

Sommersemester 2010

Übungen zu Computergraphik II - Blatt 3

Abgabe am Donnerstag, den 03. 06. 2010, 10:15 Uhr

Aufgabe 1 (Shader, 10 Punkte)

Auf der Vorlesungshomepage finden Sie das Framework `shader_2pass`. Sie können Szenen im Obj-Format laden und anzeigen. Ziel dieser Aufgabe ist es, die durch OpenGL dargestellte Szene in Echtzeit zu filtern. Da man für das Filtering zu jedem Fragment die benachbarten Fragmente einlesen muss, ist ein 2-Pass Verfahren nötig.

Im ersten Renderpass wird die Szene in ein sogenanntes Framebuffer-Objekt (FBO) gerendert. Ein FBO kann man sich vereinfacht als einen Bereich im Texturspeicher der Graphikkarte vorstellen. Im zweiten Renderpass wird das FBO als Textur eingebunden. Im Fragmentshader haben Sie nun über die Textur Zugriff auf die gerenderte Szene. Um zu gewährleisten, dass die komplette Textur verarbeitet wird, wird ein bildschirmfüllendes Rechteck gezeichnet, welches die Fragmentshaderausaufrufe "anwirft" (ein Shaderaufruf pro Texel). Die Textur-Auflösung entspricht in unserem Framework genau der Auflösung des mit `glViewport()` festgelegten OpenGL-Zeichenfensters.

- a) Schreiben Sie einen Fragmentshader, welcher einen Filterkernel K der Größe 3×3 auf die Textur T anwendet und die Fragmentfarbe entsprechend setzt, d.h.

$$\text{FragColor}(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 K(i, j)T(x - i, y - j) \quad (1)$$

Nehmen Sie `default.frag` als Gerüst für Ihren Shader. Der Kernel selbst soll in Form einer 3×3 -Matrix als Uniform-Variable an den Shader übergeben werden. Hierzu müssen Sie auch die Funktionen `FilterShader::initGLSL()` und `FilterShader::paint()` entsprechend erweitern.

- b) Erweitern Sie den Shader so, dass zwischen dem ursprünglichen und dem gefilterten Bild interpoliert werden kann. Verwenden Sie hierzu die vordefinierte Uniform-Variable `interpValue`.

Testen Sie folgende Kernel:

$$\frac{1}{4} \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Laplace

$$\frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

Gauss

Gerne dürfen Sie noch weitere (sinnvolle) Filterkernel ausprobieren.

Aufgabe 2 (Shader Bonusaufgabe, 3 Punkte)

Erweitern Sie Aufgabe 1 dahingehend, dass die Matrix, die den Filterkernel beschreibt, in der GUI des Programms eingegeben werden kann. Die Veränderung soll sich sofort auf die OpenGL-Szene auswirken.