Vortrag anlässlich des Fachseminars

"Aktuelle Themen der Graphischen Datenverarbeitung"

Talisman: Commodity Realtime 3D Graphics for the PC

von Jay Torborg, James T. Kajiya (Microsoft Corporation)

Ivo Sele

Inhalt

1. Einführung

Talisman ist eine neue 3D Graphik und Multimedia Hardware Architektur von Microsoft. Sie nutzt die zeitliche und räumliche Kohärenz von 3D-Animationen aus und reduziert dadurch die Kosten von Animationen in hoher Qualität.

Das Hauptziel der Talisman-Architektur ist die Allgegenwart von bewegter 3D Graphik. Traditionellerweise wird 3D Graphik als Tool eingesetzt, z.B. zur Visualisierung von komplexen Beziehungen oder für CAD. Das Ziel der Entwickler von Talisman ist es, 3D Graphik als Medium zu etablieren, z.B. für virtuelle Welten.

Die Voraussetzung für ein erfolgreiches Medium ist eine breite Verfügbarkeit. Denn nur ein grosses Publikum ermöglicht interessante, abwechslungsreiche Inhalte. Für eine breite Verwendung ist ein tiefer Preis wichtig. Jeder der 3D Animationen verwenden will, sollte sich die entsprechende Technologie leisten können. Des weiteren ist für eine breite Palette von Anwendungen ein minimaler Level an Funktionen und Qualität erforderlich.

Grundsätzlich gibt es zwei verschiedene Möglichkeiten kostengünstige Graphik Hardware herzustellen. Ein oft begangener Weg ist es, eine abgespeckte Version konventioneller Hardware herzustellen. Das Risiko dabei ist, dass jede Kostenersparnis zu einem Qualitätsverlust führt. Eine andere Möglichkeit ist die Entwicklung neuer Architekturen, die grundlegende Unterschiede zur konventionellen Graphik Pipeline aufweisen. Die Talisman-Architektur wählt diesen Ansatz.

Neben dem Preis ist die Bildqualität ein wichtiges Kriterium. Die Autoren des Papers gehen davon aus, dass die Beurteilung des Qualitätskriteriums abhängig von der Anwendung ist. Bei Anwendungen wie CAD ist eine präzise, genaue Darstellung der Formen notwendig. Ein low-cost System hat eine wesentlich tiefere Frame-Rate, stellt aber genau die gleichen Bilder dar. Für den Einsatz von 3D Graphik als Medium sind andere Eigenschaften wichtig. Die Animationen sollten möglichst flüssig und synchronisierbar mit Sound und Video sein. Die Möglichkeit der Interaktion mit kurzer Reaktionszeit ist ebenfalls notwendig. Die Genauigkeit der Formen und ihrer geometrischen Beziehungen sowie die Bildqualität sind Leistungsmassstäbe. Konkret heisst das, dass Interaktion stets in Echtzeit bei Frame-Raten, die der Bildwiederholfrequenz (72-85 Hz) entsprechen, möglich sein soll. Der Unterschied zwischen teuren und billigen Systemen besteht in der Genauigkeit der Formen und der Bildqualität.

2. Grundprobleme bei Echtzeit 3D Graphik

Eine konventionelle 3D Graphikpipeline ist in verschiedene Stufen unterteilt. Bei jedem neuen Frame muss die gesamte Szene die folgenden Schritte durchlaufen:

- Übersetzung des Modells in den Koordinatenraum der Applikation und Clipping
- Transformation der Objekte
- Beleuchtung neu berechnen
- Tesselation (Zerlegung in Dreiecke)

- Schattierung
- Z-Buffer zur Behandlung von versteckten Flächen
- Texture Mapping unter Berücksichtigung der Perspektive
- Spezialeffekte wie Nebel
- Alpha Blending für transparente Objekte
- Anti-Aliasing (Glättung der Kanten)
- Schreiben der Pixel des endgültigen Bilds

Die ersten vier Schritte sind eher Mathematik-intensiv und laufen daher üblicherweise auf der CPU des Hostrechners. Die restlichen Schritte sind sehr Speicher-intensiv. Sie lassen sich durch entsprechende Hardware stark beschleunigen.

Bei Verwendung einer konventionellen 3D Graphikpipeline ergeben sich die folgenden fundamentalen Probleme: Der Bedarf an Speicher Bandbreite ist enorm, da für jedes neue Bild alle Schritte der Pipeline durchlaufen werden müssen. Das Bild wird jedesmal komplett neu berechnet. Ein weiteres Problem ist die Latenzzeit des Systems (system latency), also unerwünschte Verzögerungen. Für low-cost Systeme stellt die benötigte Speichermenge ein weiteres Problem dar.

Der Bedarf an Speicherbandbreite bei einer konventionellen Graphik Pipeline geht von 190 MByte/Sek bei einer Auflösung von 640*480 mit 16 Bit Farbtiefe und 30Hz Update-Rate sowie simplen Texturen bis zu 12'000 MByte/Sek bei hoher Auflösung und Update-Rate sowie Texturen hoher Qualität und Anti-Aliasing. Es gibt nach wie vor grosse Leistungsunterschiede zwischen 3D Beschleunigern für den PC und high-end Graphik Workstations. Der Hauptgrund dafür ist die enorme Bandbreite die für 3D-Animationen in hoher Auflösung und mit hoher Frame-Rate benötigt wird. Eine solche Speicherbandbreite ist immer noch sehr teuer.

Der Speicherbedarf einer 3D Graphik Pipeline liegt für die gleichen Auflösungen und Bildqualitäten wie vorher zwischen 2.6 und 45 MByte. Neben dem Frame-Buffer benötigen auch fortgeschrittene Texture Mapping und Anti-Aliasing Verfahren sehr viel Speicher. Für high-end Systeme ist der Speicherbedarf jedoch nicht das Hauptproblem, da die benötigte Bandbreite nur durch einen grossen Hauptspeicher erreichbar ist.

Bei den Fortschritten der DRAM-Technologie zwischen 1976 und 1995 fällt vor allem der stark gesunkene Preis pro MByte auf. Die Ursache dafür ist die enorme Kapazitätssteigerung. Bei der Zugriffszeit und Bandbreite waren die Verbesserungen jedoch weit weniger markant. Diese Entwicklung macht wiederum deutlich, dass die Speichermenge nicht das Hauptproblem ist. Das Problem der Bandbreite und Zugriffszeit wurde jedoch nicht wesentlich entschärft. Es bleibt also nach wie vor ein sehr teures Unterfangen, 3D-Animationen hoher Qualität mit einer 3D Graphikpipeline zu erreichen.

3. Bildverarbeitung und 3D Graphik

Ein grosser Teil der Leistung einer konventionellen Graphikpipeline ist nicht wirklich nutzbar. Eine konventionelle Pipeline kann in jedem Frame bei voller Leistung ein völlig verschiedenes geometrisches Modell darstellen. Der Viewpoint kann sich von einem Frame zum nächsten beliebig bewegen. Zudem kann jedes beliebige Muster als Texture Map dienen, obwohl die Mehrheit optisch nicht von zufälligem Rauschen unterschieden werden kann.

Die Talisman-Architektur nutzt die zeitliche Kohärenz der Modelle, der Bewegung und des Viewpoints sowie die räumliche Kohärenz der Texturen und des Displays. Dadurch lässt sich eine wesentliche Verminderung des immensen Bedarfs an Speicherbandbreite und -menge erreichen.

Eine weitere fundamentale Methode die in Talisman verwendet wird ist es, Bildsynthese wenn möglich durch Bildverarbeitung zu ersetzen, z.B. kann ein sich entfernendes Objekt durch eine simple Skalierung simuliert werden.

4. Grundkonzepte der Hardware-Architektur

Die Talisman-Architektur verwendet zusammengesetzte Bildschichten (Image Layers). Es wird kein Frame Buffer verwendet; statt dessen werden unabhängige Bildschichten bei Video-Rate zusammengesetzt, um das Ausgangssignal zu erzeugen. Diese Bildschichten können einzeln gerendert und manipuliert werden. Normalerweise wird für jedes unabhängige Objekt eine eigene Bildschicht verwendet. Das hat den Vorteil, dass die Update-Raten pro Objekt gesetzt werden können (z.B. müssen Objekte im Hintergrund nicht so oft neu gerendert werden wie Objekte im Vordergrund). Jeder Pixel in einer Schicht enthält Farb- und Alpha-Information, damit diese Schichten anschliessend zum endgültigen Bild zusammengesetzt werden können. Alle affinen Transformationen können mit Videorate auf den Bildschichten ausgeführt werden. Die Rendering Rate der einzelnen Bildschichten ist also variabel, Transformationen (z.B. Bewegung) erfolgen jedoch mit Videorate. Dadurch wird eine flüssige Dynamik garantiert. Wie schon angetönt lassen sich viele 3D Transformationen durch 2D Bildverarbeitung simulieren. Durch die Verwendung von 2D Transformationen mit vorher gerenderten Bildern für dazwischen liegende Frames sind grosse Einsparungen möglich. Verschiedene Bildschichten können mit verschiedenen Auflösungen gerendert werden, z.B. kann der Himmel mit einer geringen Auflösung gerendert werden. Anschliessend wird er auf Displaygrösse skaliert und durch hoch-qualitative Filterung verbessert. Das Talisman System kann diese trade-offs zwischen Geschwindigkeit und Genauigkeit automatisch verwalten.

Ein weiteres wichtiges Konzept ist die Bildkompression. Talisman macht intensiven Gebrauch davon. Traditionellerweise wird Kompression für 3D Graphiksysteme wegen der Berechnungskomplexität nicht verwendet. Aufgrund des im Talisman-System verwendeten Chunkings (Erklärung weiter unten) ist eine effiziente Komprimierung von Bildern und Texturen möglich. Das führt zu einer wesentlichen Verbesserung des Preis-Leistungs-Verhältnisses. Der für die Kompression verwendete Algorithmus ist ähnlich wie JPEG. Er heisst TREC (Texture And Rendering Engine Compression) und wurde speziell für diese Zwecke entwickelt. In der Regel lassen sich Kompressionsfaktoren von über 10:1 erreichen. Aufgrund der Komprimierung kann ein einziges Bildformat (32 Bit true color) für alle Applikationen verwendet werden.

Chunking, also die Aufteilung der Bildschichten in Pixel-Regionen ist ein weiteres Grundkonzept. Aufgrund des zufälligen Zugriffs auf Pixel in traditionellen 3D Graphik Systemen ist die Verwendung von Kompression für Display Buffers nicht praktikabel. Bei Talisman ist jede Bildschicht unterteilt in Pixel-Regionen, sogenannte Chunks (32x32 Pixel in der Referez-Implementation). Die Objekte werden vorsortiert in Bins, je nachdem in welchen Chunk sie anschliessend gerendert werden. Dieser Vorgang heisst chunking. Objekte, die Chunk-Grenzen überlappen, werden in jedem Chunk in dem sie sichtbar sind referenziert. Diese Datenstruktur wird modifiziert, wenn Objekte sich von einem Chunk in einen anderen bewegen. Die Verwendung dieser Pixel-Regionen führt zwar zu einem gewissen Overhead, bietet aber wesentliche Vorteile: Der Depth Buffer muss nur so gross wie ein Chunk sein, da immer nur ein Chunk auf einmal gerendert wird. Dadurch wird eine effiziente Implementation des Depth Buffers

(direkt auf dem Chip) möglich. Anti-Aliasing wird ebenfalls wesentlich einfacher, weil jeder Chunk separat behandelt werden kann. Die erforderliche Datenmenge wird stark reduziert, was die Implementation von sophistizierten Anti-Aliasing Algorithmen erlaubt. Zudem können einzelne Chunks nach dem Rendern direkt komprimiert werden.

Das letzte Grundkonzept ist das Rendering in mehreren Durchgängen (multi-pass rendering). Dadurch kann verhindert werden, dass alle gerenderten Bilder das gleiche charakteristische Aussehen haben. Talisman verwendet ein einziges Shared Memory System für die Speicherung der komprimierten Texturen und Bildschichten. Diese Architektur erlaubt es, Daten die beim Rendern entstehen durch den Texture Processor zurückzuführen und als Daten für das Rendering einer neuen Bildschicht zu verwenden. Durch die Kombination von multi-pass Rendering mit verschiedenen Kompositionsmodi, Texture Mapping Verfahren und einer flexiblen Schattierungssprache bietet das Talisman-System eine Vielfalt an Rendering-Effekten.

5. Referenz Implementation

Die Talisman-Architektur ermöglicht verschiedenste Implementationen mit unterschiedlichen Leistungsdaten. Die Referenz-Implementation ist eine Erweiterungskarte für den PCI Bus. Sie bietet neben 3D- und Windows-Beschleunigungs Funktionalität, MPEG Playback sowie die Funktionalität eines Video Konferenz Boards, einer Soundkarte und eines Modems.

Für die Referenz Implementation wurden folgende Standardkomponenten verwendet: Der Media DSP ist verantwortlich für Video Codecs, Audio Verarbeitung und front-end Bildverarbeitung (Transformationen, Beleuchtung etc.). Der Media (Video) DAC beinhaltet zusätzlichen einen USB seriellen Kanal und einen Media Channel. Ein separater Audio Chip ist für die Audio AD/DA Wandlung zuständig.

Einige VLSI Komponenten werden speziell für die Referenz-Implementation entworfen: Der Polygon Object Processor ist zuständig für Scan-Conversion, Schattierung, Texture Mapping, Entfernen von versteckten Flächen und Anti-Aliasing. Der Image Layer Compositor arbeitet mit Videorate. Er führt affine Transformationen auf Bildschichten durch und schickt die resultierenden Pixel zum Compositing Buffer. Der Compositing Buffer bietet Platz für zwei 32 scanlines grosse Farbpuffer und einen Alpha-Puffer.

Als Shared Memory werden 4MB Rambus Speicher verwendet. Dieser bietet eine höhere Bandbreite als Standard-DRAM zu einem nur unwesentlich höheren Preis. Das Shared Memory speichert die komprimierten Bildschichten und Texturdaten, Code und Daten für den DSP sowie diverse Puffer.

5.1 Polygon Object Processor

Wie bereits erwähnt ist der Polygon Object Processor zuständig für Scan-Conversion, Schattierung, Texture Mapping, Entfernen von versteckten Flächen und Anti-Aliasing. Viele Elemente des Polygon Object Processors werden auch in konventionellen 3D Graphik Pipelines verwendet. Deshalb werden hier nur einige besondere Teile beschrieben.

Die Initial Evaluation Komponente wird benötigt, weil Polygone in 32x32 Pixel grossen Chunks bearbeitetet werden. Dadurch fängt die Bearbeitung von Dreiecken üblicherweise nicht am Scheitelpunkt des Dreiecks an. Hier wird der Schnittpunkt des Chunks mit dem Dreieck berechnet. Die Pixel Engine führt Berechnungen auf Pixel-Ebene aus. Dazu gehören Komposition, Tiefen-Pufferung und Fragment-Erzeugung für teilweise bedeckte Pixel. Der Fragment Resolve Block führt den endgültigen Anti-Aliasing Schritt aus, indem Pixel-Fragmente mit teilweiser Bedeckung oder Transparenz aufgelöst werden.

Eines der grössten Probleme bei der Entwicklung dieses Chips, war es mit der langen Latenzzeit beim Zugriff auf Texture Daten zurechtzukommen. Das Aufrechterhalten der vollen Pixel-Rate beim Rendern war eine wesentliche Zielsetzung des Designs. Daher ist es wichtig, dass stets Texel für die Texture Filter Engine verfügbar sind. Dieses Problem wird durch die Verwendung von zwei getrennten Rasterizers gelöst. Der Pre-Rasterizer ist zuständig für die Berechnung der Texel-Adressen während der eigentliche Rasterizer die Farb-, Tiefen-, und Pixel Adressen-Interpolation für das Rendering durchführt. Beide Rasterizer berechnen Informationen für die gleichen Pixel. Sie sind jedoch durch einige hundert Taktzyklen getrennt. Die Adressen der Pixel werden mittels einer Pixel-Queue vom Pre-Rasterizer an den Rasterizer weitergereicht.

5.2 Image Layer Compositor

Der Image Layer Compositor erzeugt den Graphikoutput aus einer Sammlung von nach Tiefe sortierten Schichten. Einige der Teile sind identisch mit den beim Polygon Object Processor verwendeten. Der Image Layer Compositor muss die Bilder von multiplen, sich überlappenden Objekten bei vollen Videoraten zusammensetzen. Daher ist die Rendering Rate acht mal höher als beim Polygon Object Processor.

Es werden ebenfalls zwei Rasterizer verwendet. Da dieser Chip keine perspektivischen Transformationen durchführt, wird nur simple bi-lineare Filterung verwendet und nur lineare Address-Kalkulationen sind notwendig. Dadurch wird der Rasterizer wesentlich einfacher. Gleichzeitig ist eine höhere Pixel-Rate erforderlich. Der Pre-Rasterizer muss also wesentlich weiter vor dem Rasterizer arbeiten als beim Polygon Object Processor. Daher wird keine Pixel-Queue verwendet, sondern die Parameter werden im zweiten Rasterizer neu berechnet.

6. Schlussfolgerung

Das Ziel der Talisman-Architektur ist die Allgegenwart von Echtzeit 3D Graphik. Besonders wichtig für die Erreichbarkeit dieses Ziels ist eine ausreichende Leistung und ein tiefer Preis.

Nun möchte ich auf die Frage eingehen, ob das Talisman-System dieses Ziel erreicht beziehungsweise erreichen kann. Bezüglich der Leistung würde ich diese Frage mit ja beantworten. Das System bietet bezogen auf sein Einsatzgebiet als Medium eine sehr gute Leistung. Die Frage, ob die angepeilten tiefen Kosten erreicht werden, ist nicht so leicht zu beantworten. Laut Microsoft wird der Verkaufspreis einer Talisman-Implementation im Bereich zwischen $200 bis $500 liegen. Ein unabhängiger Bericht von MicroDesign Resources kommt dagegen zum Schluss, dass dieser vermutlich über $600 betragen wird. Für eine weite Verbreitung wäre dagegen ein Preis von ca. $200 bis $350 nötig. Die besprochene Talisman-Architektur ist also eher eine Maximallösung. Längerfristig ist jedoch ein tieferer Preis möglich und das System könnte sich durchaus durchsetzen.