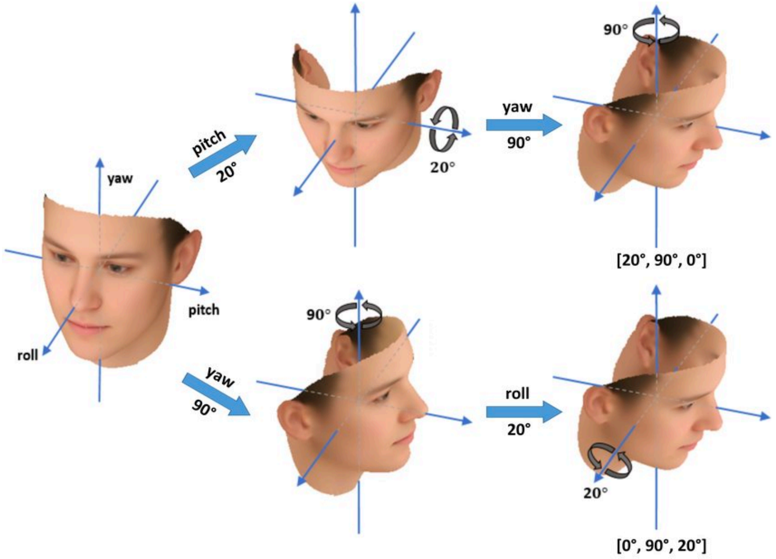
Modeling 3D Orientation/Rotation



Avinash Sharma
Center for Visual IT, KCIS, IIIT Hyderabad

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

CVIT

Representation for 3D Rotation

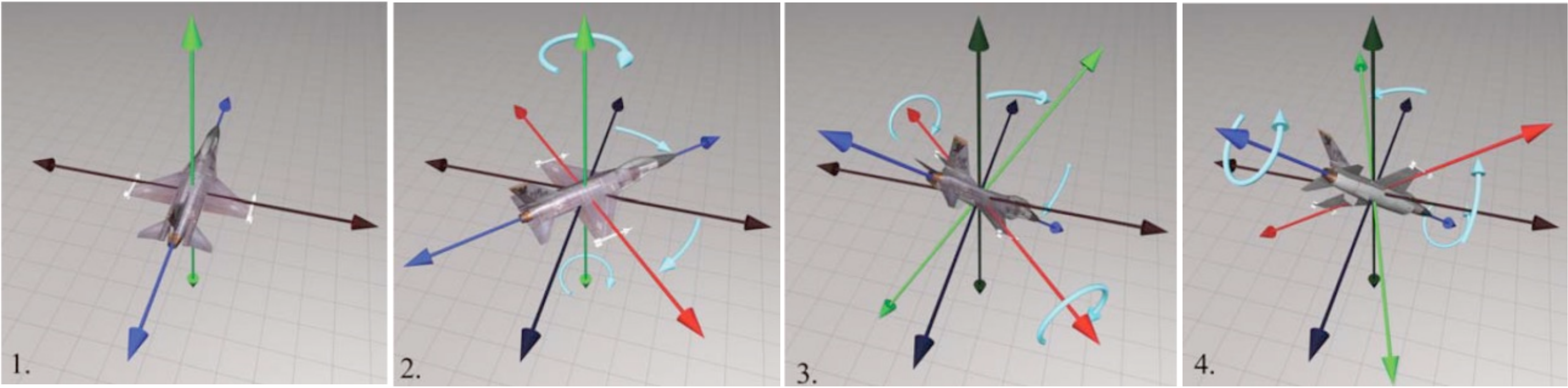
- Is there a simple means of representing a 3D Rotation? (analogous to Cartesian coordinates?)
- There are several popular options though:
  - Rotation matrices (3x3)
  - Euler angles
  - Rotation vectors (axis/angle)
  - Quaternions

Representation for Rotation

- 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.
- This means that we can represent an orientation with 3 numbers
- A sequence of rotations around principle axes is called an ***Euler Angle Sequence.***

Euler Angle Sequence

- Which axes are being rotated around in this sequence?



Yaw (R_Y) Pitch (R_X) Roll (R_Z)

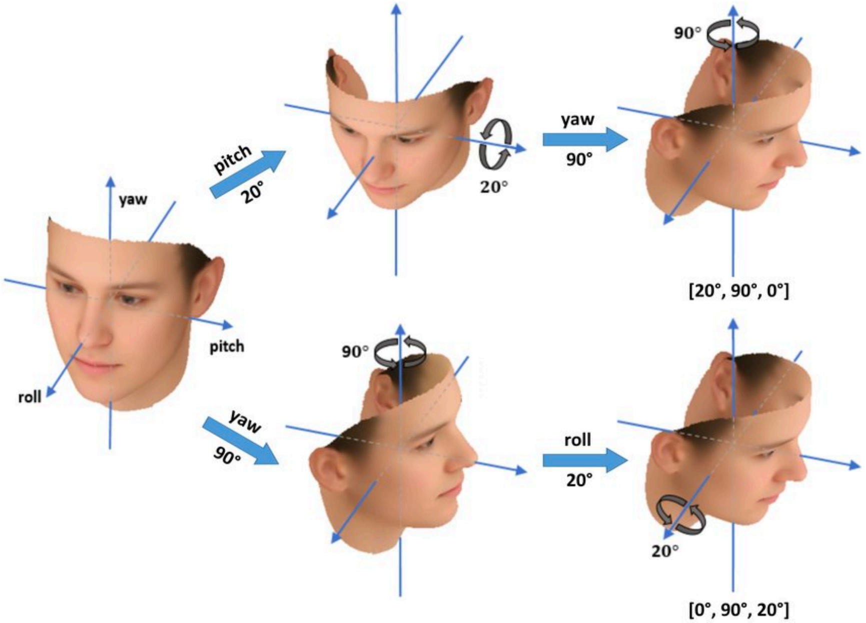
Euler Angles to Matrix Conversion

$$\mathbf{R}_z \cdot \mathbf{R}_y \cdot \mathbf{R}_x = \begin{bmatrix} c_z & -s_z & 0 \\ s_z & c_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_y & 0 & s_y \\ 0 & 1 & 0 \\ -s_y & 0 & c_y \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_x & -s_x \\ 0 & s_x & c_x \end{bmatrix}$$

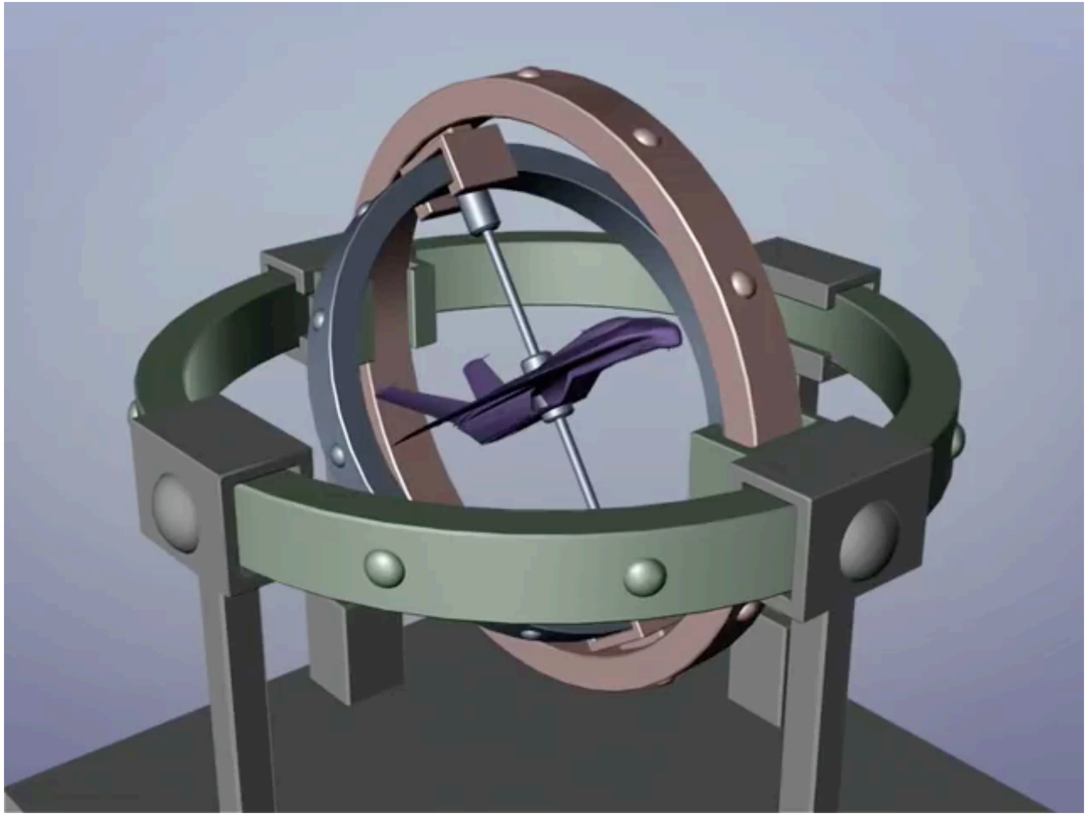
$$= \begin{bmatrix} c_y c_z & s_x s_y c_z - c_x s_z & c_x s_y c_z + s_x s_z \\ c_y s_z & s_x s_y s_z + c_x c_z & c_x s_y s_z - s_x c_z \\ -s_y & s_x c_y & c_x c_y \end{bmatrix}$$

Gimbal Lock

- Rz(roll) Rx(pitch) Ry(yaw)
- Gimbal Lock occurs when any two axes collapsed onto each other
- To avoid this, typically a fixed pitch angle is allowed for VR

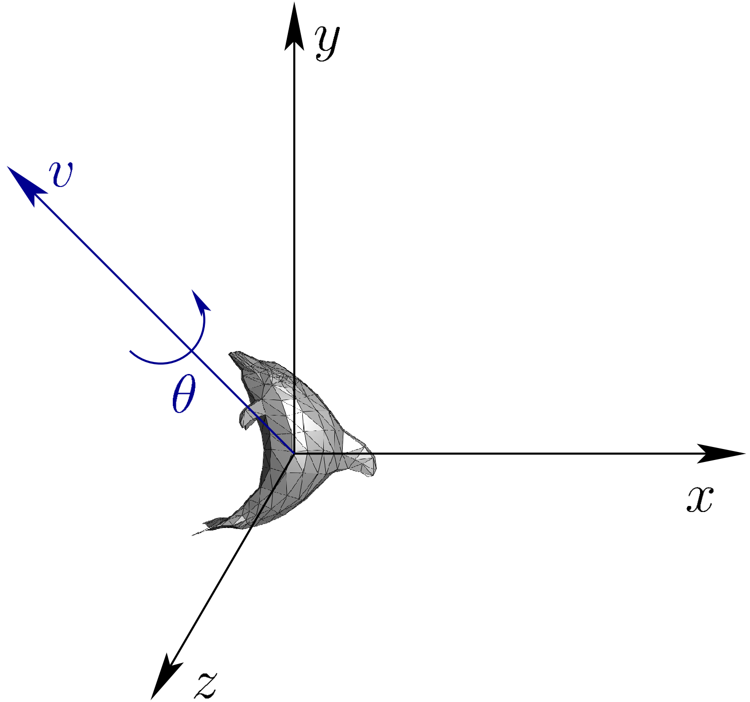


Gimbal Lock



Axis/Angle Representation

- Euler's Theorem also shows that any two orientations can be related by a single rotation about some axis (not necessarily a principle axis)
- This means that we can represent an arbitrary orientation as a rotation about some unit axis by some angle (4 numbers) (Axis/Angle form)



Quaternion

- Alternative to Euler Angles
- Developed by Sir William Rowan Hamilton [1843]
- Quaternions are 4-D complex numbers with one real axis and three imaginary axes: ***i***,***j***,***k***
- $$\mathbf{q} = q_0 + iq_1 + jq_2 + kq_3$$

Quaternion

- Given an angle and axis, easy to convert to and from quaternion
  - Euler angle conversion to and from arbitrary axis and angle difficult
- Quaternions allow stable and constant interpolation of orientations
  - Cannot be done easily with Euler angles

Quaternion

- We will use only unit length quaternions

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

- These correspond to the set of 4D vectors
- They form the ‘surface’ of a 4D hypersphere of radius 1

Quaternion as Rotation

- A quaternion can represent a rotation by angle θ around a unit vector **a**:

$$\mathbf{q} = \begin{bmatrix} \cos\frac{\theta}{2} & a_x \sin\frac{\theta}{2} & a_y \sin\frac{\theta}{2} & a_z \sin\frac{\theta}{2} \end{bmatrix}$$

or

$$\mathbf{q} = \left\langle \cos\frac{\theta}{2}, \mathbf{a} \sin\frac{\theta}{2} \right\rangle$$

- If **a** is unit length, then **q** will be also

Quaternion	Axis-Angle	Description
$(1,0,0,0)$	$(\text{undefined}, 0)$	Identity rotation
$(0,1,0,0)$	$((1,0,0),\pi)$	Pitch by π
$(0,0,1,0)$	$((0,1,0),\pi)$	Yaw by π
$(0,0,0,1)$	$((0,0,1),\pi)$	Roll by π
$(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}},0,0)$	$((1,0,0),\pi/2)$	Pitch by $\pi/2$
$(\frac{1}{\sqrt{2}},0,\frac{1}{\sqrt{2}},0)$	$((0,1,0),\pi/2)$	Yaw by $\pi/2$
$(\frac{1}{\sqrt{2}},0,0,\frac{1}{\sqrt{2}})$	$((0,0,1),\pi/2)$	Roll by $\pi/2$

Quaternion as Rotation

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

$$= \sqrt{\cos^2\frac{\theta}{2} + a_x^2\sin^2\frac{\theta}{2} + a_y^2\sin^2\frac{\theta}{2} + a_z^2\sin^2\frac{\theta}{2}}$$

$$= \sqrt{\cos^2\frac{\theta}{2} + \sin^2\frac{\theta}{2}\left(a_x^2 + a_y^2 + a_z^2\right)}$$

$$= \sqrt{\cos^2\frac{\theta}{2} + \sin^2\frac{\theta}{2}|\mathbf{a}|^2} = \sqrt{\cos^2\frac{\theta}{2} + \sin^2\frac{\theta}{2}}$$

$$= \sqrt{1} = 1$$

Rotation using Quaternion

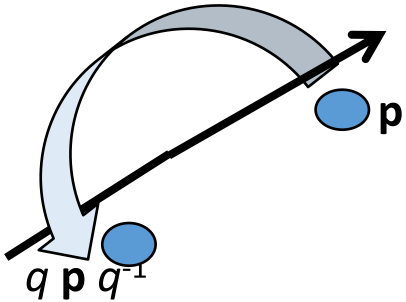
- Let $q = \cos(\theta/2) + \sin(\theta/2)\,\mathbf{u}$ be a unit quaternion: $|q| = |\mathbf{u}| = 1$
- Let point $\mathbf{p} = (x,y,z) = x\,\mathbf{i} + y\,\mathbf{j} + z\,\mathbf{k}$
- Then the product $q\,\mathbf{p}\,q^{-1}$ rotates the point $\mathbf{p}$ about axis $\mathbf{u}$ by angle θ
- Inverse of a unit quaternion is its ***conjugate***

  …just negate the imaginary part

$$
\begin{aligned}
q^{-1} &= (\cos(\theta/2) + \sin(\theta/2)\,\mathbf{u})^{-1} \\
&= \cos(-\theta/2) + \sin(-\theta/2)\,\mathbf{u} \\
&= \cos(\theta/2) - \sin(\theta/2)\,\mathbf{u}
\end{aligned}
$$

- Composition of rotations $q_{12} = q_1\,q_2 \neq q_2\,q_1$

$$q = \underset{\theta}{\cos}\frac{\theta}{2} + \underset{\mathbf{u}}{\mathbf{u}}\sin\frac{\theta}{2}$$



Quaternion to Rotation Matrix

- To convert a quaternion to a rotation matrix:

$$\begin{bmatrix} 1-2q_2^2-2q_3^2 & 2q_1q_2+2q_0q_3 & 2q_1q_3-2q_0q_2 \\ 2q_1q_2-2q_0q_3 & 1-2q_1^2-2q_3^2 & 2q_2q_3+2q_0q_1 \\ 2q_1q_3+2q_0q_2 & 2q_2q_3-2q_0q_1 & 1-2q_1^2-2q_2^2 \end{bmatrix}$$

Rotation Matrix to Quaternion

- Matrix to quaternion is not hard
  - it involves a few 'if' statements,
  - a square root,
  - three divisions,
  - and some other stuff
- tr(M) is the trace
  - sum of the diagonal elements

$$q_0 = \frac{1}{2}\sqrt{tr(\mathbf{M})}$$

$$q_1 = \frac{m_{21} - m_{12}}{4q_0}$$

$$q_2 = \frac{m_{02} - m_{20}}{4q_0}$$

$$q_3 = \frac{m_{10} - m_{01}}{4q_0}$$

Quaternion Multiplication

- We can perform multiplication on quaternions
  - we expand them into their complex number form

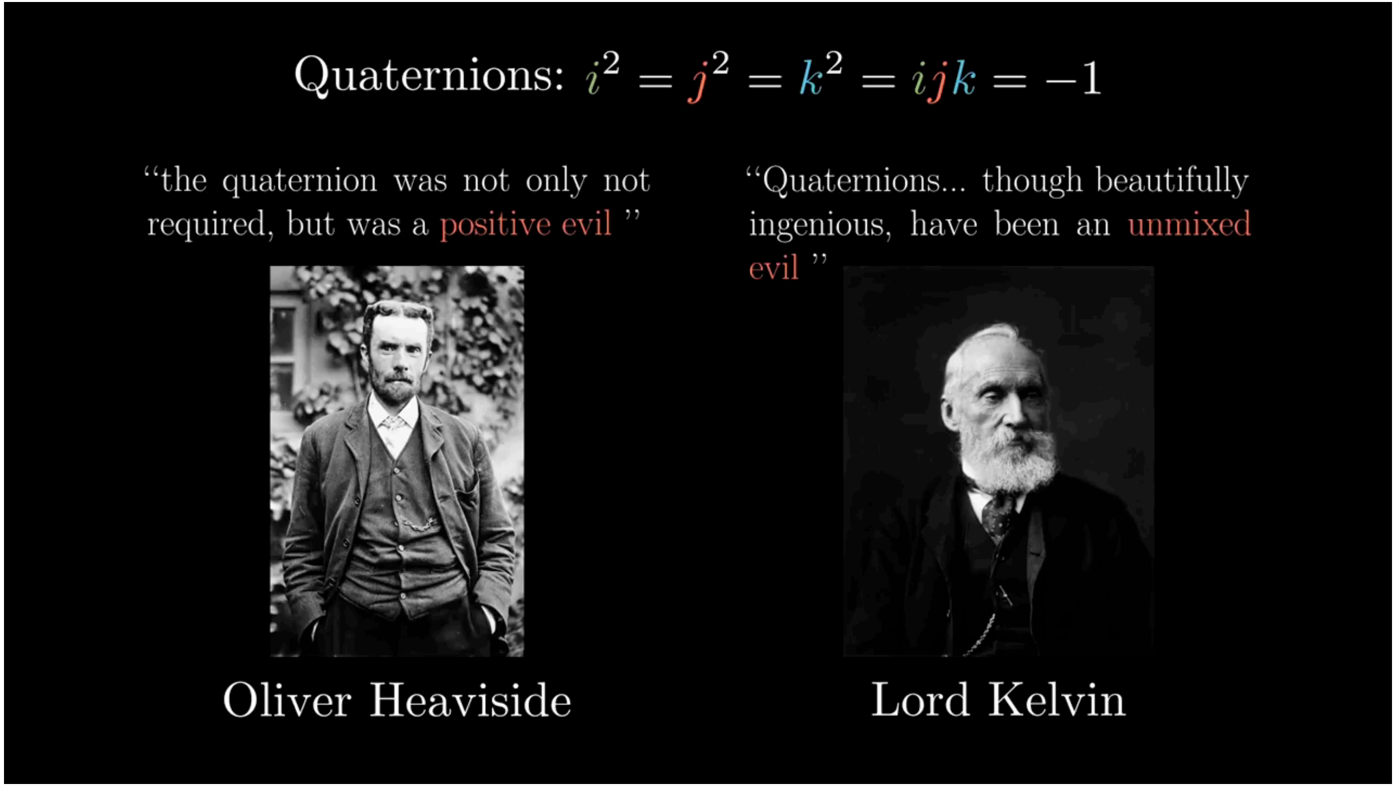
$$\mathbf{q} = q_0 + iq_1 + jq_2 + kq_3$$

- If **q** represents a rotation and **q'** represents a rotation, **qq'** represents **q** rotated by **q'**
- This follows very similar rules as matrix multiplication (i.e., non-commutative)

$$\mathbf{qq}' = \left(q_0 + iq_1 + jq_2 + kq_3\right)\left(q_0' + iq_1' + jq_2' + kq_3'\right)$$
$$= \left\langle ss' - \mathbf{v}\cdot\mathbf{v}', s\mathbf{v}' + s'\mathbf{v} + \mathbf{v}\times\mathbf{v}'\right\rangle$$

It's just like multiplying 2 rotation matrices together….

Quaternion Visualization



https://eater.net/quaternions/video/intro

Thank You !

Questions ?