CS 4732: Computer Animation

Interpolation

Robert W. Lindeman
Associate Professor
Interactive Media & Game Development
Department of Computer Science
Worcester Polytechnic Institute
gogo@wpi.edu

Quaternion Rotation

- A quaternion is a scalar and a vector
  - q = [s, v]
  - [s, v] = [-s, -v]
- To rotate a vector *v* using a quaternion
  - Represent the vector as [0, *v*]
  - Represent the rotation as *q*
  - Using quaternion multiplication

$$v' = Rot_q(v) = qvq^{-1}$$

  - Note: The scalar value for *v′* is always zero

Composing Quaternion Rotations

- Rotating a vector *v* by quaternion *p* followed by quaternion *q* is like a rotation using *qp*.

$$
\begin{aligned}
Rot_q(Rot_p(v)) &= Rot_q(pvp^{-1}) \\
&= qpvp^{-1}q^{-1} \\
&= (qp)v(qp)^{-1} \\
&= Rot_{qp}(v)
\end{aligned}
$$

Composing Quaternion Rotations

- TO rotate a vector *v* by quaternion *q* followed by its inverse quaternion q^{-1}

$$Rot_{q^{-1}}(Rot_q(v)) = Rot_{q^{-1}}(qvq^{-1})$$

$$= q^{-1}qvq^{-1}q$$

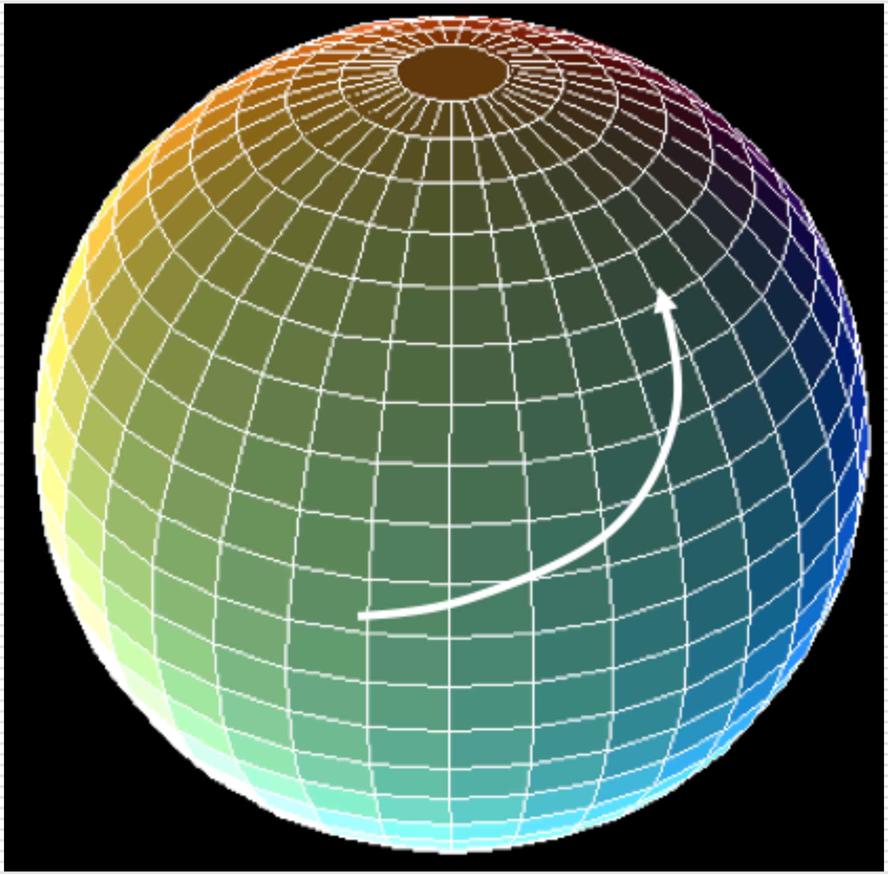
$$= v$$

Quaternion Interpolation

- A quaternion is a point on a 4D unit sphere
  - Unit quaternion: $q=(s,x,y,z)$, $||q|| = 1$
    - Forms a subspace: 4D sphere
- Interpolating quaternions means moving between two points on the 4D unit sphere
  - A unit quaternion at each step – another point on the 4D unit sphere.
  - Move with constant angular velocity along the greatest circle between the two points on the 4D sphere

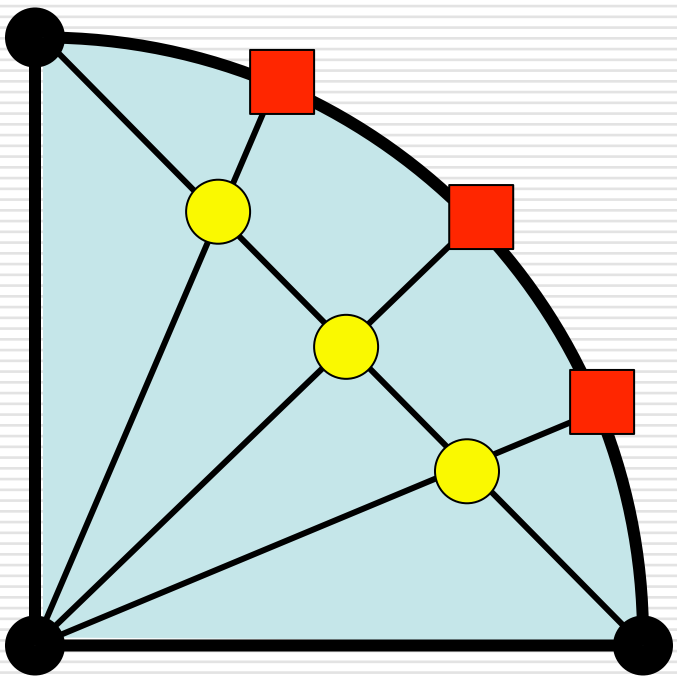
Quaternion Interpolation (cont.)

- Move with constant angular velocity along the greatest circle between the two points on the 4D unit sphere



Linear Interpolation

- Linear interpolation generates unequal spacing of points after projecting onto a circle



Spherical Linear Interpolation (slerp)

- Want equal increment along an arc connecting two points on a spherical surface

$$slerp(q_1, q_2, u) = q_1 \frac{\sin((1-u)\theta)}{\sin\theta} + q_2 \frac{\sin(u\theta)}{\sin\theta}$$

- Where
  - u goes from 0 to 1
  - $\theta = \cos^{-1}(q_1 \bullet q_2)$
- NOTE: Normalize to get a unit quaternion

Spherical Linear Interpolation (slerp)

- Let $q = \alpha q_1 + \beta q_2$
- We can solve, given:

$$\|q\| = 1$$

$$q_1 \bullet q_2 = \theta$$

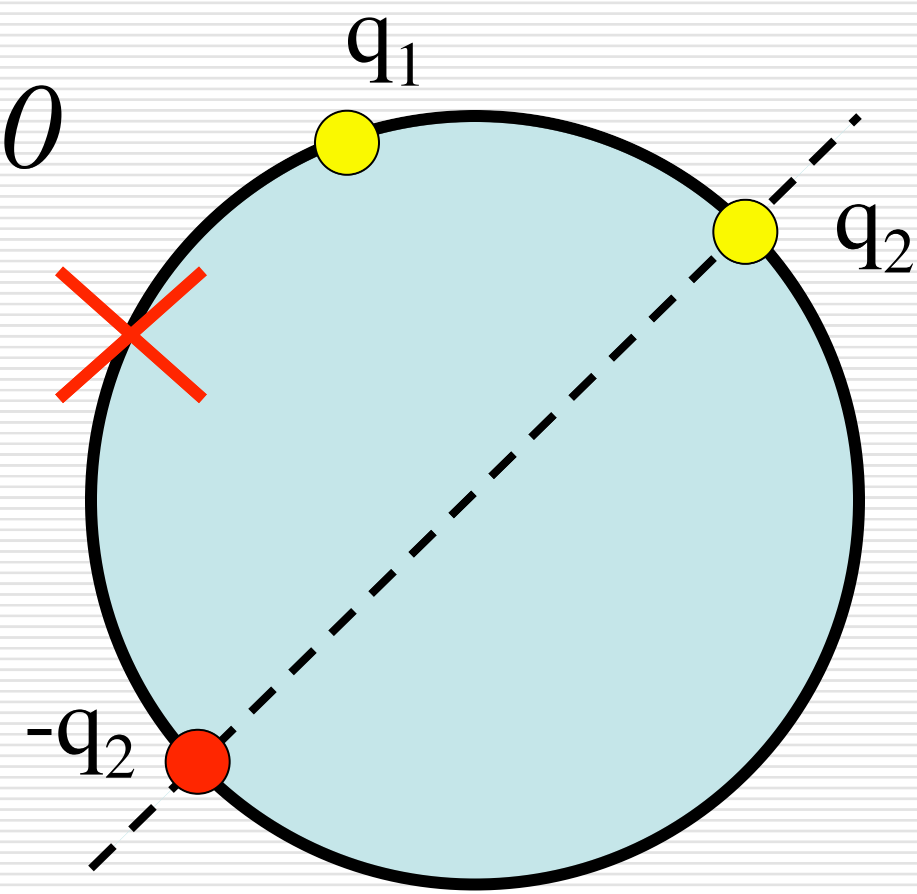
$$q_1 \bullet q = u\theta$$

to give:

$$slerp(q_1, q_2, u) = q_1 \frac{\sin((1-u)\theta)}{\sin\theta} + q_2 \frac{\sin(u\theta)}{\sin\theta}$$

Spherical Linear Interpolation (slerp)

$$q_1 \bullet q_2 > 0$$



Spherical Linear Interpolation (slerp)

- Recall that q and $-q$ represent the same rotation
- What is the difference between:

  ***slerp*(q_1, q_2, *u*)** and ***slerp*(q_1, $-q_2$, *u*)**

  - One of these will travel less than 90 degrees, while the other will travel more than 90 degrees across the sphere
  - This corresponds to rotating the ‘short way’ or the ‘long way’
- Usually, we want to take the short way, so we negate one of them if their dot product is < 0

Quaternion Summary

- Advantages
  - Good, smooth interpolation (slerp)
  - No gimbal lock
  - Can be compsed much more efficiently
    - Eight multiplies and four divides
- Disadvantages
  - Impossible to visualize
  - Unintuitive
- Good for internal representations of rotation.