



AUFGABE 1: Programmierung eines Ray Tracers

Aufbauend auf einem Framework der vorangegangenen Übungen soll schrittweise ein Ray Tracer entwickelt werden. Ein Dreiecksmodell soll geladen und mittels der OpenGL Rendering-Pipeline dargestellt werden. Ist die gewünschte Ansicht ausgewählt, so soll ein Ray Trace der Szene erzeugt werden.

- a) Nach der „Ray Trace“ Aktion des Benutzers sind in der Funktion `updateView()` die aktuellen Kameraparameter auszulesen, die für das Erzeugen von Primärstrahlen benötigt werden. Mittels `gluUnProject(...)` können Start- und Endpunkte für die Strahlen bestimmt werden. Die Ansicht soll auf Wunsch wiederhergestellt werden können. Anschließend sollen die Parameter von der Funktion `createPrimaryRays()` verwendet werden, die für alle Pixel Strahlen erzeugt und diese verfolgt. Die Dichte der Strahlen soll regelbar sein, so dass ein Subsampling möglich ist (Supersampling ist optional). (1 Punkt)
- b) Implementieren Sie eine Funktion `intersectTriangle()`, die Schnittpunkte mit den Dreiecken des aktuell geladenen Modells bestimmt. Der vorderste (= sichtbare) dieser Schnitte soll als Punkt mittels `GL_POINTS` gerendert werden (in 3D-Koordinaten!). Bei Sub- oder Supersampling sollte `glPointSize()` entsprechend angepasst werden. Die Normale des Dreiecks soll durch lineare Interpolation aus den Vertex-Normalen ermittelt werden (baryzentrische Koordinaten verwenden). Zunächst kann sie mittels `glNormal()` übergeben werden. (1 Punkt)
- c) Lesen sie die in OpenGL aktiven Lichtquellen aus. Verwenden Sie nun das Modell für ambientes und ideal diffuses Licht und schalten Sie `GL_LIGHTING` aus. Die resultierende Farbe ermittelt sich aus den Farben des Lichts und des Materials (komponentenweise Multiplikation), unter Berücksichtigung der Reflektionseigenschaften und der Reflektionswinkel. Verwenden Sie zur Demonstration verschiedenfarbige Lichtquellen und Materialien. (1 Punkt)
- d) Berechnen sie schattierte Bereiche der Szene mittels Schattenfählern. Erzeugen Sie sekundäre ideal reflektierende Strahlen und führen Sie die Rekursion durch. Die gerichtet diffuse Materialkomponente kann für die Spiegelung verwendet werden (Transmission ist optional). (1 Punkt)
- e) Schreiben Sie eine Routine zum Speichern des Bildes. Das Format kann frei gewählt werden (PPM wird der Einfachheit halber empfohlen, die Spezifikation findet man im Netz). (1 Punkt)

Bonuspunkte: Für die Implementierung der folgenden Extrafunktionalitäten werden Bonuspunkte vergeben.

- a) Kd-Baum / modifizierter BSP Baum zur effizienten Traversierung der Szene. (2 Punkte)
- b) Schärfentiefe (Depth of field). (1 Punkt)
- c) Motion Blur (verschiedene Positionen müssen interpoliert werden – z.B. Translation) (1 Punkt)
- d) Halbschatten bei ausgedehnten Lichtquellen (1 Punkt)
- e) Gerichtet diffuse Streuung (1 Punkt)
- f) Bump Mapping (z.B. prozedural mittels Perlin Noise) (1 Punkt)
- g) Texture Mapping (1 Punkt), mit Mip Mapping (+1 Punkt)

AUFGABE 2: Theoriefragen

1. Warum werden bei JPEG 8x8 Blöcke verwendet? Diskutieren Sie den Effekt kleinerer bzw. größerer Blöcke (1 Punkt)
2. Wie müsste man welche Parameter des JPEG-Verfahrens wählen, damit die Kompression verlustfrei wäre? (1 Punkt)
3. Diskutieren Sie, wie man im Ray-Tracing-Verfahren den Schatten von semi-transparenten Objekten bestimmen kann. (1 Punkt)
4. In der Praxis geht man bei Radiosity Verfahren so vor, dass die Szene nur dort fein unterteilt wird, wo dies auch zur Darstellung der Radiosity Funktion notwendig ist. Dafür beginnt man mit einer groben Aufteilung und unterteilt, wenn man einen großen Fehler durch die konstante Approximation auf dem Flächenstück befürchten muss. Entwerfen sie eine Funktion, mit der der potentielle Approximationsfehler nach oben abgeschätzt werden kann. (1 Punkt)
5. Beschreiben Sie, wie man Ray Tracing und Radiosity geeignet kombinieren kann. (1 Punkt)