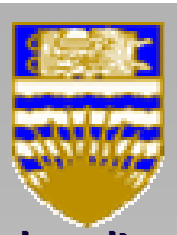
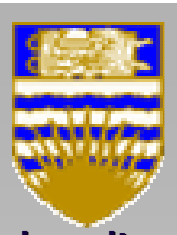


University of
British Columbia

An Introduction to Computer Animation

Michiel van de Panne

Overview

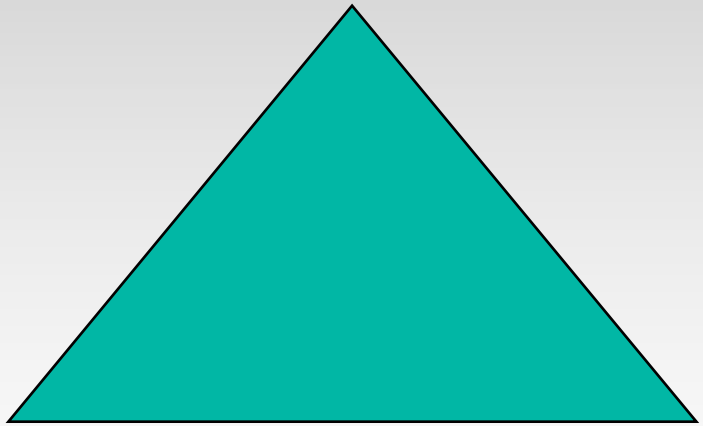


University of
British Columbia

(1) Creating Animations

(2) Representing Rotations

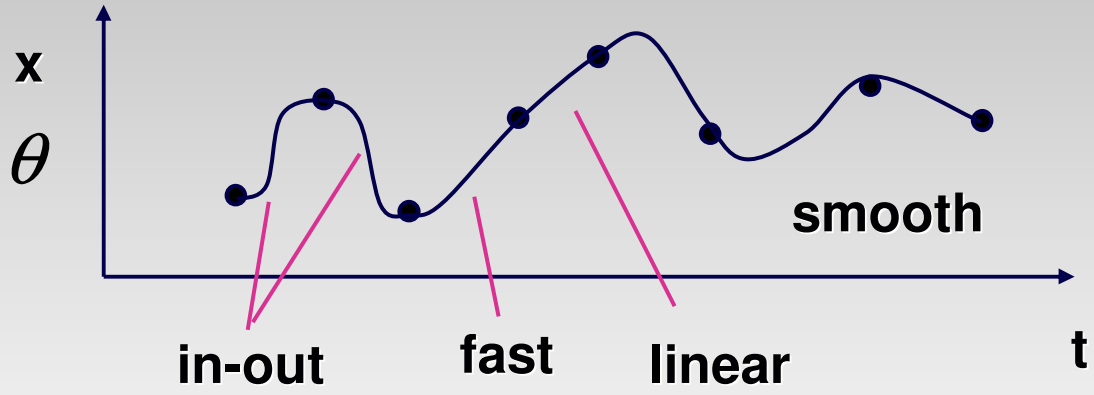
(1) Creating Animations



Representing motion

University of British Columbia

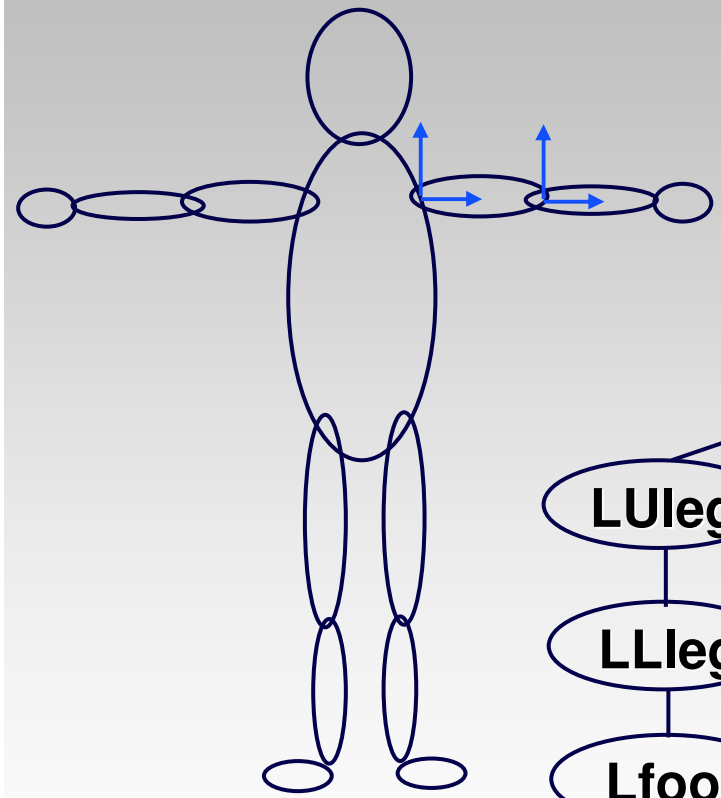
- DOF vs time
- cubic polynomial curves

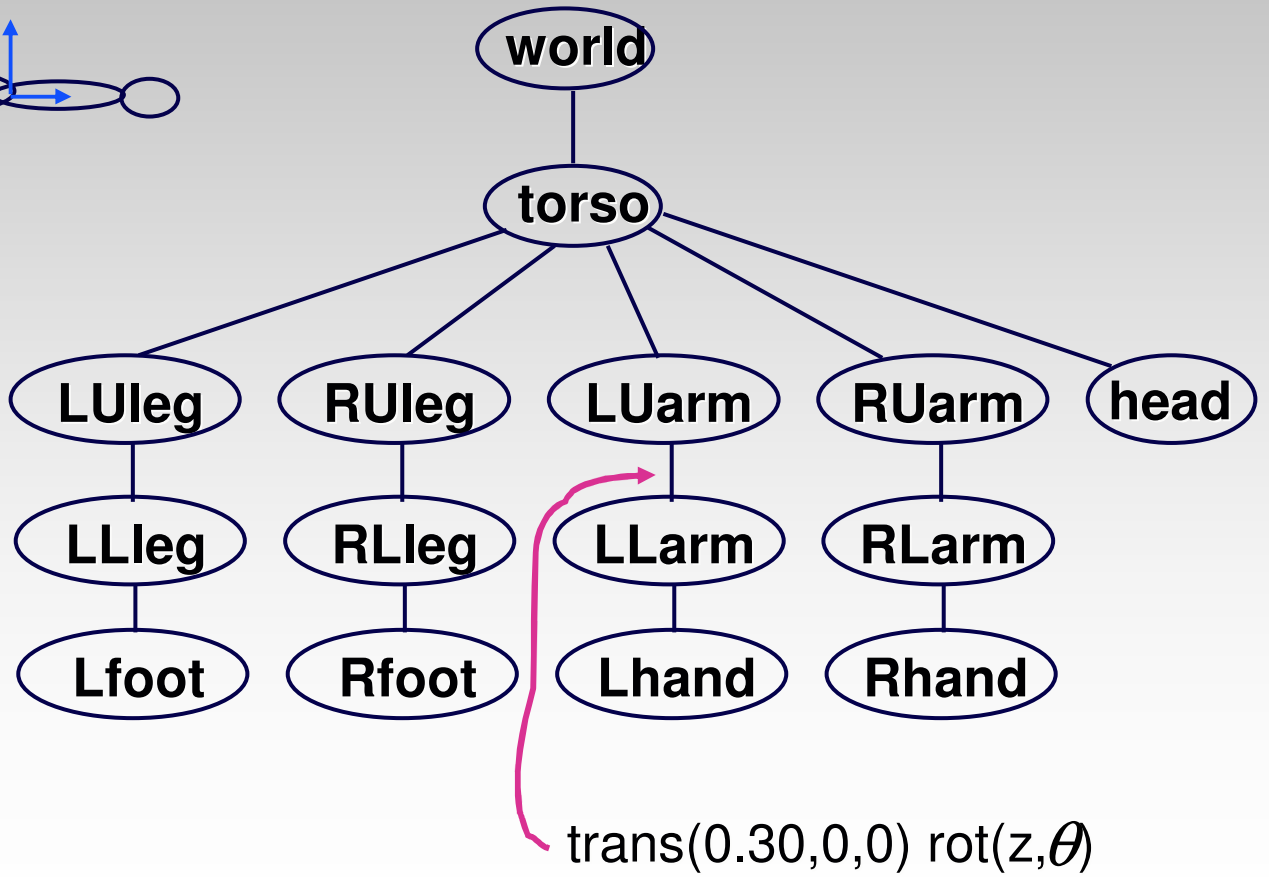


- alternative for motion through space:

(2) Representing Rotations

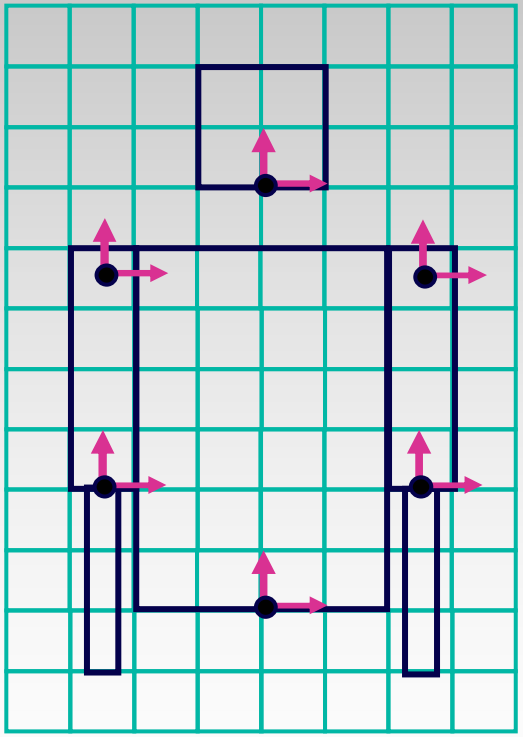
University of British Columbia

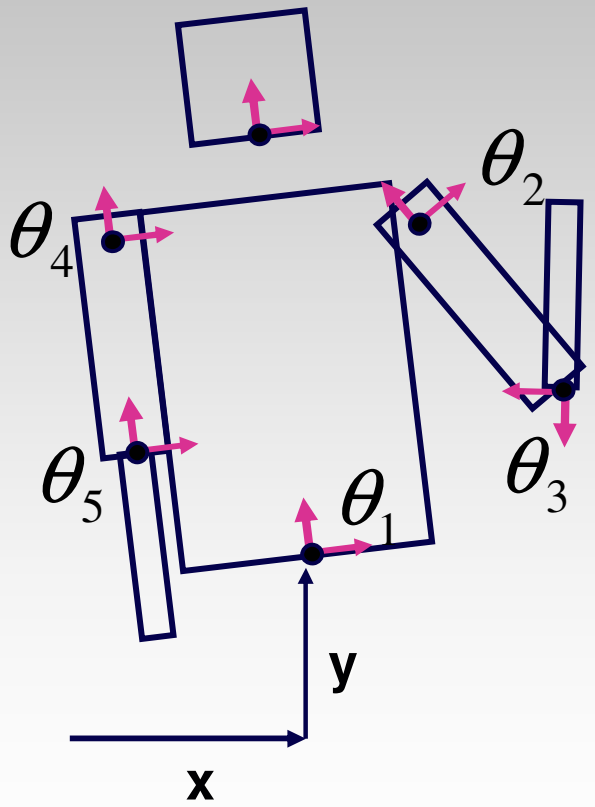




Transformation Hierarchies

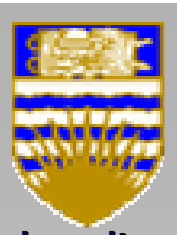
Example






```
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)
```

Rotation DOFs



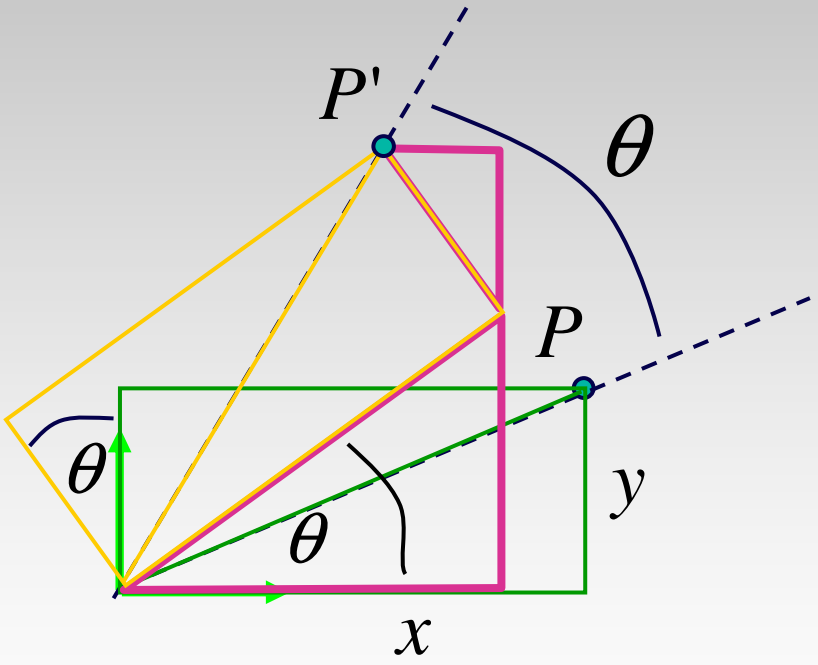
University of
British Columbia

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

Transformations

University of British Columbia

Rotation



$$Rotate(z,\theta)$$

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

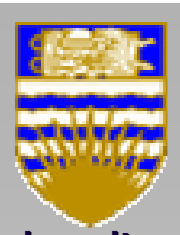
$$y' = x\sin\theta + y\cos\theta$$

$$z' = z$$

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

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

3x3 Rotation Matrix

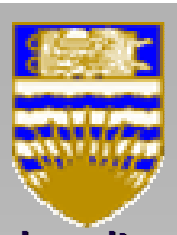


University of
British Columbia

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

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

3x3 Rotation Matrix



University of British Columbia

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

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

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

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

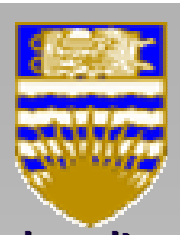
$$a \bullet b = 0 \qquad |a| = 1$$

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

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

... and determinant = 1

Rotations

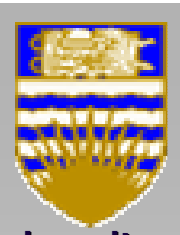


University of
British Columbia

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*

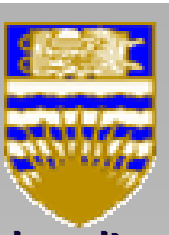
Rotations



University of
British Columbia

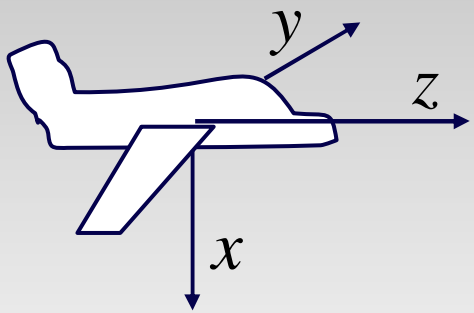
- orientation vs rotation?
- how to specify?
- how to interpolate?
- 2D vs 3D

Fixed Angle Representations



University of British Columbia

- fixed sequence of 3 rotations
  - *RPY orientation:* z, y, x



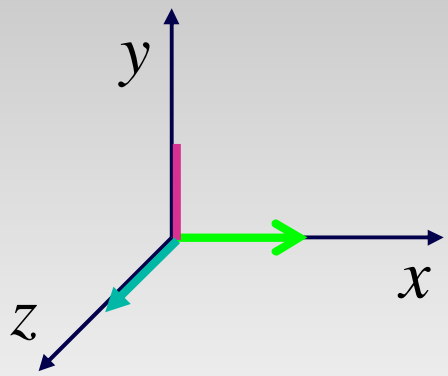
roll **pitch** **yaw**

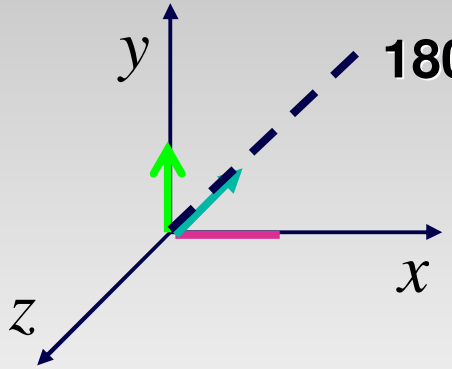
$$R_{RPY} = Rot(z,\alpha)\,Rot(y,\beta)\,Rot(x,\gamma)$$

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

Euler's Rotation Theorem

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





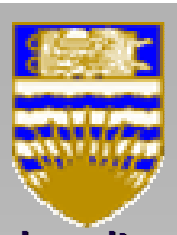
180 deg about line in xy-plane

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

where

$$c = \cos\theta$$
$$v = 1 - \cos\theta$$
$$s = \sin\theta$$

Quaternions



University of
British Columbia

- review of complex numbers

$$i^2 = -1$$

$$z = a + bi$$

- quaternions

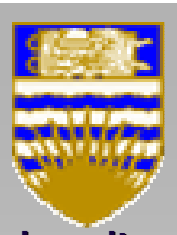
$$q = w + xi + yj + zk$$

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

s $\quad$ $\vec{v}$

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

Quaternions



University of
British Columbia

- rotation of a vector

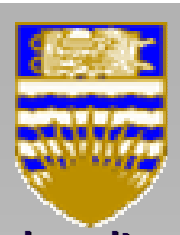
$$\vec{v}' = Rot(\vec{k},\theta)\,\vec{v} = q \cdot \tilde{v} \cdot \bar{q}$$

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

- two successive rotations

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

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

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