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

Sommersemester 2009
Übungsblatt 4
9. Juni 2009

Übung ”Augmented Reality”

Abgabetermin:
Die Lösung zu diesem Übungsblatt ist bis zum 09. Juni 2009 abzugeben.

Inhalt:
In diesem Blatt wird das Multimarker-Feature des AR Toolkit eingeführt, wodurch die Trackingstabilität erheblich verbessert wird. Außerdem wird anhand eines zweckfreien und verspäteten „Tanz in den Mai“ die Darstellung von VRML-Objekten und die Interaktion mittels Kollisionen im 3D-Raum wiederholt und vertieft.

Aufgabe 18 (P) Tanz in den Mai

Ein großes Problem des markerbasierten Trackings mit dem AR Toolkit ist, dass ein Marker stets voll sichtbar sein muss. Sobald auch nur ein kleiner Teil davon abgedeckt ist, wird nichts mehr erkannt und die Anwendung schlägt fehl. Mit dem Multimarker-Feature wird dieses Problem gemildert. Hierdurch können mehrere Marker in fester räumlicher Anordnung angegeben werden. Sobald mindestens einer davon sichtbar ist, können die Positionen der übrigen abgeleitet werden.
Bei dieser Aufgabe empfiehlt es sich, die Szenengraphenkonzepte des zweiten Blatts anzuwenden. Schreiben Sie sich also auf einem Blatt Papier die räumlichen Beziehungen zwischen der Kamera, diversen Markern und Tieren auf und nutzen Sie diese Notation zur Laufzeitberechnung der nötigen räumlichen Transformationen.

a) Machen Sie sich mit dem Headerfile `AR/arMulti.h` vertraut. Hilfreich ist dabei u.a. die Beispielanwendung `multiTest`. Schauen Sie sich besonders die Funktionen `arMultiReadConfigFile` und `arMultigetTransMat` an.

b) Laden Sie sich das Multimarker-Pattern `patt_mm_a16.pdf` und die zugehörigen Beschreibungsdateien von der Übungsseite. Drucken Sie es aus und erstellen Sie ein passendes Konfigurationsfile (Tipp: Die Abstände zwischen und Größen der Marker sind in einem Raster von 5mm gegeben).

c) Zeigen Sie auf dem großen Multi-Pattern einen Maibaum an. Diesen können Sie entweder als VRML-Modell erstellen oder direkt in OpenGL als Sammlung von alternierenden weißen und blauen Kegelstümpfen codieren. Kegelstümpfe können Sie mittels `gluNewQuadric`, `gluCylinder` und `gluDeleteQuadric` erzeugen.

d) Lassen Sie eine per Kommandozeilenoption bestimmbare Anzahl von Schafen um den Maibaum tanzen. Die Schafe sollen dabei immer in die Bewegungsrichtung blicken, gleichmäßig um den Baum verteilt sein und mit konstanter Geschwindigkeit um den Baum kreisen. Die Geschwindigkeit soll mit den Cursortasten regulierbar sein (nach rechts schneller, nach links langsamer).

e) Der große böse Wolf sitzt auch diesmal auf einem Marker. Zeichnen Sie ihn, und bringen Sie die Schafe zum sofortigen Stillstand, wenn der Wolf ihnen (d.h. dem Maibaum) zu nahe kommt. Lassen Sie auch jetzt alle Schafe in Richtung des Wolfes schauen. Benutzen Sie dazu eine ähnliche Vorgehensweise wie in Aufgabe 17(a). Beachten Sie dabei, dass die endgültige Position des Schafs durch eine Kette von Rotations- und Translationsoperationen erzeugt wurde.

Aufgabe 19 (P) Slider

Mit einem Streifen von Markern ist es möglich, einen Slider als Eingabegerät zu realisieren.

a) Laden Sie von der Übungsseite die Datei `slidermarker.tgz`, durcken Sie das Slidermarker-PDF und kleben Sie es auf einen Karton.

b) Schreiben Sie eine Anwendung, welche alle 7 Marker als großen Multimarker erkennt und außerdem einen Slider durch Verdeckungen implementiert. Sorgen Sie dafür, dass nur unter folgenden Bedingungen der Wert des Sliders verändert wird:

- Die beiden äußersten Patterns sind sichtbar.
- Alle bis auf maximal zwei nebeneinander liegende innere Patterns sind sichtbar.
- Dieser Zustand hält für minimal 200ms an. Benutzen Sie hierfür die Funktion `currentTime()` aus der `math_library`, die Sie auf der Übungsseite finden.
- Ist genau ein Pattern verdeckt, so wird der Slider auf den entsprechenden Prozentwert gesetzt (also 0, 25, 50, 75, 100), sind zwei nebeneinander liegende Patterns verdeckt, so wird der Slider auf den Mittelwert gesetzt (also 12,5, 37,5, 62,5, 87,5).

c) Benutzen Sie die `itrans` Werte der `struct ARMultiEachMarkerInfoT` um den aktuellen Wert des Sliders grafisch anzuzeigen. Hierzu empfiehlt sich die Verwendung von `glBegin(GL_QUADS); glVertex3d...; glEnd()`.

d) Zeigen Sie zusätzlich in einer anderen Farbe immer die volle mögliche Bandbreite des Sliders an.

e) Verbinden Sie den soeben implementierten Slider mit einer der vorhergehenden Aufgaben, indem Sie deren Tastatursteuerung durch die Slidersteuerung ersetzen.

Anmerkung zur `math_library` (prakt12_default.zip):
Datei von der Vorlesungsseite herunterladen, ins ARTK entpacken und das Projekt in die ARTK-Solution importieren. Anschließend in den Projekteigenschaften kontrollieren, ob der Configuration Type „Static Library (.lib)“ ist bzw. dies ggf. einstellen. Anschließend das Output-Folder des Projekts als zusätzlichen Library-Pfad für das Projekt von Aufgabe 19 einstellen, die Library explizit als zu importierende Library in den Linker-Eigenschaften angeben und natürlich die `math_library.h` includen.
Bei weiteren Problemen einfach auf http://www.die-informatiker.net fragen oder mir eine Mail (fabian.hennecke_AT_ifi.lmu.de) schicken.