

Punktbasierte Multiskalendarstellung

Michael Wand

WSI/GRIS, Eberhard Karls Universität Tübingen
Sand 14, 72076 Tübingen
wand@gris.uni-tuebingen.de

Abstract: Die Visualisierung komplexer dreidimensionaler Szenen und Datensätze ist eines der grundlegenden Probleme der Computergraphik. In dieser Arbeit wird ein neuer Ansatz zur effizienten Darstellung komplexer Szenen entwickelt: die Verwendung punktbasierter Multiskalenmodelle. Dabei werden geometrische Objekte durch eine Auswahl von Stichprobenpunkten repräsentiert. Diese werden auf den Objekten so plaziert, daß eine möglichst schnelle Darstellung mit kleinen Stichprobenmengen möglich wird. Mit diesem neuen Ansatz ist es erstmals möglich, hoch komplexe Szenen mit komplexer Topologie in Echtzeit zu visualisieren.

1 Einleitung

1.1 Problemstellung

Die effiziente Darstellung komplexer Szenen ist ein klassisches Problem der Computergraphik. Einerseits gibt es in vielen Gebieten (CAD, Virtual Reality, 3D-Spiele, Visualisierung) den Bedarf, sehr große dreidimensionale Datensätze in Echtzeit zu visualisieren. Andererseits stellt auch leistungsfähige Graphikhardware nur eine begrenzte Rechenleistung zur Verfügung. Um dennoch extrem große Szenen darstellen zu können, benötigt man so genannte *output-sensitive* Darstellungsalgorithmen, deren Rechenzeit möglichst wenig von der Szenenkomplexität sondern im wesentlichen nur von der Bildauflösung abhängt.

Grundsätzlich kann man in der Computergraphik zwischen zwei verschiedenen Ansätzen für Darstellungsalgorithmen unterscheiden: *Projektions-* und *Strahlverfolgungsalgorithmen* [Fo 96]. Projektionsalgorithmen arbeiten alle Objekte der Szene nacheinander ab, projizieren die Flächen der Objekte auf den Bildschirm und bestimmen dann die Sichtbarkeit der Objektteile für jeden Pixel des Bildes. Der wichtigste (heute fast ausschließlich eingesetzte) Projektionsalgorithmus ist der z-Buffer Algorithmus, bei dem für jeden Pixel der Tiefenwert des bis dato nächsten Objektfragmentes gespeichert wird um so das am nächsten zum Betrachter gelegene Objektfragment zu bestimmen. Strahlverfolgungsalgorithmen dagegen arbeiten alle Pixel nacheinander ab und bestimmen jeweils das erste Objekt, das einen Sehstrahl vom Auge durch einen Pixel schneidet. Die Suche nach dem ersten Objekt kann dabei durch räumliche Suchdatenstrukturen beschleunigt



Abbildung 1: Verschiedene Verfahren zur Reduktion der Szenenkomplexität

werden, so daß man in der Regel eine stark output-sensitive Laufzeit erhält. Leider ist der Grundaufwand für die Objektsuche auch bei hoch-optimierter Implementation immer noch so groß, daß eine Echtzeitdarstellung nur bedingt möglich ist [Wal 01].

Projektionsalgorithmen können dagegen sehr effizient (auch mit weitgehender Hardwareunterstützung) implementiert werden, so daß prinzipiell eine Echtzeitdarstellung möglich wird. Aufgrund der objektweisen Arbeitsweise benötigen solche Algorithmen allerdings linearen Aufwand (Aufwand linear in der Anzahl von Objekten, aus denen die Szene besteht). Eine Echtzeitdarstellung von sehr komplexen Szenen ist daher zunächst nicht möglich. Um auch solche Fälle handhaben zu können, muß der klassische Ansatz des Projektionsalgorithmus erweitert werden: Bevor die Objekte verarbeitet werden, durchlaufen sie ein oder mehrere Filterstufen, die die Komplexität der Szene soweit reduzieren, daß eine ausreichend schnelle Darstellung garantiert werden kann.

Für solche Filterstufen gibt es zwei grundlegende, orthogonale Ansätze: Der erste ist die *Vereinfachung* von Objekten. In der Regel werden dazu *Multiskalenrepräsentationen* verwendet. Dabei wird eine Datenstruktur vorgehalten, in der die Objekte der Szene in verschiedenen Auflösungsstufen repräsentiert sind. Für die Darstellung wird dann die Auflösung der Modelle an den zur Darstellung notwendigen Detaillierungsgrad angepaßt. Der zweite Ansatz ist die konservative Verdeckungsrechnung („*occlusion culling*“). Dabei werden nichtsichtbare Szenenteile identifiziert und von vornherein von der Bearbeitung ausgeschlossen. In dieser Arbeit wird nur das Vereinfachungsproblem betrachtet. Die vorgestellten Ansätze können jedoch recht einfach mit bekannten Verdeckungsrechenalgorithmen kombiniert werden, um auch Szenen mit hohem Verdeckungsgrad effizient verarbeiten zu können.

Der klassische Ansatz für Vereinfachung von Szenen ist die *automatische Netzvereinfachung* (siehe z.B. [PS 97] für einen ausführlicheren Überblick). Dabei wird angenommen, daß die Objekte der Szene durch Netze von Primitiven (meist Dreiecke) beschrieben sind. Diese werden durch automatisches Zusammenfassen vereinfacht und damit dem benötigten Detaillierungsgrad angepaßt (siehe Abbildung 1a, b). Die Haupteinschränkung dieser Methode besteht darin, daß in jedem Zwischenschritt der Vereinfachung ein gültiges Dreiecksnetz bereitgestellt werden muß. Bei einer sehr starken Vereinfachung von Objekten mit komplexer Netztopologie führt dies oft zu Problemen. Eine Klasse von alternativen Ansätzen wird als „*bildbasierte Darstellung*“ („*image based rendering*“) bezeichnet. Hier besteht die Idee darin, geometrische Objekte beliebiger

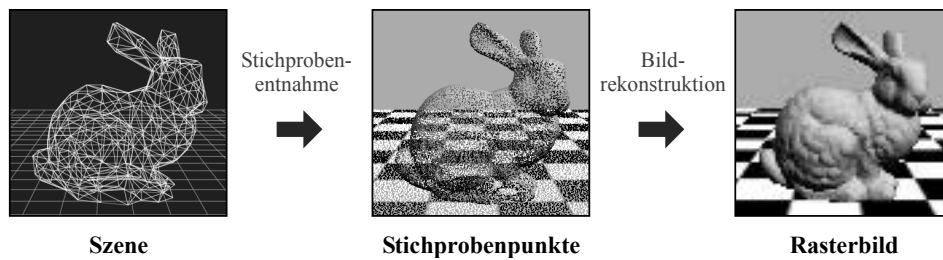


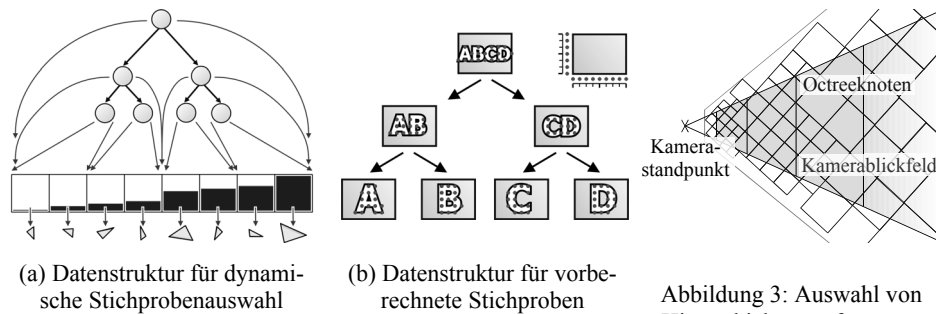
Abbildung 2: Prinzip der punktbasierter Multiskalendarstellung („bunny“-Modell aus [St 05])

Komplexität durch vorberechnete Bilder zu ersetzen [Go 96, LH 96]. Topologische Einschränkungen sind hierbei natürlich unproblematisch, dafür entstehen allerdings neue Probleme: Da nur Bilder für diskrete Ansichten der Szene verfügbar sind, treten Parallaxefehler in den Darstellungen auf die nur durch eine Verfeinerung der Diskretisierung behoben werden können. Dies führt allerdings im Gegenzug wiederum zu einem hohen Speicheraufwand, der eine Anwendung auf große Szenen praktisch ausschließt.

1.2 Punktbasierter Multiskalenmodelle

In dieser Arbeit wird ein neuer Ansatz zur Vereinfachung komplexer Szenen vorgeschlagen: Punktbasierter Multiskalenmodelle. Das Prinzip ist einfach: Anstatt die ursprünglichen geometrischen Primitive zur Darstellung der Szene zu verwenden, wird zunächst eine Stichprobe von Oberflächenpunkten genommen (Abbildung 2). Die Punkte werden dabei so gewählt, daß sie im späteren Bild annähernd gleichverteilt sind. Dann kann im zweiten Schritt ein Bild aus den Stichprobenpunkten rekonstruiert werden. Dazu kann ein einfacher und effizienter Projektionsalgorithmus verwendet werden, so daß eine Darstellung auch von extrem hoch detaillierten Szenen in Echtzeit möglich wird.

Die Idee, Stichprobenpunkte zur Darstellung von Szenen zu verwenden ist in der Computergraphik nicht neu. Punktprimitive werden bereits seit den 80er Jahren zur Darstellung irregulärer Phänomene wie Wolken, Nebel oder Feuer angewendet (siehe z.B. [Re 83]). Später wurden punktbasierter Repräsentationen von Geometrie auch als allgemeines geometrisches Primitiv [LW 85] angewandt, u.a. auch zur Umgehung von Parallaxeproblemen beim bildbasierter Rendering (z.B. [GD 98, Sh 98]). In dieser Arbeit wurden zum ersten mal Punktwolken mit räumlich adaptiver Auflösung zur Repräsentation komplexer Szenen angewandt und untersucht, wobei frühere Ansätze verallgemeinert wurden: Ansätze für hierarchische Szenenrepräsentationen gehen bereits auf Clark [Cl 76] zurück, die in dieser Arbeit verwendeten Datenstrukturen (und ihre Analyse) haben Ähnlichkeit zu Hierarchien aus farbigen, halbtransparenten Boxen, die von Chamberlain et al. [Ch 96] beschrieben wurden. Parallel zu dieser Arbeit wurden ähnliche Vorschläge auch von anderen Autoren gemacht [Pf 00, RL 00]. Einige Ideen aus diesen Arbeiten wurden, wie später beschrieben, im Verlauf der Arbeit aufgenommen und integriert. Im folgenden sollen nun verschiedene Vorschläge für punktbasierter Multiskalendarstellungen näher erläutert werden.



(a) Datenstruktur für dynamische Stichprobenauswahl

(b) Datenstruktur für vorberechnete Stichproben

Abbildung 4: Datenstrukturen für die Erzeugung von Stichprobenpunkten

Abbildung 3: Auswahl von Hierarchieknoten für gegebene Kameraparameter

2 Datenstrukturen

Ziel punktbasierter Multiskalendarstellungsverfahren ist es, die Szene so mit Stichprobenpunkten abzutasten, daß diese später im Bild gleichmäßig verteilt sind (und sich so ein Bild in gleichmäßig guter Qualität rekonstruieren läßt). Dazu muß es möglich sein, gezielt die Stichprobendichte in verschiedenen Regionen der Szene zu variieren. Daher werden zunächst die Primitive, aus denen die Szene besteht (in der Regel Oberflächendreiecke), in einen *Octree* einsortiert. Ein *Octree* ist eine Baumdatenstruktur die dadurch entsteht, daß man einen Würfel um die gesamte Szene solange in 8 Kindwürfel zerteilt, bis in jedem Würfel nur noch eine feste Anzahl an Primitiven liegt. Damit erhält man eine Hierarchie von Gruppen von Primitiven. Je tiefer man in der Hierarchie absteigt, desto enger sind die Primitive räumlich lokalisiert. Für jede solche Gruppe soll nun eine Approximation durch eine Wolke von Stichprobenpunkten bereitgestellt werden. Dazu wurden zwei Möglichkeiten untersucht:

2.1 Dynamische Stichprobenentnahme

Bei der ersten Variante der Datenstruktur wird in jedem Knoten des *Octree* eine Liste aller enthaltenen Primitive gespeichert. Für jede Gruppe von Primitiven wird eine Liste aufsummierter Flächeninhalte bereitgehalten (siehe Abbildung 4a). Mit Hilfe dieser vorberechneten Datenstruktur ist es möglich, für die Gruppe von Primitiven in jedem Knoten der Hierarchie effizient zufällige Stichprobenpunkte auszuwählen, die auf der Oberfläche dieser Primitive gleichverteilt sind. Dazu wird eine Zufallszahl bestimmt im Bereich der Flächeninhaltssummen des Knotens. Dann wird das erste Primitiv ausgewählt (und ein zufälliger Punkt auf diesem Primitiv bestimmt), dessen Flächeninhaltssumme größer ist als diese Zahl. Dadurch ist die Wahrscheinlichkeit, das Primitiv auszuwählen, proportional zu seinem Flächeninhalt. Die Suche benötigt lediglich $O(\log n)$ Zeit für n Primitive, da eine binäre Suche angewandt werden kann. Damit ist die Zeit für die Stichprobenwahl nur sehr schwach von der Szenenkomplexität abhängig (der Logarithmus wächst nur sehr schwach mit steigendem n) – eine wichtige Voraussetzung um extrem detaillierte Szenen bearbeiten zu können.

2.2 Vorberechnete Stichproben

Alternativ zu den Auswahllisten für zufällige Stichprobenentnahme können die Stichproben auch fest vorberechnet werden, und selbst in der Hierarchie gespeichert werden. Diese Methode ist zwar weniger flexibel in der Zuteilung von Stichprobenpunkten an Gruppen von Objekten, dafür ist die Stichprobenentnahme aber effizienter. Diese Idee geht auf die Arbeit von Pfister et al. [Pf 00] zurück, die parallel zur dem Verfahren mit dynamischer Stichprobenauswahl entwickelt wurde. Das Prinzip ist in Abbildung 4b dargestellt: Für jeden Würfel in der Octree-Hierarchie wird ein fester Bruchteil der Seitenlänge als Punktabstand festgelegt und Stichprobenpunktmengen mit diesem Abstand vorberechnet. Dadurch, daß die Boxen mit jeder Hierarchiestufe in ihrer Seitenlänge halbiert werden, erhält man so beim Abstieg in die Hierarchie immer genauere Approximationen der Geometrie. Die Hierarchie endet, sobald die Kosten für die Approximation die der ursprünglichen Geometrie übersteigen (z.B.: jedes Dreieck mit mehr als drei Punkten wird nicht weiter mit Punkten abgetastet sondern selbst in der entsprechenden Hierarchiestufe abgelegt).

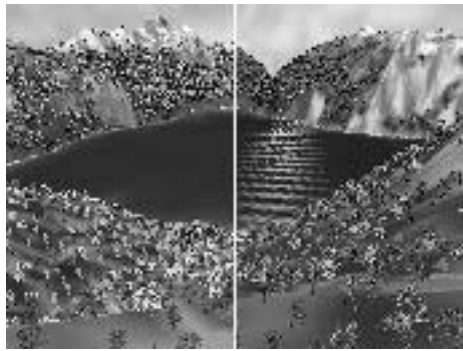
Da während der Vorbereitung mehr Zeit zur Verfügung steht als während der Echtzeitdarstellung, können aufwendigere Verfahren als eine rein zufällige Auswahl zur Verteilung der Stichprobenpunkte benutzt werden. Hier wurden verschiedene Varianten untersucht. Zufällige Stichprobenmengen können z.B. verbessert werden, indem man zu jedem Punkt eine Nachbarschaft bestimmt und überflüssige Punkte (d.h. Punkte, die sehr dicht an anderen Nachbarn liegen) löscht. Die Stichprobendichten, die notwendig sind, um Flächen sicher abzudecken, können damit um eine Größenordnung reduziert und die Darstellung entsprechend beschleunigt werden.

3 Projektionsverfahren

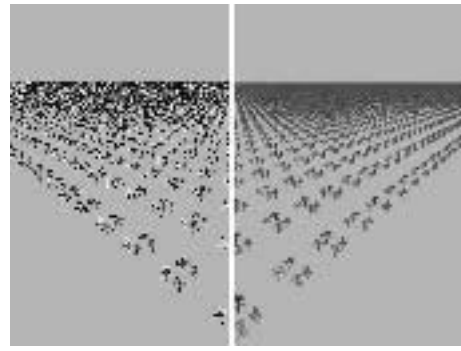
3.1 Der Darstellungsalgorithmus

Mit den Datenstrukturen kann nun ein effizienter Projektionsalgorithmus zur Darstellung komplexer Szenen konstruiert werden. Das Vorgehen ist dabei recht einfach: Dazu sei ein Kamerastandpunkt gegeben, der sich in Echtzeit durch eine große Szene bewegt, für die eine der oben beschriebenen Datenstrukturen vorberechnet wurde. Nun wird, beginnend mit dem Wurzelknoten der Hierarchie, folgender Algorithmus rekursiv ausgeführt¹: Zunächst wird geprüft, ob der entsprechende Begrenzungswürfel des Octreeknötens im Blickfeld der Kamera liegt. Ist dies nicht der Fall, bricht die Rekursion ab. Anderenfalls wird geprüft, ob die gewünschte Stichprobendichte bereits erreicht ist. Dies ist der Fall, wenn die Stichprobenpunkte nach Projektion auf den Bildschirm einen geringeren Abstand als ein Pixel aufweisen. Bei genügend dichten Punkten werden diese auf den Bildschirm gezeichnet (ein Tiefentest für jeden gezeichneten Pixel sorgt dafür, das verdeckte Flächen nicht durchscheinen); anderenfalls wird die Rekursion fortgesetzt.

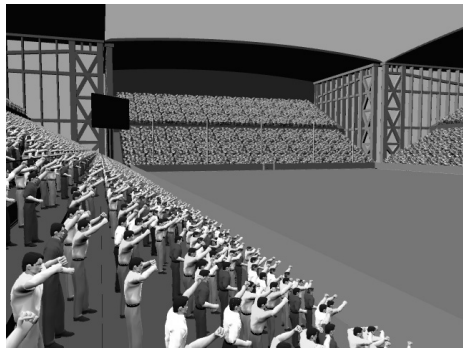
¹ Der Einfachheit halber wird hier der Algorithmus für vorberechnete Stichprobenpunkte beschrieben. Bei dynamischer Stichprobenauswahl ist das Vorgehen ähnlich, siehe [Wan 01].



a) Landschaftsszene: $4 \cdot 10^8$ Dreiecke, dynamische Stichprobenentnahme, Rechenzeit 1/10-20sec, je nach Verfahren und Bildqualität.



b) Eine extrem komplexe Szene: 10^{15} Primitive, Rechenzeiten von 0,25-120 sec, je nach Verfahren und Bildqualität.



c) Animierte Massenszene (vorberechnetes Sampling): Stadion mit Fußballfans, 10^8 Dreiecke. Darstellungsgeschwindigkeit: 9 Bilder/sec.



d) Animierte Massenszene (vorberechnetes Sampling): Verhaltenssimulation einer Tierherde $4,2 \cdot 10^7$ Dreiecke. Darstellungsgeschwindigkeit: 8 Bilder/sec.

Abbildung 5: Projektionsverfahren – Ergebnisse (alle Bilder in 640×480 Pixeln Auflösung, 1.5Ghz Pentium M System mit nVidia GeForce FX Go5650 Grafikkarte)

3.2 Analyse

Die Laufzeit dieses Algorithmus läßt sich analytisch abschätzen. In beiden Varianten (dynamisch und vorberechnete Stichprobenentnahme) belaufen sich die Kosten für die Traversierung der Hierarchie auf $O(\log \tau + h)$. Dabei bezeichnet τ das Verhältnis zwischen Durchmesser der Szene und dem minimalen Betrachterabstand (da die perspektivische Abbildung in der Kameraebene singularär ist, muß immer ein minimaler Betrachterabstand definiert werden). τ selbst kann zwar sehr groß werden, geht aber nur logarithmisch ein. h ist die Höhe des Octrees, in der Praxis auch ein sehr kleiner Wert (typischerweise logarithmisch in der Anzahl der Primitive). In der Praxis spielen diese Kosten daher keine wesentliche Rolle. Entscheidend sind die Kosten für die Stichprobepunkte. Diese betragen $O(a)$ für vorberechnetes und $O(a \log a \log n)$ für dynamisches

Stichprobenentnahme. a ist dabei die projizierte Fläche aller sichtbarer und verdeckter Objekte auf dem Bildschirm (hier muß zusätzlich noch angenommen werden, daß die Orientierungen von Objekten und die Blickfelder der Kamera gleichverteilt sind, um pathologische Fälle auszuschließen). Bei dynamischer Stichprobenbestimmung geht die projizierte Fläche als $a \log a$ ein, da zufällige Stichproben genommen werden: Um sicher zu gehen, daß a Pixel auch sicher mit Stichproben gefüllt werden können und keine Löcher bleiben, müssen $O(a \log a)$ Punkte gezeichnet werden (dieses Phänomen ist als Coupon Collector's Problem in der Literatur bekannt). Letztlich geht noch der Logarithmus der Szenenkomplexität (Anzahl der Primitive) n ein, da Stichproben mit binärer Suche bestimmt werden. Aus diesen Gründen ist das dynamische Stichprobenverfahren auch in der Praxis etwas langsamer; allerdings lassen sich die Nachteile durch geschicktes Caching in gewissem Maße ausgleichen. Zusammenfassend kann man sagen, daß die Darstellungszeit mit Hilfe der neuen Verfahren im wesentlichen nur noch von der projizierten Fläche (durch Objekte bedeckte Pixel) abhängt, Szenenkomplexität und Ausdehnung spielen fast keine Rolle mehr (die logarithmischen Einflüssen sind in der Praxis von geringer Auswirkung).

3.3 Ergebnisse

Wie gut funktionieren die Verfahren tatsächlich in der Praxis? Abbildung 5 zeigt einige Darstellungen hoch komplexer Szenen mit Hilfe von punktbasierten Multiskalenprojektionsverfahren. Abbildung a) und b) zeigen komplexe Landschaftsszenen, die mit der neuen Technik in Echtzeit dargestellt werden können. Die Laufzeit hängt im wesentlichen nur von der Bildqualität und nicht von der Szenenkomplexität ab. Man beachte, daß die Darstellung von Szene b) mit gewöhnlichen (linearzeit) Projektionsverfahren auch auf der schnellsten derzeit verfügbaren PC-Graphikhardware mehrere Tage gedauert hätte. Abbildung 5 c) und d) zeigen eine Ergebnisse einer Erweiterung: Die Datenstruktur für vorberechnete Stichprobenpunkte läßt sich sehr leicht auf animierte Szenen verallgemeinern [WS 02]. Dabei können allgemeine „Keyframeanimationen“ (hier: Abfolgen von Dreiecksnetzen beliebiger Topologie mit bekannter Interpolation über die Zeit) behandelt werden. Die Garantien an Laufzeit und Bildqualität bleiben dabei (unter einigen praxisnahen Annahmen über die Animation) erhalten. Die Szene in Abbildung c) besteht aus hundert Millionen Dreiecken, die sich bewegende Fußballfans darstellen. Sie kann mit 9 Bildern pro Sekunde dargestellt werden (die verwendete Hardware kann an sich nur etwa 30 Millionen Dreiecke pro Sekunde verarbeiten, hätte ohne Multiskalendarstellung also etwa 30 mal länger für die Darstellung gebraucht). Ein ähnliches Beispiel ist in Szene d) dargestellt: Hierbei wird eine Herde aus 1.300 Tieren simuliert, wobei jedes aus etwa 30.000 Dreiecken modelliert ist und sich selbstständig durch die Szene bewegt. Auch hier ermöglicht das Multiskalenverfahren eine Echtzeitdarstellung.

4 Erweiterungen

Strahlverfolgungsverfahren: Die punktbasierten Multiskalenmethoden lassen sich auch anwenden, um Strahlverfolgungsmethoden zu beschleunigen. Dabei werden auch Sekundärstrahlen weiterverfolgt [Fo 96] um Spiegelungen, Brechungen und Schatten zu

berechnen. Das technische Hauptproblem ist nun, daß die Dichte der Stichprobenpunkte nun nicht mehr einfach vom Abstand zum Betrachter abhängt sondern komplex in der Szene variieren kann (z.B. wenn Strahlen durch ein Brennglas fallen). Dieses Problem kann gelöst werden, indem die partiellen Ableitungen von Strahlschnittpunkten nach den Bildschirmkoordinaten betrachtet werden, um die Strahlaufweitung und damit die notwendige Auflösung festzulegen [Ig 99]. In der Praxis sind die Laufzeitgewinne durch eine Multiskalendarstellung der Szene nicht so dramatisch wie bei Projektionsverfahren, da Strahlverfolgungsverfahren durch die verwendeten Suchdatenstrukturen von vornherein eine stärker output-sensitive Laufzeit aufweisen. Dennoch lassen sich vor allem Spezialeffekte wie weiche Schatten, Antialiasing oder unscharfe Reflektionen von rauen Oberflächen deutlich effizienter behandeln [WS 03a].

Volumendarstellung: Die hier vorgeschlagenen hierarchischen Datenstrukturen lassen sich auch auf halbtransparente Volumenmodelle anwenden, wie sie oft in der Medizin zur Visualisierung von Tomographiedaten auftreten. Eine ähnliche Analyse wie im Fall von reinen Oberflächenmodellen zeigt, daß eine solche Darstellung die Laufzeit von linearer auf logarithmische Zeit verringern kann. Die Hierarchien können in einer komprimierten Waveletdarstellung repräsentiert werden, die den Speicherbedarf für die Volumenmodelle drastisch reduzieren. Insgesamt lassen sich mit dieser Methode sehr große Datensätze in Echtzeit darstellen, für die dies zuvor als unmöglich galt. So konnte z.B. erstmals der bekannte „*Visible Human*“ Datensatz (mehrere Gigabyte Rohdaten) bei guter Bildqualität in Echtzeit visualisiert werden [Gu 02].

Soundrendering: Eine weitere, auf den ersten Blick scheinbar nicht sehr naheliegende Anwendung ist die Wiedergabe von Ton in komplexen Szenen mit vielen Tonquellen. Hier kann fast das gleiche stichprobenbasierte Darstellungsverfahren wie für die Bilderzeugung angewandt werden, um den Toneindruck eines sich bewegenden Beobachters in Echtzeit zu approximieren. Für diese Schätzung aus Stichproben läßt sich eine stochastische Konvergenz gegen das tatsächliche Tonsignal, unabhängig von der Anzahl der Originaltonquellen, zeigen. Mit diesem Ansatz war zum erstenmal eine Echtzeittondarstellung in Szenen mit Millionen von Tonquellen möglich, wobei der Toneindruck bei Benutzung aktueller PC-Hardware nur geringfügig schlechter als der aller Tonquellen war (die aber unmöglich in Echtzeit wiedergegeben werden könnten) [WS 04].

5 Diskussion und Ausblick

Die hier beschriebene Arbeit stellt einen neuen Ansatz zur Darstellung extrem detailreicher Szenen vor – die Darstellung aus Stichprobenpunkten. Dieses Paradigma erlaubt zum erstenmal extrem komplexe Oberflächenmodelle mit Milliarden von Primitiven und mehr in Echtzeit zu visualisieren. Insbesondere läßt sich das Verfahren auch sehr einfach auf animierte Szenen verallgemeinern, was mit alternativen Ansätzen (z.B. Netzvereinfachung) bisher noch nicht so einfach und in dieser Allgemeinheit möglich ist. Der Stichprobenpunktansatz läßt sich auch in anderen Gebieten anwenden. So lassen sich auch Strahlverfolgungsalgorithmen, Volumendarstellungen und sogar Tondarstellungen in komplexen Szenen mit ähnlichen Ansätzen deutlich beschleunigen.

Die Verwendung von Stichprobenpunkten ist nicht immer die beste Methode: In bestimmten Fällen (z.B. bei der Darstellung glatter Oberflächen) sind z.B. Netzvereinfachungsmethoden durchaus deutlich effizienter. Für das neue Paradigma spricht aber die Einfachheit und Allgemeinheit: Die punktbasierten Verfahren lassen sich oft auch dann noch einsetzen, wenn klassische Verfahren nicht mehr anwendbar sind. In vielen Fällen (z.B. Szenen mit komplexer Netztopologie, animierte Szenen, auch für Tondarstellung) gibt es heute noch keine praktikablen Alternativen für eine Echtzeitdarstellung komplexer Szenen. Die Verwendung von Mengen von Stichprobenpunkten kann auch in anderen Gebieten als der reinen Szenendarstellung hilfreich sein, z.B. bei der Simulation bestimmter globaler Beleuchtungsphänomene [WS 03b].

Für weitere Arbeiten in dieser Richtung wird sicherlich ein wichtiges Augenmerk auf dem Aspekt der Modellierung liegen. Derzeit werden die komplexe Szenen (alle hier gezeigten komplexen Beispiele) durch Instantiierung beschrieben, d.h. aus vielen Instanzen gleicher Objekte und Objektgruppen zusammengesetzt. Die damit beschreibbaren Szenen sind natürlich noch eingeschränkt. Wünschenswert wäre, auch Szenen ohne strukturelle Redundanz handhaben zu können. Ein Ansatz, der bereits untersucht wurde, ist hierbei die Auslagerung der Szenendatenstrukturen auf Hintergrundspeicher [K1 02]; darüber hinaus kommen aber noch viele weitere Methoden zur effizienten Modellierung großer Szenen in Betracht (wie z.B. prozedurale Modellierung [Fo 96]). Neben der effizienten Codierung sind auch neue Modellierungswerkzeuge notwendig, damit komplexe Szenen auch noch effizient von menschlichen Modellierern erstellt werden können und die neuen Darstellungsmöglichkeiten damit in der Praxis voll nutzbar werden.

Literaturverzeichnis

- [Cl 76] Clark, J.H.: Hierarchical geometric models for visible surface algorithms. In: Communications of the ACM, 19(10), 547-554, 1976.
- [Ch 96] Chamberlain, B., DeRose, T., Lischinski, D., Salesin, D., Snyder, J.: Fast Rendering of Complex Environments Using a Spatial Hierarchy. In: Proc. Graphics Interface '96, 132-141, 1996.
- [Fo 96] Foley, J. D.; van Dam, A.; Feiner, S. K.; Hughes, J. F.: Computer Graphics: principals and practice, second edition in C. Addison-Wesley, 1996.
- [GD 98] Grossman, J. P., Dally, W. J.: Point Sample Rendering. In: Rendering Techniques '98, 181-192, Springer, 1998.
- [Go 96] Gortler, S.J., Grzeszczuk, R., Szeliski, R., Cohen, M.F.: The Lumigraph. In: SIGGRAPH 96 Conference Proceedings, 43-54, 1996.
- [GH 97] Garland, M.; Heckbert, P.S.: Surface simplification using quadric error metrics. In: SIGGRAPH 97 Proceedings, 209 - 216, 1997.
- [Gl 91] Glassner, A. (editor): An Introduction to Ray Tracing, Academic Press, 4th printing, 1991.
- [Gu 02] Guthe, S., Wand, M., Gonser, J., Straßer, W.: Interactive Rendering of Large Volume Data Sets. In: Visualization 2002 Conference Proceedings, 2002.
- [Ig 99] Igehy, H.: Tracing Ray Differentials. In: SIGGRAPH 99 Proceedings, 179 – 186, 1999.

- [Kl 02] Klein, J., Krokowski, J., Fischer, M., Wand, M., Wanka, R., Meyer auf der Heide, F.: The Randomized Sample Tree: A Data Structure for Interactive Walkthroughs in Externally Stored Virtual Environments. In: Proceedings of ACM Symposium on Virtual Reality Software and Technology (VRST), 2002.
- [LH 96] Levoy, M., Hanrahan, P.: Light Field Rendering. In: SIGGRAPH 96 Proceedings, Annual Conference Series, 31-42, 1996.
- [LW 85] Levoy, M., Whitted, T.: The Use of Points as a Display Primitive. Technical report, University of North Carolina at Chapel Hill, 1985.
- [Pf 00] Pfister, H., Zwicker, M., van Baar, J., Gross, M.: Surfels: Surface Elements as Rendering Primitives. In: SIGGRAPH 2000 Proceedings, Annual Conference Series, 335-342, 2000.
- [PS 97] Puppo, E.; Scopigno, R.: Simplification, LOD and Multiresolution Principals and Applications. In: EUROGRAPHICS '97 Tutorial Notes, 1997.
- [Re 83] Reeves, W. T.: Particle Systems – A Technique for Modeling a Class of Fuzzy Objects. In: Computer Graphics (SIGGRAPH 83 Proceedings), 17(3), 359-376, 1983.
- [RL 00] Rusinkiewicz, S., Levoy, M.: Qsplat: A Multiresolution Point Rendering System for Large Meshes. In: SIGGRAPH 2000 Proceedings, 343 – 352, 2000.
- [Sh 98] Shade, J.W., Gortler, S. J., He, L., Szneliski, R.: Layered Depth Images. In: SIGGRAPH 98 Proceedings, 231–242, 1998.
- [St 05] Stanford Computer Graphics Laboratory: The Stanford 3D Scanning Repository, 2005. <http://graphics.stanford.edu/data/3Dscanrep/>
- [Wal 01] Wald, I., Benthin, C., Wagner, M., Slusallek, P.: Interactive Rendering with Coherent Ray-Tracing. In: Computer Graphics Forum (EUROGRAPHICS 2001 Proceedings), 20(3), 153-164, 2001.
- [Wan 01] Wand, M., Fischer, M., Peter, I., Meyer auf der Heide, F., Straßer, W.: The Randomized z-Buffer Algorithm: Interactive Rendering of Highly Complex Scenes. In: SIGGRAPH 2001 Proceedings, Annual Conference Series, 361 – 370, 2001.
- [WS 02] Wand, M., Straßer, W.: Multi-Resolution Rendering of Complex Animated Scenes, In: EUROGRAPHICS 2002 Conference Proceedings (best student paper award), 2002.
- [WS 03a] Wand, M., Straßer, W.: Multi-Resolution Point-Sample Raytracing. In: Graphics Interface 2003 Conference Proceedings, 2003.
- [WS 03b] Wand, M., Straßer, W.: Real-Time Caustics. In: EUROGRAPHICS 2003 Conference Proceedings, 2003.
- [WS 04] Wand, M., Straßer, W.: A Real-Time Sound Rendering Algorithm for Complex Scenes. Technical report WSI-2003-5, Wilhelm Schickard Institute for Computer Science, Graphical-Interactive Systems (WSI/GRIS), University of Tübingen, 2003.

Michael Wand wurde am 28. November 1974 in Paderborn geboren. Nach Schulausbildung (Abitur Juni 1994) und Zivildienst (bis Oktober 1995) folgte ein Studium der Informatik an der Universität-GH Paderborn. Abschluß mit Diplom im März 2000. Seit Juni 2000 ist er wissenschaftlicher Mitarbeiter am Lehrstuhl für graphisch-interaktive Systeme (Prof. Straßer) der Universität Tübingen, wo er im Juni 2004 promovierte (Veröffentlichung der Arbeit im Oktober 2004). Hauptforschungsinteressen sind Techniken zur effizienten Modellierung und Darstellung komplexer dreidimensionaler Szenen.