Fachseminar-Vortrag
Graphische Datenverarbeitung
Prof. M. Gross

Interactive Multiresolution Surface Viewing

gehalten am 13.5.1997
von Daniel Estermann an der
Abteilung für Informatik, ETH Zürich

auf der Grundlage der gleichnamigen Publikation [1] von
Andrew Certain, Jovan Popovic, Tony DeRose, Tom Duchamp, David Salesin und Werner Stuetzle (im folgenden Autoren genannt),
University of Washington

Inhalt

1 Motivation

Oberflächen spielen eine zentrale Rolle in vielen dreidimensionalen Computergraphik-Anwendungen. Dabei bereitet die steigende Komplexität dieser Objekte mehrere Probleme: Sie benötigen viel Speicherplatz und können somit nur langsam übertragen werden. Oft sind sie zudem für eine interaktive Darstellung auf aktueller Hardware zu detailliert.

Ein einfacher Lösungsansatz – sowohl für die Übertragung wie auch für die dynamische Darstellung – besteht darin, eine vorher berechnete Folge von Näherungen mit niedrigerer Auflösung zu übertragen. Während der Darstellung eines groben Bildes treffen die höher aufgelösten Gitterdaten beim Benutzer ein. Leider hat dieser Ansatz einen grossen Nachteil: Die zu übertragenden und zu speichernden Daten sind umfangreicher als die Gitterdaten voller Auflösung. Es muss daher ein Kompromiss gefunden werden zwischen Granularität (Auflösungsunterschied zwischen zwei aufeinanderfolgenden Modellen) und Übertragungszeit sowie Speicherplatz.

Mehrere Arbeiten haben gezeigt, dass *Multiresolution Analysis* (MRA) sich zur Lösung dieser Probleme eignet. Es bietet Kompression, progressive Übertragung und eine adaptive Auflösungskontrolle für komplexe Gitternetze. Bis zum Erscheinungszeitpunkt von [1] weisen jedoch alle Arbeiten mindestens zwei Mängel auf. Erstens werden nur entweder Farbe oder Geometrie in Multiresolution-Form repräsentiert, nie aber beide gleichzeitig. Zweitens sind die Algorithmen zur Rekonstruktion und Darstellung der Objekte zu langsam für eine interaktive Anwendung.

Ziele

Die Autoren erreichen mit der Separation von Farbe und Geometrie eine effiziente Implementation folgender Merkmale:

- **Progressive Übertragung:** Eine attraktive Methode für die Darstellung eines komplexen Objektes ist, mit einer grob aufgelösten Version zu beginnen, um die Darstellung dann sukzessive zu verfeinern. Mit Multiresolution Analysis bedeutet dies, zuerst das Basisgitter zu übertra-

gen, und dann die Beiträge der Wavelet-Koeffizienten mit absteigendem Gewicht miteinzuberechnen.

- **Detailstufen:** Durch Weglassen der kleinsten Farb- und Geometriekoeffizienten erhält man gröber aufgelöste Näherungen. Damit kann auf Kosten der Details eine gewünschte Bildwiederholrate aufrecht erhalten werden, selbst bei wechselnder Maschinenbelastung. Überflüssige Details können auch dann weggelassen werden, wenn das Objekt weit vom Betrachter entfernt ist (Abbildung 1).
- **Automatische Texturgenerierung:** Die Aufspaltung von Farbe und Geometrie erlaubt es, von Texture-Mapping-Hardware Gebrauch zu machen. Für eine gegebene Zahl von Gitterpunkten liefert Texture-Mapping eine weitaus bessere Approximation. Farbe kann also auf diesbezüglich beschleunigten Systemen immer in voller Auflösung dargestellt werden, da dies die Bildwiederholrate nicht beeinflusst.
- **Anpassung an Benutzerbedürfnisse:** Bei Abwesenheit von Texture-Mapping-Hardware hängt die Anzahl der dargestellten Polygone von der Schwelle ab, bei der man die Wavelets „abschneidet". Diese Schwellen können wiederum für Farbe und Geometrie getrennt gewählt werden. Im Allgemeinen existieren viele solche Kombinationen, die in der gleichen Anzahl Polygone resultieren. Automatisch diejenige Kombination zu finden, welche am besten aussieht, ist sehr schwierig, da dies stark vom Objekt selbst abhängt. Stattdessen überlassen die Autoren diese Entscheidung dem Benutzer.

All diese Merkmale weist auch die von den Autoren entwickelte Demonstrations-Anwendung auf. Sie zeigt eindrücklich, mit welcher Geschwindigkeit komplexe 3D-Modelle z.B. über das World Wide Web betrachtet werden können.

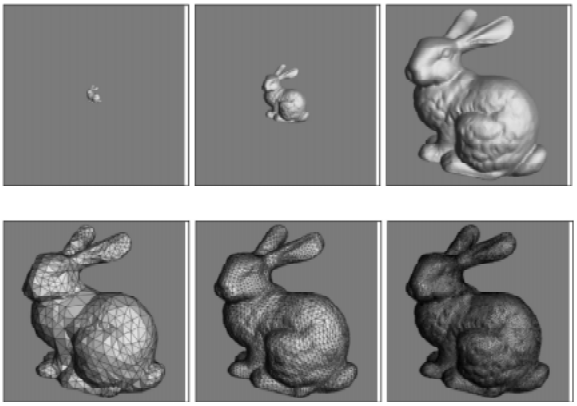

Abbildung 1: Detailstufen, abhänig von der Darstellungsgrösse (aus [2]).

2 Multiresolution Analysis

In diesem Abschnitt versuche ich, einen groben Überblick über die Grundlagen der Multiresolution Analysis zu geben. Als Quellen dienten mir dabei in erster Linie [5], [3] und [1].

Idee

Die zentrale Idee von Multiresolution Analysis ist, eine Funktion mittels Filterung in einen grob aufgelösten Teil und eine Serie von Detailinformationen mit steigender Auflösung zu zerlegen. Diese lokalen Korrekturterme nennt man Wavelet-Koeffizienten, sie speichern die Details des Objekts in verschiedenen Auflösungen. Zum Beispiel zeigt Abbildung 2(b) den grobaufgelösten Teil des Gitters in Abbildung 2(a).

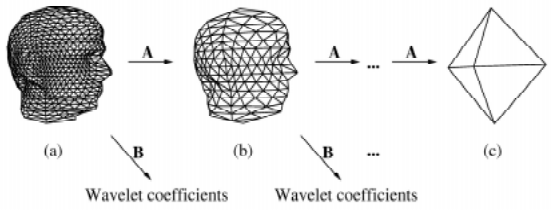

Abbildung 2:
Multiresolution-Zerlegung (aus [3])

Die Extraktion des groben Teils ist linear, somit kann sie als Multiplikation mit einer Matrix A^j geschrieben werden. Die Wavelet-Koeffizienten, welche die Detailinformation repräsentieren, können analog durch Multi-

plikation mit B^j berechnet werden. Invertiert man die sogenannten Analyse-Filter A^j und B^j, so erhält man die Synthese-Filter P^j und Q^j. Die Synthese, also die Rückgewinnung des Originals ausgehend vom Basisgitter mit Hilfe der Wavelet-Koeffizienten, kann auch als Subdivision beschrieben werden: Aufteilen jeder Dreiecksfläche an den Kantenmittelpunkten, und Verschieben der neuen Knoten gemäss den Wavelet-Koeffizienten.

Mathematische Grundlagen

Unser Startpunkt ist eine geschachtelte Serie von Vektorräumen $V^0 \subset V^1 \subset V^2 \subset \ldots$. Mit steigendem j nimmt auch die Auflösung der Funktionen in V^j zu. Die Basisfunktionen $\phi_i^j(x)$ der Räume V^j werden auch *Skalierungsfunktionen* genannt. Vorderhand wollen wir die Funktionen auf die Klasse der stückweise linearen Funktionen einschränken. Eine Funktion, definiert auf einem Gitternetz M^0, wird J-fach stückweise linear genannt, wenn sie stückweise linear ist auf M^J, wobei M^J durch J-fache rekursive Unterteilung von M^0 entstanden ist (Abbildung 3). In [2] wird gezeigt, wie man eine allgemeine Oberfläche approximiert, sodass sie dieses Kriterium erfüllt.

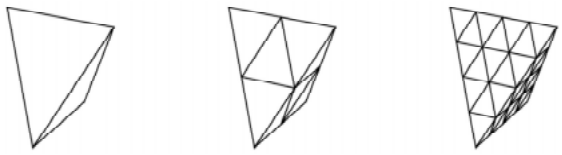

Abbildung 3:
Rekursive 4-zu-1-Subdivision eines Tetraeders: (a) M^0, (b) M^1, (c) M^2 (aus [1])

Der nächste Schritt ist die Definition von Wavelet-Räumen. Für jedes j definieren wir W^j als das orthogonale Komplement von V^j in V^{j+1}. Das heisst W^j enthält alle Funktionen in V^{j+1}, die orthogonal sind zu jenen in V^j. Man schreibt deshalb oft

$$W^j = V^{j+1} - V^j.$$

Die Basisfunktionen $\psi_i^j(x)$ für W^j nennt man *Wavelets*. Mit einer solchen Wavelet-Basis können wir jede J-fach stückweise lineare Funktion f auf M^0 als Linearkombination von Skalierungsfunktionen und Wavelets verschiedener Stufen ausdrücken.

Für die Orthogonalität brauchen wir ein weiteres Grundelement der Multiresolution Analysis, das innere Produkt zweier Funktionen

$$\langle f \,|\, g \rangle := \sum_T \int_{x \in T} f(x) g(x)\, dx \,,$$

Dabei läuft die Summation über alle Knoten von M^0; das Flächenelement dx ist normalisiert, damit alle Flächen Einheitsgrösse haben.

Bei der Wahl der Wavelet-Basen gilt es darauf zu achten: (i) dass die grobaufgelösten Gitter gute Näherungen des Originals sind; (ii) die Grösse eines Wavelet-Koeffizienten dessen Wichtigkeit entspricht, entsprechend dem Fehler, der beim Nullsetzen des Koeffizienten entsteht; und (iii) dass Analyse- und Synthesefilter eine lineare Komplexität in der Anzahl Knotenpunkte aufweisen.

Um Forderung (ii) zu erfüllen, sollten die gewählten Wavelets orthogonal zu den Skalierungsfunktionen sein. Auf der anderen Seite wollen wir deren Support so klein wie möglich halten, um Forderung (iii) nachzukommen. Unglücklicherweise kollidieren aber diese beiden Ziele. Da kleiner räumlicher Support für die Anwendung essentiell ist, lockern wir die Orthogonalitätsanforderungen.

k-disc-Wavelets

Die Autoren von [1] wählen den von Lounsbery [6] vorgeschlagenen Ansatz: Man setzt die Grösse k des Supports a priori fest und konstruiert dann *biorthogonale Wavelets*, die den Raum $\underline{W}^j$ aufspannen und so orthogonal wie möglich zu den Skalierungsfunktionen in $\underline{V}^j$ sind.

Konkret stelle man sich einen Knotenpunkt i eines Gitters M^{j+1} vor, der auf dem Mittelpunkt der Kante e von M^j liegt. Das k-disc-Wavelet mit Mittelpunkt am Knoten i ist eine Fuktion der Form

$$\underline{\psi}_i^j = \underline{\phi}_i^{j+1} + \sum_{v \in N_k} s_v^j \underline{\phi}_i^{j+1} \,,$$

wobei N_k für eine Menge Knoten der Stufe j in der Umgebung von Knoten i steht. Diese Umgebungen N_k sind rekursiv definiert. Die

Umgebung N_0 für das 0-disc-Wavelet besteht aus den Endpunkten der Kante *e*; die Umgebung N_k enthält die Knoten von allen Flächen, die an N_{k-1} grenzen (Abbildung 4). Das Wavelet, welches nur aus der Skalierungsfunktion der Stufe $j+1$ besteht, wird *lazy*-Wavelet genannt. Es ist sehr einfach zu handhaben, ist aber weit von Orthogonalität entfernt.

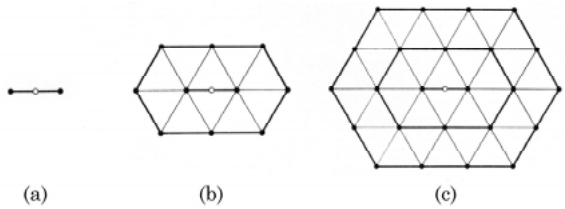

Abbildung 4:
Knoten innerhalb der k-discs: (a) 0-disc, (b) 1-disc, (c) 2-disc; (aus [5])

Die Koeffizienten s_{iv}^j sind so gewählt, dass die Norm der orthogonalen Projektion von $\underline{\psi}_i^j$ auf $\underline{V}^j$ minimiert wird. Man erhält sie durch Lösen des folgenden Gleichungssystems:

$$\sum_{v \in N_k} \left\langle \underline{\phi}_u^j \middle| \underline{\phi}_v^j \right\rangle s_{iv}^j = -\left\langle \underline{\phi}_u^j \middle| \underline{\phi}_v^{j+1} \right\rangle$$

für alle $u \in N_k$.

Man beachte die Lokalität des Systems um Knoten *i*. Für grössere k hängt das System von den Knoten eines grösseren Gebietes ab, der Support also nimmt zu.

Umwandlung von (farbigen) Polygonmodellen in Multiresolution–Form

Multiresolution Analysis eines farbigen Gitternetzes basiert auf der Bedingung, dass *M* parametrisch definiert ist durch zwei vektorwertige *J*-fach stückweise lineare Funktionen: f_{geom} und f_{color} bilden ein Dreiecks-Basisgitter auf $\mathbf{R}^3$ ab.

Typischerweise liegt aber *M* nicht in dieser Form vor, sondern in Form von Knoten, Kanten, Flächen, Knotenkoordinaten und Knotenfarben. *M* muss also zuerst parametrisiert werden. Dies geschieht mit dem 'Remeshing'-Algorithmus von Eck [2]. Dessen Resultat ist ein Basisgitter mit wenigen Knoten, eine Parametrisierung ρ: $M^0 \rightarrow M$, und eine Näherung von ρ durch die bereits erwähnte Geometriefunktion f_{geom}: $M^0 \rightarrow \mathbf{R}^3$.

Die Zerlegung in Wavelets liefert dann einen Vektor von 3 Koeffizienten für jedes Wavelet. Diese Koeffizientenvektoren werden nach absteigender Länge sortiert.

Nun zur Analyse der Farbfunktion: Die Farbe ist ursprünglich an den Knotenpunkten von *M* gegeben und kann somit für jeden Punkt der Oberfläche interpoliert werden. Die erhaltene Parametrisierung ρ der Geometrie führt zu einer ähnlichen Funktion γ auf M^0. Um eine *J*-fach stückweise lineare Näherung f_{color} von γ zu erhalten, tastet man γ an den Knoten von M^J ab. Wie bei der Geometrie wird dann die Analyse mittels Filterbank durchgeführt, womit man die Koeffizienten der Farbwavelets erhält.

3 Implementation

Dieses Kapitel gibt einen kurzen Überblick über die im entwickelten Multiresolution Viewer verwendeten Konzepte.

Algorithmen & Datenstrukturen

Die Autoren gehen davon aus, dass die Objektdaten bereits in Multiresolution-Form vorliegen. Intern wird das Objekt in Form eines Baums gespeichert. Die Wurzel hat soviele Kinder wie das Basisgitter Flächen. Diese Flächenknoten haben jeweils vier Nachkommen, entsprechend derer 4-zu-1-Subdivision. Die Zeichnung des Objekts geschieht durch eine Traversierung dieses Baums, indem f_{geom} und f_{color} an den Knoten ausgewertet werden und für jedes Blatt ein farbiges Polygon erzeugt wird. Während der progressiven Übertragung treffen laufend Wavelet-Koeffizienten ein, welche neue Blätter im Baum erzeugen. Koeffizienten ab einer gewählten Schwelle können ignoriert werden, was den Baum verkleinert und somit die Traversierung beschleunigt.

Bei Verwendung von Texture-Mapping-Hardware bewirkt das Eintreffen eines Farbwavelets eine Veränderung des Texturspeichers. Die anfänglichen Texturen der Basispolygone erhält man durch Interpolation der Eckpunktfarben. Die Tatsache, dass die Geometrie parametrisch durch eine stückweise lineare Funktion repräsentiert

wird, vereinfacht die (sonst nicht triviale) Berechnung der Texturkoordinaten.

Viewer–Architektur

Der Viewer, in OpenGL und Motif für Silicon Graphics Workstations geschrieben, agiert als eine Hilfsanwendung für den Netscape Navigator. Zu Beginn einer Übertragung holt sich der Viewer zuerst das Basisgitter, dann öffnet er zwei separate HTTP-Verbindungen, je eine für Geometrie- und Farbwavelets. Während dem Empfang dieser Daten wird das Modell periodisch neugezeichnet. Die Bildfolge auf der Titelseite zeigt das Resultat einer Übertragung über eine 64 KBit-Leitung (nach 3, 17, 59 und 180 Sekunden).

Die Qualität der Anzeige kann auf verschiedene Weise über Schieberegler beeinflusst werden. In einem Modus kann der Benutzer die Anzeigerate wählen, der Viewer passt die Modellkomplexität dementsprechend an. Alternativ können entweder die Anzahl der Geometrie- und Farbwavelets oder die Anzahl Polygone der Approximation gewählt werden.

Bei der Wahl von Wiederholrate oder Polygonzahl lässt sich zusätzlich das Verhältnis von Farb- zu Geometrieauflösung eingestellen. Wenn der Rechner Texture-Mapping beschleunigt, können Farbinformationen in voller Auflösung dargestellt werden, ohne die Anzahl Polygone zu erhöhen. Der softwaremässige Rechenaufwand verändert sich dabei nicht.

4 Ergebnis

Bisherige Arbeiten wurden in zwei Bereichen erweitert. Erstens zeigen die Autoren, wie Multiresolution Analysis unter Trennung von Geometrie und Farbe durchgeführt wird. Zweitens entwickelten sie Datenstrukturen und Algorithmen, welche die Anzeige von komplexen Objekten trotz beschränkter Übertragungsbandbreite und Rechenkapazität bei interaktiven Wiederholraten ermöglichen. Da ich leider keinen Zugang zum entwickelten Prototyp-Viewer hatte, muss ich den angegebenen Zahlen und Bildern vertrauen.

Die Fortschritte bei der Netzbandbreite sowie der Rechenleistung werden teilweise durch immer komplexere Daten und Programme zunichte gemacht. Beispielsweise hat sich *Multimedia* vom Schlagwort regelrecht zum „Ressourcenfresser“ entwickelt. Aus diesem Grund glaube ich, dass die vorgestellte Technik in vielen künftigen Systemen eine Rolle spielen wird.

5 Ausblick

Die Anwendung von Multiresolution Analysis für Oberflächen ist eine relativ junge Entwicklung in der Computergraphik. In jüngster Zeit wurde auf diesem Gebiet intensiv geforscht, nichtsdestotrotz bleiben noch viele Fragen offen und Probleme ungelöst.

Seit der Veröffentlichung von [1] ist bereits eine neue Arbeit erschienen, die das Thema Multiresolution Analysis um die Editierung von Flächen erweitert. Oberflächen können damit in verschiedenen Detailstufen interaktiv verändert werden.

Die hier behandelte Technik ist wahrscheinlich noch nicht der Weisheit letzter Schluss. Momentan ist es Aufgabe des Benutzers, bei gegebener Obergrenze von Anzahl Polygonen das Verhältnis zwischen Farb- und Geometrieauflösung zu wählen. Heuristiken zur automatischen Wahl wären wünschenswert. Auch könnte die Objektrepräsentation mit Hilfe von *k*-disc-Wavelets in mehrfacher Hinsicht verbessert werden:

- Bei den beschriebenen Methoden bleibt die topologische Struktur des Originalobjekts erhalten. Beispielsweise für ein einfaches Objekt mit vielen winzigen Löchern wäre es denkbar, das Ausgangsgitter in eine „einfachere“ Oberfläche zu zerlegen.
- Eine von der Bildschirmauflösung abhängige Fehlermetrik könnte verwendet werden, um bessere Bilder mit noch weniger Dreiecken zu erzeugen.
- Viele Objekte in der Praxis weisen scharfe Diskontinuitäten auf, die in groben Auflösungen erhalten sein sollten. Möglicherweise gelingt dies mit Wavelets mit Unstetigkeiten, die sich dem Objekt anpassen.

Andere Verfahren

Ein anderes Verfahren, das ebenfalls die Schwierigkeiten bei der Speicherung, Übertragung und Anzeige von komplexen Gitter-

modellen angeht, nennt sich *Progressive Meshes* [4]. Die Grundidee besteht darin, die Veränderungen aufzuzeichnen, die ein Optimierer auf ein Gitter anwendet. Dieses wird dann durch das Basisgitter und die Optimierungen in umgekehrter Reihenfolge repräsentiert.

- Mit dieser Methode können auch Oberflächen verlustfrei dargestellt werden, welche die Anforderungen von stückweiser Linearität nicht erfüllen.
- Oft liefert Progressive Meshes bessere Approximationen. Ausgeprägte Unstetigkeiten wie scharfe Kanten bleiben beispielsweise in der maximalen Optimierung erhalten.
- Das Verfahren von Hoppe kann allerdings nicht so intuitiv zur Editierung von Gittern erweitert werden.
- Bei Progressive Meshes kann der Fehler nicht auf ein Maximum beschränkt werden.

6 Danksagungen

Ich möchte mich bei Oliver Staadt für die freundliche Unterstützung und Ausstattung mit Unterlagen bedanken.

7 Literaturverzeichnis

[1] Andrew Certain, Jovan Popovic, Tony D. DeRose, Tom Duchamp, David Salesin und Werner Stuetzle, 1996. **Interactive Multiresolution Surface Viewing**. *SIGGRAPH '96 Conference Proceedings*, Seiten 91-98.

[2] Matthias Eck, Tony D. DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery und Werner Stuetzle, 1995. **Multiresolution Analysis of Arbitrary Meshes**. *SIGGRAPH '95 Conference Proceedings*, Seiten 173-162.

[3] Michael Lounsbery, Tony D. DeRose und Joe Warren, 1993 zur Publikation vorgeschlagen. **Multiresolution Analysis for Surfaces of Arbitrary Topological Type**. *ACM Transactions on Graphics*, Vol. 16, Nr. 1, Januar 1997, Seiten 34-73.

[4] Hugues Hoppe, 1996. **Progressive Meshes**. *SIGGRAPH '96 Conference Proceedings*, Seiten 19-26.

[5] Eric Stollnitz, Tony D. DeRose und David Salesin**. Wavelets for Computer Graphics: Theory and Applications**. Morgan-Kaufmann, 1996.