



## 12. Übung zur Vorlesung „Computergrafik I“

Wintersemester 2005/06

20. Januar 2006

Abgabe: 30.01.2006 in der Übung

### Aufgabe 12.1:

(3 Punkte)

In einer Szene für ‘Ray Tracing’ sei die Spezifikation der Kamera durch die Parameter

$$eye = (0, 0, -5), \quad \vec{u} = (1, 0, 0), \quad \vec{v} = (0, 1, 0),$$

den Blickwinkel mit  $30^\circ$ ,  $N = 5$  und den ‘aspect ratio’ 1.5 gegeben. Für die Werte

$$nRows = 480, \quad nCols = 640$$

berechne man die Parametergleichung für den  $rc$ -Strahl mit  $r = 100$ ,  $c = 200$ .

### Aufgabe 12.2:

(3 Punkte)

Angenommen ein Ellipsoid  $W$  wird aus einer Einheitskugel durch die Befehle

```
translate 2 4 9, scale 1 4 4, sphere
```

gebildet. Die Einheitskugel wird also erst skaliert und anschliessend verschoben.

Stellen Sie die Transformationsmatrix  $M$  in homogenen Koordinaten auf, berechnen Sie deren Inverse  $M^{-1}$  und finden Sie den Schnittpunkt von  $W$  mit dem Strahl

$$(10, 20, 5) + (-8, -12, 4)t.$$

### Aufgabe 12.3:

(6 Punkte)

Entwerfen und programmieren Sie einen einfachen Ray-Traycer, der selbstleuchtende Objekte korrekt rendert, vernachlässigen Sie also ambiente, diffuse und spekulare Beleuchtungskomponenten. Mit dem Programm soll eine Szene bestehend aus drei Kugeln korrekt gerendert werden.

Experimentieren Sie mit verschiedenen Blockgrößen der Pixel, die Sie auf Tastendruck variieren.

Hinweis: Hilfsfiles zur Lösung der Programmieraufgabe finden Sie auf der Homepage zur Vorlesung.

Hinweise zu den Quelltexten:

- Die Raytracing-Routine (`void Camera::raytrace(Scene&, int)` in der Datei `Camera.cpp`) wird aus der `display`-Funktion des OpenGL-Programms heraus aufgerufen.
- Implementieren Sie den Schleifenrumpf der Raytracing-Methode entsprechend den Vorgaben aus der Vorlesung.
- Implementieren Sie die Schnittpunkt-Berechnung zwischen einer Kugel und einem Strahl.
  - Der Rumpf der Methode `bool Sphere::hit(Ray&, Intersection&)` ist in der Datei `SDL.cpp` vorgegeben.
  - Die Methode zur Schnittpunkt-Berechnung wird in der Methode `Color3 Scene::shade(Ray&)` für alle Objekte aufgerufen.
  - Als Argumente erhält die Schnittpunkt-Berechnung den aktuellen Strahl und eine Referenz auf ein `Intersection`-Objekt, welches zur Verwaltung der gefundenen Schnittpunkte dient und in `SDL.h` definiert ist.
  - Zur Transformation des gegebenen Strahls in das generische Koordinatensystem steht die Funktion `void makeGenericRay(Ray&, Affine4&, Ray&)` in `SDL.cpp` zur Verfügung.
- Zum Testen des Raytracers steht die Szene-Datei `testszene.dat` zur Verfügung.