

## Praktikum Augmented Reality

### **Abgabetermin:**

Die Lösung zu diesem Übungsblatt ist bis zum 8.06.2004 abzugeben.

Bauen Sie wie gewohnt auf Ihren Lösungen der letzten Aufgabenblätter auf, insbesondere der Funktion `matrixToQuaternion` aus Aufgabe 9c. Die übrigen Quaternionenoperationen (u.a. `quaternionToMatrix`, `quaternionToAxisAngle`, `axisAngleToQuaternion`) können Sie auf der Praktikums homepage herunterladen.

### **Aufgabe 17 (P) Tangible User Interface**

Rechensysteme zwingen ihre Nutzer leider viel zu oft, sich ihnen anzupassen, anstatt daß die Systeme sich den Nutzern anpassen. Eine Idee, diesen bedauernswerten Zustand zu verbessern, sind Benutzerschnittstellen zum Anfassen, sogenannte *Tangible User Interfaces (TUIs)*. Hiroshi Ishii vom MIT Media Lab<sup>1</sup> ist der Pionier dieser Interaktionsmetapher, bei der mittels alltäglicher Objekte mit einem Rechner interagiert wird.

Die Bildverarbeitung des AR Toolkit reicht aus, um einfache TUIs zu bauen, und genau das sollen Sie in dieser Aufgabe machen.

- Basteln Sie einen Drehregler! Dazu bietet es sich an, auf einer Trägerplatte einen Referenzmarker anzubringen, und dagegen verdrehbar einen zweiten Marker. Visualisieren Sie einen nett anzusehenden Drehknopf auf dem zweiten Marker, und evtl. noch eine einfache Kontrollinformation auf dem Referenzmarker.
- Bestimmen Sie mittels Quaternionen und deren Achsen/Winkel Rotationsäquivalent (s. Vorbemerkung Aufgabe 9) den Winkel zwischen diesen beiden Markern, aus der sich die aktuelle Stellung des Drehreglers ergibt. Nutzen Sie die gewonnene Information, um in der Visualisierung irgendeinen Parameter interaktiv zu verändern.
- Basteln Sie ein weiteres TUI Ihrer Wahl! Um die Aufgabe nicht unnötig zu erschweren, soll auch dieses Kontrollelement eine eindimensionale Ausgabe liefern (wie der Drehregler, der einen Wert zwischen 0 und  $2\pi$  ausgibt).

### **Aufgabe 18 (P) Quaternions from Hell**

Nachdem Sie in den letzten Aufgaben Quaternionen als zu Matrizen äquivalente Rotationsdarstellung kennengelernt haben, sollen Sie jetzt etwas wirklich Sinnvolles mit ihnen bewerkstelligen. Wie der Name dieser Aufgabe aber schon andeutet, müssen zuerst noch ein paar kleine Voraussetzungen geschaffen werden.

Eine wunderschöne Anwendung von Quaternionen liegt in der idealen Interpolation zwischen zwei Rotationen. Gegeben seien also zwei Quaternionen  $q_1$  und  $q_2$ , und ein Interpolationsfaktor

---

<sup>1</sup><http://tangible.media.mit.edu/>

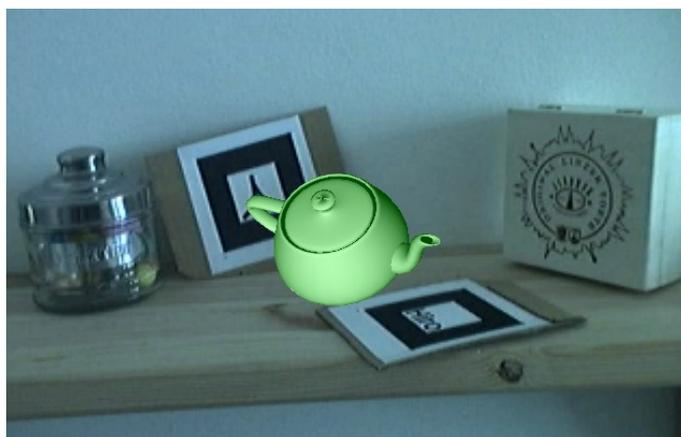
$h \in [0; 1]$ . Ist nun  $h = 0$ , so soll die Ausgabe  $q_1$  sein, ist  $h = 1$ , so soll  $q_2$  ausgegeben werden. Dazwischenliegende Werte von  $h$  führen zu Interpolationen.

Der in dieser Aufgabe zu implementierende Algorithmus heißt *Slerp* (*Spherical Linear Interpolation*), in der Menge der Einheitsquaternionen, die man sich als Kugel mit Radius 1 im vierdimensionalen Raum vorstellen kann, erzeugt er eine Bahn konstanter Winkelgeschwindigkeit auf einem Großkreis dieser Kugel. Details und Beweise finden sich in Dam, Koch und Lillholm, *Quaternions, Interpolation and Animation*<sup>2</sup>.

Der Algorithmus arbeitet wie folgt (NB:  $q_1 \cdot q_2$  bezeichnet das innere Produkt zweier Quaternionen, *nicht* deren Multiplikation):

$$\begin{aligned} \cos \Omega &= q_1 \cdot q_2 \\ \text{Slerp}(q_1, q_2, h) &= \frac{q_1 \sin((1-h)\Omega) + q_2 \sin(h\Omega)}{\sin \Omega} \end{aligned}$$

- Implementieren Sie eine Funktion `quaternionSlerp(double *q1, double *q2, double h)`!
- Lassen Sie ein Objekt Ihrer Wahl (z.B. eine Teekanne) zwischen zwei Markern hin- und herpendeln. Das Objekt soll also zuerst auf dem einen Marker stehen, sich dann mit konstanter (Winkel-)Geschwindigkeit zum anderen Marker hin bewegen, auf diesem auch kurz stehen und sich dann wieder an den Ausgangsort zurückbewegen. Interpolieren Sie dazu die Position linear und die Rotation mittels Slerp.
- Benutzen Sie das in der vorigen Aufgabe entworfene TUI, um die Geschwindigkeit des Objekts zu regulieren.



<sup>2</sup><http://www.diku.dk/forskning/image/teaching/Studentprojects/Quaternion/>