Photogrammetry & Robotics Lab

Coordinate Transformations & Representations for Rotations

Cyrill Stachniss

The slides have been created by Cyrill Stachniss.

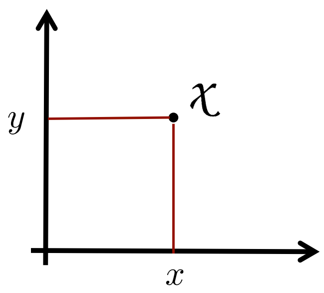
Coordinates? Transformation?

- Basically all geometric problems involve points in the 2D or 3D world
- How to represent points?
- Often, we need to **transform** points

Example

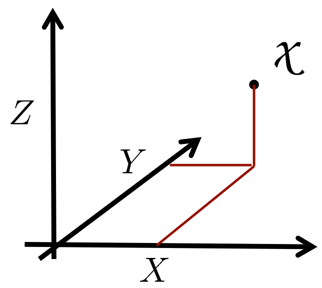
- The position of a robot can be represented by a point in space
- If the robot moves, we can model this by transforming that point

A Point in 2D



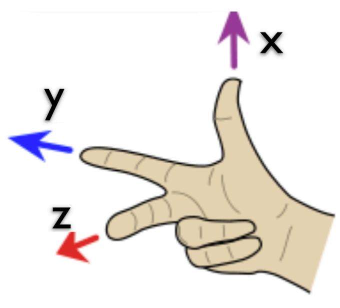
$$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

A Point in 3D

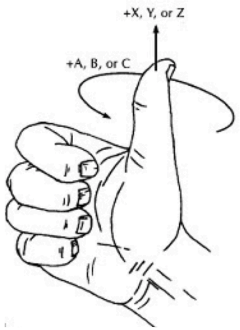


$$\boldsymbol{X} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Right-Handed C.S. in 3D



axes



positive rotation

[Image Courtesy: E. Olson]

Notation (in my lectures)

Point $\mathcal{X}$ (or $\mathcal{Y}$ or $\mathcal{P}$)

- in homogeneous coordinates $\mathbf{x}$
- in Euclidian coordinates $\boldsymbol{x}$

2D vs. **3D** space

- lowercase = 2D; capitalized = 3D

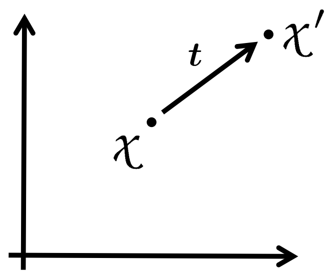
Plane $\mathcal{A}$

- in homogeneous coordinates $\mathbf{A}$

Matrix R

Quaternion $\mathbf{q}$

Translation



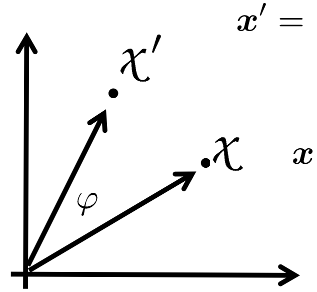
$$\boldsymbol{x}' = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

$$\boldsymbol{t} = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\mathcal{X}' = \mathcal{T}(\mathcal{X}) : \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Rotation

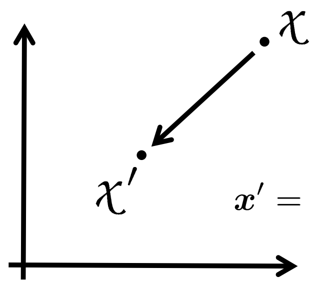


$$\boldsymbol{x}' = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\mathcal{X}' = \mathcal{R}(\mathcal{X}) : \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Scale Change



$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

$\boldsymbol{x}' = \begin{bmatrix} x' \\ y' \end{bmatrix}$

$$\mathcal{X}' = \mathcal{D}(\mathcal{X}) : \quad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \lambda x \\ \lambda y \end{bmatrix}$$

What is the Difference?

$$\mathcal{X}' = \mathcal{T}(\mathcal{X}) : \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

$$\mathcal{X}' = \mathcal{R}(\mathcal{X}) : \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\mathcal{X}' = \mathcal{D}(\mathcal{X}) : \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Different Operations

$$\mathcal{X}' = \mathcal{T}(\mathcal{X}) : \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

SUM

$$\mathcal{X}' = \mathcal{R}(\mathcal{X}) : \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

MATRIX MULT.

$$\mathcal{X}' = \mathcal{D}(\mathcal{X}) : \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

MATRIX MULT.

No Single Operation to Describe a Transformation in the Euclidian Space

$$\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \qquad \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

sum vs. matrix multiplication

This will lead us to using an **alternative representation** in the next lecture

In the Euclidian Space...

... transformations require to combine matrix multiplications **and** additions

$$\mathcal{X}' = \mathcal{T}(\mathcal{R}(\mathcal{D}(\mathcal{X})))$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \lambda x \\ \lambda y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Rotation and Translation are **NOT** commutative

- You cannot change the order of executing translations and rotations

$$\mathcal{T}(\mathcal{R}(\mathcal{X})) \neq \mathcal{R}(\mathcal{T}(\mathcal{X}))$$

- Here is why:

$$\mathcal{T}(\mathcal{R}(\mathcal{X})) = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

$$\mathcal{R}(\mathcal{T}(\mathcal{X})) = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} x + t_x \\ y + t_y \end{bmatrix}$$

Similarity Transform in 2D

- Scale change (1 DoF)
- Rotation (1 DoF)
- Translation (2 DoF)
- In sum: **4 DoF**

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} \lambda x \\ \lambda y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Similarity Transform in 3D

- Scale change (1 DoF)
- Rotation (3 DoF)
- Translation (3 DoF)
- In sum: **7 DoF**

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \lambda R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} t_X \\ t_Y \\ t_Z \end{bmatrix}$$

Representations of Rotation

Rotations Using Matrices

- A rotation is a special transformation

$$\mathcal{R} : \mathbb{R}^n \to \mathbb{R}^n \qquad \boldsymbol{x}' = R\boldsymbol{x}$$

- with $|R| = 1 \qquad R^{-1} = R^\top$

Properties

- Rotations do not change the scale
- They have a fix point (rotation center)

Rotation Matrices in 2D

- Rotations can be expressed through rotation matrices
- Example:

$$R(\varphi) = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix}$$

"rotation of the 2D plane by the angle φ"

Rotation Matrices in 3D

- We can do the same in 3D

$$R(\varphi) = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix}$$

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

Rotations Around X/Y/Z Axis

- We have 3 axes to rotate about...

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

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

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

There Are Multiple Possible Representations for a Rotation

- **Rotation matrix**
- **Euler angles**
- **Rotation axis and rotation angle**
- **Quaternion**
- Matrix exponential of a skew-symmetric matrix
- ...

Rotation Matrix

- Matrix of the form

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3] = \begin{bmatrix} \boldsymbol{r}_1^\top \\ \boldsymbol{r}_2^\top \\ \boldsymbol{r}_3^\top \end{bmatrix}$$

- with the constraints

$$|\boldsymbol{c}_1|^2 = 1 \qquad |\boldsymbol{c}_2|^2 = 1 \qquad |\boldsymbol{c}_3|^2 = 1$$
$$\boldsymbol{c}_1^\top \ \boldsymbol{c}_2 = 0 \qquad \boldsymbol{c}_2^\top \ \boldsymbol{c}_3 = 0 \qquad \boldsymbol{c}_3^\top \ \boldsymbol{c}_1 = 0$$

Degrees of Freedom

- A general 3 by 3 matrix has **9** degrees of freedom
- **6** independent constraints reduce the degrees of freedom (each: -1 DoF)
- **3 degrees of freedom remain**

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

$$|\boldsymbol{c}_1|^2 = 1 \qquad |\boldsymbol{c}_2|^2 = 1 \qquad |\boldsymbol{c}_3|^2 = 1$$
$$\boldsymbol{c}_1^\top \ \boldsymbol{c}_2 = 0 \qquad \boldsymbol{c}_2^\top \ \boldsymbol{c}_3 = 0 \qquad \boldsymbol{c}_3^\top \ \boldsymbol{c}_1 = 0$$

Properties

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3] = \begin{bmatrix} \boldsymbol{r}_1^\top \\ \boldsymbol{r}_2^\top \\ \boldsymbol{r}_3^\top \end{bmatrix}$$

- Column vectors $\boldsymbol{c}_i$ of R are the images of the unit vectors $\boldsymbol{e}_i$

$$\boldsymbol{c}_i = R\boldsymbol{e}_i = [\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3]\boldsymbol{e}_i$$

- Row vectors $\boldsymbol{r}_i$ of R are the vectors, which have been rotated into the unit vectors $\boldsymbol{e}_i$

$$\boldsymbol{r}_i^\top = \boldsymbol{e}_i^\top R = \boldsymbol{e}_i \begin{bmatrix} \boldsymbol{r}_1^\top \\ \boldsymbol{r}_2 \\ \boldsymbol{r}_3 \end{bmatrix}$$

Rotation Matrix Summary

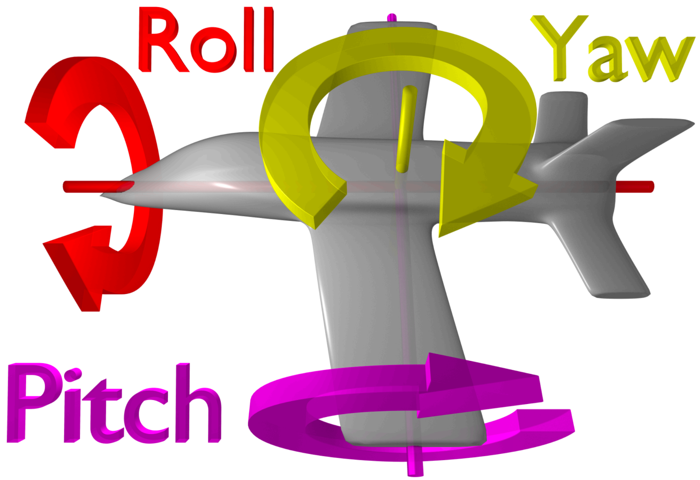
- 3D rotations can be expressed through a 3 by 3 matrix with $|R| = 1, R^{-1} = R^\top$
- Over-parameterized representation
- Commonly used as an exchange format for rotations
- Easy composition of rotations
- Suboptimal for state estimation problems (9 parameters + constraints)

Euler Angles

Euler Angles

- A rotation consists of three rotations around fixed axes (e.g., z-y-x axes)
- Useful for visualizing rotations
- Commonly used for describing planes, vehicles, robots, sensors, ...
- Minimal representation: 3 variable for 3 degrees of freedom

Example



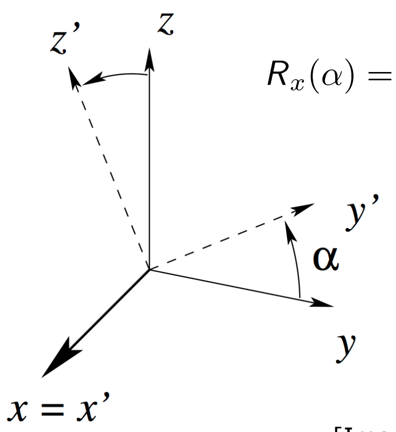
[Image Courtesy: Wikipedia Commons, User: ZeroOne]

Where Do the Axis Point to?

- **Warning: Different conventions!**
- Often: x-axis point forward
- Ground vehicles: y to the left
- Aerospace/marine: y to the right
- Satellites: ...
- Clockwise vs. counter clockwise rotations
- ...

Example: Rotate Around X Axis

Rotation: $(\alpha, 0, 0)$



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

$$\begin{aligned} x' &= x \\ y' &= y\cos\alpha - z\sin\alpha \\ z' &= y\sin\alpha + z\cos\alpha \end{aligned}$$

[Image Courtesy: W. Förstner]

Three Rotation Axes

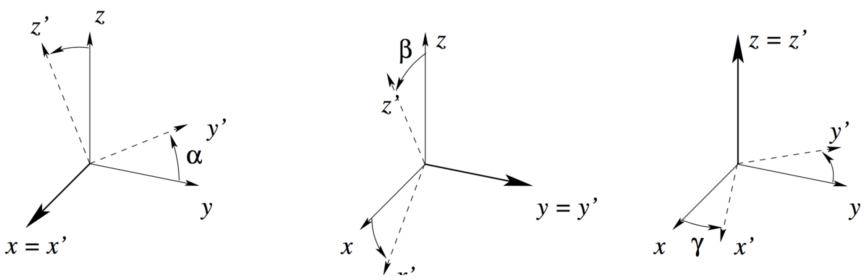
An arbitrary 3D rotation can be expressed by three rotations around three axes

$$R_1(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{bmatrix}$$ "x"

$$R_2(\varphi) = \begin{bmatrix} \cos\varphi & 0 & \sin\varphi \\ 0 & 1 & 0 \\ -\sin\varphi & 0 & \cos\varphi \end{bmatrix}$$ "y"

$$R_3(\varphi) = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ "z"

Three Rotation Axes



$R_x(\alpha)$ $\quad$ $R_y(\beta)$ $\quad$ $R_z(\gamma)$

[Image Courtesy: W. Förstner]

Rotation Composition

- 3 rotations around 3 axis must be combined
- In which order should that be done?

$$R_A(\alpha,\beta,\gamma) = R_3(\gamma)R_2(\beta)R_1(\alpha)$$
$$R_B(\alpha,\beta,\gamma) = R_1(\alpha)R_2(\beta)R_3(\gamma)$$
$$R_C(\alpha,\beta,\gamma) = R_1^\top(\alpha)R_2^\top(\beta)R_3^\top(\gamma) = R_A^\top$$
$$R_D(\alpha,\beta,\gamma) = R_3^\top(\gamma)R_2^\top(\beta)R_1^\top(\alpha) = R_B^\top$$

...

The Order Makes a Difference!

$$R_A(\alpha,\beta,\gamma) = R_3(\gamma)R_2(\beta)R_1(\alpha)$$

$$\begin{bmatrix} \cos\gamma\cos\beta & -\sin\gamma\cos\alpha + \cos\gamma\sin\beta\sin\alpha & \sin\gamma\sin\alpha + \cos\gamma\sin\beta\cos\alpha \\ \sin\gamma\cos\beta & \cos\gamma\cos\alpha + \sin\gamma\sin\beta\sin\alpha & -\cos\gamma\sin\alpha + \sin\gamma\sin\beta\cos\alpha \\ -\sin\beta & \cos\beta\sin\alpha & \cos\beta\cos\alpha \end{bmatrix}$$

$$R_B(\alpha,\beta,\gamma) = R_1(\alpha)R_2(\beta)R_3(\gamma)$$

$$\begin{bmatrix} \cos\gamma\cos\beta & -\sin\gamma\cos\beta & \sin\beta \\ \cos\gamma\sin\beta\sin\alpha + \sin\gamma\cos\alpha & -\sin\gamma\sin\beta\sin\alpha + \cos\gamma\cos\alpha & -\cos\beta\sin\alpha \\ -\cos\gamma\sin\beta\cos\alpha + \sin\gamma\sin\alpha & \sin\gamma\sin\beta\cos\alpha + \cos\gamma\sin\alpha & \cos\beta\cos\alpha \end{bmatrix}$$

Rotation Matrix to Euler Angles

- Given R, we can compute (α,β,γ)
- It is important to specify the convention for (α,β,γ)
- Assumptions:
  - Known axes and order of the rotations
  - No singularity in Euler angle representation

Rotation Matrix to Euler Angles

$$R_A(\alpha,\beta,\gamma) = R_3(\gamma)R_2(\beta)R_1(\alpha)$$

$$\begin{bmatrix} \cos\gamma\cos\beta & -\sin\gamma\cos\alpha+\cos\gamma\sin\beta\sin\alpha & \sin\gamma\sin\alpha+\cos\gamma\sin\beta\cos\alpha \\ \sin\gamma\cos\beta & \cos\gamma\cos\alpha+\sin\gamma\sin\beta\sin\alpha & -\cos\gamma\sin\alpha+\sin\gamma\sin\beta\cos\alpha \\ -\sin\beta & \cos\beta\sin\alpha & \cos\beta\cos\alpha \end{bmatrix}$$

if $\cos\beta \neq 0$

$$\alpha = \operatorname{atan2}(R_{32}, R_{33})$$
$$\beta = \operatorname{atan2}\left(-R_{31}, \sqrt{R_{32}^2+R_{33}^2}\right)$$
$$\gamma = \operatorname{atan2}(R_{21}, R_{11})$$

Singularity

$$R_A(\alpha,\beta,\gamma) = R_3(\gamma)R_2(\beta)R_1(\alpha)$$

$$\begin{bmatrix} \cos\gamma\cos\beta & -\sin\gamma\cos\alpha+\cos\gamma\sin\beta\sin\alpha & \sin\gamma\sin\alpha+\cos\gamma\sin\beta\cos\alpha \\ \sin\gamma\cos\beta & \cos\gamma\cos\alpha+\sin\gamma\sin\beta\sin\alpha & -\cos\gamma\sin\alpha+\sin\gamma\sin\beta\cos\alpha \\ -\sin\beta & \cos\beta\sin\alpha & \cos\beta\cos\alpha \end{bmatrix}$$

$$\cos\beta = 0 \Rightarrow \begin{bmatrix} 0 & -\sin\gamma\cos\alpha+\cos\gamma\sin\alpha & \sin\gamma\sin\alpha+\cos\gamma\cos\alpha \\ 0 & \sin\gamma\sin\alpha+\cos\gamma\cos\alpha & -\cos\gamma\sin\alpha+\sin\gamma\cos\alpha \\ -1 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & -\sin(\gamma-\alpha) & \cos(\gamma-\alpha) \\ 0 & \cos(\gamma-\alpha) & \sin(\gamma-\alpha) \\ -1 & 0 & 0 \end{bmatrix}$$

Singularity ("Gimbal Lock")

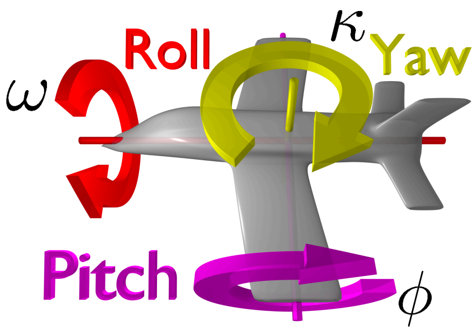
$$\begin{bmatrix} 0 & -\sin(\gamma-\alpha) & \cos(\gamma-\alpha) \\ 0 & \cos(\gamma-\alpha) & \sin(\gamma-\alpha) \\ -1 & 0 & 0 \end{bmatrix}$$

- $\beta = \pm 90^\circ$ but α, γ cannot be determined
- Singularity at $\beta = 90^\circ$
- Illustration: $\beta = 90^\circ$ makes the third rotation axis equal to the first one (except sign), thus only the difference $\gamma - \alpha$ can be inferred, but not α, γ

Discontinuity

- Rotation angles limited to $[0, 2\pi)$
- "Wrap around"
- Values at the borders of $[0, 2\pi)$ can lead to jumps in the parameters
- Problematic in state estimation

Often Used Conventions
3-2-1 = Yaw-Pitch-Roll OR
1-2-3 = Roll-Pitch-Yaw
(Omega, Phi, Kappa)



[Image Courtesy: Wikipedia Commons, User: ZeroOne]

Euler Angles Summary

- Three rotations around fixed axes
- Different variants (3-2-1, 1-2-3, ...)
- Useful for visualizing rotations
- Minimal representation
- Singularities
- Discontinuities
- No direct composition of rotations
- Suboptimal for general state estimation problems

Axis-Angle Representation

Result from Euler's Theorem

- Every composition of 3D rotations can be expressed a single rotation around a single rotation axis
- From Euler angles to a single rotation

- We can simple use a vector to encode the rotation axis and one scalar for the rotation angle
- 4 parameters (3+1)

Axis-Angle vs. Normalized Axis-Angle Representation

- Angle plus rotation axis has length one

  $\theta, \boldsymbol{r} = [r_1, r_2, r_3]^\top \quad \text{with} \quad \|\boldsymbol{r}\| = 1$
- 4 parameters plus 1 constraint

- Alternative: encode angle onto length

  $\boldsymbol{r} = [r_1, r_2, r_3]^\top \quad \text{with} \quad \|\boldsymbol{r}\| = \theta$
- Minimal representation with 3 params.

Axis-Angle to Rotation Matrix

- Rodrigues Formula:

$$R_{r,\theta} = I_3 + \sin\theta\ S_r + (1 - \cos\theta)\ S_r^2$$

- with $S_r \doteq \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix}$
- Alternatively

$\boldsymbol{\theta} = \theta \boldsymbol{r}$

$$R_{\boldsymbol{\theta}} = I_3 + \frac{\sin|\boldsymbol{\theta}|}{|\boldsymbol{\theta}|}\ S_\theta + \frac{1 - \cos|\boldsymbol{\theta}|}{|\boldsymbol{\theta}|^2}\ S_\theta^2$$

Rotation Matrix to Axis-Angle

- Angle: $\theta = \operatorname{atan2}(|\boldsymbol{a}|, \operatorname{tr}\ R - 1)$
- with:

$$\boldsymbol{a} = -\begin{pmatrix} r_{23} - r_{32} \\ r_{31} - r_{13} \\ r_{12} - r_{21} \end{pmatrix} = 2\sin\theta\ \boldsymbol{r}$$

$$\operatorname{tr}(R) = r_{11} + r_{22} + r_{33} = 1 + 2\cos\theta$$

Rotation Matrix to Axis-Angle

For the axis, we have three cases:

- 1: $\theta = 0 \leftrightarrow R = I$: singularity
- 2: $\theta = \pi$: sign of the axis is irrelevant and we have

  $\boldsymbol{r}\boldsymbol{r}^\top = \frac{R + I}{2}$ => r given by any column of R

- 3: otherwise $\boldsymbol{r} = \frac{\boldsymbol{a}}{|\boldsymbol{a}|}$

Axis Angle Summary

- Singularity for $\theta = 0$
- Rotations limited to $(-\pi, \pi]$
- Human readable
- Minimal representation if normalized
- No direct composition of rotations (e.g., requires to build rotation matrix)

Quaternions

Quaternions

- Alternative way for modeling rotations
- Described by W. Hamilton in 1843 (also: Rodrigues, 1840; Gauss 1819)
- Quaternions form an algebra and can be seen as a complex number with a 3-dimensional complex component
- Partially human-readable form but manipulations not fully intuitive
- Offer outstanding properties...

William Hamilton in 1843, Brougham Bridge in Dublin

[Image Courtesy: Wikipedia; User: Cone83]

Singularities and Discontinuities

Quaternions are the only 4-parameter representation for rotations that

- is **unique**, except of the sign
- has **no singularities**
- has **no discontinuities**

"Quaternions are almost minimal"

Definition (seen as a Complex Number)

- Quaternion $\mathbf{q}$ is a 4D vector

$$\mathbf{q} = \begin{bmatrix} q \\ \boldsymbol{q} \end{bmatrix} \quad \text{with} \quad \boldsymbol{q} = [q_1, q_2, q_3]^\top$$

- can be interpreted as a 1D real number q and a 3D complex part $\boldsymbol{q}$

$$\mathbf{q} = q + q_1\, i + q_2\, j + q_3\, k$$

- with 3 complex units i, j, k
- for which holds $i^2 = j^2 = k^2 = ijk = -1$

Definition (seen as a Tuple)

- Quaternion $\mathbf{q}$ is a 4D vector

$$\mathbf{q} = \begin{bmatrix} q \\ \boldsymbol{q} \end{bmatrix} \quad \text{with} \quad \boldsymbol{q} = [q_1, q_2, q_3]^\top$$

- that can also be viewed as a tuple

$$\mathbf{q} = (q, \boldsymbol{q})$$

- and that follows rules of a certain algebra (for addition, multiplication, inversion, ...)

Addition

Adding two quaternions is (intuitively) defined as the element-wise sum

$$\mathbf{p} = \mathbf{q} + \mathbf{r}$$

$$\begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} q_0 + r_0 \\ q_1 + r_1 \\ q_2 + r_2 \\ q_3 + r_3 \end{bmatrix} \qquad (p, \boldsymbol{p}) = (q + r, \boldsymbol{q} + \boldsymbol{r})$$

Multiplication

Multiplication is defined differently from what one might suspect

$$\mathbf{p} = \mathbf{qr}$$

$$(p, \boldsymbol{p}) = (qr - \boldsymbol{q} \cdot \boldsymbol{r}, r\boldsymbol{q} + q\boldsymbol{r} + \boldsymbol{q} \times \boldsymbol{r})$$

$$\begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} q_0 r_0 - q_1 r_1 - q_2 r_2 - q_3 r_3 \\ q_1 r_0 + q_0 r_1 - q_3 r_2 + q_2 r_3 \\ q_2 r_0 + q_3 r_1 + q_0 r_2 - q_1 r_3 \\ q_3 r_0 - q_2 r_1 + q_1 r_2 + q_0 r_3 \end{bmatrix}$$

Multiplication and Unit Quat.

- Because of the cross-product

$$(p, \boldsymbol{p}) = (qr - \boldsymbol{q} \cdot \boldsymbol{r}, r\boldsymbol{q} + q\boldsymbol{r} + \boldsymbol{q} \times \boldsymbol{r})$$

- the multiplication is **not** commutative!

- **Unit quaternion** is a quaternion with

$$\mathbf{q} \quad \text{with} \quad |\mathbf{q}| = 1$$

Inverse

The inverse $\mathbf{q}^{-1}$ of a quaternion is defined as

$$\mathbf{q}^* = \begin{bmatrix} q \\ -\boldsymbol{q} \end{bmatrix}$$

"conjugate"

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{|\mathbf{q}|^2}$$

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

Quaternions and Rotations

- Quaternions can be used to model rotations

$$\mathbf{q} = \begin{bmatrix} q \\ \boldsymbol{q} \end{bmatrix} = \begin{bmatrix} \cos(\theta/2) \\ \sin(\theta/2)\boldsymbol{r} \end{bmatrix}$$

- with the normalized rotation axis

$$\boldsymbol{r} = [r_1, r_2, r_3]^\top \quad \text{with} \quad \|\boldsymbol{r}\| = 1$$

Example: Quaternion Rotation

- A rotation of θ around $\boldsymbol{r}$ with

$$\theta = 30^\circ \qquad \boldsymbol{r} = \begin{bmatrix} 0.6 \\ 0.8 \\ 0.0 \end{bmatrix}$$

- Equation: $\mathbf{q} = \begin{bmatrix} q \\ \boldsymbol{q} \end{bmatrix} = \begin{bmatrix} \cos(\theta/2) \\ \sin(\theta/2)\boldsymbol{r} \end{bmatrix}$

- Result (in different notations):

$$\mathbf{q} = \begin{bmatrix} \cos(15^\circ) \\ \sin(15^\circ) \begin{bmatrix} 0.6 \\ 0.8 \\ 0.0 \end{bmatrix} \end{bmatrix}$$

$$\mathbf{q} = (\cos(15^\circ), \sin(15^\circ)(0.6, 0.8, 0.0))$$

$$\mathbf{q} = \cos(15^\circ) + \sin(15^\circ)(0.6\, i + 0.8\, j + 0.0\, k)$$

Quaternion & Axis-Angle

- The pure presentation of a quaternion is similar to axis-angle
- Quaternion

$$\mathbf{q} = \begin{bmatrix} q \\ \boldsymbol{q} \end{bmatrix} = \begin{bmatrix} \cos(\theta/2) \\ \sin(\theta/2)\boldsymbol{r} \end{bmatrix} \quad \text{with} \quad \|\boldsymbol{r}\| = 1$$

- Axis-Angle

$$\theta, \boldsymbol{r} = [r_1, r_2, r_3]^\top \quad \text{with} \quad \|\boldsymbol{r}\| = 1$$

Executing a Rotation

- Through left and right multiplication, we can perform a rotation of the vector part $\boldsymbol{p}$ of $\mathbf{p}$ ($\mathbf{p} = (0, \boldsymbol{p})$) by $\mathbf{q}$:

$$\mathbf{p}' = \mathbf{q}\mathbf{p}\mathbf{q}^{-1}$$

Executing a Rotation

- Through left and right multiplication, we can perform a rotation of the vector part $\boldsymbol{p}$ of $\mathbf{p}$ ($\mathbf{p} = (0, \boldsymbol{p})$) by $\mathbf{q}$:

$$\mathbf{p}' = \mathbf{q}\mathbf{p}\mathbf{q}^{-1}$$

Example:

- 3D point to rotate: $\boldsymbol{p}$
- Define point as: $\mathbf{p} = (0, \boldsymbol{p})$
- Define rotation: $\mathbf{q} = (\cos(\theta/2), \sin(\theta/2)\boldsymbol{r})$
- Rotate $\mathbf{p}' = \mathbf{q}\mathbf{p}\mathbf{q}^{-1}$

Executing a Rotation

- Through left and right multiplication, we can perform a rotation of the vector part $\boldsymbol{p}$ of $\mathbf{p}$ ($\mathbf{p} = (0, \boldsymbol{p})$) by $\mathbf{q}$:

$$\mathbf{p}' = \mathbf{q}\mathbf{p}\mathbf{q}^{-1}$$

- Rotations can be easily composed by quaternion multiplication

$$\mathbf{q}\mathbf{p} = (qp - \boldsymbol{q} \cdot \boldsymbol{p}, p\boldsymbol{q} + q\boldsymbol{p} + \boldsymbol{q} \times \boldsymbol{p})$$

Composition

Rotations can be easily composed by quaternion multiplication

$$\mathbf{p}' = \mathbf{q}'\mathbf{p}\mathbf{q}'^{-1} \qquad \mathbf{p}'' = \mathbf{q}''\mathbf{p}'\mathbf{q}''^{-1}$$

$$\mathbf{p}'' = \mathbf{q}''\,(\mathbf{q}'\,\mathbf{p}\,\mathbf{q}'^{-1})\,\mathbf{q}''^{-1}$$

Composition

Rotations can be easily composed by quaternion multiplication

$$\mathbf{p}' = \mathbf{q}'\mathbf{p}\mathbf{q}'^{-1} \qquad \mathbf{p}'' = \mathbf{q}''\mathbf{p}'\mathbf{q}''^{-1}$$

$$\begin{aligned}\mathbf{p}'' &= \mathbf{q}''\,(\mathbf{q}'\,\mathbf{p}\,\mathbf{q}'^{-1})\,\mathbf{q}''^{-1} \\ &= (\mathbf{q}''\mathbf{q}')\,\mathbf{p}\,(\mathbf{q}'^{-1}\mathbf{q}''^{-1}) \\ &= \underbrace{(\mathbf{q}''\mathbf{q}')}_{\mathbf{q}}\;\mathbf{p}\;\underbrace{(\mathbf{q}''\mathbf{q}')^{-1}}_{\mathbf{q}^{-1}} \\ &= \mathbf{q}\mathbf{p}\mathbf{q}^{-1}\end{aligned}$$

Composition

- Thus, the rotation $\mathbf{q}$ is obtained by multiplying the rotation quat. $\mathbf{q}'', \mathbf{q}'$

$$\mathbf{q} \quad = \quad \mathbf{q}''\mathbf{q}'$$

- and executing the rotation by

$$\mathbf{p}'' \quad = \quad \mathbf{q}\mathbf{p}\mathbf{q}^{-1}$$

- This corresponds to the rotation matrices $R_{\mathbf{q}} \quad = \quad R_{\mathbf{q}''}R_{\mathbf{q}'}$

Quaternion and Rotation Matrix

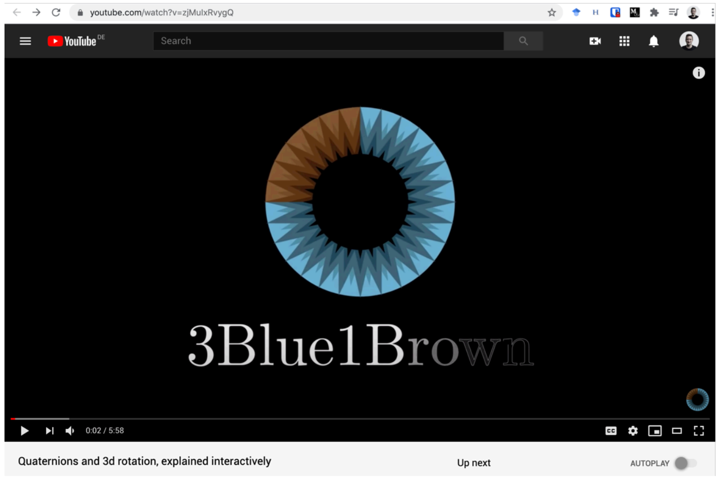
- A unit quaternion can be transformed into a rotation matrix by

$$R_{\mathbf{q}} = \begin{bmatrix} 1-2\left(q_2^2+q_3^2\right) & 2\left(q_1q_2-q_0q_3\right) & 2\left(q_1q_3+q_0q_2\right) \\ 2\left(q_2q_1+q_0q_3\right) & 1-2\left(q_1^2+q_3^2\right) & 2\left(q_2q_3-q_0q_1\right) \\ 2\left(q_3q_1-q_0q_2\right) & 2\left(q_3q_2+q_0q_1\right) & 1-2\left(q_1^2+q_2^2\right) \end{bmatrix}$$

- Conversion in the other direction

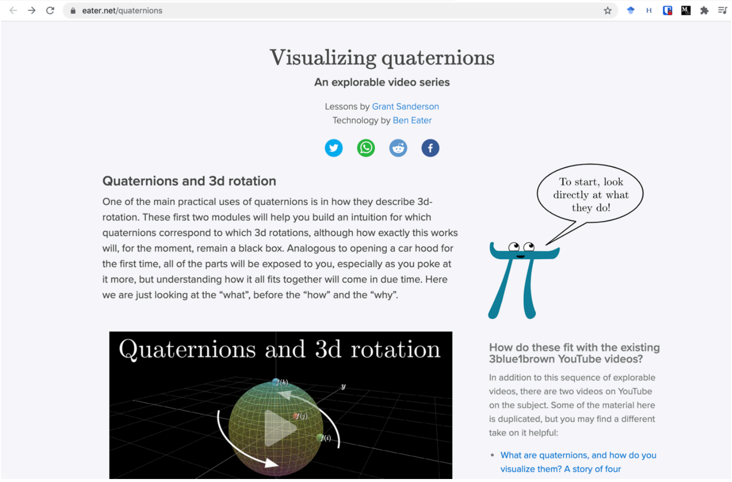
$$\begin{aligned} q_0 &= \frac{1}{2}\sqrt{1+\mathrm{tr}R} \\ q_1 &= (R_{32}-R_{23})/(4q_0) \\ q_2 &= (R_{13}-R_{31})/(4q_0) \\ q_3 &= (R_{21}-R_{12})/(4q_0) \end{aligned}$$

Outstanding Video Explanation



https://www.youtube.com/watch?v=zjMuIxRvygQ

Outstanding Interactive Tutorial



https://eater.net/quaternions

Quaternion Summary

- 4-parameter presentation
- Unique, except of the sign
- No singularities, no discontinuities
- Easy composition of rotations
- Allow for angular interpolation (SLERP)
- Often used for state estimation
- Attractive way for handling rotations
- Partially human readable, often seen as confusing (at first)

Summary (1)

- We can express rotations using different representations
- Representation differ w.r.t.:
  - Readability by humans
  - Singularities
  - Discontinuities
  - Minimal vs. over-parameterized
  - Uniqueness
  - Simplicity of rotation composition

3D Rotation Parameterizations

	#params	Ease of Use[1]	Unique?	Readable?	Singularities?
Rotation Matrix	9	★★	yes	no	no
Euler Angles	3	★★★	no	yes	yes
Rotations Axis and Angle	4(3)	★★	(no)	(yes)	yes
Quaternion	4(3)	★★	yes	no	no

([1] highly subjective rating)

Summary (2)

- There is no "best" representation
- Rotation matrices often serve as an exchange format
- Euler angler can be visualized well
- State estimation often relies on quaternions
- Angle-axis representations are human readable (link to quaternions)

Slide Information

- The slides have been created by Cyrill Stachniss as part of the photogrammetry and robotics courses.
- **I tried to acknowledge all people from whom I used images or videos. In case I made a mistake or missed someone, please let me know.**
- The photogrammetry material heavily relies on the very well written lecture notes by Wolfgang Förstner and the Photogrammetric Computer Vision book by Förstner & Wrobel.
- Parts of the robotics material stems from the great Probabilistic Robotics book by Thrun, Burgard and Fox.
- If you are a university lecturer, feel free to use the course material. If you adapt the course material, please make sure that you keep the acknowledgements to others and please acknowledge me as well. To satisfy my own curiosity, please send me email notice if you use my slides.

Cyrill Stachniss, cyrill.stachniss@igg.uni-bonn.de