Raytracing

Tobias G. Pfeiffer

Berlin, 25. November 2007

Inhaltsverzeichnis

1	Einführung Grundlagen des Renderings Das Raytracing-Verfahren			1
2				2
3				3
	3.1	Schnittpunkt- und Normalenbestimmung		
		3.1.1	Kugel	5
		3.1.2	Dreieck	5
		3.1.3	Achsenparalleler Quader	6
		3.1.4	Auswählen des sichtbaren Objekts	7
	3.2	Farbbe	estimmung eines Punktes	8
		3.2.1	Direkter Lichteinfall	8
		3.2.2	Gerichtete Reflexion und Transmission	9
		3.2.3	Diffuse Reflexion und Transmission	12
		3.2.4	Vergleich der Beleuchtungsarten	13
4	Laufzeitbetrachtungen und Optimierung			15
	4.1	4.1 Analyse der Laufzeit		15
	4.2	4.2 Verbesserungen der Laufzeit		15
5	Ausblick			19
Li	Literatur			

1 Einführung

Ein grundlegendes Problem in der Computergrafik ist es, aus einer dreidimensionalen Szene, deren Geometrien mit Materialien und Texturen versehen sind, eine Grafik zu berechnen, die demjenigen Bild entspricht, das ein menschlicher Betrachter von einer bestimmten Position in der Szene aus sehen würde; dieser Vorgang heißt *rendern*. Raytracing (deutsch: "Strahlenverfolgung") ist ein Verfahren, das dieses Problem löst und dabei durch die Verwendung verschiedener physikalischer Gesetze der Lichtausbreitung einen hohen Grad an Realismus erreicht.

Der folgende Text gibt einen Überblick über die allgemeine Funktionsweise des Raytracings und die dort verwendeten Techniken. Zunächst sollen die Grundlagen des Renderings erläutert werden, anschließend wird erklärt, wie die einzelnen Schritte des Verfahrens ablaufen. Danach wird die Laufzeit eines allgemeinen Raytracing-Algorithmus' analysiert und Formen der Optimierung vorgestellt.

2 Grundlagen des Renderings

Ziel des Renderings ist es, ein zweidimensionales Abbild einer dreidimensionalen Szene zu erzeugen, so wie ein Betrachter in der Szene sie sehen würde. Folgende Problemparameter sind also gegeben:

- eine Menge an geometrischen Primitiven (z. B. Polygone, Kugeln, implizit gegebene Flächen), bestimmt durch ihre Koordinaten im Raum, und deren assoziierte Materialien bzw. Texturen
- eine Menge an Lichtquellen mit deren Eigenschaften (Position, Farbe, Intensität, Form)¹
- eine Betrachterposition ("Kameraposition") und Blickrichtung

Es soll als Ausgabe eine Raster-Grafik berechnet werden, die im Idealfall nicht mehr von einer Fotografie der gleichen Szene zu unterscheiden ist.



Abbildung 1: Einfall von Lichtstrahlen in das Auge

Dafür muss zunächst untersucht werden, wie das menschliche Auge Bilder wahrnimmt und die gewonnenen Erkenntnisse entsprechend auf das Gebiet der Computergrafik übertragen werden. Die Funktionsweise des Auges ist in Abbildung 1 skizziert: Licht einer gewissen Wellenlängenzusammensetzung (*Spektrum*) fällt durch die Pupille ein und trifft auf die Netzhaut, auf der sich Rezeptoren für bestimmte Wellenlängen des Lichtes befinden. Diese unterschiedlichen Wellenlängen werden vom Menschen als unterschiedliche Farben wahrgenommen, wobei es mehrere Spektren gibt, die die gleiche wahrgenommene Farbe erzeugen. Ein "Bild" zu berechnen, ist also äquivalent dazu, für jeden Bereich einer Fläche Intensität und Spektrum des dort auftreffenden Lichts zu bestimmen. [DSC93] behandelt

¹Es soll im Folgenden versucht werden, zwischen der *Farbe*, d. h. der Wellenlängenverteilung, und der *Intensität*, d. h. der Lichtstärke, angemessen zu unterscheiden, wo dies möglich ist.



Abbildung 2: Umsetzung der Augenfunktion in der Computergrafik

umfassend die physikalischen Eigenschaften von Licht, die in der Computergrafik gebraucht werden, wie beispielsweise Ausbreitung, Brechung und Farbe.

In der Computergrafik wird aus technischen Gründen die Position von Brennpunkt und Netzhaut vertauscht. Wie in Abbildung 2 gezeigt, wird dazu eine *Bildebene*, die senkrecht auf der *Blickrichtung* des *Blickpunktes* steht, definiert. Eine rechteckige Fläche auf dieser Bildebene wird in der Breite und in der Höhe in kleine Quadrate unterteilt, die den Pixeln auf dem zu berechnenden Bild entsprechen.

Die Aufgabe eines Renderverfahrens ist es nun, für jeden dieser Pixel einen Farbwert zu berechnen, der sich aus den in diesem Bereich sichtbaren Objekten ergibt. Da das, was der Mensch als "Farbe" wahrnimmt, sich jedoch ausschließlich aus den im Auge ankommenden Lichtwellen verschiedener Wellenlänge ergibt, bedeutet dies, dass im Wesentlichen die Lichtausbreitung in der modellierten Szene berechnet werden muss.

3 Das Raytracing-Verfahren

Das Raytracing-Verfahren basiert darauf, die Ausbreitung der Lichtstrahlen in der Szene zu verfolgen; es ist damit ein *globales Beleuchtungsmodell*. Beim *Forward-Raytracing* wird von der Lichtquelle ausgehend das ausgesandte Licht verfolgt, auch über mehrere reflektierende Oberflächen hinweg, bis es entweder auf der Bildebene ankommt, sich aus der Szene hinaus bewegt oder zu schwach wird, um irgendeinen Einfluss auf das Bild haben zu können. Da nur ein winziger Bruchteil der anfangs ausgesandten Lichtstrahlen jemals im Bild ankommt, wird dieses Verfahren in der Praxis nicht verwendet.

Wenn allgemein vom Raytracing-Verfahren die Rede ist, ist damit in der Regel das *Backward-Raytracing* gemeint. Dies bedeutet, dass vom Blickpunkt aus der eintreffende Lichtstrahl rückwärts verfolgt und dessen Ursprung auf einem Objekt in der 3D-Szene bestimmt wird. Anschließend kann durch verschiedene Verfahren, unter anderem durch weitere Aussendung von Strahlen und Einsatz eines *lokalen Beleuchtungsmodells* (siehe Kapitel ??), die Farbe und Intensität des in Richtung des Blickpunktes ausgesandten Lichtes von diesem Punkt aus berechnet werden, die dann der Farbe an dem jeweiligen Bildpixel entspricht.

3.1 Schnittpunkt- und Normalenbestimmung

Zunächst muss für jeden Pixel dasjenige Objekt bestimmt werden, das vom Blickpunkt aus in Richtung dieses Pixels "gesehen" wird. Sei dazu der Blickpunkt $v \in \mathbb{R}^3$, die Blickrichtung $d \in \mathbb{R}^3$ und die Bildebene durch

$$E := \left\{ b + \lambda r + \mu u \mid \lambda \in [0, w_{\max}], \mu \in [0, h_{\max}] \right\}$$

gegeben (siehe Abbildung 3).



Abbildung 3: Bildpunkt und Bildebene: Bezeichnungen

Dabei legt *b* legt die untere linke Ecke des Bildausschnitts im Raum fest, $u \in \mathbb{R}^3$ ist der Vektor, der nach oben zeigt, (um verschiedene Rotationen der Kamera zu erlauben) und *r* ergibt sich als Kreuzprodukt von *u* und *d*. h_{max} und w_{max} sind die gewünschten Pixelanzahlen in der Höhe bzw. der Breite. Dann entspricht ein Bildpixel (x, y) gerade der Punktmenge $\{b + \lambda r + \mu u \mid \lambda \in [x, x + 1], \mu \in [y, y + 1]\}$ mit dem Pixelmittelpunkt $m := b + (x + \frac{1}{2})r + (y + \frac{1}{2})u$.

Der vom Blickpunkt durch diesen Pixelmittelpunkt gehende "Blickstrahl" hat

also die Gleichung $v + \lambda(m - v), \lambda \ge 0$. Nun kann für jedes Primitiv der Szene der Schnittpunkt mit diesem Strahl berechnet werden, wenn es einen gibt. Im Folgenden werden für einige Primitivtypen Methoden zur Schnittpunktbestimmung vorgestellt. Da zur Berechnung der Lichtausbreitung später auch die Normalen am Schnittpunkt benötigt werden, wird auch gleich die Bestimmung eines Normalenvektors erklärt.

3.1.1 Kugel

Die Kugel² ist ein sehr beliebtes Motiv in Raytracing-Szenen, da sich der Schnittpunkt eines Strahls mit einer Kugel sehr schnell feststellen lässt. Eine Kugel mit Mittelpunkt $p \in \mathbb{R}^3$ und Radius $r \in \mathbb{R}, r \ge 0$ ist die Punktmenge $\{x \in \mathbb{R}^3 \mid |x - p| = r\}$. Nun gilt $|x - p| = r \Leftrightarrow |x - p|^2 = r^2 \Leftrightarrow |x - p|^2 - r^2 = 0$ und durch Einsetzen der Strahlgleichung $v + \lambda d$ erhält man:

$$0 = |v + \lambda d - p|^{2} - r^{2} = \langle v + \lambda d - p, v + \lambda d - p \rangle - r^{2}$$
$$= \langle v - p, v - p \rangle + 2\lambda \langle v - p, d \rangle + \lambda^{2} \langle d, d \rangle - r^{2}$$
$$= \lambda^{2} |d|^{2} + 2\lambda \langle v - p, d \rangle + |v - p|^{2} - r^{2}$$
$$\iff 0 = \lambda^{2} + \lambda \frac{2 \langle v - p, d \rangle}{|d|^{2}} + \frac{|v - p|^{2} - r^{2}}{|d|^{2}}$$

(Hierbei ist $\langle ., . \rangle$ das euklidische Skalarprodukt.) Dann gilt unter der Voraussetzung |d| = 1 (d. h. mit normiertem Richtungsvektor):

$$\lambda_{1,2} = -\langle v - p, d \rangle \pm \sqrt{\langle v - p, d \rangle^2 - |v - p|^2 - r^2}$$

Der Einheitsnormalenvektor auf einem Punkt x auf einer Kugel ist gerade der normierte Richtungsvektor vom Kugelmittelpunkt p zu diesem Punkt auf der Oberfläche:

$$n = \frac{1}{r}(x - p)$$

3.1.2 Dreieck

Ein Dreieck mit den Eckpunkten $a, b, c \in \mathbb{R}^3$ ist die Menge { $\alpha a + \beta b + \gamma c \in \mathbb{R}^3 | \alpha, \beta, \gamma \ge 0, \alpha + \beta + \gamma = 1$ }, wobei jeder Punkt in diesem Dreieck eindeutig bestimmte Koeffizienten α, β, γ hat; dies nennt man die *baryzentrische Darstellung*.

²Eigentlich geht es hier, mathematisch betrachtet, um eine *Sphäre* oder *Kugelfläche*, es soll jedoch im Folgenden der im allgemeinen Sprachgebrauch eher verbreitete Begriff der Kugel verwendet werden.

Der Strahl ist gegeben durch den Term $v + \lambda d$ mit $\lambda \ge 0$, dabei soll |d| = 1 sein. Dann ist eine Lösung für die Gleichung

$$v + \lambda d = \alpha a + \beta b + \gamma c$$

gesucht. Man kann nun $\gamma = 1 - \alpha - \beta$ einsetzen und umformen in $\alpha(a - c) + \beta(b - c) - \lambda d = v - c$. Dies lässt sich als lineares Gleichungssystem schreiben:

$$\begin{pmatrix} a-c & b-c & -d \end{pmatrix} \begin{pmatrix} lpha \\ eta \\ \lambda \end{pmatrix} = \begin{pmatrix} v-c \end{pmatrix},$$

welches mit den bekannten Verfahren für lineare Gleichungssysteme (z. B. Gauß-Algorithmus) gelöst werden kann. Existiert eine Lösung mit $\alpha, \beta, 1 - \alpha - \beta \ge 0, \lambda > 0$, so schneidet der Strahl das Dreieck, andernfalls geht er außen daran vorbei oder das Dreieck liegt hinter dem Startpunkt des Strahls. Den Schnittpunkt erhält man dann mit $v + \lambda d$, die Normale n, unabhängig von der erhaltenen Lösung, durch das Kreuzprodukt $(a - c) \times (b - c)$. Hierbei muss man jedoch beachten, dass unter Umständen die Normale auf die Rückseite des Dreiecks zeigt, was in der Regel nicht erwünscht ist. Dazu prüfe man, ob $\langle v - c, n \rangle > 0$; andernfalls sollte man das Vorzeichen des Normalenvektors ändern.

Der Fall Strahl–Ebene ist eine Verallgemeinerung des Falls Strahl–Dreieck; man ersetze a-c und b-c durch die beiden die Ebene aufspannenden Richtungsvektoren (dann ist c Stützvektor) und lasse beliebige Werte für α und β zu.

3.1.3 Achsenparalleler Quader

Ein Quader wird im Raytracing-Verfahren häufig als Container für andere Objekte verwendet (z. B. als *Bounding Box*) und man will im Allgemeinen nur wissen, ob der Strahl diese Box schneidet oder nicht. Ein Quader kann als Schnittmenge von drei senkrecht aufeinander stehenden Paaren paralleler Ebenen betrachtet werden. Eine Ebene ist eindeutig bestimmt durch ihre Normale n und den Abstand a zum Ursprung. Betrachtet man nun nur achsenparallele Ebenen, so reicht die Angabe der Normalenrichtung $d \in \{x, y, z\}$ (d. h. die Normale zeigt in Richtung der x-, y- oder z-Achse) und der Abstand zum Ursprung. Ein Ebenenpaar ist dann gegeben durch (a_{d1}, a_{d2}) (sei o.B.d.A. $a_{d1} \le a_{d2}$), die zwischen den beiden Ebenen enthaltene Punktmenge ist $\{p \in \mathbb{R}^3 \mid a_{d1} \le p_d \le a_{d2}\}$ und ein achsenparalleler Quader ist die Schnittmenge dreier Ebenenpaare in verschiedene Koordinatenrichtungen.

Sei $v + \lambda r$ ein Strahl mit einem Intervall [a, b] der möglichen Parameterwerte, wobei dieses mit $[0, \infty[$ initialisiert wird. Nun kann man für jede Koordinatenrichtung das Intervall der Parameterwerte ausrechnen, in dem das von den Ebenen eingeschlossene Gebiet geschnitten wird und das mit dem Strahl gespeicherte Intervall entsprechend verkleinern:

$$\lambda_1 := \frac{v_d - a_{d1}}{r_d} \qquad \qquad \lambda_2 := \frac{v_d - a_{d2}}{r_d} \tag{1}$$

$$\lambda_{\min} := \min(\lambda_1, \lambda_2) \qquad \qquad \lambda_{\max} := \max(\lambda_1, \lambda_2) \qquad (2)$$

$$:= \max(a, \lambda_{\min}) \qquad b := \min(b, \lambda_{\min}) \qquad (3)$$

Ist das Intervall [a, b] nach Behandlung aller drei Koordinatenrichtungen nicht leer, so schneidet der Strahl den durch die Ebenen berandeten Quader im Punkt $v + a \cdot r$.

3.1.4 Auswählen des sichtbaren Objekts

a

Hat man eine Liste von λ_i für den Blickstrahl berechnet, wählt man daraus das kleinste aus, denn Punkte mit größeren Werten für λ sind sicherlich weiter vom Blickpunkt entfernt als Punkte mit kleineren Werten und werden folglich von diesen verdeckt (siehe Abbildung 4). Im Allgemeinen wird es durch numerische Rechenungenauigkeiten so sein, dass der berechnete Punkt *nicht* auf der Oberfläche eines Objekts liegt, sondern etwas davor oder dahinter. Damit bei der weiteren Aussendung von Strahlen nicht das Objekt selbst den ersten Schnitt erzeugt, kann es notwendig werden, das berechnete λ noch ein wenig nach unten zu korrigieren, damit der Punkt nicht *hinter* der Oberfläche liegt.



Abbildung 4: Das erste geschnittene Primitiv ist das vom Betrachter aus sichtbare Objekt

Damit ist dieser Abschnitt, die Bestimmung des sichtbaren Punktes, dessen Farbe nun noch ermittelt werden muss, abgeschlossen. Im Abschnitt 4 finden sich Methoden, diesen Teil des Verfahrens deutlich zu beschleunigen.

3.2 Farbbestimmung eines Punktes

Aus dem vorherigen Abschnitt kennt man den Punkt $p \in \mathbb{R}^3$ auf einem Primitiv $P \subset \mathbb{R}^3$, die Normale $n \in \mathbb{R}^3$ sowie die Richtung $r \in \mathbb{R}^3$, in der der Betrachter auf p sieht, und muss nun die Farbe bestimmen, die von dort in Richtung des Blickpunktes v (also in Richtung -r) ausgesandt wird. Dazu überlegt man sich zunächst, welche Arten von Licht von p ausgehen können:

- Licht, das direkt von platzierten Lichtquellen der Szene auf p fällt und in alle Richtungen reflektiert wird³
- gerichtet reflektiertes Licht, wie z. B. bei einem Spiegel
- gerichtet transmittiertes Licht, wie z. B. bei einer Glaskugel
- diffus reflektiertes Licht, wie auf matten Oberflächen zu beobachten
- diffus transmittiertes Licht, wie z. B. bei einer Milchglasscheibe

Diese Arten des Lichteinfalls sind unterschiedlich kompliziert zu bestimmen und müssen für das Berechnen eines realistisch wirkenden Bildes kombiniert werden.

Hat man für *p* die möglichen Lichtquellen und -einfallsrichtungen berechnet, kann man mittels eines geeigneten Shading-Verfahrens in Abhängigkeit von Spektrum, Intensität, Entfernung, Einfallswinkel und gegebenenfalls weiteren Parametern das ausgesandte Licht berechnen; dies ist jedoch nicht das Thema dieser Arbeit (siehe stattdessen z. B. Kapitel **??**). Im Folgenden soll daher nur gezeigt werden, was die genannten Kategorien des Lichteinfalls auszeichnet und wie man sie berechnen kann.

3.2.1 Direkter Lichteinfall

In der einfachsten Form des Raytracing muss bestimmt werden, von welcher der Lichtquellen Licht am Punkt *p* ankommt. Sicherlich macht es für die Erscheinung eines Objekts einen massiven Unterschied, ob es direkt von einer Lichtquelle beschienen wird oder bezüglich dieser Lichtquelle im Schatten eines anderen, lichtundurchlässigen (*opaken*) Objekts liegt.

³Genau genommen ist die Reflexion von Licht, das direkt von Lichtquellen kommt, nur ein Sonderfall der diffusen Reflexion. Dort ist der Fall gegeben, dass man zufällig weiß, aus welcher Richtung sehr starkes Licht kommt; dies wird aber nach den gleichen physikalischen Gesetzen behandelt wie die diffuse Reflexion im Allgemeinen.

Zu prüfen ist folglich, ob und in welchem Maße Licht von der Menge $L \subset \mathbb{R}^3$ der gegebenen Lichtquellen auf p fällt. Dazu erstellt man für jede Lichtquelle $l \in L$ einen Strahl $p + \lambda(l - p)$ (einen sog. *Schattenstrahl*) und berechnet, ob es ein Objekt gibt, das diesen Strahl für $\lambda \in [0, 1]$ schneidet. Wenn dies so ist, bekommt pkein direktes Licht von l, denn der Weg der Lichtstrahlen von l zu p ist durch ein Objekt versperrt. Andernfalls erhält p direktes Licht von l, das dann teilweise zum Blickpunkt reflektiert wird.

Die Intensität des in Richtung des Blickpunktes ausgesanten Lichts verhält sich dabei nach [Gla89] proportional zum Cosinus des Winkels zwischen Lichteinfall und Oberflächennormale. Hier gilt das *Lambertsche Gesetz* $I = L \cdot A \cdot \cos \alpha$, wobei I die Lichtstärke, L die Leuchtdichte und A die leuchtende Fläche ist. Dies bedeutet, dass die Intensität des ausgesandten Lichts am stärksten ist, wenn die Lichtquelle senkrecht über der Oberfläche steht; wenn sie sehr stark von der Seite strahlt, ist die Intensität geringer (vergl. Abbildung 5).



Abbildung 5: Auswirkung des Einfallswinkels auf die ausgesandte Lichtstärke

3.2.2 Gerichtete Reflexion und Transmission

Aus der Erfahrung weiß man, dass auf spiegelnden Oberflächen der direkte Lichteinfall die Aussendung von Licht in Richtung des Blickpunktes nur bedingt beeinflusst. Steht man in einem bestimmten Winkel vor einem Spiegel, so sieht man nur noch das Licht, das in einem bestimmten anderen Winkel eingefallen ist. Dabei gilt für perfekte Spiegel das bekannte Gesetz "Einfallswinkel = Ausfallswinkel"; dies bedeutet, dass der Winkel α zwischen Betrachtungsrichtung r und Oberflächennormalen n der gleiche ist wie der Winkel β zwischen der Richtung l, aus der das gespiegelte



Abbildung 6: Zu verfolgende Lichtstrahlen bei idealer gerichteter Reflexion

Licht kommen muss, und der Oberflächennormalen, siehe Abbildung 6. Des weiteren liegen die Betrachtungsrichtung, Normalenvektor und Lichtursprungsrichtung in einer Ebene.

Es kann (unter der Prämisse |r| = |n| = 1) mittels der beiden angegebenen Nebenbedingungen $\langle -r, n \rangle = \langle n, l \rangle$ und $\exists \alpha, \beta : l = \alpha r + \beta n$ für den ausgehenden Vektor mit Länge 1 folgende Formel bestimmt werden:

$$l = r - 2 \langle n, r \rangle \cdot n$$

(Die Gültigkeit der Forderungen ist leicht nachzuprüfen, die Herleitung findet sich in [Gla89].)



Abbildung 7: Zu verfolgende Lichtstrahlen bei idealer gerichteter Transmission

Ähnlich geht man bei Objekten vor, die aus einem Material bestehen, das zu einem Teil lichtdurchlässig ist. Auch hier kann das Licht, das von einem Punkt in eine gewisse Richtung ausgesandt wird, bei einem ideal transmittierenden Material aus genau einer Richtung kommen, man nennt das Material dann *transparent*. Dabei tritt das Phänomen der Lichtbrechung auf, d. h. der Winkel α zwischen Betrachtungsrichtung r und der Oberflächennormalen n ist abhängig vom Winkel γ zwischen der Richtung t, aus der das transmittierte Licht kommen muss, und der Oberflächennormalen, siehe Abbildung 7. Wieder liegen die Betrachtungsrichtung, Normalenvektor und Lichtursprungsrichtung in einer Ebene.

Der Grad der Brechung hängt dabei immer von den beiden Materialien ab, an deren Grenze der Lichtstrahl gebrochen wird. Jedes Material hat einen *Brechungsin*dex η , der das Verhältnis der Lichtgeschwindigkeit in diesem Material im Vergleich zur Lichtgeschwindigkeit im Vakuum angibt. Dann gilt nach *Snells Gesetz*

$$\frac{\sin\alpha}{\sin\gamma} = \frac{\eta_r}{\eta_t}$$

Es kann passieren, dass beim Übergang von einem dichteren in ein weniger dichtes Material Totalreflexion auftritt, d. h. der Winkel zwischen einfallendem Licht und der Normalen ist so groß, dass keine Brechung mehr statt findet, sondern das Licht an der Innenseite der Fläche komplett reflektiert wird.

Unter der Vorraussetzung |r| = |n| = 1 erhält man für den ausgehenden Vektor t in das zweite Material hinein die Formel:

$$t = \frac{\eta_r}{\eta_t} \cdot r + \left(\frac{\eta_r}{\eta_t} \cos \alpha - \sqrt{1 + \left(\frac{\eta_r}{\eta_t}\right)^2 \cdot (\cos \alpha - 1)}\right) \cdot n$$

(Die Herleitung findet sich, mit anderen Bezeichnungen, ebenfalls in [Gla89].) Ist der Term unter der Wurzel negativ, so ist dies gerade das Zeichen dafür, dass Totalreflexion auftritt; es ist dann stattdessen die Formel für die ideale Reflexion zu verwenden.

In die jeweils berechnete Richtung l bzw. t kann man nun einen Strahl von p aus schicken und das Objekt finden, das diesen Strahl als erstes schneidet. Anschließend muss das von diesem Objekt am Schnittpunkt in Richtung -l (bzw. -t) ausgesandte Licht berechnet werden. Dies ist offenbar genau das gleiche Problem, wie das Ursprungsproblem, das an p in Richtung -r ausgesandte Licht zu bestimmen, daher kann man hier einen rekursiven Ansatz wählen; dies nennt man dann *rekursives Raytracing*, vergl. Abbildung 8. Der Rekursionsanker kann dann beispielsweise das Verlassen der Szene, eine bestimmte Rekursionstiefe oder ein zu gering werdender Anteil an der Gesamtintensität des von p ausgehenden Lichts sein. Ein Beispiel, bei dem intensiv von der Möglichkeit des rekursiven Raytracings Gebrauch gemacht wurde, ist in Abbildung 9 zu sehen.



Abbildung 8: Beispiel für rekursives Raytracing und dazugehöriger Strahlenbaum



Abbildung 9: Reflexion und Transmission an einem lebensnahen Beispiel (aus [Pov])

3.2.3 Diffuse Reflexion und Transmission

Die im vorigen Abschnitt beschriebene gerichtete Reflexion findet so nur auf glatten, spiegelnden Oberflächen statt. Auf rauen, matten Oberflächen findet stattdessen diffuse Reflexion statt; dies bedeutet, dass einfallendes Licht aus einer Richtung in alle Richtungen gleichmäßig ausgesandt wird. Daraus ergibt sich umgekehrt sofort das Problem, dass nicht eindeutig bestimmbar ist, aus welcher Richtung das Licht kommt, das in Richtung des Betrachters ausgesandt wird. Ein ähnliches Problem tritt auf, wenn ein Material zwar lichtdurchlässig ist, aber nicht die Möglichkeit einer gerichteten Transmission bietet, weil es im Innern viele Partikel gibt, die das Licht umlenken, wie z. B. bei Milchglas; man nennt das Material dann *transluzent*. Auch hier ist das Problem, dass Licht aus vielen Richtungen zum Farbwert am betrachteten Punkt beitragen könnte.

Eine mögliche Herangehensweise an dieses Problem ist, den Einfluss der diffusen Reflexion bzw. Transmission zu vernachlässigen oder durch ein globales Umgebungslicht zu simulieren. Eine Alternative dazu bietet das *stochastische* bzw. *verteilte Raytracing*, das in [Coo89] umfassend beschrieben wird. Die Idee bei diesem Verfahren ist, statt nur eines Strahls in eine feste Richtung mehrere, mehr oder weniger zufällig verteilte Strahlen in *ungefähr* diese Richtung auszusenden und die erhaltenen Werte zu mitteln.

3.2.4 Vergleich der Beleuchtungsarten

Im Folgenden soll veranschaulicht werden, welche Konsequenzen die Berechnung der oben vorgestellten Arten des Lichteinfalls hat. Die verwendeten Bilder wurden aus [Wik] entnommen.



Abbildung 10: *Raytracing nur mit Verdeckungsberechnung (Raycasting)*



Abbildung 11: Raytracing mit jeweils einem Schattenstrahl pro Punkt

In Abbildung 10 sieht man eine Szene, bei der nur Primärstrahlen ausgesandt wurden und die Farbe der Textur als einzige Größe für die Berechnung der Lichtaussendung in Richtung des Blickpunktes herangezogen wurde. Als Folge davon kann man sehen, dass korrekt festgestellt wurde, welches Objekt ein anderes Objekt verdeckt. Jede weitere Tiefeninformation geht jedoch verloren, da es keine Schatten und keine Auswirkungen von Lichtquellen gibt.

In Abbildung 11 wurde zusätzlich von jedem Punkt auf einem getroffenen Objekt ein Schattenstrahl in Richtung der Lichtquelle (bzw. einem Punkt im Zentrum der Lichtquelle) ausgesandt. Man erkennt nun bereits die dreidimensionale Gestalt der Körper und deren Lage relativ zur Lichtquelle. Die Körper werfen Schatten mit harten Kanten, da "im Schatten sein" hier eine binäre Eigenschaft ist, es gibt keinen Halbschatten. An Wänden und Decke des Szene sowie an der Grenze von Licht und Schatten auf den Kugeln erkennt man den Einfluss des Einfallswinkels des Lichts (siehe Abschnitt 3.2.1).





Abbildung 12: *Raytracing mit jeweils einem Schattenstrahl pro Punkt und rekursiver Reflexions-/Transmissionsbestimmung*

Abbildung 13: Raytracing mit rekursiver Reflexions-/Transmissionsbestimmung und teilweise diffuser Reflexion

In Abbildung 12 wurde die hintere Kugel als spiegelnd und die vordere als teils transparent, teils spiegelnd modelliert. Von dem jeweils getroffenen Punkt der Kugel wurden also rekursiv weitere Strahlen im jeweiligen Ausfalls- bzw. Brechungswinkel ausgesandt. In der vorderen Kugel erkennt man die Spiegelung der orangen Wand und der Lichtquelle und die durch Lichtbrechung im Inneren der Kugel umgedrehte blaue Wand, die sich hinter der Kugel befindet. Wände, Decke und Schatten sehen genauso aus wie im vorherigen Bild, da diese nicht perfekt spiegeln und daher keine veränderte Behandlung erfahren.

In Abbildung 13 wurden zusätzlich zu den bisher eingesetzten Techniken zur Berechnung der diffusen Reflexion einige Strahlen in zufällige Richtungen ausge-

sandt. Dies kann man insbesondere an den Schatten mit weichen Rändern erkennen, die dadurch entstehen, dass von Punkten im Randbereich des Schattens einige Lichtstrahlen die Lichtquelle treffen, andere nicht, und die Mittelung der so erhaltenen Farbwerte zu einem Ergebnis zwischen "vollständig im Schatten" und "nicht im Schatten" führt. Auch der Bereich an der Decke um die Lichtquelle herum ist nun deutlich weicher berandet.

4 Laufzeitbetrachtungen und Optimierung

4.1 Analyse der Laufzeit

Raytracing ist ein Verfahren, das hochqualitative Ergebnisse liefern kann, jedoch oft zum Preis einer langen Laufzeit. Zunächst einmal ist (von einer konstanten Anzahl von Strahlen pro Pixel ausgehend) die Anzahl der ausgesandten Primärstrahlen proportional zur Anzahl der Pixel des zu erzeugenden Bildes. Dies bedeutet, dass die Renderzeit direkt von der Auflösung des zu erzeugenden Bildes abhängt.

Ohne weitere Optimierungen muss jeder Strahl gegen alle n in der Szene vorhandenen Objekte getestet werden. Dies gilt nicht nur für die O(xy) Primärstrahlen, sondern auch für alle Sekundärstrahlen, wie zum Beispiel die bei der Reflexionsoder Transparenzbestimmung ausgesandten Strahlen. Es soll daher zunächst die Anzahl s der Sekundärstrahlen bestimmt werden. In jeder Ebene des Strahlenbaums (vergl. Abbildung 8) wird zu jeder Lichtquelle ein Strahl gesandt und ein Reflexionsstrahl und ein Transmissionsstrahl erzeugt. Die letzten beiden erzeugen wieder Schnittpunkte, von denen das Gleiche aus geschieht. Die Anzahl der Sekundärstrahlen im Strahlenbaum mit Tiefe d ist also:

$$s_d = (|L|+2) + 2(|L|+2) + 4(|L|+2) + \ldots = (2^d - 1)(|L|+2) \in O(2^d),$$

wobei |L| die Anzahl der Lichtquellen ist. Wenn für jeden Strahl n Objekte auf Schnitt getestet werden müssen und d als Obergrenze für die Tiefe des Strahlenbaums festgelegt wird, ergibt sich als Gesamtanzahl der Schnittoperationen:

$$O(xy \cdot (1+s_d)n) = O(n \cdot xy \cdot 2^d)$$

4.2 Verbesserungen der Laufzeit

Die im vorherigen Abschnitt bestimmte Anzahl der Schnittpunktbestimmungen ist linear in der Anzahl der Pixel, der Objekte und exponentiell in der Tiefe des Strahlenbaums. Nach [AK89] werden 95% der Laufzeit eines Raytracing-Rendervorgangs auf Schnittpunkttests verwendet; es besteht also – gerade in Bildern mit großem *d*, d. h. vielen Spiegelungen (vergl. Abbildung 9) – durchaus Optimierungsbedarf.

In [AK89] werden die Techniken zur Beschleunigung in folgende Kategorien eingeteilt:

- schnellere Schnittpunktbestimmung (schnellere bzw. weniger Strahl-Objekt-Schnittoperationen)
- weniger Strahlen, z. B. durch adaptive Tiefenkontrolle des Strahlenbaums
- verallgemeinerte Strahlen, d. h. verfolge ein Volumen (z. B. Kegel) statt eines Strahls

Die Geschwingkeit der Strahl-Objekt-Schnittoperationen kann man verbessern, indem man Algorithmen zur Schnittpunktbestimmung entwickelt, die mit möglichst wenig Elementaroperationen (Multiplikation, Addition etc.) auskommen, wie sie beispielsweise bei [Han89] und [Hai89] vorgestellt werden. Da diese Optimierungen häufig wenig anschaulich sind, sollen im Folgenden nur Methoden zur Reduktion der *Anzahl* der Strahl-Objekt-Schnittoperationen erläutert werden.

Wie oben beschrieben, muss prinzipiell jeder Strahl gegen jedes Objekt getestet werden. Dies ist jedoch, gerade bei zunehmend komplexer werdenden Szenen, ein Problem; insbesondere, weil nur der erste Schnittpunkt für den Strahl interessant ist. Eine Idee, die die Laufzeit eines Raytracing-Algorithmus' beschleunigen kann, ist es daher, effiziente Indexstrukturen einzuführen; zwei Möglichkeiten dafür werden im Folgenden erläutert.

Objekthierarchien Eine Möglichkeit, die Anzahl der Schnittpunkttests zu reduzieren, ist das Verschachteln von mehreren Objekten in ein gemeinsames Elternobjekt, das einfacher auf Schnitt mit einem Strahl zu prüfen ist. Es können beispielsweise mehrere Primitive in einer Kugel zusammengefasst werden, diese dann wieder in einem größeren Objekt, usw., bis zum Schluss alle Objekte in einem großen Container-Objekt zusammengefasst sind. Ein Beispiel für diese Situation ist in Abbildung 14 gezeigt.

Wird nun ein Strahl in den Raum geschossen, wird zunächst das Wurzelobjekt auf Schnitt geprüft. Gibt es einen Schnittpunkt, werden alle Kindobjekte geprüft, dann ggf. deren Kinder usw., bis es entweder kein Objekt mehr gibt, das den Strahl schneidet, oder so weit in der Objekthierarchie hinabgestiegen wurde, bis ein Strahl-Primitiv-Test durchgeführt werden muss.



Abbildung 14: *Objekthierarchie mit Kreisen als Container-Objekten (7 Kreisschnitt- und 1 Primitivschnitt- statt 11 Primitivschnittests)*

Bei der Auswahl der Art der Container-Objekte (z. B. Kugel, Quader, Zylinder) muss man zwischen Geschwingkeit des Schnittpunkttests und Genauigkeit der Unterobjektapproximation abwägen. Sicherlich ist es besser, wenn das Container-Objekt so genau wie möglich das enthaltene Objekt annähert, denn dann ist die Wahrscheinlichkeit geringer, dass man den Container trifft, obwohl das eigentliche Objekt nicht getroffen wird. Andererseits wird der Aufwand des Strahl-Container-Schnittpunkttests höher, je komplizierter der Container ist und unter Umständen gewinnt man dann keinen Geschwingkeitsvorteil mehr. Eine Erweiterung dieses Verfahrens ist beispielsweise die Verwendung mehrerer Container-Objekte unterschiedlichen Typs, die *alle* geschnitten werden müssen, bevor ein Test mit dem enthaltenen Primitiv durchgeführt wird.

Zur weiteren Optimierung kann man zu einem Strahl $v + \lambda r$ außer Startpunkt vund Richtung r auch noch ein Intervall $[a, b] \subset \mathbb{R}$ speichern, aus dem der Parameter λ sein darf. Davon kann man profitieren, wenn man eine Tiefensuche im Container-Baum durchführt und nach jedem erfolgreichen Strahl-Primitiv-Test das obere Ende des Intervalls so setzt, dass es gerade noch den berechneten Schnittpunkt enthält, aber keine Punkte, die weiter vom Startpunkt des Strahls entfernt sind. Dann braucht ein Container-Objekt, dessen nächster Punkt erst bei einem $\lambda > b$ geschnitten wird, nicht weiter untersucht zu werden, denn es kann sicher keine Objekte mehr enthalten, die näher an v sind als eines der bisher gefundenen.

Raumaufteilung Eine Alternative zur Gruppierung von Objekten ist die Partition des Raumes in Bereiche, die dann nur noch eine Teilmenge der Objekte enthalten.

Ein Beispiel dafür ist das *Binary Space Partitioning* (BSP)⁴, bei dem durch Ebenen eine Aufteilung des Raums in Halbräume vorgenommen wird, bis jeder Bereich nur noch eine festgelegte Anzahl an Primitiven enthält, vergl. Abbildung 15. Zu jedem Blatt des BSP-Baums wird dann eine Liste der Objekte, die im entsprechenden Bereich liegen, gespeichert.



Abbildung 15: BSP-aufgeteilter Bereich und entsprechender BSP-Baum

Wenn nun der erste Schnittpunkt eines Strahls mit einem Objekt aus der Szene bestimmt werden soll, wird, beginnend bei der Wurzel des BSP-Baums, immer dasjenige Kind des Knotens zuerst betrachtet, das den Bereich repräsentiert, der näher am Startpunkt des Strahls liegt. Mit dieser Methode erreicht man zuerst das Blatt des Baums, dessen assoziierter Bereich am nächsten am Startpunkt liegt. Gibt es einen Strahl-Primitiv-Schnittpunkt in diesem Bereich, kann abgebrochen werden, denn es gibt sicherlich kein Primitiv, das noch näher am Startpunkt des Strahls ist. Andernfalls muss man den zum jeweiligen Elternknoten gehörenden anderen Bereich prüfen und ggf. wieder eine Ebene nach oben gehen. Durch Anpassung der Grenzen des Intervalls, aus dem die Parameterwerte λ des Strahls kommen können, kann man bestimmte Unterbäume gleich vollständig von der Suche ausschließen.

Mit Hilfe der in diesem Abschnitt vorgestellten Techniken ist es möglich, die Anzahl der Schnittpunkttests pro Strahl in eine Größenordnung von $O(\log n)$ zu bringen.

⁴Sowohl BSP-Bäume als auch ähnliche Strukturen wie Octrees oder *k-d*-Bäume sind in jedem Standardwerk über Algorithmen ausführlich beschrieben. [dB93] beschäftigt sich zu großen Teilen mit speziellen Techniken zur Reduktion von Schnittpunkttests.

5 Ausblick

In der vorliegenden Arbeit wurden die Grundlagen des Renderings im Allgemeinen und des Raytracings im Speziellen erläutert. Es wurde ausgeführt, welche Arten des ausgesandten Lichts es gibt und wie man für einen gewissen Punkt in der Szene die beeinflussenden Lichtquellen bestimmt. Die durch die verschiedenen Techniken erzeugten Effekte wurden demonstriert, eine Laufzeitanalyse durchgeführt und Techniken der Geschwingkeitsoptimierung vorgestellt.

Ein Thema, das in dieser Arbeit nicht ausführlich behandelt wurde, sich aber für den interessierten Leser zur Vertiefung anbietet, ist das stochastische Raytracing. Damit lassen sich mit vergleichsweise einfachen Methoden Effekte wie weiche Schatten, Tiefenunschärfe, verschwommene Reflexionen, Kantenglättung und Bewegungsunschärfe simulieren.

Literatur

- [AK89] James Arvo and David Kirk. A survey of ray tracing acceleration techniques. In *An Introduction to Ray Tracing*. 1989.
- [Coo89] Robert L. Cook. Stochastic sampling and distributed ray tracing. In An Introduction to Ray Tracing. 1989.
- [dB93] Mark de Berg. *Ray Shooting, Depth Orders and Hidden Surface Removal.* Springer-Verlag, 1993.
- [DSC93] Heiko Duin, Günter Symanzik, and Ute Claussen. *Beleuchtungsalgorithmen in der Computergrafik.* Springer-Verlag, 1993.
- [Gla89] Andrew S. Glassner. Surface physics for ray tracing. In *An Introduction* to Ray Tracing. 1989.
- [Hai89] Eric Haines. Essential ray tracing algorithms. In *An Introduction to Ray Tracing*. 1989.
- [Han89] Pat Hanrahan. A survey of ray-surface intersection algorithms. In *An Introduction to Ray Tracing*. 1989.
- [Pov] POV-Ray Hall of Fame. http://hof.povray.org. Stand: 11.11.2007).
- [Wik] Raytracing. http://de.wikipedia.org/wiki/Raytracing. Stand: 11.11.2007.