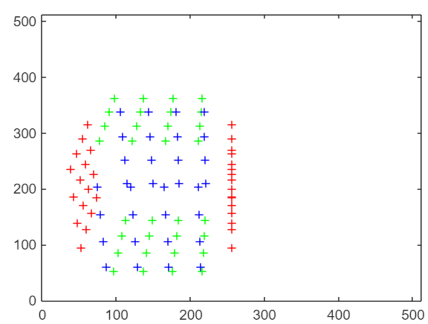
	Ground True	Estimated values	diff
Tx	0	6.7009e-04	6.7009e-04
Ty	0	1.8468e-16	1.8468e-16
Tz	4	4.	4.

ii) After calculating Tx, Ty, Tz, aspect ratio, fx, fy from the ground True & Estimated. we see that only Tx Ty influence the image center.
so we should have fixed image center when using direct parameter calibration.

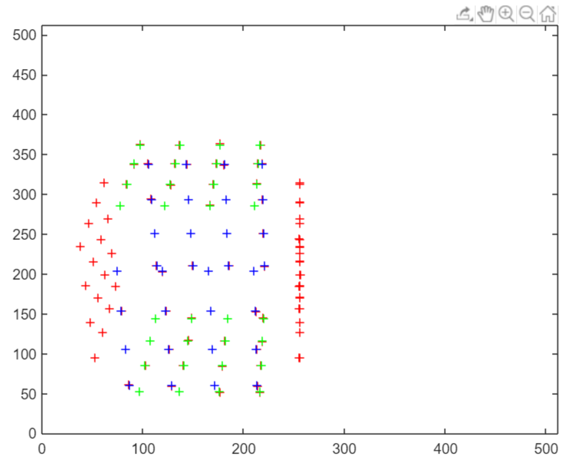
ii. If the ratio of an image is different than the ratio of your screen, we may not see the whole image. Images won't fit properly if the screen is narrower than the image. We know the first 2 rows of R and first two components from the T. By this 3rd row will calculate from the R1 and R2. Next is to Determine the sign s positive or negative, assuming s as positive, test sign of s use the image point. We got X = -1 and x = 1 so s is negative in this case.

if we do not know the aspect ratio of the camera, we can't find the image center using Orthocenter Theorem. Because the vanishing point is not unique, and we can't get the unique orthocenter if we do not know the aspect ratio.





iii. Adding some noise, 0.1 mm random error to 3D points and 0.5pixel random error to 2D points. Once noise is added we can Then we can constraints by using SVD.



iii> Random noise:-

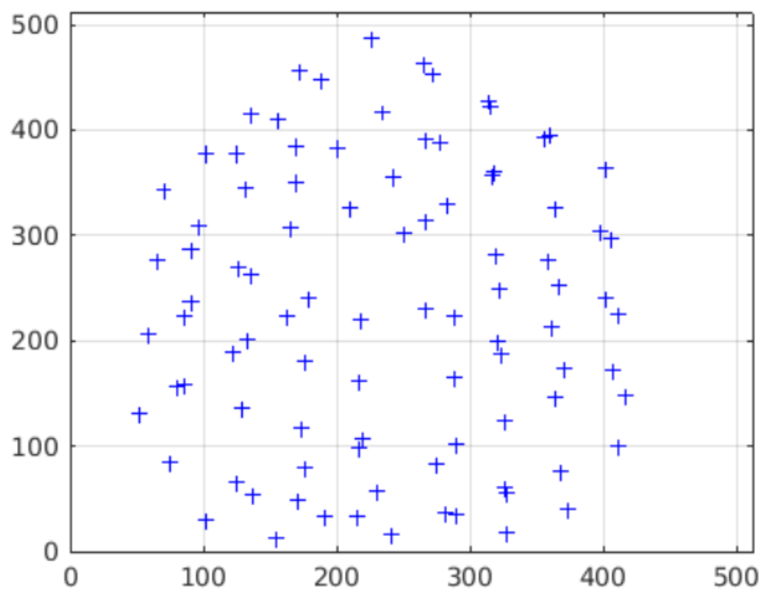
After adding 0.1mm random error to 3D points & 0.5 pixels random error to 2D points.

	Ground True	noise add	diff
aspect raito	0.75	0.7491	0.0009
Fx	930.9	926.9095	3.9905
Fy	1241.2	1242.7	1.5
Tx	6.7009e-0.4	-2.6265e-0.4	0.00093
Ty	1.8468e-16	-6.5654e-04	0.00055
Tz	4.	3.9453	0.05

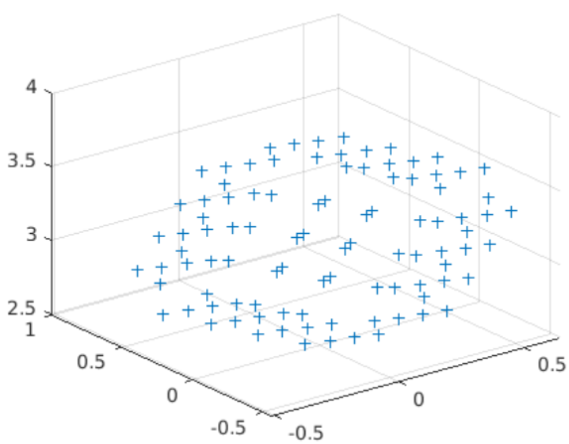
As the result aspect raito, Fx, Fy, TY, Tz, Tx are all changed by adding noise.

We can conclude that Orthocenter method is very sensitive to the extrinsic Parameter.

WE can conclude that orthocenter method is sensitive to the extrinsic parameter.



500
400
300
200
100
0
0
100
200
300
400
500



4
3.5
3
2.5
1
0.5
0
-0.5
-0.5
0
0.5