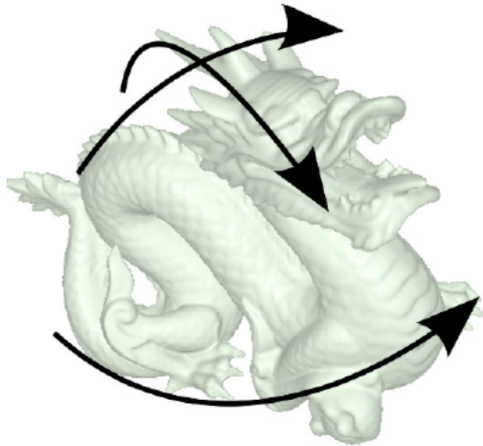
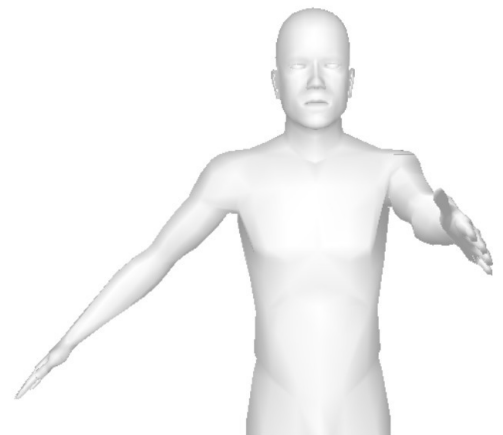




Rotation - Quaternion



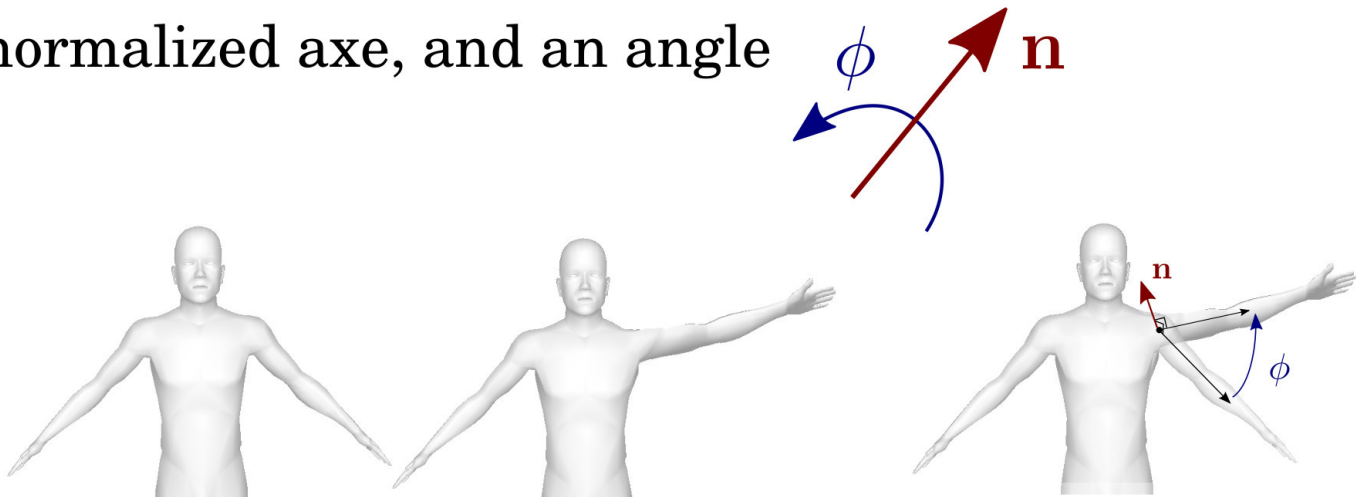
Rotation

A 3D rotation can be described by:

An orthonormal matrix

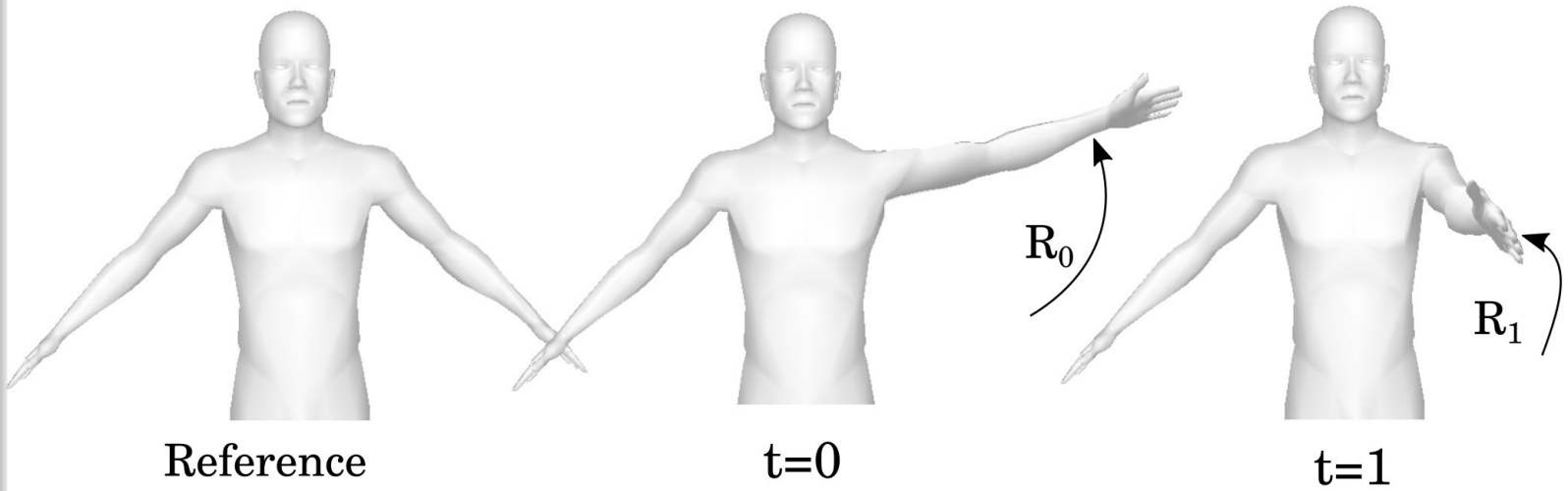
$$R(\mathbf{n}, \phi) = \begin{pmatrix} \cos(\phi) + n_x^2(1 - \cos(\phi)) & n_x n_y(1 - \cos(\phi)) - n_z \sin(\phi) & n_y \sin(\phi) + n_x n_z(1 - \cos(\phi)) \\ n_z \sin(\phi) + n_x n_y(1 - \cos(\phi)) & \cos(\phi) + n_y^2(1 - \cos(\phi)) & -n_x \sin(\phi) + n_y n_z(1 - \cos(\phi)) \\ -n_y \sin(\phi) + n_x n_z(1 - \cos(\phi)) & n_x \sin(\phi) + n_y n_z(1 - \cos(\phi)) & \cos(\phi) + n_z^2(1 - \cos(\phi)) \end{pmatrix}$$

A normalized axe, and an angle



Rotation interpolation

Given 2 rotations, how to interpolate them ?

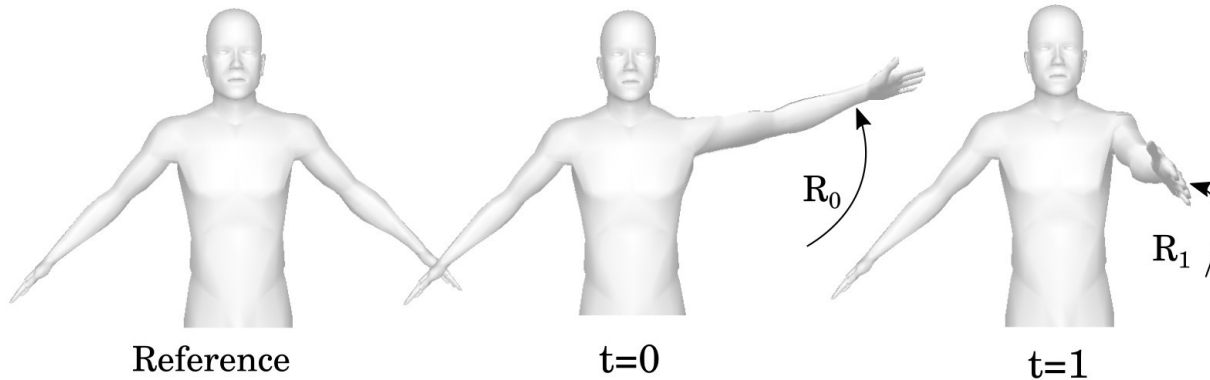


What is $R_{1/2}$ at $t=0.5$?

Problem: $0.5 R_0 + 0.5 R_1$ is not a rotation!

Rotation space is not a vectorial space!

Rotation interpolation



More general question:

Given R_0 and R_1 ,

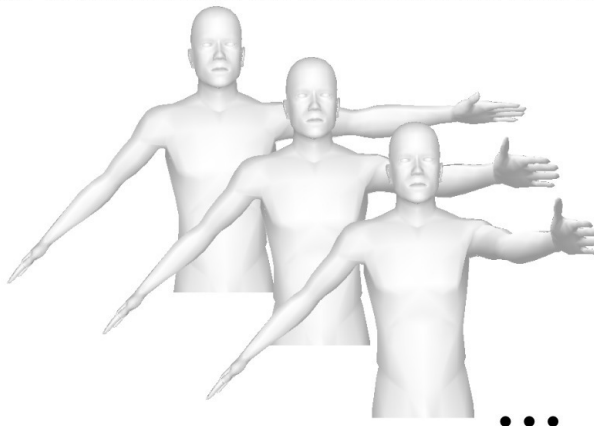
How to compute a continuous animation between the keyframe 0 and keyframe 1 ?

$R(t=0.1)$

$R(t=0.2)$

$R(t=0.3)$

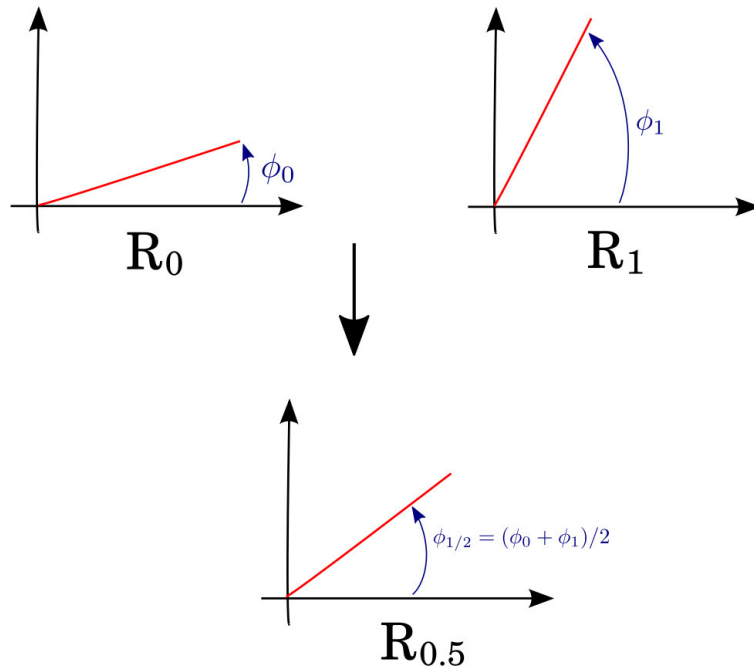
...



Rotation interpolation

In 2D, easy solution

=> Interpolate the angle



But in 3D, there is also two normalized angles

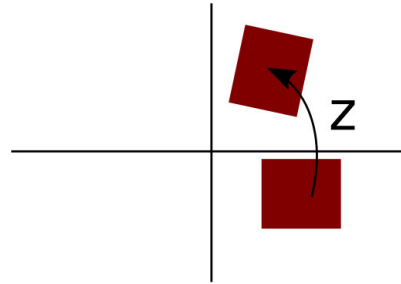
Complex number

Remember that complex numbers can model rotation in 2D

$$z = e^{i\theta}$$

$$z = x + y i$$

$$|z| = 1$$



Given 2 rotation

the middle one is

$$z_0 = e^{i\theta_0}$$

$$z_1 = e^{i\theta_1}$$

$$z_{1/2} = e^{i(\theta_0 + \theta_1)/2}$$

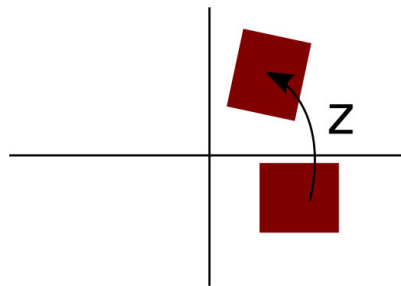
Complex number

Remember that complex numbers can model rotation in 2D

$$z = e^{i\theta}$$

$$z = x + y i$$

$$|z| = 1$$



More generally:

Given 2 rotation the interpolated one is

$$z_0 = e^{i\theta_0}$$

$$z_1 = e^{i\theta_1}$$

$$z_\alpha = e^{i((1-\alpha)\theta_0 + \alpha\theta_1)}$$

$$\alpha \in [0, 1]$$

=> Nice formalism of the complex number for rotation

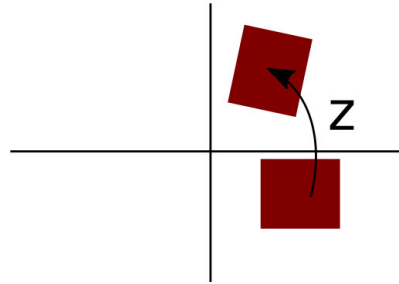
Generalisation to quaternion

Complex numbers can model rotation in 2D

$$z = e^{\mathbf{i}\theta}$$

$$z = x + y \mathbf{i}$$

$$|z| = 1$$



Quaternion can model rotation in 3D

$$q = x + y \mathbf{i} + z \mathbf{j} + w \mathbf{k} \in \mathbb{R}^4$$

$$|q| = 1$$



$$i^2 = -1 \quad ij = k$$

$$j^2 = -1 \quad jk = i$$

$$k^2 = -1 \quad ik = -j \quad ijk = -1$$

Quaternion

Unit quaternion are rotations in 3D

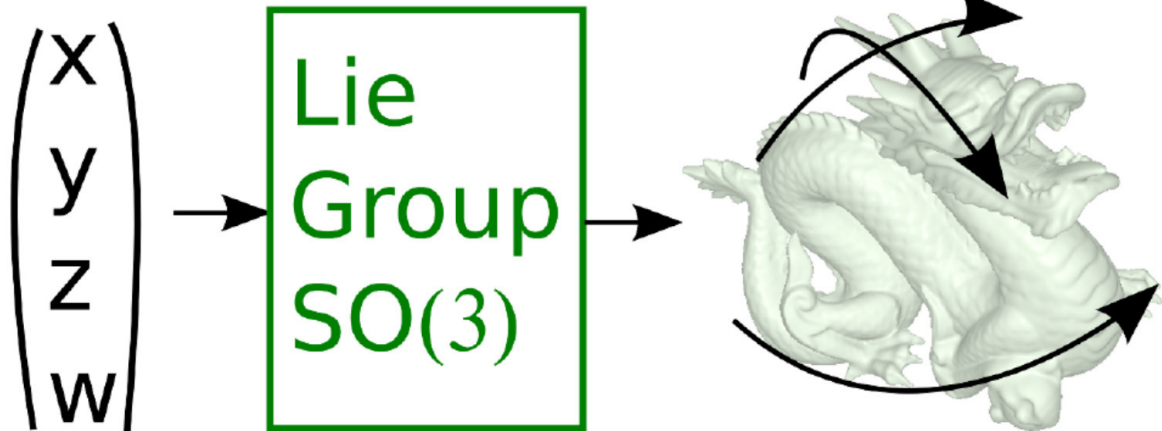
$$q = x + y \mathbf{i} + z \mathbf{j} + w \mathbf{k}$$

$$|q| = 1$$

+ Short representation of rotation $q_1 q_2 \Leftrightarrow R_1 R_2$
(non comutative product)

Algebre

rotations



Equivalence rotation/quaternion

Quaternion $q = (q_0, q_1, q_2, q_3) \Rightarrow$ Matrix R

$$R = \begin{pmatrix} 1 - 2(q_1^2 + q_2^2) & 2(q_0q_1 - q_3q_2) & 2(q_0q_2 + q_3q_1) \\ 2(q_0q_1 + q_3q_2) & 1 - 2(q_0^2 + q_2^2) & 2(q_1q_2 - q_3q_0) \\ 2(q_0q_2 - q_3q_1) & 2(q_1q_2 + q_3q_0) & 1 - 2(q_0^2 + q_1^2) \end{pmatrix}$$

Efficient representation of multiplication: $q = (\mathbf{v}, w)$

$$q_0 \ q_1 = (w_0 \mathbf{v}_1 + w_1 \mathbf{v}_0 - \mathbf{v}_0 \times \mathbf{v}_1, w_0 w_1 - \mathbf{v}_0 \cdot \mathbf{v}_1)$$

Applying a rotation to a 3D point

Given a 3D point $p = (x, y, z)$

And a unit quaternion q

The rotation of p by q is given by p'

where $p' = (z'_0, z'_1, z'_2)$

with $z' = q z \bar{q}$

$\bar{q} = (-q_0, -q_1, -q_2, q_3)$ is the conjugated quaternion.

Quaternion interpolation

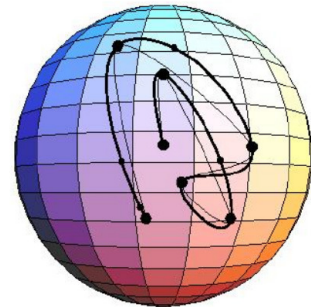
Quaternion/rotation can be interpolated using the SLERP.

Spherical Linear intERPoltation

$$\text{SLERP}(q_0, q_1, \alpha) = \frac{\sin(\alpha(1 - \theta))}{\sin(\theta)} q_0 + \frac{\sin(\alpha\theta)}{\sin(\theta)} q_1$$

$$\cos(\theta) = q_0 \cdot q_1$$

$$\alpha \in [0, 1]$$



Other approach

Matrix interpolation also exists.

Generalization of linear interpolation in Lie Group

$$R = (R_2 R_1^{-1})^\alpha R_1$$

But: requires matrix exponential

- Computational cost
- Numerical instabilities

=> Quaternion SLERP is more efficient

Application to camera trajectory

