Quaternions

- "extension" of complex numbers
- by Hamilton
  - algebra of quaternions hence often denoted with H
  - abstract symbols i,j,k with: $i^2 = j^2 = k^2 = ijk = -1$
  - quaternion z: $z = a + bi + cj + dk$
  - *a* also denoted as scalar- and *(b,c,d)* as vector part
  - notation: $z = (a, \mathbf{v})^T$, $\mathbf{v} = (b,c,d)$

Quaternions

multiplication of quaternions $q_i = (s_i, \mathbf{v_i})$

$$q_1 q_2 = \left(s_1 s_2 - \mathbf{v}_1 \cdot \mathbf{v}_2, s_1 \mathbf{v}_2 + s_2 \mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2\right)$$

dot-product $$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

cross-product $$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \times \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{pmatrix}.$$

Quaternions

$$q \;=\; a + bi + cj + dk$$

conjugate $\overline{q}$ of q : $\quad \overline{q} = a\text{-}bi\text{-}cj\text{-}dk$

norm $|q| = \sqrt{q\,\overline{q}} = \sqrt{\overline{q}\,q} = \sqrt{a^2 + b^2 + c^2 + d^2}$

unit quaternion $(\,|q'| = 1\,) : \mathrm{q'} = \mathrm{q} \,/\, |q|$

Quaternions for Rotations

- ***unit*** quaternions can represent rotations
- kind of axis-angle representation
  - ***v*** ~ axis, *a* ~ angle
  - though: *a* not something like angle in radians or so due to normalization
  - hence not straightforward to interpret

Quaternions for Rotations

note: point p =$(x,y,z)^T$ (or vector **v**)

- represented as quaternion
- with scalar part = Zero
- i.e., p = $(0,x,y,z)^T$

like for homogeneous coordinates:

- representation (3D point/vector or 4D quat.)
- is assumed to be clear from the context

Quaternions for Rotations

quaternion rotation: $p' = q\ p\ \bar{q}$

rotate by angle θ around unit axis $\mathbf{v}$:

use $q = (\cos(\theta / 2),\ \mathbf{v}\sin(\theta / 2))$

Conversion of quaternion to rotation matrix

z = a + bi + cj + dk (with |z| = 1)

$$\begin{pmatrix} a^2+b^2-c^2-d^2 & 2bc-2ad & 2bd+2ac \\ 2bc+2ad & a^2-b^2+c^2-d^2 & 2cd-2ab \\ 2bd-2ac & 2cd+2ab & a^2-b^2-c^2+d^2 \end{pmatrix}.$$

(Dis)Advantages Quaternions

quaternions

- no gimbal lock
- 4 parameters
- vector rotation:
  24 mul + 18 add (42)
- chaining rotations:
  16 mul + 12 add (28)
- numerical stability:
  rounding errors – normalize
  => still a rotation (close to intended one)
- easy to interpolate "smooth" rotations between 2 orientations

rotation matrix

- gimbal lock (with Euler angles)
- 9 parameters
- vector rotation:
  9 mul + 6 add (15)
- chaining rotations:
  27 mul + 18 add (45)
- numerical stability: harder to ensure orthogonality of rotation matrix, i.e., that it is a rotation at all

more detailed perf. discussion (incl. also axis-angle):
David Eberly,
Rotation Representations and Performance Issues
http://geometrictools.com/Documentation/RotationIssues.pdf

Rodrigues' (Rotation) Formula

intuitive axis angle formula

- rotate **v**
- by angle theta
- around a normalized axis **k**

$$\mathbf{v}' = \mathbf{v}\cos\theta + (\mathbf{k}\times\mathbf{v})\sin\theta + \mathbf{k}(\mathbf{k}\cdot\mathbf{v})(1-\cos\theta)$$

2nd version

- rotate in a plane spanned by **a** and **b**
- angle from **a** to **b**

$$\mathbf{k} = \frac{\mathbf{a}\times\mathbf{b}}{|\mathbf{a}\times\mathbf{b}|} = \frac{\mathbf{a}\times\mathbf{b}}{|\mathbf{a}||\mathbf{b}|\sin\alpha}$$

Recap: Homogeneous Coordinates

- 3D coordinates
  - translation as addition
- homogeneous coordinates
  - $(a,b,c,s)^T$
    - s = scale
    - mapped to 3D as $(a/s, b/s, c/s)^T$
    - kinematics: s = 1
  - translation and rotation as matrix product

Homogeneous Matrix

translation followed by rotation

$$H = \begin{pmatrix} & & \\ & R & t \\ & & \\ & p & s \end{pmatrix} = \begin{pmatrix} R_{1,1} & R_{1,2} & R_{1,3} & t_1 \\ R_{2,1} & R_{2,2} & R_{2,3} & t_2 \\ R_{3,1} & R_{3,2} & R_{3,3} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Dual Interpretation

the homogeneous matrix is

- an operator
  - transforming points (respectively, rigid bodies)
- a coordinate system
  - R-part ~ basis (frame axes)
  - t-part ~ origin

Semantics of rotation

several cases:

1) frame (as coordinate system) remains fixed but the **scene/body rotates**

- denoted here as fixed axes (FiAx)
- point p simply changed to p’ by rotation matrix R

$$\mathbf{p}' = \mathbf{R}\,\mathbf{p}$$

Semantics of rotation

several cases:

2) physical scene remains fixed but the **observation frame moves**

- denoted here as rotated-axes (RoAx)
- always involves two “canonical” frames
  - start frame F_S before and
  - end frame F_E after the rotation

Notation from Craig-textbook

John J. Craig. **Introduction to Robotics**. Prentice Hall
1st ed. 1986, 2nd ed. 1989, 3rd ed. 2004

- variables
  - uppercase = matrices
  - lowercase = scalars
  - or if bold = vectors (Craig: uppercase, too)
- hat (aka carat) denotes unit vector

Notation from Craig-textbook

- left superscript
  - denotes frame in which vector/point or frame is resolved, i.e., "written in"
  - ${}^{A}\boldsymbol{p}$: point ***p*** as "seen" in frame ***A***
  - ${}^{A}_{B}\boldsymbol{F}$, ${}^{A}_{B}\boldsymbol{T}$: frame/transform ***B*** as "seen" in frame ***A***
- left subscript
  - denotes new, resulting frame
  - ${}^{S}_{E}R$: rotation *R* from S to E
  - (superscript = start S, subscript = end E)
- right super-/subscript
  - superscript: for transpose, inverse, etc.
  - subscript: for description/indexing, etc.

Note: semantics of rotation

tricky part: case 2) (observer moves)

(without translation) coordinates of the same physical object are related in both frames by

$$ {}^{\mathrm{S}}\mathbf{p} = {}^{\mathrm{S}}_{\mathrm{E}}\mathbf{R} \; {}^{\mathrm{E}}\mathbf{p} $$

Note:

- here “before” coordinates on left hand side (LHS)
- case 1) of FiAx: “after” coordinates on LHS

Note: semantics of rotation

[still case 2)]

$$ {}^{\mathrm{S}}\mathbf{p} = {}^{\mathrm{S}}_{\mathrm{E}}\mathbf{R}\ {}^{\mathrm{E}}\mathbf{p} $$

Note:

- here "before" coordinates on left hand side (LHS)
- case 1) of FiAx: "after" coordinates on LHS

advantage: rotation matrix to rotate a scene/body within FiAx is the same rotation matrix to rotate the frame while keeping the scene fixed

Recap: Inverses of Rotations

- inverted rotation matrices
- are simply transposed matrices

$$R_x(\alpha)^{-1} = R_x(\alpha)^T \quad R_y(\beta)^{-1} = R_y(\beta)^T \quad R_z(\gamma)^{-1} = R_z(\gamma)^T$$

Inverse of the Homogeneous Matrix

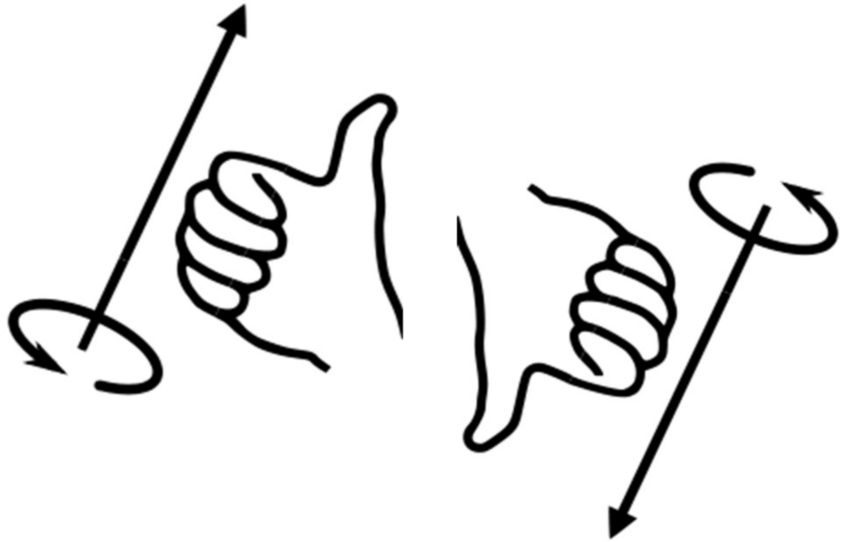
$$H^{-1} = \begin{pmatrix} & R & & t \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} & R^T & & -R^T t \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

intuition:

- do inverse rotations in reverse order
- translation vector needs to be transformed to match previous coordinate system

Inverse Rotation of a Quaternion

- **-q** is same rotation as **q**
  (think in terms of axis-angle)
- the conjugate is the inverse rotation



$$q = a + bi + cj + dk$$

conjugate $\overline{q}$ of q : $\quad \overline{q} = a\text{-}bi\text{-}cj\text{-}dk$

Inverse Rotation of a Quaternion

Notes on notations:

- conjugate hence occasionally denoted with q^{-1} (also often: q^*)
- hat/carat occasionally used on quaternions
  - to stress that it represents a rotation
  - as it denotes a unit vector

Chaining Frames, resp. Quaternions

given spatial transforms *T* as homogeneous matrices

$$ {}^{A}_{C}T_3 = {}^{A}_{C}T_2 \cdot {}^{A}_{B}T_1 $$

$$ = {}^{A}_{B}T_1 \cdot {}^{B}_{C}T_2 $$

rotating from *S* over *E* to *F*

$$ \mathcal{F}_\mathrm{S} \rightarrow \mathcal{F}_\mathrm{E} \rightarrow \mathcal{F}_\mathrm{F} $$

$$ {}^{\mathrm{S}}_{\mathrm{F}}\check{\mathbf{q}} = {}^{\mathrm{S}}_{\mathrm{E}}\check{\mathbf{q}} \diamond {}^{\mathrm{E}}_{\mathrm{F}}\check{\mathbf{q}} $$

(where ◊ denotes the quaternion product)