

Datum: 7.4.2005, 15:30

Fächer (Prüfer):

Introduction to High performance Computing (Bischof)

Computational Differentiation (Bischof)

Computergrafik I (Kobbelt)

Virtual Reality (Lenka Jerabkova)

Prüfling: J.B.

Note: 1.0

----- Kobbelt: Computergrafik I-----

K: Extendet Coordinates - wofür?

Affine Abbildungen, Translation als lineare Abbildung nur möglich mit 4. Komponente

K: Wie sieht eine allgemeine Rotationsmatrix um X aus? Translationsmatrix?

=> Matrizen hingeschrieben ( $M_R$ ,  $M_T$ )

K: Translations- / Rotationsmatrix zusammengemerged in eine Matrix M, was wird zuerst ausgeführt, Translation oder Rotation?

=> Überleg, hm M ist ja gleich  $M_R * M_T$ , also zuerst die Translation

K: FALSCH! Überlegen Sie noch einmal

=> okay, sorry geirrt, also erst die Rotation. Allgemein kann man ja verschiedene Abbildungen kombinieren in einer Matrix, indem man die entsprechenden Matrizen in der richtigen Reihenfolge miteinander multipliziert

K: Okay, richtig :-). Wie wird diffuse Reflektion berechnet? Schreiben Sie bitte die Formel auf

=> Bild gezeichnet + Formel dazugeschrieben

K: Wie funktioniert denn ihr Lieblingsverfahren zur Schattensimulation?

=> Shadow Maps erklärt, perspektivische und projektive Aliasfehler genannt, perspektivische Shadow Map erklärt (NDC)

K: Wie funktionieren Shadow Volumes?

=> Idee erklärt, Feststellen ob ein Punkt im Schatten liegt über Strahl ins Unendliche, läßt sich in OpenGL mit Stencil Buffer erledigen

K: Welches Verfahren ist besser in einer Szene mit bewegter Lichtquelle, z.B. eine Kerze?

=> Shadow Maps haben ja die Eigenschaft, aus Sicht der Lichtquelle zu rendern, was bei einer Kerze ja schwierig ist (strahlt in alle Richtungen), Shadow Volumes müssen dafür bei Veränderung der Lichtquelle neu berechnet werden.

K: Läßt sich die Berechnung der Shadow-Volumes nicht irgendwie auf der Hardware durchführen?

=> Klar (hatte Mario in der Übung mal erwähnt): Jeden Punkt und damit auch jede Kante verdoppeln, Vertex Shader Programm guckt dann ob die Kante eine schattenwerfende Kante ist (Normalenvektoren) und zieht sie aus Sicht der Lichtquelle ins Unendliche

K: Wie kann man allgemein Volumen visualisieren?

=> es gibt direkte und indirekte Verfahren, direkt ist z.B. Ray Casting. Dazu wollte er dann die Gleichung sehen (stetig und diskret)

K: Wie würde man für einen Strahl anhand dieser Gleichung die akkumulierten Farbwerte berechnen?

=> Beide Richtungen möglich (von vorn nach hinten oder von hinten nach vorn), im ersten Modus muß man dann in jedem Schritt die Farbe und die akkumulierten Opazitäten updaten

K: Was ist besser bzw. wo ist der Unterschied?

=> Vorn nach hinten, denn wenn die Opazität schon einen bestimmten Wert erreicht hat sieht man dahinter eh nichts mehr und kann frühzeitig abbrechen

K: Kennen Sie ein weiteres Verfahren zur Volumen-Visualisierung, z.B. auf der Hardware?

=> Volume-Slicing: 3D-Textur des Volumens in Grafikkarte laden, Alpha-Blending zur Darstellung des gesamten Volumens

K: Wie rasterisiert man Linien? Und was bedeutet eigentlich rasterisieren?

=> Rasterisieren = Die Linie auf einem diskreten Pixelgitter zeichnen, Bresenham erläutert: Idee, Update der Decision-Variable, 2 Additionen und 1 Multiplikation pro Pixel

----- ab jetzt Bischof: Computational Differentiation -----

B: Schreiben Sie mir mal den Adjoint-Code für dieses Beispiel auf ( $Z = X * Y$ ) und für dieses ( $Z = Z * Y$ )

=> hingeschrieben, für den komplizierten Fall hab ich noch mal den hypothetischen Code aufgeschrieben um sicher zu gehen

B: Und hier noch ein Beispiel (loop i:  $Z = Z * X(i)$ )

=> hingeschrieben, aber dabei was vergessen ( loop i rückwärts:  $X(i) += Z * AdZ$ ,  $AdZ = AdZ * X(i)$  )

B: Na ja das funktioniert ja so erst mal nicht. Was fehlt?

=> Äh, keine Ahnung... Prof. Bischof hat dann etwas mit dem Zaunpfahl gewinkt... Ach ja stimmt, die Werte von Z werden ja überschrieben. Also die verschiedenen Verfahren erklärt wie man das in Griff kriegt: Store and Restore, Storage Expansion, Recompute

B: Und was sind da die jeweiligen Probleme?

=> Storage Expansion erweitert Z um eine Dimension => Speicherbedarf, Recompute keinen zusätzlichen Speicherbedarf, aber dafür Laufzeitbedarf

B: Wie sähe das Recomputing in diesem Beispiel aus?

=> In die loop i-Schleife noch eine loop j-Schleife eingebaut, in der das jeweilige Z neu berechnet wird und ein bisschen zu Initialisierung. erzählt

B: Wie kann man bei einer dünnbesetzten Jacobi-Matrix seine Vorteile rausschlagen?

=> Compressed Jacobian erläutert (er hatte ein Beispiel hingemalt), Seeding selektiert dabei die Spalten (FM) bzw. Zeilen (RM) die man "zusammenfassen" kann. Coloring erwähnt zum Finden der orthogonalen Spalten.

B: Wie sähe die Seed-Matrix für dieses Beispiel aus?

=> So daß die erste Spalte erhalten bleibt und Spalten 2 – 5 aufeinander addiert werden. Dann die Seed-matrix hingeschrieben, okay

B: Und was macht man in einem solchen Fall? (Beispiel aus Predator-Prey-System hingemalt)

=> Splitting: Einmal zusammenfassen, dann die verlorenen Informationen mit einer weiteren Seed-Matrix sichern. War nicht so wirklich sauber, da hat er mir aber gut bei weitergeholfen

----- Introduction to High Performance Computing -----

B: Hypercube: Wie funktioniert Broadcast?

=> Spanning tree Broadcast, erst in Dimension 0, dann Dimension 1... hab ich ihm direkt anhand Beispiel hingemalt

B: Wie bettet man Ring in Hypercube ein?

=> Crey-Code erklärt, hingeschrieben, daneben den Ring gemalt, klar

B: Wie sieht ein Omega-Netzwerk aus? Zeichnen Sie mal für 8 Prozis. Wie müssen die Stufen geschaltet werden?

=> Erklärt (zyklischer Linksshift mit evtl. Invertierung des letzten Bits), gezeichnet, Startadresse XOR (bitweise) Zieladresse gibt die Stufen, bei denen Crossover geschaltet werden muß, die anderen Stufen werden durchgeschaltet

B: Wie kann man denn eine fette Jacobi-Matrix auf mehreren Prozessoren zerlegen?

=> Partitionierung, allgemeinen Fall aufgemalt. Die Diagonalmatrizen können parallel Cholesky-zerlegt werden, die Blöcke in der letzten Zeile auch (NACH den Diagonalmatrizen), der Block unten rechts ist serieller Engpass

B: Wie findet man eine Partitionierung?

=> Knotenseparator

B: Okay. Wie macht man aus einem Kantenseparator einen Knotenseparator?

=> Überleg (hatten wir nicht in der Vorlesung), zwischen den Kanten Knoten einfügen?

B: Nee, wäre ja Verschwendung. Hat dann ein bisschen herumgemalt und die Ecken von einer Partition in die andere dick gezeichnet. Schauen sie sich mal die Kanten an. Idee?  
=> Aaaaah, klar man könnte einfach die Endpunkte der Kanten zwischen den Partitionen als den Knotenseparator nehmen.

B: Richtig! Wie funktioniert denn die spektrale Methode?  
=> Laplace-Matrix erwähnt, den Lösungsvektor  $x$  beschrieben, der angibt, welcher Knoten in welcher Partition landet.  $x^T * L * x$  entspricht dann den Kosten des Schnitts (  $4 * Cut$  )

B: Wie löst findet man dazu die optimale Lösung?  
=> Relaxierung, Fiedler Vektor (Eigenvektor zum 2.-kleinsten Eigenwert), mittels Median aus dem Fiedler-Vektor eine gültige Lösung machen

B: Warum nicht den Eigenvektor zum kleinsten Eigenwert?  
=> Das wäre der triviale Fall: Alle Knoten in eine Partition

-----Ab jetzt Lenka Jerabkova: Virtual Reality -----

J: Betrachterzentrierte Projektion - wofür?  
=> korrekte Perspektive, ermöglicht durch Tracking

J: Wieso Frustrum-Abbildung in den Einheitswürfel? Wie ist der erste Schritt um von einer evtl. Schrägprojektion

J: Wofür und wie funktioniert Tracking?  
=> Optisch, Elektromagnetisch. Benötigt um die korrekte Augenposition zu kennen

J: Was passiert wenn zwei Menschen in der Cave sind, aber nur einer getrackt wird?  
=> Habe dann eine Zeichnung bekommen, die schon in der Übung mal drankam. Der getrackte Betrachter sieht das Objekt dort, wo es auch sein soll, der ungetrackte sieht es am falschen Ort.

J: Wo denn?  
=> Habe dann ein bisschen an der Skizze herumgemalt, war aber falsch. Dann hab ich mich irgendwie herausgeredet, immer noch falsch.

J: Zeichnen Sie die Skizze doch einmal neu  
=> Neu gezeichnet, dabei ist es mir dann klar geworden. Der ungetrackte Betrachter sieht das Objekt dort, wo der Strahl von der (falschen) Projektion zum linken und zum rechten Auge sich schneiden

J: Richtig! Was ist denn Haptik und wie wird es in der VR eingesetzt?  
=> Kurz erklärt, Phantom erwähnt

J: Wie rendert man eine Wand haptisch?  
=> Prinzip der Gegenkraft erwähnt, die proportional zur Eintauchtiefe in der Wand ist. Schwierig ist es dabei, die Skalierung zu finden: Wenn die Feder zu weich ist, fühlt sich die Wand nicht hart an; ist die Feder zu hart, wird man quasi aus der Wand herauskatapultiert.

J: Was können hierbei für Schwierigkeiten auftauchen, zum Beispiel abhängig von der Form des zu rendernen Objektes?  
=> Sehr dünne Wand: Man berührt z.B. einen Tisch oben und wird nach unten herausgezogen. Lösung: Virtueller Proxy, der speichert, von wo man in die Wand hereingekommen ist und die Gegenkraft entsprechend lenkt.

Zusammenfassung: Die Prüfung verlief sehr angenehm, bei einigen Aussetzern haben die Prüfer mich wieder auf die richtige Fährte gebracht und gemerkt, daß es keine Verständnislücken waren. Ich habe etwa 7 Wochen lang für die Prüfung gelernt. HPC und CD ausschließlich anhand des Vorlesungsmaterials, CGI mit dem Skript von Sandip. Über Google findet man nette Applets, die z.B. den Bresenham-Algorithmus oder Polygon-Clipping veranschaulichen. Viel Erfolg bei der Prüfung!