

Sommersemester 2007

Übungen zu Computergraphik II - Blatt 2

Abgabe am Montag, dem 21. 05. 2007, 13:00 Uhr

Aufgabe 1 (Raytracing, 6 Punkte)

Wir haben das mittlerweile wohlbekannte Raytracing-Framework um eine Funktion zum Laden von Dreiecksnetzen erweitert. Dazu wird das *OBJ*-Format verwendet. Auf der Vorlesungshomepage finden Sie einen Link zu einer genaueren Beschreibung dieses Formats, was für das Lösen dieser Aufgabe allerdings nicht von Bedeutung ist. Die Dateien `spaceship.xml` und `chicken.xml` enthalten Szenen mit solchen Dreiecksnetzen.

Implementieren Sie in der Funktion `Triangle::intersect()` in `Triangle.cpp` einen Strahl-Dreieck-Schnitttest.

Hinweis: Schauen Sie sich in Ihren Computergraphik-I-Unterlagen noch einmal das Kapitel über baryzentrische Koordinaten an.

Aufgabe 2 (Raytracing, 14 Punkte)

Implementieren Sie **zwei** (7+7 Punkte) der folgenden Methoden des Distribution Ray Tracings:

- (i) Antialiasing
- (ii) Weiche Schatten
- (iii) Tiefenschärfe
- (iv) Spekulare Reflexion

oder **eine** (14 Punkte) dieser Distribution Ray Tracing-Techniken:

- (v) Effiziente Soft Shadows
- (vi) Motion Blur

Hinweise:

- zu (i): Generieren Sie in der Funktion `Raytracer::render()` statt lediglich eines Strahls pro Pixel, mehrere Strahlen, und mitteln Sie das Ergebnis.
- zu (ii): Verfolgen Sie in der `Raytracer::shade()` mehrere Schattenstrahlen. Gehen Sie dabei von einer festen Grösse der Lichtquelle aus. (Wenn Sie viel Zeit und Lust haben, dürfen Sie das Framework natürlich auch gerne um eine flexible Klasse für Flächenlichtquellen erweitern, bei der man die Grösse variabel einstellen kann. Wenn Sie eine solche Klasse implementieren, müssen Sie keine zweite Distribution Ray Tracing-Technik mehr implementieren, um volle Punktzahl für diese Aufgabe zu erhalten).

- zu (iii): Generieren Sie in der Funktion `Raytracer::render()` statt lediglich eines Strahls pro Pixel mehrere Strahlen und mitteln Sie das Ergebnis. Auch hier gilt: Natürlich dürfen Sie das Framework auch gerne um eine Klasse `LensCamera` erweitern. Wie bei dem Algorithmus für weiche Schatten müssen Sie in diesem Fall keine zweite Technik mehr implementieren.).
- zu (iv): Generieren Sie in der Funktion `Raytracer::traceRay()` mehrere reflektierte Strahlen und addieren Sie die einzelnen Beiträge gemäß dem Phong-Modell auf.
- zu (v): Erweitern Sie die Funktion `Raytracer::shade()` um das Verfahren für effiziente Soft Shadows. Sie können hier von einer festen Grösse der Lichtquellen ausgehen.
- zu (vi): Erweitern Sie die einzelnen geometrischen Objekte um einen Start- und Zielpunkt und legen Sie eine Zeit fest, die diese Objekte für diese Strecke benötigen (Bei Kugeln können Sie beispielsweise Start- und Zielposition des Mittelpunktes angeben. Die Dreiecksnetze können mittels der Funktion `moveTo()` bewegt werden). Wählen Sie in `Raytracer::render()` einen zufälligen Zeitpunkt für jedes Pixel, und interpolieren Sie die Bewegung linear zwischen Start- und Zielpunkt.

Auf der Vorlesungshomepage steht ein neues Framework mit den oben beschriebenen Erweiterungen bereit. Dieses Framework enthält noch keine Funktionalität zur Berechnung der gebrochenen Strahlen, der reflektierten Strahlen, usw...., da dieses Bestandteil des 1. Übungszettels war, dessen Abgabe bekanntlich auf kommenden Donnerstag verschoben wurde. An diesem Donnerstag wird es durch ein neues Framework ersetzt, welches auch die Lösungen des 1. Übungszettels enthält. So können auch Gruppen, die den ersten Übungszettel nicht gelöst haben, den 2. Übungszettel bearbeiten.

Schauen Sie sich das neue Framework bitte VOR! der nächsten Übungsstunde genau an, so dass Sie dort dann eventuelle Fragen formulieren können.