Course Diagram

3D Transformations | Dead Reckoning | Basic Probability

Bayesian Inference | Extended Kalman/Information Filter

Projective camera geometry | Multi View Geometry | Feature Matching

Bundle Adjustment | VAN, SLAM | Graphical models

Incremental Smoothing and Mapping (iSAM)

Advanced topics (subject to progress in class)

Multi-Robot SLAM & VAN | Belief Space Planning

Objectives of this Lecture

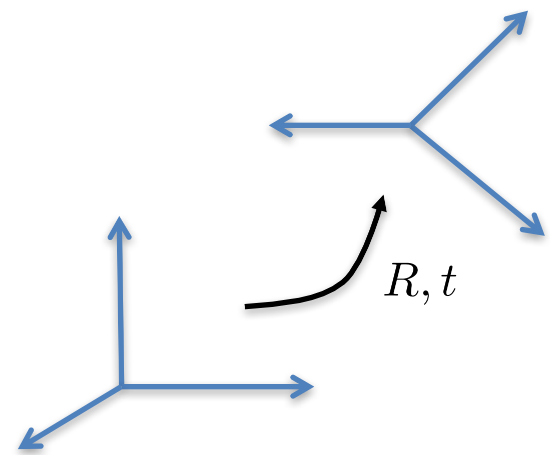
- Learn 3D transformations between different coordinate frames
- Learn different orientation parameterizations
- Use homogenous coordinates and 6 DOF pose/frame

Outline

- 6 DOF Pose
- 3D Transformations
- Orientation Parameterizations
- Navigation State

6 DOF Frame, Pose

- 6 degree of freedom (DOF) frame, pose:
  - Defines a coordinate frame relative to another coordinate reference frame
  - 3D rigid transformation
    - rotation R
    - translation t



- Also known as Euclidean transformation (special Euclidean group SE(3))

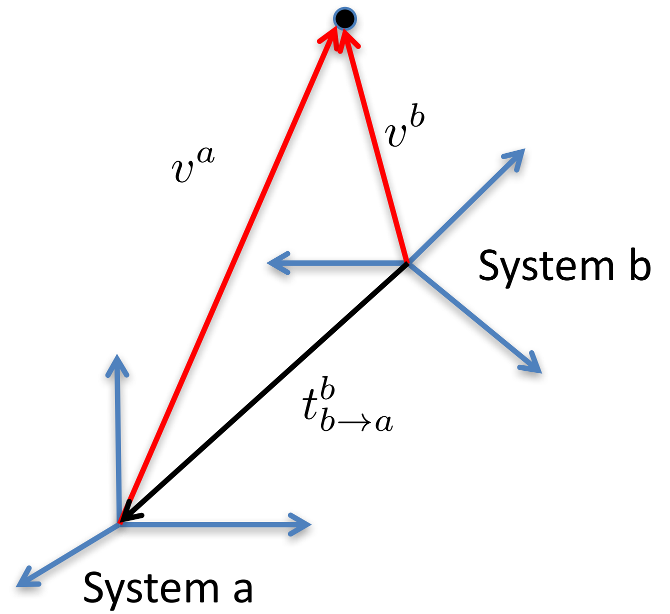
3D Rigid Transformations

- Euclidean transformation

$$v^b = Rv^a + t$$

- In detail:

$$v^b = R_a^b v^a + t_{b\to a}^b$$



R_a^b : rotation from a to b

$t_{b\to a}^b$: translation from b to a, expressed in system b
(origin of system a relative to system b)

3D Rigid Transformations

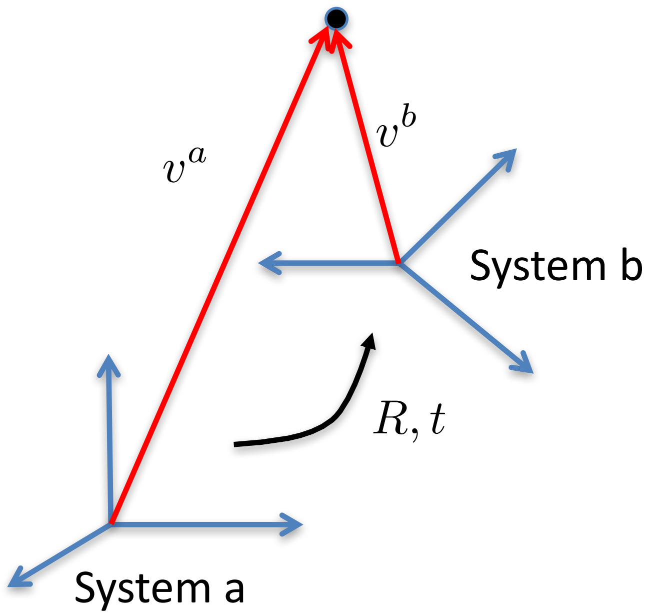
- Euclidean transformation

$$v^b = Rv^a + t$$

- In matrix notation:

$$\begin{pmatrix} v^b \\ 1 \end{pmatrix} = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \begin{pmatrix} v^a \\ 1 \end{pmatrix}$$

4x4 matrix



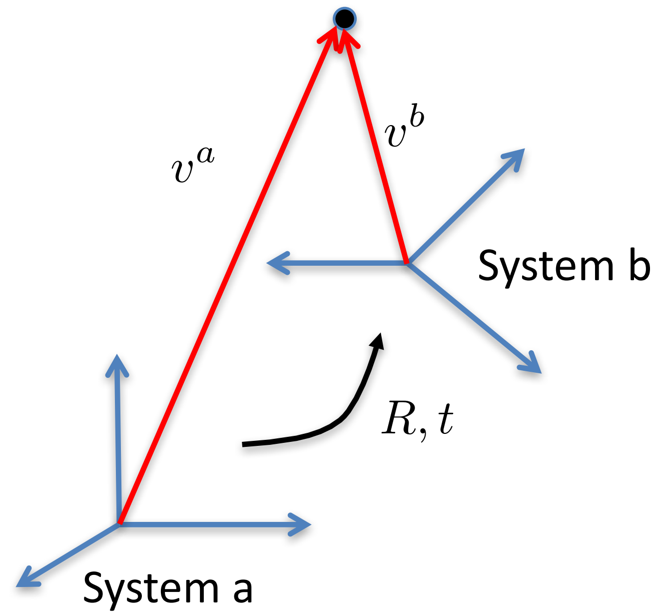
3D Rigid Transformations

- 3D point/vector: $\mathbf{v} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$
- Augmented coordinates: $\bar{\mathbf{v}} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$

$$\begin{pmatrix} v^b \\ 1 \end{pmatrix} = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \begin{pmatrix} v^a \\ 1 \end{pmatrix}$$

$$\Leftrightarrow$$

$$\bar{v}^b = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \bar{v}^a$$



- Common notations: ${}^b_aT$, ${}_bT_a$
(from system a to b)

3D Rigid (Euclidean) Transformation

- 6 degrees of freedom transformation:
  - translation t (3 DOFs)
  - rotation R (3 DOFs)

$$T = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

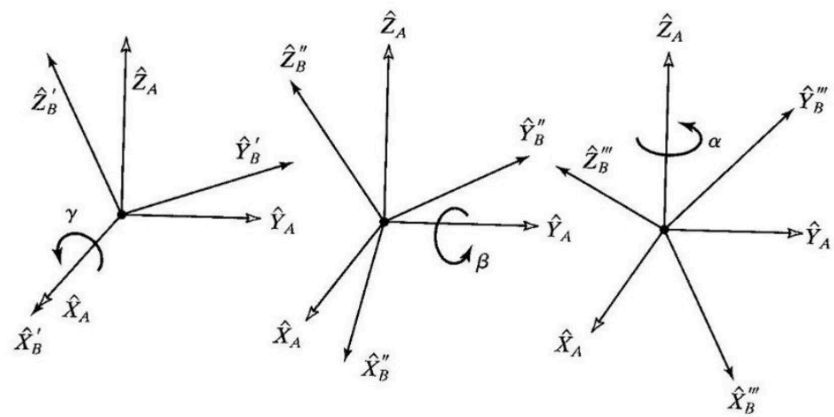
- Different parameterizations of 3D rotation exist (next slides)

Rotations

- Euler's Theorem:

  Any two independent orthonormal coordinate frames can be related by a sequence of rotations (not more than three) about coordinate axes, where no two successive rotations may be about the same axis.

Leonard Euler (1707-1783)



Orientation Parameterizations

- Rotation matrix
- Euler angles
- Euler vector
- Quaternion

Rotation Matrix

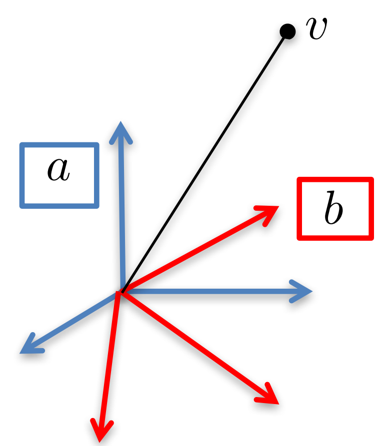
- Orthonormal 3x3 matrix

special orthogonal group

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \in SO(3)$$

- Columns correspond to coordinate axes
- Disadvantage: over-parameterization (9 parameters for 3 DOFs)
- R_a^b: rotation <u>from</u> system a <u>to</u> system b

$$v^b = R_a^b v^a$$



- Composition: $R_a^c = R_b^c R_a^b$
- Inverse: $R^{-1} \equiv R^T$

Euler Angles

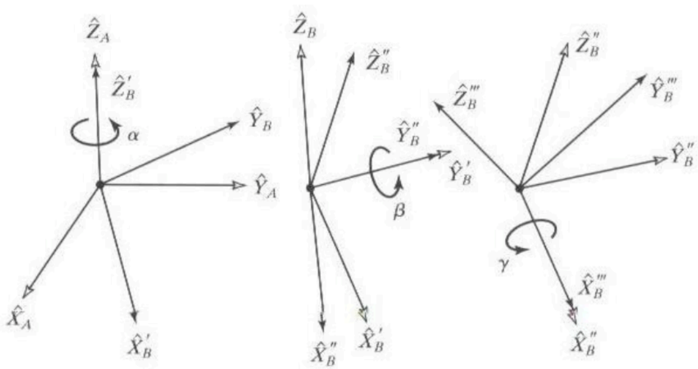
- Orientation is represented by 3 numbers
- Euler Angle Sequence: A sequence of rotations around principle axes
- There are 12 different sequences (without successive rotations about the same axis)
- Order does matter!

XYZ	XZY	XYX	XZX
YXZ	YZX	YXY	YZY
ZXY	ZYX	ZXZ	ZYZ

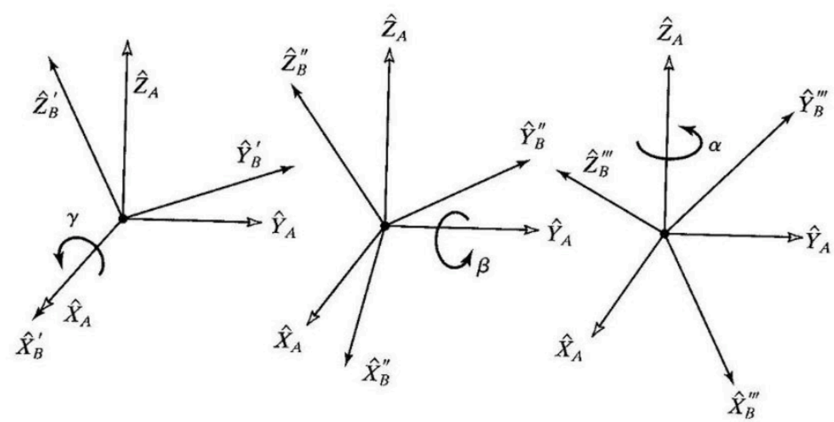
Euler Angles

- Orientation is represented by 3 numbers
- Euler Angle Sequence: A sequence of rotations around principle axes
- There are 12 different sequences (without successive rotations about the same axis)
- Order does matter!

ZYX



XYZ



From Euler Angles to Rotation Matrix

- Rotation matrices about principal axes:

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

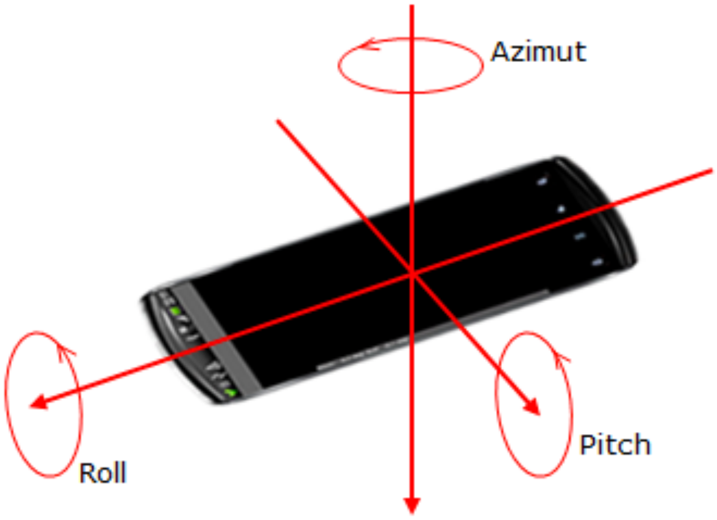
$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

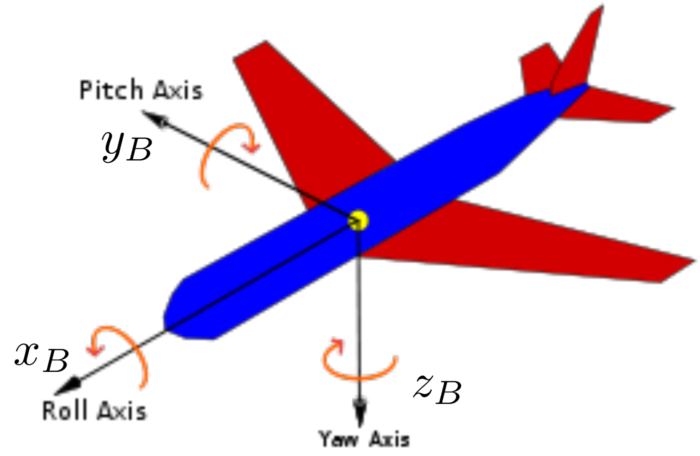
$$R_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- To represent orientation as a matrix, multiply a sequence of rotation matrices

From Euler Angles to Rotation Matrix

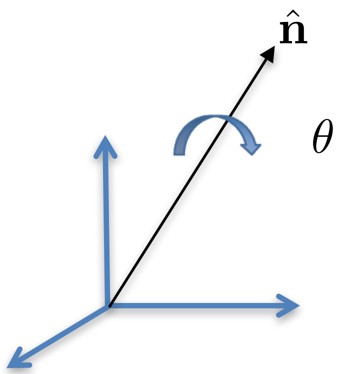
- No standard convention
- Common conventions
  - roll-pitch-yaw (rpy) $R_z(\psi)\, R_y(\theta)\, R_x(\phi)$
  - yaw-pitch-roll (ypr) $R_x(\phi)\, R_y(\theta)\, R_z(\psi)$





Euler Vector

- Euler's rotation theorem: any 3D rotation can be described by a single rotation about some axis



- $\hat{\mathbf{n}}$: axis of rotation, known as Euler axis
- θ : rotation angle

- Conversion to rotation matrix via Rodriguez' formula:

$$R\left(\hat{\mathbf{n}},\theta\right) = I + \sin\theta\,[\hat{\mathbf{n}}]_\times + (1-\cos\theta)\,[\hat{\mathbf{n}}]_\times^2$$

- Conversion from rotation matrix to Euler vector:

$$\theta = \cos^{-1}\left[\frac{1}{2}\left(\mathrm{trace}\left(R\right)-1\right)\right] \qquad \hat{\mathbf{n}} = \frac{1}{2\sin\theta}\begin{pmatrix} r_{32}-r_{23} \\ r_{13}-r_{31} \\ r_{21}-r_{12} \end{pmatrix}$$

Quaternions

- Invented by W.R. Hamilton in 1843
- A quaternion has 4 components

$$\mathbf{q} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}$$

- Quaternions are an extension to complex numbers
- Of the four components, one is a “real” scalar number, and the other three form a vector in an imaginary ijk space:

$$\mathbf{q} = q_0 + iq_1 + jq_2 + kq_3$$

with

$$i^2 = j^2 = k^2 = ijk = -1$$
$$i = jk = -kj$$
$$j = ki = -ik$$
$$k = ij = -ji$$

Unit Quaternions as Rotations

- Unit quaternion:

$$\mathbf{q} = \left[\begin{array}{cccc} q_0 & q_1 & q_2 & q_3 \end{array}\right]$$

$$\|\mathbf{q}\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1$$

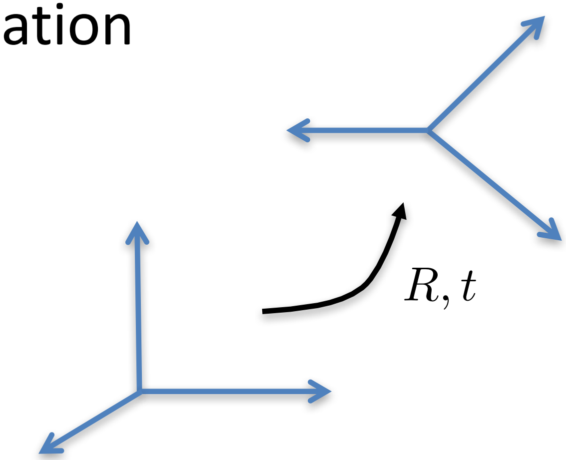
- A unit quaternion can represent a rotation by an angle θ around a unit axis $\mathbf{a}$:

$$\mathbf{q} = \left[\cos\frac{\theta}{2} \quad a_x \sin\frac{\theta}{2} \quad a_y \sin\frac{\theta}{2} \quad a_z \sin\frac{\theta}{2}\right]$$

- Can be transformed to a rotation matrix and vice versa

Back to 6 DOF Pose

- 6 DOF pose represents a 3D rigid transformation
  - Translation
  - Rotation



$$T = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- We will occasionally use the pose notation:

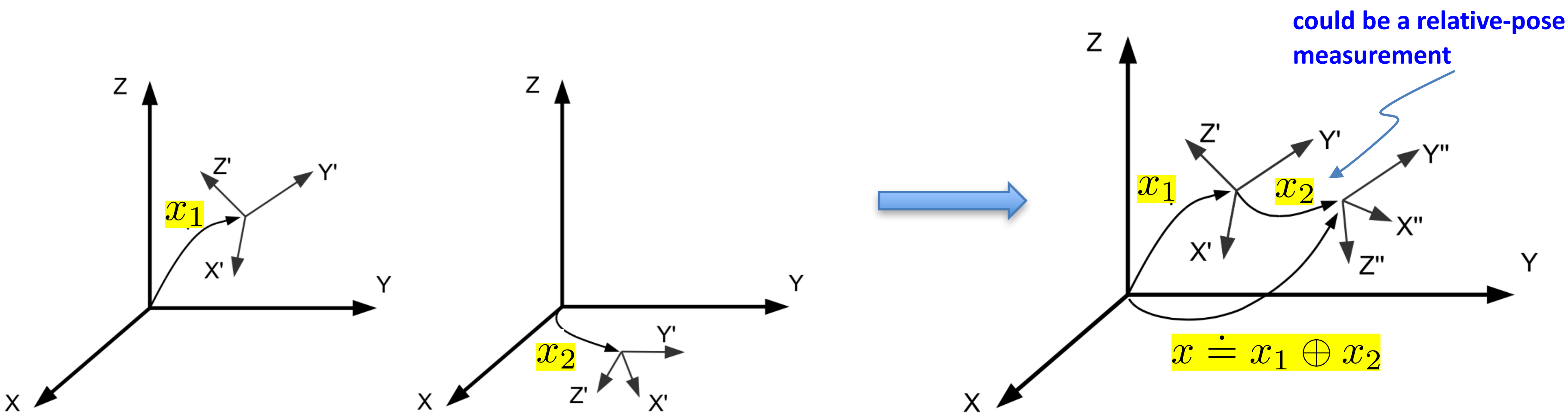
$$x = \{R, t\}$$

- Next:
  - Pose composition

Pose/Transformation Composition

Intuition

- Pose describes a transformation relative to some reference frame
- Consider two poses: x_1 and x_2
- Composition (notation $x_1 \oplus x_2$):
  - Calculate transformation by first following $x_1$and then x_2 (Consider x_2 being expressed relative to x_1)



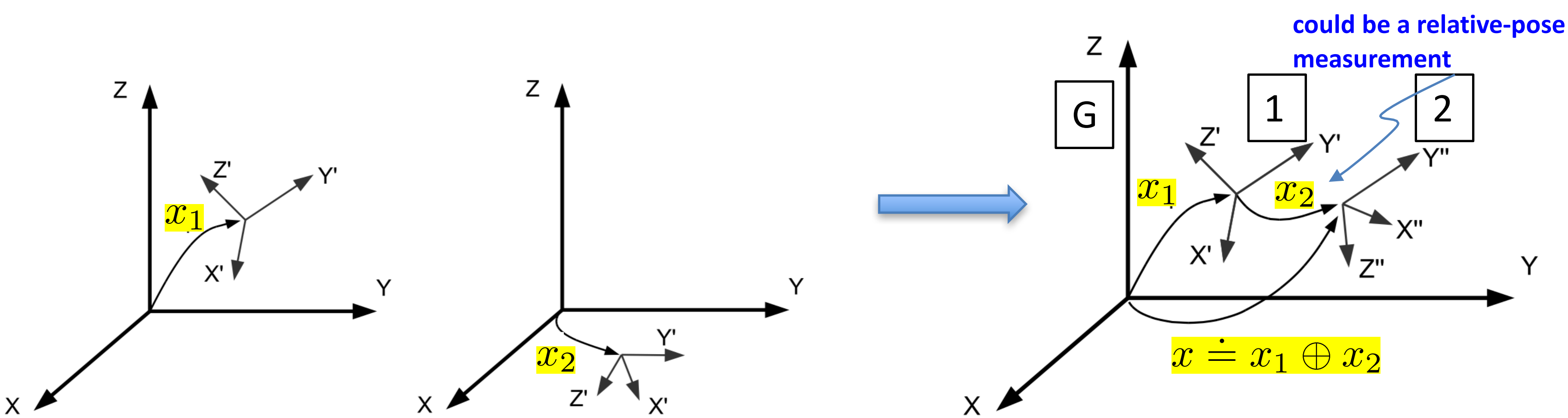
Diagrams adapted from Blanco14tr

Pose/Transformation Composition

More formally:

$$x_1 = \{R_1, t_1\}$$
$$x_2 = \{R_2, t_2\}$$

$$\Rightarrow \quad x \doteq x_1 \oplus x_2 = \{R_1 R_2, R_1 t_2 + t_1\}$$



Diagrams adapted from Blanco14tr

Pose/Transformation Composition

More formally:

$$x_1 = \{R_1, t_1\}$$
$$x_2 = \{R_2, t_2\}$$

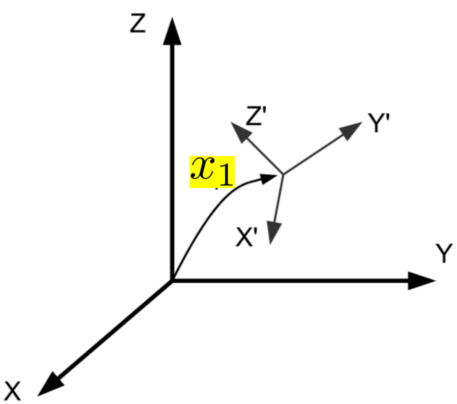
$$x \doteq x_1 \oplus x_2 = \{R_1 R_2, R_1 t_2 + t_1\}$$

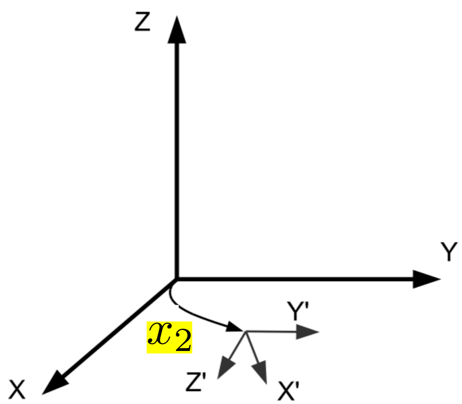
Explicitly:

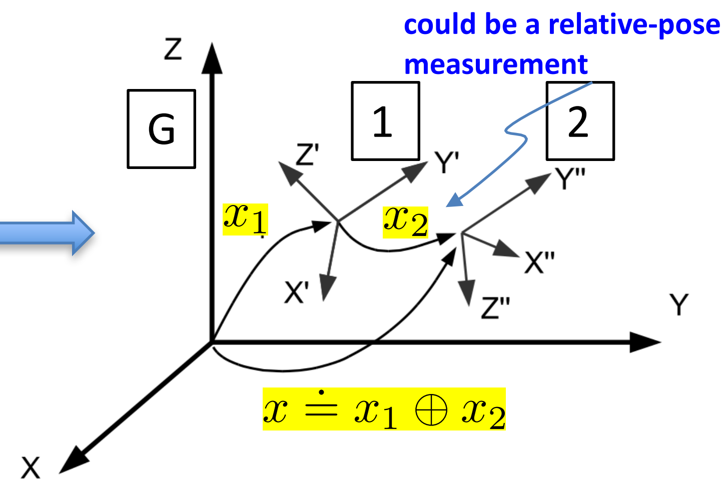
$${}^{G}_{1}T \doteq \begin{bmatrix} R_1^G & t_{G\to 1}^G \\ 0^T & 1 \end{bmatrix}$$

$${}^{1}_{2}T \doteq \begin{bmatrix} R_2^1 & t_{1\to 2}^1 \\ 0^T & 1 \end{bmatrix}$$

$${}^{G}_{2}T = {}^{G}_{1}T\,{}^{1}_{2}T = \begin{bmatrix} R_1^G R_2^1 & R_1^G t_{1\to 2}^1 + t_{G\to 1}^G \\ 0^T & 1 \end{bmatrix} = \begin{bmatrix} R_2^G & t_{G\to 2}^G \\ 0^T & 1 \end{bmatrix}$$





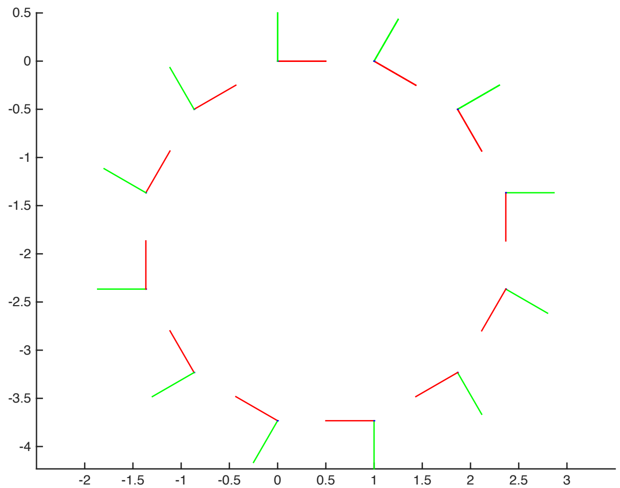


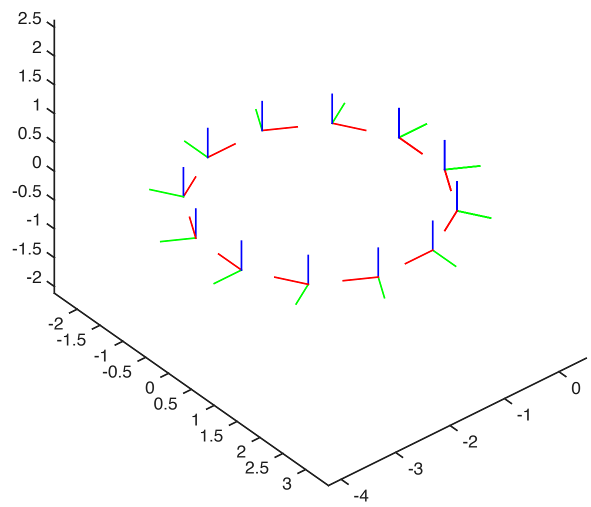
Diagrams adapted from Blanco14tr

Example

- Relative transformation (between two adjacent frames)
  - rotation around z axis by 30 deg
  - translation along x axis by 1 m

$$ {}^{i-1}_{i}T \doteq \begin{bmatrix} 0.866 & 0.5 & 0 & 1 \\ -0.5 & 0.866 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$





Platform/Camera Pose

- A minimal representation of a pose (of the robot, camera etc.) is described by 6 parameters:
  - 3D Cartesian coordinates $\mathbf{t} = \begin{pmatrix} x & y & z \end{pmatrix}^T$
  - Three Euler angles ϕ, θ, ψ
- The corresponding state is: $\mathbf{x} = \begin{pmatrix} x & y & z & \phi & \theta & \psi \end{pmatrix}^T \in \mathbb{R}^6$
- A 3D Euclidian transformation representation can be trivially obtained (how?)

$$\begin{bmatrix} R & \mathbf{t} \\ 0^T & 1 \end{bmatrix}$$

Navigation State

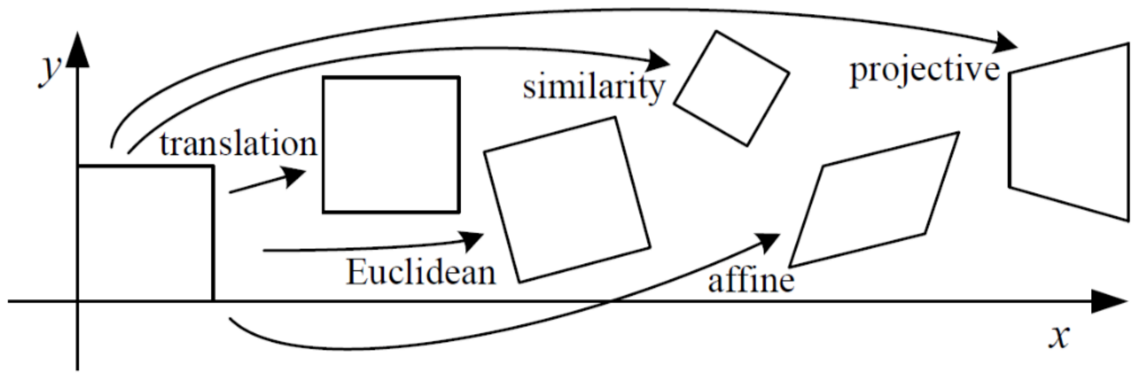
- In the navigation context, the state typically includes additional parameters:
  - Velocity (in particular, required when using inertial sensors)
  - Calibration parameters of different sensors (e.g. IMU, camera)

- For example – typical state with basic IMU calibration parameters:

$$\mathbf{x} = \begin{pmatrix} x & y & z & \phi & \theta & \psi & v_x & v_y & v_z & b_x^a & b_y^a & b_z^a & b_x^g & b_y^g & b_z^g \end{pmatrix}^T \in \mathbb{R}^{15}$$

6 DOF pose (position & orientation) **velocity** **accelerometer bias** **gyroscope bias**

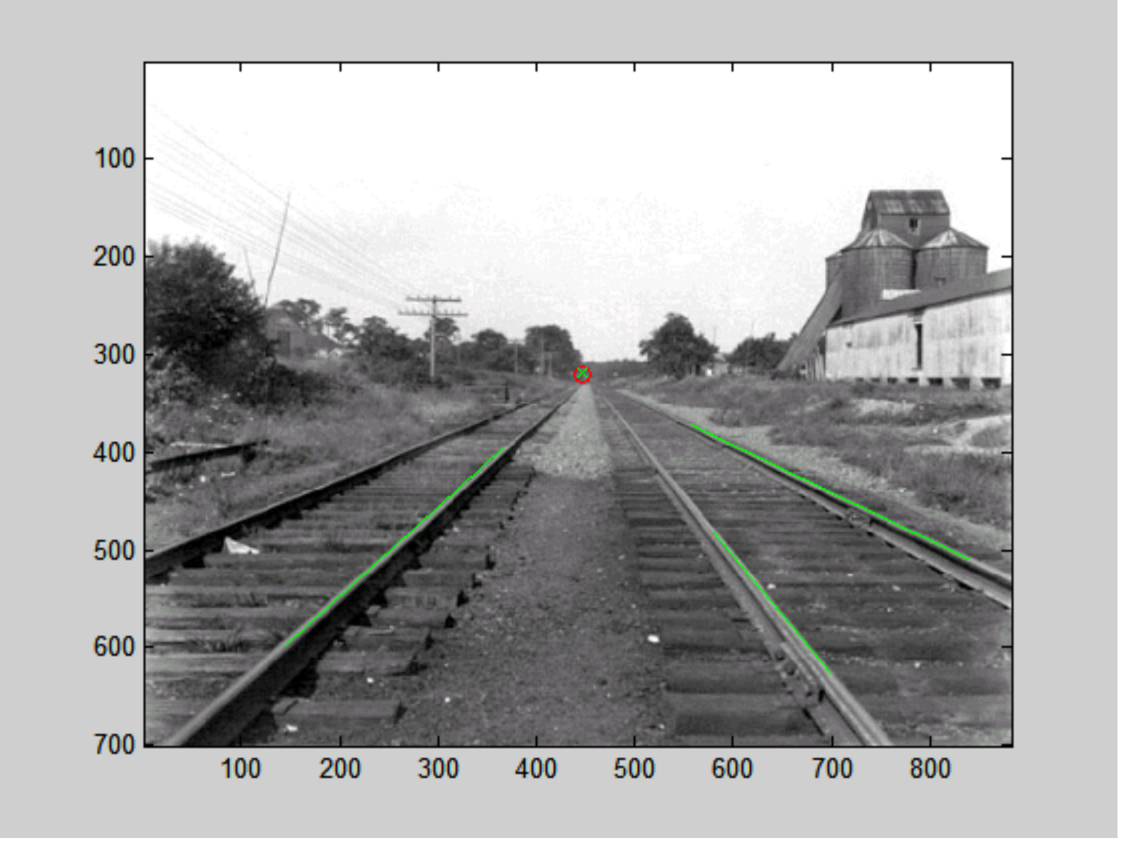
Additional Types of 3D Transformations



Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} \boldsymbol{I} \mid \boldsymbol{t} \end{bmatrix}_{3\times 4}$	3	orientation	
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{3\times 4}$	6	lengths	
similarity	$\begin{bmatrix} s\boldsymbol{R} \mid \boldsymbol{t} \end{bmatrix}_{3\times 4}$	7	angles	
affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{3\times 4}$	12	parallelism	
projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{4\times 4}$	15	straight lines	

Projective Transformation (a Glimpse)

- Preserves lines
- … More details in ~2 weeks



Recap - Lessons Learned

- 6 DOF Pose
- 3D Transformations
  - Euclidean (rotation and translation)
  - More general (e.g. projective transformation)
- Pose composition
- Navigation state

- Next: How to calculate where we are? Most basic - dead reckoning