

Übung zu Computergraphik I

– Übungsblatt 8 –

Lehrstuhl für Computergraphik
und Multimediasysteme

Rene Winchenbach, Jan Mußmann

Abgabe: Bis spätestens **10. Dezember**, 10 Uhr
Besprechung: **Dienstag 17. Dezember und Mittwoch 18. Dezember**
Gesamtpunktzahl nach Übungsblatt 6: 46 von 65

Hinweis: Die Programmieraufgaben müssen per E-Mail an Jan Mußmann eingereicht werden. Geben Sie dabei bitte immer Ihren **Namen**, Ihre **Matrikelnummer**, sowie Ihre **Übungsgruppe (Di. / Mi.)** an. Geben Sie nur die von Ihnen **geänderten Dateien** ab (`GLWidget.cpp`, `vs.gls1` und `fs.gls1`).

Aufgabe 1 Texturfilterung (3 Punkte)

In Abbildung 1 ist eine Graustufentextur mit 8×8 Pixeln zu sehen, die auf die Texturkoordinaten $s, t \in [0, 1]$ abgebildet werden. Außerdem sei die Texturcoordinate $S = \begin{pmatrix} 2/3 \\ 11/20 \end{pmatrix}$ gegeben.

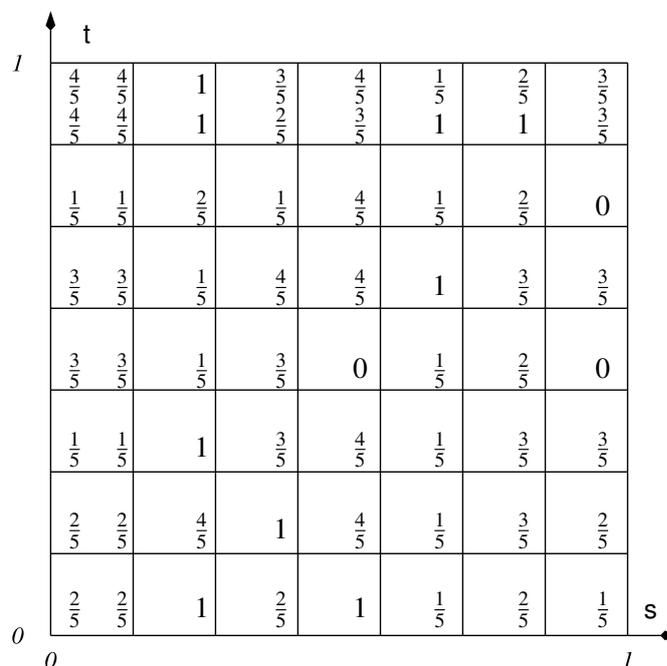


Abbildung 1: Textur mit Graustufen. Die Kreuzungspunkte sind die Mittelpunkte der Texel. Entsprechend sind die Grauwerte unmittelbar neben den Kreuzungspunkten angegeben.

- 1.1 Bestimmen Sie durch Nearest-Neighbor-Lookup den Grauwert für die Texturcoordinate S .
- 1.2 Bestimmen Sie mittels bilinearer Interpolation den Grauwert für die Texturcoordinate S .
- 1.3 Welchen Grauwert erhalten Sie für die Texturcoordinate $S_2 = \begin{pmatrix} 3/2 \\ 4/3 \end{pmatrix}$ durch Nearest-Neighbor-Lookup, wenn Texturkoordinaten durch Clamping fortgesetzt werden? Welchen Grauwert erhalten Sie, wenn Texturkoordinaten durch Repeat fortgesetzt werden? Schreiben Sie das konkrete Ergebnis hin und begründen Sie.

Aufgabe 2 Programmieren mit Texturen (3 Punkte)

In der folgenden Aufgabe soll das Programmgerüst `ueb08.zip` durch die Texturierung eines Objekts erweitert werden. Abbildung 2 zeigt einen Candy-Stick. Die Geometrie des Candy-Sticks entspricht der eines Zylinders. Die rot-weißen Streifen sollen mittels der Textur aus Abbildung 2(b) realisiert werden. In der Methode `setupCandyStick` wird bereits die Geometrie eines Zylinders erzeugt. Ebenso wird bereits eine Textur geladen und gebunden, sodass nur die Texturkoordinaten gesetzt und die Textur-Funktion implementiert werden müssen. Im Urzustand sollte das Programmgerüst einen rotierenden weißen Zylinder anzeigen.



Abbildung 2: *Rendering des Candy Stick 2(a) bestehend aus einer Zylinder-Geometrie und einer Textur mit diagonalen roten und weißen Streifen 2(b).*

- 2.1 Geben Sie für jeden Vertex des Zylinders Texturkoordinaten an. Passen Sie hierfür die Funktion `setupCandyStick()` in `GLWidget.cpp` an. Die Textur soll den Zylinder genau einmal komplett umwickeln. Es sollen keine sichtbaren Nahtstellen entstehen.
- 2.2 Im Programmgerüst kann mittels Tastaturtaste "m" zwischen den beiden Textur-Funktionen Replace und Modulate gewechselt werden. Die Beleuchtungsberechnung ist bereits im Vertex Shader umgesetzt. Die Farbe des beleuchteten Zylinders wird dem Fragment Shader als `vec3 vcolor` übergeben und kann dort verwendet werden. Implementieren Sie die beiden Textur-Funktionen im Fragment Shader in Abhängigkeit von der `uniform`-Variable `texEnvMode`.

Hinweis: Stellen Sie beim Ausführen des Programms sicher, dass sich die Textur `stripes.png` in Ihrem Arbeitsverzeichnis befindet.