

Sommersemester 2007

Übungen zu Computergraphik II - Blatt 5

Abgabe am Donnerstag, dem 19. 07. 2007, 13:00 Uhr

Aufgabe 1 (GLSL-Shader, 4+4+7 Punkte)

Implementieren Sie folgende GLSL-Shader:

- Erzeugen Sie mittels eines Vertexshaders einen Welleneffekt. Deformieren Sie dazu die Geometrie des Objekts in Abhängigkeit von der Zeit mittels `cos` und `sin`. Zum Testen des Shaders eignet sich besonders das Objekt `plane.obj`.
- Erzeugen Sie mittels eines Vertexshaders einen Morphingeffekt zwischen der Geometrie eines Objektes und einer Kugel. Interpolieren Sie dazu, in Abhängigkeit von der Zeit, die Position jedes Vertices zwischen dessen eigentlicher Position im Objekt und seiner Position projiziert auf eine Kugeloberfläche. Zum Testen des Shaders eignet sich besonders das Objekt `tesselatedcube.obj`.
- Implementieren Sie das aus der Vorlesung bekannte Bump-Mapping mittels GLSL-Shadern.

Sie können Ihre Shader entweder mittels des auf der Homepage bereitstehenden Frameworks, oder auch mit einer GLSL-Shader-IDE (RenderMonkey, GLSEditorSample,...) entwickeln.

Aufgabe 2 (GPGPU, 5 Punkte)

Ziel dieser Aufgabe ist die Berechnung eines Distanzfeldes auf der GPU. Gegeben ist ein diskretes Feld der Grösse $n \times m$ und eine Menge von Punkten und Geraden. In jeder Feldkoordinate steht der Abstand zu dem ihm am nächsten liegenden Punkt bzw. Linie. Um das Distanzfeld auf der GPU zu berechnen, werden die Punkte und Linien nacheinander an die Shader Programme übergeben, d.h. $\#Iterationen = \#Punkte + \#Linien$. Im Fragmentprogramm wird für jede Koordinate die Distanz zum aktuell übergebenen Objekt berechnet. Ist der Wert kleiner, wird der bisherige Distanzwert durch den neu berechneten ersetzt.

Implementieren Sie einen GLSL-Fragmentshader `distfield_point.frag` bzw. `distfield_line.frag`, welcher die Distanz der aktuellen Fragmentposition zu einem Punkt bzw. Gerade berechnet und gegebenenfalls updated. Das Distanzfeld wird durch eine Textur repräsentiert. Punkte und Linien werden über eine Uniform-Variable übergeben. Jedem Punkt/Linie ist eine Farbe zugewiesen, die ebenfalls durch eine Uniform-Variable übergeben wird. In der Alpha-Komponente der Textur soll die minimale Distanz und in den RGB-Komponenten die zugehörige Farbe gespeichert werden.

Sie können das auf der Homepage stehende Framework verwenden. Ein Gerüst für die beiden Fragmentshader ist enthalten, ebenso einfache Beispieldatensätze (Distanzfeldgrösse, Punkte/Linien-Menge) incl. Screenshots der resultierenden Distanzfelder.