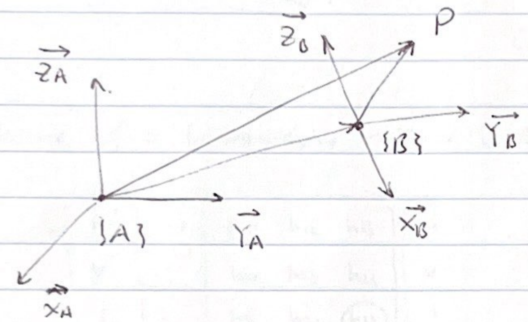
$\boxed{\text{CSCI 5561 COMPUTER VISION}}$ Lecture. Oct. 17th.

COORDINATION TRANSFORMATION



Notation:

$\vec{X_A}$: unit vector in A frame

$\vec{Y_A}$ . . . .

$\vec{Z_A}$ . . .

$\vec{X_B}$: unit vector in B frame.

⋮

▷ Consider 3D point. P. and vector $\vec{AP}$. $\vec{BP}$. $\vec{AB}$

$$\boxed{\vec{AP} = \vec{AB} + \vec{BP}} \qquad \text{Eq. 1.}$$

Has nothing related to coordination until now.

▷ Express them in. A. frame or B frame.

$$\vec{AP} = (\vec{X_A}\ \vec{Y_A}\ \vec{Z_A}) \underbrace{\begin{pmatrix} {}^A P_x \\ {}^A P_y \\ {}^A P_z \end{pmatrix}}_{{}^A p} $$

${}^A p$ → coordinates in A frame.

$$\vec{AB} = (\vec{X_A}.\ \vec{Y_A}.\ \vec{Z_A}) \begin{pmatrix} {}^A t_x \\ {}^A t_y \\ {}^A t_z \end{pmatrix}$$

— ${}^A t$: translation.

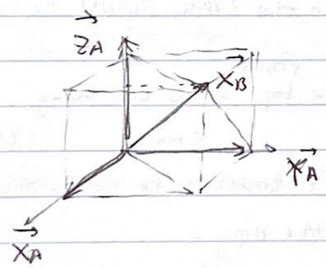
$$\vec{^BP} = \begin{pmatrix} \vec{X_B} & \vec{Y_B} & \vec{Z_B} \end{pmatrix} \begin{pmatrix} ^Bp_x \\ ^Bp_y \\ ^Bp_z \end{pmatrix}$$ — ^BP: coordinates in B frame.

▷ Insert back to Eq. 1.

$$\underbrace{\begin{pmatrix} \vec{X_A} & \vec{Y_A} & \vec{Z_A} \end{pmatrix} \begin{pmatrix} ^Ap_x \\ ^Ap_y \\ ^Ap_z \end{pmatrix}}_{\rightarrow\, ^Ap} = \begin{pmatrix} \vec{X_A} & \vec{Y_A} & \vec{Z_A} \end{pmatrix} \underbrace{\begin{pmatrix} ^At_x \\ ^At_y \\ ^At_z \end{pmatrix}}_{\rightarrow\, ^At} + \begin{pmatrix} \vec{X_B} & \vec{Y_B} & \vec{Z_B} \end{pmatrix} \underbrace{\begin{pmatrix} ^Bp_x \\ ^Bp_y \\ ^Bp_z \end{pmatrix}}_{\rightarrow\, ^Bp}$$

▷ Express in the same coordinate system.

$$\begin{pmatrix} \vec{X_B} & \vec{Y_B} & \vec{Z_B} \end{pmatrix} = \begin{pmatrix} \vec{X_A} & \vec{Y_A} & \vec{Z_A} \end{pmatrix} \underbrace{\begin{bmatrix} \vec{X_B}\cdot\vec{X_A} & \vec{Y_B}\cdot\vec{X_A} & \vec{Z_B}\cdot\vec{X_A} \\ \vec{X_B}\cdot\vec{Y_A} & \vec{Y_B}\cdot\vec{Y_A} & \vec{Z_B}\cdot\vec{Y_A} \\ \vec{X_B}\cdot\vec{Z_A} & \vec{Y_B}\cdot\vec{Z_A} & \vec{Z_B}\cdot\vec{Z_A} \end{bmatrix}}_{\downarrow}$$



^AR_B: express B axis in A frame.

▷ $$\begin{pmatrix} \vec{X_A} & \vec{Y_A} & \vec{Z_A} \end{pmatrix} \begin{pmatrix} ^Ap_x \\ ^Ap_y \\ ^Ap_z \end{pmatrix} = \begin{pmatrix} \vec{X_A} & \vec{Y_A} & \vec{Z_A} \end{pmatrix} {}^AR_B \begin{pmatrix} ^Bp_x \\ ^Bp_y \\ ^Bp_z \end{pmatrix} + \begin{pmatrix} \vec{X_A} & \vec{Y_A} & \vec{Z_A} \end{pmatrix} \begin{pmatrix} ^At_x \\ ^At_y \\ ^At_z \end{pmatrix}$$

$$\Downarrow$$

▷ Differentiate between R/T as a 'operator', and a way to describe a coordinate.

$$^Ap = {}^AR_B\, {}^Bp + {}^At$$

^AT_B

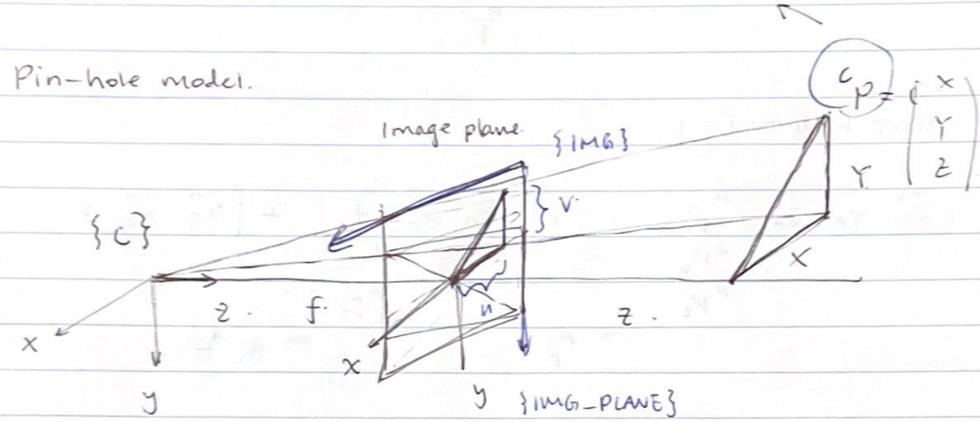
$$\Downarrow$$

$$\begin{pmatrix} ^Ap \\ 1 \end{pmatrix} = \begin{bmatrix} ^AR_B & ^At \\ 0 & 1 \end{bmatrix} \begin{bmatrix} ^Bp \\ 1 \end{bmatrix}$$

CAMERA MODEL

IMPORTANT: P in camera frame.

Pin-hole model.



projection relationship: $\boxed{\dfrac{f}{z} = \dfrac{u}{X} = \dfrac{v}{Y}}$

in the {IMG_PLANE} coordination frame.

{IMG_PLANE} to {IMG}: 1) scale ~~by~~ from W_{CCD} to W_{img}
(mm) (640 pxs).

2) shift/translate to left-top corner as the origin.

$$\begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} u \cdot \dfrac{W_{img}}{W_{CCD}} + t_x \\ v \dfrac{H_{img}}{H_{CCD}} + t_y \end{pmatrix} = \begin{bmatrix} f \dfrac{X}{Z} \cdot \dfrac{W_{img}}{W_{CCD}} + t_x \\ f \cdot \dfrac{Y}{Z} \cdot \dfrac{H_{img}}{H_{CCD}} + t_y \end{bmatrix}$$

$$= \begin{bmatrix} \dfrac{f}{Z} \cdot \dfrac{W_{img}}{W_{CCD}} & \cancel{t_x}\, 0 & t_x \\ 0 & \dfrac{f}{Z} \cdot \dfrac{H_{img}}{W_{CCD}} & t_y \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Remove Z from matrix

$$Z \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = \begin{bmatrix} f \cdot \dfrac{W_{img}}{W_{CCD}} & 0 & Z t_x \\ 0 & f \cdot \dfrac{H_{img}}{H_{CCD}} & Z t_y \\ 0 & 0 & Z \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} f \dfrac{W_{img}}{W_{CCD}} & 0 & t_x \\ 0 & f \dfrac{H_{img}}{H_{CCD}} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

INTRINSIC MATRIX

▷ Note that. ${}^{c}p$ is in the camera frame. ${}^{c}p = \begin{bmatrix} {}^{c}x \\ {}^{c}y \\ {}^{c}z \end{bmatrix}$

▷ In practice, we have points in the world frame $\{w\}$, ${}^{w}p$

~~thef~~ therefore, we need to transform them into. ~~wo.~~ camera frame.

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K\,{}^{c}p = K \cdot \underbrace{\begin{bmatrix} {}^{c}R_w & {}^{c}t \end{bmatrix}}_{\text{transform } w \text{ to } c} \cdot {}^{w}p$$

$\boxed{\text{Entrisic matrix}}$

▷ In practice, we. have. camera poses. with respect to the world,

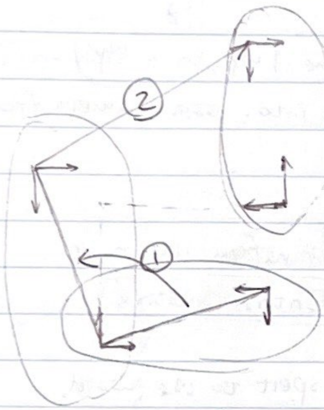
i.e. we have. ${}^{w}T_c$. $t = \begin{bmatrix} {}^{w}R_c & {}^{w}t \\ 0 & 1 \end{bmatrix}$

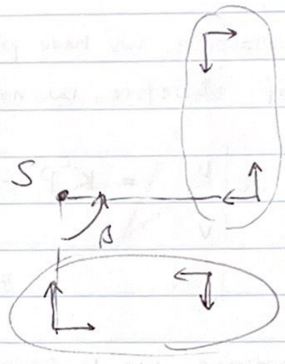
Therefore, we writing code or usage. sometimes we use.

$${}^{c}T_w = \begin{bmatrix} {}^{c}R_w & {}^{c}t \\ 0 & 1 \end{bmatrix} = {}^{w}T_c^{-1} = \begin{bmatrix} {}^{w}R_c^{-1} & -{}^{w}R_c^{-1} \cdot {}^{w}t \\ 0 & 1 \end{bmatrix}$$

Pay extra attention to this.

OTHER REPRESENTATION OF ROTATION





Rotation first. then. second

Rotate along axis.

S: screw axis.

$R = Rot(\hat{\omega}, \theta)$

Rotation $Rot(\hat{\omega}, \theta)$ as an operator.

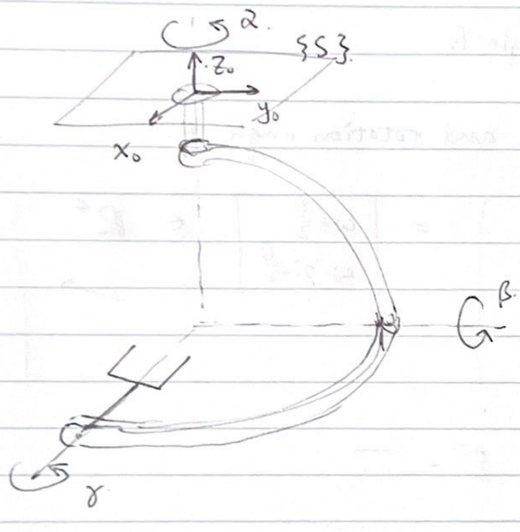
$$^{A}P_2 = Rot(\hat{\omega}, \theta)\ ^{A}P_1$$

Note: operator is different than the 'description' from a frame.

$$Rot(\hat{\omega}, \theta) = \begin{bmatrix} c\theta + \hat{\omega}_1^2(1-c\theta), & \hat{\omega}_1\hat{\omega}_2(1-c\theta) - \hat{\omega}_3 s\theta, & \hat{\omega}_1\hat{\omega}_3(1-c\theta) + \hat{\omega}_2 s\theta \\ w_1 w_2(1-c\theta) + \hat{\omega}_3 s\theta, & c\theta + \hat{\omega}_2^2(1-c\theta), & \hat{\omega}_2\hat{\omega}_3(1-c\theta) - \hat{\omega}_1 s\theta \\ w_1 w_3(1-c\theta) - \hat{\omega}_2 s\theta, & \hat{\omega}_2\hat{\omega}_3(1-c\theta) + \hat{\omega}_1 s\theta, & c\theta + \hat{\omega}_3^2(1-c\theta) \end{bmatrix}$$

$$Rot(\hat{x}, \theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \quad \cdots$$

▷ Euler Angles (ZYX)



1) Rotate by Z-axis of {S} α

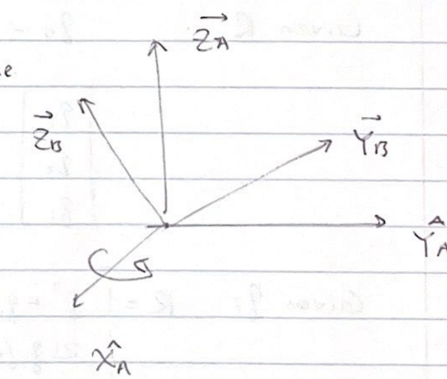
2) Rotate by y-axis of body-axis β

3) Rotate by ~~z~~ x-axis of body frame γ.

$$Rot(\alpha, \beta, \gamma) = \underline{Rot(\hat{z}, \alpha) \cdot Rot(\hat{y}, \beta) \cdot Rot(\hat{x}, \gamma)}$$

The order matters!

▷ Roll-Pitch-Yaw. (X-Y-Z) Fixed frame



1) Rotate by X-axis in $\underline{\{A\}}$ by γ.

2) Rotate by y-axis in $\underline{\{A\}}$ by β.

3) Rotate by Z-axis in {A} by α.

$$Rot(\alpha, \beta, \gamma) = Rot(\hat{z}, \alpha) \cdot Rot(\hat{y}, \beta) \cdot Rot(\hat{x}, \gamma).$$

Same as the previous one.

Quaternion.

Numerical stable to small angle θ.

For a rotation axis $\hat{\omega}\cdot\theta$. and rotation angle θ.

$$q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos\frac{\theta}{2} \\ \omega_x \cdot \sin\frac{\theta}{2} \\ \omega_y \sin\frac{\theta}{2} \\ \omega_z \sin\frac{\theta}{2} \end{bmatrix} = \begin{bmatrix} \cos\frac{\theta}{2} \\ \omega \cdot \sin\frac{\theta}{2} \end{bmatrix} \in R^4 \rightarrow \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix}$$

Property: $\|q\| = 1$ $\quad q^{-1} = \bar{q}$

Given R.

$$q_0 = \frac{1}{2}\sqrt{1 + r_{11} + r_{22} + r_{33}}$$

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \frac{1}{4q_0}\begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}$$

Given q:

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_0 q_2 + q_1 q_3) \\ 2(q_0 q_3 + q_1 q_2) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_0 q_1 + q_2 q_3) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

Note; in some pkgs. q is written as $\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$ instead of $\begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix}$