Quaternionen - mathematischer Hintergrund und ihre Interpretation als Rotationen

Markus Lust

Februar 2001

Computergrafik-Seminar
Universität Koblenz-Landau

21. Februar 2001

Zusammenfassung

Dieser Text entstand im Frühjahr 2001 im Rahmen des Computergrafik-Seminars bei Prof. Giesen, angeboten im Wintersemester 2000/2001 an der Universität Koblenz-Landau, Abteilung Koblenz. Die folgenden Seiten sollen eine Einführung geben in die Welt der Quaternionen und ihr historischer und mathematischer Hintergrund soll aufgedeckt werden. Ziel ist es, zu erläutern, wie man Quaternionen zur Repräsentation von Rotationen in der Computergrafik einsetzt und ihre Vor- und Nachteile gegenüber herkömmlichen Methoden sollen diskutiert werden.

Inhaltsverzeichnis

1 Historischer Hintergrund

Lange Zeit befand sich William Hamilton auf der Suche nach einer Erweiterung der komplexen Zahlen ins 3-dimensionale. Die Aufgabe klang einfach. Es gab eine Regel zur Multiplikation von Tupeln (komplexe Zahlen aus einem Realteil und einem Imaginärteil) und gesucht war eine Regel zur Multiplikation von Tripeln. Jeden Morgen im Oktober 1948, wenn Hamilton an den Frühstückstisch kam fragte ihn sein jüngster Sohn: "Papa, kannst Du Tripel multiplizieren?- worauf dieser jedesmal resigniert antwortete: "Nein, ich kann sie nur addieren und subtrahieren." Eines Tages bei einem Spaziergang mit seiner Frau am Royal Canal in Dublin kam ihm die entscheidende Idee. Er war so fasziniert, dass er sie mit einem Messer in die Holzbalken der Brougham Bridge ritzte. Aber seine Lösung war sogar für ihn selbst höchst merkwürdig, denn sie war 4-dimensional. In einer ersten Idee setzte er die vierte Komponente gleich Null und nannte das Ergebnis ein reines Quaternion. Genutzt hat im 19. Jahrhundert niemand seine Quaternionen. Erst später griff Prof. Gibbs aus Yale seine Idee auf, experimentierte ein wenig herum und "erfand" sozusagen (auf reinen Quaternionen aufbauend) das Vektor- und das Skalarprodukt.

2 Quaternionen

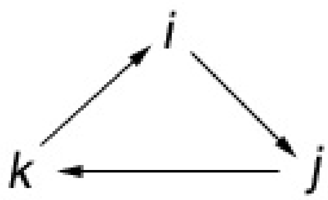
Hamilton war auf der Suche nach einer Erweiterung der komplexen Zahlen ins 3-dimensionale. Komplexe Zahlen lassen sich entweder als $a+b\cdot i$ oder als (a,b) (also als 2D-Vektor) schreiben. Bei der Multiplikation dieser Zahlen muss beachtet werden, dass $i^2=-1$ gilt.
Hamilton überlegte nun, wie zwei Zahlen der Art $a+b\cdot i+c\cdot j$ bzw. (a,b,c) miteinander multiplizieren sollte. Die Addition und Subtraktion führte er komponentenweise aus, wie wir das von heutigen Vektoren kennen. Bei der Multiplikation treten nun die Terme i^2, j^2 und $i\cdot j$ auf. Es gilt analog zu den komplexen Zahlen $i^2=j^2=-1$ und $j\cdot i=-i\cdot j$ muss gelten, damit die euklidische Länge eines Vektors multiplikativ bleibt, damit also weiterhin $l(v\cdot w)=l(v)\cdot l(w)$ gilt.
Statt nun $i\cdot j=0$ zu setzen, was diese Gleichung auch erfüllt hätte, setzte Hamilton $i\cdot j=k$, er führte also eine vierte Dimension ein.
Damit ist ein Quaternion wie folgt definiert:

Definition 1
Ein Quaternion q ist ein 4-Tupel (w,x,y,z), wobei gilt: $q:=w+x\cdot i+y\cdot j+z\cdot k$

Nach dieser Definition kann man also ein Quaternion auffassen als eine komplexe Zahl mit drei verschiedenen imaginären Größen i, j und k oder als einen 4-dimensionalen Vektor mit den Komponenten w, x, y und z. Damit ließe sich ein 3D-Vektor darstellen als ein Quaternion mit dem Realteil $w=0$. Auch üblich ist die Schreibweise $q=[w,v]$ mit der reellen Größe w und dem 3D-Vektor $\mathbf{v}$. Das kann man rechtfertigen, indem man i, j und k als 3D-Normalen-Einheitsvektoren auffasst, die, wenn man sie mit x, y bzw. z multipliziert und die Ergebnisse dann addiert, den Vektor (x,y,z) ergeben.

2.1 Die imaginären Größen

Mit folgendem Diagramm lässt sich der mathematische Zusammenhang zwischen den drei imaginären Größen beschreiben:



In Pfeilrichtung gelesen heißt das: $i \cdot j = k$, $j \cdot k = i$ und $k \cdot i = j$. Gegen die Pfeilrichtung erhält man jeweils ein Minuszeichen, also z.B. $j \cdot i = -k$. Damit gilt also unter anderem $i \cdot j \neq j \cdot i$. Das bedeutet, dass R^4 einen sogenannten *Schiefkörper* aufspannt, da alle Körpereigenschaften außer der Kommunitativität bzgl. der Multiplikation gelten. Außerdem gilt wie bei den komplexen Zahlen $i^2 = j^2 = k^2 = -1$.

2.2 Rechenregeln

Addition:
Die Addition ist wie folgt definiert und funktioniert genau wie bei den komplexen Zahlen bzw. wie bei Vektoren, nämlich komponentenweise:

$$\begin{aligned} q_1 + q_2 &= [w_1, v_1] + [w_2, v_2] \\ &= w_1 + x_1 i + y_1 j + z_1 k + w_2 + x_2 i + y_2 j + z_2 k \\ &= (w_1 + w_2) + (x_1 + x_2) j + (y_1 + y_2) j + (z_1 + z_2) \\ &= [w_1 + w_2, v_1 + v_2] \end{aligned}$$

Ein Beispiel:
$[1, (2, 3, 4)] + [2, (4, 6, 8)] = [3, (6, 9, 12)]$

Multiplikation:
Die Multiplikation ist wegen der oben beschriebenen Regeln für die Imaginärteile nicht kommutativ und daher wie folgt definiert:
$q_1 \cdot q_2 = [w_1, v_1] \cdot [w_2, v_2] = [w_1 \cdot w_2 - v_1 \cdot v_2\ ,\ v_1 \times v_2 + w_1 \cdot v_2 + w_2 \cdot v_1]$
Diese Definition ergibt sich aus den Rechenregeln für relle und komplexe Zahlen wenn man q_1 und q_2 als 4-dimensionale komplexe Zahlen betrachtet. Der Beweis hierfür ist zu finden in [DAM 98].
Ein Beispiel:
$[1, (0, 1, 1)] \cdot [1, (1, 0, 0)] =$
$(1 + 0i + j + k) \cdot (1 + i + 0j + 0k) =$
$1 + i + j \cdot i + k \cdot i =$
$1 + i - k + k + j =$
$1 + i + j = [1, (1, 1, 0)]$

Das Konjugierte:
Definition 2

Das Konjugierte eines Quaternions q ist $q^* = [w, -v]$
Auch diese Definition orientiert sich an den komplexen Zahlen.

Der Betrag (die Länge):
Definition 3

Der Betrag (Die Länge) eines Quaternions q ist $\|q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$

Ein *Einheitsquaternion* ist ein Quaternion q mit $\|q\| = 1$. Enheitsquaternionen werden später bei der Rotation eine entscheidende Rolle spielen.

Das Inverse:
Definition 4

Das Inverse eines Quaternions q ist $q^{-1} = \frac{q^*}{\|q\|^2}$.

Für Einheitsquaternionen gilt somit: $q^{-1} = q^*$

2.3 Die Polarform

Satz:
Ein Quaternion q kann geschrieben werden als
$q = \|q\| \cdot (cos(\Theta) + i \cdot sin(\Theta) + j \cdot sin(\Theta) + k \cdot sin(\Theta))$.
Einheitsquaternionen lassen sich schreiben als $q = [cos(\Theta), \mathbf{n} \cdot sin(\Theta)]$, wobei **n** ein Vektor der Länge 1 ist.

Beweis:
Es gilt:
$\|q\| = \|[w, v]\| = w^2 + x^2 + y^2 + z^2 = 1$
$\Rightarrow x^2 + y^2 + z^2 = \|v\|^2 \leq 1$
$\Rightarrow -1 \leq \|v\| \leq 1$
$\Rightarrow \exists \Theta : sin(\Theta) = \|v\|$
$\Rightarrow v = \|v\| \cdot \frac{v}{\|v\|} = sin(\Theta) \cdot \frac{v}{\|v\|} = \mathbf{n} \cdot sin(\Theta)$
Hierbei ist **n** ein Einheitsvektor.
Da gilt:
$w^2 + \|v\|^2 = 1$ folgt:
$w^2 = \|v\|^2 - 1 \Rightarrow w^2 = cos^2(\Theta) \Rightarrow w = cos(\Theta)$
Aus all dem folgt dann, dass ein Einheitsquaternion geschrieben werden kann als $q = [cos(\Theta), \mathbf{n} \cdot sin(\Theta)]$.

Man nennt Θ den *Winkel* und **n** die *Achse* des Quaternions (siehe Abb.1). Diese beiden Begriffe dienen als Grundlage für die Rotation mittels Quaternion-Multiplikation, die im nächsten Abschnitt beschrieben werden soll.

3 Rotation

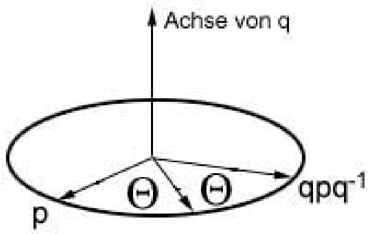


Abbildung 1: Rotation des Vektors r um die Achse n mit dem Winkel Θ

Satz
Sei $p = [0, v]$ Die Quaternion-Darstellung des Vektors v. Die Operation $q \cdot p \cdot q^{-1}$ (oder $q \cdot p \cdot q^*$ falls q ein Einheitsquaternion ist) rotiert den Vektor v um die Achse von q mit dem Doppelten des Winkels von q. Die Dreh-Richtung ergibt sich aus der aus der Physik bekannten Rechte-Hand-Regel.

Der Beweis dieses Satzes bleibt aus, er kann bei [DAM 98] nachgelesen werden. Stattdessen wird anhand eines Beispiels eine Rotation durchgeführt und danach werden ein paar Beobachtungen gemacht, die plausibel machen, dass es sich bei der vorgestellten Operation tatsächlich um eine Rotation handelt. Als Beispiel soll der Vektor $v = (1, 2, 0)$ rechtsherum um die z-Achse um 60 Grad rotiert werden. Das v entsprechende Quaternion p ist $[0, (1, 2, 0)]$ oder $i + 2j$.
Um das gesuchte Rotationsquaternion zu erhalten, muss man die Achse und den Winkel bestimmen. Die Achse ist $(0, 0, -1)$, da es sich um die z-Achse handelt, die Achse die Länge 1 haben muss und es sich um eine Rechts-Drehung handelt. Der Winkel beträgt $30°$, denn die Quaternion-Rotation repräsentiert ja einen doppelten Rotationswinkel. Damit ist das gesuchte Einheitsquaternion q für die Rotation des Vektors v $[cos(30°), (0, 0, -1) \cdot sin(30°)] = [cos(30°), (0, 0, -0.5)]$.
Die Rotation wird durch $q \cdot p \cdot q^*$ ausgeführt.

Es gilt:

$$\begin{aligned}
q \cdot p \cdot q^* &= (cos(30°) - 0,5k) \cdot (i + 2j) \cdot (cos(30°) + 0,5k) \\
&= (i \cdot cos(30°) + 2j \cdot cos(30°) - 0,5ki - kj) \cdot (cos(30°) + 0,5k) \\
&= (i \cdot cos(30°) + 2j \cdot cos(30°) - 0,5j + i) \cdot (cos(30°) + 0,5k) \\
&= (i \cdot cos^2(30°) + 0,5ik \cdot cos(30°) + 2j \cdot cos^2(30°) + jk \cdot cos(30°) - 0,5j \cdot cos(30°) \\
&\quad -0,25jk + i \cdot cos(30°) + 0,5ik) \\
&= (i \cdot cos^2(30°) - 0,5j \cdot cos(30°) + 2j \cdot cos^2(30°) + i \cdot cos(30°) - 0,5j \cdot cos(30°) \\
&\quad -0,25i + i \cdot cos(30°) - 0,5j)
\end{aligned}$$

$= i \cdot (cos^2(30°) + 2cos(30°) - 0,25) + j \cdot (2cos^2(30°) - cos(30°) - 0,5)$
$= i \cdot (0,75 + 1,73 - 0,25) + j \cdot (1,5 - 0,86 - 0,5)$
$= 2,23i + 0,14j$
oder in Vektorschreibweise: $(2.23, 0.14, 0)$

Dass es sich um den richtigen Vektor handelt kann leicht nachgerechnet werden. Folgende Indizien sprechen ebenfalls für eine korrekte Rotation:

- p wurde als Quaternion mit $w = 0$ aufgefasst und am Wert von w hat sich nichts geändert.
- Am Wert der z-Koordinate hat sich nichts geändert, was wir von einer Rotation um die z-Achse auch erwarten.
- Die Länge des Vektors v blieb unverändert: $\|v\| = \sqrt{1+4+0} = \sqrt{5} = \sqrt{2,23^2 + 0,14^2}$

3.1 Konkatenation von Rotationen

Satz
Es seien die Rotationen R_1 und R_2 durch q_1 bzw. q_2 repräsentiert. Dann repräsentiert $q = q_2 \cdot q_1$ die Rotation $R = R_1 R_2$
Beweis:
R wird repräsentiert durch $q_2 \cdot (q_1 \cdot p \cdot q_1^*) \cdot q_2^*$
$= (q_2 \cdot q_1) \cdot p \cdot (q_1^* \cdot q_2^*)$
$= (q_2 \cdot q_1) \cdot p \cdot (q_2 \cdot q_1)^*$
$= q \cdot p \cdot q^*$ mit $q = q_2 \cdot q_1$.

Als Beispiel dazu soll eine Rotation um 180° um die x-Achse gefolgt von einer Rotation um 180° um die y-Achse gefolgt von einer Rotation um 180° um die z-Achse betrachtet werden. i, j und k repräsentieren jeweils die einzeln Rotationen da ja z.B. $[cos(90°), (1,0,0) \cdot sin(90°)] = [0, (1,0,0)] = i$ eine Rotation um 180° um die x-Achse darstellt. Die Gesamtrotation ist dann $i \cdot j \cdot k = -1$. Da die Gesamtrotation eine Null-Rotation darstellt, sollte daran auch an der Änderung der Reihenfolge der einzelnen Rotationen nichts ändern. Es ist aber $j \cdot i \cdot k = 1$, was auf den ersten Blick dazu im Widerspruch steht. Jedoch ist es tatsächlich so, dass sowohl $[1, (0,0,0)]$ als auch $[-1, (0,0,0)]$ eine Null-Rotation repräsentieren.

3.2 Potenz von Quaternionen

Satz
Das Quaternion q^x rotiert einen Vektor x-mal soweit wie das Quaternion q.

Der Satz folgt direkt aus der Beziehung $q^x = \|q\|^x \cdot [cos(x\Theta), \mathbf{n} \cdot sin(x\Theta)]$, der Beweis dafür bleibt aber aus.

3.3 Die Rotationsmatrix

Da die Orientierung eines Objekts im Raum meistens in Form einer Matrix gegeben ist und man Quaternionen vor allem zur Interpolation zwischen zwei Orientierungen einsetzt, muss man sehr oft Quaternionen in Matrizen umwandeln. Wenn das Quaternion $[w, (x, y, z)]$ gegeben ist, dann lässt sich daraus die folgende Rotationsmatrix ableiten:

$$\begin{pmatrix} 1-2(y^2+z^2) & 2(xy-wz) & 2(xz+wy) \\ 2(xy+wz) & 1-2(x^2+z^2) & 2(yz-wx) \\ 2(xz-wy) & 2(yz+wx) & 1-2(x^2+y^2) \end{pmatrix}$$

Auf den ersten Blick scheint es ein Vorteil zu sein, dass diese Matrix keine trigonometrischen Funktionen enthält. Aber für die Rotation muss das Quaternion ja in Polarform gegeben sein, dann sind die w, x, y und z trigonometrische Funktionen.

4 Vergleich zw. Euler'scher Geometrie und Quaternionen

Nun, da die Rotation im Raum mittels Quaternionenmultiplikation vorgestellt wurde, muss man sich fragen, welche Vor- und Nachteile dieses Vorgehen mit sich bringt. Traditionell werden Rotationen in der Computergrafik mit Methoden der Linearen Algebra durchgeführt. In der Euler'schen Geometrie hat man dazu ein ein rechtwinkliges Koordinatensystem mit den 3 Achsen x, y und z. Eine Rotation wird durchgeführt, indem man einen Punkt einzeln um die drei Achsen dreht. Diese drei Drehungen nennt man auch *roll*, *pitch* und *yaw*. Im folgenden werden die Vor- und Nachteile der beiden Verfahren gegenübergestellt, im wesentlichen nach [DAM 98].

Nachteile der Euler'schen Geometrie

- Solange man eine Rotation in einzelnen Drehungen um die jeweiligen Achsen ausdrücken kann, ist das Euler'sche Modell intuitiv. Das ändert sich jedoch, wenn man um eine beliebige Achse drehen möchte, denn es ist nicht intuitiv zu erkennen, wie die drei Winkel lauten, mit denen man einen Punkt um die Achse $(1, 1, 1)$ dreht.

- Die Reihenfolge der einzelnen Rotationen ist nicht beliebig, was sich auch in der Nicht-Kommutativität der Matrizenmultiplikation niederschlägt. Eine Rotation um $90°$ um die x-Achse, $90°$ um die y-Achse und $0°$ um die z-Achse liefert ein unterschiedliches Ergebnis, je nachdem um welche Achse zuerst gedreht wird.

- Es kann zum Gimbal-Lock-Problem kommen. Dabei handelt es sich um ein Konzept aus der Luft- und Raumfahrt-Industrie [DAM 98], wo Gyroskope verwendet werden. Ein Gyroskop besteht aus drei konzentrischen Ringen wie Abb. 2 zeigt.

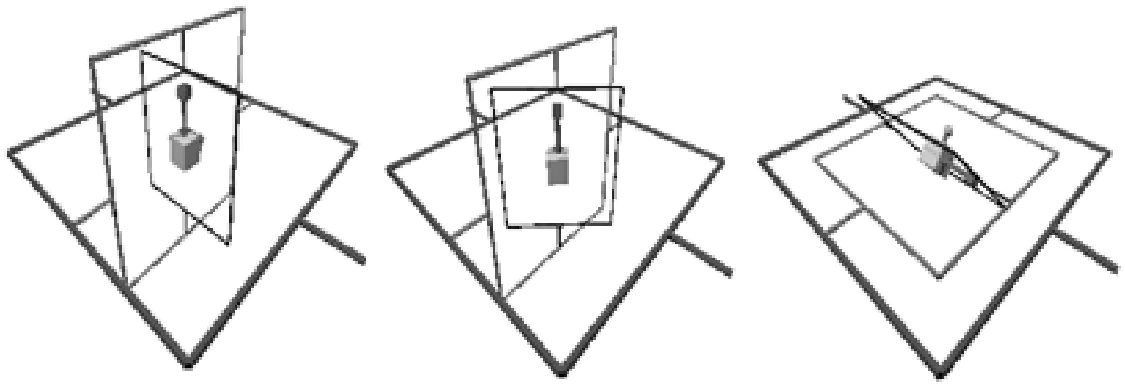


Abbildung 2: Zwei Rotationen, die bei einem Gyroskop zum Gimbal-Lock führen

Das erste Bild in Abb. 2 zeigt den Ausgangszustand des Gyroskops, alle drei Achsen stehen senkrecht aufeinander. Eine Drehung um die z-Achse resultiert im zweiten Bild von Abb.2. Darauf folgt eine Drehung um die y-Achse um 90°. Eine Drehung um die z-Achse ist nun nicht mehr möglich, wie die dritte Grafik zeigt. Die einzelnen Rotationen in der Euler'schen Geometrie sind also nicht voneinander unabhängig.

- Lineare Matrizen-Interpolation kann nicht verwendet werden. Wenn man zwischen zwei Orientierungen eines Vektors interpolieren will, kann in der Euler'schen Geometrie nicht die intuitive Methode der linearen Interpolation zwischen 2 Matrizen verwendet werden. Als Funktion lautet diese Verfahren: $LinMat(M_0, M_1, h) = M_0 \cdot (1 - h) + M_1 \cdot h$. Da die interpolierten Matrizen aber in der Regel nicht orthonormal sind und sogar neben der Rotation auch noch Effekte wie Skalierung und Translation auftreten, ist diese Methode hier unbrauchbar.

- Die Rotationsmatrizen sind mehrdeutig, es lässt sich bei gegebener Matrix keine eindeutige Folge von Euler-Winkel besitmmen, die der Rotation entspricht.

- Ein zusätzlicher Rechenaufwand entsteht, weil die Orthonormalität der Matrizen immer bewahrt werden muss.

- Die in der Matrix enthaltenen Informationen sind redundant, bei homogenen Matrizen sind 6 Nullen enthalten, bei 3x3-Matrizen benutzt man 9 Stellen um 4 Freiheitsgrade (ein Winkel und eine 3D-Achse) zu repäsentieren. Ausserdem muss man 6 Bedingungen ständig überprüfen (jede Zeile muss ein Einheitsvektor sein und die Spalten müssen zueinander orthogonal sein).

Vorteile der Euler'schen Geometrie

- Bei der Euler'schen Geometrie handelt es sich um bekannte Mathematik, die zur Grundausbildung im Mathematikunterricht gehört und daher vorausgesetzt werden kann.

- Auch ein entscheidender Vorteil ist die Tatsache, das die Methoden und Funktionen zum Standard der meisten Grafikpakete gehören wohingegen Quaternionen-Operationen erst implementiert und effizient gemacht werden müssen.

- Die homogene Matrix enthält mehr Informationen, sie kann nicht nur zum Rotieren sondern auch zum Skalieren, Verschieben etc. genutzt werden.

Nachteile von Quaternionen

- Quaternionen gehören nicht zum Umfang der mathematischen Standardausbildung. Auch in den meisten Grafikpaketen sind Quaternionen-Methoden nicht unterstützt.

- Mit Quaternionen kann man lediglich Rotationen darstellen. Das bedeutet, dass am besten Matrizen und Quaternionen verwendet werden sollten. Dabei entstehen zusätzliche Kosten für die Umrechnung zwischen den beiden Konzepten.

Vorteile von Quaternionen

- Quaternionen haben eine offensichtliche geometrische Interpretation. Drehachse und -winkel können direkt abgelesen beziehungsweise kodiert werden (siehe Abb. 3).

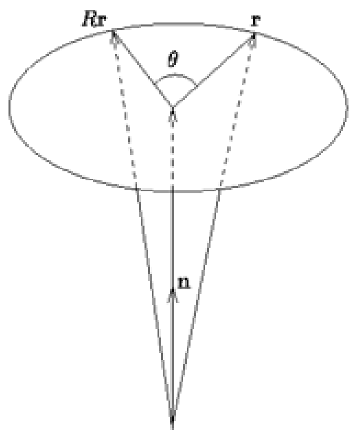


Abbildung 3: Achse und Winkel

- Quaternionen sind vom Koordinatensystem unabhängig und der Benutzer muss sich keine Gedanken über die Konventionen bezüglich der Reihenfolge der Rotationsachsen machen.

- Mit Interpolationsverfahren wie SLERP(Spherical Linear Interpolation)und SQUAD (Spherical and Quadrangle Quaternion Interpolation) lassen sich die “schönsten“ und sanftesten Interpolationen zwischen zwei Orientierungen bestimmen. Diese eignen sich hervorragend für Kamerafahrten. Ausführliche Informationen dazu siehe [DAM 98].

- Die Repräsentation mit Quaternionen ist kompakt. Vier Variablen stehen für genau die 4 Constraints aus Euler's Theorom (Ein Winkel und eine 3D-Achse). Anstatt der 6 Bedingungen die man bei den Euler-Winkeln einhalten muss, braucht man hier nur eine zu bewahren: Die Rotations-Quaternionen sollten Einheitsquaternionen sein.
- Das Gimbal-Lock-Problem tritt nicht auf, weil nicht einzeln um 3 Achsen gedreht wird sondern der Punkt auf der Oberfläche einer im Ursprung liegenden Einheits-Kugel bewegt wird.
- Die Konkatenation von Rotationen ist mit Quaternionen deutlich effizienter: Multipliziert man zwei 3x3-Matrizen so muss man 27 Multiplikationen und 18 Additionen durchführen. Bei der Multiplikation von Quaternionen kommt man mit einem entsprechend effizienten Algorithmus [BOB 98] nur auf 8 Multiplikationen und 4 Divisionen.

Literatur

[DAM 98] Erik B. Dam, Martin Koch, Martin Lillholm: *Quaternions, Interpolation and Animation*, Technical report DIKU-TR-98/5 University of Copenhagen, 07/1998

[BOB 98] Nick Bobick: *Rotating Objects Using Quaternions*,
Artikel aus "Game Developer", Game Developer Media Group 02/1998