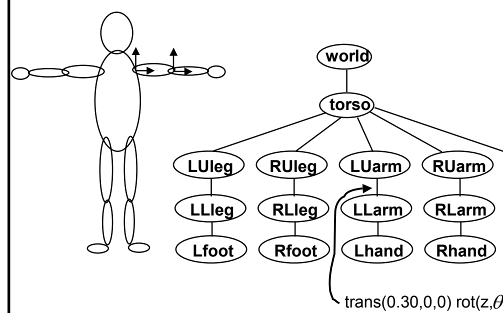
University of British Columbia

Animation: Representations

Transformation Hierarchies

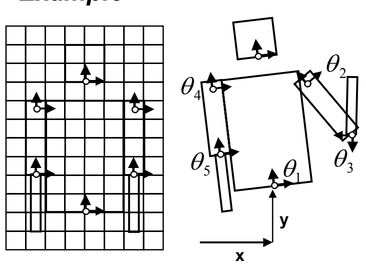
University of British Columbia



Transformation Hierarchies

University of British Columbia

Example




```
glTranslate3f(x,y,0);
glRotatef(θ1,0,0,1);
DrawBody();
glPushMatrix();
   glTranslate3f(0,7,0);
   DrawHead();
glPopMatrix();
glPushMatrix();
   glTranslate(2.5,5.5,0);
   glRotatef(θ2,0,0,1);
   DrawUArm();
   glTranslate(0,-3.5,0);
   glRotatef(θ3,0,0,1);
   DrawLArm();
glPopMatrix();
... (draw other arm)
```


Rotations

University of British Columbia

- orientation: absolute "book is lying face up"
  - *but require a reference coordinate frame*
- rotation: relative, non-unique
  - *spin globe +100 or –260*
- rotation matrix in n-D

$$v' = R \cdot v \qquad (A \cdot B)C = A(B \cdot C)$$
$$|v'| = |v| \qquad R_1 \cdot R_2 = R_3$$
$$R^T \cdot R = I \qquad R_A^{-1} = R$$
$$\det(R) = +1 \qquad I = R$$

continuous group $SO(n)$

Rotation DOFs

University of British Columbia

- 2D: 1 DOF
- 3D: 3 DOF
- 4D: 6 DOF

Rotations

University of British Columbia

SO(3)

- rotations do not commute $A \cdot B \neq B \cdot A$
- require at least 4 parameters for a smooth parameterization
  - *analogy: surface of the earth*
    - 2D surface, 3 params
- combing the hairy ball
  - *camera orientation: view object from any dir*

3x3 Rotation Matrix

University of British Columbia

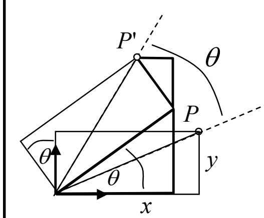
$$\begin{bmatrix} x' \\ y' \\ z' \\ h' \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ h \end{bmatrix}$$

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & T_x \\ m_{21} & m_{22} & m_{23} & T_y \\ m_{31} & m_{32} & m_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformations

University of British Columbia

Rotation



$Rotate(z,\theta)$

$$x' = x\cos\theta - y\sin\theta$$
$$y' = x\sin\theta + y\cos\theta$$
$$z' = z$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & & \\ \sin\theta & \cos\theta & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

glRotatef(angle,x,y,z);
glRotated(angle,x,y,z);

3x3 Rotation Matrix

University of British Columbia

- 9 elements
- 3 orthogonality constraints
- renormalization algorithms
- extracting pure rotational component (polar decomp)

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

$$R = \begin{bmatrix} \vec{a} & \vec{b} & \vec{c} \end{bmatrix}$$

$$R^{-1} = R^T$$

$a \bullet b = 0 \qquad |a| = 1$

$b \bullet c = 0 \qquad |b| = 1$

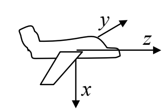
$a \bullet c = 0 \qquad |c| = 1$

... and determinant = 1

Fixed Angle Representations

University of British Columbia

- fixed angle representations
  - *RPY orientation:* z, y, x



roll **pitch** **yaw**

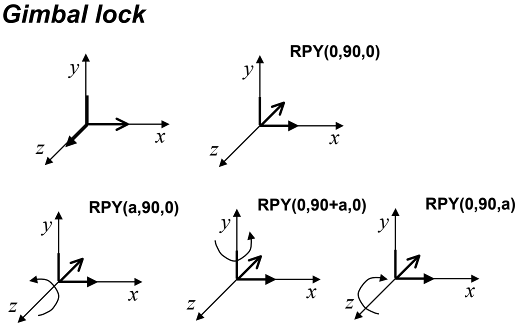
$$R_{RPY} = Rot(z,\alpha)\,Rot(y,\beta)\,Rot(x,\gamma)$$

- can use many ordering of axes
- Euler angles: z, x, z

Fixed Angle Representations

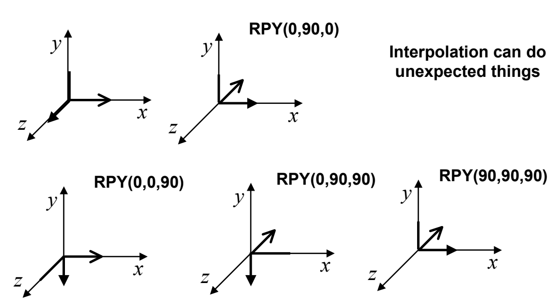
University of British Columbia

Gimbal lock



Fixed Angle Representations

University of British Columbia

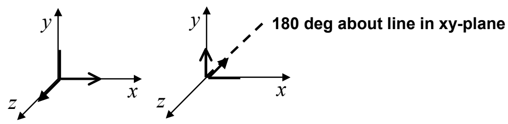


Interpolation can do unexpected things

Euler's Rotation Theorem

University of British Columbia

- can always go from one orientation to another with one rotation about a single axis



$$Rot(\vec{k},\theta) = \begin{bmatrix} k_x^2 v + c & k_x k_y v - k_z s & k_x k_z v + k_y s \\ k_x k_y v + k_z s & k_y^2 v + c & k_y k_z v - k_x s \\ k_x k_z v - k_y s & k_y k_z v + k_x s & k_z^2 v + c \end{bmatrix}$$

where

$c = \cos\theta$

$v = 1 - \cos\theta$

$s = \sin\theta$

Interpolation using Rot(k,θ)

University of British Columbia

$$R_A \longrightarrow R_B$$

$R(t)\,R_A$ **t=0** $R(t) = I$

t=1 $R(t) = R_B R_A^{-1}$

solve for k,θ such that $R_B R_A^{-1} = Rot(\vec{k},\theta)$

use $Rot(k, t\theta) R_A$

© Michiel van de Panne

Exponential Map

University of British Columbia

- idea: encode amount of rotation into magnitude of $\vec{k}$

$$\left|\vec{k}\right| = \theta \qquad Rot(\vec{k}, \left|\vec{k}\right|) \qquad \Re^3 \longrightarrow SO(3)$$

- axis definition undefined for no rotation
- singularities for $\left|\vec{k}\right| = 2\pi n$

© Michiel van de Panne

Quaternions

University of British Columbia

- review of complex numbers

$i^2 = -1$

$z = a + bi$

- quaternions

$q = w + xi + yj + zk$

$[x \quad y \quad z \quad w] = (s, \vec{v})$ $\qquad s = w$

$\vec{v} = [x \quad y \quad z]$

© Michiel van de Panne

Quaternions

University of British Columbia

$i^2 = -1 \qquad i \cdot j = -j \cdot i = k$

$j^2 = -1 \qquad j \cdot k = -k \cdot j = i$

$k^2 = -1 \qquad k \cdot i = -i \cdot k = j$

RH rule

- unit quaternions

$w^2 + x^2 + y^2 + z^2 = 1$

- addition $(s_1, v_1) + (s_2, v_2) = (s_1 + s_2, v_1 + v_2)$
- multiplication

$$(s_1, v_1) \cdot (s_2, v_2) = (s_1 \cdot s_2 - v_1 \bullet v_2, s_1 \cdot v_1 + s_2 \cdot v_2 + v_1 \times v_2)$$

© Michiel van de Panne

Quaternions

University of British Columbia

$$Rot(\vec{k},\theta) = (\cos\frac{\theta}{2}, \sin\frac{\theta}{2}\vec{k})$$

- rotation of a vector

$\vec{v}' = Rot(\vec{k},\theta)\vec{v} = q \cdot \tilde{v} \cdot \overline{q}$

$\tilde{v} = (0, \vec{v}) \qquad \overline{q} = (s, -\vec{v})$

- two successive rotations

$q_2(q_1 \cdot \tilde{v} \cdot \overline{q}_1) q_2$

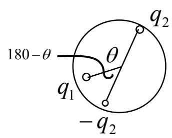
© Michiel van de Panne

Orientation Interpolation

- linear interpolation of quaternions
- but q and –q represent the same orientation

$q_1 \rightarrow q_2$ **or** $q_1 \rightarrow -q_2$ **?**

choose shorter path, use dot product to compute

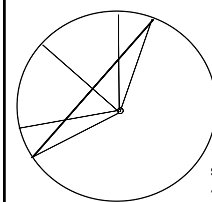


$$\cos\theta = q_1 \cdot q_2 = s_1 \cdot s_2 + v_1 \bullet v_2$$

Orientation Interpolation

SLERP instead of LERP

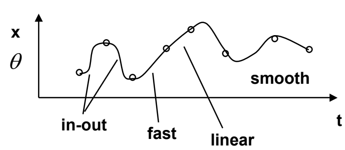
$$slerp(q_1, q_2, u) = \frac{\sin((1-u)\theta)}{\sin\theta} q_1 + \frac{\sin(u\theta)}{\sin\theta} q_2$$



smooth interpolation of multiple orientations:
-construct smooth curve on the 4D sphere

Representing motion

- DOF vs time



- alternative for motion through space: