

Prüfungsprotokoll Praktische Prüfung

Prüfer: Prof. Kobbelt

Fächer: CG1 / CG2 / Einführung in Datenbanken (Buchprüfung Kapitel 1-14)

Dauer: ca 55 Minuten

Datum: 22.10.05 Note: 1.3

E-mail: Maximilian.Moellers@rwth-aachen.de

Prof. Kobbelt: Gut, dann wollen wir mal anfangen. Erzählen sie mir doch mal was von der Rendering Pipeline. *Die Standardfrage schlechthin*

Ich: Aufgemalt und zu jedem Bereich ein paar Sätze verloren. *Hab zwar clipping vergessen, was mir ziemlich schnell aufgefallen ist, aber da Prof. Kobbelt sich nichts hat anmerken lassen, das dann nicht korrigiert. Erspart Fragen *G**

Prof. Kobbelt: Es gibt dann ja in den neueren OpenGL Versionen weitergehende Möglichkeiten, die nicht der normalen Rendering Pipeline folgen.

Ich: Ja, es gibt da ja Pixel / Vertex Shader die einzelne Berechnungsschritte komplett ersetzen. Vertex Shader machen halt das T&L, die Pixel Shader gehen nachher auf den Framebuffer ein.

Prof. Kobbelt: Gibt es denn noch andere Methoden Objekte zu rendern. z.b. wenn diese aus Laserscans kommen?

Ich: Jo Volume Rendering mittels direkt / indirekten Methoden. Hab diese aufgelistet und er wollte dann ziemlich genau Ray Casting erklärt bekommen, was also in diesem tollen Integral bzw. in der diskretisierten Form der Summe resultiert. Habe mich dann ziemlich beim B2F und F2B rendern verhaspelt, was sicherlich einer der Hauptgründe für die Note war.

Prof. Kobbelt: Übergang zu Meshdecimation :

Ich: jeweils 2-3 Sätze zu Resampling, Incremental Decimation, Vertex clustering (u.a. Fehlerquadriken erwähnt)

Prof. Kobbelt: Wie werden denn Fehlerquadriken berechnet?

Ich: Entweder Eigenvektor mit minimalem Eigenwert oder (*nach Hinweisen von Kobbelt*) in diesem Fall als Schnitt von mind 3 Ebenen (siehe $Ax = b$ im CG2 Skript).

Prof. Kobbelt: Diese wird man ja z.b. bei Incremental Decimation nicht in jedem neuen Schritt neu berechnen.

Ich: *na super.. Transferleistungen* Hmm. ja z.B. bei nem Vertex Contract könnte man die aus den ursprünglichen Knoten übernehmen.

Prof. Kobbelt: Genau. Und gibt es da irgendwas bei der Gewichtung zu beachten?

Ich: Hmmm. Die in der Mitte liegen ja näher an dem entstehenden Knoten und müssten somit stärker gewichtet sein.

Prof. Kobbelt: Begründung leider falsch, der Rest stimmt aber. Es werden einfach die Quadriken addiert und somit die in der Mitte liegenden Ebenen, welche ja an beiden Knoten liegen doppelt gewertet. Übergang zu Global Illumination. Wir hatten da ja so eine Gleichung:

Ich: Beide Versionen der Rendering Equation erklärt, wobei ich feststellen durfte, das ich eine davon einfach falsch gelernt hatte. *Ziemlich peinlich das Ganze :-/* Direkt dann die Begriffe Flux / Radiositiy / Radiance definiert.

Prof. Kobbelt: Wie kann man denn das nun lösen, wenn einen nur diffuse Reflexionen interessieren?

Ich: Classical Radiosity erklärt (Patches, $B_i = E_i + \dots$, Form Faktoren). Lösung durch nen iterativen Solver (Gathering). Iterationsformel hingeschrieben. Erwähnt, dass das funktioniert, da es eine von-Neumann-Reihe ist (was ja nichts anderes als ne geometrische Reihe für quadratische Matrizen ist), die zur gesuchten invertierten Matrix konvergiert. Alternativ Lösung durch Shooting, dies erklärt.. Nachteil bei beiden $O(n^2)$ Interaktionen, dafür Hierarchical Radiosity- i , $O(n)$ Interaktionen.

Prof. Kobbelt: OK nun zu DB. Was ist denn der Unterschied von objektorientierten und relationalen?

Ich: Entwicklung analog zu Programmiersprachen; das eine sind flache Tabellen, bei den anderer gewähren Klassenmethoden einheitlichen Zugang zu den Daten. Kapselung, Vererbung....

Prof. Kobbelt: Wie stellt man denn an so eine Datenbank die Anfragen?

Ich: SQL; beispiel statement.. zusammenhang zur relationalen Algebra.

Prof. Kobbelt: Wie modelliert man denn so Relationen?

Ich: ER-Modell anhand eines Beispiels, dazu noch die Anomalien anhand eines Beispiels. *ausnahmsweise keine Normalformen*

Prof. Kobbelt: Wie sehen denn die unterliegenden Datenstrukturen aus?

Ich: *zu blöd, dass die Datenstrukturen Vorlesung schon so lange her ist. Also einfach mal kreativ drauflosgeraten.* Je nach Anfragentyp Hashing oder z.b. B*-Bäumen.

Prof. Kobbelt: Wie funktioniert denn so ein B-Baum?

Ich: k bis 2k Kinderknoten, bei zu vielen rekursives Splitten bis zur wurzel, bla bla. B* hat dann den Vorteil, dass nur die Blätter die Daten enthalten, und somit die oberen lvl des Baums im Hauptspeicher gehalten werden können. Zusätzlich sind die Blätter über ne doppeltverkettete Liste verknüpft und es werden somit Bereichsanfragen erleichtert.

Prof. Kobbelt: Wie kann man das dann erweitern, dass man nach mehreren Kriterien sucht?.

Ich: keine Ahnung.

Prof. Kobbelt: Für jedes Kriterium einen Suchbaum, und dann auf dieselben Blätter zeigen. Warten sie dann bitte draussen.

Tja.. das waren dann also ca 250h lernen und trotzdem durch ein paar dumme Fehler (Ray Casting, RE) keine 1.0, wobei das auch gerechtfertigt ist. Prof Kobbelt kann man als Prüfer echt empfehlen, wenn da nicht das Fach wäre *G*. Es war eine extrem nette Atmosphäre, und es kam kein Stress auf. Wenn ich etwas nicht wusste, wurde mir einige Tipps gegeben, mit denen ich dann weiter kam oder er mir es dann erklärt hat. Datenbanken würde ich erst als letztes lernen, da es vom relevanten Umfang als auch von der Schwierigkeit doch deutlich unter CG1 / CG2 liegt. Viel Erfolg bei deiner Prüfung! Ich werd meinen Freiversuch nutzen!

p.s. es existieren noch diverse weitere CG1/CG2/GM1/GM2 Protokolle. Da ich jedoch nicht deren Autor ermitteln konnte, werde ich diese nicht veröffentlichen. Vielleicht kann man die ja von Kommilitonen bekommen.