SCHOOL of ELECTRICAL ENGINEERING & COMPUTING
FACULTY of ENGINEERING & BUILT ENVIRONMENT

The UNIVERSITY of NEWCASTLE

Comp3320/6370 Computer Graphics

LECTURE 12: Quaternions

Based on chapter 4 of the book “Real-Time Rendering”
by Akenine-Möller et al., 4th edition 2018.

Definition

- A quaternion is a 4-tuple expresses as

$$\begin{aligned}\hat{\mathbf{q}} &= (\mathbf{q}_v,\ q_w)\\ &= iq_x + jq_y + kq_z + q_w\\ &= \mathbf{q}_v + q_w\end{aligned}$$

$$i^2 = j^2 = k^2 = -1$$

$$\begin{aligned} ij &= -ji = k\\ jk &= -kj = i\\ ki &= -ik = j\end{aligned}$$

$$\mathbf{q}_v = iq_x + jq_y + kq_z = (q_x, q_y, q_z)$$

- Multiplication:

$$\hat{\mathbf{q}}\,\hat{\mathbf{r}} = (\mathbf{q}_v \times \mathbf{r}_v + q_w \mathbf{r}_v + r_w \mathbf{q}_v,\ q_w r_w - \mathbf{q}_v \cdot \mathbf{r}_v)$$

$$\hat{\mathbf{q}}\,\hat{\mathbf{r}} \neq \hat{\mathbf{r}}\,\hat{\mathbf{q}}$$

$$s\,\hat{\mathbf{q}} = \hat{\mathbf{q}}\,s = (s\,\mathbf{q}_v, s\,q_w)$$

- Addition:

$$\hat{\mathbf{q}} + \hat{\mathbf{r}} = (\mathbf{q}_v + \mathbf{r}_v,\ q_w + r_w)$$

Operations

- Conjugate: $\hat{\mathbf{q}}^* = (-\mathbf{q}_v,\ q_w)$
- Norm: $$\begin{aligned} n(\hat{\mathbf{q}}) &= \hat{\mathbf{q}}\hat{\mathbf{q}}^* = \hat{\mathbf{q}}^*\hat{\mathbf{q}} = \mathbf{q}_v \cdot \mathbf{q}_v + q_w^2 \\ &= q_x^2 + q_y^2 + q_z^2 + q_w^2 \\ &= \|\hat{\mathbf{q}}\|^2 \end{aligned}$$
- Identity: $\hat{\mathbf{i}} = (\mathbf{0},\ 1)$ $$\begin{aligned} \hat{\mathbf{i}}\,\hat{\mathbf{q}} &= (\mathbf{0} \times \mathbf{q}_v + 1\,\mathbf{q}_v + q_w \mathbf{0},\ 1\,q_w - \mathbf{0} \cdot \mathbf{q}_v) \\ &= (\mathbf{0} + \mathbf{q}_v + \mathbf{0},\ q_w - 0) \\ &= \mathbf{q}_v + q_w \end{aligned}$$
- Inverse: $$\hat{\mathbf{q}}^{-1} = \frac{1}{n(\hat{\mathbf{q}})}\hat{\mathbf{q}}^*$$

$$n(\hat{\mathbf{q}}) = \hat{\mathbf{q}}\hat{\mathbf{q}}^* \qquad 1 = \frac{\hat{\mathbf{q}}\hat{\mathbf{q}}^*}{n(\hat{\mathbf{q}})} \qquad \hat{\mathbf{q}}^{-1} = \frac{(\hat{\mathbf{q}}^{-1}\hat{\mathbf{q}})\hat{\mathbf{q}}^*}{n(\hat{\mathbf{q}})}$$

Rules

- Conjugate rules:

$$(\hat{\mathbf{q}}^*)^* = \hat{\mathbf{q}}$$
$$(\hat{\mathbf{q}}+\hat{\mathbf{r}})^* = \hat{\mathbf{q}}^*+\hat{\mathbf{r}}^*$$
$$(\hat{\mathbf{q}}\hat{\mathbf{r}})^* = \hat{\mathbf{r}}^*\hat{\mathbf{q}}^*$$

- Norm rules:

$$n(\hat{\mathbf{q}}^*) = n(\hat{\mathbf{q}})$$
$$n(\hat{\mathbf{q}}\hat{\mathbf{r}}) = n(\hat{\mathbf{q}})n(\hat{\mathbf{r}})$$

- Linearity:

$$\hat{\mathbf{p}}(\hat{\mathbf{q}}+\hat{\mathbf{r}}) = \hat{\mathbf{p}}\hat{\mathbf{q}}+\hat{\mathbf{p}}\hat{\mathbf{r}}$$
$$(\hat{\mathbf{q}}+\hat{\mathbf{r}})\hat{\mathbf{p}} = \hat{\mathbf{q}}\hat{\mathbf{p}}+\hat{\mathbf{r}}\hat{\mathbf{p}}$$

- Associativity:

$$\hat{\mathbf{p}}(\hat{\mathbf{q}}\hat{\mathbf{r}}) = (\hat{\mathbf{p}}\hat{\mathbf{q}})\hat{\mathbf{r}}$$

Alternate Form

- Unit quaternions, $n(\hat{\mathbf{q}})=1$, can be written as

$$\hat{\mathbf{q}} = \sin(\phi)\mathbf{u}_q + \cos(\phi)$$

for some unit 3D vector $\mathbf{u}_q$

- Note that for $\hat{\mathbf{q}} = \sin(\phi)\mathbf{u}_q + \cos(\phi) = (\mathbf{q}_v,\ q_w)$

$$\mathbf{q}_v = \sin(\phi)\mathbf{u}_q \qquad q_w = \cos(\phi)$$

- Another form is $\hat{\mathbf{q}} = e^{\phi\,\mathbf{u}_q}$

which gives rise to

$$\log(\hat{\mathbf{q}}) = \log(e^{\phi\,\mathbf{u}_q}) = \phi\,\mathbf{u}_q$$

$$\hat{\mathbf{q}}^{\boxed{t}} = (\sin(\phi)\mathbf{u}_q + \cos(\phi))^t = e^{\phi\,t\,\mathbf{u}_q} = \sin(\phi\,t)\mathbf{u}_q + \cos(\phi\,t)$$

Rotations

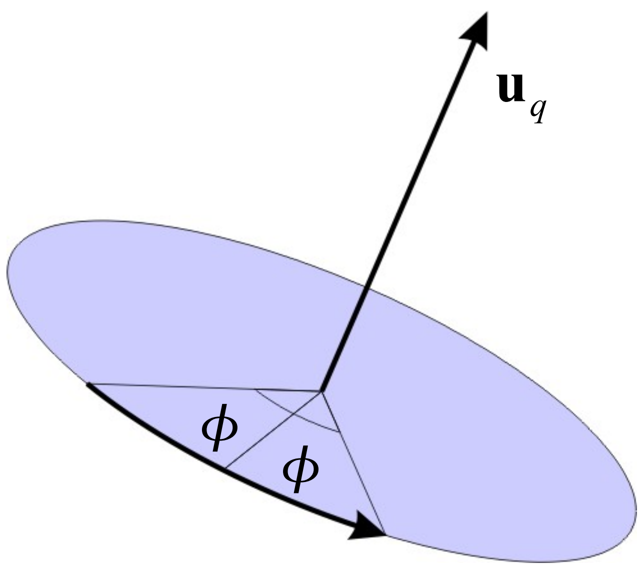
- Given a point $\mathbf{p} = (p_x \quad p_y \quad p_z \quad p_w)^T$
  create a quaternion as $\hat{\mathbf{p}} = ip_x + jp_y + kp_z + p_w$
- Given a unit quaternion $\hat{\mathbf{q}} = (\sin(\phi)\mathbf{u}_q,\ \cos(\phi))$

$$\hat{\mathbf{q}}\,\hat{\mathbf{p}}\,\hat{\mathbf{q}}^{-1}$$

  rotates the point $\mathbf{p}$ about the axis $\mathbf{u}_q$ by an angle of 2ϕ

- Since $n(\hat{\mathbf{q}}) = 1$

$$\hat{\mathbf{q}}\,\hat{\mathbf{p}}\,\hat{\mathbf{q}}^{-1} = \hat{\mathbf{q}}\,\hat{\mathbf{p}}\,\hat{\mathbf{q}}^{*}$$



- Given another unit quaternion $\hat{\mathbf{r}}$
  we can first apply $\hat{\mathbf{q}}$ then $\hat{\mathbf{r}}$ by

$$\hat{\mathbf{r}}(\hat{\mathbf{q}}\,\hat{\mathbf{p}}\,\hat{\mathbf{q}}^{*})\hat{\mathbf{r}}^{*} = (\hat{\mathbf{r}}\,\hat{\mathbf{q}})\hat{\mathbf{p}}(\hat{\mathbf{r}}\,\hat{\mathbf{q}})^{*} = \hat{\mathbf{c}}\,\hat{\mathbf{p}}\,\hat{\mathbf{c}}^{*}$$

$$\hat{\mathbf{c}} = \hat{\mathbf{r}}\,\hat{\mathbf{q}}$$

Matrix Conversion

- Matrix-Vector multiplication is more efficient than $\hat{\mathbf{q}}\,\hat{\mathbf{p}}\,\hat{\mathbf{q}}^*$
- We can convert a unit quaternion $\hat{\mathbf{q}}$ to a matrix $\mathbf{M}^q$ with

$$\mathbf{M}^q = \begin{pmatrix} 1-2(q_y^2+q_z^2) & 2(q_xq_y-q_wq_z) & 2(q_xq_z+q_wq_y) & 0 \\ 2(q_xq_y+q_wq_z) & 1-2(q_x^2+q_z^2) & 2(q_yq_z-q_wq_x) & 0 \\ 2(q_xq_z-q_wq_y) & 2(q_yq_z+q_wq_x) & 1-2(q_x^2+q_y^2) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Converting from the matrix back to a quaternion

$$\begin{aligned} m_{21}^q - m_{12}^q &= 4q_wq_x \\ m_{02}^q - m_{20}^q &= 4q_wq_y \\ m_{10}^q - m_{01}^q &= 4q_wq_z \\ \mathrm{tr}(\mathbf{M}^q) &= 4q_w^2 \end{aligned}$$

$$q_w = \frac{1}{2}\sqrt{\mathrm{tr}(\mathbf{M}^q)} \qquad q_x = \frac{m_{21}^q - m_{12}^q}{4q_w}$$

$$q_y = \frac{m_{02}^q - m_{20}^q}{4q_w} \qquad q_z = \frac{m_{10}^q - m_{01}^q}{4q_w}$$

Matrix Conversion

- What if $q_w = 0$?

$$\mathbf{M}^q = \begin{pmatrix} 1-2(q_y^2+q_z^2) & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) & 0 \\ 2(q_x q_y + q_w q_z) & 1-2(q_x^2+q_z^2) & 2(q_y q_z - q_w q_x) & 0 \\ 2(q_x q_z - q_w q_y) & 2(q_y q_z + q_w q_x) & 1-2(q_x^2+q_y^2) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 2q_x^2-1 & 2q_x q_y & 2q_x q_z & 0 \\ 2q_x q_y & 2q_y^2-1 & 2q_y q_z & 0 \\ 2q_x q_z & 2q_y q_z & 2q_z^2-1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$q_x = \sqrt{\frac{m_{00}^q - 1}{2}} \qquad q_y = \sqrt{\frac{m_{11}^q - 1}{2}} \qquad q_z = \sqrt{\frac{m_{22}^q - 1}{2}}$$

Spherical Linear Interpolation

- Given two unit quaternions $\hat{\mathbf{q}}$ and $\hat{\mathbf{r}}$, we can interpolate between the two with

$$\hat{\mathbf{s}}(\hat{\mathbf{q}}, \hat{\mathbf{r}}, t) = (\hat{\mathbf{r}}\hat{\mathbf{q}}^{-1})^t \hat{\mathbf{q}}$$

for $t \in [0, 1]$

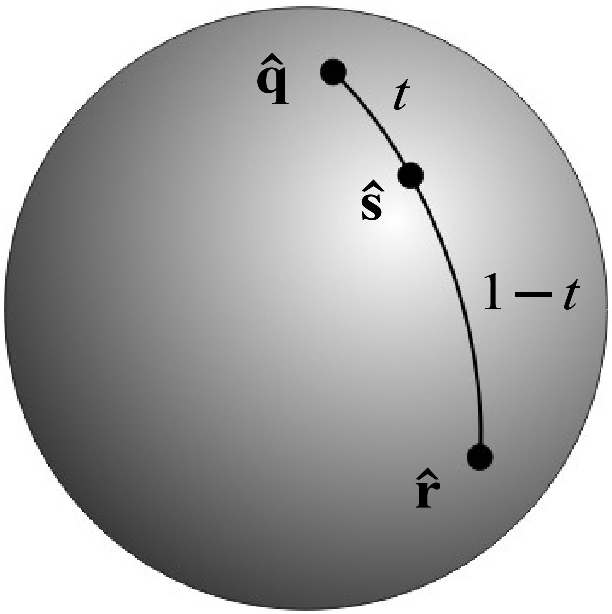
If the length of the arc from $\hat{\mathbf{q}}$ to $\hat{\mathbf{r}}$ is l then the length of the arc from $\hat{\mathbf{q}}$ to $\hat{\mathbf{s}}$ is $l\,t$ and $\hat{\mathbf{s}}$ to $\hat{\mathbf{r}}$ is $l(1-t)$

- Easier implemented in software as

$$\hat{\mathbf{s}}(\hat{\mathbf{q}}, \hat{\mathbf{r}}, t) = \frac{\sin(\phi(1-t))}{\sin(\phi)}\hat{\mathbf{q}} + \frac{\sin(\phi t)}{\sin(\phi)}\hat{\mathbf{r}}$$

$$\cos(\phi) = q_x r_x + q_y r_y + q_z r_z + q_w r_w$$

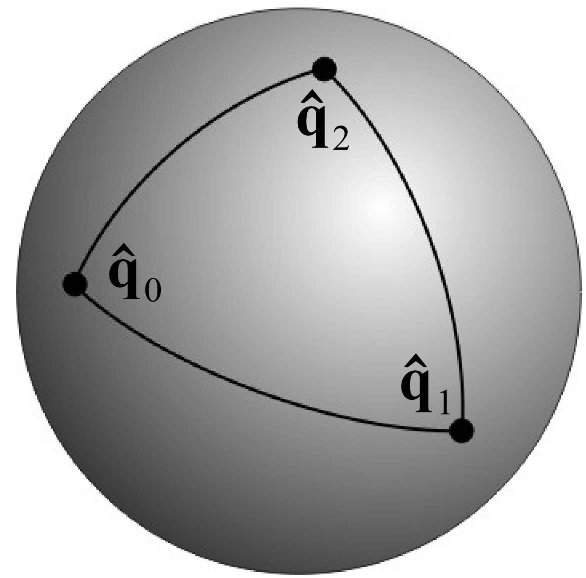
$$\hat{\mathbf{s}}(\hat{\mathbf{q}}, \hat{\mathbf{r}}, t) = \mathrm{slerp}(\hat{\mathbf{q}}, \hat{\mathbf{r}}, t)$$



Slerping Through Many Quaternions

- Suppose we have many orientations, and we want to move from one to the other

$$\hat{\mathbf{q}}_0,\ \hat{\mathbf{q}}_1, \ldots, \hat{\mathbf{q}}_{n-1}$$



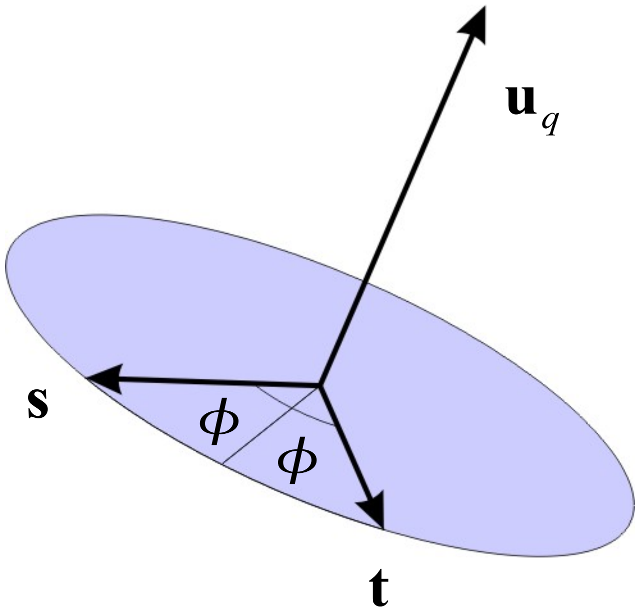
- If we interpolate between successive quaternions we observe sudden jerks as the direction changes between interpolations.
- To make the interpolations smooth splines are used.

$$\mathrm{squad}(\hat{\mathbf{q}}_i, \hat{\mathbf{q}}_{i+1}, \hat{\mathbf{a}}_i, \hat{\mathbf{a}}_{i+1}, t) = \mathrm{slerp}(\mathrm{slerp}(\hat{\mathbf{q}}_i, \hat{\mathbf{q}}_{i+1}, t), \mathrm{slerp}(\hat{\mathbf{a}}_i, \hat{\mathbf{a}}_{i+1}, t), 2t(1-t))$$

$$\hat{\mathbf{a}}_i = \hat{\mathbf{b}}_i = \hat{\mathbf{q}}_i \exp\left(-\frac{\log(\hat{\mathbf{q}}_i^{-1}\hat{\mathbf{q}}_{i-1}) + \log(\hat{\mathbf{q}}_i^{-1}\hat{\mathbf{q}}_{i+1})}{4}\right)$$

Rotation from $\mathbf{s}$ to $\mathbf{t}$

- How can we rotate from direction $\mathbf{s}$ to direction $\mathbf{t}$?
- Normalize $\mathbf{s}$ and $\mathbf{t}$
- Rotates through 2ϕ radians,
- about the vector $\mathbf{u}_q$

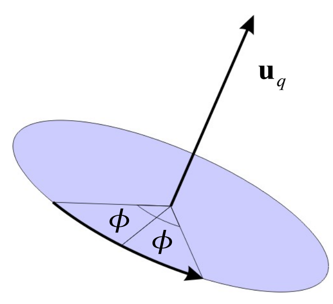


$$\mathbf{u}_q = \frac{\mathbf{s} \times \mathbf{t}}{\|\mathbf{s} \times \mathbf{t}\|}$$

$$\mathbf{s} \cdot \mathbf{t} = \cos(2\phi)$$
$$\|\mathbf{s} \times \mathbf{t}\| = \sin(2\phi)$$

$$\hat{\mathbf{q}} = (\sin(\phi)\mathbf{u}_q,\ \cos(\phi))$$

$$\hat{\mathbf{q}} = \left(\frac{1}{\sqrt{2(1+\mathbf{s}\cdot\mathbf{t})}}(\mathbf{s} \times \mathbf{t}),\ \frac{\sqrt{2(1+\mathbf{s}\cdot\mathbf{t})}}{2} \right)$$



Rotation from $\mathbf{s}$ to $\mathbf{t}$

- Often it is more convenient to represent the rotation from $\mathbf{s}$ to $\mathbf{t}$ in matrix form

$$\mathbf{R}(\mathbf{s},\mathbf{t}) = \begin{pmatrix} a+hv_x^2 & hv_xv_y-v_z & hv_xv_z+v_y & 0 \\ hv_xv_y+v_z & a+hv_y^2 & hv_yv_z-v_x & 0 \\ hv_xv_z-v_y & hv_yv_z+v_x & a+hv_z^2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where

$$\mathbf{v} = \mathbf{s}\times\mathbf{t} \qquad a = \mathbf{s}\cdot\mathbf{t} \qquad h = \frac{1-a}{\mathbf{v}\cdot\mathbf{v}}$$