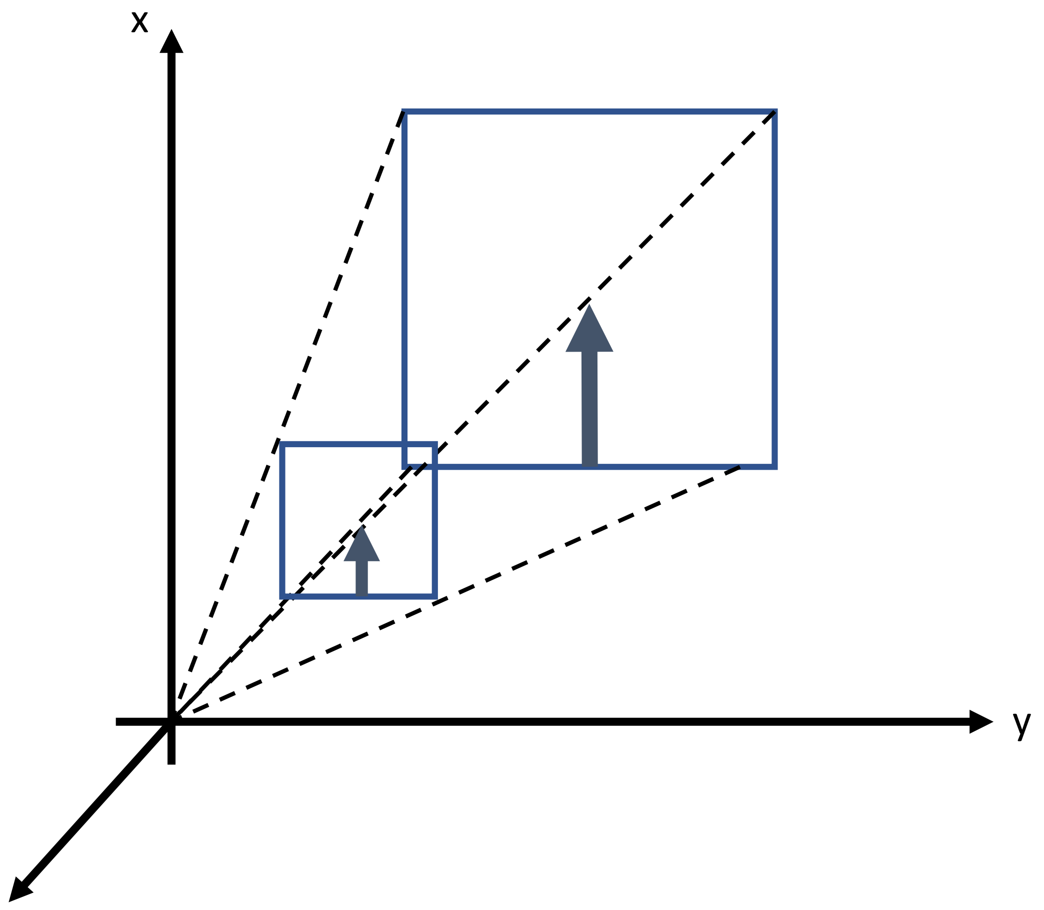
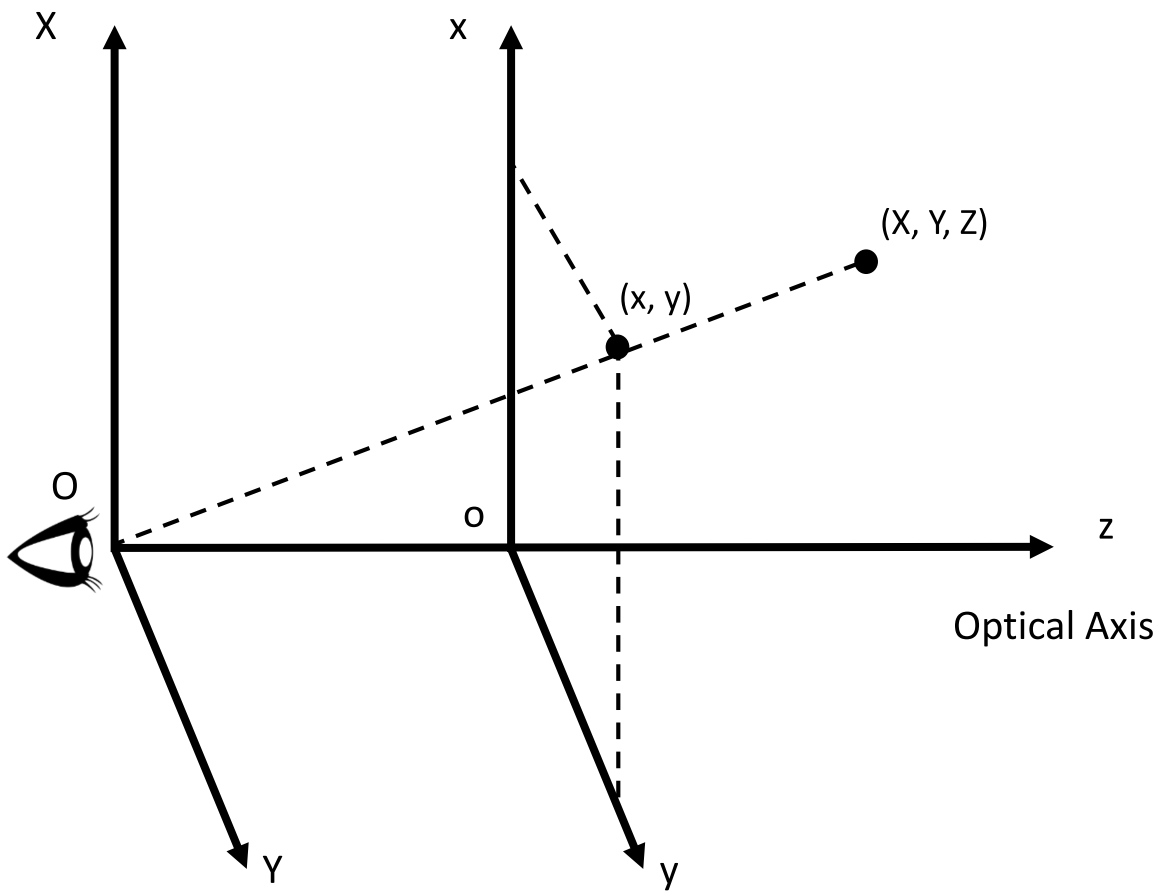
Projective Geometry

Textbook:

Geometric Computation for Machine Vision - K. Kanatani, Oxford Press

Projective Geometry

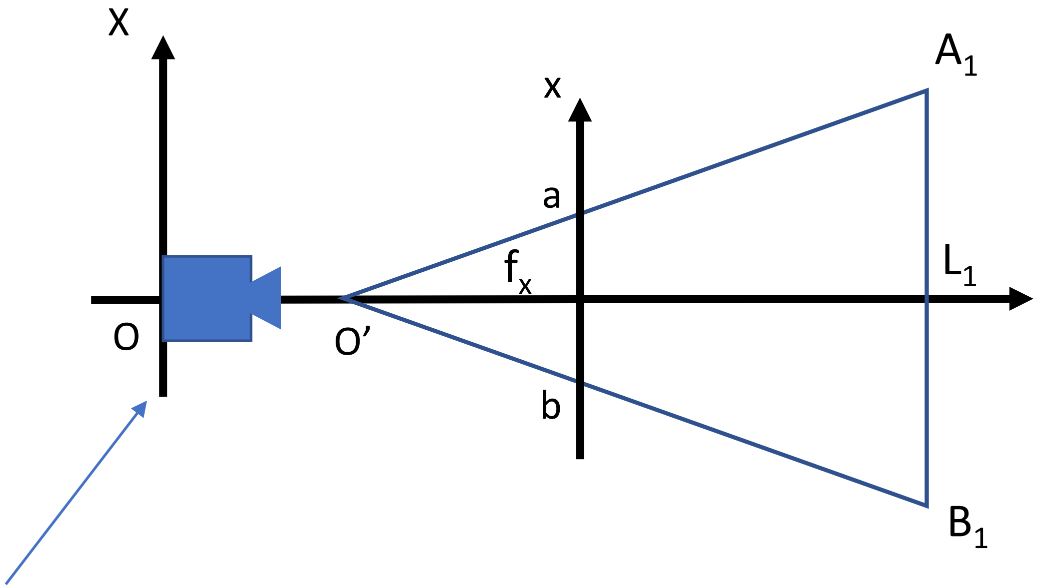




Projective Geometry

Camera Parameters

- f_x, f_y – focal length in x,y directions
- x_o, y_o – image center (found by calibration)
- f_x/f_y – aspect ratio



Back side of hand phone

How to find f_x?

1. Point the camera to a wall
2. Mark two points on the wall (A_1 & B_1)
3. Measure A_1B_1 and OL_1 using a tape in mm
4. Read image points in pixels

$$a = (x_a, y_a)$$
$$b = (x_b, y_b)$$

5. Relocate the camera and obtain

$$A_2B_2\,, OL_2$$
$$a_2 = (x'_a, y'_a)$$
$$b_2 = (x'_b, y'_b)$$

two images

6. Solve

$$\frac{f_x}{O'L_1} = \frac{x_a - x_b}{A_1B_1} \qquad \left\{ \begin{array}{l} \dfrac{f_x}{OL_1 - OO'} = \dfrac{x_a - x_b}{A_1B_1} \\[2ex] \dfrac{f_x}{OL_2 - OO'} = \dfrac{x'_a - x'_b}{A_2B_2} \end{array} \right.$$

Hence, given an image point (row, column)

For simplicity, we assume $f_x = f_y = f$

$$\begin{array}{ll} x = -(row - x_0) & in\ pixels \\ y = column - y_0 & in\ pixels \\ f & in\ pixels \end{array}$$

N-vector of a Point

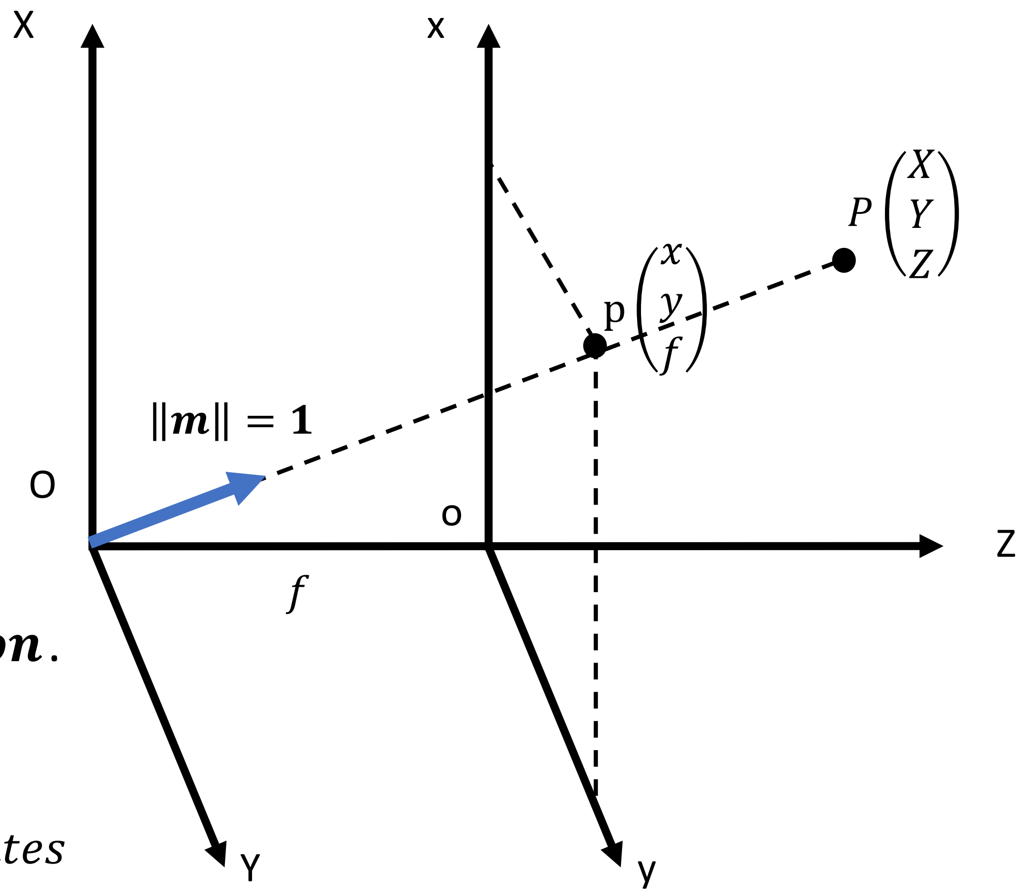
A point in the world $P(X, Y, Z)$ will be projected to the image plane $p(x, y)$ through a simple linear relation:

$$\frac{X}{Z} = \frac{x}{f} \quad , \qquad \frac{Y}{Z} = \frac{y}{f}$$

Hence $\quad (x, y) = (f\frac{X}{Z}, f\frac{Y}{Z})$

where $\boldsymbol{f}$ is a constant called $\boldsymbol{focal\ length}$.
This equation is also referred to as the $\boldsymbol{perspective\ projection}$.

$image\ plane$: $2D$ projective space
(x, y, f) : $homogeneous\ coordinates$
(x, y) : $image\ coordinates\ or\ inhomogeneous\ coordinates$



Multiply a non-zero number to homogeneous coordinates, the resulting vector represent the same point; hence, we can define a unit vector by normalizing $u = (u_x, u_y, f)$

$$N[u] = \frac{u}{||u||} = \frac{u}{\sqrt{u_x^2 + u_y^2 + f^2}}$$

A point (a, b) in the image plane can be

$$\boldsymbol{m} = \pm N[\begin{pmatrix} a \\ b \\ f \end{pmatrix}]$$

Note that $\pm\boldsymbol{m}$ represent the same point (by definition). $\boldsymbol{m}$ is also called the $\boldsymbol{N-vector}$

N-vector of a Line

Define a line l on the image plane.

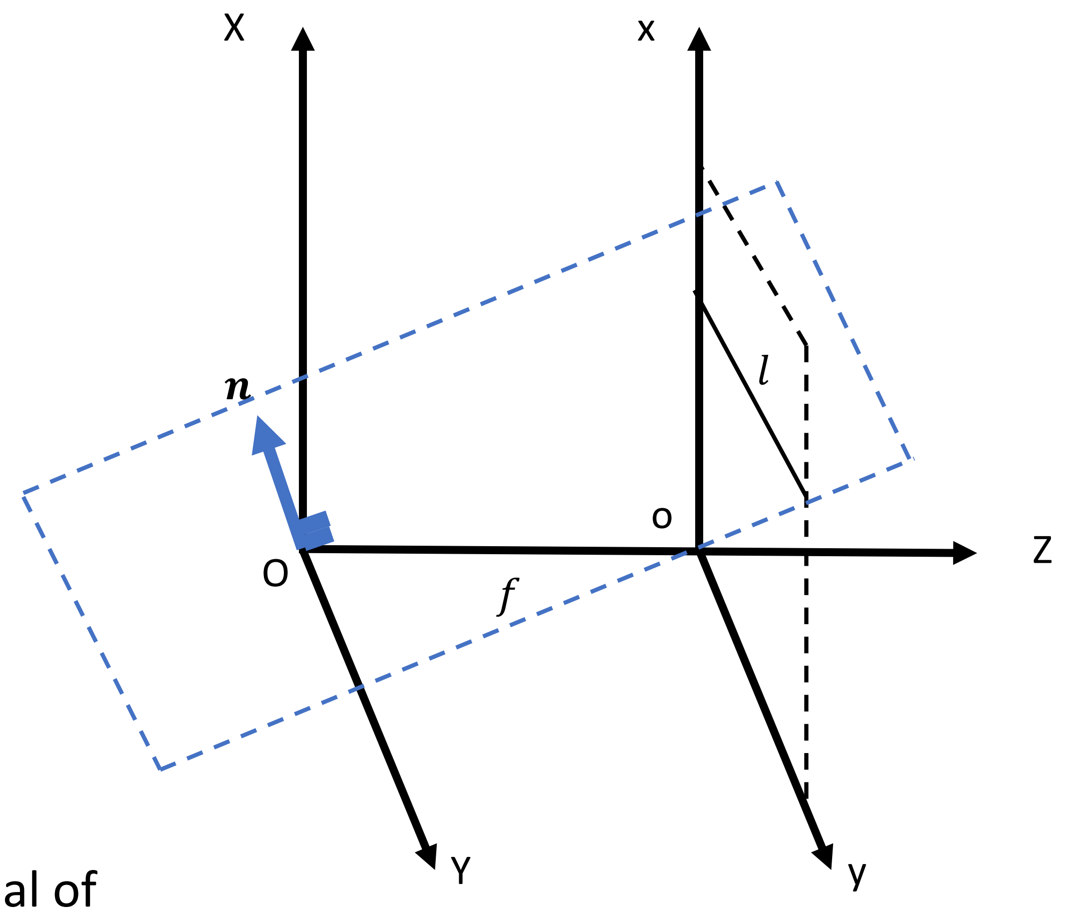
$$Ax + By + C = 0$$

The N-vector is given as

$$n = \pm N[\begin{pmatrix} A \\ B \\ C/f \end{pmatrix}]$$

This line can be interpreted as the intersection of two planes:

$$Z = f \quad \text{and} \quad AX + BY + CZ/f = 0$$



where the plane $AX + BY + CZ/f = 0$ passes through the viewpoint $O$and intersects the image plane along l. The normal of this plane is $(A, B, C/f)^T$.

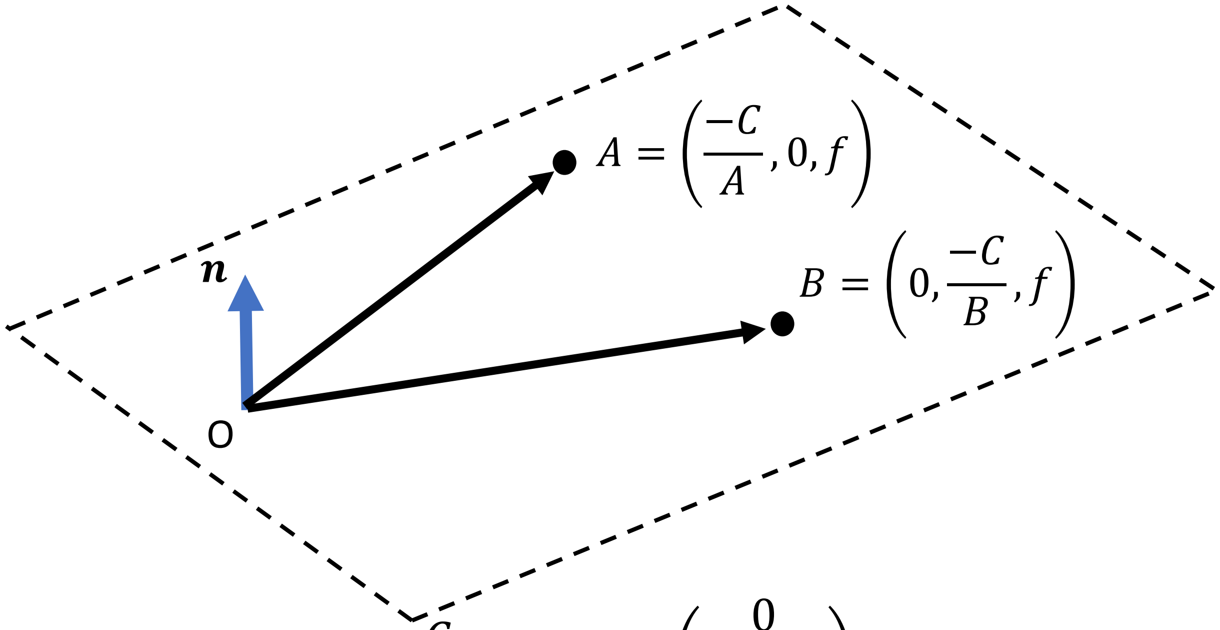
We can say that the N-vector of a line l can be interpreted as the unit normal vector normal to the plane passing through the viewpoint O and intersecting the image plane along l.

Proof

A plane in 3D is defined as; $Ax + By + Cz + D = 0$

when the plane passes through the origin, i.e. $D = 0$,
the plane becomes; $Ax + By + Cz = 0$

and $\boldsymbol{n} = \pm \begin{pmatrix} A \\ B \\ C \end{pmatrix}$ is the plane normal.



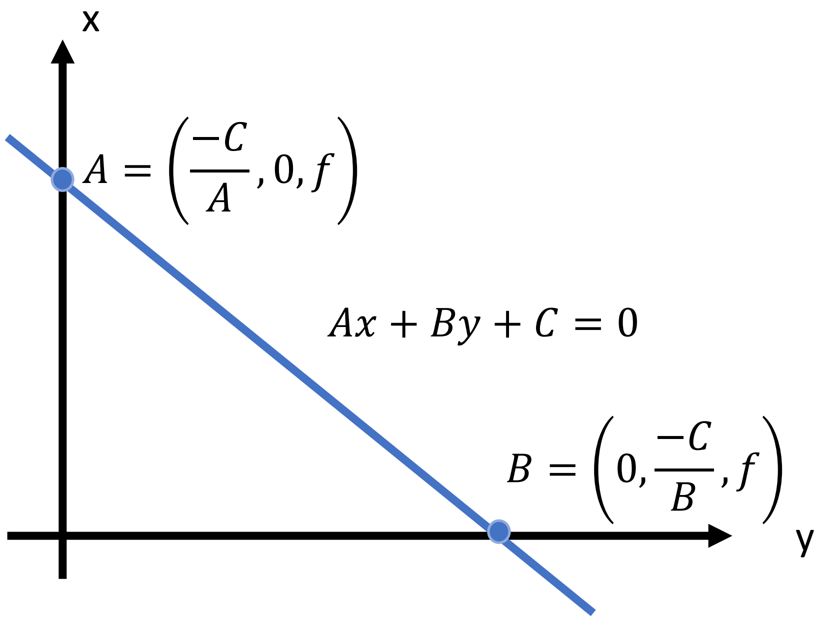
A: $\quad let\ \ y = 0, \qquad x = \frac{-C}{A} \Rightarrow \overrightarrow{OA} = \begin{pmatrix} -C/A \\ 0 \\ f \end{pmatrix}$

B: $\quad let\ \ x = 0, \qquad y = \frac{-C}{B} \Rightarrow \overrightarrow{OB} = \begin{pmatrix} 0 \\ -C/B \\ f \end{pmatrix}$

$$\vec{\boldsymbol{n}} = \overrightarrow{OB} \times \overrightarrow{OA} = \begin{pmatrix} 0 \\ -C/B \\ f \end{pmatrix} \times \begin{pmatrix} -C/A \\ 0 \\ f \end{pmatrix} = \begin{vmatrix} i & j & k \\ 0 & -C/B & f \\ -C/A & 0 & f \end{vmatrix}$$

$$= \begin{vmatrix} -C/B & f \\ 0 & f \end{vmatrix} i - \begin{vmatrix} 0 & f \\ -C/A & f \end{vmatrix} j + \begin{vmatrix} 0 & -C/B \\ -C/A & 0 \end{vmatrix} k$$

$$= \left(\frac{-C}{B} f, \frac{-C}{A} f, \frac{-C^2}{AB}\right) = \frac{\boldsymbol{-Cf}}{\boldsymbol{AB}} \left(\boldsymbol{A}, \boldsymbol{B}, \frac{\boldsymbol{C}}{\boldsymbol{f}}\right)$$



Computation of points & lines

a) Given two image lines l and l', the intersection P can be found as the vector (outer) product of their N-vectors. A plane in 3D is defined as;

$$\boldsymbol{m} = \pm N[\boldsymbol{n} \times \boldsymbol{n}']$$

Example: Find the intersection P of two image lines.

$$l_1\colon\ a_1x + b_1y + c_1 = 0$$
$$l_2\colon\ a_2x + b_2y + c_2 = 0$$

The N-vectors of l_1 and l_2 are;

$$\boldsymbol{n}_1 = \pm N[\begin{pmatrix} a_1 \\ b_1 \\ c_1/f \end{pmatrix}] \quad \text{and} \quad \boldsymbol{n}_2 = \pm N[\begin{pmatrix} a_2 \\ b_2 \\ c_2/f \end{pmatrix}]$$

Expand the above equations and solve for; $\boldsymbol{n}_1 \times \boldsymbol{n}_2$ we have; $\boldsymbol{m} = \pm N[\begin{pmatrix} (b_1c_2 - c_1b_2)/f \\ (c_1a_2 - a_1c_2)/f \\ a_1b_2 - b_1a_2 \end{pmatrix}] = \pm N[\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix}]$

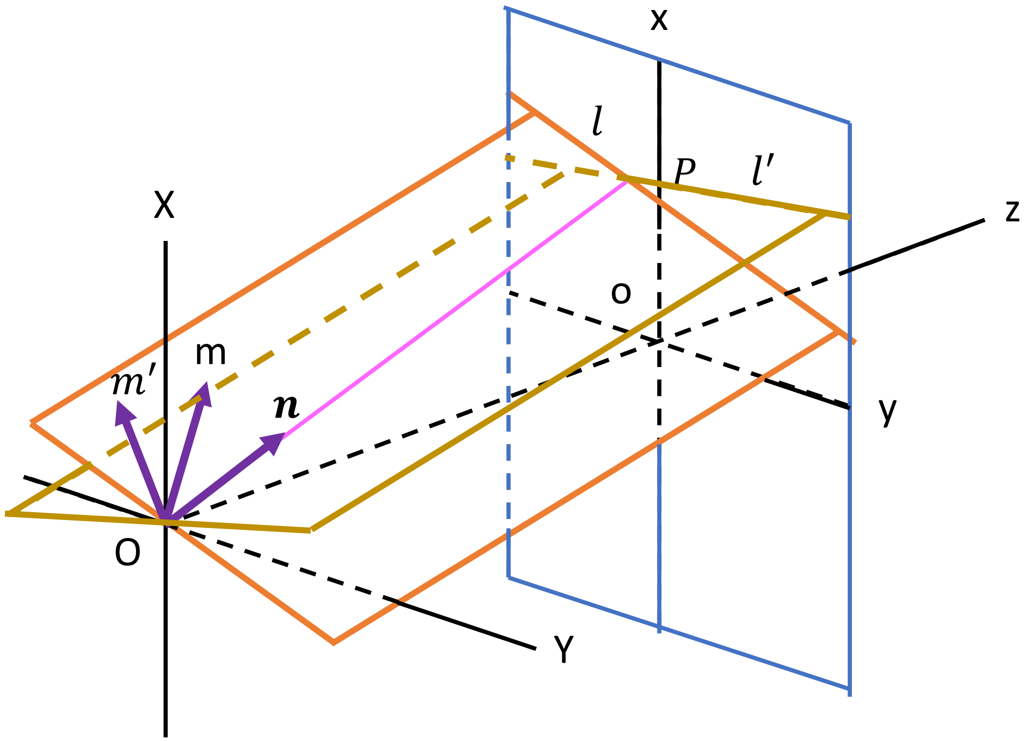
Or, the corresponding image coordinates are; $\left(\dfrac{b_1c_2 - c_1b_2}{a_1b_2 - b_1a_2}, \dfrac{c_1a_2 - a_1c_2}{a_1b_2 - b_1a_2}\right)$

If the two lines are parallel, $a_1b_2 - b_1a_2 = 0$, no solution will be found.

Computation of points & lines

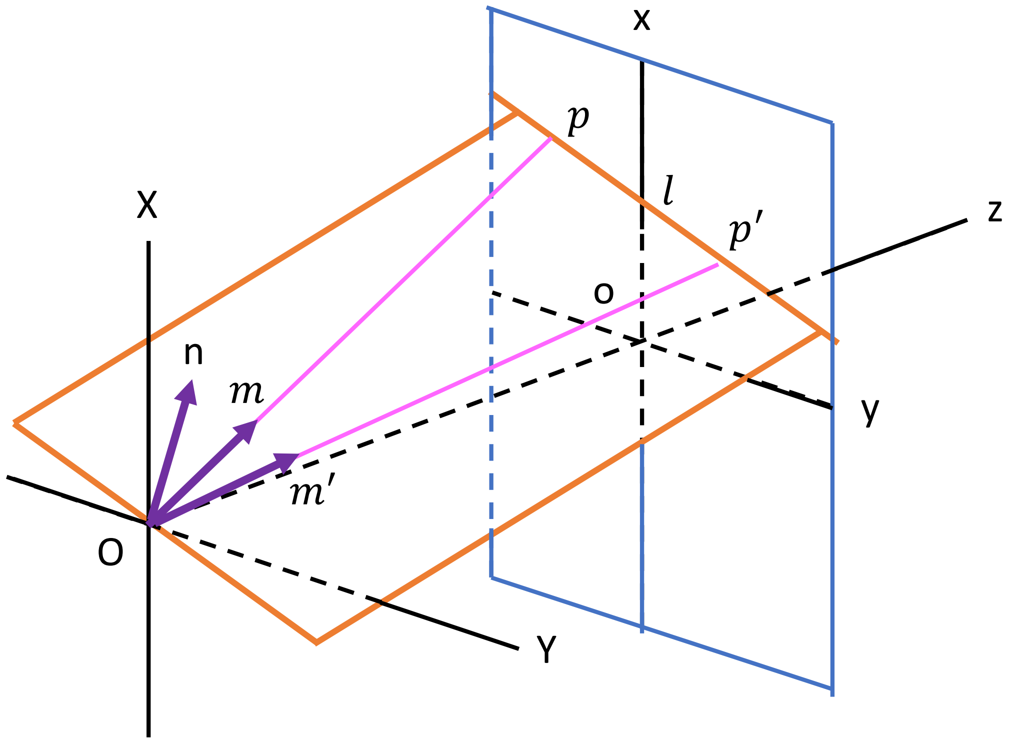
b) The N-vector $\boldsymbol{n}$ of the join of the two images points P and P' whose N-vectors are $\boldsymbol{m}$ and $\boldsymbol{m}'$, respectively, is given by;

$$\boldsymbol{n} = \pm N[\boldsymbol{m} \times \boldsymbol{m}']$$



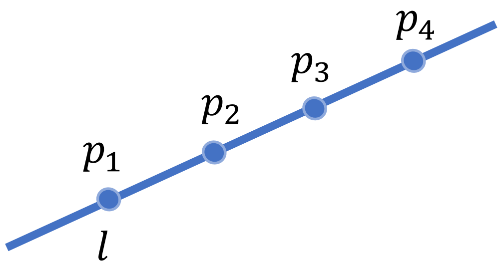
c) An image point P of N-vector $\boldsymbol{m}$ and an image line l of N-vector $\boldsymbol{n}$ are incident to each if the inner product is zero;

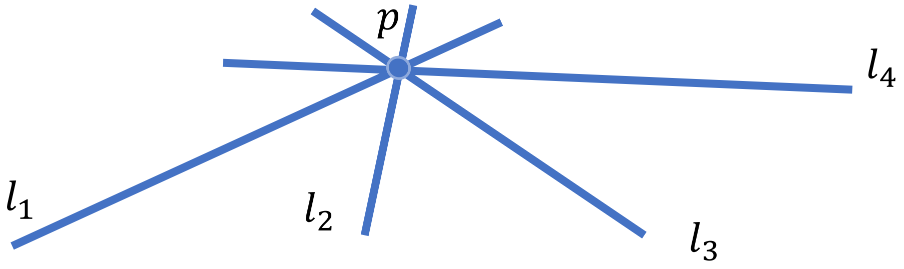
$$(m, n) = 0$$



Collinearity and Concurrency

- Points are ***collinear*** if there exists a line passing through all of them.
- Lines are ***concurrent*** if there exists a point that is on all of them.





Example-1: Show that image points are collinear if and only if the ***rank*** of their N-vector is less than three.

By definition, image points P_i of N-vectors $\boldsymbol{m}_i$, $i = 1$, ….., N, are collinear if f there exists a unit vector ***n*** such that ($\boldsymbol{m_i}$ **, n**) = 0, , $i = 1$, ….., N. This means that the vectors $\boldsymbol{m}_i$ are all perpendicular to n, which implies that any three of them are linearly dependent. Hence, the rank is less than three.

Conversely, if the rank is less than three, any three of the vectors m_i are coplanar, meaning that they all lie on a common plane. If we let ***n*** be the unit surface normal to it, we have ($\boldsymbol{m_i}$ **, n**) = 0, , $i = 1$, ….., N.

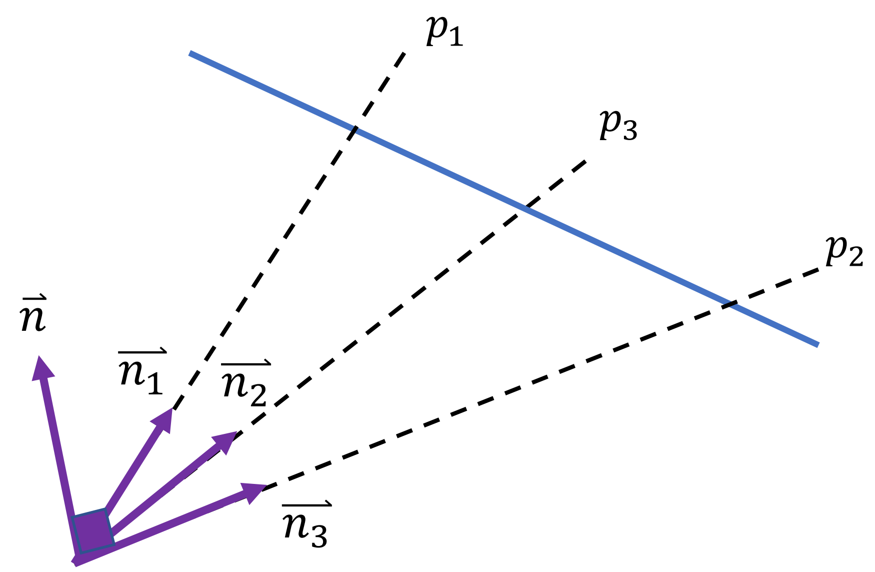
Collinearity and Concurrency

Example-2: Three image points of N-vectors m_1, m_2 and m_3 are collinear iff the scalar triple product

$$|m_1, m_2, m_3| = 0$$

Note :
$|m_1, m_2, m_3| = (m_1 \times m_2, m_3)$

$$|\overrightarrow{x_1},\ \overrightarrow{x_2},\ \overrightarrow{x_3}\ |$$
$$= (\overrightarrow{x_1} \times \overrightarrow{x_2}\ , \overrightarrow{x_3})$$
$$= 0$$



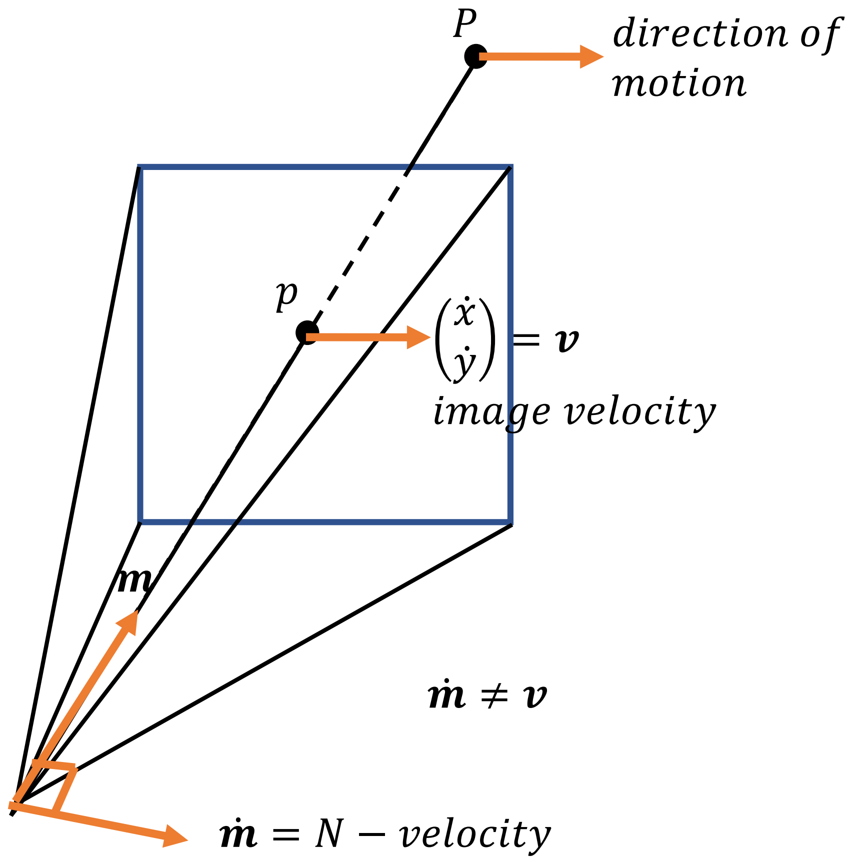
Translational Motion

1) If an image point (x, y) is moving with image velocity $(\dot{x}, \dot{y})$, its N-vector $\dot{\boldsymbol{m}}$ is given by;

$$\dot{\boldsymbol{m}} = \pm\left(\frac{1}{\sqrt{x^2+y^2+f^2}}\begin{pmatrix}\dot{x}\\ \dot{y}\\ 0\end{pmatrix} - \frac{1}{\left(\sqrt{x^2+y^2+f^2}\right)^3}\begin{pmatrix}x\\ y\\ f\end{pmatrix}\right)$$

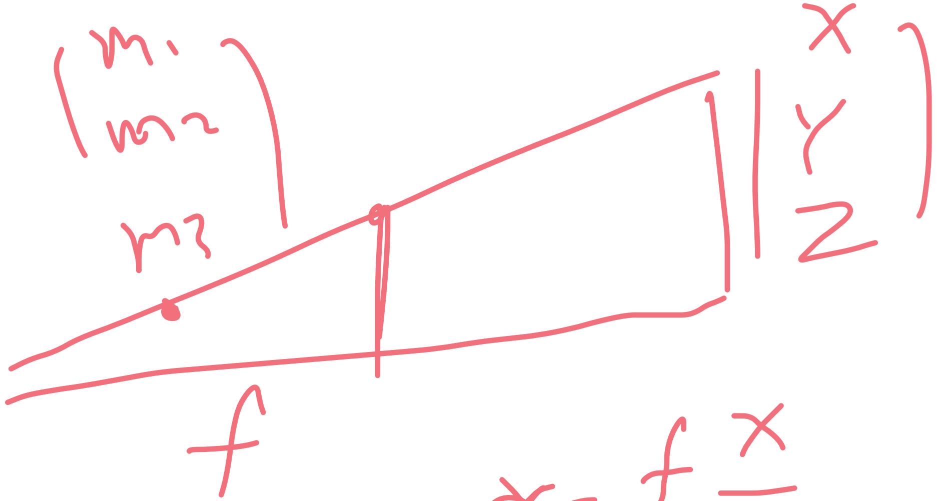
This is obtained by differentiating below with respect to time t, and $\dot{\boldsymbol{m}}$ is not normalized into a unit vector. $\dot{\boldsymbol{m}}$ is called N-velocity.

$$\boldsymbol{m} = \frac{\pm 1}{\sqrt{x^2+y^2+f^2}}\begin{pmatrix}x\\ y\\ f\end{pmatrix}$$



Translational Motion

$$\boldsymbol{m} = \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} \qquad \dot{\boldsymbol{m}} = \begin{pmatrix} \dot{m}_1 \\ \dot{m}_2 \\ \dot{m}_3 \end{pmatrix} = \begin{pmatrix} \frac{dm_1}{dt} \\ \frac{dm_2}{dt} \\ \frac{dm_3}{dt} \end{pmatrix}$$



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

$$= f\frac{m_1}{m_3}$$

2) If an image point (x, y) is moving with N-velocity $\dot{\boldsymbol{m}}$, its image velocity $(\dot{x}, \dot{y})$ is given by;

$$\boldsymbol{v} = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \frac{f}{m_3}\begin{pmatrix} \dot{m}_1 \\ \dot{m}_2 \end{pmatrix} - \frac{f\dot{m}_3}{m_3^2}\begin{pmatrix} m_1 \\ m_2 \end{pmatrix}$$

Proof: Image coordinates are given by the projection equation, hence; $x = f\frac{m_1}{m_3}$, $\quad y = f\frac{m_2}{m_3}$

The image velocity is obtained by differentiating these.

Translational Motion

Example: A rigid object is subjected to constant translational motion. It was found at (row,column)=(302,311) in frame one. After 40 milliseconds, it was found at (322,300) in frame two. Compute the image velocity and the N-velocity. The camera $f = 800\ pixels$, image center = (255,255) and images are (512,512).

1) Conversion

$$x_0 = -(302 - 255)$$
$$y_0 = 311 - 255$$
$$x_1 = -(322 - 255)$$
$$y_1 = 300 - 255$$

2) Image Velocity

$$\dot{x} = \frac{x_1 - x_0}{\Delta t}$$
$$\dot{y} = \frac{y_1 - y_0}{\Delta t}$$

$$v = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \Rightarrow 2\mathrm{D}$$

3) N-velocity

$$\dot{m} = \pm\left(\frac{1}{\sqrt{x^2 + y^2 + f^2}}\begin{pmatrix} \dot{x} \\ \dot{y} \\ 0 \end{pmatrix} - \frac{1}{\left(\sqrt{x^2 + y^2 + f^2}\right)^3}\begin{pmatrix} x \\ y \\ f \end{pmatrix}\right)$$

Translational Motion

3) The N-vector and the N-velocity of a moving point are orthogonal to each other.

$$(m, \dot{m}) = 0$$

4) If m and $\dot{m}$ are the N-vector and N-velocity, respectively, of a translating space point, the N-vector of its trajectory is;

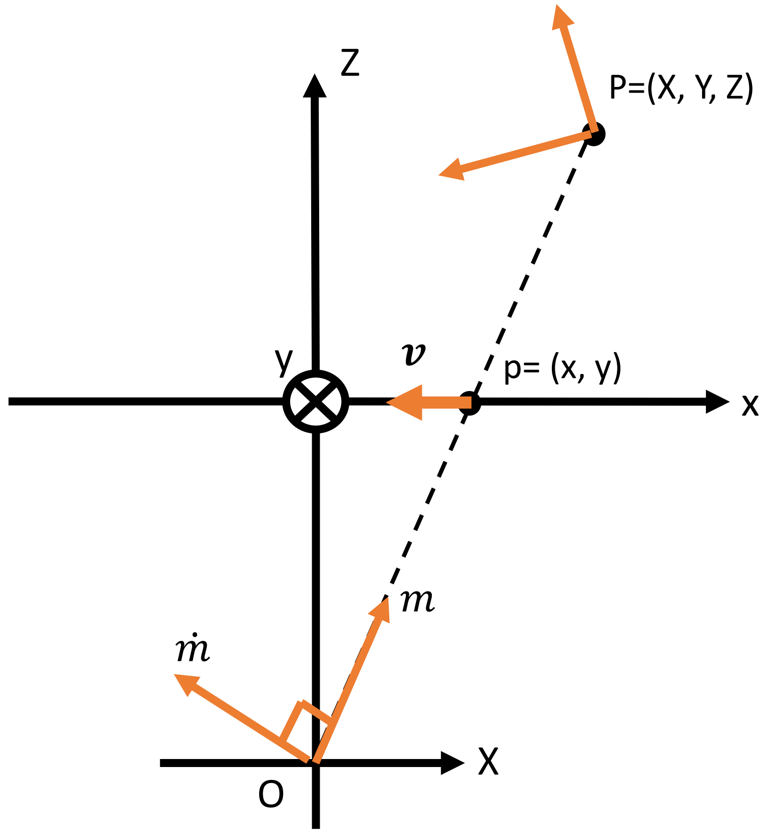
$$n = \pm N[m \times \dot{m}]$$

$$\dot{m} = \frac{\partial m}{\partial t} \quad , \qquad m = \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix}$$

$$\dot{m} \perp m \Rightarrow (\dot{m}, m) = 0$$

$$v = \begin{pmatrix} \frac{\partial x}{\partial t} \\ \frac{\partial y}{\partial t} \end{pmatrix} = image\ velocity; \qquad x = f\frac{m_1}{m_3} \quad , \qquad y = f\frac{m_2}{m_3}$$

$$v = \frac{f}{m_3}\begin{pmatrix} \dot{m}_1 \\ \dot{m}_2 \end{pmatrix} - \frac{f\dot{m}_3}{m_3^2}\begin{pmatrix} m_1 \\ m_2 \end{pmatrix}$$



Analysis of Translational Motion

Example: A point is observed at time t_1 as (X_1, Y_1, Z_1) and at time t_2 as (X_1', Y_1', Z_1'). What's its image velocity?

Solution:

Image point at t_1: $x_1 = f\dfrac{X_1}{Z_1}$, $y_1 = f\dfrac{Y_1}{Z_1}$

Image point at t_2: $x_1' = f\dfrac{X_1'}{Z_1'}$, $y_1' = f\dfrac{Y_1'}{Z_1'}$

Its image velocity:

$$v = \begin{pmatrix} \dfrac{\Delta x}{\Delta t} \\ \dfrac{\Delta y}{\Delta t} \end{pmatrix} = \begin{pmatrix} \dfrac{x_1' - x_1}{t_2 - t_1} \\ \dfrac{y_1' - y_1}{t_2 - t_1} \end{pmatrix}$$

Analysis of Translational Motion

Example: N image points from one object were observed with noise. Find its image velocity using least squares.

$$t_1: (x_1, y_1), (x_2, y_2), \ldots (x_N, y_N)$$

$$t_2: (x_1', y_1'), (x_2', y_2'), \ldots (x_N', y_N')$$

Solution:

$$\Delta t = t_2 - t_1 = 1\ (for\ simplicity)$$

$$Assume\ the\ optimal\ solution:\ v_m = \begin{pmatrix} v_x \\ v_y \end{pmatrix}$$

We observed,

$$v_1 = \begin{pmatrix} x_1' - x_1 \\ y_1' - y_1 \end{pmatrix}, \ldots, v_N = \begin{pmatrix} x_N' - x_N \\ y_N' - y_N \end{pmatrix}$$

Each observation has produced an error

$$\varepsilon_i = (v_m - v_i) = \begin{pmatrix} v_x - (x_i' - x_i) \\ v_y - (y_i' - y_i) \end{pmatrix}$$

Analysis of Translational Motion

The least squares error is;

$$\varepsilon = \sum_{i=1}^{N} \|\varepsilon_i\|^2 = \sum_{i=1}^{N} \varepsilon_i{}^T . \varepsilon_i = \sum_{i=1}^{N} \{[v_x - (x_i' - x_i)]^2 + [v_x - (y_i' - y_i)]^2\}$$

To minimize the error;

$$\frac{\partial \varepsilon}{\partial v_x} = 0 \ , \qquad \frac{\partial \varepsilon}{\partial v_y} = 0$$

Hence;

$$v_m = \begin{pmatrix} \frac{1}{N} \sum_{i=1}^{N} (x_i' - x_i) \\ \frac{1}{N} \sum_{i=1}^{N} (y_i' - y_i) \end{pmatrix}$$

x & y are independent variables.

3D Rotation

The least squares error is;

$$R^{-1} = R^{T} \ , \qquad RR^{T} = 1 \ (orthogonal)$$

$$\det(R) = 1$$

Let $(\theta_x, \theta_y, \theta_z)$ represent rotation about x, y & z axes.

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{pmatrix} \qquad R_y = \begin{pmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{pmatrix} \qquad R_z = \begin{pmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R = R_z R_y R_x \ \ (order\ is\ important)$$

3D rotation example for stereo image rectification

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$$

Let (e_1, e_2, e_3) represent orthogonal unit vectors, for example;

$$e_1 = (1,0,0)^T \quad - \ x\ axis$$
$$e_2 = (0,1,0)^T \quad - y\ axis$$
$$e_3 = (0,0,1)^T \quad - z\ axis$$

exam

1) Giving a vector $P_1 = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix}$ (note: $(x_1, y_{1,} z_1)$ is also coordinates of P_1 in e_1, e_2, e_3), rotate it by R.

$$P_2 = RP_1$$

2) We can rotate e_1, e_2, e_3 one-by-one, becoming e_1', e_2', e_3'

$$e_1' = \begin{pmatrix} x_1' \\ y_1' \\ z_1' \end{pmatrix}, \quad e_2' = \begin{pmatrix} x_2' \\ y_2' \\ z_2' \end{pmatrix}, \quad e_3' = \begin{pmatrix} x_3' \\ y_3' \\ z_3' \end{pmatrix}$$

Then the rotation matrix R is given as

$$R = \begin{pmatrix} x_1' & x_2' & x_3' \\ y_1' & y_2' & y_3' \\ z_1' & z_2' & z_3' \end{pmatrix}$$

For example;

$$e_1' = Re_1 = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} r_{11} \\ r_{21} \\ r_{31} \end{pmatrix} \quad or \quad \boxed{r_{ij} = e_i^T . e_j'}$$

Conversion from world coordination to local coordination

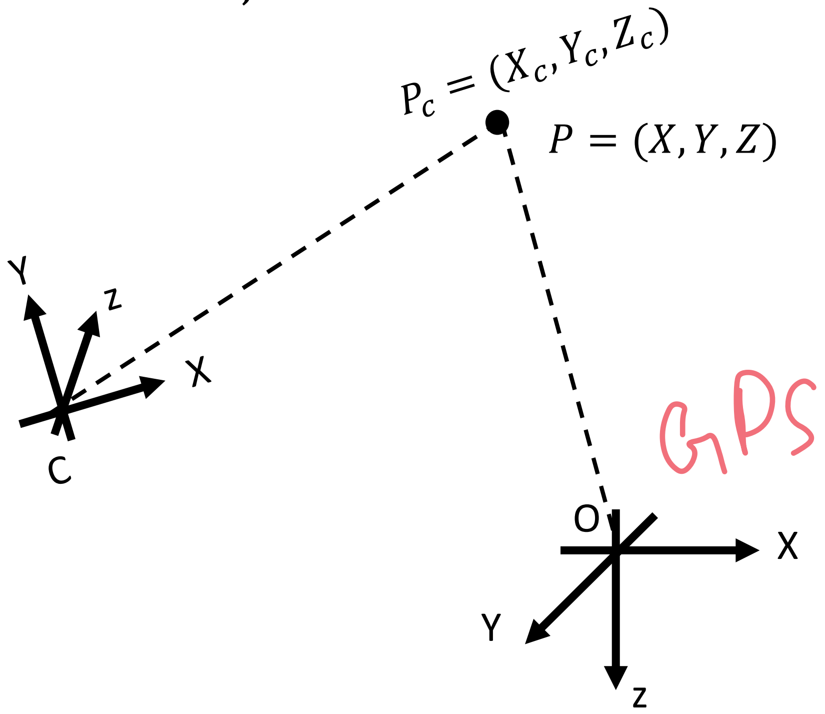
Giving a world coordinate $P = (X, Y, Z)$(e.g. satellite), we have a local camera at θ, C.

Then;

$$P_c^T = \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = R_{(\theta)}^T (P^T - C^T)$$

$$P_c = (X_c, Y_c, Z_c) \quad - in\ local\ camera\ frame$$

$$\theta = attitude$$



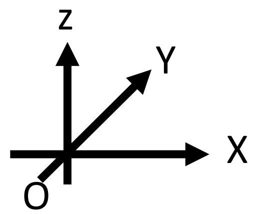
Camera Motion (ego motion)

Let the camera in the world coordinate system be denoted as (C, θ), where C is (X, Y, Z) (*location*) and $\theta = (\theta_X, \theta_Y, \theta_Z)$.
The relative motion from C_1 to C_2 is given as;

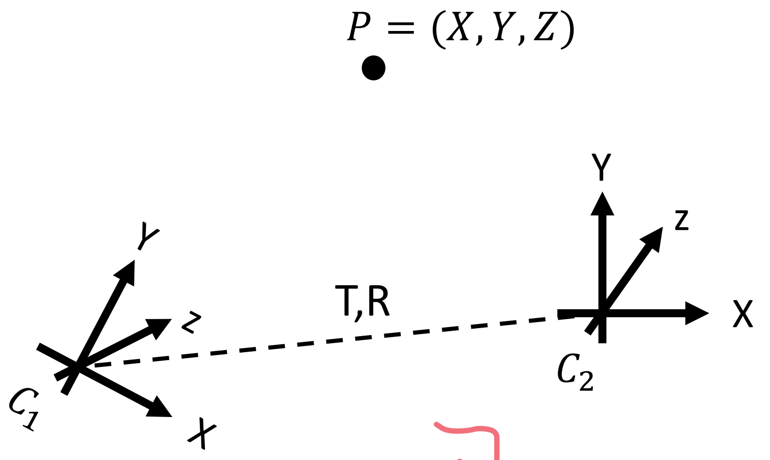
$$T = (C_2 - C_1)^T = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} \; ; \; T - translation$$

$$C_2^T = R(C_1^T + T) \; ; \; R - relative\, rotation$$

$$\boxed{R = R_{(\theta_2)} . R_{(\theta_1)}^T}$$



(world coordinate)



Change coordinate systems

A world point $\mathrm{P} = (X, Y, Z)$ in the camera's view at C_1 is $P_1 = (X_1, Y_1, Z_1)$. The camera is then subjected to an ego-motion (θ, t). This point is then in the C_2 location be viewed as $P_2 = (X_2, Y_2, Z_2)$.

$$P_2^T = R_{(\theta)}^T (P_1^T - t) \quad or \quad P_1^T = R_{(\theta)} (P_2^T + t)$$

Question

1) Given 2 sets of 3D points and their correspondence, compute the ego motion.

2) Noise are present in data, use RANSAC to compute ego motion.

3) A robot is equipped with camera and GPS. Assume a target travelling at constant velocity $v_t = \begin{pmatrix} \dot{x}_t \\ \dot{y}_t \end{pmatrix}$.

   Find the target's location and v_t.