

Sommersemester 2010

Übungen zu Computergraphik II - Blatt 1

Abgabe am Donnerstag, den 29. 04. 2010, 10:15 Uhr

Aufgabe 1 (Prozedurale Textur, 2 Punkte)

Betrachten wir den durch die parametrische Funktion

$$T : [0, 1]^2 \rightarrow \mathbb{R}^3, (u, v) \mapsto \begin{pmatrix} (R + r \cos(2\pi v)) \cos(2\pi u) \\ (R + r \cos(2\pi v)) \sin(2\pi u) \\ r \sin(2\pi v) \end{pmatrix}$$

gegebenen Torus, wobei r den kleinen Radius und R den großen Radius des Torus darstellt. Die Oberfläche dieses Torus soll nun auf folgende Art und Weise eingefärbt werden: ist $P = (x, y, z)$ ein Punkt der Oberfläche des Kreisringes, dann ergebe sich seine RGB-Farbe $C(P) \in [0, 1]^3$ wie folgt:

$$C(P) = \begin{pmatrix} r(P) \\ g(P) \\ b(P) \end{pmatrix} = \begin{pmatrix} 1, \text{ wenn } x > 0, \text{ sonst } 0 \\ 1, \text{ wenn } y > 0, \text{ sonst } 0 \\ 1, \text{ wenn } z > 0, \text{ sonst } 0 \end{pmatrix}$$

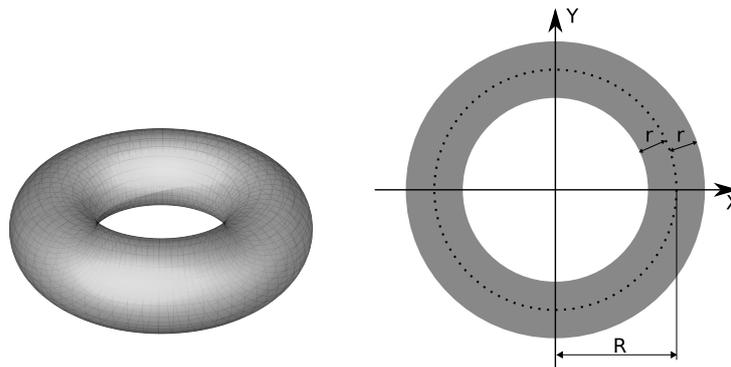


Abbildung 1: Torus

Ihre Aufgabe: Welche Färbung erhält die Oberfläche des Torus?

Aufgabe 2 (Texturkoordinaten, 4 Punkte)

Skizzieren Sie, wie die Textur aus Abb. 2 auf die Polygone in Abb. 3 mit den angegebenen Texturkoordinaten abgebildet wird. Gehen Sie davon aus, daß der Texturwiederholungsparameter auf `GL_REPEAT` gesetzt ist.

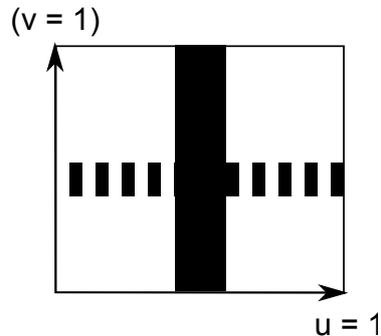


Abbildung 2: Textur

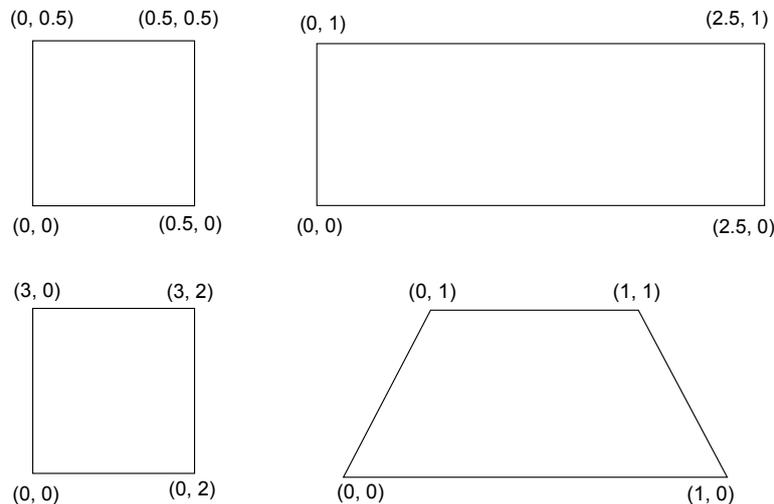


Abbildung 3: Die Koordinaten sind in der Form (u,v) angegeben.

Aufgabe 3 (Texturen in OpenGL, 6 Punkte)

Auf der Vorlesungshomepage finden sie das Framework *texture*. Im Programm wird ein Rechteck im Bereich $(0,0,0)$ bis $(1,0,10)$ gezeichnet. In der Funktion `generateTexture()` wird eine einfache, vierfarbige Textur erzeugt, die jetzt auf das Polygon gelegt werden soll.

- Aktivieren Sie die Texturierung und laden Sie die Textur auf die Graphikkarte hoch. Die Textur soll die Farbwerte des Polygons ersetzen (`GL_REPLACE`). Setzen Sie nun die Texturkoordinaten, so daß die Textur in z-Richtung (= v-Richtung der Textur) fünfmal wiederholt wird. Setzen Sie zunächst beide Texturfilter auf `GL_NEAREST`.
- Verändern Sie mit Hilfe der Texturmatrix die Texturkoordinaten, so daß der Eindruck einer sich auf den Betrachter zu bewegendem Textur entsteht. In den Funktionen `GLwidget::slotTextureFaster()` und `GLwidget::slotTextureSlower()` soll die Geschwindigkeit verstellt werden können.
- Ergänzen Sie die Funktionen `GLwidget::slotTexFilterNearest()`, `GLwidget::slotTexFilterLinear()` und `GLwidget::slotTexFilterMipMap()`, so daß die Texturfilter auf `GL_NEAREST`, `GL_LINEAR` und `GL_LINEAR_MIPMAP_LINEAR` gesetzt werden.