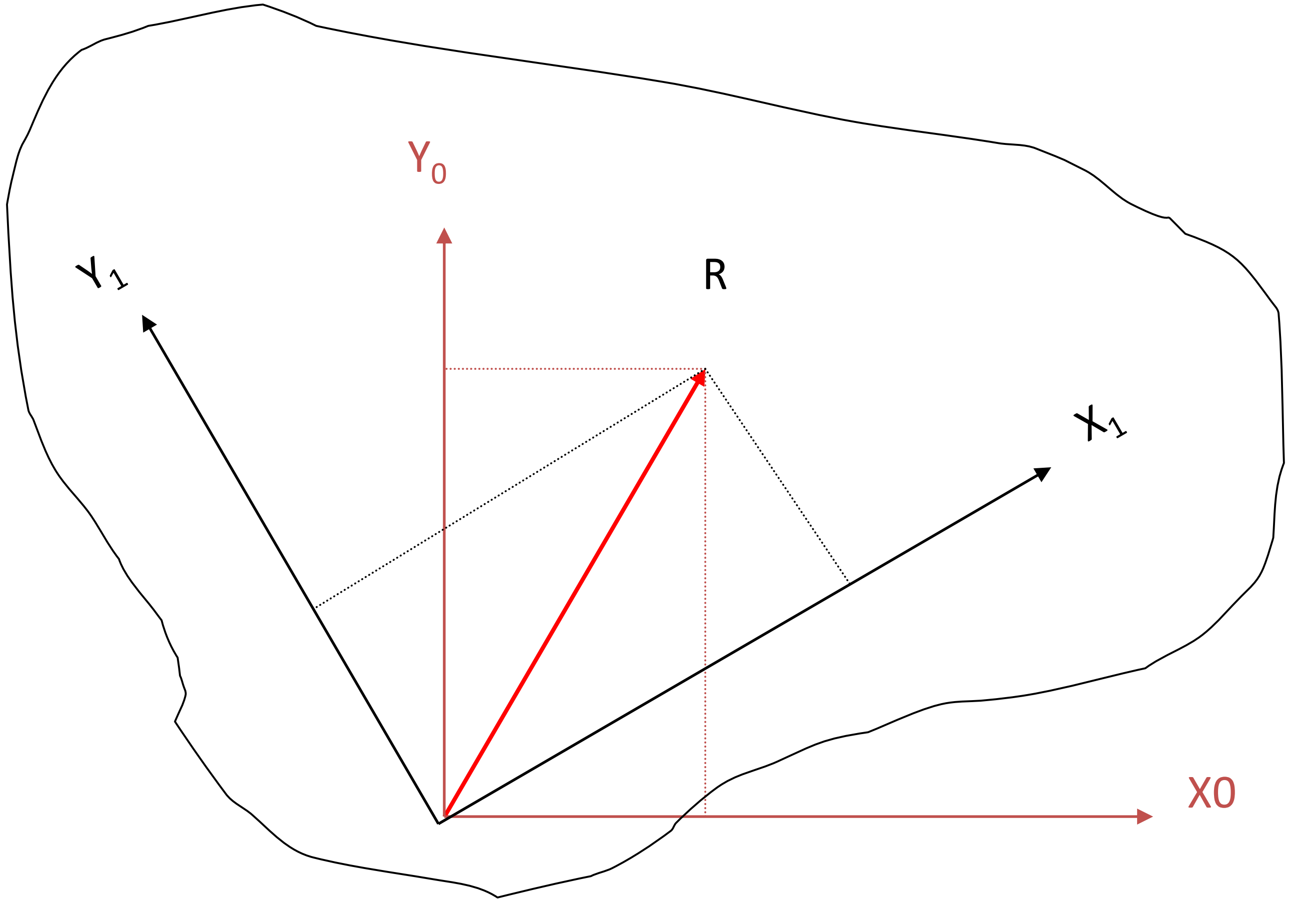
Computer vision
3D Transformations

Le Thanh Ha, Ph.D
Assoc. Prof. at University of Engineering and Technology,
Vietnam National University
ltha@vnu.edu.vn; lthavnu@gmail.com; 0983 692 592

Coordinate Frames



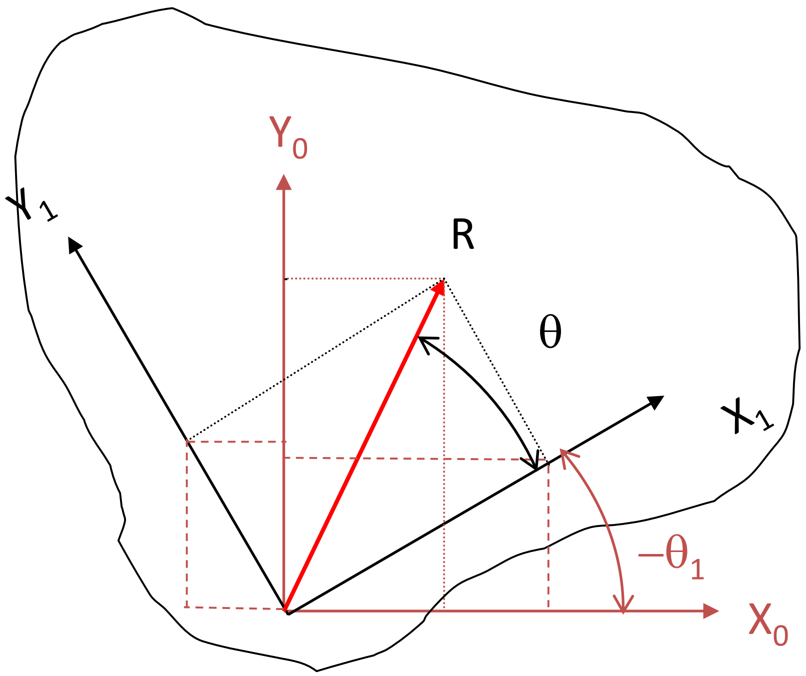
Coordinate Frames

- Interested in Some Point described by a Vector R
- We need to express R which we know in Frame 1 in Frame 0

$$x_1 = R \cdot \sin(\theta)$$
$$y_1 = R \cdot \cos(\theta)$$

$$x_0 = x_1 \cdot \cos(\theta_1) - y_1 \cdot \sin(\theta_1)$$
$$y_0 = x_1 \cdot \sin(\theta_1) + y_1 \cdot \cos(\theta_1)$$

Equation is independent of values



Coordinate Frames

- Now lets put it in matrix form

$$x_0 = x_1 \cdot \cos(\theta_1) - y_1 \cdot \sin(\theta_1)$$
$$y_0 = x_1 \cdot \sin(\theta_1) + y_1 \cdot \cos(\theta_1)$$

$$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

$$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = [T] \cdot \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

- So what if we want to map the other way?

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = [T]^{-1} \cdot \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

- What is the inverse of T? Why?

Coordinate Frames

- If we look at the columns and rows of T we see that they have a norm of one.
- Also if we take the dot product of the columns we find they are orthogonal to each other.
- So T is an ortho-normal Matrices. Thus its transpose is its inverse.

$$[T] = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) \end{bmatrix} \qquad [T]^{-1} = \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) \\ -\sin(\theta_1) & \cos(\theta_1) \end{bmatrix}$$

- This was a simple 2DOF example what about 3.
- If we project a Z axes out the plane generated by the X and Y axes, then a rotation around the Z axes will not affect the Z position of the vector R.

Coordinate Frames

- The 3D transformation axes about the Z axes is:

$$[T]_z = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Similarly for rotations around the X or Y axes we get

$$[T]_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \qquad [T]_Y = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix}$$

- Shorthand notation is often used
  $\cos(\xi)=c\,\xi$ $\sin(\xi)=s\,\xi$

Coordinate Transformations

- For multiple transformation we simply multiply by more matrices.
- If we have multiple rotations about the same axes we can just add the angles of the matrices. Does this make sense.

$$[T(\gamma+\phi)]_z = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\phi)\cdot\cos(\gamma) - \sin(\phi)\cdot\sin(\gamma) & -\cos(\phi)\cdot\sin(\gamma) - \sin(\phi)\cdot\cos(\gamma) + & 0 \\ \sin(\phi)\cdot\cos(\gamma) + \cos(\phi)\cdot\sin(\gamma) & -\sin(\phi)\cdot\sin(\gamma) + \cos(\phi)\cdot\cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\sin(\phi+\gamma) = \sin(\phi)\cos(\gamma) + \cos(\phi)\sin(\gamma)$$
$$\cos(\phi+\gamma) = \cos(\phi)\cos(\gamma) - \sin(\phi)\sin(\gamma)$$

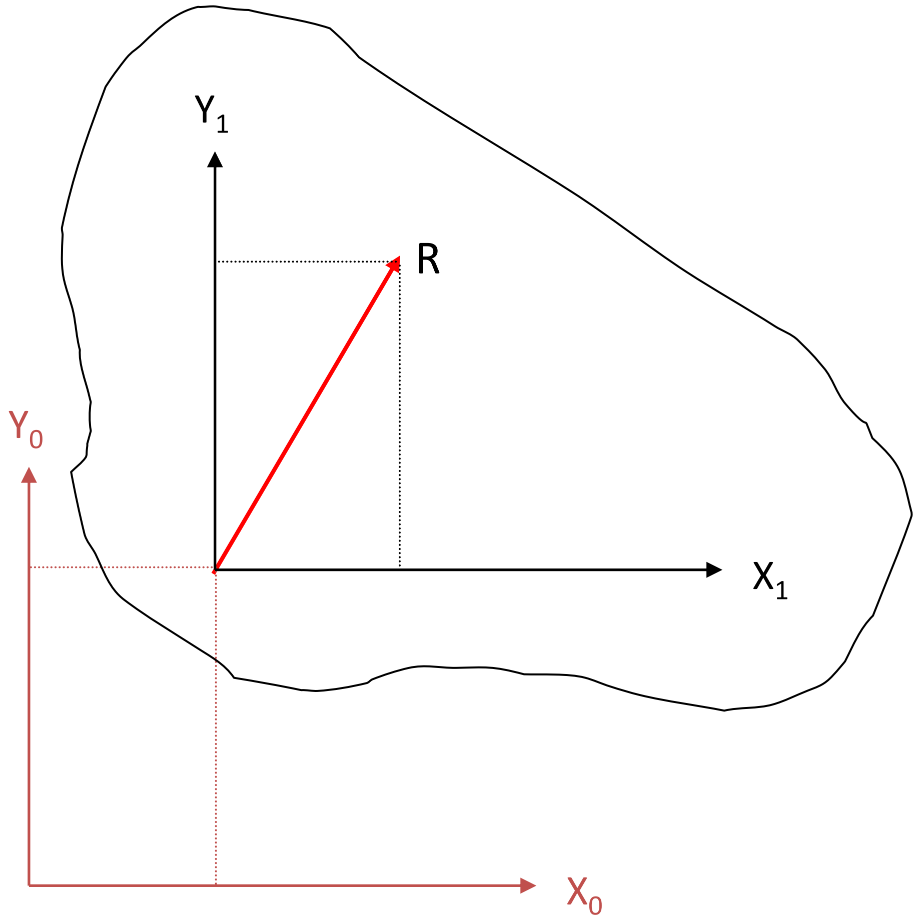
$$[T(\gamma+\phi)]_z = \begin{bmatrix} \cos(\phi+\gamma) & -\sin(\phi+\gamma) & 0 \\ \sin(\phi+\gamma) & \cos(\phi+\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Substituting we get

Coordinate Transformation

- So now we can express a vector in a reference frame rotated to an arbitrary angle in space.
- What if we want to express it in a translated frame

Translating Coordinate Frames



Translating Coordinate Frames

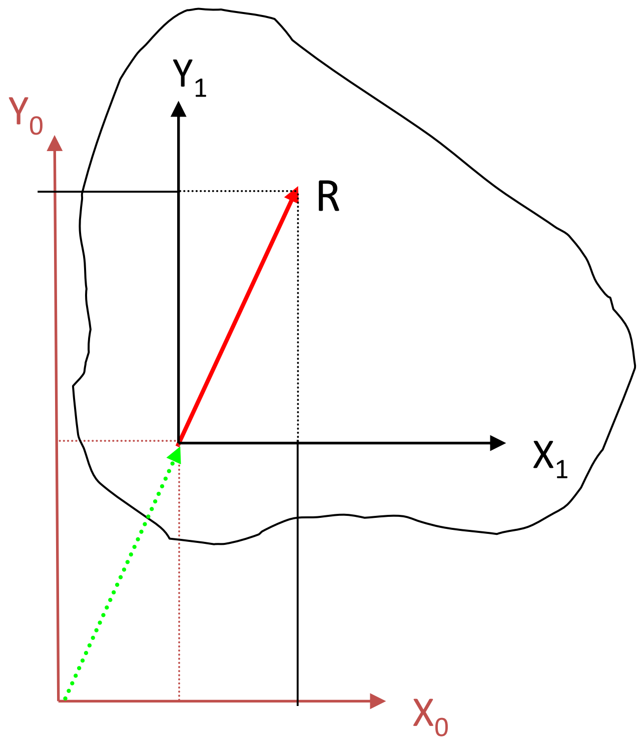
- Now we can translate and rotate.

$$x_0 = x_1 + \Delta x$$

$$y_0 = y_1 + \Delta y$$

- Or in terms of vectors

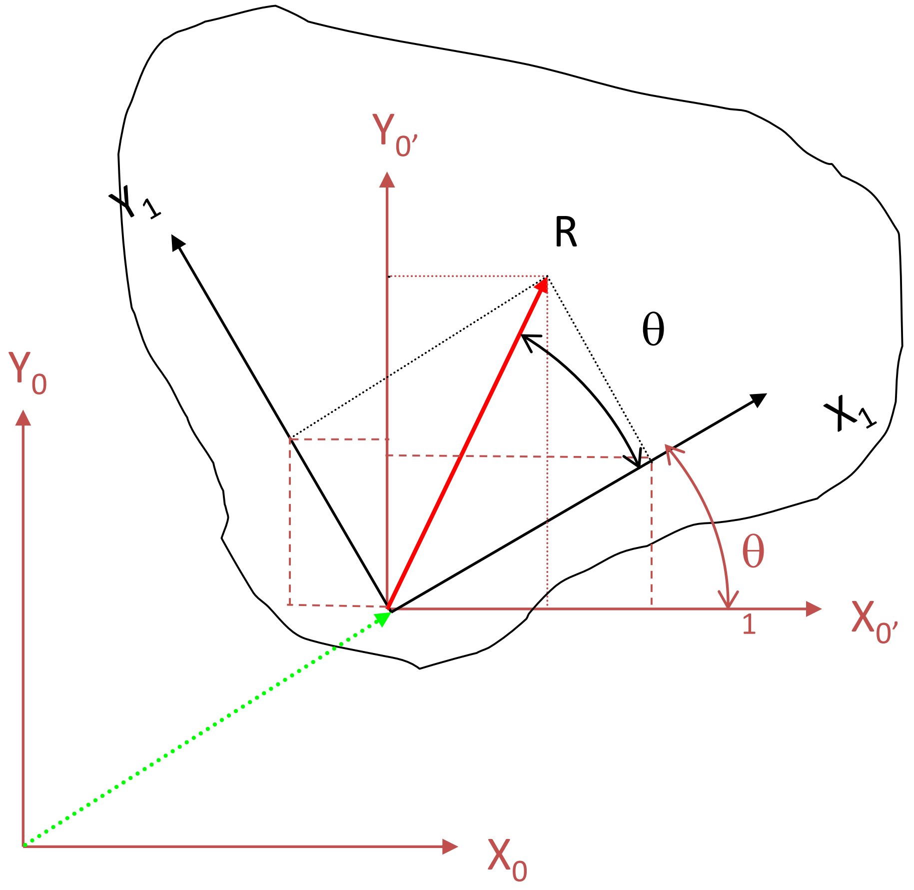
$$R_0 = R_1 + \Delta R$$



What if we have a rotation and a translation

- First we can rotate the frames
- then we can translate

$$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = [T] \cdot \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$



Homogeneous Transformation Matrices

- 3x3 Rotation Matrix

$$T_1 = \begin{bmatrix} C1 & -S1 & 0 \\ S1 & C1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- 3x1 Displacement Vector

$$R_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$

- For a displacement and a rotation

$$[R_0] = [T] \cdot [R_1] + [\Delta R]$$

4x4 Homogeneous Matrix

- If we want to perform a rotation and a translation with one operation

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = [T] \cdot \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \qquad T = \begin{bmatrix} C1 & -S1 & 0 \\ S1 & C1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- We can create a homogeneous Transformation Matrix

$$T_H = \left[\begin{array}{ccc:c} C1 & -S1 & 0 & \Delta x_0 \\ S1 & C1 & 0 & \Delta y_0 \\ 0 & 0 & 1 & \Delta z_0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] \qquad \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 0 \end{bmatrix} = [T_H] \cdot \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 0 \end{bmatrix}$$

Homogeneous Transformation Matrices

- What does a pure translation look like

$$T_T = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- What does a pure rotation look like

$$T_R = \begin{bmatrix} C1 & -S1 & 0 & 0 \\ S1 & C1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Multiply to move from one coordinate to another.

D-H Parameters

- Denavit-Hartenberg (D-H) are used to describe a robot link.
- One Coordinate System is created for each link.
  - Each Axes is orthogonal
  - Use the right hand rule
- Link are assumed to be rigid
- Four Parameters are used
  - d_n Link Offset
  - a_n Link Length (Common Normal Distance)
  - θ_n Joint Angle
  - α_n Link Twist Angle

Notation

- Coordinate systems are represented with brackets {B}, {0}, etc.
- Vectors
  - Lets Look at a Vector P Described in Frame A
  - Leading Subscript describes the frame in which the Vector is described or Referenced
  - Individual Elements of a vector are described by a trailing subscript

Frame of Reference

$$^{A}P = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

Matrix Notation

New Frame of Reference

$^{A}_{B}R$ $\qquad\qquad$ $^{A}_{B}T$

Original Frame of Reference

Homogenous Transformations Represent 3 Things

- Describe a Frame
- Map from one Frame to another
- Act as an Operator to move within a Frame

$$ {}^{A}_{B}T $$

HMI

HUMAN MACHINE INTERACTION LABORATORY

Transforms Describe Frames

- Frames can be described by A Homogenous Transformation Matrices

$$ {}^{A}_{B}T = \left[\begin{array}{ccc|c} & & & {}^{A}P_{BorgX} \\ & {}^{A}_{B}R & & {}^{A}P_{BorgY} \\ & & & {}^{A}P_{BorgZ} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] $$

- Description of Frame
  - Columns of ${}^{A}_{B}R$ are the Unit Vectors defining the directions of the principle axes of {B} in terms of {A}

$$ {}^{A}_{B}R = \begin{bmatrix} {}^{A}\hat{X}_{B} & {}^{A}\hat{Y}_{B} & {}^{A}\hat{Z}_{B} \end{bmatrix} = \begin{bmatrix} {}^{B}\hat{X}_{A} \\ {}^{B}\hat{Y}_{A} \\ {}^{B}\hat{Z}_{A} \end{bmatrix} $$

  - Rows of ${}^{A}_{B}R$ are the Unit Vectors defining the directions of the principle axes of {A} in terms of {B}
  - ${}^{A}P_{Borg}$ is the location of the origin of {B} in terms or {A}

Mapping Between Frames

- Maps vector from Frame {B} to Frame {A}

- ${}^{A}_{B}R$ will rotate a vector to project its components originally described in {B} in the {A} of Frame

$$
{}^{A}_{B}T = \left[\begin{array}{ccc|c} & & & {}^{A}P_{BorgX} \\ & {}^{A}_{B}R & & {}^{A}P_{BorgY} \\ & & & {}^{A}P_{BorgZ} \\ \hline 0 & 0 & 0 & 1 \end{array}\right]
$$

- ${}^{A}P_{Borg}$ will translate the vector to adjust its origin from frame {B} to its new origin in {A}

$$
{}^{B}P \mapsto {}^{A}P
$$

HMI

HUMAN MACHINE INTERACTION LABORATORY

Acts as an Operator within a coordinate frame

- Operator will rotate and translate a vector
- R Defines the angle to rotate about
- P_{Borg} Defines the distance to translate
- Usually drop subscripts

$$T = \left[\begin{array}{ccc|c} & & & P_{2orgX} \\ & R & & P_{2orgY} \\ & & & P_{2orgZ} \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

- T Operates on ${}^{A}P_1$ to create ${}^{A}P_2$