Ludwig-Maximilians-Universität München
LFE Medieninformatik
Prof. Dr. Andreas Butz
Dipl.-Medieninf. Sebastian Boring
Dipl. Inf. Arnd Vitzthum

Wintersemester 2006/2007
Übungsblatt 3
13. November 2006

Übung "Augmented Reality"

Abgabetermin:
Die Lösung zu diesem Übungsblatt ist bis zum 20. November 2006 abzugeben.

Inhalt:
In diesem Blatt wird das Konzept hierarchischer Transformationen im Szenengraphen vorgestellt und zur Kollisionserkennung benutzt. Außerdem werden Quaternionen als weitere Form der Rotationsdarstellung eingeführt. Sie eignen sich besonders zur Animation von Objekten. Als Objekte werden nun VRML-Modelle eingeführt. Zum Abschluss wird der große böse Wolf ein armes Schaf erschrecken.

Aufgabe 14 (H) Der Szenegraph

Der Szenegraph ist eine in der Computergraphik häufig verwendete hierarchische Darstellungsform von 3D-Szenen. Es handelt sich dabei um einen DAG (Directed Acyclic Graph), oft auch um einen Baum. In diesem Graph kann man drei Typen von Knoten unterscheiden. Gruppenknoten dienen der Zusammenfassung mehrerer Kinder zur Hierarchisierung von Objekten, Transformationsknoten beschreiben geometrische Transformationen, also z.B. Rotationen oder Translationen, und Strukturknoten beschreiben geometrische Objekte, z.B. ein Tisch oder Stühle.

Beim Rendern durchläuft man den Graphen nun mit einer Tiefensuche, wobei die Kinder von Gruppenknoten in fester Ordnung abgearbeitet werden.

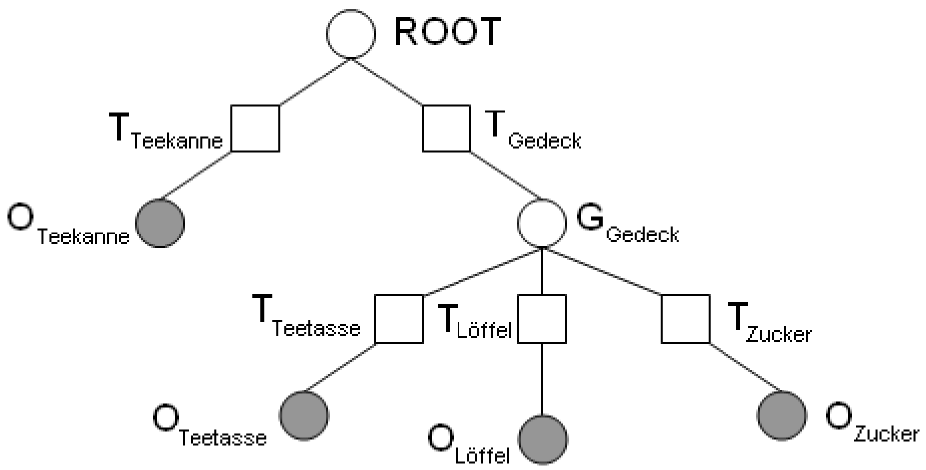

Abbildung 1: Beispielszenegraph: Ein Teegedeck

Geht man in einem Szenengraphen "von unten nach oben", so entspricht dies der Ausführung von *invertierten* Transformationen. Hierdurch ist es z.B. möglich, die Position eines Stuhls relativ zum Tisch durch folgende Berechnung zu bestimmen:

$$\mathbf{T}_{gesucht} = \mathbf{T}_{Tisch}^{-1}$$

- Berechnen Sie analytisch die Linksinverse $\mathbf{H}^{-1,L}$

$$\mathbf{H}^{-1,L} = \begin{pmatrix} \mathbf{R}^* & t^* \\ 0 & 1 \end{pmatrix} \text{ mit } \mathbf{H}^{-1,L} \cdot \begin{pmatrix} \mathbf{R}_x & t_x \\ 0 & 1 \end{pmatrix} = \mathbf{I}$$

  in Abhängigkeit von den Parametern $\mathbf{R}_x$ und t_x einer homogenen Transformationsmatrix. *Tipp:* Da $\mathbf{R}_x$ eine orthonormale Rotationsmatrix ist, gilt $\mathbf{R}_x^{-1} = \mathbf{R}_x^T$.

Aufgabe 15 (H) Quaternionen

Ähnlich wie komplexe Zahlen zur Darstellung von Rotationen im zweidimensionalen benutzt werden, können die von *William Rowan Hamilton* 1843 entdeckten *Quaternionen* zur Darstellung von Rotationen im dreidimensionalen Raum verwendet werden. In dieser Aufgabe sollen Sie ein paar der wesentlichen Eigenschaften von Quaternionen zeigen.

Quaternionen sind hyperkomplexe Zahl vom Rang 4 und lassen sich wie folgt darstellen:

$$q = q_0 + \mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3 = q_0 + \mathbf{q}$$

wobei für $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ gilt:

$$\begin{aligned} \mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1 \qquad & \mathbf{ij} = \mathbf{k} = -\mathbf{ji} \\ \mathbf{jk} = \mathbf{i} = -\mathbf{kj} \qquad & \mathbf{ki} = \mathbf{j} = -\mathbf{ik} \end{aligned}$$

Die *Addition* zweier Quaternionen ist als Addition der entsprechenden vier Komponenten definiert; zwei Quaternionen sind genau dann gleich, wenn alle vier Komponenten gleich sind.

Die Darstellung $q = q_0 + \mathbf{q}$ eines Quaternions kann auch als Kombination eines Skalars mit einem dreidimensionalen Vektor interpretiert werden. Das Quaternion $q^* = q_0 - \mathbf{q}$ eines Quaternions $q = q_0 + \mathbf{q}$ wird *komplex Konjugierte von q* genannt.

Die *Norm* $N(q)$ eines Quaternions ist wie folgt definiert:

$$\begin{aligned} N^2(q) &= q^* q = \\ &= q_0^2 + q_1^2 + q_2^2 + q_3^2 = |q|^2 \end{aligned}$$

Ein *Einheitsquaternion* ist ein Quaternion mit der Norm $N(q) = 1$.

a) Zeigen Sie: Die Multiplikation bei Quaternionen ist nicht kommutativ. Berechnen Sie hierzu auf der Basis der obigen Gleichungen das Produkt $r = pq$ zweier Quaternionen $q = q_0 + \mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3$ und $p = p_0 + \mathbf{i}p_1 + \mathbf{j}p_2 + \mathbf{k}p_3$.

b) Zeigen Sie: Das *Einselement* der Quaternionenmultiplikation ist $q = 1 + \mathbf{0}$.

c) Zeigen Sie: Das *inverse Element* der Quaternionenmultiplikation ist $q^{-1} = q^*/N^2(q)$.

d) Zeigen Sie: $(pq)^* = q^* p^*$.

Man kann zeigen, dass für jedes Einheitsquaternion $q = q_0 + \mathbf{q}$ die Operation $\mathbf{v}' = q\mathbf{v}q^*$ einen in homogenen Koordinaten dargestellten Punkt $\mathbf{v}$ im dreidimensionalen Raum auf einen Punkt $\mathbf{v}'$ derart abbildet, dass dies einer Drehung im Raum um die Achse $\mathbf{q}$ um den Winkel $2 \cdot acos(q_0)$ entspricht.

e) Berechnen Sie den Rotationsoperator $L_q(\mathbf{v}) = q\mathbf{v}q' =: \mathbf{v}'$ in Matrixdarstellung (d.h. als 3×3 Matrix R_q mit $\mathbf{v}' = R_q\mathbf{v}$). Berechnen Sie dazu zuerst eine Darstellung der einzelnen Komponenten von $\mathbf{v}'$ in Abhängigkeit der Komponenten von $\mathbf{v}$. Diese (lineare) Abhängigkeit kann dann in einer 3×3 Matrix codiert werden.

f) Zeigen Sie: Das Produkt pq der Quaternionen p und q definiert einen Rotationsoperator L_{pq} der einer Sequenz von Rotationsoperatoren, L_q gefolgt von L_p, entspricht.

Aufgabe 16 (P) Einbindung von VRML-Modellen in ARToolkit Anwendungen

VRML (**V**irtual **R**eality **M**arkup **L**anguage) ist ein Standard zur Beschreibung von 3D-Szenen. Die meisten 3D-Modellierungsprogramme unterstützen den Export von Modellen im VRML 97-Format. Das ARToolkit unterstützt durch die Bibliothek *OpenVRML* die Einbindung solcher Objekte, wodurch eine leichtere Darstellung komplexer 3D-Szenen ermöglicht wird.

VRML 97 beherrscht durch *JavaScript* gesteuerte Animationen. Diese werden wir im Praktikum allerdings nicht benutzen.

a) Machen Sie sich mit der Headerdatei `AR/arvrml.h` vertraut. Schauen Sie sich außerdem die Konfigurationsdateien der mit dem AR Toolkit gelieferten `simpleVRML` an.
Tip: Aus welchem Grund auch immer funktionieren die Parameter `translation` und `rotation` in der Konfigurationsdatei nicht.

b) Schreiben Sie eine Anwendung, die auf zwei Markern (*Hiro* und *Kanji*) die beiden VRML-Modelle `sheep.wrl` und `wolf.wrl` anzeigt. Nutzen Sie dafür die Funktionen `arVrmlLoadFile` und `arVrmlDraw`.

Aufgabe 17 (P) Der große böse Wolf und das kleine ängstliche Schaf

Wie Sie sicher wissen, ist der große böseWolf böse, und deshalb hat das kleine Schaf Angst vor ihm. Diese Angst äußert sich darin, dass das Schaf sich zum Wolf dreht und unmerklich zittert, sobald der Wolf ihm zu nahe kommt.

Die Darstellung dieser Szene mit einem Szenegraphen bietet den Vorteil, beliebige Beziehungen zwischen Objekten herleiten zu können. Um die Position und Orientierung von Objekt A im Koordinatensystem des Objekts B darzustellen, sucht man einen Pfad im Szenengraphen von B nach A, traversiert diesen und multipliziert die jeweils passierten Transformationen aufeinander auf – wenn man von "unten nach oben" geht, natürlich die Inversen davon. Als Beispiel kann man in Abbildung 1 die Position der Teekanne relativ zum Zucker folgendermaßen ausdrücken:

$$\mathbf{T}_{gesucht} = \mathbf{T}^{-1,L}_{Teekanne} \cdot \mathbf{T}_{Gedeck} \cdot \mathbf{T}_{Zucker}$$

Wie Sie sicher wissen, ist der große Wolf böse, und deshalb hat das kleine Schaf Angst vor ihm. Diese Angst äußert sich darin, dass das Schaf sich zum Wolf dreht und unmerklich zittert, sobald der Wolf ihm zu nahe kommt. Erweitern Sie Ihre Anwendung aus Aufgabe 14(b) derart, dass sich das Schaf zum Wolf dreht sobald der ”Wolfmarker” weniger als `15 cm` vom ”Schafmarker” entfernt ist. Die nötige Drehung können Sie wie folgt ermitteln:

- Bestimmen Sie den Vektor vom Schaf- zum Wolfmarker mit Hilfe hierarchischer Transformationskonstrukte (Szenegraph).
- Projizieren Sie diesen Vektor auf die (x, y)-Ebene des Schafmarkers.
- Drehen Sie das Schafmodell um die z-Achse des Schafmarkersystems, so daß es in die Richtung des projizierten Vektors schaut.