
Aufgabenblatt 3

Sichtbarkeit mittels Raycasting

Der zentrale Inhalt dieser Aufgabenserie wird es sein, Raycasting zu implementieren. Im Kontext der Sichtbarkeit wird durch Raycasting berechnet, welche Objekte von einem von der Kamera ausgehenden Strahl getroffen werden und somit sichtbar sind. Dabei sollen Schnittberechnungen mit Kugeln und Dreiecken möglich sein. Informieren Sie sich über Raycasting ausgehend von der Vorlesung vom 26. Mai. Diese Aufgabenserie setzt voraus, dass Aufgabenblatt 2 vollständig gelöst wurde.

Folgendes muss in dieser Aufgabe implementiert werden:

1. *main(...)*: Kommentieren Sie den Code ein, der für die Erzeugung der Kamera *cam* und dem Hinzufügen der Kamera zur Szene verantwortlich ist.
2. *main(...)*: Erzeugen Sie mindestens ein Objekt der Klasse *Sphere* und fügen Sie dieses mittels *Scene::addSphere(...)* zur Szene hinzu.
3. *main(...)*: Führen Sie alle notwendigen Transformationen an den Modellen durch (siehe Abbildung 1).
4. *main(...)*: Erzeugen Sie ein Objekt der Klasse *SolidRenderer* und führen Sie *SolidRenderer::renderRaycast()* aus.
5. Die Funktion *SolidRenderer::renderRaycast()* ist gegeben und führt *SolidRenderer::computeImageRow(size_t row)* aus. Die Aufteilung in Zeilen ermöglicht eine einfache Parallelisierungsstrategie. Im Quelltext stehen Ihnen drei Optionen zur Verfügung. Den schnellsten Code erhalten Sie mittels OpenMP. Sollte OpenMP auf Ihrem System nicht funktionieren, können Sie auf TBB ausweichen. Um optimierten Code mit Parallelisierung zu erhalten, sollten Sie mit *ccmake* (benötigt ggf. ein zusätzliches Paket) den *CMAKE_BUILD_TYPE* auf *Release* stellen.
6. Implementieren Sie die Methode *SolidRenderer::computeImageRow(size_t rowNum)*. Hier soll für jede Spalte innerhalb der Zeile ein Strahl generiert werden. Hierfür steht Ihnen die Funktion *Camera::getRay(size_t column, size_t row)* zur Verfügung. Befassen Sie sich mit der Struktur *HitRecord*. Dieses Objekt wird zum Speichern notwendiger Informationen für einen Strahl genutzt. Erzeugen sie für einen Pixel einen *HitRecord* und initialisieren Sie die für das Raycasting notwendigen Variablen (*color*, *parameter*, *triangleId*, *sphereId*). Innerhalb dieser Methode soll mit dem Aufruf von *Scene::intersect(...)* überprüft werden, ob ein Objekt getroffen wurde. Ist dies der Fall, setzen Sie die Farbe eines Pixels mit *Image::setValue(...)*. Wird nichts getroffen, so soll das Pixel die Hintergrundfarbe $[0,0,0.7]$ erhalten.

7. Implementieren Sie die Methode `Scene::intersect(...)`. Gehen Sie dabei über alle Kugeln ihrer Szene und rufen Sie für jede Kugel `Scene::sphereIntersect(...)` auf. Bei einem Treffer muss der `HitRecord` aktualisiert werden. Gehen sie analog in allen Modellen Ihrer Szene über alle Dreiecke und rufen Sie `Scene::triangleIntersect(...)` auf. Beachten Sie dabei, dass Sie temporär ein Dreieck in Abhängigkeit ihrer Transformationen erzeugen müssen. Wenn eine Kugel **oder** ein Dreieck von einem Strahl getroffen wurde, gibt die Methode `true` zurück. Der Parameter `epsilon` kann genutzt werden um numerische Ungenauigkeiten zu behandeln.
8. Implementieren Sie den Schnittpunkttest für einen Strahl mit einer impliziten Kugel `Scene::sphereIntersect(...)`. Orientieren Sie sich dabei an der Vorlesung (Sichtbarkeit). Weitere Hinweise finden Sie im Netz.
9. Implementieren Sie den Schnittpunkttest für einen Strahl mit einem Dreieck `Scene::triangleIntersect(...)`. Orientieren Sie sich dabei an der Vorlesung (Sichtbarkeit). Weitere Hinweise finden Sie im Netz.

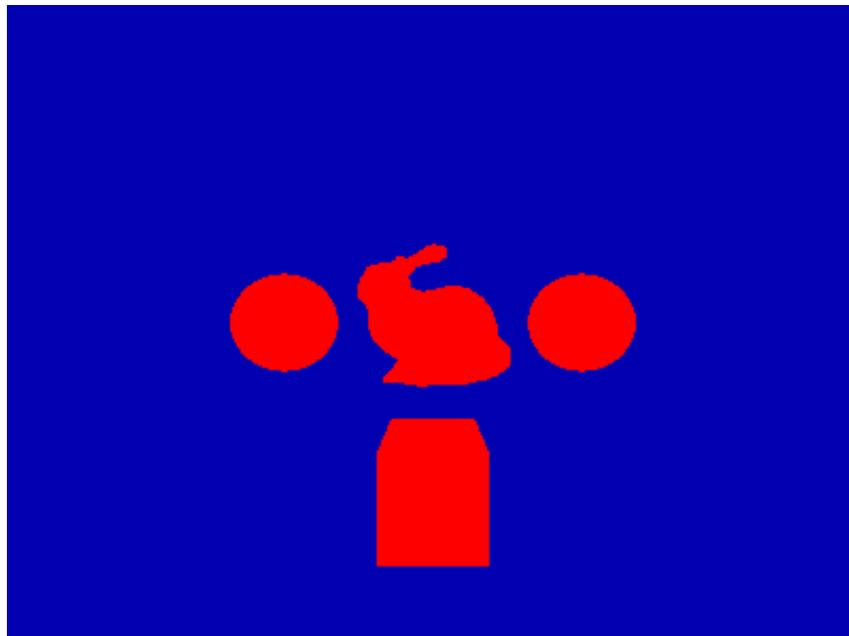


Abbildung 1: Zielsetzung der Sichtbarkeitsberechnung durch Raycasting (Beispiel)
Anmerkung: Für dieses Rendering wurde der Cube um 100 Einheiten in negativer y Richtung verschoben. Das Bunny-Modell wurde 10 Einheiten in negativer y Richtung und 30 Einheiten in negativer z Richtung verschoben und um 170 Grad **um** die y-Achse rotiert (bitte beachten Sie, dass das Modell um die eigene Achse gedreht wurde). Die Kugeln haben einen Radius von 50 Einheiten. Eine Kugel hat ihr Zentrum im Punkt $[-150, 0, -30]$ und die zweite Kugel hat ihr Zentrum im Punkt $[150,0,-30]$.