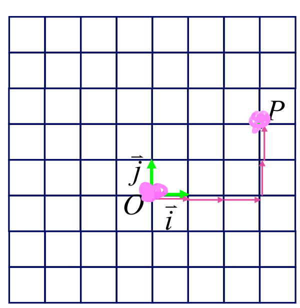
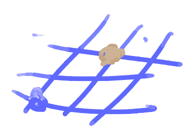
Affine Transformations (review)

- coordinate frames
- translate, scale, rotate, shear
- transformation hierarchies

Coordinate Frame



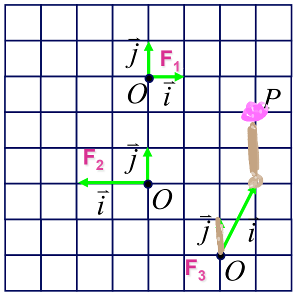
coordinate frame:
basis vectors + Origin



$$P = O + x\vec{i} + y\vec{j}$$

$$P = O + 3\vec{i} + 2\vec{j}$$

Working with Frames



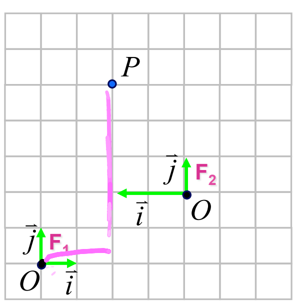
$$P = O + x\vec{i} + y\vec{j}$$

F_1 P(3,-1)

F_2 P(-1.5,2) -1.5, 2

F_3 P(1, 2)

Transformations as a change of frame



$$P = O + x\vec{i} + y\vec{j}$$

$$\begin{bmatrix} x \\ y \end{bmatrix}_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}_2 + x_2 \begin{bmatrix} 1 \\ 0 \end{bmatrix}_2 + y_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix}_2$$

F2

$$\begin{bmatrix} x \\ y \end{bmatrix}_1 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}_1 + x_2 \begin{bmatrix} -2 \\ 0 \end{bmatrix}_1 + y_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix}_1$$

F1

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_1 = \begin{bmatrix} -2 & 0 & 4 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_2 \quad \begin{matrix} x_2 \\ y_2 \\ 1 \end{matrix} \quad \begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix}$$

check: P_2(1,3) becomes P_1( )

1,3

2,5 ✓

$$P_1 = MP_2$$

University of
British Columbia

Affine transformations

- linear transformation + translations
- can be expressed as a 3x3 matrix + 3 vector

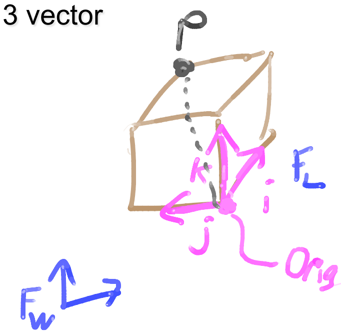
$$P' = M \cdot P + T$$

4x4 matrices

origin

$$\underbrace{\begin{bmatrix}x'\\y'\\z'\\1\end{bmatrix}}_{W} = \begin{bmatrix} i & j & k & o \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 1\end{bmatrix}\underbrace{\begin{bmatrix}x\\y\\z\\1\end{bmatrix}}_{L} \quad P_L$$

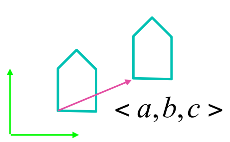
$$P_W = M P_L$$



University of
British Columbia

Translation

translate(a,b,c)



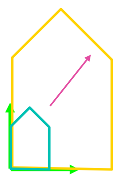
$$\begin{bmatrix}x'\\y'\\z'\\1\end{bmatrix} = \begin{bmatrix} 1 & & & a \\ & 1 & & b \\ & & 1 & c \\ & & & 1\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

glTranslatef(a,b,c);
glTranslated(a,b,c);

© Michiel van de Panne

Scaling

scale(a,b,c)

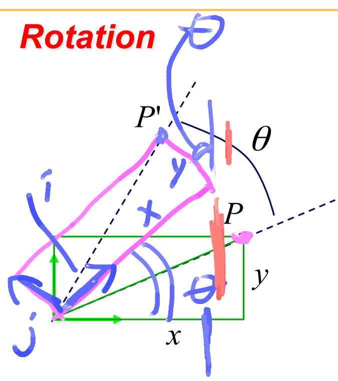


$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a & & & \\ & b & & \\ & & c & \\ & & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

glScalef(a,b,c);
glScaled(a,b,c);

University of
British Columbia

Rotation



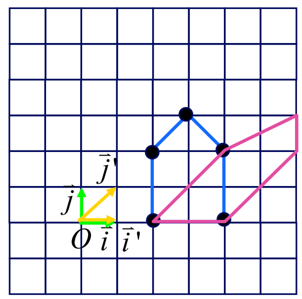
$Rotate(z,\theta)$

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

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{array}{c} \begin{array}{cccc} i & j & & \end{array} \\ \begin{bmatrix} c\theta & -s\theta & & \\ s\theta & c\theta & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \end{array} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

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

Shear

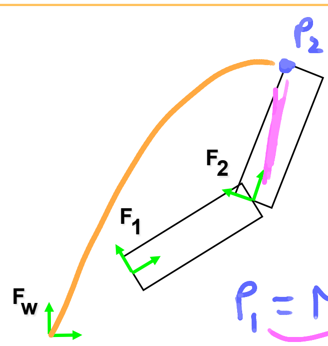


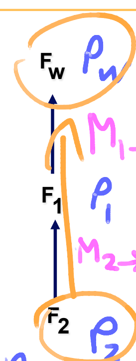
$$\begin{array}{c} \\ \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} \end{array} = \begin{array}{c} \begin{array}{cccc} i' & j' & & \end{array} \\ \begin{bmatrix} 1 & 1 & & \\ 0 & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \end{array} \begin{array}{c} \\ \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \end{array}$$

shear

University of
British Columbia

Transformation Hierarchies





Scene graph

$P_1 = M_{2\to 1} P_2$

$P_W = M_{1\to W} P_1$

$P_W = M_{1\to W} M_{2\to 1} P_2$

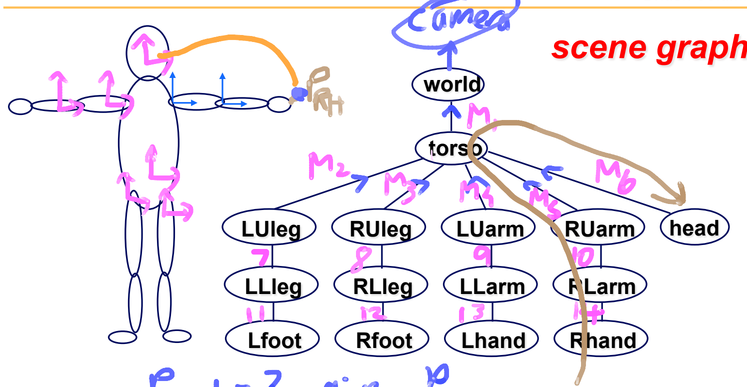
R to L

order for matrices

Transformation Hierarchies

University of British Columbia

scene graph



$P_{Head} = ?$ given P_{RH}

$P_{head} = M_6^{-1} M_5 M_{10} M_{14} P_{RH}$

Representing Rotations

Which of the following are rotation matrices?

$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \checkmark \qquad \begin{bmatrix} 1 & -1 & 1 \\ 1 & 1 & -1 \\ 1 & 0 & 1 \end{bmatrix} \times \qquad \begin{bmatrix} 1 & & \\ & -1 & \\ & & -1 \end{bmatrix}$$

$\det(R) = +1$

$a \times b = c$

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

numbers: 9

constraints: $|a| = 1$, $a \cdot b = 0$; $|b| = 1$, $b \cdot c = 0$; $|c| = 1$, $a \cdot c = 0$

DOF = # numbers − # constraints $= 9 - 6 = 3$

SO(3)

special orthogonal

Choices for Representing Rotations

x

	# numbers	# constraints	# DoF
(1) Rotation matrix	9	6	3
(2) Euler angles	3	0	3
(3) Angle - axis	4	1	3
(4) Quaternion	4	1	3

2 Euler Angles

3 successive rotations about changing axes

"gimbal lock": lose a DoF because of axis alignment

XYZ: $Rot(x, \alpha)\, Rot(y', \beta)\, Rot(z'', \gamma)$

ZYX: $Rot(z, \alpha)\, Rot(y', \beta)\, Rot(z'', \gamma)$

XYX: $Rot(x, \alpha)\, Rot(y', \beta)\, Rot(x'', \gamma)$

XXY:

12 possible orderings for valid Euler angles

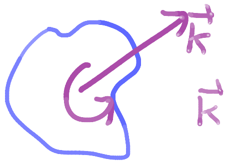
issues:

- unique? ZYX$(0°, 90°, 0°)$ ZYX$(90, 90, 90)$

3 Angle - Axis

Euler's Theorem

can go between any two orientations with only one rotation about some axis



$\vec{k} = \langle x, y, z \rangle$

glRotate(θ, x, y, z)

$$Rot(k,\theta) = \begin{bmatrix} xxC + c & xyC - zs & xzC + ys \\ yxC + zs & yyC + c & yzC - xs \\ zxC - ys & zyC + xs & zzC + c \end{bmatrix}$$

4 numbers

θ, k_x, k_y, k_z

constraints:

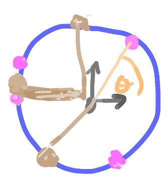
$|k| = 1$

where

$$\begin{aligned} c &= \cos\theta \\ s &= \sin\theta \\ C &= 1 - c \end{aligned}$$

4 Unit Quaternions

desired feature:



θ or $(\cos\theta, \sin\theta)$

point on a circle

179°, −135°, 90°

θ, ϕ

or x, y, z and $x^2+y^2+z^2=1$

point on a sphere

Multiplication of complex numbers

$e^{i\pi} + 1 = 0$

complex numbers

$z = a + bi \qquad i^2 = -1$

$z = \cos\theta + \sin\theta\, i$

$z_1 z_2 = (a_1 + b_1 i)(a_2 + b_2 i)$

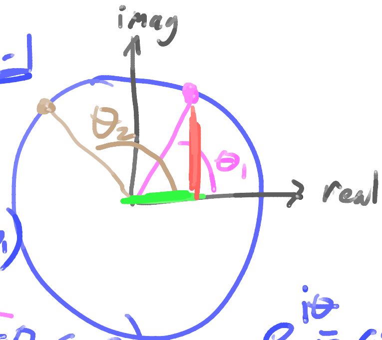
$= a_1 a_2 - b_1 b_2 + i(a_1 b_2 + a_2 b_1)$

$= \cos\theta_1 \cos\theta_2 - \sin\theta_1 \sin\theta_2$

$+ i(\cos\theta_1 \sin\theta_2 + \cos\theta_2 \sin\theta_1)$

$= \cos(\theta_1 + \theta_2) + i \sin(\theta_1 + \theta_2)$

multiplication = addition of angles



$e^{i\theta} = \cos\theta + i \sin\theta$

$e^{i\theta_1} e^{i\theta_2} = e^{i(\theta_1 + \theta_2)}$

$z_1 \; z_2$

Quaternions

$q = w + xi + yj + zk$

$= [w \;\; x \;\; y \;\; z]$

$= [s, \; v]$

unit quaternion $\quad \|q\| = 1 \qquad x^2 + y^2 + z^2 + w^2 = 1$

addition: $\quad q_1 + q_2 = [w_1 + w_2 \;\; x_1 + x_2 \;\; y_1 + y_2 \;\; z_1 + z_2]$

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

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

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

Quaternion Multiplication

$$q_1 \otimes q_2 = (w_1 + x_1 i + y_1 j + z_1 k)(w_2 + x_2 i + y_2 j + z_2 k)$$

	w_2	$x_2 i$	$y_2 j$	$z_2 k$
w_1	$w_1 w_2$	$w_1 x_2 i$	$w_1 y_2 j$	$w_1 z_2 k$
$x_1 i$	$w_2 x_1 i$	$-x_1 x_2$	$x_1 y_2 k$	$-x_1 z_2 j$
$y_1 j$	$w_2 y_1 j$	$-y_1 x_2 k$	$-y_1 y_2$	$y_1 z_2 i$
$z_1 k$	$w_2 z_1 k$	$x_2 z_1 j$	$-y_2 z_1 i$	$-z_1 z_2$

1 2 3 4

$$(s_1, \vec{v}_1) \times (s_2, \vec{v}_2) = \left(\underbrace{s_1 s_2}_{1} - \underbrace{\vec{v}_1 \cdot \vec{v}_2}_{2},\ \underbrace{s_1 \vec{v}_2}_{3} + \underbrace{s_2 \vec{v}_1}_{4} + \vec{v}_1 \times \vec{v}_2 \right)$$

Quaternion for an axis-angle rotation

$$Rot(k, \theta) = \left(\underbrace{\cos\tfrac{\theta}{2}}_{w},\ \underbrace{\sin\tfrac{\theta}{2}\, \vec{k}}_{x,y,z} \right)$$

Composition using multiplication

$q = q_1 \otimes q_2$ equivalent $M = M_1 M_2$

$q_1 \otimes q_2 \neq q_2 \otimes q_1$ ✗ commutative

$(q_1 \otimes q_2) \otimes q_3 = q_1 \otimes (q_2 \otimes q_3)$ ✓ associative

Quaternion rotation of a point (really a vector V)

$$Rot(\vec{k},\theta)\vec{V} = q \otimes \tilde{V} \otimes \bar{q}$$

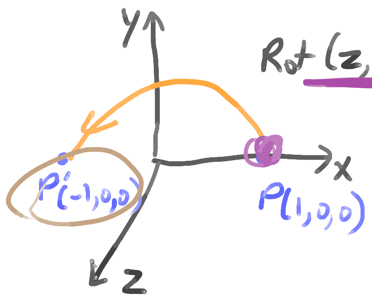
$$V' = MV$$

$$\tilde{V} = (0, V)$$

$$\bar{q} = (s, -v) = [w\ -x\ -y\ -z]$$

$$\left(\cos\frac{\theta}{2},\ \sin\frac{\theta}{2} k\right)$$

Example:



$Rot(z, 180°)$

$$\tilde{V} = [0, \langle 1,0,0 \rangle]$$

$$q = [0, \langle 0,0,1 \rangle]$$

$$\bar{q} = [0, \langle 0,0,-1 \rangle]$$

$$(q \otimes \tilde{V}) \otimes \bar{q} = (k \otimes i) \otimes (-k)$$

$$= j \otimes -k$$

$$= -i$$

$$= [0, \underbrace{\langle -1,0,0 \rangle}_{V'}]$$

Converting between representations

3x3 rotation matrix

Euler angles

$$M_1 = Rot(z, \theta_1)$$

$$M_2 = Rot(x, \theta_2)$$

$$M_3 = Rot(z, \theta_3)$$

$$M = M_1 M_2 M_3$$

angle-axis

$$Rot(k, \theta)$$

quaternions

$$Q = \begin{bmatrix} 1-2y^2-2z^2 & 2xy-2zw & 2xz+2yw \\ 2xy+2zw & 1-2x^2-2z^2 & 2yz-2xw \\ 2xz-2yw & 2yz+2xw & 1-2x^2-2y^2 \end{bmatrix}$$

Composition of Successive Rotations

3x3 matrices $M = M_1 M_2$

Euler angles $ZYZ(30°, -20°, 50°)$

Angle-axis $ZYZ(\ ,\ ,\ )$

$Rot(k, \theta)$

need to convert to matrices (and back)

Quaternions $q = q_1 \otimes q_2$

Quaternion Features

- compact 4 numbers (vs 9 for a matrix)
- continuous
- unique (almost: q and $-q$ represent the same orientation)
- simple composition

Reality Check all widely used.

3x3 matrices: part of transformation matrix

Euler angles: most compact, mocap data

Axis-angle: glRotate(θ, x, y, z)

Quaternions: best choice for interpolation

Linear Interpolation of Quaternions

(Book 3.3)

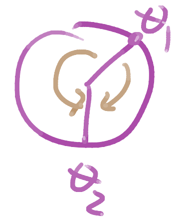
Idea: just interpolate the components

$Q(t) = (1-t)Q_1 + tQ_2 \qquad t \in [0,1]$

potential issues

① interpolate from $Q_1 \rightarrow Q_2$

$Q_1 \rightarrow -Q_2$



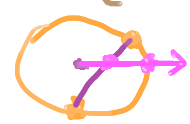
$\cos\theta_1 = Q_1 \cdot Q_2$

$\cos\theta_2 = Q_1 \cdot (-Q_2)$

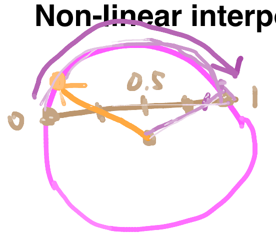
take smaller of θ_1 and θ_2

② no longer a unit quaternion?

→ easy to renormalize



Non-linear interpolation speed



evenly spaced linear interpolation

≠ evenly spaced orientations

solution: Spherical Linear Interpolation (SLERP)

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

$v=0$

$v=1 \qquad v \in [0,1]$

$\cos\theta = q_1 \cdot q_2$

→ smooth quaternion interpolation thru a sequence of points also possible

Linear Interpolation of Rotation Matrices

e.g

$$0.5\underbrace{\begin{bmatrix}1 & & \\ & 1 & \\ & & 1\end{bmatrix}}_{\text{no rotation}} + 0.5\underbrace{\overbrace{\begin{bmatrix}0 & -1 & \\ 1 & 0 & \\ & & 1\end{bmatrix}}^{M}}_{90^\circ \text{ rot around } z} = \begin{bmatrix}0.5 & -0.5 & \\ 0.5 & 0.5 & \\ & & 1\end{bmatrix} \overset{?}{=} \frac{1}{\sqrt{2}} Rot(z, 45^\circ)$$

object shrinks!

Better:

$$M = Rot(z, 90^\circ)$$

$$M^2 = Rot(z, 90^\circ)Rot(z, 90^\circ)$$

$$= Rot(z, 180^\circ)$$

$$M^{0.5} = Rot(z, 45^\circ)$$

matrix exponential → expensive, and a few caveats (but it does work!)