Ludwig-Maximilians-Universität München
LFE Medieninformatik
Prof. Dr. Andreas Butz
Fabian Hennecke

Sommersemester 2009
Übungsblatt 2
5. Mai 2009

Übung "Augmented Reality"

Abgabetermin:
Die Lösung zu diesem Übungsblatt ist bis zum 15. Mai 2009 abzugeben.

Inhalt:
Dieses Übungsblatt gibt eine Einführung in die Augmented Reality mit dem ARToolkit. Außerdem werden die wesentlichen Eigenschaften von Rotationsmatrizen zur Beschreibung dreidimensionaler Orientierung wiederholt und homogene Koordinaten und Matrizen eingeführt.

Aufgabe 7 (P) C und ARToolkit

Das ARToolkit ist eine Softwarebibliothek in der Programmiersprache C für markerbasiertes optisches Tracking. Es wird hauptsächlich von Dr. Hirokazu Kato an der Osaka University in Japan, entwickelt, und wird durch das Human Interface Technology Laboratory (HIT Lab.) der University of Washington, sowie das HIT Lab NZ an der University of Canterbury, Neuseeland, unterstützt.

a) Laden Sie sich den Quellcode des ARToolkit von der Praktikumshomepage herunter.

b) Machen Sie sich mit den mitgelieferten Beispielprogrammen vertraut, übersetzen Sie das simpleTest Beispiel und testen Sie das übersetzte Programm. Drucken Sie sich dazu den entsprechenden Marker aus.

c) Senden Sie einen Screenshot Ihres laufenden Programms (mit Ihnen im Bild) als Lösung für diese Aufgabe ein.

Aufgabe 8 (P) GLUT - OpenGL Utility Toolkit

GLUT, das OpenGL Utility Toolkit, ist eine plattformunabhängige Bibliothek zum Schreiben von OpenGL Programmen. GLUT bietet einige Funktionen die das Zeichnen von Objekten und die Interaktion mit Tastatur und Maus einfacher machen.

a) Laden und übersetzen Sie das Programm `prakt02_default.tgz` von der Übungswebsite.

b) Finden Sie die Funktion zum Zeichnen einer Teekanne, und ersetzen Sie den Würfel aus dem Beispielprogramm durch eine Teekanne.

c) Finden Sie die Funktion zum Auswerten von Taststatureingaben. Ändern Sie ihr Programm, so dass auf Druck der Taste ”w” die Teekanne abwechelnd als Drahtgittermodell (Tipp: Funktion `glutWireTeapot`) und als echte Teekanne gezeichnet wird. Benennen Sie das Programm und die relevanten Dateien in *prakt02_teapot* um und schicken Sie die Quelldateien an die Übungsleitung.

<u>Aufgabe 9 (H)</u> Rotationsmatrizen

Bewegungen von Objekten im mehrdimensionalen Raum können auf verschiedene Weisen dargestellt werden. Ein Beispiel ist die Darstellung durch Translationen und Rotationen. Rotationen können durch Matrizen beschrieben werden. Ein Punkt ~x im zweidimensionalen Raum kann mit Hilfe einer Matrix um den Winkel a (in mathematisch positiver Drehrichtung) rotiert werden:

$$R_\alpha = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}$$

a) Berechnen Sie den Punkt der durch die Rotation des Punktes $\begin{pmatrix} 1 \\ 2 \end{pmatrix}$ um 90° entsteht. Zeigen Sie, dass $R_\alpha \cdot R_\beta \cdot \vec{x} = R_\beta \cdot R_\alpha \cdot \vec{x}$ für beliebige α und β gilt. Warum ist diese Aussage immer gültig?

Jede Rotation im Dreidimensionalen kann intuitiv dargestellt werden indem die Basisvektoren des $\mathbf{R}^3$ auf drei neue Basisvektoren abgebildet werden. In diesem neuen Koordinatensystem wird dann das Objekt dargestellt. Diese lineare Abbildung entspricht genau einer 3×3 Matrix deren Spalten die neuen Basisvektoren sind.

b) Geben Sie die Rotationsmatrix an, die den Punkt x = $(4,2,0)^T$ auf den Punkt $(0,2,-4)^T$ abbildet.

c) Kann es Transformationen geben bei denen nur eine Koordinatenachse verändert wird? Falls ja, welche Art von Transformationen sind dies?

Eine Rotation im 3D-Raum um eine beliebige Achse x und einen beliebigen Winkel f, kann nach Euler immer in eine Serie von Rotationen um drei linear unabhängige Achsen zerlegt werden. Hierbei ist ω die Rotation um die *x*-Achse,

$$R(\omega,0,0) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & \sin\omega \\ 0 & -\sin\omega & \cos\omega \end{pmatrix}$$

φ die Rotation um die y-Achse,

$$R(0,\varphi,0) = \begin{pmatrix} \cos\varphi & 0 & -\sin\varphi \\ 0 & 1 & 0 \\ \sin\varphi & 0 & \cos\varphi \end{pmatrix}$$

und κ die Rotation um die z-Achse.

$$R(0,0,\kappa) = \begin{pmatrix} \cos\kappa & \sin\kappa & 0 \\ -\sin\kappa & \cos\kappa & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

d) Zeigen Sie, dass gilt:

$$R(\omega,0,0) \cdot R(0,\varphi,0) \cdot R(0,0,\kappa) \cdot \vec{x} \neq R(0,0,\kappa) \cdot R(0,\varphi,0) \cdot R(\omega,0,0) \cdot \vec{x}\ .$$

Begründen Sie diese Eigenschaft.

e) Wie lautet die Matrix, die eine solche Rotation rückgängig macht?

<u>Aufgabe 10 (H)</u> Homogene Transformationsmatrizen

Die *Translationsmatrix* $T = (t_X, t_Y, t_Z)^T$ stellt eine Verschiebung vom Ursprung des Koordinatensystems dar. Die allgemeine Transformationsgleichung

$$p' = R \cdot p + t$$

ist nicht linear und kann daher nicht invertiert werden. Für Szenegraphen ist die Invertierung jedoch zwingend notwendig. Durch Hinzufügen einer weiteren Dimension, d.h.

$$\begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} \mapsto \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} = \begin{pmatrix} \lambda \cdot p_x \\ \lambda \cdot p_y \\ \lambda \cdot p_z \\ \lambda \end{pmatrix}$$

kann diese Gleichung wieder in eine lineare Abbildung umgewandelt werden:

$$p^* = \begin{pmatrix} p_{x'} \\ p_{y'} \\ p_{z'} \\ 1 \end{pmatrix} = \begin{pmatrix} & & & t_x \\ & R(\omega,\varphi,\kappa) & & t_y \\ & & & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix}$$

Diese 4×4 Matrixdarstellung rigider Transformationen im 3D-Raum ist in der Computergraphik sehr gebräuchlich und wird *Homogene Transformationsmatrix* genannt.

Zeigen Sie, dass die o.g. lineare Gleichung genau der Berechnung $p' = R \cdot p + t$ entspricht.

<u>Aufgabe 11 (P)</u> Transformationen in OpenGL

In *OpenGL* werden Transformationen auf Objekte so angewandt, dass zuerst das Koordinatensystem transformiert wird und dann das Objekt im neuen Koordinatensystem gezeichnet wird.

a) Was bedeutet das für die Transformationen (Rotationen, Translationen)?

b) Wenden Sie Ihr neu erworbenes Wissen über Transformationen an, indem Sie in Ihrem ARToolkit Programm aus Aufgabe 8c einen Würfel im Ursprung mit dem *OpenGL*-Befehl `glScalef` zuerst so ungleichmässig skalieren daß er sieben mal so hoch wie lang und breit ist. Rotieren Sie ihn dann um 45° um die x-Achse und verschieben Sie ihn an einen Punkt im Raum. Zeichen Sie mit OpenGL den Würfel nach jedem Schritt, so daß die Abfolge der Transformationen sichtbar wird.

c) Experimentieren Sie mit der Reihenfolge der Transformationen und beobachten Sie die Effekte. Verwenden Sie die Tastaturfunktion aus Aufgabe 8c um durch Druck auf die Taste ”r” zwischen verschiedenen Reihenfolgen umzuschalten. Senden Sie Ihr Ergebnisprogramm mit dem Namen `prakt05` wie in Aufgabe 8 beschrieben ein!

<u>Aufgabe 12 (P)</u> Transformationen mit dem Matrix-Stack

OpenGL stellt mit den Funktionen `glPushMatrix` und `glPopMatrix` eine komfortable Möglichkeit zur Verfügung, Transformationsmatrizen auf einem Keller zwischenzuspeichern. Dies kann man z.B. dazu benutzen, eine Transformation eines Objekts vorzugeben und diese jeweils mit mehreren voneinander unabhängigen weiteren Transformationen zu koppeln.

a) Schreiben Sie ein Programm, das auf einem Marker zwei Objekte anzeigt. Zu Beginn soll ein Objekt 10cm links des Markers, das andere 10cm rechts des Markers angezeigt werden.

b) Sorgen Sie dafür, dass durch Drücken der Pfeiltasten die Objekte näher zusammen bzw. auseinander rücken. Durch Druck auf die Taste ”r” soll der Startzustand wiederhergestellt werden. Benutzen Sie für die Implementierung die *OpenGL*-Stackoperationen.

Machen Sie sich zum Lösen dieser Aufgabe zuerst mit den im *ARToolkit* verwendeten Koordinatensystemen vertraut, eine gute Erklärung finden Sie in der auf der Übungswebsite hinterlegten Präsentation von Hirokazu Kato. Die eigentliche Translation von Objekten geschieht durch Veränderung der *OpenGL*-Modelviewmatrix mittels `glTranslate3f` an geeigneter Stelle.

Aufgabe 13 (P) Mehrere Marker in ARToolkit

Das ARToolkit stellt die Möglichkeit zur Verfügung, dass mehrere Marker simultan erkannt werden können. Dies kann z.B. dazu verwendet werden, verschiedene Objekte auf unterschiedlichen Markern anzuzeigen.

a) Entwerfen Sie ein Programm, das drei beliebige Marker des ARToolkit simultan erkennt und verschiedene Objekte darauf anzeigt.

b) Sorgen Sie dafür, dass diese Objekte durch Drücken der Leertaste die Markerplätze tauschen. In Ihrer Lösung sollten die Markerbeschreibungen (welche wurden verwendet) enthalten sein.