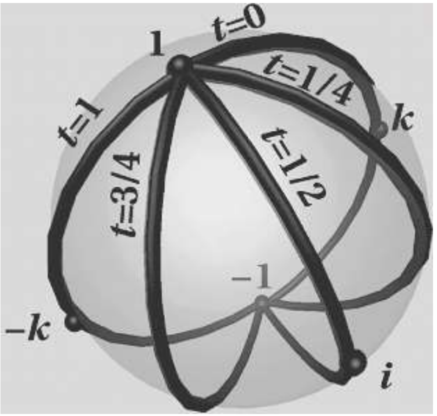
# 3D rotations and quaternions

darko zikic

04 may 2004

supervisor: wolfgang wein
seminar: methods and tools in medical imaging
chair for computer aided medical procedures [ prof. navab ]
technical university of munich

# next 45 minutes: rotations with quaternions




- Why ?
  - what it‘s all about?
  - why bother using quaternions at all?
- How ?
  - provided we want to use quaternions:
    - what are they?
    - and how can we use them?

# agenda

- 3D transformations overview
  - homogeneous transformations
  - VQS transformations
  - comparison
- quaternions
- rotations with quaternions
- summary

# 3D transformations

# homogeneous transformations

- a general transformation matrix (4x4):

$$\begin{bmatrix} A & t \\ v^T & u \end{bmatrix}$$

- A: affine matrix (3x3). includes the rotation matrix R
- R: the rotation matrix(3x3)
- t: the translation vector (3x1)
- v: enables projective transformations (3x1)

# hom. transformations: rotation (euler angles)

- rotation about the X-axis:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha) & sin(\alpha) \\ 0 & -sin(\alpha) & cos(\alpha) \end{bmatrix}$$

- rotation about the Y-axis:

$$B = \begin{bmatrix} cos(\beta) & 0 & sin(\beta) \\ 0 & 1 & 0 \\ -sin(\beta) & 0 & cos(\beta) \end{bmatrix}$$

- rotation about the Z-axis:

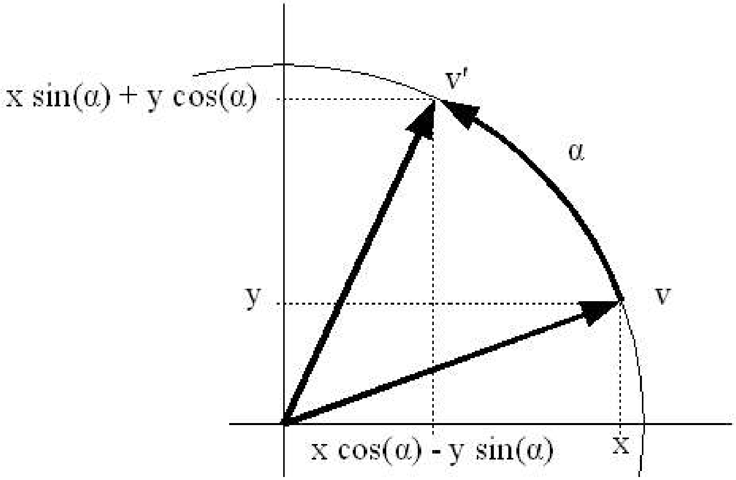
$$C = \begin{bmatrix} cos(\gamma) & sin(\gamma) & 0 \\ -sin(\gamma) & cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Rotation in 3D space: R = CBA

$$R = \begin{bmatrix} \cos(b)\cos(c), & -\cos(a)\sin(c) - \sin(a)\sin(b)\cos(c), & \sin(a)\sin(c) - \cos(a)\sin(b)\cos(c) \\ \cos(b)\sin(c), & \cos(a)\cos(c) - \sin(a)\sin(b)\sin(c), & -\sin(a)\cos(c) - \cos(a)\sin(b)\sin(c) \\ \sin(b), & \sin(a)\cos(b), & \cos(a)\cos(b) \end{bmatrix}$$

# 3D rotation example

- rotation about the z axis




$$\begin{bmatrix} cos(\alpha) & -sin(\alpha) & 0 \\ sin(\alpha) & cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} cos(\alpha) \cdot x - sin(\alpha) \cdot y \\ sin(\alpha) \cdot x + cos(\alpha) \cdot y \\ z \end{bmatrix}$$

# 3D transformations: VQS*

*[ vector | quaternion | scalar ]

# an interlude: quaternions in 60 seconds

- quaternions intuitively: 3D vector extended by a real number

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} \longrightarrow \begin{bmatrix} r \\ a \\ b \\ c \end{bmatrix}$$

3D vector quaternion

- multiplication and addition defined
- quaternions rotate vectors by multiplication
  - r describes the rotation angle
  - a,b,c are the rotation axis

# VQS* transformations

*[ vector | quaternion | scalar ]

- VQS is a triplet (v,q,s): vector, quaternion, scalar
- transformation:

$$p' = \underbrace{s}_{\text{scaling}} \cdot \underbrace{(q \cdot p \cdot q^{-1})}_{\text{rotation}} + \underbrace{v}_{\text{translation}}$$

# homogenous vs. VQS*

*comparison of the two methods as transformation implementations

| homogenous transformations | VQS |
|---|---|
| + non-uniform scaling, shearing and projective transformations possible | + intuitive / simple<br>+ small size<br>+ non-ambigous<br>+ no gimbal lock (loss of DOF)<br>+ easier rotation interpolation |
| – not intuitive / user-unfriendly<br>– unnecessarily large (results in numerical problems)<br>– ambigous<br>– gimbal lock (loss of DOF)<br>– rotation interpolation difficult | – only rotation, translation and scaling possible |

quaternions are the “weapon of choice“ for rigid body transformations

# possible combination of transformation methods in VR / AR systems

**modelling**
homogeneous transformation

**transforming**
VQS transformation

**viewing**
homogeneous transformation

# quaternions

# quaternions: intuition

- a 3D vector extended by a real number to a quadruple

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} \longrightarrow \begin{bmatrix} r \\ a \\ b \\ c \end{bmatrix}$$

3D vector — quaternion

- multiplication and addition defined
- something like 4D complex numbers

# quaternions: definition

- quaternion Q: $Q = 1 \cdot q_1 + i \cdot q_2 + j \cdot q_3 + k \cdot q_4$
- i,j,k: symbolic characters with: $i^2 = -1,\ j^2 = -1,\ k^2 = -1$
- axiomatic properties:

$$ir = ri,\ jr = rj,\ kr = rk$$

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

- defined addition and multiplication

# quaternions: notation

$$
\begin{aligned}
Q &= 1 \cdot q_1 + i \cdot q_2 + j \cdot q_3 + k \cdot q_4 \\
&= [q_1, [q_2, q_3, q_4]] \\
&= [q_r, q_v] \\
&= [q_1, q_2, q_3, q_4]
\end{aligned}
$$

- real quaternion: Q with $q_v = 0$
- vector quaternion: Q with $q_r = 0$

# quaternion operations: addition

- component-wise addition:

$$\begin{aligned} p + q &= (p_1, p_2, p_3, p_4) + (q_1, q_2, q_3, q_4) \\ &= (p_1 + q_1, p_2 + q_2, p_3 + q_3, p_4 + q_4) \end{aligned}$$

- properties:
  - associativity
  - commutativity

# quaternion operations: multiplication

- multiplication-formula:

$$p \cdot q = [p_r, p_v] \cdot [q_r, q_v]$$

$$= (p_1 + ip_2 + jp_3 + kp_4) \cdot (q_1 + iq_2 + jq_3 + kq_4)$$

$$= (p_1q_1 + p_2q_2 + p_3q_3 - p_4q_4) + i(p_1q_2 + p_2q_1 + p_3q_4 - p_4q_3)$$
$$+ j(p_1q_3 + p_3q_1 + p_4q_2 - p_2q_4) + k(p_1q_4 + p_4q_1 + p_2q_3 - p_3q_2)$$

$$= [p_rq_r - p_vq_v, p_rq_v + q_rp_v + p_v \times q_v]$$

- properties:
  - associativity
  - distributivity over addition
  - no commutativity

# quaternion operations: multiplication II

- multiplication of vector quaternions:

  $$p \cdot q = [p_r, p_v] \cdot [q_r, q_v] = [p_r q_r - p_v q_v, p_r q_v + q_r p_v + p_v \times q_v]$$

  with $p_r = 0$ and $q_r = 0$ simplifies to:

  $$p \cdot q = [-p_v q_v, p_v \times q_v]$$

- special cases:
  - p and q parallel: $p \cdot q = [-p_v q_v, 0]$
  - p and q perpendicular: $p \cdot q = [0, p_v \times q_v]$

# other quaternion properties:

- for a quaternion $Q = [q_r, q_v]$

conjugate: $\bar{Q} = [q_r, -q_v]$

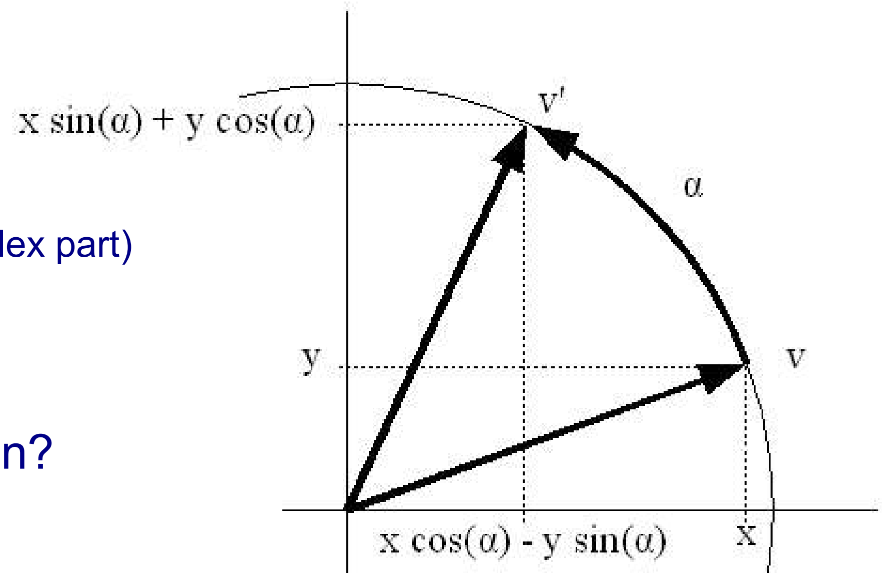
inverse: $Q^{-1} = \frac{\bar{Q}}{|Q|^2}$

unit quaternions: $|Q| = 1$

inverse for unit quaternions: $Q^{-1} = \bar{Q}$

# quaternions & rotations

# rotation intuition




- rotation in 2D
  (can be expressed by complex numbers
  cos(α) being the real part, sin(α) the complex part)

- What do we need for a 3D rotation?
  - angle (sin, cos)
  - rotation axis

- How do we get this information into quaternions?
  - for |Q|=1 there is a Φ such that:

$$Q = [q_r, q_v] = [cos(\phi), sin(\phi)q'_v] \text{ with } q'_v = q_v/|q_v|$$

# unit quaternions rotate vectors

- rotation of a perpendicular vector $(Q \perp V)$
  - $QV$ is a rotation about $q_v$ by $\phi = cos^{-1}(q_r)$
  - $QVQ^{-1}$ is a rotation about $q_v$ by $2\phi = 2cos^{-1}(q_r)$
- general rotations in 3D-space
  - $QVQ^{-1}$ is a rotation about $q_v$ by $2\phi = 2cos^{-1}(q_r)$
- rotation composition
  - succesive rotation by Q and then by P: $PQVQ^{-1}P^{-1}$

# summary

# once again...

+ intuitive / simple
+ small size
+ non-ambigous
+ no gimbal lock (loss of DOF)
+ easier rotation interpolation

# lessons learned

- what are quaternions?
- how to rotate with quaternions?
- why to rotate with quaternions?

# readings


- Allen, Bishop and Welch, SIGGRAPH 2001, “Tracking: Beyond 15 Minutes of Thought“
- Albrecht Beutelspacher, Vieweg, “Lineare Algebra“
- Dam, Koch and Lillholm, University of Copenhagen,
  “Quaternions, Interpolation and Animation“, Tecnical Report
- Hart, Francis and Kauffman, “Visualizing Quaternion Rotation“
- Hartley, Zisserman, “Multiple View Geometry in Computer Vision“
- Nassir Navab, TUM, “Computer Vision“ (Lecture Notes)
- Marc Pollefeys, UNC, “Multiple View Geometry“ (Lecture Notes)
- Holloway, Robinett, “The Visual Display Transformation for Virtual Reality“
- Ken Shoemake, “Animating Rotations with Quaternion Curves“
- Leandra Vicci, UNC, “Quaternions and Rotations in 3-Space“

the end