

Abhängigkeit der Graphikleistung von der Systemleistung bei Computern unter Windows NT

Stephan Würmlin

Departement Informatik, ETH Zürich

wuermlin@vis.ethz.ch

Projektarbeit für die Vorlesung “Computer Systems Performance Analysis and Benchmarking”

Zusammenfassung

Dieser Bericht beschreibt den Zusammenhang der Systemleistung und der Graphikleistung, anhand von zwei Dell Computern unter Windows NT.

Die Systemleistung beinhaltet folgende Komponenten: Prozessor, Speicher und Bus. Diese werden mit dem SPEC CPU95 Benchmark Programm gemessen.

Die Graphikleistungsmessungen werden ausschliesslich mit der OpenGL API durchgeführt. Es wird Viewperf, ein Anwendungsbenchmark Programm, verwendet. Viewperf misst die reale Anwendungsleistung. Zusätzlich wird die rohe, optimale Graphikleistung mit dem GLperf Benchmark Programm gemessen. Unser Interesse galt dabei den Pixel- und Dreieckfüllraten.

1. Einleitung

Windows PC's mit billigen 3D Beschleunigern scheinen professionellen Graphikarbeitsstationen von Silicon Graphics in punkto Leistung immer näher zu kommen. Doch sind diese Leistungen nur bis zu einem gewissen Komplexitätsgrad möglich. Dies hat einfache Gründe. Die ganzen Geometrieberechnungen, die in einem SGI System von einem eigens dafür gemachten Chip durchgeführt wird, werden bei herkömmlichen Windows PC's vom Hauptprozessor durchgeführt, was diesen vollständig auslastet.

In dieser Arbeit soll untersucht werden, wann Komplexitätsgrenzen erreicht werden und wie der Zusammenhang zwischen der Systemleistung (Busbandbreite, Prozessor, Speicher) und der Graphikleistung ist.

Es werden keine Messungen mit der Direct3D Programmierungsschnittstelle durchgeführt.

2. Computer Konfigurationen

Es werden zwei Computersysteme untersucht. Das eine ist ein Dell Optiplex Pro System, das andere ein Dell Precision 410 System. Wir werden von der ersten Maschine als *Matisse*, und von der zweiten als *Capilla* sprechen.

Beide Maschinen unterstützen MMX, die Multimedia Erweiterung von Intel, und besitzen ähnliche PCI Bus Architekturen, mit 32 bit Hauptspeicher Adressierung und einen 64 bit breiten Datenbus.

System	<i>Matisse</i>	<i>Capilla</i>
Prozessor	Intel Pentium Pro	Intel Pentium II
Taktrate	180 MHz	400 Mhz
Level 1 Cache	16 KByte	32 KByte
Level 2 Cache	256 KByte	512 KByte
Hauptspeicher	128 MByte	64 MByte
Betriebssystem	Windows NT 4 SP3	Windows NT 4 SP3

Table 1: Computer Konfigurationen

Capilla besitzt eine doppelt so hohe Taktrate und doppelt so viel Level 1 und 2 Cache als Matisse, und ein Pentium II Prozessor anstelle eines älteren Pentium Pro Prozessors.

3. Graphikkarte

In beiden Maschinen ist eine FireGL 1000 Pro Karte von Diamond Multimedia Systems mit dem Permedia 2 Chip von 3Dlabs eingebaut. Diese Karte unterstützt sowohl die Direct3D Graphik API als auch die OpenGL API.

Weitere Details finden sich in Tabelle 2.

Zu bemerken ist, dass der Permedia 2 Chip keine trilineare Filterung unterstützt.

Die FireGL 1000 Pro besitzt einen integrierten Geometrie Pipeline Prozessor, den Delta 3D von 3Dlabs. Dieser besitzt folgende Fähigkeiten: “Single-Pixel” Divisionsfunktion für schnellere perspektivische Korrektur, 8-bit Farben Texturpalette, welche die Speicheranforderungen der Texturkompression reduzieren und schnellere Z-Buffering Modi verbessern auch die 3D Leistung.

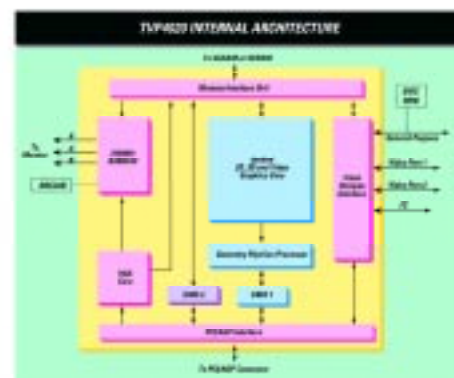


Figure 1: Blockdiagramm TI TVP4020

<i>Diamond FireGL 1000 Pro</i>	
Graphikprozessor	3Dlabs Permedia 2
Graphikcontroller	Texas Instruments TVP 4020
Architektur	64 bit
Bus	PCI/AGP
Video RAM	8 MByte DRAM (dynamische Teilung in Frame und Textur Speicher)
3D Operationen	Gouraud & Flat Shading 16 bit Z-Buffer Alpha Blending Bilinear Filtering Hardware Texture Mapping
RAMDAC Speed	230 MHz
Chip Clock Speed	83 MHz
Bandbreite	664 MByte/s
AGP Features	AGP 1x (266MByte/s) Full Sideband Addressing DMA Mode Texturing
Geometry Pipeline Processor	3Dlabs Delta 3D
Peak Rates (published)	1 Mio. Polygons/s 80 Mio. Pixels/s
Bios Version	Matisse: 1.54 / Capilla: 1.49
Driver Version	4.00.1381.0277

Table 2: Diamond FireGL 1000 Pro

4. Der SPEC CPU Benchmark

Der SPEC CPU95 Benchmark, misst die Leistung des Prozessors, der Speicherhierarchie und des Compilers eines Computersystems.

CPU95 seinerseits benutzt die CINT95 Tests mit acht Programmen, um die Festkommaeinheit, und die CFP95 Tests mit zehn Programmen, um die Gleitkommaeinheit

zu messen. CPU95 ist nicht dazu konstruiert, andere Computersystemkomponenten zu messen, weshalb es sich gut dazu eignet, den Einfluss der Systemleistung auf die Graphikleistung zu betrachten.

Das "C" im Namen von CINT95 und CFP95 steht für Component, und steht dafür, dass nicht ein System getestet wird (das würde mit einem "S" angegeben), sondern es handelt sich dabei um ein Komponenten Benchmark Programm.

SPEC CPU95 wird von der Open Systems Group (OSG) der SPEC entwickelt. Die '95 Version ist das zweite Update des Benchmarks. SPEC CPU89, das Original, wurde 1989 entwickelt.

SPEC95 wurde entwickelt, um vergleichbare Messungen von Computersystemleistungen mit einem gut bekannten, rechenintensiven Workload durchzuführen. Um dies auch auf verschiedenen Plattformen durchführen zu können, entschied sich die OSG den Benchmark als Quellcode zu unterhalten. Obwohl das Benchmark Programm oft nur als Prozessorbenchmark bezeichnet wird, misst es eigentlich folgende drei Komponenten eines Computersystems:

- Prozessor
- Speicher Hierarchie
- Compiler

Das Programm wurde nicht entwickelt, um andere Komponenten zu testen, wie zum Beispiel Graphik, Netzwerk, I/O oder Fähigkeiten des Betriebssystems.

SPEC CPU95 ist für viele Betriebssysteme erhältlich, unter anderem für viele UNIX Versionen und Windows NT.

Benchmark	Anwendungsbereich	Spezielle Aufgabe
099.go	Spiele, Künstliche Intelligenz	Spielt das "Go" Spiel gegen sich selber.
124.m88ksim	Simulation	Simuliert den Motorola 88100 Prozessor und lässt Dhystone und ein Memory Test Programm laufen.
126.gcc	Programmierung, Compilierung	Compiliert einen vorbereiteten Quellcode in optimierten SPARC Assembler Code.
129.compress	Kompression	Textdateikompression unter Benutzung der Lempel-Ziv Codierung.
130.li	Sprachinterpreter	Lisp Interpreter.
132.jpeg	Bildbearbeitung	JPEG Bildkomprimierung mit verschiedenen Parametern.
134.perl	Shell Interpreter	Text- und numerische Manipulationen.
147.vortex	Datenbank	Errichtet und manipuliert drei relationale Datenbanken.

Table 3: CINT95 Benchmarks

Benchmark	Anwendungsbereich	Spezielle Aufgabe
101.tomcatv	Flüssigkeitsdynamik, Geometrische Translation	Generiert ein zweidimensionales Koordinatensystem um generelle geometrische Bereiche.
102.swim	Wettervorhersage	Löst flache Wassergleichungen mit Hilfe von Approximationen mit finiten Differenzen.
103.su2cor	Quantenphysik	Quark-Gluon Theorie, Berechnung von Massen elementarer Teilchen.
104.hydro2d	Astrophysik	Hydrodynamische Navier-Stokes Gleichungen.
107.mgrid	Elektromagnetismus	Berechnung eines 3D Potentialfeldes.
110.applu	Flüssigkeitsdynamik, Mathem.	Löst ein Matrix System mit Pivot-Strategien.
125.turb3d	Simulation	Simuliert Turbulenzen in einem kubischen Bereich.
141.apsi	Wettervorhersage	Berechnet Statistiken über Temperaturen und Schadstoffe in einem Bereich.
145.fpppp	Chemie	Berechnet Ableitungen von Multielektronen.
146.wave5	Elektromagnetismus	Löst Maxwell'sche Gleichungen in einem kartesischen Netz.

Table 4: CFP95 Benchmarks

5. Der Viewperf Benchmark

Viewperf ist ein portables OpenGL Performance Benchmark Programm, welches in C geschrieben wurde. Ursprünglich wurde es von IBM entwickelt, erweitert wurde es dann von SGI und Digital.

Viewperf ist sehr flexibel zum Evaluieren der OpenGL Performance von Computer Systemen. Derzeit gibt es Implementationen für diverse Plattformen, zum Beispiel für UNIX, Linux, Windows NT, Windows 95 und OS/2.

Viewperf wurde 1994 in die Standard Performance Evaluation Corporation (SPEC) eingegliedert, welche sich zur Aufgabe macht, Benchmarks für Computer zu unterhalten. Die OpenGL Performance Characterization (OPC) Projektgruppe der SPEC publizierte Viewperf als ersten OpenGL Benchmark der Graphics Performance Characterization (GPC) Gruppe.

Die OPC Gruppe unterhält den Quellcode von Viewperf, der für alle über Internet erhältlich ist.

Viewperf misst die 3D Rendering Performance eines Systems. Dazu wird die OpenGL API benutzt. Die OPC Gruppe arbeitet mit unabhängigen Software-Entwicklern (ISV: engl. Independent Software Vendors) zusammen, um Tests, Datensätze und Gewichte zusammenzustellen, welche Viewsets genannt werden. Jedes Viewset repräsentiert den Graphik Rendering Mix einer aktuellen, real existierenden Anwendung. Die ISVs, welche die Viewsets entwickelt haben, bestimmen für jeden Test, der im Bericht aufgeführt wird, ein Gewicht, das die Wichtigkeit des Tests in der ganzen Anwendung wiedergibt.

Viewperf liegt auf der Homepage der OpenGL Performance Characterization Group zum Download bereit. Der Quellcode ist dazu da, Leistungsvergleiche diverser Hardware Plattformen anzustellen. Viewperf ist lauffähig auf den gängigsten Prozessoren, so zum Beispiel Alpha, Intel, MIPS und PowerPC. Window Umgebungen werden sowohl X als auch Windows unterstützt.

Die Zahlen, welche die Performance angeben, werden in der Einheit "Bilder pro Sekunde" angegeben, was für den Benutzer einfach nachzuvollziehen ist. Für jeden Rendering Test in einem Viewset wird eine Zahl ausgegeben.

Viewperf bewertet eine grosse Anzahl von OpenGL Befehlen, Operationen und Modi. Unter anderem kann Texture Mapping, Alpha Blending, Fogging, Anti-Aliasing, Depth Buffering und Lighting bewertet werden. Das Zeichnen der folgenden OpenGL Primitiven kann gemessen werden: points, lines, line_strips, line_loops, triangles, triangle_strips, triangle_fans, quads und polygons.

In Viewperf können Bildschirmschnapschüsse aufgenommen werden. Diese werden im PNG oder PPM Format abgespeichert. Die Schnapschüsse dienen der Qualitätsanalyse und der Verifikation des Benchmarks.

Viewsets kann jeder entwickeln. Da Viewperf anwendungsorientiert ist, muss man zuerst die vorhandene Anwendung analysieren. Daraus muss man Datensätze zusammenstellen, die repräsentativ für die Anwendung sind und ähnliche Komplexität aufweisen. Wie man ein Viewset genau zusammenstellt, kann man auf der Homepage der OPC Gruppe nachlesen.

Für Tests stellt die OPC Gruppe sechs Viewsets zur Verfügung. Diese sind in Tabelle 3 beschrieben.

Name	Vendor	Description	Based on	# of Tests
ProCDRS	Parametric Technology (PTC)	Modeling und Rendering Application for Computer-Aided Industrial Design (CAID)	Pro/DESIGNER	7
DX	Industrial Business Machines (IBM)	Scientific Data Visualization and Analysis Package	Visualization Data Explorer	10
DRV	Intergraph	3D Computer Model Review Package for plant design models	Design Review	10
AWadvs	Alias/Wavefront	Integrated workstation-based 3D animation system that offers a comprehensive set of tools for 3D modeling, animation, rendering, image composition, and video output	Advanced Visualizer	10
Light	Lightscape Technology Inc.	Radiosity Visualization Application with a physically based lighting interface	Lightscape Visualization System	4
CDRS	Parametric Technology (PTC)	Modeling und Rendering Application for Computer-Aided Industrial Design (CAID)	CDRS, originally created by Evans and Sutherland	7

Table 5: Viewperf Viewsets

6. Der GLperf Benchmark

GLperf ist ein Programm, welches die Leistung von OpenGL Graphikoperationen misst. Diese Operationen werden auf "Low-Level" Primitiven, wie Punkten, Linien, Polygonen, Pixeln etc. angewandt und beinhalten keine ganzen Modelle wie in Viewperf.

Mit GLperf versucht man einerseits die Methode zu standardisieren, mit welcher Computerverkäufer die Leistung von OpenGL Graphikoperationen auf ihren Systemen angeben. Andererseits ermöglicht GLperf einen umfassenden Einblick in die Graphikleistung eines Systems.

GLperf wurde hauptsächlich von John Spitzer von Silicon

Graphics entwickelt und ist jetzt wie Viewperf in die OpenGL Performance Characterization Gruppe der SPEC integriert.

GLperf ermöglicht die explizite Eingabe der OpenGL Zustände und der Datenart, die das Graphiksystem zu bewältigen hat. Das Ziel von GLperf ist es möglichst grosse Flexibilität in der Leistungsmessung über einen grossen Bereich von Szenarien und Modi zu erreichen.

GLperf dient als Serviceprogramm. Die vom Benutzer gewünschten Zustände und Tests werden in einem Skript gespeichert, die dann von GLperf analysiert und ausgeführt werden.

Die Skripts erlauben dem Benutzer die Veränderung der

meisten Attribute, die die Leistung verändern können. Dies beinhaltet die meisten OpenGL Zustandsvariablen (e.g. Depth Functions, Alpha Functions, Fogging Functions, Texture Options, Polygon Mode, Number Of Lights, etc.).

Zusätzlich kann man folgendes spezifizieren:

- den Ablaufmodus (immediate oder in einer Display List (compile, execute, compile_and_execute, list deletion))
- den Farbmodus (RGB α oder als Index in einer Look-Up Table)
- die Grösse der Primitive (e.g. point, line, polygon) in Anzahl Pixel
- die Form der Primitive (das Verhältnis für Polygone)
- die Orientierung der Primitive (horizontal, vertikal oder zufällig)
- den Prozentsatz der Front-Facing beziehungsweise der Back-Facing Primitiven
- den Prozentsatz der trivial akzeptierten, geclipten und zurückgewiesenen Primitiven
- den Typ der Projektionsmatrix (perspektivisch oder parallel)
- die Datenmenge, die für jede Primitive zur Verfügung gestellt wird (color data (none, per facet, per vertex), normal data (none, per facet, per vertex), texture data (none, per vertex))

Obwohl sowohl Viewperf wie auch GLperf Graphikleistung messen, wurden sie doch für zwei verschiedene Ziele entwickelt.

Während Viewperf ein ganzes Modell mit unterschiedlichen Primitivengrössen zeichnet, welche den typischen Grössen in richtigen Anwendungen entsprechen, weist GLperf künstlich jeder Primitive eine spezielle Grösse zu. GLperf ist daher ein synthetisches Benchmark Programm. Während Viewperf den Rendering Mix einer Anwendung simuliert und die Leistung misst, offeriert GLperf eine viel kontrollierbare Konfiguration zum Messen der höchsten Leistung eines Systems.

Ein weiterer Unterschied besteht in der Ausgabe der Resultate. Viewperf Tests resultieren in einer Angabe in Bilder pro Sekunde, während GLperf Tests eine Messung in Anzahl gezeichneter Primitiven pro Sekunde ergeben. Primitiven sind dabei zum Beispiel Punkte, Linien, Pixel, Dreiecke etc.

Um repräsentative Resultate von GLperf zu erhalten, hat die OPC Gruppe 13 Skripte veröffentlicht, welche die interessantesten Bereiche testen. Sie sind aufgeteilt in 10 Skripte im RGB-Farbraum und 3 Skripte im indiziertem Farbraum.

Das Interesse galt den Pixelfüllraten und den Dreieckfüllraten. Aus diesem Grund werden nur zwei der dreizehn Skripts benutzt:

FillRate.rgb

Dieses Skript misst wieviele Pixel pro Sekunde gezeichnet werden können

Die Tests werden im "Display List" Modus gezeichnet mit 500×500 Pixel grossen Quadraten.

TriFill.rgb

Dieses Skript misst den Einfluss der Vergrößerung der Primitive auf das Zeichnen von Dreiecken.

Die Tests werden mit Dreiecksstreifen durchgeführt.

Das Postfix des Skriptnamens gibt die jeweilige Zugehörigkeit zum Farbraum an.

7. Analyse der Daten

SPEC CPU95.

Zuerst zu den SPECmarks. Der SPEC Benchmark wurde nur für Capilla durchgeführt. Dies ist die fettgedruckte Zahl in der Tabelle 3.

Die anderen Zahlen wurden von der SPEC Homepage genommen, da es nur darum geht einen Überblick über die Systemleistung zu erhalten. Für die Analyse wurden aber die publizierten Daten verwendet.

System	<i>Matisse</i>	<i>Capilla</i>	<i>C/M</i>
CINT_base95	7.28	11.6 (15.3)	2.10
CFP_base95	4.99	11.2	2.24

Table 6: SPECmarks

Die Verhältnisse C/M, C bezeichnet Capilla, M Matisse, bezeichnen den Speedup von Capilla gegenüber Matisse. Das Verhältnis der Taktraten von Capilla gegenüber Matisse ist $400/180 = 2.22$. Die Verhältnisse der SPECmarks reflektieren in etwa das Taktratenverhältnis.

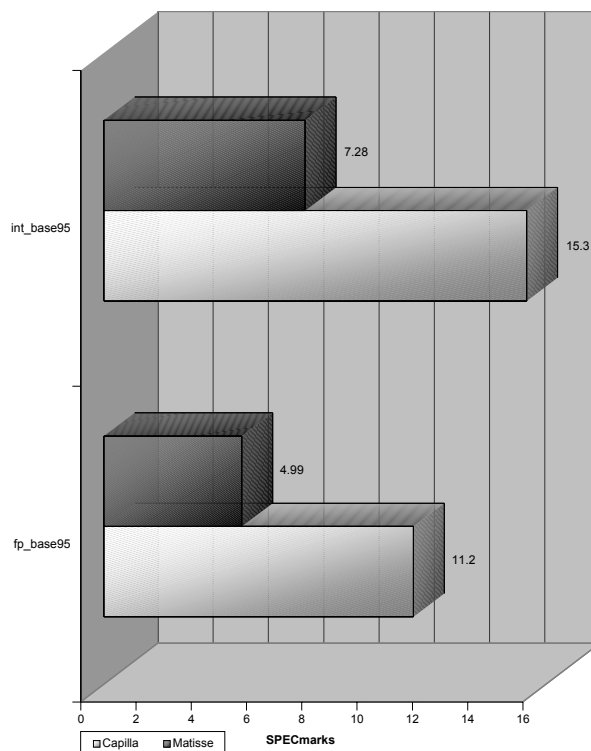


Figure 2: SPECmarks

Viewperf.

Replikation. Die Tests der Viewsets wurden bei beiden Maschinen fünf mal laufen gelassen. Der ermittelte Wert für ein Test entspricht dem arithmetischen Mittel aus den Einzeltests.

Netzwerk. Die Maschinen waren beide am Netzwerk. Der Netzwerk Overhead ist nicht berücksichtigt worden.

Benutzermodus. Die Tests wurden im "Multi User Mode" durchgeführt.

Bildschirmauflösung. Die Auflösung war bei beiden Maschinen 1280 × 1024 Pixel bei 16 bit Farbtiefe.

Die Viewperf Resultate sind in Tabelle 4 festgehalten.

System	<i>Matisse</i>	<i>Capilla</i>	<i>C/M</i>
ProCDRS	1.926	5.556	2.89
DX	3.886	7.619	1.96
DRV	2.360	4.710	2.00
AWadvs	4.801	10.213	2.13
Light	0.4675	1.010	2.16
CDRS	14.92	36.05	2.42

Table 7: Viewperf Resultate

Die Verhältnisse liegen zwischen 1.96 und 2.89, der Mittelwert liegt bei 2.26. Dieser entspricht etwa dem C/M Verhältnis der Floating-Point SPECmarks.

Dies zeigt, dass die Graphikleistung extrem vom Prozessor abhängt, da er die Geometrieberechnungen durchführt. Die Verhältnisse variieren, jenachdem wie geometrieintensiv die Berechnungen sind.

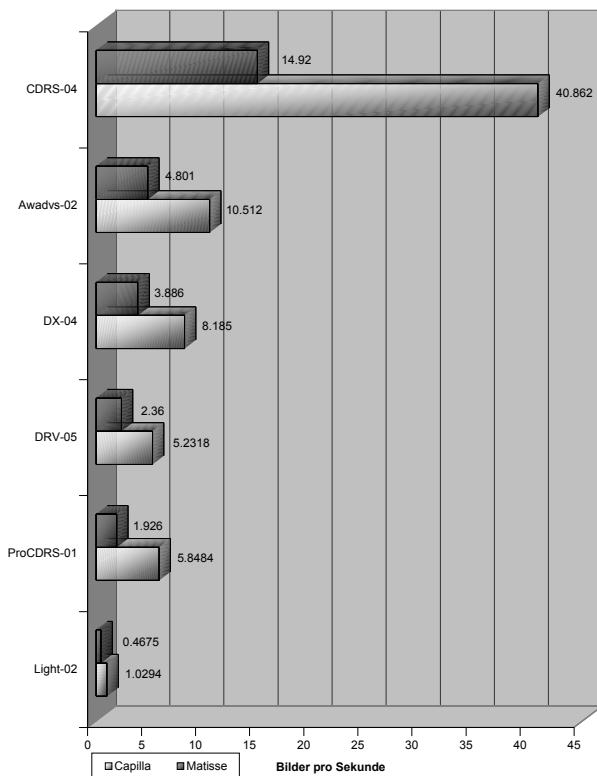


Figure 3: Viewperf Resultate

GLperf.

Replikation. Die Tests wurden, bedingt durch die lange Laufzeit, bei beiden Maschinen nur einmal laufen gelassen.

Netzwerk. Der Netzwerk Overhead ist wiederum nicht berücksichtigt worden.

Benutzermodus. Die Tests wurden ebenfalls im "Multi User Mode" durchgeführt.

Bildschirmauflösung. Die Auflösung war bei beiden Maschinen 1280 × 1024 Pixel bei 16 bit Farbtiefe.

Pixelfüllraten. Alle Ergebnisse in Millionen Pixel pro Sekunde - Display List, 500 × 500 Pixel grosse Quadrate, RGB Farbmodus. Die Textur ist 64 × 64 Pixel gross und im RGB Farbmodus.

System	<i>Matisse</i>	<i>Capilla</i>	<i>C/M</i>
flat	30.3	62.3	2.06
flat, Z	46.4	77.6	1.67
smooth	79.2	81.9	1.03
smooth, Z	46.4	77.6	1.67
smooth, Z, nearest neighbor	24.37	39.4	1.62
smooth, Z, linear	17.61	25.4	1.44
smooth, Z, trilinear	7.577	13.61	1.80

Table 8: GLperf Resultate: **Pixelfüllraten.** flat bezeichnet Flat Shading, Z: eingeschaltetes Z-Buffering, smooth: Smooth Shading, nearest neighbor: Textur Filterung mit "Nearest Neighbor" Algorithmus, linear: Textur Filterung mit "Linear" Algorithmus, trilinear: Textur Filterung mit "Trilinear" Algorithmus

Interessant sind hier vor allem die "Smooth Shaded"-Werte. Wie man sieht ist der Unterschied vernachlässigbar, was zeigt, dass hier die Graphikkarte (bzw. die Raster Engine der Graphikkarte) der limitierende Faktor ist.

Ausschliesslich beim Test im "Flat Shaded" Modus mit ausgeschaltetem Z-Buffering sind die Geometrieberechnungen des Hauptprozessors ausschlaggebend.

Die Werte mit der tilinearen Filterung sind interessant. Laut 3Dlabs beherrscht der Permedia 2 Chip diesen Modus nicht. Eigentlich gäbe es da zwei Lösungen. Einerseits könnte der Chip die Berechnungen an den Hauptprozessor abgeben oder andererseits einen einfacheren Modus durchführen. Dies scheint auch der Fall zu sein. Meiner Meinung nach führt der Chip bilineare Filterung durch, was der Chip auch beherrscht. Dies führt jedoch zu einer Beeinträchtigung der Bildqualität.

Wie man sieht, entspricht der Test im Smooth Shaded Mode etwa den publizierten Höchstleistungswerten (Tabelle 2).

Dreieckfüllraten. Alle Ergebnisse in Dreiecke pro Sekunde - Display List, Triangle Strip, RGB Farbmodus, Flat Shaded, Z-Buffered.

Wie man an den Werten merkt, ist die Leistung am Anfang stark von der Rasterung, also von der Graphikkarte, abhängig. Die Werte liegen nahe beieinander.

Mit zunehmender Komplexität (Anzahl Pixel pro Dreieck) scheint wieder die Geometrieberechnungen eine Rolle zu spielen.

Interessant ist, dass bei etwa 12 Pixeln pro Dreieck ein Umschwung bemerkbar ist. Capilla kann die Dreieckfüllrate länger halten und Matisse erleidet einen Einbruch. Das zeigt klar, dass da der Prozessor ins Spiel kommt.

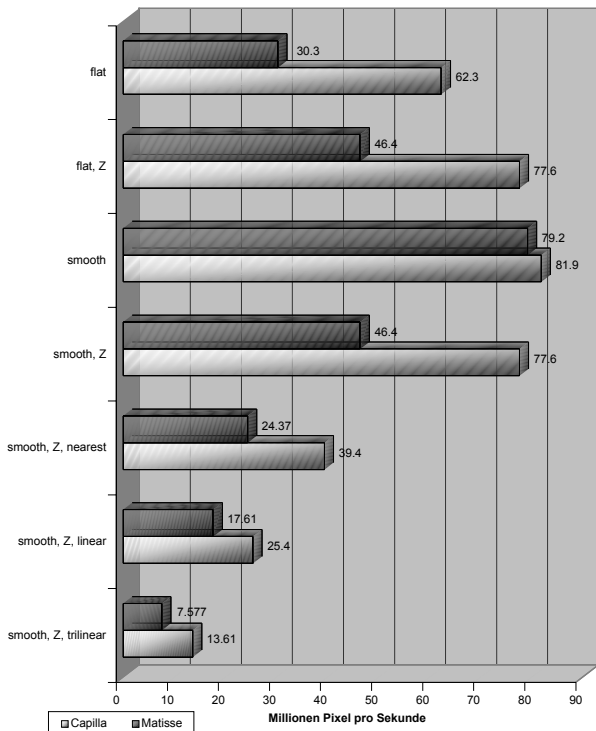


Figure 4: Pixelfüllraten

Der Einbruch ist bei ca. 9 Pixel pro Dreieck für Matisse. Capilla's Einbruch ist bei ca. 13 Pixel pro Dreieck. Der Einbruch ist der Punkt, wo die Dreieckfüllrate unter 5% des Maximums fällt.

Anzahl Pixel pro Dreieck	Matisse	Capilla	C/N
1	833000	1010000	1.21
9	792000	1000000	1.26
13	661000	959000	1.45
17	591000	890000	1.51
34	405000	654000	1.62
254	118000	184000	1.56

Table 9: GLperf Resultate: Dreieckfüllraten. Flat Shaded, Z-Buffered.

Wiederum kann man sehen, dass die publizierten Höchstleistungen (Tabelle 2) den Werten mit einem Pixel pro Dreieck entsprechen. Dies ist nicht sehr praxisnah.

8. Fazit

Wie in den Tests gesehen, ist die Graphikleistung stark vom Prozessor abhängig. Dies gilt aber nur solange, wie das Graphiksystem grössere Kapazität besitzt.

Für reale Applikationen, wie mit Viewperf gemessen, scheint das Pentium II System mit der Graphikkarte viel besser zusammenspielen.

Das grosse Fragezeichen ist der Geometrie Pipeline Prozessor. Die Resultate zeigen, dass die Geometrieleistungen stark vom Prozessor abhängen. Die Fähigkeiten des Geometrie Pipeline Prozessor sollten aber den Hauptprozessor entlasten. Dies scheint nicht der Fall zu sein oder sie werden nicht vollständig ausgenutzt.

Ein Aspekt, der hier nicht getestet wird, ist die Geschwindigkeit, mit der Daten zwischen Prozessor und Graphikkarte verschoben werden. Das Bussystem, hier der Accelerated Graphics Port (AGP) von Intel, kann auch ein Flaschenhals werden. Wie in Tabelle 2 angegeben, sieht man, dass die Bandbreitenlimitierung von AGP (1x) bei 266MByte/s liegt. Die maximale Bandbreite des Graphikcontrollers liegt aber bei 664 MByte/s. Hier wäre eine Karte mit einem AGP 2x Interface vielleicht besser.

In einer weiteren Testserie könnte man die Leistung der Systeme noch genauer evaluieren. Dazu wäre es allerdings notwendig, dass beide Systeme gleichviel Arbeitsspeicher, Level 1 und Level 2 Cache hätten.

Interessant wäre auch, die Graphikkarte variabel zu halten und nur ein System zu testen. Ein Beispiel wäre eine Graphikkarte mit dem Nvidia Riva TNT Chip und der schon getesteten Permedia 2 Karte zu verwenden. Da könnte man sicher auch wieder Schlüsse über die Rolle des Prozessors für die Graphikleistung ziehen, da der Riva TNT Chip wesentlich leistungsfähiger ist:

- 128 Bit Architektur
- 180 Mio. Pixel/sec Füllrate
- 6 Mio. Dreiecke/sec
- 1.8 GByte/sec Frame Buffer Bandbreite
- AGP 2x 534 MByte/s oder AGP 4x 1068 MByte/s
- Zwei Textur Engines (Twin Texel (TNT) Arch.)
- 16MByte SDRAM/SGRAM

Ein anderer interessanter Aspekt, wäre zu versuchen, die Leistungsgrenzen der Graphikkarte zu erreichen. Dazu wäre aber ein viel leistungsfähigeres System zum Beispiel mit einem DEC Alpha Prozessor nötig.

Referenzen

- [1] Markus H. Gross. *Graphische Datenverarbeitung*. Skript zur Vorlesung, ETH Zürich, 1998.
- [2] Rai Jain. *The Art Of Computer Systems Performance Analysis*. John Wiley & Sons, 1991.
- [3] The OpenGL Performance Characterization Project. *The GLperf Benchmark*. Homepage under <http://www.spec.org/gpc/opc.static/glpind%7E1.html>.
- [4] The OpenGL Performance Characterization Project. *The Viewperf Benchmark*. Homepage under <http://www.spec.org/gpc/opc.static/opcview.html>.
- [5] The Standard Performance Evaluation Corporation. *The SPEC CPU95 Benchmark*. Homepage under <http://www.spec.org/osg/cpu95>.