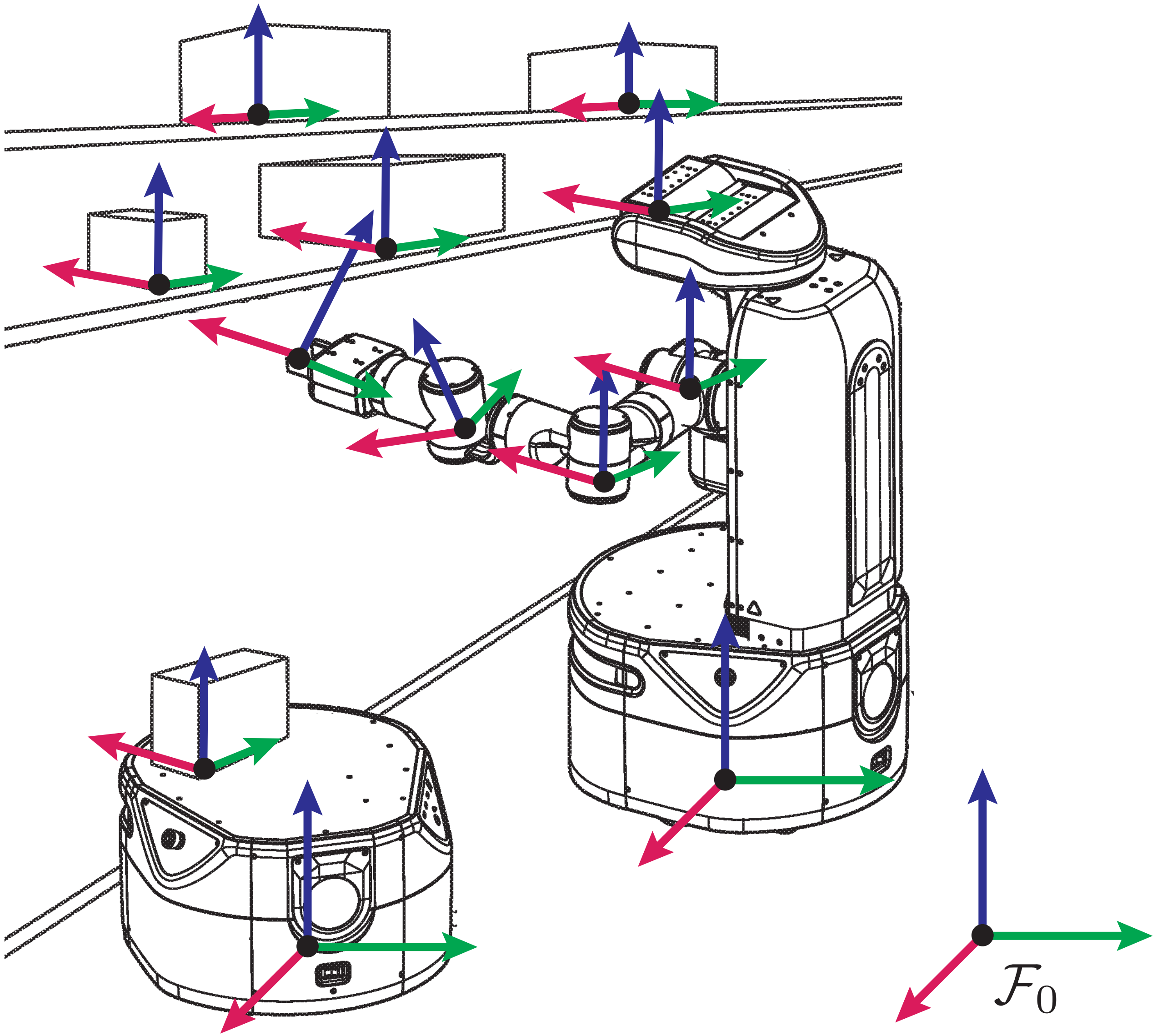
Rotations & Transformations

Lecture 3

Fall 2022

Coordinate Frames



If we want a robot to be able to pick a box and place it somewhere. What spatial information do we need to know?

Global frame
Robot Frame
Camera frame

Pose of robots
Pose of gripper
Pose of all joints
Pose of target boxes
Pose of obstacle boxes

fetchrobotics.com

Points & Vectors

"origin of frame 0 in frame 1"

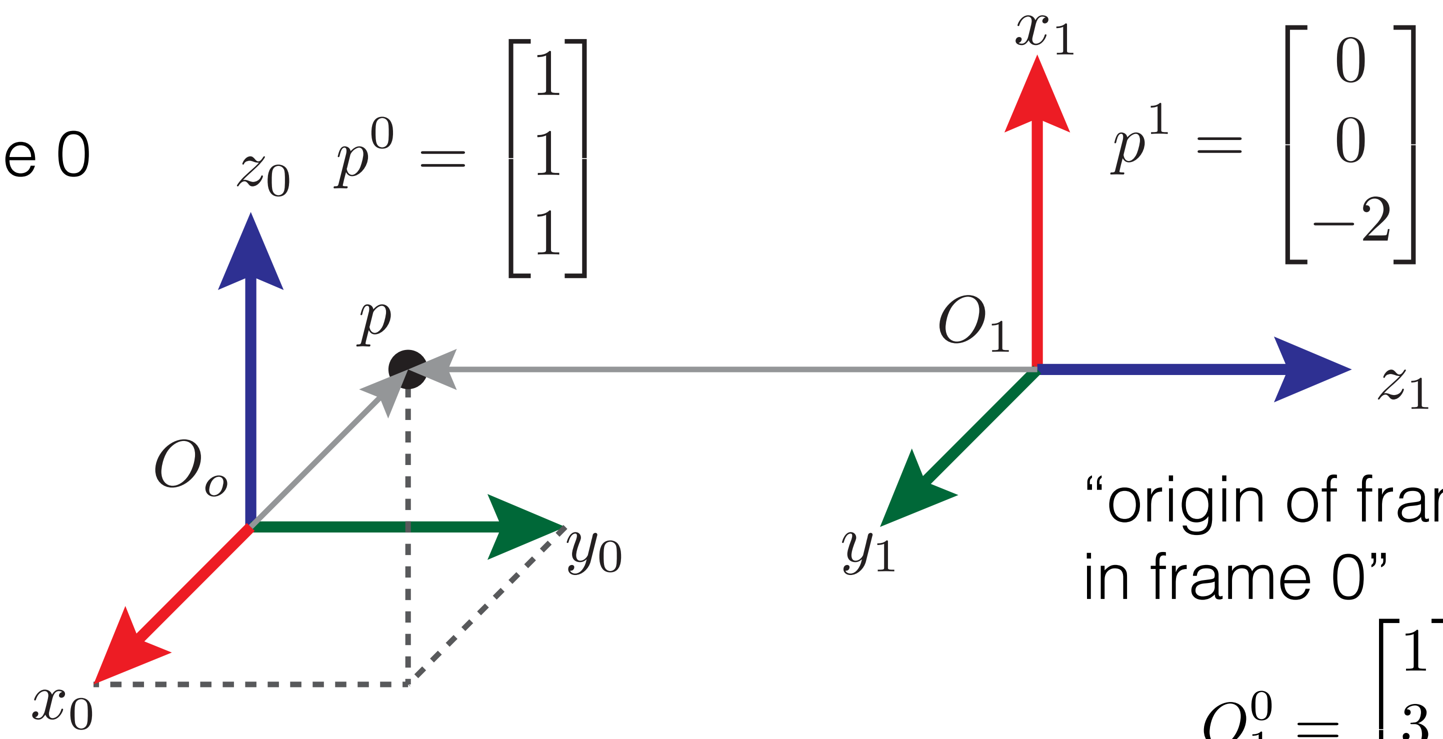
$$O_0^1 = \begin{bmatrix} -1 \\ -1 \\ -3 \end{bmatrix}$$

"point p in frame 0"

$$p^0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

"point p in frame 1"

$$p^1 = \begin{bmatrix} 0 \\ 0 \\ -2 \end{bmatrix}$$



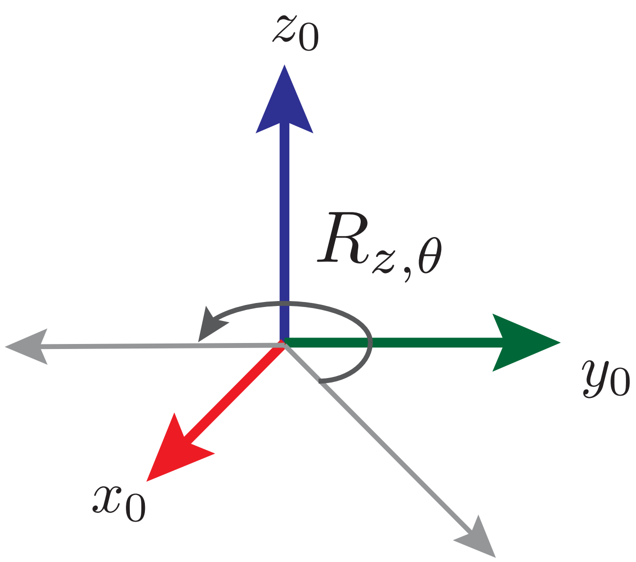
"origin of frame 1 in frame 0"

$$O_1^0 = \begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix}$$

3D Rotations

Direction Cosine Matrix (DCM)

$$R_1^0 = \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 & z_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 & z_1 \cdot y_0 \\ x_1 \cdot z_0 & y_1 \cdot z_0 & z_1 \cdot z_0 \end{bmatrix}$$



Rotation around any single axis:

$$R_{z,\theta} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

$$R_{y,\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

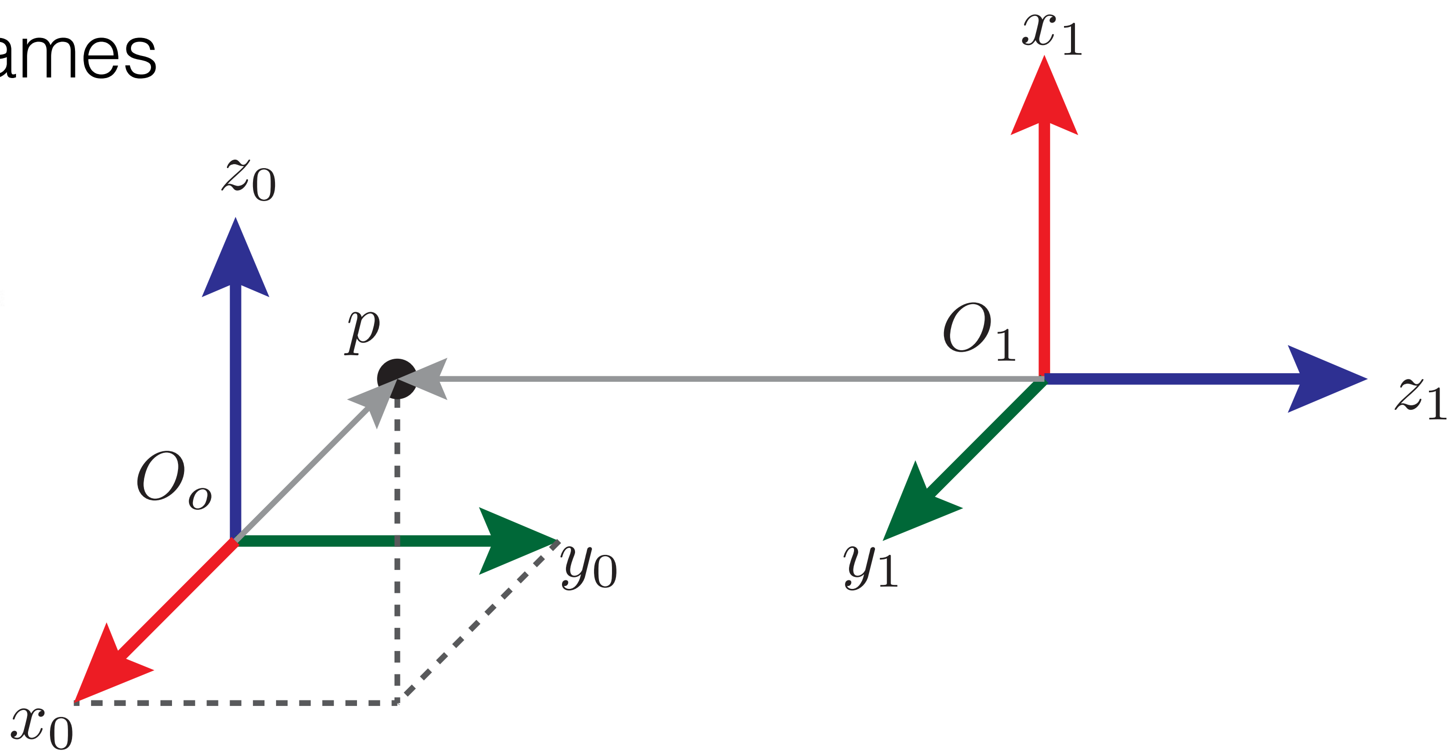
Interpretation of Rotation Matrix

- Represents coordinate transform of a point p in two different frames (at the same origin)
- Gives the orientation of a transformed coordinate frame relative to another fixed coordinate frame (at the same origin)
- Operator rotating a vector in a coordinate system to a new vector in the same coordinate system – rigid body rotation

Coordinate Transformation

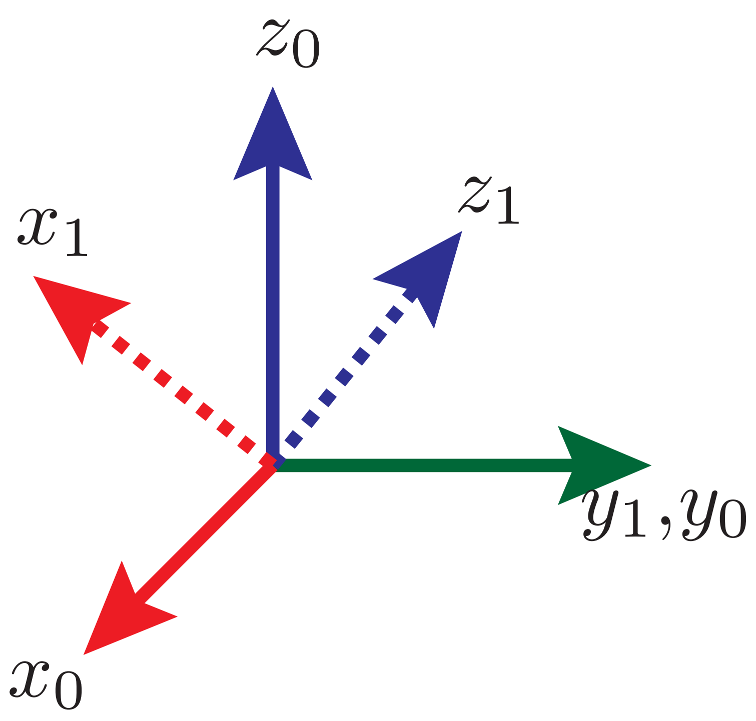
- Give representation of points in different reference frames

$$p^0 \quad = \quad R_1^0 p^1 + d_1^0$$



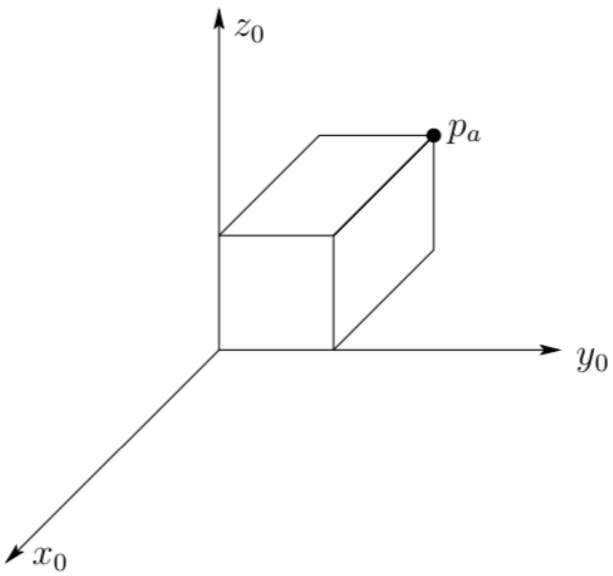
Orientation

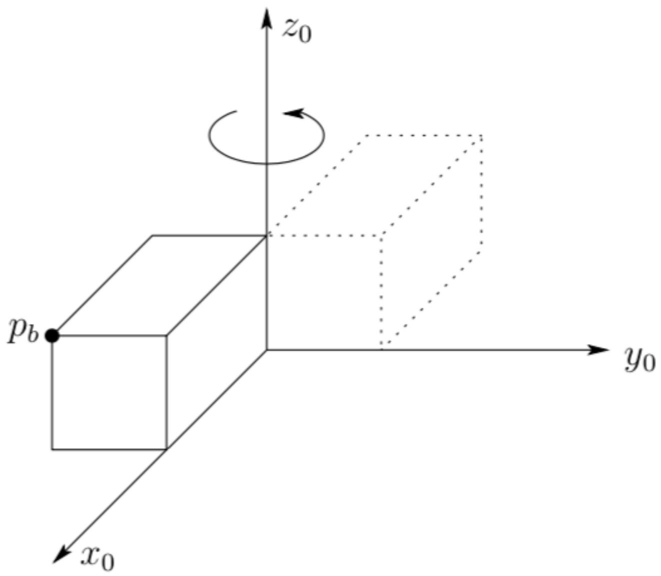
- Represents orientation of one frame with respect to a fixed frame
- Example: orientation of end effector in World Frame

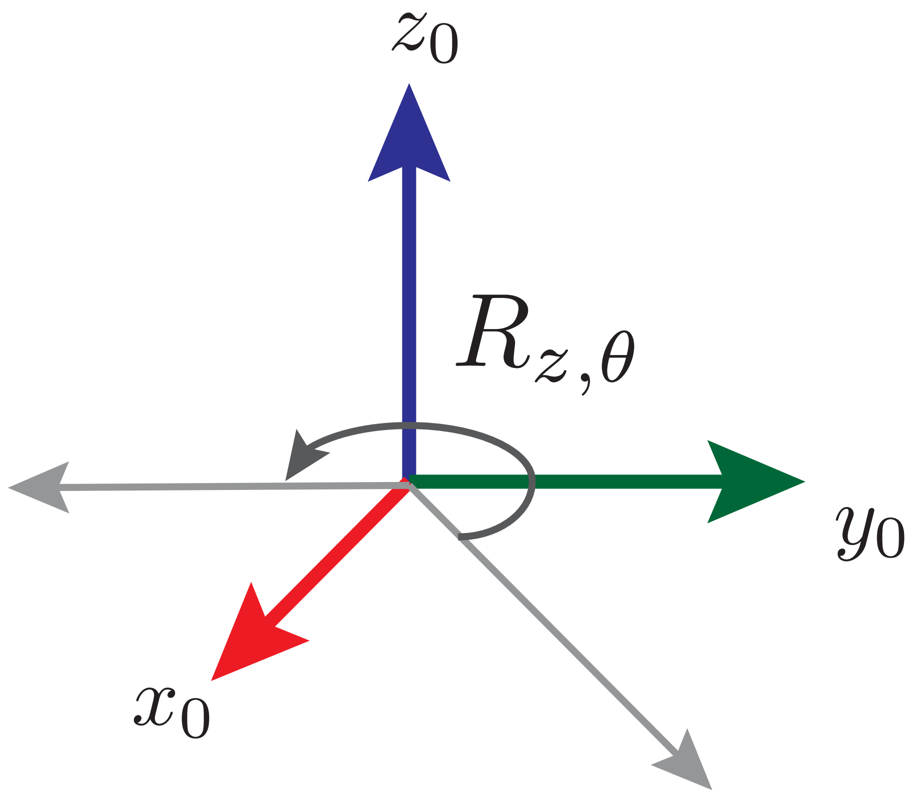


Rotation Operator

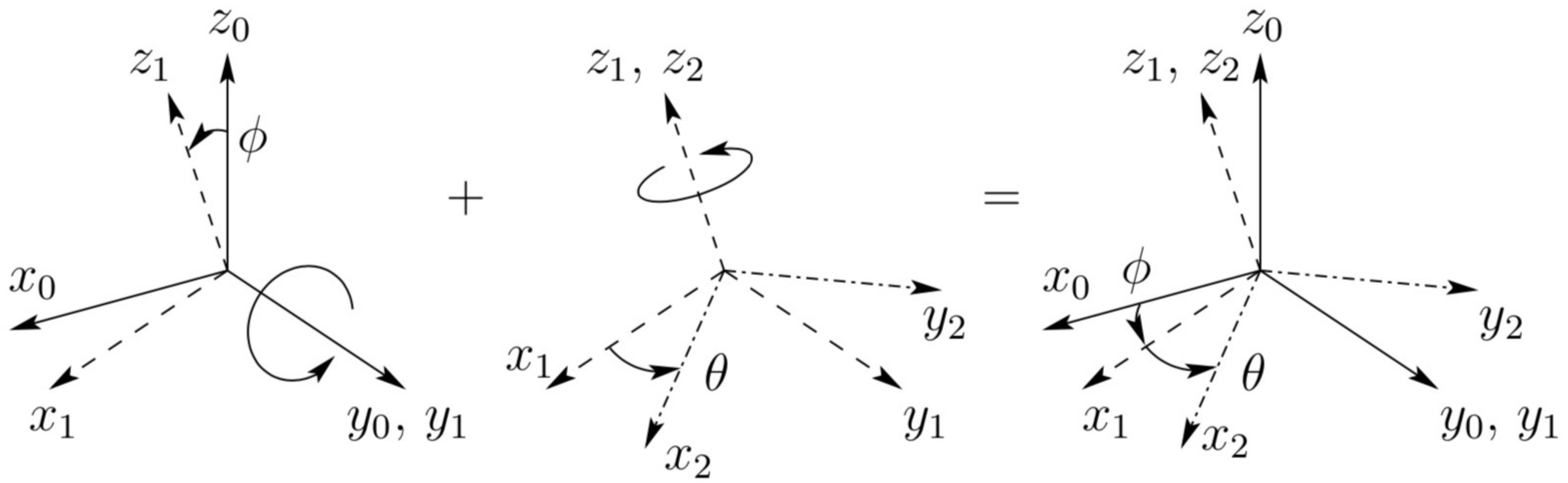
- Transform a vector or point to a new vector in the same frame
- Rigid body rotation







Sequential Operations - Current Frame



$$p^0 = R_{y,\phi}p^1$$

$$p^1 = R_{z,\theta}p^2$$

$$p^0 = R_{y,\phi}R_{z,\theta}p^2$$

Post-multiply for current frame

Sequential Operations - Fixed Frame

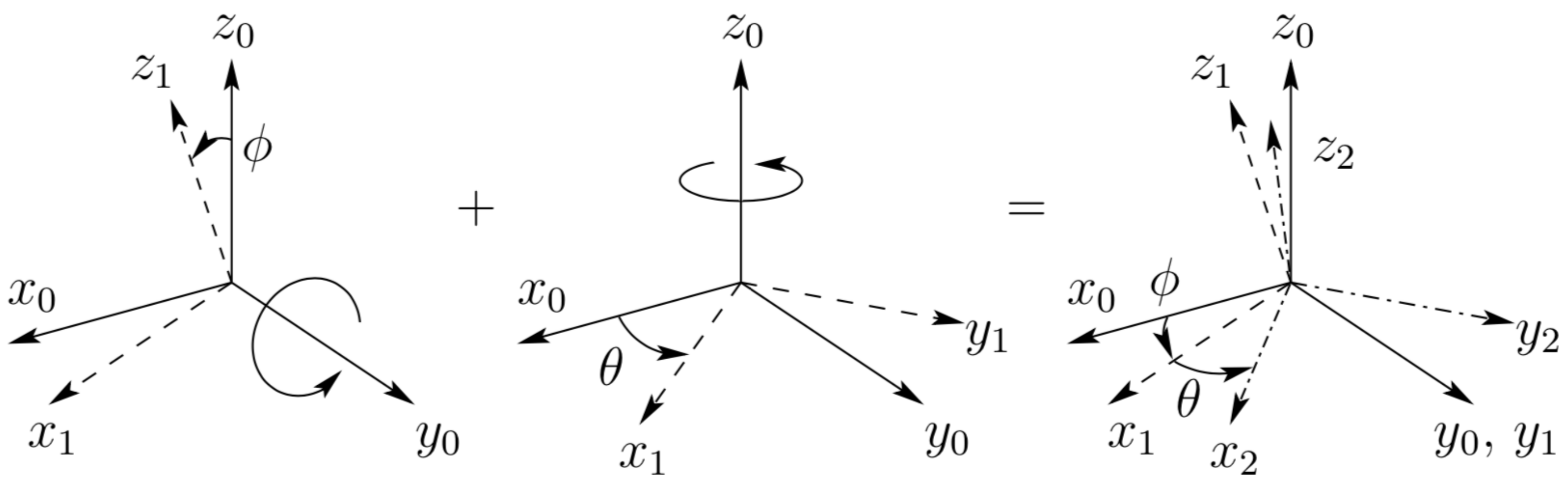
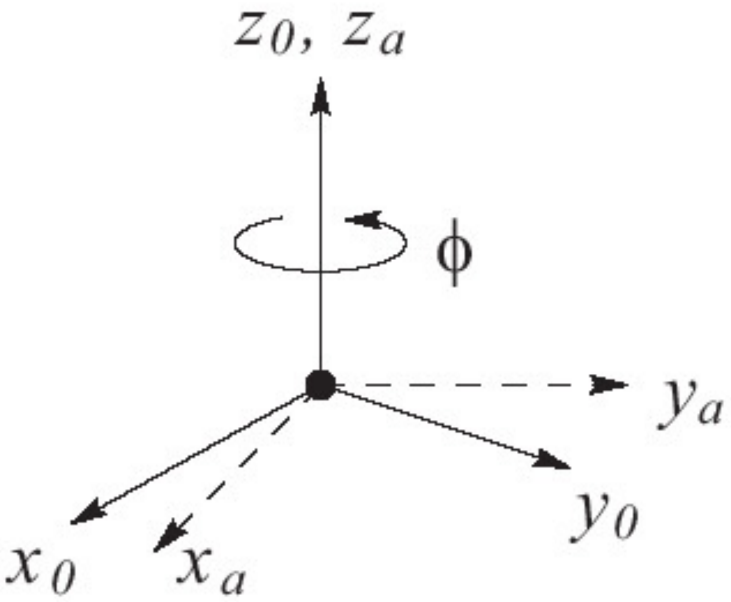


Figure 2.10: Composition of rotations about fixed axes.

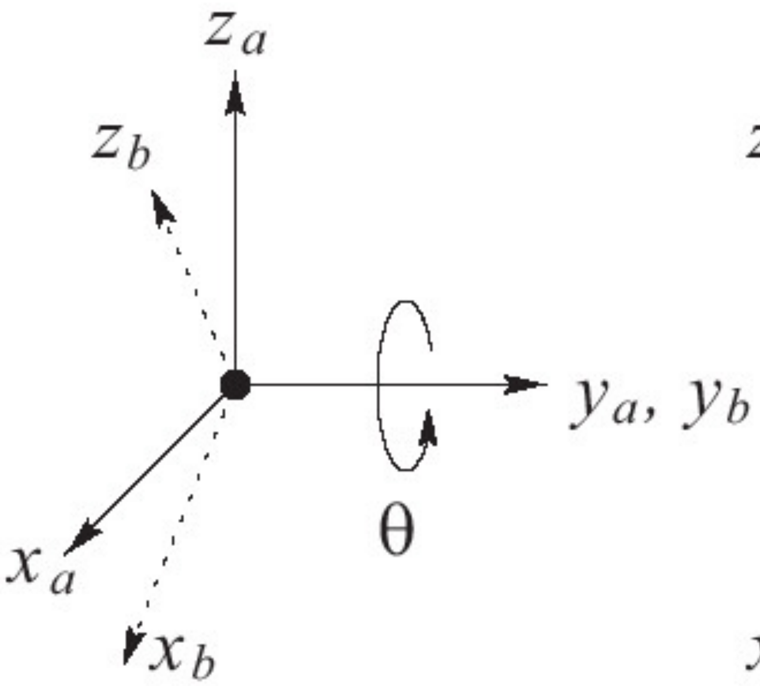
$$
\begin{aligned}
p^0 &= R_{y,\phi} p^1 \\
&= R_{y,\phi} \left[R_{y,-\phi} R_{z,\theta} R_{y,\phi} \right] p^2 \\
&= R_{z,\theta} R_{y,\phi} p^2 .
\end{aligned}
$$

Pre-multiply for fixed frame

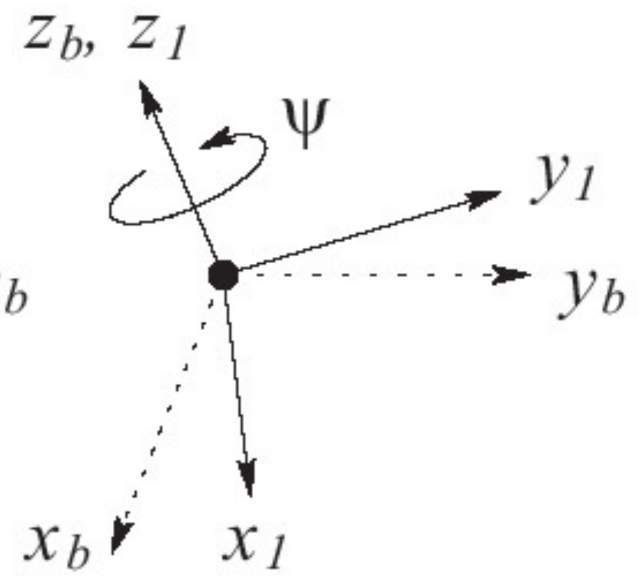
Euler Angle Representation



Rotate around Z



Rotate around current Y



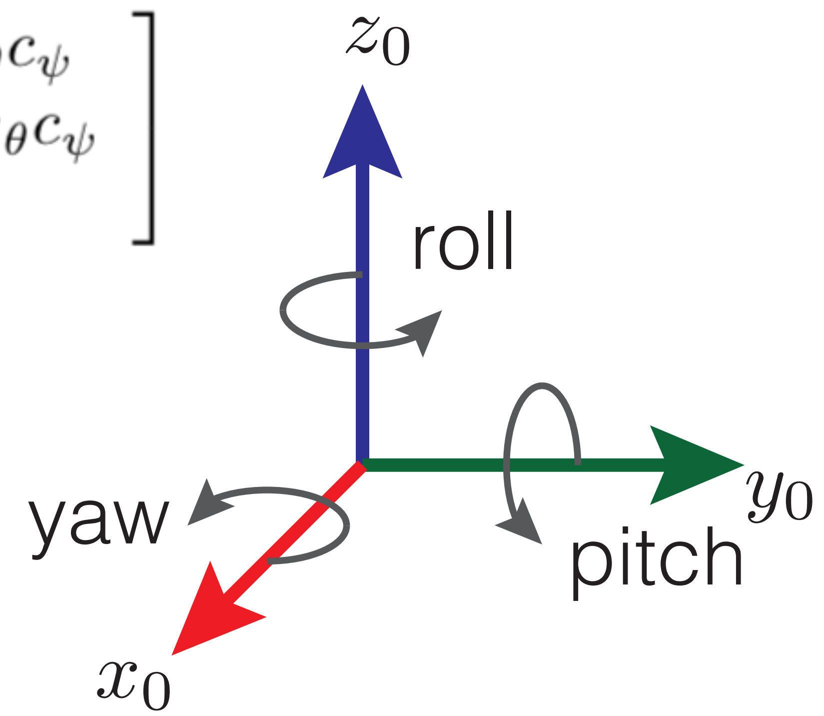
Rotate around current Z

Euler Angles

$$
\begin{aligned}
R_{ZYZ} &= R_{z,\phi} R_{y,\theta} R_{z,\psi} \\
&= \begin{bmatrix} c_\phi & -s_\phi & 0 \\ s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\phi c_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta \\ s_\phi c_\theta c_\psi + c_\phi s_\psi & -s_\phi c_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta \\ -s_\theta c_\psi & s_\theta s_\psi & c_\theta \end{bmatrix}
\end{aligned}
$$

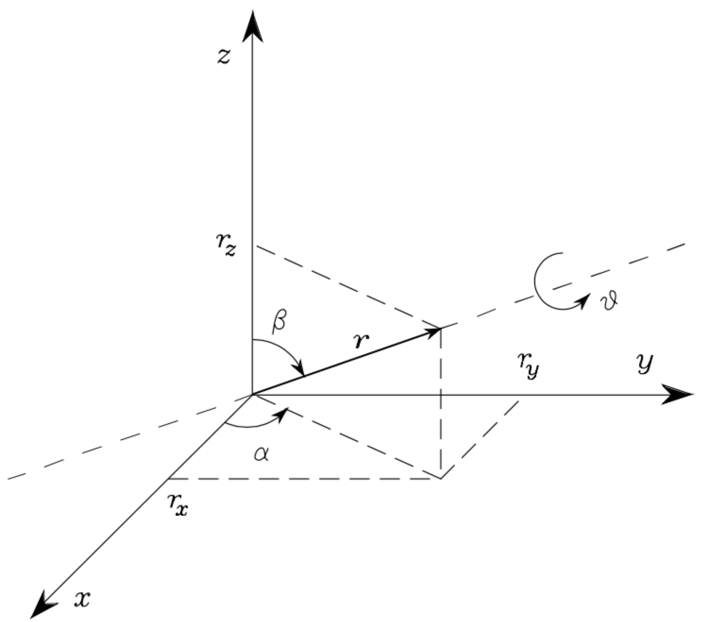
Tait-Bryan (Roll, Pitch, Yaw)

$$
\begin{aligned}
R_{XYZ} &= R_{z,\phi} R_{y,\theta} R_{x,\psi} \\
&= \begin{bmatrix} c_\phi & -s_\phi & 0 \\ s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\psi & -s_\psi \\ 0 & s_\psi & c_\psi \end{bmatrix} \\
&= \begin{bmatrix} c_\phi c_\theta & -s_\phi c_\psi + c_\phi s_\theta s_\psi & s_\phi s_\psi + c_\phi s_\theta c_\psi \\ s_\phi c_\theta & c_\phi c_\psi + s_\phi s_\theta s_\psi & -c_\phi s_\psi + s_\phi s_\theta c_\psi \\ -s_\theta & c_\theta s_\psi & c_\theta c_\psi \end{bmatrix}
\end{aligned}
$$



Angle-Axis Representation

Rotation around arbitrary axis:



$$\boldsymbol{R}(\vartheta, \boldsymbol{r}) = \boldsymbol{R}_z(\alpha)\boldsymbol{R}_y(\beta)\boldsymbol{R}_z(\vartheta)\boldsymbol{R}_y(-\beta)\boldsymbol{R}_z(-\alpha)$$

$$\sin\alpha = \frac{r_y}{\sqrt{r_x^2 + r_y^2}} \qquad \cos\alpha = \frac{r_x}{\sqrt{r_x^2 + r_y^2}}$$

$$\sin\beta = \sqrt{r_x^2 + r_y^2} \qquad \cos\beta = r_z.$$

$$\boldsymbol{R}(\vartheta, \boldsymbol{r}) = \begin{bmatrix} r_x^2(1-c_\vartheta)+c_\vartheta & r_x r_y(1-c_\vartheta)-r_z s_\vartheta & r_x r_z(1-c_\vartheta)+r_y s_\vartheta \\ r_x r_y(1-c_\vartheta)+r_z s_\vartheta & r_y^2(1-c_\vartheta)+c_\vartheta & r_y r_z(1-c_\vartheta)-r_x s_\vartheta \\ r_x r_z(1-c_\vartheta)-r_y s_\vartheta & r_y r_z(1-c_\vartheta)+r_x s_\vartheta & r_z^2(1-c_\vartheta)+c_\vartheta \end{bmatrix}.$$

Unit Quaternions

$$\mathcal{Q} = \{\eta, \boldsymbol{\epsilon}\} \quad \eta = \cos\frac{\vartheta}{2} \quad \boldsymbol{\epsilon} = \sin\frac{\vartheta}{2}\boldsymbol{r}$$

$$\eta^2 + \epsilon_x^2 + \epsilon_y^2 + \epsilon_z^2 = 1$$

$$\boldsymbol{R}(\eta, \boldsymbol{\epsilon}) = \begin{bmatrix} 2(\eta^2 + \epsilon_x^2) - 1 & 2(\epsilon_x\epsilon_y - \eta\epsilon_z) & 2(\epsilon_x\epsilon_z + \eta\epsilon_y) \\ 2(\epsilon_x\epsilon_y + \eta\epsilon_z) & 2(\eta^2 + \epsilon_y^2) - 1 & 2(\epsilon_y\epsilon_z - \eta\epsilon_x) \\ 2(\epsilon_x\epsilon_z - \eta\epsilon_y) & 2(\epsilon_y\epsilon_z + \eta\epsilon_x) & 2(\eta^2 + \epsilon_z^2) - 1 \end{bmatrix}$$

Multiplying Quaternions:

$$\boldsymbol{R}_1\boldsymbol{R}_2 \longrightarrow \mathcal{Q}_1 * \mathcal{Q}_2 = \{\eta_1\eta_2 - \boldsymbol{\epsilon}_1^T\boldsymbol{\epsilon}_2, \eta_1\boldsymbol{\epsilon}_2 + \eta_2\boldsymbol{\epsilon}_1 + \boldsymbol{\epsilon}_1 \times \boldsymbol{\epsilon}_2\}$$

Homogeneous Transformation

- Matrix representation of a rigid body motion
- Points and Vectors must be represented in Homogeneous Coordinates
- Contains the rotation and translation in single matrix by embedding in higher dimension

$$H = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix}; R \in SO(3),\ d \in \mathbb{R}^3 \qquad p = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Basic Transformations

$$\mathrm{Trans}_{x,a} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad Rot_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha & 0 \\ 0 & s_\alpha & c_\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathrm{Trans}_{y,b} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad Rot_{y,\beta} = \begin{bmatrix} c_\beta & 0 & s_\beta & 0 \\ 0 & 1 & 0 & 0 \\ -s_\beta & 0 & c_\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathrm{Trans}_{z,c} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad Rot_{\mathrm{z},\gamma} = \begin{bmatrix} c_\gamma & -s_\gamma & 0 & 0 \\ s_\gamma & c_\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Inverse Rigid Body Transformation

- To find the inverse, we find the transpose of the rotation part of the matrix
- We also reverse the displacement through multiplication with the inverse rotation

$$H^{-1} = \begin{bmatrix} R^T & -R^T d \\ 0 & 1 \end{bmatrix}$$

$$H^{-1}H = \begin{bmatrix} R^T & -R^T d \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R^T R & R^T d - R^T d \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix} = I.$$

$$HH^{-1} = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R^T & -R^T d \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} RR^T & -RR^T d + d \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix} = I.$$

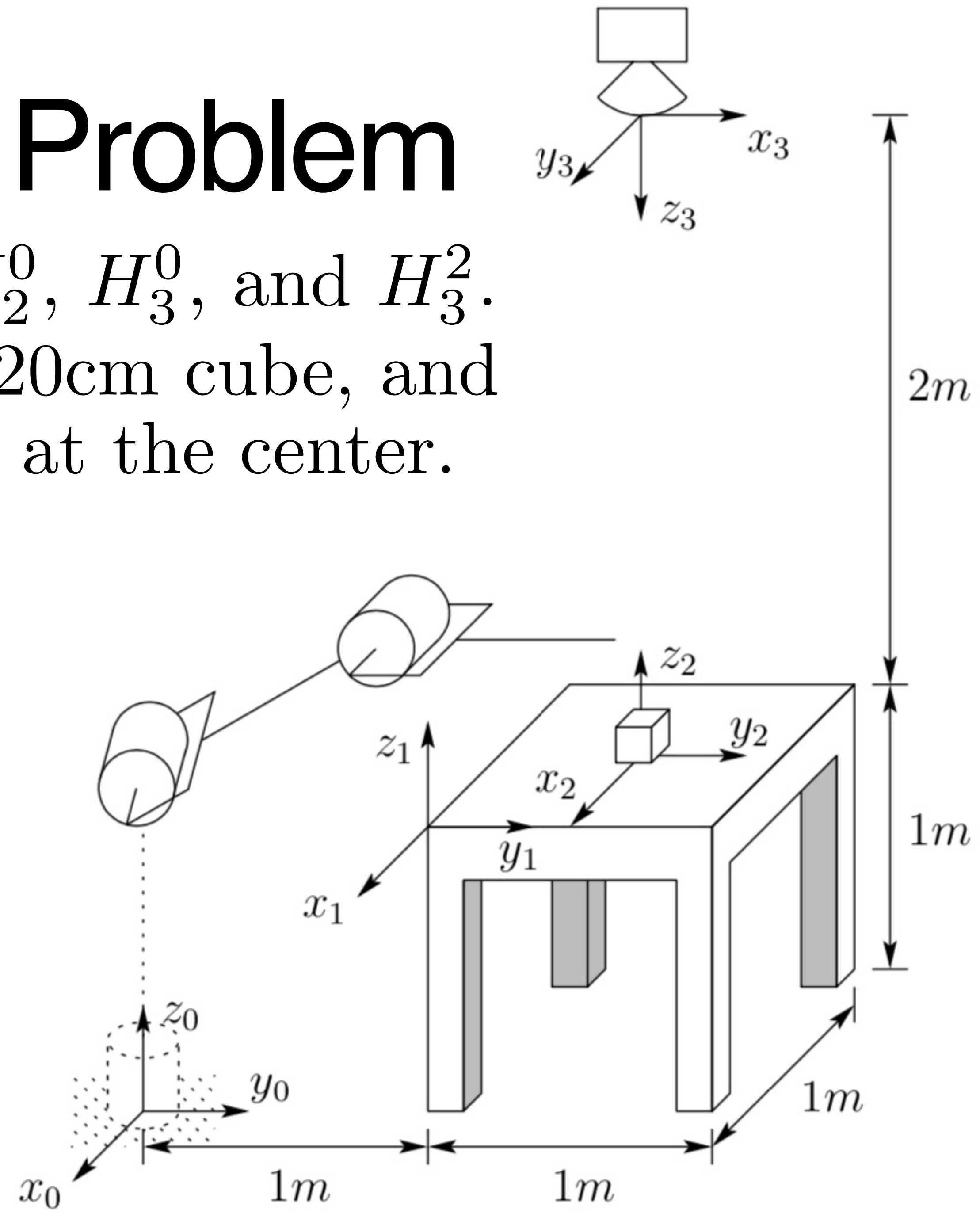
Example Homogeneous Transformation

Example 2.8 *The homogeneous transformation matrix H that represents a rotation of α degrees about the current x-axis followed by a translation of b units along the current x-axis, followed by a translation of d units along the current z-axis, followed by a rotation of θ degrees about the current z-axis, is given by*

$$
\begin{aligned}
H &= Rot_{x,\alpha} Trans_{x,b} Trans_{z,d} Rot_{z,\theta} \qquad (2.100) \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha & 0 \\ 0 & s_\alpha & c_\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} 1 & 0 & 0 & b \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} c_\theta & -s_\theta & 0 & 0 \\ s_\theta & c_\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} c_\theta & -s_\theta & 0 & b \\ c_\alpha s_\alpha & c_\alpha c_\theta & -s_\alpha & -s_\alpha d \\ s_\alpha s_\theta & s_\alpha c_\theta & c_\alpha & c_\alpha d \\ 0 & 0 & 0 & 1 \end{bmatrix}.
\end{aligned}
$$

Worked Problem

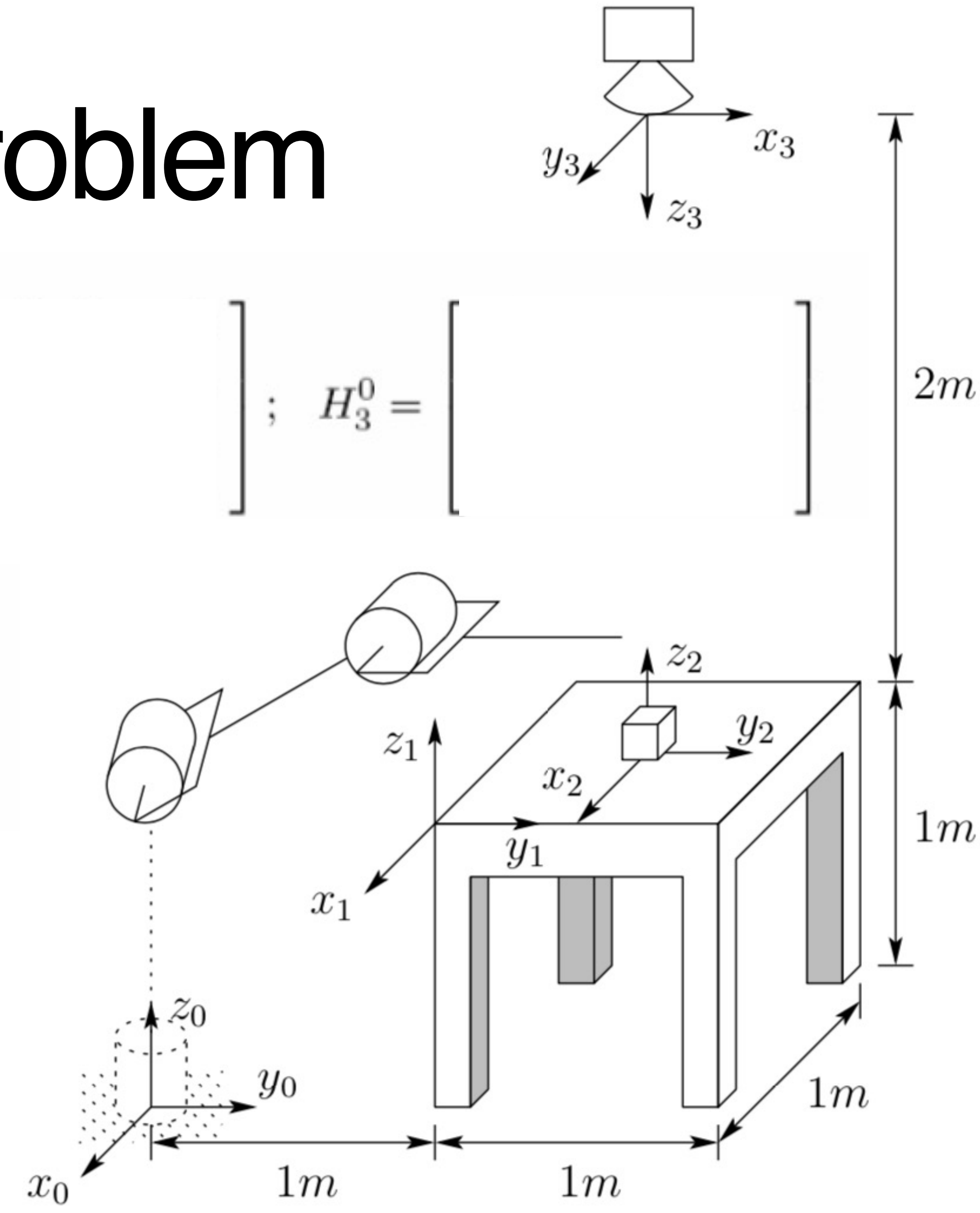
Find H_1^0, H_2^0, H_3^0, and H_3^2. The block is a 20cm cube, and the origin o_2 is at the center.



Worked Problem

$$H_1^0 = \begin{bmatrix} & & & \\ & & & \\ & & & \\ & & & \end{bmatrix}; \quad H_2^0 = \begin{bmatrix} & & & \\ & & & \\ & & & \\ & & & \end{bmatrix}; \quad H_3^0 = \begin{bmatrix} & & & \\ & & & \\ & & & \\ & & & \end{bmatrix}$$

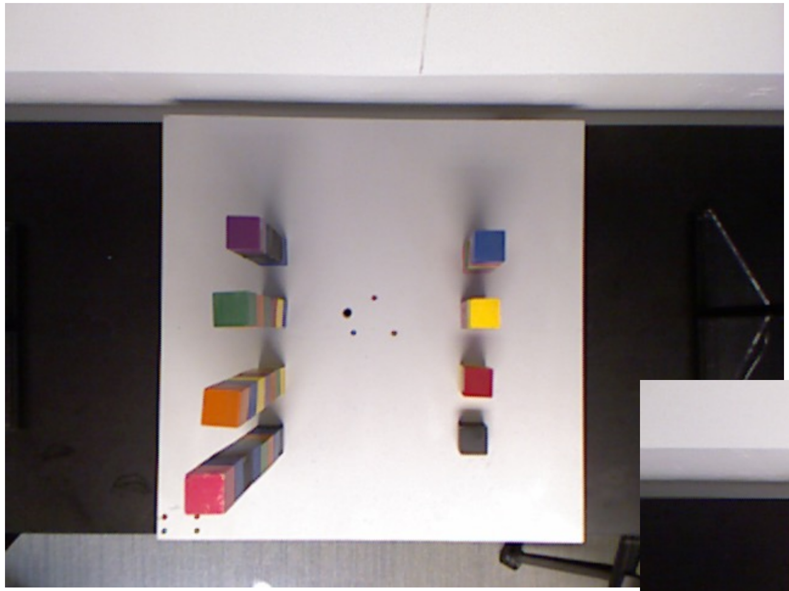
$$H_3^2 = \begin{bmatrix} & & & \\ & & & \\ & & & \\ & & & \end{bmatrix}$$



Transformations for

Given an object detected in an image,
what is its location in workspace coordinates?

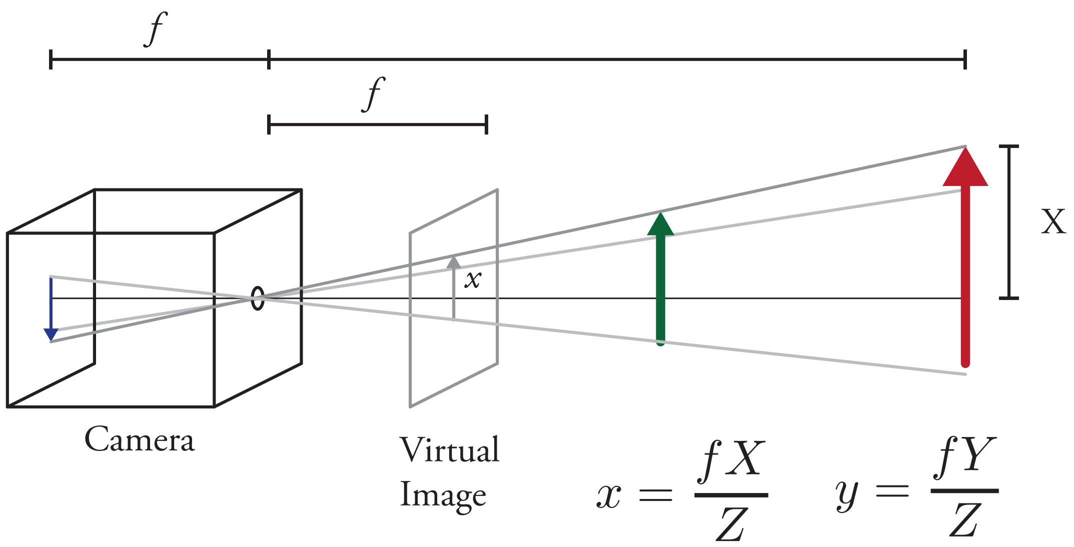
$$\begin{bmatrix} u \\ v \end{bmatrix} \rightarrow \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$



Let's start by looking at
how an image is formed.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \begin{bmatrix} u \\ v \end{bmatrix}$$

Pinhole Model



$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \frac{1}{Z_c} \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix} \left[\mathbf{I} \quad | \quad 0 \right] \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

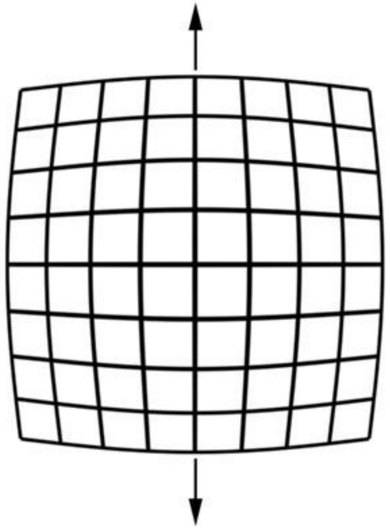
Camera intrinsic matrix

- Transforms points in Camera Frame to pixel coordinates
- Depends on geometry of camera, physical characteristics of sensor, details of the image digitization

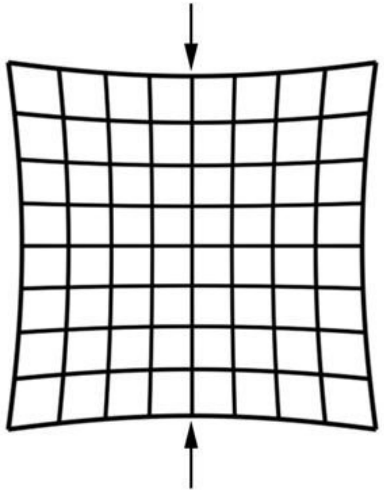
$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z_c} \begin{bmatrix} \alpha & s & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & | & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

Lens Distortion

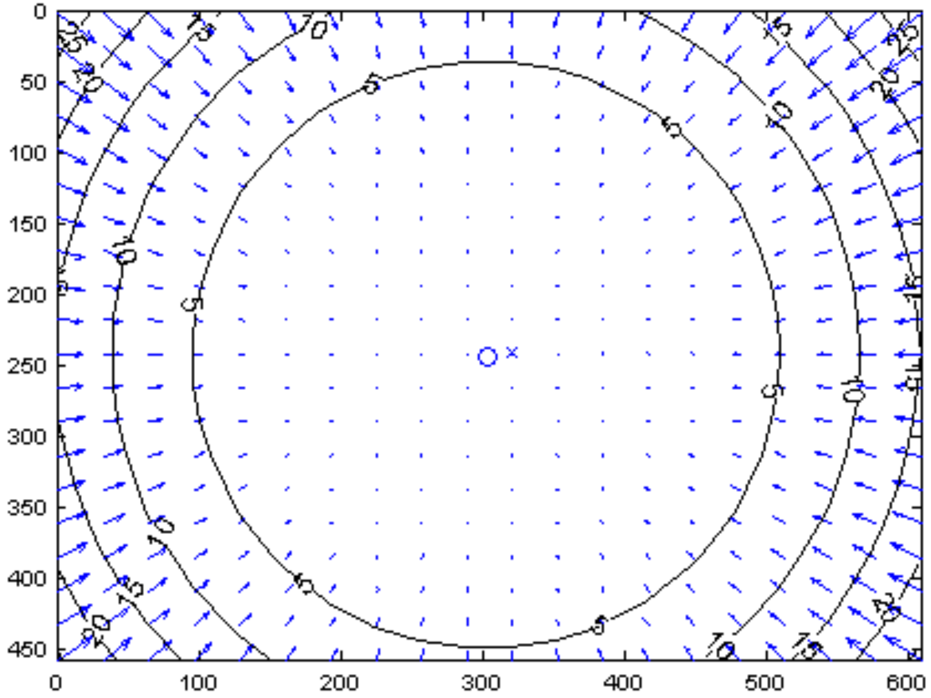
- Real cameras can have distortion
- Can correct with polynomial model
- cv2.undistort()



Barrel distortion



Pincushion distortion



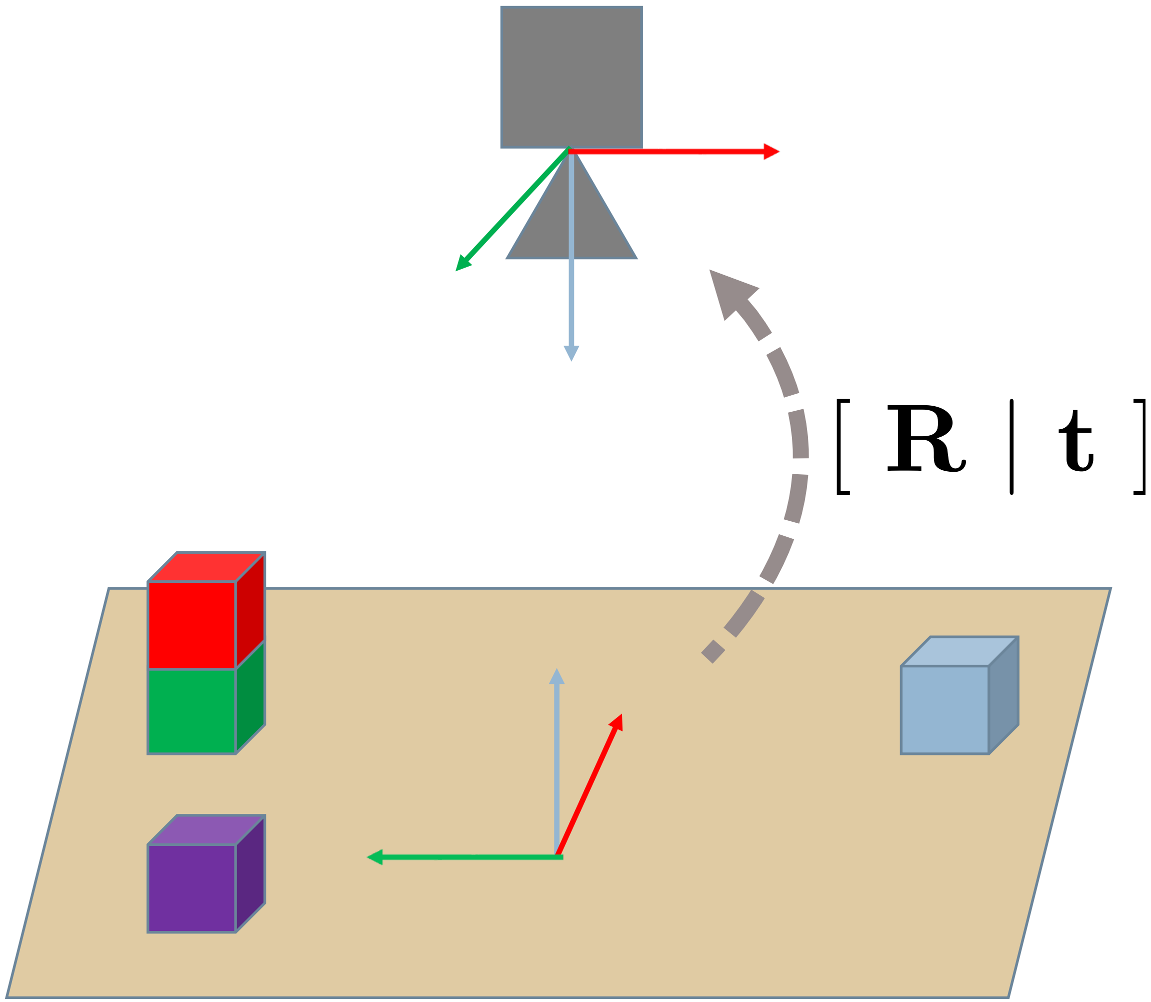
$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u_d \\ v_d \end{bmatrix} (1 + k_1 r^2 + k_2 r^4)$$

ROS /camera_info

- The Realsense node publishes the intrinsic matric on the camera_info topic.
- This intrinsic matrix comes from factory calibration.
- May be more accurate than we can measure with checkerboard in lab.
- Does not include distortion parameters, but these may be minimal.

Camera Extrinsics

- We need the transformation to take points in the world frame and put them in the camera frame
- Carefully position camera and measure (naïve)
- Use fiducials (Apriltags)
- Calculate from pixel/point correspondances (cv2.solvePnP)



Full Camera Matrix
Workspace to Camera

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\mathbf{P} = \underbrace{\mathbf{K}}_{\text{intrinsic}} [\, \underbrace{\mathbf{R} \mid \mathbf{t}}_{\text{extrinsic}} \,]$$

intrinsic

extrinsic

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z_c} \begin{bmatrix} \alpha & s & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \overset{\text{projection}}{[\mathbf{I} \quad | \quad 0]} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

Camera to Image Frame (Intrinsic)

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} & \mathbf{R} & & \mathbf{T} \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

World to Camera (Extrinsic)

Pixel to World frame

- With depth information we can undo the scale by $1/Z_c$

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = Z_c(u, v)\ \mathbf{K}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

- Change to homogeneous coordinates and multiply inverse extrinsic matrix

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \mathbf{H}^{-1} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$