

# Übung zu Computergraphik I

## – Übungsblatt 9 –

**Lehrstuhl für Computergraphik  
und Multimediasysteme**

Andreas Görlitz, Hendrik Hochstetter, John Rickard, Rene Winchenbach

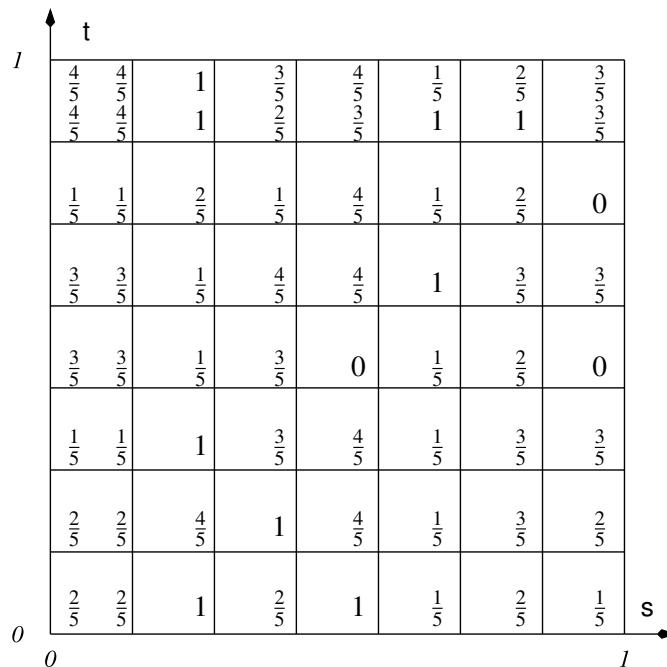
**Abgabe:** Für Studenten mit 5 LP verpflichtend  
bis spätestens Dienstag 05. Januar 2016, 10 Uhr.

**Besprechung:** Dienstag 12. Januar 2016 und Mittwoch 13. Januar 2016

**Hinweise:** Schriftliche Aufgaben bitte mit Name, Matrikelnummer und Übungsgruppe beschriften und zusammengeheftet in den Pappkarton vor Büro H-A 7115/1 werfen. Programmieraufgaben bitte per Mail mit Name und Matrikelnummer an Ihren jeweiligen Tutor senden. Geben Sie dabei nur Ihre modifizierte(n) Quelltextdatei(n) als Anhang ab.

### Aufgabe 1 Texturfilterung (1 Punkt)

In Abbildung 1 ist eine Graustufentextur mit  $8 \times 8$  Pixeln zu sehen, die auf die Texturkoordinaten  $s, t \in [0, 1]$  abgebildet werden. Außerdem sei die Texturkoordinate  $\mathbf{S} = \begin{pmatrix} 2/3 \\ 11/20 \end{pmatrix}$  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 Texturkoordinate  $S$ .
- 1.2 Bestimmen Sie mittels bilinearer Interpolation den Grauwert für die Texturkoordinate  $S$ .
- 1.3 Welchen Grauwert erhalten Sie für die Texturkoordinate  $S_2 = \begin{pmatrix} 3/2 \\ 4/3 \end{pmatrix}$  durch Nearest-Neighbor-Lookup, wenn Texturkoordinaten durch Clamping fortgesetzt werden? Welchen Grauwert erhalten Sie, wenn Texturkoordinaten durch Repeat fortgesetzt werden?

## Aufgabe 2 Programmieren mit Texturen (1 Punkt)

In der folgenden Aufgabe soll das Programmgerüst `ueb09.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. 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.

### Aufgabe 3 Das Aliasing-Problem und MipMapping (1 Punkt)

In dieser Aufgabe sollen Sie anhand einer kleinen Rechnung das Aliasing-Problem, das u.a. beim Texturieren regelmäßig auftritt, untersuchen. In Abbildung 3 ist der Graph der Funktion  $f(x) = \frac{1+\sin(50x^2)}{2}$  abgebildet.

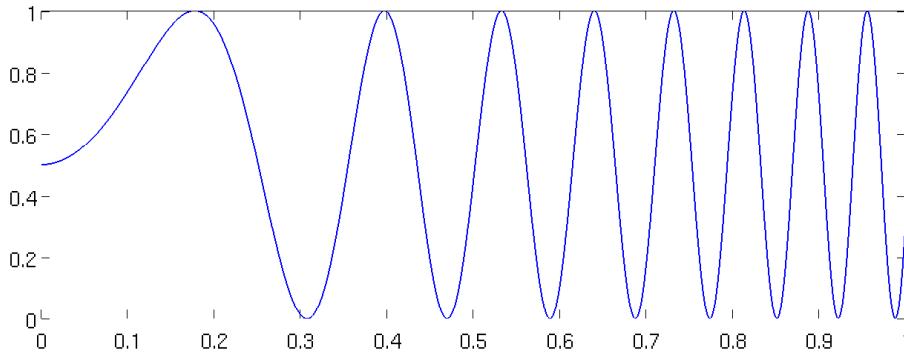


Abbildung 3: Graph der Funktion  $f(x)$ .

- 3.1 Es sei  $B$  eine 1D-Textur mit 28 Texeln, die die Graustufenwerte  $f\left(\frac{i}{27}\right), i \in \{0, \dots, 27\}$  haben. Berechnen Sie für die Texturkoordinaten  $S_1 = \frac{10}{13}$ ,  $S_2 = \frac{11}{13}$  und  $S_3 = \frac{12}{13}$  den Graustufenwert der Textur  $B$  mittels Nearest-Neighbor-Lookup.  
 Was fällt an diesen Werten im Vergleich zum Kurvenverlauf aus Abbildung 3 auf und womit ist dies zu erklären?
- 3.2 Um auftretenden Aliasing-Effekten entgegenzuwirken, wird bei der Texturierung häufig Mip-Mapping verwendet. Berechnen Sie für die Texturkoordinaten  $S_i, i \in \{1, 2, 3\}$  die Graustufenwerte des nächst größeren MipMap-Levels. Das nächst größere MipMap-Level  $B'$ , mit halbierter Auflösung von  $B$ , wird durch Mittlung der Graustufenwerte in  $B$  bestimmt.  
 Welcher Effekt soll durch die Verwendung von MipMapping erreicht werden und wie gut gelingt dies im obigen Beispiel?