

DIPLOMARBEIT

UNIVERSITÄTSZENTRUM INFORMATIK

Martin-Luther-Universität Halle-Wittenberg

Der Quinary-Space-Partition-Tree

Oliver Heyme

(2005)

Allgemeine Angaben

Die Diplomarbeit wurde in der Arbeitsgruppe "Allgemeine Informatik, Computergraphik" von Doz. Dr. Peter Schenzel, Institut für Informatik, Fachbereich Mathematik und Informatik der Martin-Luther-Universität angefertigt.

Zur Kontaktaufnahme benutzen Sie bitte die Email-Adresse direktor@uzi.uni-halle.de.

Zusammenfassung

Die 3D-Computergrafik ist heutzutage aus der Produktion von Kinofilmen, der Architektur- und Produktvisualisierung und vielen anderen Bereichen nicht mehr wegzudenken. Durch sie ist es mit einfachen Mitteln möglich Szenarien, Objekte und Charaktere welche nicht, oder noch nicht real existieren zu modellieren, zu animieren und darzustellen.

Die Darstellung dieser 3D-Objekte wird als Rendering bezeichnet und stellt eine große Forschungsrichtung der Computergrafik dar. Aus diesem Grund existieren mehrere Verfahren zur Generierung von 2D-Bildern aus 3D-Szenarien, die meist auf eine Art der Anwendung spezialisiert sind. So gibt es neben dem Rasterization-Verfahren, welches vor allem bei der Realtime-Darstellung von 3D-Szenarien verwendet wird, das Scanline- und Raytracing-Verfahren. Die beiden letztgenannten Algorithmen finden größtenteils Anwendung im Bereich der fotorealistischen Bildsynthese.

Beim Raytracing werden Strahlen von einer virtuellen Kamera aus, mit allen Objekten der 3D-Szene auf Schnittpunkte geprüft. Von den gefundenen Intersections wird diejenige mit dem geringsten Abstand zum Strahlursprung ausgewählt und schattiert.

Zur exakten Berechnung von Schatten können wieder Strahlen, mit dem zu schattierenden Schnittpunkt als Ursprung und den Positionen der Lichtquellen als Richtung, auf Intersections mit der Szene geprüft werden. Diese Möglichkeit der rekursiven Anwendung des Algorithmus kann zur Lösung der meisten Probleme der fotorealistischen Computergrafik verwendet werden. Die bekanntesten Beispiele sind die Be-

rechnung der physikalisch korrekten Reflektion bzw. Brechung an Oberflächen, sowie die globale Beleuchtung.

Nachteil des Raytracing-Verfahrens ist die große Zahl an durchzuführenden Schnittpunktberechnungen. So muss, bei einer naiven Herangehensweise, jeder Strahl mit allen Objekten der Szene getestet werden. Das kann bei einem 2D-Bild mit einer Auflösung von x mal y Pixeln, einer 3D-Szene mit n Objekten und l Lichtquellen zu $(x \cdot y \cdot n) \cdot (l + 1)$ Intersectionberechnungen führen. Um diesen Nachteil auszugleichen werden Beschleunigungsstrukturen verwendet. Diese sollen die Anzahl der Schnittpunktberechnungen auf einen wesentlichen geringeren Aufwand pro Strahl reduzieren. Die Idee ist dabei, die Szene vor dem eigentlichen Rendering in kleinere Unterszenen aufzuteilen und beim Prüfen eines Strahles nur Objekte zu testen, die in vom Strahl getroffenen Unterszenen liegen.

Ein großer Nachteil der derzeit existierenden Beschleunigungsstrukturen ist die Festlegung auf statische 3D-Szene. Das heißt, bei animierten Objekten in der Szene muss die gesamte, oder zumindest ein Teil der Beschleunigungsstruktur neu berechnet werden. Bei der Berechnung von Bildern und Animationen mittels Raytracing verbraucht das Generieren und Abspeichern von Beschleunigungsstrukturen zusätzliche Ressourcen. Diese Problematik ist einer der Hauptgründe, warum für umfangreiche 3D-Animationsfilme immer noch das Scanline-Verfahren vorrangig genutzt wird, welches keine solche Struktur benötigt. So wird bei den derzeit erfolgreichsten 3D-Animationsfilmen wie z.B. „Die Unglaublichen“ und „Findet Nemo“ der Scanline-Renderer Pixar RenderMan® verwendet.

Es besteht beim Raytracing komplexer 3D-Szene die Forderung nach speichereffizienten Beschleunigungsstrukturen, die sich schnell erstellen lassen, wenig Speicher verbrauchen und trotzdem eine akzeptable Renderinggeschwindigkeit bieten.

Im Rahmen dieser Diplomarbeit wird eine Modifikation der Beschleunigungsstruktur Binary-Space-Partition-Tree (BSP) untersucht. Der BSP teilt eine Szene durch rekursive Anwendung entlang einer Teilungsebene in zwei Unterszenen gleicher Größe.

Der davon abgeleitete Quinary-Space-Partition-Tree (QSP) verwendet pro Rekursion vier statt einer Teilungsebene und teilt somit die Szene in einem Schritt in fünf statt zwei Unterszenen (Voxel) auf.

Ziel war die Entwicklung und Implementierung von effizienten Algorithmen zum Erstellen und Durchlaufen der neuen Struktur. Dabei wurden die Möglichkeiten moderner CPUs beachtet und bestmöglich ausgenutzt. Die Implementierung wurde auf Basis des Raytracing-Renderers rayGina durchgeführt und als Zusatzmodul in das System integriert, ohne Eingriffe oder Änderungen am Kern vorzunehmen.

Der Quinary-Space-Partition-Tree erreicht bei geringeren Baumtiefen und somit niedrigerem Speicherverbrauch gute bis sehr gute Renderingzeiten. Kann aber bei großen 3D-Szenen, durch den wesentlich komplexeren Algorithmus zum Durchlaufen der Struktur nicht an die Bestzeiten des Binary-Space-Partition-Tree heranreichen. Dennoch stellt er eine ressourcenschonende Alternative dar.

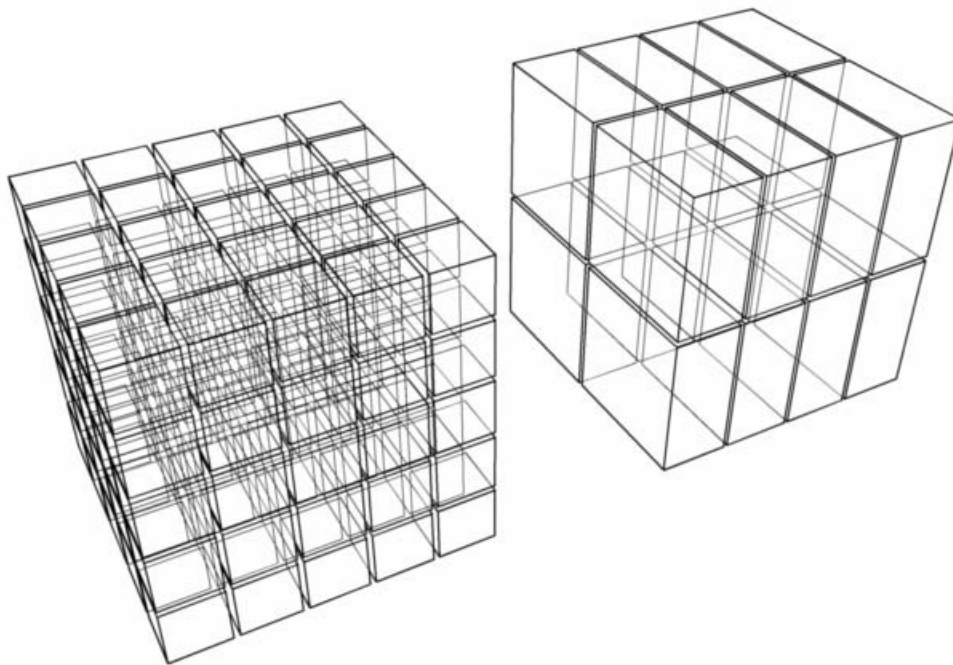


Abbildung 1: Vergleich der Struktur bei gleichem Speicherverbrauch
(links QSP 125 Unterteilungen, rechts BSP 16 Unterteilungen)