Quaternionen

Mario Konrad, Mario.Konrad@gmx.net

7. August 2007

Inhaltsverzeichnis

1 Was sind Quaternionen? 2
- 1.1 Allgemein . . . 2
- 1.2 Das Einheitsquaternion . . . 2
- 1.3 Alternative Darstellungsformen . . . 2

2 Rechnen mit Quaternionen 2
- 2.1 Addition . . . 3
- 2.2 Subtraktion . . . 3
- 2.3 Multiplikation . . . 3
- 2.4 Division . . . 4

3 Rotationen im $\mathbb{R}^3$ mit Quaternionen 4
- 3.1 Definition . . . 4
- 3.2 Verkettete Rotation . . . 4
- 3.3 Source Code in C . . . 4

1 Was sind Quaternionen?

1.1 Allgemein

$$q = w + ix + jy + kz \tag{1}$$
$$q' = w - ix - jy - kz \tag{2}$$

mit

$$i^2 = -1, \quad j^2 = -1, \quad k^2 = -1$$

Bei Multiplikationen untereinander verhalten sich diese ähnlich wie das Vektorprodukt:

$$i \cdot j = -j \cdot i = k$$
$$j \cdot k = -k \cdot j = i$$
$$k \cdot i = -i \cdot k = j$$

Der Betrag:

$$\|q\| = \sqrt{q \cdot q'} = \sqrt{w^2 + x^2 + y^2 + z^2}$$

Der Kehrwert:

$$q^{-1} = \frac{q'}{q \cdot q'}$$

mit

$$qq' = w^2 + x^2 + y^2 + z^2 = \|q\|^2$$

1.2 Das Einheitsquaternion

$$\|q\| = 1 \qquad \Longrightarrow \qquad q^{-1} = q' \tag{3}$$

1.3 Alternative Darstellungsformen

Als Vektor:

$$q = (w, x, y, z) \tag{4}$$

Als Skalar und Vektor:

$$s = w \qquad \text{und} \qquad \vec{v} = (x, y, z)$$

dann ist

$$q = (s, v) \tag{5}$$

2 Rechnen mit Quaternionen

Für alle folgenden Kapitel soll gelten

$$q_1 = (w_1, x_1, y_1, z_1)$$
$$q_2 = (w_2, x_2, y_2, z_2)$$

2.1 Addition

$$a = q_1 + q_2 = (w_1 + w_2, x_1 + x_2, y_1 + y_2, z_1 + z_2) \tag{6}$$

Die Addition ist assoziativ:

$$q_1 + (q_2 + q_3) = (q_1 + q_2) + q_3$$

Die Addition ist kommutativ:

$$q_1 + q_2 = q_2 + q_1$$

Es existiert ein neutrales Element bezüglich der Addition:

$$q_1 + e = q_1 \qquad \text{mit} \qquad e = (0, 0, 0, 0) \quad \forall q_1$$

Es existiert ein inverses Element bezüglich der Addition:

$$q_1 + (-q_1) = 0 \qquad \forall q_1$$

2.2 Subtraktion

$$a = q_1 - q_2 = (w_1 - w_2, x_1 - x_2, y_1 - y_2, z_1 - z_2) \tag{7}$$

Die Subtraktion ist assoziativ:

$$q_1 - (q_2 + q_3) = (q_1 - q_2) - q_3 = q_1 - q_2 - q_3$$

Die Subtraktion ist **nicht** kommutativ:

$$q_1 - q_2 \neq q_2 - q_1$$

Es existiert ein neutrales Element bezüglich der Subtraktion:

$$q_1 - e = q_1 \qquad \text{mit} \qquad e = (0, 0, 0, 0) \quad \forall q_1$$

2.3 Multiplikation

Die Multiplikation ist assoziativ:

$$(q_1 \cdot q_2) \cdot q_3 = q_1 \cdot (q_2 \cdot q_3)$$

Die Multiplikation ist **nicht** kommutativ:

$$q_1 \cdot q_2 \neq q_2 \cdot q_1$$

Die Multiplikation ist distributiv:

$$q_1 \cdot (q_2 + q_3) = q_1 \cdot q_2 + q_1 \cdot q_3$$

Es existiert ein neutrales Element bezüglich der Multiplikation:

$$q_1 \cdot e = q_1 \qquad \text{mit} \qquad e = (1, 0, 0, 0) \quad \forall q_1$$

2.4 Division

$$\frac{q_1}{q_2} = q_1 \cdot q_2^{-1} = q_1 \cdot \frac{q_2}{q_2 \cdot q_2'} = \frac{1}{q_2 \cdot q_2'} \cdot q_1 \cdot q_2 \tag{8}$$

3 Rotationen im $\mathbb{R}^3$ mit Quaternionen

3.1 Definition

Quaternionen eignen sich hervorragend zur Rotation von Vektoren im dreidimensionalen Raum. Anders als bei den eulerschen Winkel kann hier der *gimbal-lock* nicht entstehen. Dafür ist ein wenig mehr rechenaufwand erforderlich. Ein anderer Vorteil ist, dass um eine beliebige Drehachse rotiert werden kann.

Nehmen wir ϕ als Drehwinkel um die Drehachse $\vec{u} = (u_1, u_2, u_3)$, mit $\vec{u}$ als Einheitsvektor, an. Wir erhalten dann das Quaternion $q = (w, x, y, z)$ wie folgt:

$$\begin{aligned} w &= \cos\left(\frac{\phi}{2}\right) \\ x &= u_1 \cdot \sin\left(\frac{\phi}{2}\right) \\ y &= u_2 \cdot \sin\left(\frac{\phi}{2}\right) \\ z &= u_3 \cdot \sin\left(\frac{\phi}{2}\right) \end{aligned}$$

oder in der Schreibweise als Linearkombination:

$$s = \cos\left(\frac{\phi}{2}\right) \qquad \vec{v} = \sin\left(\frac{\phi}{2}\right) \cdot \vec{u}$$

Nun wollen wir den Vektor $\vec{a} = (a_1, a_2, a_3)$ um die definierte Drehachse $\vec{u}$ drehen. Dazu müssen wir den Vektor $\vec{a}$ erweitern: $\vec{p} = (0, a_1, a_2, a_3)$

$$p_{rot} = q \cdot p \cdot q' \tag{9}$$

3.2 Verkettete Rotation

Sollen zwei Drehungen um jeweils eine Achse durchgeführt werden, so kann dies wie folgt berechnet werden:

$$p_{rot} = r \cdot q \cdot p \cdot q' \cdot r' \tag{10}$$

mit q und r als Quaternionen.

3.3 Source Code in C

Die Multiplikation zweier Quaternionen:

```
void qmul(float * s, float v[],   // q  = (s,v)
          float s1,  float v1[],  // q1 = (s1,v1)
```

```
          float s2,  float v2[])  // q2 = (s2,v2)
{
   *s   = s1*s2 - v1[0]*v2[0] - v1[1]*v2[1] - v1[2]*v2[2];
   v[0] = s1*v2[0] + s2*v1[0] + v1[1]*v2[2] - v1[2]*v2[1];
   v[1] = s1*v2[1] + s2*v1[1] - v1[0]*v2[2] + v1[2]*v2[0];
   v[2] = s1*v2[2] + s2*v1[2] + v1[0]*v2[1] - v1[1]*v2[0];
}
```

Die Rotationsfunktion:

```
void rot(float p[], float u[], float angle)
{
   float s = 0.0; float v[3]; // q
   float vi[3]; // q'
   float f;
   float ts, tv[3];

   s = cos(angle*M_PI/180.0/2.0);
   f = sin(angle*M_PI/180.0/2.0);
   v[0] = u[0] * f; v[1] = u[1] * f; v[2] = u[2] * f;
   vi[0] = -v[0]; vi[1] = -v[1]; vi[2] = -v[2];
   qmul(&ts, tv, s, v, 0.0, p);
   qmul(&f, p, ts, tv, s, vi);
   f = 1.0 / (s*s + v[0]*v[0] + v[1]*v[1] +v[2]*v[2]);
   p[0] *= f; p[1] *= f; p[2] *= f;
}
```

Die Anwendung:

```
float p[] = { 1.0, 0.0, 0.0 };
float u[] = { 0.0, 1.0, 0.0 };
rot(p, u, 90.0);  // p 90 Grad um u drehen
```