Computer Graphics

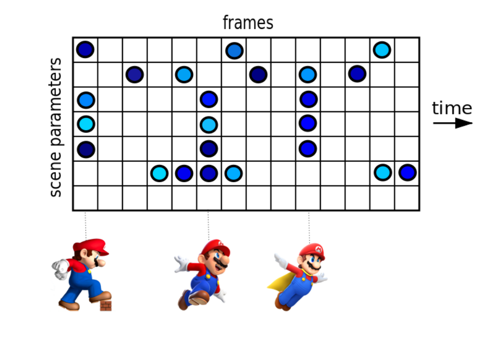
Dr. rer. nat. Martin Möddel

April 6, 2021

Institut für Biomedizinische Bildgebung

Rotations in Computer Graphics

Motivation



- Animations → keyframing
- Standard interpolation technqiques lead to practical problems
- Besides transations, rotations are the only motion leaving the shape of an object unchanged

Rotations

- The set of all distance preserving transformations of the euclidean space of dimension n is the orthogonal group $O(n)$
- Equivalently, it is the group of orthogonal $n \times n$ matrices with the matrix multiplication as group operation

$$O(n) = \left\{ A \in \mathbb{R}^{n \times n} | A^T A = A A^T = I_n \right\} \tag{1}$$

- An important subset of these transformations are the matrices, which have determinant 1, the special orthogonal group

$$SO(n) = \left\{ A \in \mathbb{R}^{n \times n} | A^T A = A A^T = I_n \text{ and } \det(A) = 1 \right\} \tag{2}$$

- We also refer to this group as rotation group, since its elements are rotations
- In this lecture we will specifically discuss the group of rotations in 3D Euclidean space $SO(3)$

Basic Rotations

- Basic rotations rotate around one of the canonical axis x, y, z
- They are specified by an angle φ (right hand rule) like

$$\mathrm{R}_x(\varphi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{pmatrix} \tag{3}$$

$$\mathrm{R}_y(\varphi) = \begin{pmatrix} \cos\varphi & 0 & \sin\varphi \\ 0 & 1 & 0 \\ -\sin\varphi & 0 & \cos\varphi \end{pmatrix} \tag{4}$$

$$\mathrm{R}_z(\varphi) = \begin{pmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5}$$

General Rotations

Figure 1: "Portrait of Leonhard Euler (1707-1783)" by the Kunstmuseum Basel in public domain

General Rotations

- General rotations can be achieved by concatenation of the standard rotations
- One way is to use Euler angles α, β, and γ, which can also be used to describe the orientation of a rigid body
- The classic Euler angles correspond to three rotations around two axes
    - *z-x-z*, *x-y-x*, *y-z-y*, *z-y-z*, *x-z-x*, *y-x-y*
- Another possibility to describe rotations are the Tait-Bryan angles with the corresponding rotations are around
    - *x-y-z*, *y-z-x*, *z-x-y*, *x-z-y*, *z-y-x*, *y-x-z*

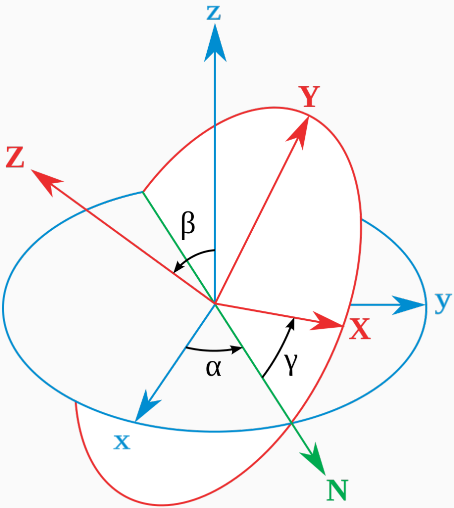
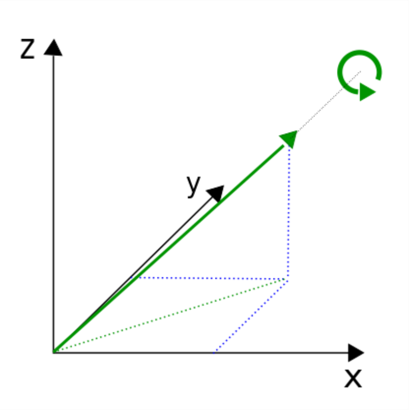


Figure 2: "Euler angles" by Lionel Brits licensed under the CC BY 3.0

Question

Find the Euler angles for a rotation of $\varphi = 80^\circ$ around the rotation axis $\mathrm{n} = \frac{1}{\sqrt{3}}(1,1,1)^T$.



General Rotations

Reminder
Within our camera coordinate system we have the viewing direction aligned along the negative z-axis, the upwards direction aligned along the y-axis and the right hand side along the x-axis.

- For camera transformations we use the y-x-z rotation
    - The Rotation by $\alpha = h$ (head) around the y makes the camera shake its head
    - The Rotation by $\beta = p$ (pitch) around the x makes the camera nod its head
    - The Rotation by $\gamma = r$ (roll) around the z makes the camera roll around the viewing direction
- Other rotation orders can also be useful depending on the orientation of your object in your local coordinate system

General Rotations

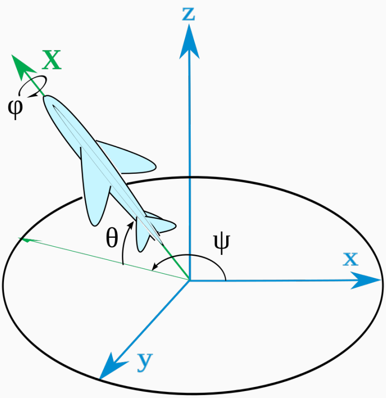


Figure 3: “Plane with ENU embedded axes” by Juansempere licensed under CC BY-SA 3.0. Rotation order is φ (yaw), θ (pitch), ψ (roll) around z-y-x

General Rotation - Angle Extraction

- In some situations it might be useful to extract the Euler parameters h, p and r from

$$\mathsf{R} = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \tag{6}$$

- This can be done using

$$\mathsf{E}\left(h, p, r\right) = \begin{pmatrix} \cos r \cos h - \sin r \sin p \sin h & -\sin r \cos p & \cos r \sin h + \sin r \sin p \cos h \\ \sin r \cos h + \cos r \sin p \sin h & \cos r \cos p & \sin r \sin h - \cos r \sin p \cos h \\ -\cos p \sin h & \sin p & \cos p \cos h \end{pmatrix} \tag{7}$$

- Whenever we are not in a gimbal lock situation we get

$$h = \text{atan2}(-R_{31}, R_{33}) \tag{8}$$

$$p = \arcsin(R_{32}) \tag{9}$$

$$r = \text{atan2}(-R_{12}, R_{22}) \tag{10}$$

- Due to the ambiguity of arcsin we might not be able to recover the angle, if R was not created with an angle $p \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$

General Rotations - Rotation About Arbitrary Axis

- Sometimes one has only a normalized direction r and a rotation angle α specified and wants to write down the corresponding rotation matrix
- Completing r to a orthonormal basis $(\mathsf{r}, \mathsf{s}, \mathsf{t})$ we define the transformation from the *rst*-coordinate system to the canonical *xyz*-coordinate system

$$\mathsf{M} = \begin{pmatrix} \mathsf{r} & \mathsf{s} & \mathsf{t} \end{pmatrix} \tag{11}$$

- The final transformation can be obtained by transforming from *rst* to *xyz*, followed by a rotation by α around the *x*-axis (old *r*-axis) and a back transformation into *rst*-coordinates

$$\mathsf{R}(\mathsf{r}, \alpha) = \mathsf{M}^T \mathsf{R}_x(\alpha) \mathsf{M} \tag{12}$$

General Rotations - Rotation About Arbitrary Axis

- To obtain a second direction we can use

$$\tilde{s} = \begin{cases} (0, -r_z, r_y), \text{ if } |r_x| < |r_y| \text{ and } |r_x| < |r_z| \\ (-r_z, 0, r_x), \text{ if } |r_y| < |r_x| \text{ and } |r_y| < |r_z| \\ (-r_y, r_x, 0), \text{ if } |r_z| < |r_x| \text{ and } |r_z| < |r_y| \end{cases} \tag{13}$$

and normalize this direction

$$s = \frac{\tilde{s}}{\|\tilde{s}\|_2} \tag{14}$$

- The last direction can be found using the cross product

$$t = r \times s \tag{15}$$

General Rotations - Rotation About Arbitrary Axis

Euler's rotation theorem
Any displacement of a rigid body such that a point on the rigid body remains fixed, is equivalent to a single rotation about an axis that runs through the fixed point.

- The theorem does not state, which axis, but merely that such an axis exists
- Apart from the initial fixed points all other points on the axis remain fixed

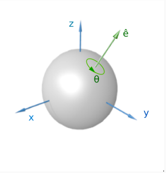


Figure 4: Each rotation can be expressed by a single angle and direction

Quaternions

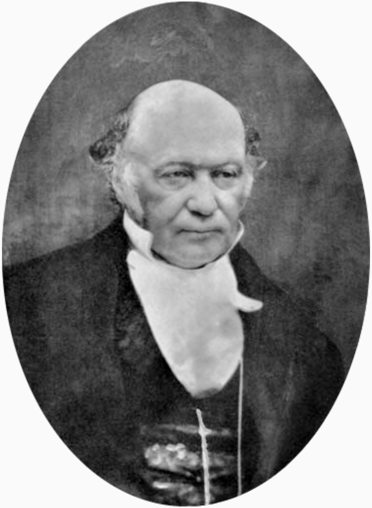


Figure 5: Sir William Rowan Hamilton (1805-1865)

Quaternions

Figure 6: "William Rowan Hamilton Plaque Plaque on Broome Bridge on the Royal Canal commemorating William Rowan Hamilton's discovery. The plaque reads: Here as he walked by on the 16th of October 1843 Sir William Rowan Hamilton in a flash of genius discovered the fundamental formula for quaternion multiplication $i^2 = j^2 = k^2 = ijk = -1$ & cut it on a stone of this bridge." by JP licensed under CC BY-SA 2.0

Definition
A quaternion $\hat{\mathrm{q}} \in \mathbb{H}$ can be defined in the following equivalent ways

$$\hat{\mathrm{q}} = (q_s, \mathrm{q}_v) = q_s + iq_x + jq_y + kq_z, \tag{16}$$

where $\mathrm{q}_v = iq_x + jq_y + kq_z$ is the imaginary part of the quaternion, q_s is its real part and the imaginary units i, j, k satisfy

$$i^2 = j^2 = k^2 = -1 \tag{17}$$

$$jk = -kj = i \tag{18}$$

$$ki = -ik = j \tag{19}$$

$$ij = -ji = k \tag{20}$$

Quaternions - Scalar and Vector Part

Definition
Let $(q_s, \mathrm{q}_v) = q_s + iq_x + jq_y + kq_z$ be any quaternion then we refer to q_s as its **scalar part** and to q_v or $iq_x + jq_y + kq_z$ as its **vector part**.

- A quaternion is called real if its vector part is zero
- A quaternion is called pure imaginary if its scalar part is zero

Quaternions - Vector Space

Definition
The set of quaternions $\mathbb{H}$ forms a real vector space with component wise addition

$$(q_s + iq_x + jq_y + kq_z) + (p_s + ip_x + jp_y + kp_z) = (q_s + p_s) + i(q_x + p_x) + j(q_y + p_y) + k(q_z + p_z) \quad (21)$$

and scalar multiplication

$$\lambda(q_s + iq_x + jq_y + kq_z) = \lambda q_s + i\lambda q_x + j\lambda q_y + k\lambda q_z, \quad (22)$$

with $\lambda \in \mathbb{R}$

Definition
The Hamilton product $\hat{\mathrm{q}}\hat{\mathrm{p}}$ of two quaternions is defined by

$$
\begin{aligned}
\hat{\mathrm{q}}\hat{\mathrm{p}} = &+ (q_s p_s - q_x p_x - q_y p_y - q_z p_z) && (23)\\
&+ i(q_y p_z - q_z p_y + p_s q_x + q_s p_x) && (24)\\
&+ j(q_z p_x - q_x p_z + p_s q_y + q_s p_y) && (25)\\
&+ k(q_x p_y - q_y p_x + p_s q_z + q_s p_z) && (26)\\
&= (q_s p_s - \mathbf{q_v} \cdot \mathbf{p_v}, \mathbf{q_v} \times \mathbf{p_v} + p_s \mathbf{q_v} + q_s \mathbf{p_v}), && (27)
\end{aligned}
$$

where $\cdot$ is the dot product and $\times$ is the cross product.

The corresponding identity element is

$$\hat{\mathrm{i}} = (1, 0). \quad (28)$$

Remark
The Hamilton product is not commutative.

Quaternions - Inverse

Definition
The conjugate of a quaternion $\hat{\mathrm{q}}$ is given by

$$\hat{\mathrm{q}}^* = (q_s, -\mathrm{q}_v). \tag{29}$$

The norm of a quaternion $\hat{\mathrm{q}}$ is given by

$$n(\hat{\mathrm{q}}) = \sqrt{\hat{\mathrm{q}}^*\hat{\mathrm{q}}} = \sqrt{\hat{\mathrm{q}}\hat{\mathrm{q}}^*} \tag{30}$$

The inverse of a quaternion $\hat{\mathrm{q}}$ is given by

$$\hat{\mathrm{q}}^{-1} = \frac{1}{n(\hat{\mathrm{q}})^2}\hat{\mathrm{q}}^* \tag{31}$$

Quaternions - Rules

Using the definitions above it is quite simple to derive

- $(\hat{q}^*)^* = \hat{q}$
- $(\hat{q} + \hat{p})^* = \hat{q}^* + \hat{p}^*$
- $(\hat{q}\hat{p})^* = \hat{p}^*\hat{q}^*$
- $n(\hat{q}^*) = n(\hat{q})$
- $n(\hat{q}\hat{p}) = n(\hat{q})n(\hat{p})$
- $\hat{p}(\alpha\hat{q} + \beta\hat{r}) = \alpha\hat{p}\hat{q} + \beta\hat{p}\hat{r}$
- $(\alpha\hat{q} + \beta\hat{r})\hat{p} = \alpha\hat{q}\hat{p} + \beta\hat{r}\hat{p}$
- $\hat{p}(\hat{q}\hat{r}) = (\hat{p}\hat{q})\hat{r}$

Definition
A unit quaternion $\hat{\mathrm{q}} \in \mathbb{H}$ satisfies $n(\hat{\mathrm{q}}) = 1$.

- For each unit quaternion $\hat{\mathrm{q}} = (q_s, \mathrm{q}_v)$ there exists a unit vector $\mathrm{u}_q \in \mathbb{R}^3$ and angle $\varphi \in \mathbb{R}$ such that

$$\hat{\mathrm{q}} = (\cos\varphi, \sin\varphi \mathrm{u}_q) \tag{32}$$

- Vice versa each quaternion of the form

$$(\cos\varphi, \sin\varphi \mathrm{u}_q), \tag{33}$$

with $\varphi \in \mathbb{R}$ and unit vector $\mathrm{u}_q \in \mathbb{R}^3$ is a unit quaternion

Quaternions - Rotations

- Let r, $\|r\|_2 = 1$ be an arbitrary rotation axis and φ a rotation angle
- These can be used to construct the unit quaternion

$$\hat{q} = (\cos\frac{\varphi}{2}, \sin\frac{\varphi}{2} r) \tag{34}$$

- For this unit quaternion

$$\varrho_{\hat{q}} : \mathbb{H} \to \mathbb{H} \tag{35}$$

$$\hat{x} \mapsto \hat{q}\hat{x}\hat{q}^* \tag{36}$$

  is a rotation on the imaginary part of $\mathbb{H}$ around r by the angle φ
- We can rotate $x \in \mathbb{R}^3$ by φ around $r \in \mathbb{R}^3$ by concatenation of the various mappings

$$x \mapsto (0, x) = \hat{x} \mapsto \hat{q}\hat{x}\hat{q}^* = (0, x') \mapsto x' \tag{37}$$

Quaternions - Rotations

- Given two unit quaternions $\hat{q}_1$ and $\hat{q}_2$ representing two rotations we can concatenate these rotations by

$$\hat{q}_2(\hat{q}_1\hat{x}\hat{q}_1^*)\hat{q}_2^* = (\hat{q}_2\hat{q}_1)\hat{x}(\hat{q}_2\hat{q}_1)^* \tag{38}$$

- I.e. the concatenation of rotations is equivalent to the multiplication of the corresponding unit quaternions
- It directly follows that the inversion of a rotation is equivalent to the inversion of the corresponding quaternion $\hat{q}^{-1} = \hat{q}^*$
- Note that $\hat{q}$ and $-\hat{q}$ define the same rotation

Quaternions - Rotations from One Vector to Another

Question
A common operation is, where we want to rotate one unit vector s into another t. What steps do you need to perform to implement such a rotation with quaternions ?

Quaternions - Spherical Linear Interpolation

- Another problem often encountered is the interpolation between rotations, e.g. given two orientations of a camera finding a smooth transition in between
- One can not simply interpolate the rotation matrices R_1 and R_2, as in general

$$\alpha \mathsf{R}_1 + (1-\alpha)\mathsf{R}_2 \notin SO(3) \tag{39}$$

- However we can define an easy interpolation method for unit quaternions, which achieves this task
- Let $\hat{\mathsf{q}}_1$ and $\hat{\mathsf{q}}_2$ be two unit quaternions and $t \in [0,1]$ an interpolation parameter then

$$\hat{\mathsf{s}}(\hat{\mathsf{q}}_1, \hat{\mathsf{q}}_2, t) = (\hat{\mathsf{q}}_1 \hat{\mathsf{q}}_2^{-1})^t \hat{\mathsf{q}}_2 \tag{40}$$

is the interpolated unit quaternion, where
$\hat{\mathsf{q}}^t = (\cos\varphi, \sin\varphi \mathsf{u}_q)^t = (\cos(\varphi t), \sin(\varphi t)\mathsf{u}_q)$

Quaternions - Spherical Linear Interpolation

- We have

$$\hat{s}(\hat{q}_1, \hat{q}_2, 1) = \hat{q}_1 \tag{41}$$

$$\hat{s}(\hat{q}_1, \hat{q}_2, 0) = \hat{q}_2 \tag{42}$$

- For fixed $\hat{q}_1$ and $\hat{q}_2$ this interpolation constitutes the shortest path (geodesies) from $\hat{q}_1$ and $\hat{q}_2$ on the four dimensional unit sphere
- The computed unit quaternion $\hat{s}(\hat{q}_1, \hat{q}_2, t)$ rotates with a constant speed around a fixed axis with the parameter t
- The method can be extended to find a smooth spline through a series of unit quaternions
- Achieving something similar with Euler angles is quite involved

Quaternions - Spherical Linear Interpolation

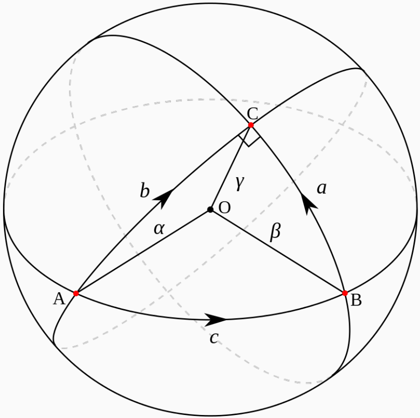


Figure 7: Geodesies on the $\mathbb{S}^2$