Ausnutzung der Beleuchtungskohärenz mit Irradiance Gradients

Studienarbeit im Fach Informatik

vorgelegt von

Matthias Ott

geb. am 23. Juli 1972 in Ansbach

angefertigt am

Institut für Mathematische Maschinen und Datenverarbeitung
Lehrstuhl für Graphische Datenverarbeitung
Friedrich-Alexander-Universität Erlangen-Nürnberg

Betreuer: Philipp Slusallek

Beginn der Arbeit: 1. März 1997

Abgabe der Arbeit: 10. November 1997

Inhaltsverzeichnis

Abbildungsverzeichnis

Kapitel 1

Einführung

Die diffuse Beleuchtung trägt bei vielen Szenen stark zur Gesamtbeleuchtung bei. Es existieren aber noch keine Ansätze, diesen Beleuchtungsanteil effizient in bestehende Raytracingmethoden zu integrieren. Im Augenblick wird meist ein konstanter Term hierfür verwendet, was die tatsächliche Lichtverteilung aber nur sehr grob annähern kann.

1.1 Ein primitiver Ansatz

Eine einfache Möglichkeit, die Irradiance an jeder sichtbaren Stelle der Szene zu berechnen, würde sein, den indirekten Lichtanteil dort jedesmal neu zu erzeugen. Wenn man hier die Rekursionstiefe der Irradiance-Berechnung mit der Zahl R begrenzen würde und in jedem Rekursionsschritt S Strahlen in den Halbraum verschicken würde[1], müßte man insgesamt $(X \cdot Y) \cdot S^R$ Strahlen verschicken, wobei $X \cdot Y$ die Bildgröße darstellt. Dies bedeutet, daß man beispielsweise bei einer maximalen Rekursionstiefe von 5, einer Strahlanzahl von 512 und einer Bildgröße von $1024 \cdot 768$ Pixeln $2.8 \cdot 10^{19}$ Strahlen verschicken würde. Diese Zahl überfordert die heutigen Computersysteme aber noch bei weitem.

1.2 Überblick über Beleuchtungsberechnungsmethoden

1.2.1 Theorie des Raytracing

Zur realistischen Berechnung eines Bildes aus einer künstlich erzeugten Szene ist das exakte Beachten der Beleuchtung und des Lichtaustausches zwischen den einzelnen Objekten sehr wichtig. Raytracing ist ein weitverbreitetes Verfahren, um dies zu erreichen. Hier wird meist die Lichtverteilung vom Blickpunkt aus in die Szene betrachtet.
Wenn hiermit aber auch diffuse Beleuchtung berechnet werden soll, löst jeder einzelne vom Blickpunkt verschickte Strahl, wenn er auf eine reflektierende Fläche trifft, rekursiv eine große Anzahl weiterer Strahlen aus, was den Zeitverbrauch des Verfahrens stark erhöht. Dieses Problem soll nun durch den hier vorgestellten Ansatz, der ursprünglich von Gregory J. Ward

[1] Bei der Irradiance-Berechnung ist der gesamte sichtbare Halbraum relevant.

(siehe [10] und [9]) entwickelt und in seinem Raytracing-Paket Radiance [6] implementiert wurde, gelöst werden.

1.2.1.1 Strahlen im Raytracing

Die Zeit, die ein Raytracing-Algorithmus benötigt, um eine Szene zu berechnen, ist sehr stark von der Anzahl der verschossenen Strahlen abhängig. Ein allgemeiner Ansatz kann wie folgt aufgegliedert werden:

1. **Schnittpunkt mit der Szenengeometrie berechnen**
   Der Aufwand ist von der Komplexität der Szene abhängig, bei komplexeren Grundprimitiven ist der Zeitaufwand zur Schnittpunktberechnung größer, wobei aber die Anzahl der Strahlen nur linear mit der Pixelanzahl des erzeugten Bildes wächst.

2. **Direkte Beleuchtung durch Lichtquellen berechnen**
   Dies hängt von der Zahl der Lichtquellen ab, welche aber meist im Vergleich zu den in einer Szene enthaltenen Objekten nicht sehr hoch ist. Es müssen in jedem Schritt genauso viele Strahlen verschossen werden, wie Lichtquellen existieren.

3. **Spekularen Anteil berechnen**
   Hierbei kann der Aufwand schon größer werden, es existieren aber bereits effiziente Algorithmen von Kajiya [4] und Cook [3], um dies zu lösen.

4. **Indirekte Beleuchtung berechnen**
   Dies stellt hingegen ein Problem bei Raytracing-Algorithmen dar, denn es existieren meist sehr viele reflektierende Flächen in einer Szene, die zu der indirekten Beleuchtung beitragen. In jedem Schritt müssen hier Strahlen zu allen sichtbaren reflektierenden Flächen geschickt werden, wobei die Strahlen dieses Verfahren rekursiv erneut auslösen.

Bis jetzt war es meist üblich, einen konstanten indirekten Lichtanteil zu verwenden, um die Komplexität von Punkt 4 zu umgehen. Hierdurch kann aber das Berechnungsergebnis bei vielen reflektierenden Flächen in einer Szene nur sehr grob approximiert werden.
Wie man im direkten Vergleich von Bild B.1, das ohne diffuse Beleuchtung berechnet wurde, mit Bild B.2, in dem der durch Irradiance Gradients berechnete diffuse Anteil enthalten ist, erkennen kann, trägt die diffuse Beleuchtung hier maßgeblich zur Gesamtbeleuchtung bei.

1.3 Ziel der Studienarbeit

- Die primäre Aufgabe dieser Arbeit sollte es sein, das Irradiance Gradients-Verfahren, das von Gregory J. Ward [9] entwickelt wurde, als ein spezielles LightOp in das Framework der Lighting Networks in das Programmpaket Vision zu integrieren.

- Ferner sollte das Verfahren flexibel in ein Klassensystem zerlegt werden, um zu ermöglichen, daß man diesen Algorithmus später einfach erweitern und ändern kann.

- Außerdem sollte das Verfahren, bei möglicher Ausnutzung der bereits vorhandenen Kohärenzinformationen, erweitert werden.

1.4 Strukturierung der Arbeit

In Kapitel 2 werden die grundlegenden Ideen der Beleuchtungsberechnung kurz angerissen. Danach wird der Irradiance Gradients-Algorithmus auf theoretischer Basis vorgestellt.
In Kapitel 3 werden dann die programmiertechnischen Belange erörtert, die nötig waren, um den Algorithmus zu realisieren und in Vision zu integrieren.
In Kapitel 4 werden einige mögliche und auch realisierte Erweiterungen des Algorithmus vorgestellt.
In Kapitel 5 wird ein zusätzlich entwickeltes Tool beschrieben, mit dem sich die wichtigsten Parameter graphisch visualisieren lassen.
In Kapitel 6 erfolgt schließlich eine Bewertung des gesamten Ansatzes.
In Anhang A werden einige interne Parameter näher erklärt, die benötigt werden, wenn der implementierte Algorithmus mit Vision zusammen verwendet werden soll.
Im Anhang B sind abschließend Bildbeispiele zu sehen, die mit Vision und Irradiance Gradients berechnet wurden.

Kapitel 2

Theorie

Hier werden erst die grundlegenden Ideen der Beleuchtungsberechnung kurz angerissen. Danach wird der Irradiance Gradients-Algorithmus auf theoretischer Basis vorgestellt.

2.1 Beleuchtung

2.1.1 Grunddefinitionen

> Die direkte Beleuchtungsdichte (auch: Radiance) ist der Lichtfluß, der in eine bestimmte Richtung abgestrahlt wird pro senkrecht zu dieser Richtung projizierter Einheitsfläche.
>
> Die indirekte Beleuchtungsdichte (auch: Irradiance) ist der Lichtfluß pro Einheitsfläche, der von einer nicht selbstleuchtenden Fläche auf einer Oberfläche eintrifft.

Die Beleuchtungsdichte E an einem Punkt X wird als das Integral der Strahldichte über die projizierte Halbkugel wie folgt berechnet:

$$E = \int\limits_{\Omega_+} L(\omega) \cos \omega d\omega =$$

$$= \int\limits_0^{2\pi} \int\limits_0^{\frac{\pi}{2}} L(\Theta, \Phi) \cos \Theta \sin \Theta d\Theta d\Phi,$$

wobei Θ der polar Winkel, Φ der azimuthal Winkel und $L(\Theta, \Phi)$ die Luminance aus der Richtung (Θ, Φ) ist.

2.2 Grundlagen von Irradiance Gradients

Die Idee des hier implementieren Algorithmus ist es, daß die Anzahl der Strahlen drastisch reduziert werden kann, ohne ganz auf die Berechnung der Irradiance zu verzichten.
Folgende Beobachtungen waren für die Entwicklung des Algorithmus wichtig [10]:

- Da reflektierende Flächen in einer normalen Szene weit verbreitet sind, benötigt man viele Strahlen, um die indirekte Beleuchtung zu berechnen.
- Ein berechneter indirekter Beleuchtungswert ist unabhängig vom Beobachtungpunkt nach der Lambertschen Annahme [5].
- Die indirekte Beleuchtung ändert sich auf einer Fläche meist nur langsam, da der direkte Lichtanteil, der stärker variiert, und die dazugehörigen Schatten schon beim Punkt 2 des Raytracing-Ansatzes berechnet wurden.

Hieraus folgt die Idee, daß viele einmal berechneten Irradiance-Werte durch Caching dieser und geschickter Kohärenzausnutzung in der näheren Nachbarschaft wiederverwertet werden können.
Da es aber doch zu Änderungen in der Irradiance in einer Szene kommt, werden zusätzlich zu der Irradiance noch Gradienten, die die 1. Ableitung der Irradiance darstellen, berechnet. Diese Gradienten (siehe Sektion 2.6) können dazu benutzt werden, aus mehreren Irradiance-Werten einen neuen zu interpolieren.
Ziel es es, Diese Irradiance-Werte nur auf den tatsächlich an der Lichtverteilung beteiligten Flächen zu berechnet.

2.3 Grobaufteilung

Der gesamte Algorithmus gliedert sich in zwei Teile, einer primären und einer sekundären Methode:

```
Wenn
     gültige Wert(e) in der Nähe der Berechnungspunkte existieren
dann
     wende sekundäre Methode an
sonst
     wende primäre Methode an
```

Grob gesagt werden bei der primären Methode neue Werte durch Strahlverschicken erzeugt, bei der sekundären Methode neue Werte durch Interpolation aus bereits vorhandenen gewonnen.

2.4 Primäre Methode

2.4.1 Abtasten der Umgebung

Um einen Irradiance-Wert an einer Stelle zu berechnen, muß die gesamte Umgebung abgetastet werden. Um dies möglichst schnell zu bewältigen, wird die Umgebung durch eine Monte Carlo Methode abgetastet.

2.4.2 Abtastung über eine Halbkugel

Um das Integral über die Umgebung diskret zu berechnet, wird eine Parametrisierung benötigt, mit der der Halbraum gut abgedeckt werden kann. Hier bietet sich die Parametrisierung über eine Halbkugel an, auf der die Gewichtung der einzelnen Strahlen durch eine Cosinusgewichtung schon enthalten ist. Es darf aber keine direkte Emission von Lichtquellen berücksichtigt werden, da der direkte Anteil der Beleuchtung nicht innerhalb dieses Algorithmus berechnet wird. Um dies zu erreichen, ist es notwendig, den Lichtanteil der Emission, den ein Strahl zurückliefert, nachdem er eine Lichtquelle getroffen hat, zu vernachlässigen.

$$E = \frac{\pi}{M \cdot N} \sum_{j=1}^{M} \sum_{k=1}^{N} L(\Theta_j, \Phi_k)$$

$$\Theta_j = \sin^{-1}(\sqrt{\tfrac{j-X_j}{M}})$$
$$\Phi_k = \pi \tfrac{(k-Y_k)}{M}$$

X_j, Y_k sind gleichförmig verteilte Zufallszahlen zwischen 0 und 1 und $M \cdot N$ ist die gesamte Anzahl der Strahlen, wobei $N \approx \pi M$. Als Beispiel für solch eine Halbkugelabtastung siehe Bild B.12.

2.4.3 Verringerung der Strahlanzahl

Die initiale Strahlanzahl S, die in der ersten Rekursionstiefe verschossen wird, wird bei jeder weiteren Rekursionstiefe R mit dem Faktor X^R gewichtet, wobei die initiale Rekursionstiefe den Wert 0 hat.
Der Wert X sollte dem mittleren Reflexionskoeffizienten der Szene möglichst gut entsprechen. Wenn dieser Wert nicht bekannt ist, kann hier meist eines Wert von $0,5$ verwendet werden. Die Verringerung kann deswegen geschehen, da die Strahlen einer tieferen Rekursionstiefe immer weniger Einfluß auf die berechnete Irradiance haben, und man deswegen dort, um einen konstanten Fehler nicht zu unterschreiten, weniger Strahlen zu verschießen braucht. Dies hat den Vorteil, daß die Berechnung schneller vonstatten geht. Außerdem kommt es dadurch zwangsläufig zu einem Rekursionsabbruch, da sich die Strahlanzahl mit wachsender Rekursionstiefe der Zahl 0 annähert[1].
Eine mögliche Zahlenfolge wäre zum Beispiel:

$$32 \Rightarrow 16 \Rightarrow 8 \Rightarrow 4 \Rightarrow 0$$

Als Folgerung aus dieser Verringerung muß später aber darauf geachtet werden, daß keine Irradiance-Werte einer höheren Rekursionsstufe aus Werten einer tieferen Rekursionsstufe interpoliert werden, da sonst ein zu großer Fehler möglich wäre.

[1] Es muß eine Aufteilung von ca. $1 : \pi$ geben, was bei Zahlen < 4 nicht mehr möglich ist. Diese Aufteilung ist durch die Verteilung der Strahlen auf die Halbkugel begründet.

2.4.4 Begrenzung der Rekursionstiefe

Da das Gewicht von Strahlen aus tieferen Rekursionstiefen immer geringer wird, kann der Verfahren mit einer Schranke für die maximale Rekursionstiefe versehen werden. Wenn diese Schranke erreicht wird, wird an dieser Stelle keine neue Rekursion mehr angestoßen, sondern es wird die Standardhintergrundfarbe (siehe Sektion 4.2) an die höhere Rekursionsebene zurückgeliefert. Diese Schranke muß während einer Berechnung nicht unbedingt erreicht werden, eine Rekursion kann auch durch die Verringerung der Strahlanzahl (siehe Sektion 2.4.3) beendet werden.

2.4.5 Ablegen der Werte

Die bei der primären Methode erzeugten Werte werden in einen Speicher abgelegt, damit sie dann beim Anwenden der sekundären Methode wiederverwendet werden können.

2.5 Sekundäre Methode

Die sekundäre Methode stellt die Interpolation der bereits berechneten Irradiance-Werte dar, also den eigentlich neuen Teil von Irradiance Gradients.

2.5.1 Auswahl der Objekte

Um die passenden Werte interpolieren zu können, muß erst einmal festgestellt werden, welche der im Speicher abgelegten Werte überhaupt hierfür in Frage kommen.
Hierzu wird das in Abbildung 2.1 dargestellte Split Sphere Modell verwendet.

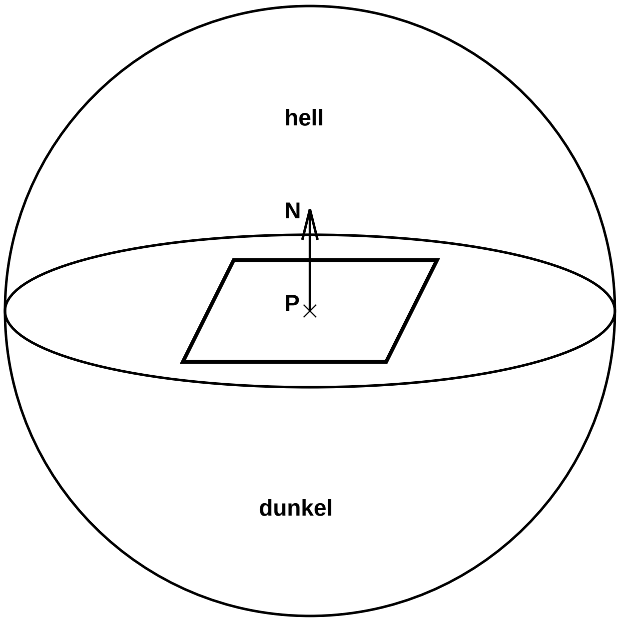

Abbildung 2.1: Das Modell einer Split Sphere

Dabei wird ein Irradiance-Wert nur dann zur Interpolation verwendet, wenn er folgende Tests besteht:

- Es werden keine Irradiance-Werte akzeptiert, die in einer tieferen als der aktuellen Rekursionsebene berechnet wurden, da der maximale Fehler, die diese beinhalten können, zu groß ist.

- Wenn der gespeicherte Wert zu weit von der aktuellen Position entfernt liegt, wird dieser Wert verworfen. Die maximal erlaubte Entfernung vom Ausgangspunkt eines Gradienten Objektes wird wie folgt berechnet:

$$\texttt{maximale Entfernung} = \frac{\sum_{i=1}^{N} \texttt{Entfernung}(i)}{N},$$

  wobei N die Anzahl der über den Halbraum verschossen Strahlen ist und $\texttt{Entfernung}(i)$ die Entfernung der i-ten getroffenen Fläche zurückgibt.

- Wenn sich die Normale des gespeicherten Wertes zu sehr von der aktuellen Oberflächennormale unterscheidet, wird dieser Wert zurückgewiesen. Hierbei wird auch noch einmal die Entfernung mit einbezogen, d.h. die Normale darf stärker abweichen, wenn die Entfernung des gespeicherten Wertes gering ist. Ferner findet auch hier, wie schon in Sektion 2.4.3 beschrieben, eine Abschwächung über die Rekursion statt, wodurch die Normale in einer tieferen Rekursionsebene stärker abweichen darf, bevor ein Wert abgewiesen wird.

- Es dürfen keine Werte zugelassen werden, deren Position P_0 sich räumlich vor der aktuellen Position P befindet, wie es in Abbildung 2.2 der Fall ist. Es könnte sonst die Situation eintreten, daß von P_0 aus fast keine anderen Flächen mehr sichtbar sind, der Gradient also dort klein ist. P wird hier aber von P_0 abgeschattet, der Gradient bei P würde also bei einer direkten Neuberechnung dort groß sein. Die Interpolation mit dem kleinen Gradienten von P_0 ergibt aber wiederum einen kleinen Gradienten, was bedeutet, daß eine Interpolation an dieser Stelle nicht wünschenswert ist.

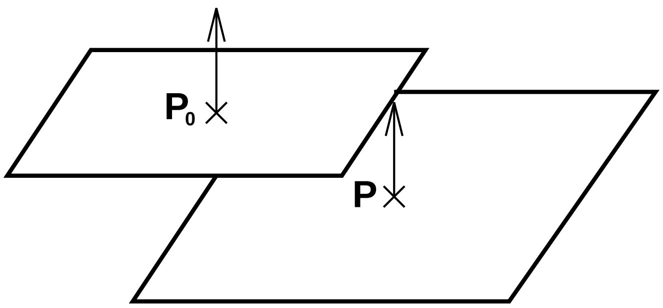

Abbildung 2.2: Abschattung von P durch P_0, der Wert von P_0 darf deswegen bei P nicht verwendet werden.

- Wie später in Kapitel 3.3.4 noch näher erläutert wird, kann die Regelmäßigkeit des Verfahrens zu sichtbaren Artefakten führen. Deshalb wird die endgültige Entscheidung durch ein Verfahren, das innerhalb von 90 % bis 110 % zufällig Werte zuläßt oder ablehnt, reguliert:

  $$\begin{array}{l}\texttt{wenn}\\ \quad P = \frac{\texttt{Entfernung des Punktes}}{\texttt{Maximal erlaubte Entfernung}}\texttt{,}\\ \quad N = \frac{\texttt{Abweichung der Normalen}}{\texttt{Maximal erlaubte Abweichung}}\texttt{,}\\ \quad \frac{P+N}{2} > (0.9 + 0.2 \cdot \texttt{random()})\texttt{,}\\ \texttt{dann}\\ \quad \texttt{lehne Wert ab}\end{array}$$

  `random()` liefert eine Zufallszahl zwischen 0 und 1.

2.5.2 Interpolation

Zur Interpolation der in Frage kommenden Irradiance-Werte werden diese mit einer gewichteten Summe aufaddiert, wobei die Irradiance-Werte noch zusätzlich durch die Gradienten (siehe Sektion 2.6) reguliert werden. Dadurch wird die Ordnung der Interpolation erhöht.
Die Interpolation erfolgt nach folgender Formel:

$$E(\vec{P}) = \frac{\sum\limits_S w_i(\vec{P}) \left[E_i + (\hat{n}_i \times \hat{n}) \cdot \vec{\nabla}_r E_i + (\vec{P} - \vec{P}_i) \cdot \vec{\nabla}_t E_i\right]}{\sum\limits_S w_i(\vec{P})}, \text{mit}$$

$$w_i(\vec{P}) = \frac{1}{\frac{\|\vec{P}-\vec{P}_i\|}{R_i} + \sqrt{1 - \hat{n} \cdot \hat{n}_i}} = \text{Gewicht des Strahls } i,$$

wobei $\vec{P}$ die Position des Testpunktes,
$\hat{n}$ die Oberflächennormale des Testpunktes,
E_i der Irradiance-Wert des Strahls i,
$\vec{P}_i$ die Position der vom Strahl i getroffenen Fläche,
$\hat{n}_i$ die Oberflächennormale der vom Strahl i getroffenen Fläche,
R_i das harmonische Mittel der Entfernung zu Objekten, die von $\vec{P}_i$ aus sichtbar sind,
S die Menge der gültigen Irradiance-Werte, $\{i : w_i(\vec{P}) > 1/a\}$ und
a die von Benutzer gewählte Fehlerschranke ist.

2.6 Gradienten

Um die Interpolation der Irradiance-Werte zu verbessern, werden, wie schon oben erwähnt, einige Zusatzinformationen benötigt. Zu diesem Zweck wird an den betreffenden Stellen der Gradient der Irradiance berechnet, der die Richtung und Größe der Änderung angibt, was für die Interpolation eine wertvolle Zusatzinformation darstellt. Die Informationen, die für die Berechnung der Gradienten nötig sind, werden schon beim Verschicken der Strahlen über die Halbkugel erhalten, was bedeutet, daß hierfür kein Zusatzaufwand notwendig ist. Dadurch kann das Berechnen der Gradienten ohne merkliche Verlangsamung des Raytracings geschehen.

2.7 Berechnung der Gradienten

Bei der Abtastung der Halbkugel ist neben der Helligkeit der getroffenen Flächen auch deren Entfernung bekannt. Aus diesen zwei Informationen für jeden Strahl kann man den Gradienten berechnen.
Da die Irradiance eine Funktion mit fünf Variablen darstellt (drei aus der Position und zwei aus der Richtung), sollte auch der Irradiance-Gradient ein fünfdimensionaler Vektor sein. Es sind aber nur je zwei Dimensionen frei, da sich der Gradient in der Tangentialebene zum Berechnungspunkt befindet. Wegen der besseren Verwendbarkeit werden hier aber zwei dreidimensionale Vektoren berechnet, wobei der eine sich auf die Rotation und der andere auf die Translation bezieht.

2.7.1 Rotationsgradient

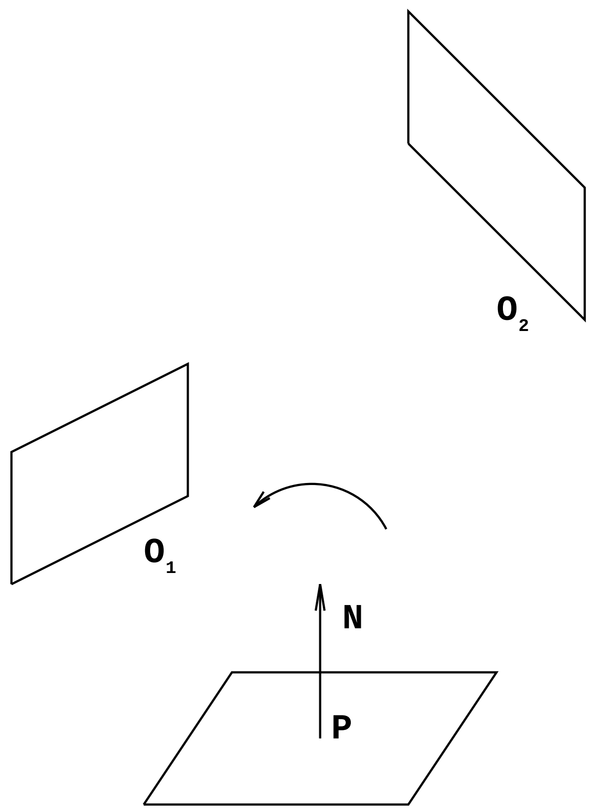

Abbildung 2.3: Der Einfluß der Fläche O_1 auf P wird größer, wenn N nach links gedreht wird.

Wie in Abbildung 2.3 dargestellt ist, hat immer die den Ausgangspunkt nähere Fläche die größere Auswirkung in Bezug auf Veränderung der Normalenrichtung. Dies bedeutet bei Abbildung 2.3, daß der Einfluß der Fläche O_1 wächst, wenn die Normale nach links gedreht wird.
Um den Rotationsgradienten über die Halbkugel zu bestimmen, werden die Ableitungen der Cosinus bei jeder beteiligten Teilfläche der Halbkugel aufsummiert[2]:

$$\vec{\nabla}_r E = \frac{\pi}{M \cdot N} \sum_{k=0}^{N-1} \left\{ \hat{v}_k \sum_{j=0}^{M-1} - \tan \Theta_j \cdot L_{j,k} \right\},$$

wobei $\hat{v}_k$ der Basisebenen Einheitsvektor in der $\phi_k + \frac{\pi}{2}$ Richtung und $M \cdot N$ die Gesamtanzahl der Strahlen ist.

Anmerkung: Der Tangens erscheint hier, da cos abgeleitet $-\sin$ ergibt, und das Strahlenverteilungsschemata über der Halbkugel schon implizit einen Cosinus als Gewichtsfaktor enthält. Hieraus ergibt sich für die Sinusgewichtung die Formel $\frac{-\sin}{\cos} = -\tan$.

2.7.2 Translationsgradient

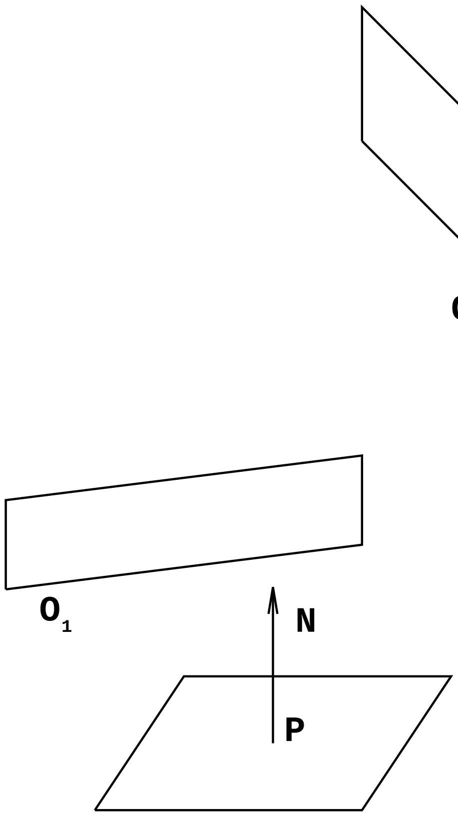

Abbildung 2.4: O_1 verdeckt O_2, O_2 wird aber sichtbar, wenn P nach rechts bewegt wird.

[2]Die Parametrisierung der Halbkugel geschieht über Θ und Φ

Die Berechnung des Translationsgradienten erweist sich auf der Halbkugel als etwas komplizierter, da hier, wie in Abbildung 2.4 sichtbar, die Verdeckungsmöglichkeit beachtet werden muß.

Hier wird betrachtet, wie der projizierte Raumwinkel jeder Zelle der Halbkugel von der Verschiebung des Mittelpunktes der Halbkugel betroffen wird (siehe Abbildung 2.5). Die Änderung des projizierten Raumwinkels, multipliziert mit der Radiance der Zelle, ergibt die Veränderung des Beitrages dieser Zelle, welche teilweise durch die Entfernung der beteiligten Fläche bestimmt wird. Umso näher die Fläche, umso höher die Rate der Änderung in Beziehung zu einer Bewegung senkrecht zu der Grenze zwischen zwei benachbarten Zellen. Tatsächlich ist es immer die Entfernung zu der näheren Zelle, die die Größe der Veränderung der Verdeckung bestimmt. Jede Zelle hat einen gleich großen projizierten Raumwinkel (z. B. $\frac{\pi}{MN}$ Steradian), deshalb sollte jede Zelle die gleiche Fläche bei der Projizierung auf eine Ebene belegen.

Die endgültige Summation wurde so angeordnet, daß sie die Differenzen in Radiance zwischen zwei benachbarten Zellen verwendet, und dadurch über die Grenzen, und nicht über die Zellen selbst summiert. Hieraus ergibt sich für den Translationsgradienten $\vec{\nabla}_p E$ diese Formel:

$$\vec{\nabla}_p E = \sum_{k=0}^{N-1} \left\{ \hat{u}_k \frac{2\pi}{N} \sum_{j=1}^{M-1} \frac{\sin\Theta_{j_-} \cdot \cos^2\Theta_{j_-}}{\min(r_{j,k}, r_{j-1,k})} \cdot (L_{j,k} - L_{j-1,k}) + \right.$$
$$\left. + \hat{v}_{k_-} \sum_{j=0}^{M-1} \frac{\sin\Theta_{j_+} - \sin\Theta_{j_-}}{\min(r_{j,k}, r_{j,k-1})} \cdot (L_{j,k} - L_{j,k-1}) \right\},$$

wobei $\hat{u}_k$ der Einheitsvektor in der Θ_k Richtung,
$\hat{v}_{k_-}$ der Einheitsvektor in der $\Theta_{k_-} + \frac{\pi}{2}$ Richtung,
Θ_{j_-} der Polarwinkel der vorherigen Grenze, $\sin_{-1}\sqrt{\frac{j}{M}}$,
Θ_{j_+} der Polarwinkel der nächsten Grenze, $\sin_{-1}\sqrt{\frac{j+1}{M}}$ und
Θ_{k_-} der Azimuthalwinkel der vorherigen Grenze, $2\pi\frac{k}{N}$ ist.

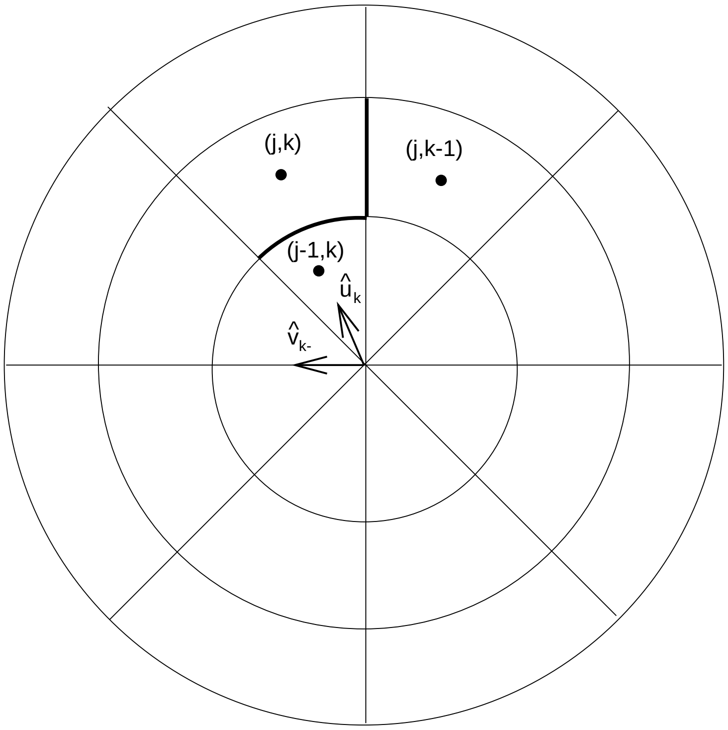

Abbildung 2.5: Die Projektion der Halbkugel in die Tangentialebene

Kapitel 3

Implementation

Hier werden die programmiertechnischen Belange erörtert, die nötig waren, um den Algorithmus zu realisieren und in Vision zu integrieren.

3.1 Das Programmpaket Vision

Das Programmpaket Vision [8] kann in drei große Bereiche eingeteilt werden. Hier wären Rendering, Global Illumination und Scene Description, wobei sich der Bereich Global Illumination noch einmal in Lighting, Shader, LightSource Shader und Volume Lighting unterteilt. Ein Lighting-Objekt wiederum kann aus mehreren LightOps bestehen (siehe Sektion 3.1.1), wobei der Algorithmus Irradiance Gradients als ein solches in Vision integriert wurde.

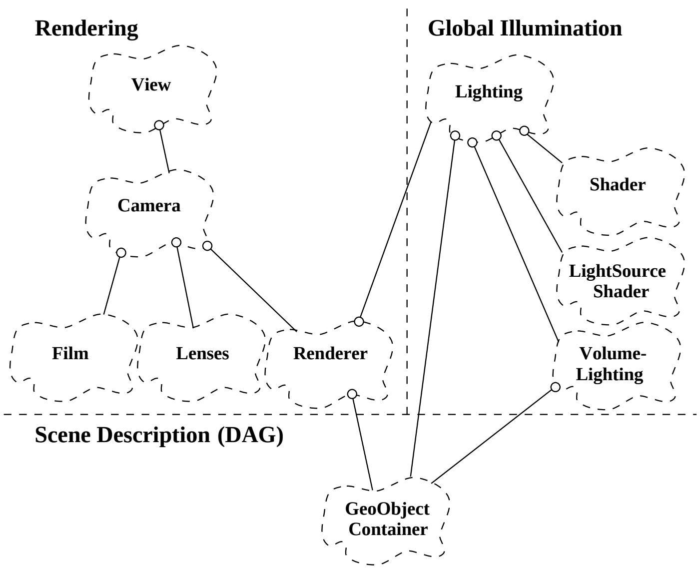

Abbildung 3.1: Die Struktur von Vision

3.1.1 Begriffserklärungen

Lighting Network: Ein Lighting Network ist ein Netz von verschiedenen Algorithmen, die alle verschiedene Aspekte der Beleuchtung berechnen. Durch dieses Netz verbundene Algorithmen berechnen die Lichtverteilung in einer Szene.

LightOp: Ein LightOp ist ein einzelner Teil eines Lighting Networks, der einen speziellen Aspekt der Beleuchtung berechnet.

Multi LightOp: Ein Multi-LightOp ist der oberste LightOp, der für die Verwaltung und Koordinierung der LightOps zuständig ist. Er teilt die Berechnung der Lichtverteilung auf die anderen LightOps auf und erzeugt aus dessen Ergebnissen das endgültige Ergebnis.

3.1.2 Integration in Vision

Nachdem Vision in der Programmiersprache C++ geschrieben ist, konnte der Algorithmus gut gekapselt als eigenes LightOp integriert werden. Dadurch war es nun möglich, in diesem LightOp nur die Funktionalität zur Berechnung der diffusen Beleuchtung zu implementieren und die Irradiance dann als Teil eines komplexeren Lighting Networks, wie z.B in Abbildung 3.2 gezeigt, zurückzuliefern.
Ferner wurde aber auch die Möglichkeit erhalten, sich zu Visualisierungszwecken nur den diffusen Anteil ausgeben zu lassen.

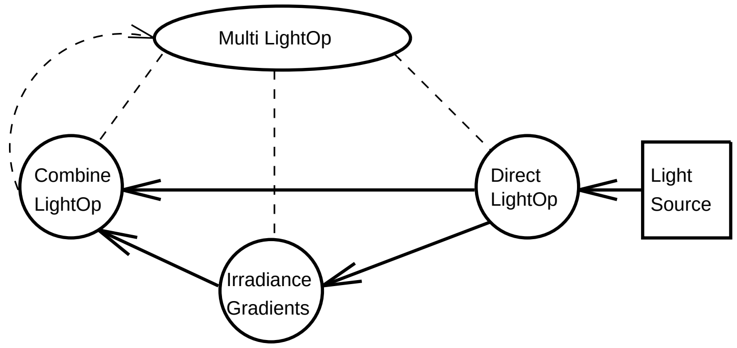

Abbildung 3.2: Ein mögliches Lighting Network

3.1.3 Konfigurierbarkeit zur Laufzeit

Vision bietet durch seine graphische Oberfläche die Möglichkeit, einzelne LightOps zur Laufzeit zu konfigurieren. Dadurch ist es möglich, verschiedene Parameter der Teilalgorithmen zu verändern. Für den hier implementierten Algorithmus kann man hiermit die in Abbildung 3.3 gezeigten Werte einstellen, wobei die einzelnen Felder noch genauer im Anhang A erklärt werden.

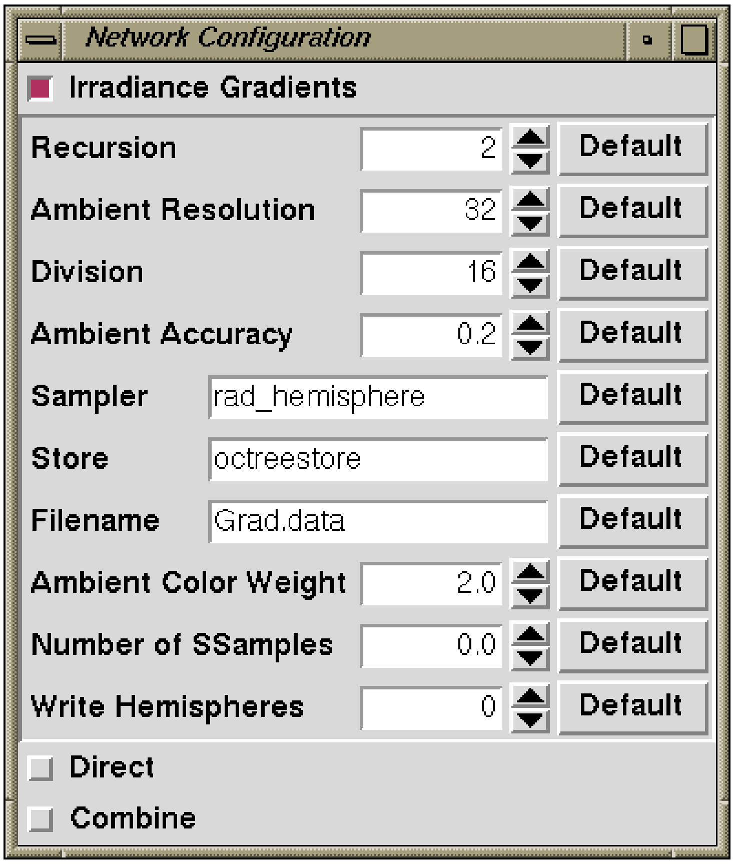

Abbildung 3.3: Die graphischen Einstellmöglichkeiten des Benutzers in Vision

3.2 Überblick über die Implementation

Zur Implementierung der speziellen Eigenschaften des Irradiance Gradients-Algorithmus wurde zuerst einmal die Klasse **LightOp** abgeleitet und dann dort Basismethoden des Verfahrens implementiert, so hauptsächlich die Entscheidung zwischen **primärer** und **sekundärer Methode** (siehe Sektion 2.3). Zur Implementierung des Speichers wurde ein neue Oberklasse erzeugt, von der dann jeweils der spezielle Speicheralgorithmus (siehe Sektion 2.4.5, auch Sektion 3.3.2) als eigene Unterklasse abgeleitet wurde. Auch für den Algorithmus zur Abtastung des Halbraumes (siehe Sektion 2.4.1) wurde eine Oberklasse erzeugt, deren abgeleitete Unterklassen dann nur die spezielle Funktionalität für das Verfahren über eine Halbkugel (siehe Sektion 2.4.2) oder über eine Ebene (siehe Sektion 4.6.1) enthalten.

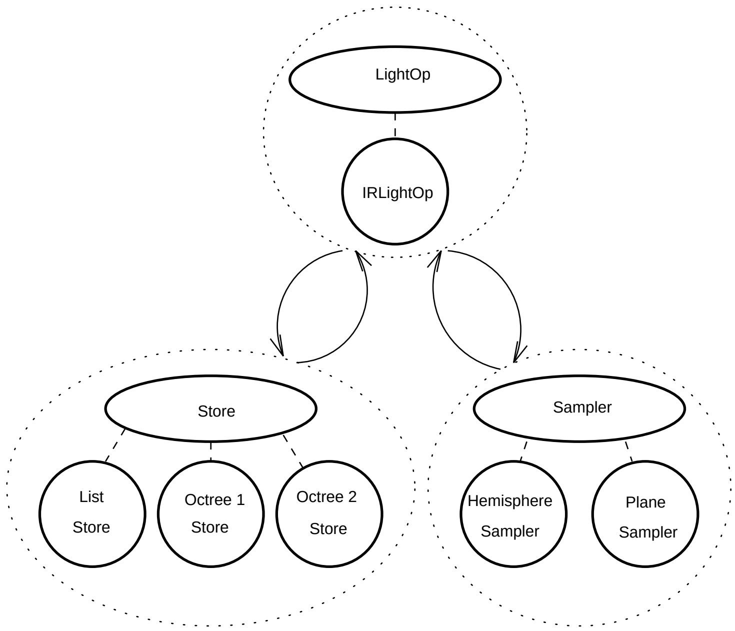

Abbildung 3.4: Die Klassenstruktur des LightOps

3.3 Einzelheiten einzelner Implementationsbereiche

3.3.1 Entscheidung über primäre oder sekundäre Methode

Zur Klärung der Frage, ob an einer Position P die primäre oder sekundäre Methode angewandt werden muß, wird an das Speicher Objekt eine Anfrage mit der Suchposition P gestellt. Das Speicher Objekt liefert daraufhin alle in Frage kommenden Irradiance-Objekte zurück. Über diese Objekte werden dann die Tests aus Sektion 2.5.1 ausgeführt. Wenn hier am Ende noch Objekte übrig bleiben, wird auf diesen die sekundäre Methode angewandt, wenn es keine passenden gespeicherten Objekte gibt, wird ein neues Irradiance-Objekt durch die primäre Methode erzeugt.

3.3.2 Speichern der Gradienten

Zur späteren Wiederverwendung müssen die Gradienten in irgendeine Art Speicher abgelegt werden, in dem sie dann leicht wieder gefunden werden können. Dieser Speicheralgorithmus ist ein wichtiger Teil der Gesamteffizienz, da das Finden im Speicher wesentlich schneller als das Neuerzeugen eines Irradiance-Objektes sein sollte, um diesen Aufwand zu rechtfertigen. Einführend wird hier zuerst in Tabelle 3.5 ein Zeitvergleich der drei implementierten Speicherarten aufgezeigt, wobei die erste Spalte den Typ, die zweite die Anzahl der berechneten Gradienten und die dritte die Zeit, die der Algorithmus benötigte, angibt.

Speicher	*Gradienten*	*Zeit (in Sekunden)*
Listen-Speicher	820	25
	3000	976
Octree-Speicher (nach Ward '88)	820	22
	3000	445
Alternativer Octree-Speicher	820	21
	3000	372

Abbildung 3.5: Vergleichszeiten der drei Speichertypen

3.3.2.1 Listen-Speicher

Diese Methode ist die einfachste der drei Ansätze. Die Idee ist es, die Gradienten in einer linearen doppelt verketteten Liste abzulegen. Dies führt dann zu einer Performance von $O(N)$. Die Zeit, die für die Ablage der Gradienten benötigt wird, ist zwar konstant, bei der Suche muß dann aber die gesamte Liste überprüft werden. Dieser Ansatz zeigt deshalb nur bei einer kleinen Anzahl von Gradienten eine akzeptable Leistung.

Eine Verbesserung der Suchgeschwindigkeit im Vergleich zu einer linearen Liste stellt nun die Verwendung eines Octrees zur Ablage der Gradienten dar. Hierbei wird der dreidimensionale Raum der Szene rekursiv jeweils in acht Bereiche unterteilt.

3.3.2.2 Octree-Speicher (nach Ward '88)

Der Grundgedanke hier ist, daß kleine Objekte, d.h. mit kleinem Gültigkeitsradius, tiefer in den Octree abgelegt werden und dadurch nur dann gefunden werden, wenn sich die Suchposition in geringer Nähe befindet. Dadurch wird die Anzahl der zu untersuchenden Objekte stark eingeschränkt. Dies wird dadurch realisiert, daß ein Irradiance-Objekt in den Octree-Knoten abgelegt wird, in dem sich seine räumliche Position befindet und dessen Seitenlänge größer als zweimal, aber nicht größer als viermal im Verhältnis zum Gültigkeitsradius des Irradiance-Objektes ist (siehe Abbildung 3.6). Ein so erzeugter Octree enthält nun Knoten, die sowohl acht Kinderknoten als auch mehrere Irradiance-Objekte selbst enthalten können. Die Suche in solch einem Octree erfolgt nun nach dem folgenden Ansatz:

```
Über jedes Kind
    Wenn
         Suchposition ist innerhalb der halben Würfelgröße des Kindes
              Durchsuche diesen Kinderknoten
```

Das heißt, daß der Algorithmus nicht nur die Knoten, die die Suchposition P enthalten, durchläuft, sondern auch Knoten, deren räumliche Grenzen weniger als die halbe Kantenlänge der aktuellen Würfelgröße entfernt liegen. Dadurch werden alle in Frage kommenden Knoten erfaßt.
Es ergibt sich, daß sich der Algorithmus im schlechtesten Fall wie $O(N)$ verhält, bei einer uniformen Verteilung der Irradiance-Objekte beträgt die Performance aber $O(\log(N))$ [9].

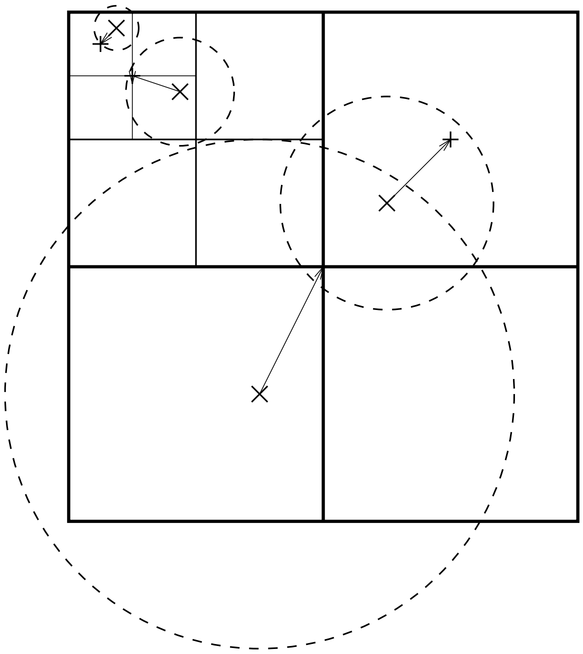

Abbildung 3.6: Das Ablageschemata der Objekte, Objekte mit größerem Gültigkeitsradius werden weiter oben abgelegt.

Ein Problem hierbei ist aber, daß der Baum sehr seitenlastig werden kann, wenn keine uniforme Verteilung der Werte gegeben ist, was z.B. der Fall ist, wenn der berechnete Ausschnitt nur einen kleinen Teil der gesamten Szene darstellt. Der oberste Octree-Würfel muß immer die gesamte Szene enthalten, da aber die Unterteilungen einen Würfel immer in acht gleich große Stücke zerlegen, werden so die meisten Objekte in einen relativ kleinen Bereich des Octree-Speichers abgelegt. Dadurch geht hier viel von der Sucheffizienz verloren.

3.3.2.3 Alternativer Octree-Speicher

Dieser Ansatz verfolgt eine etwas andere Idee der Ablage und Suche der Objekte in einem Octree. Die Irradiance-Werte werden in den Octree beim Ablegen immer nur so weit nach unten propagiert, so lange der Gültigkeitsradius noch vollständig in den Würfel des betreffenden Knotens paßt. Dies hat den Vorteil, daß die Suche nach passenden Werten einfacher erfolgen kann. Es muß immer nur ein einziger linearer Weg durch den Octree-Baum verfolgt werden, um alle betreffenden Objekte zu finden, d. h. es kann nicht passieren, daß mehrere Kinder eines Knotens durchsucht werden müssen, wie es bei dem Algorithmus von

Ward der Fall ist.
Ferner wird hier ein Objekt nur dann tiefer in den Octree geschoben, wenn der Vaterknoten bereits eine vorher festgelegte Maximalanzahl von Objekten enthält. Wenn diese Grenze überschritten wird, werden acht Unterwürfel erzeugt, wobei der Mittelpunkt der Unterteilung aus den räumlichen Mittelpunkten der abzulegenden Objekte des Oberwürfels gebildet wird. Dadurch wird erreicht, daß der Baum nicht zu sehr verzweigt wird und die räumliche Aufteilung sich der Szenengeometrie anpassen kann.

3.3.2.4 Vorsortierung der Objekte

Um die Geschwindigkeit des Verfahrens zu verbessern, werden die Irradiance-Objekte schon beim Ablegen in den Speicher nach aufsteigender Rekursionstiefe sortiert. Da Irradiance-Objekte aus tieferen Rekursionstiefen nicht für die Interpolation in geringeren Tiefen verwendet werden (siehe Sektion 2.4.3), kann dadurch das Durchsuchen der Objekte, die in einem Knoten direkt enthalten sind, erheblich verkürzt werden.

3.3.2.5 Bewertung

Der Listen Speicher sollte nur zu Testzwecken verwendet werden, er wurde hier auch nur als Validierungsmittel für die zwei Octree-Speicher implementiert.
Bei den zwei Octree-Speicher hat sich kein eindeutiger Favorit herausgebildet, hier ist es sehr stark szenenabhängig, welcher der beiden schneller ist. Einige Vergleichszeiten wurden in Tabelle 3.5 festgehalten.

3.3.3 Speichern der Objekte in einer Datei

Die erzeugten Irradiance-Objekte gehen normalerweise verloren, wenn eine Szene vollständig berechnet wurde. Da die Werte aber positionsunabhängig sind, könnte man diese zum Beispiel für die erneute Berechnung der Szene aus einem anderen Blickwinkel wiederverwenden. Außerdem lassen sich bessere Ergebnisse erzielen, wenn für eine Szene einige Irradiance-Objekte in einer niedrigen Auflösung erzeugt werden und die Szene dann mit diesen Werten erneut in einer hohen Auflösung endgültig berechnet wird.
Um dies zu erreichen, müssen die Objekte über verschiedene unabhängige Berechnungen hinweg gespeichert werden. Dies wird hier durch das Ablegen der Irradiance-Objekt in einer Datei realisiert. Da aber verschiedene Parameterwerte bei dem Algorithmus verschieden gute Ergebnisse erzeugen, muß gewährleistet sein, daß Irradiance-Objekte nur dann wiederverwertet werden, wenn sie mit den gleichen Einstellungen wie die aktuelle Berechnung erzeugt worden sind.
Hierfür werden im Kopf der Irradiance-Datei die verwendeten Parameter vermerkt. Dadurch kann man die Übereinstimmung der Parameter beim Einlesen der Datei überprüfen und das Wiederverwenden der Irradiance-Werte gegebenenfalls verhindern.

Der Kopf einer Datei mit 5000 Gradienten, einer maximalen Rekursionstiefe von 5, einer Unterteilung des Halbraumes von 512, einer ambienten Auflösung von 32 und einer ambienten Genauigkeit von 0.2 sieht z.B. wie folgt aus:

```
#Gradients v1.0
#Data: 5 512 32 0.2
Number: 5000
```

3.3.4 Extrapolation

Außer dem LightOp wurden noch einige zusätzliche Erweiterungen des Vision-Paketes benötigt, um den Algorithmus optimal einsetzen zu können. Es können sich deutliche Artefakte in den berechneten Bildern ergeben, wenn der Algorithmus am Anfang der Berechnung zur Erzeugung der Pixelwerte stark extrapolieren muß. Der Grund hierfür ist, daß ein normaler Bildebenentracer ein Bild von links oben nach rechts unten berechnet. Hierdurch kann es zu fehlerhaften Extrapolationen kommen, wie die Skizze 3.7 und das mit einem Standard-Tracer erzeugte Bild B.3 verdeutlichen. Die Skizze simuliert den Durchlauf eines Standard-Tracers. Bei Punkt X_1 sind noch keine Irradiance-Objekte vorhanden, es wird deshalb ein neues berechnet. Bei X_2 kommt es dann zu keiner neuen Berechnung eines Irradiance-Objektes, da das Objekt I dort noch gültig ist. Bei X_3 wird ein neuer Irradiance-Wert durch Strahlverschicken über den Halbraum erzeugt. Dieser Wert II hätte dann aber auch Gültigkeit bei X_2. Ähnliches passiert bei X_4. Der Wert dort wird nur aus dem Wert I unter Einbeziehung des Gradienten dort erzeugt, wobei im nachhinein auch der Wert III dort gültig gewesen wäre.

Hierdurch wird deutlich, daß es an vielen Stellen zu fehlerhaften Extrapolationen kommen kann, da dort noch nicht genug Informationen enthalten sind. Um dies zu verbessern, wurden u.a. zwei komplexere Bildebenen-Tracer entwickelt, um diese durch zu regelmäßiges Abtasten entstandene Artefakte zu vermeiden.

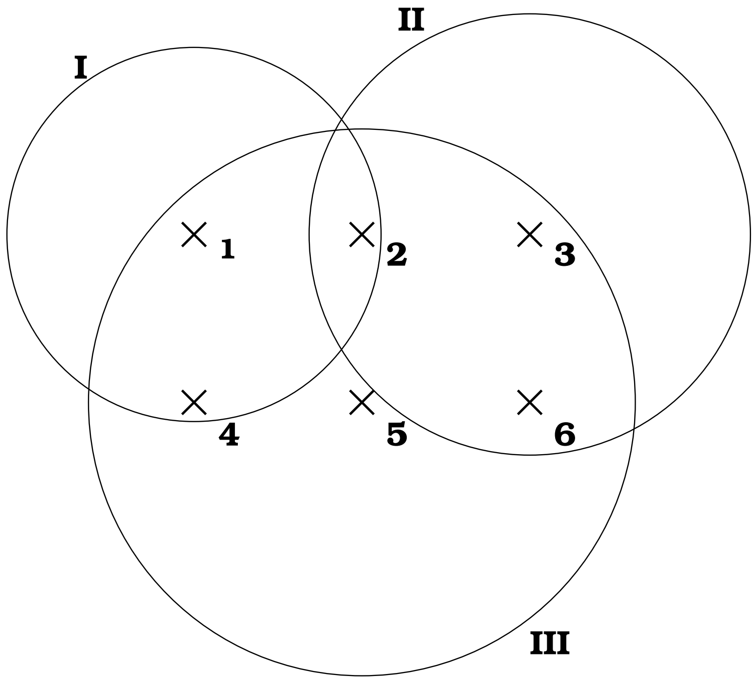

Abbildung 3.7: Die Punkte werden von X_1 nach X_6 abgearbeitet, es findet hier eine fehlerhafte Extrapolation statt.

3.3.4.1 Interlace-Tracer

Um die Regelmäßigkeiten durch lineares Abtasten von links oben nach rechts unten zu vermeiden, wurde der Interlace-Tracer entwickelt, der die Abtastungsschritte besser auf das Bild verteilt. Der Grundgedanke des Interlace-Tracers ist es, die Bildebene zuerst nur grob abzutasten und dann diese Abtastung nach und nach zu verfeinern.
Hierzu wird die Bildebene nicht Pixel für Pixel von links nach rechts und von oben nach unten abgetastet, sondern zuerst einmal nur jeder achte Pixel nach unten und rechts, Wurde dann die rechte untere Ecke erreicht, wird die Abtastung wieder von oben neu begonnen, wobei der Startpixel aber um eins nach unten bzw. rechts verschoben wird.

```
for (offset_y = 0; offset_y < 8; offset_y++) {
  for (offset_x = 0; offset_x < 8; offset_x++) {
    for (y = offset_y; y <= width; y = y + 8) {
      for (x = offset_x; x <= height; x = x + 8) {
        send_ray(x,y);
      }
    }
  }
}
```

Wie man an Bild B.4 sehen kann, werden die sichtbaren Fehler durch Extrapolation hierdurch zwar weniger, verschwinden aber nicht vollständig, was u.a. an der immer noch vorhandenen Regelmäßigkeit des Verfahrens liegt.

3.3.4.2 Zufalls-Tracer

Um die Fehler durch zu regelmäßiges Abtasten zu beheben, wurde der Tracer geschrieben, welcher das Bild mit einer gleichverteilten Zufallsfunktion abtastet, wobei zusätzlich sichergestellt ist, daß kein Pixel doppelt abgetastet wird. Dadurch können alle Artefakte vermieden werden, die durch ein zu gleichmäßiges Abtasten erzeugt wurden.

```
for (y = 0; y <= height; y++) {
  for (x = 0; x <= width; x++) {
    array[(y * (width + 1) + x] = y * width + x;
  }
}

for (s = (width + 1)*(height + 1); s >= 0; s--) {
  t = floor(random() * s);
  x = array[u] mod width;
  y = array[u] / width;
  array[t] = array[s];
  send_ray(x,y);
}
```

`random()` ist eine gleichverteilte Zufallsfunktion zwischen 0 und 1.

Wie in Bild B.5 sichtbar, hat sich der Gesamteindruck noch etwas verbessert, alle Fehler durch Extrapolation konnten aber auch dadurch nicht behoben werden.

3.3.4.3 Vorabtasten

Trotz der Verbesserungen gegenüber dem Standard-Tracer können die beiden oben genannten Verfahren eine Extrapolation aufgrund noch nicht abgetasteter Bereiche nicht ganz ausschließen.

Hier bieten sich zwei Möglichkeiten an, um dieses Problem zu lösen:

- Man läßt das Bild noch einmal mit einer bereits fertig berechneten Gradienten Datei erzeugen. Der Nachteil hiervon aber ist, daß zwei vollständige Durchläufe notwendig sind.
- Im Gegensatz dazu genügt es eigentlich, das Bild nur grob vorabzutasten und die hieraus gewonnenen Werte dann zur erneuten Berechnung des Bildes zu verwenden. Die bei der

Vorabtastung gewonnenen Pixelwerte müssen zwar bei der zweiten Abtastung auch neu berechnet werden, was aber einen geringeren Aufwand gegenüber einer zweimaligen Vollabtastung darstellt, da der Gesamtalgorithmus nur ein einziges Mal aufgerufen werden muß.

Wie man in Bild B.6 sieht, kann man durch ein Vorabtasten die Fehler durch Extrapolation beseitigen, es wird aber auch, wie die Zeiten aus Tabelle 3.8 zeigen, ein etwas größerer Zeitaufwand benötigt.

Art des Verfahrens	*Zeit (in Sekunden)*
Einmaliger Durchlauf	387
Vorabtasten (8er Schritte)	394
Zweimaliger Durchlauf	409

Abbildung 3.8: Vergleichszeiten für einfaches, zweifaches und Vor-Sampling

3.3.4.4 Bewertung

Die Ergebnisse der vorangegangenen Tests zeigen, daß die Qualität des berechneten Bildes auch sehr stark vom verwendeten Tracer abhängt. Ein zu gleichmäßiger Durchlauf kann zu sehr starken Artefakten führen, was durch einen daraufhin modifizierten Tracer-Algorithmus verbessert werden kann, wofür auch kein zusätzlicher Zeitaufwand notwendig ist.
Wenn man aber das Auftreten dieses Artefakte vollständig verhindern will, muß man die Szene mehrfach abtasten. Dies führt zu sehr guten Resultaten, es entsteht hier aber ein Mehraufwand an Zeit.

Kapitel 4

Erweiterungen des Verfahrens

Hier werden einige mögliche und auch realisierte Erweiterungen des Algorithmus vorgestellt.

4.1 Supersampling

Bei einem Abtasten der Umgebung mit relativ wenig Strahlen kann es manchmal zu einer Überbewertung einzelner Strahlen kommen, d.h. daß einige Strahlen stark von ihren Nachbarstrahlen in der Helligkeit abweichen und so in dieser Richtung eine starke Änderung der Helligkeit suggerieren. Dies kann aber durch das zufällige Treffen von kleinen lokalen Änderungen der Helligkeitsverteilung hervorgerufen werden. Diese lokalen Stellen werden dann in dem endgültigen Irradiance-Wert zu stark oder zu schwach gewichtet.
Hier hilft ein nachträgliches erneutes Abtasten der betreffenden Bereiche. Um diese Bereiche zu finden, wurde folgender Algorithmus implementiert:

> `Solange` $(S = S - 1) > 0$
> $V(S) = \sum_{i=1}^{4} \texttt{bright}(C_i) - \texttt{bright}(C_0)$,
> `Sortiere` V `absteigend nach Größe.`
> `Solange` $(N = N - 1) > 0$
>   `Verschicke zu` $V(0)$ `gehörigen Strahl erneut,`
>   `Mittele Ergebnis mit altem Wert,`
>   `Berechne` $V(0)$ `erneut,`
>   `Sortiere` V `erneut.`

N ist die vom Benutzer angegebene Zahl der Supersamples, welche auch hier mit einem Gewichtungsfaktor über die Rekursion (siehe Sektion 2.4.3) multipliziert wird. S ist die Zahl der verschickten Strahlen und $V(X)$ die am Strahl X berechnete Varianz. C_0 ist die Farbe an der aktuellen Position, C_i die Farbe des i-ten Nachbarn und `bright`(C) berechnet die Helligkeit der Farbe C.
Je höher die Varianz V einer Zelle ist, desto größer ist die Wahrscheinlichkeit für einen Fehler an dieser Stelle, und desto wichtiger ist es, dort erneut Strahlen zu verschicken.

4.2 Verbesserung der ambienten Grundhelligkeit

Obwohl mit dem hier vorgestellten Verfahren die ambiente Helligkeit berechnet wird, ist es doch sinnvoll, die ungefähre Grundhelligkeit anzugeben, d.h. einen durchschnittlichen Wert, der als gemittelte ambiente Helligkeit in einer Szene erwartet wird, da dieser im Fall des Rekursionsabbruchs verwendet wird. Wenn dieser angegebene Wert von der tatsächlichen Hintergrundhelligkeit zu weit abweicht, kann es passieren, daß das Bild zu hell oder zu dunkel wird. Dies ist besonders dann der Fall, wenn der Wert für die maximale Rekursionstiefe nicht sehr groß gewählt wurde.
Da die tatsächliche Hintergrundhelligkeit aber häufig vorher noch nicht bekannt ist, wurde ein Verfahren implementiert, daß aus den bereits berechneten Irradiance-Werten eine neue Grundhelligkeit errechnet. Der Gedanke ist, zwar am Anfang die von außen angegebene Grundhelligkeit zu verwenden, wenn dann aber schon Irradiance-Werte berechnet wurden, aus diesen und der angegebenen Grundhelligkeit eine neue, wahrscheinlich präzisere Grundhelligkeit zu interpolieren.
Hierbei kann die initiale Helligkeit gewichtet werden, um bei anfänglich noch sehr wenigen Irradiance-Werten keinen zu großen Fehler zu begehen. Die neue Grundhelligkeit E wird nach folgender Formel berechnet:

$$E = \frac{(W \cdot E_i + N \cdot E_s)}{W + N},$$

wobei E_i die angegebene Grundhelligkeit,
E_s die aufsummierte berechnete Irradiance,
W das Gewicht der angegebenen Grundhelligkeit und
N die Anzahl der bereits berechneten Irradiance-Werte ist.

4.3 Numerische Probleme

Durch numerische Ungenauigkeiten der Berechnungen beim Raytracing kann es passieren, daß ein Strahl eine dem Abschußpunkt zu nahe gelegene Wand nicht trifft, und durch diese deshalb hindurchgeht. Dieser Fehler, der auch in der Implementation in Radiance [6] vorhanden ist, kann dadurch stark abweichende Irradiance-Werte erzeugen, die sich, wie in dem mit Radiance berechneten Ausschnitt B.16 aus der Szene B.15 sichtbar, als dunkle oder helle Flecken hervorheben können.

Wenn ein Strahl, wie in Abbildung 4.1 dargestellt, in eine Ecke trifft, werden Strahlen beim anschließenden Verschießen über eine Halbkugel einen geschlossenen Körper verlassen können. So liegt in Abbildung 4.1 der Punkt P in solch einer Kante. Der Strahl S_1 wird von der Wand weg verschickt, stellt also kein Problem dar, der Strahl S_2 hingegen, der die Wand eigentlich treffen sollte, schneidet diese nicht und verursacht so einen Fehler.

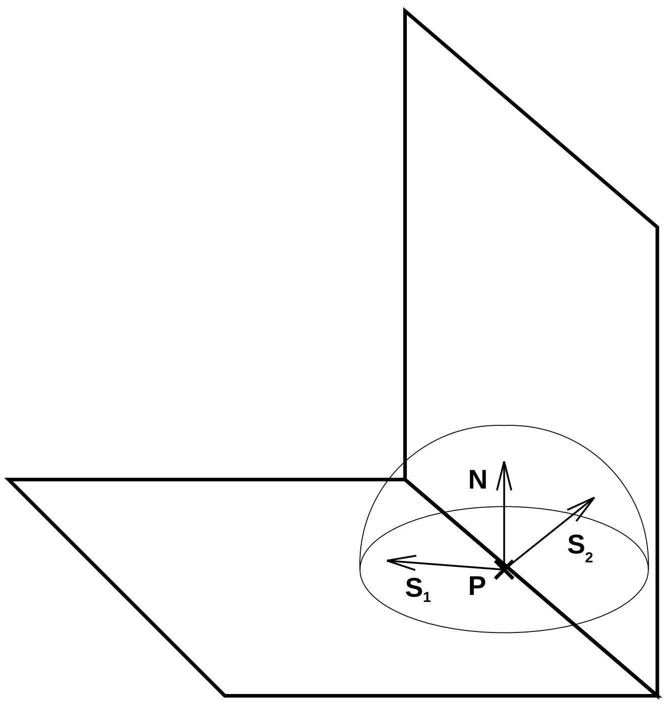

Abbildung 4.1: Der Ausgangspunkt P ist genau in einer Ecke, der Strahl S_2 kann deshalb eine geschlossene Szene verlassen.

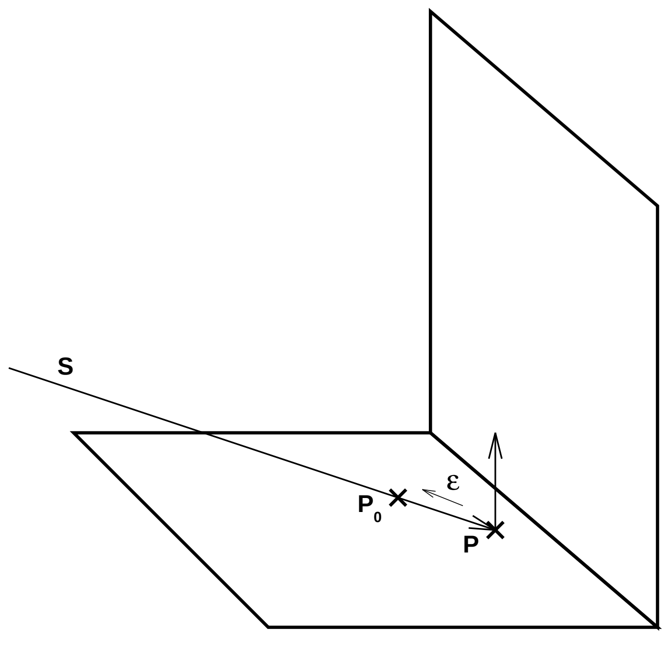

Abbildung 4.2: Verschiebung des Ausgangspunktes um ϵ, kein Strahl kann die Szene mehr verlassen.

Die hier entwickelte Lösung dieses Problems sieht vor, daß das Verschicken des Strahls nicht vom eigentlichen Ausgangspunkt auf einer Fläche ausgeht. Vielmehr wird der Startpunkt, wie in Abbildung 4.2 gezeigt, um ein Epsilon in Gegenrichtung zum einfallenden Strahl S hin verschoben. Hierdurch wird sichergestellt, das die Entfernung von Ausgangspunkt zu einer Fläche groß genug ist, um diese Fläche bei einer Schnittpunktberechnung zu treffen. Dies gewährleistet, daß die oben beschriebenen fehlerhaften Fälle hierdurch vermieden werden.

4.4 Indirekter Schattenwurf

Der direkte Schattenwurf wird zwar in einem anderen Teil des Raytracing Ansatzes behandelt, es kann aber auch bei der Berechnung des diffusen Anteils zu gewissen Schattengrenzen kommen. Hierfür zeigen sich sogenannte 'diffuse Lichtquellen' verantwortlich. Dies sind stark reflektierende Flächen, die z.B. von einer Lichtquelle direkt angestrahlt werden, und dann einen großen Teil dieses einfallenden Lichts reflektieren. Durch Verdeckung kann es dann passieren, daß sich hieraus harte Schattenkanten ergeben sollten. In Bild B.8 und B.9 wurde eine diffuse Lichtquelle dadurch erzeugt, daß eine helle Punktlichtquelle in eine nach einer Seite (hier: links) offenen Kiste in einen sonst unbeleuchteten Raum plaziert wird. Dadurch wird die der offenen Seite zugewandte Raumwand durch die reflektierenden Innenflächen der Lichtkiste angestrahlt.

Durch den Interpolationsansatz von Irradiance Gradients können solche Effekte aber, wie in Bild B.8 sichtbar, stark verwaschen werden. Die einfachste Methode, solche Artefakte zu verbessern, ist es, solche Szenen mit einer höheren Auflösung abzutasten. So wurde zur Berechnung von Bild B.9 sowohl die Anzahl der Strahlen als auch die maximale Rekursionstiefe erhöht.

Dies führt aber zu einem stark gestiegenen Zeitaufwand, wobei das Ergebnis zwar besser ist, aber noch lange nicht optimal. Solche Effekte stellen eine Schwäche des Irradiance Gradients-Algorithmus dar. In Sektion 4.6 wird später ein neuer Ansatz kurz vorgestellt, der möglicherweise auch diese Effekte korrekt berechnet.
Abbildung 4.3 und 4.4 zeigen die Positionen der in dieser Szene erzeugten Gradienten. Es wird deutlich, daß in den Gradienten alleine nicht genug Informationen enthalten sind, um diese 'diffusen Lichtquellen' korrekt darzustellen.

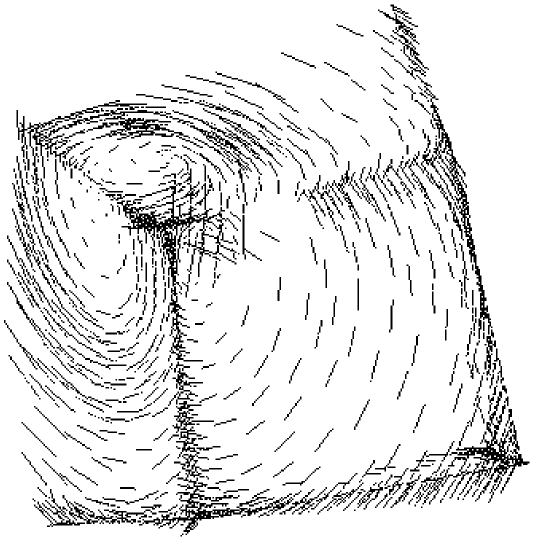

Abbildung 4.3: Rotationsgradienten bei einer diffusen Lichtquelle

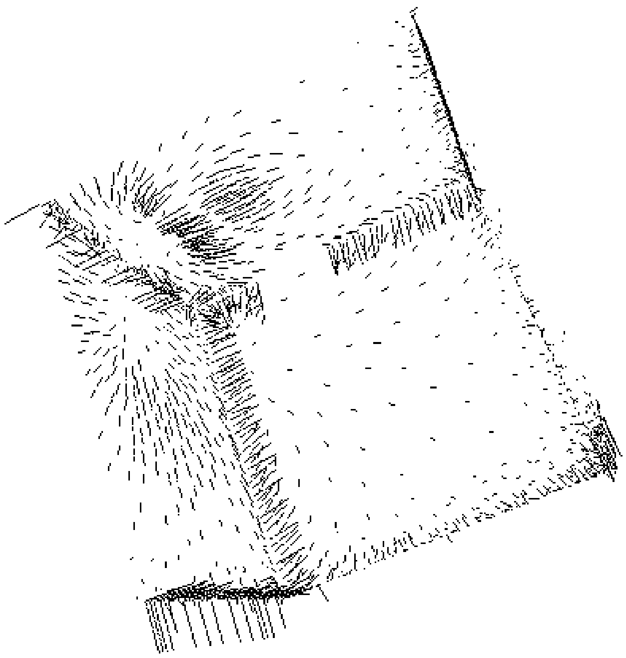

Abbildung 4.4: Translationsgradienten bei einer diffusen Lichtquelle

4.5 Mögliche Erweiterungen des Verfahrens

4.6 Interpolation von Bildern

Es wäre durchaus möglich, bei der Interpolation nicht nur die Irradiance-Werte, sondern sogar die vorher berechneten Bilder der Umgebung zu interpolieren. Hierbei wäre es aber wünschenswert, daß die Umgebung nicht über eine Halbkugelparametrisierung, sondern über eine Ebenenparametrisierung abgetastet wird. Dies würde eine einfachere Interpolation der so gewonnenen Bilder ermöglichen.
Zur Interpolation der Bilder stünde dann z.B. der Ansatz von Chen [1] zur Verfügung.

4.6.1 Abtastung über eine Ebene

Dieser Ansatz sieht vor, die Abtastung über eine ebene quadratische Einheitsfläche, die sich im geringen Abstand vom Ausgangspunkt senkrecht zu diesem befindet, zu parametrisieren. Diese Fläche wird gleichförmig in gleich große Zellen unterteilt, wobei dann ein Strahl durch jede dieser Zellen vom Ausgangspunkt ab in die Umgebung geschickt wird. Dadurch entsteht ein planares Bild der Raumes, was sehr gut für eine Interpolation verwendet werden kann. Nachdem bei diesem Abtasten aber noch keine Gewichtung der einzelnen Richtungen erfolgt, was aber für die Irradiance-Berechnung notwendig ist, wird zu jeder Zelle auf der Einheitsfläche der betreffende Formfaktor F nach folgender Formel berechnet [2]:

$$F_{dA_i \to A_j} = \frac{1}{2\pi} \sum_{i=1}^{n} \beta_i N_i \bullet \overline{(R_i \times R_{(i+1)\%n})},$$

wobei N_i die Normale der Zelle i und β_i der Winkel zwischen R_i und $R_{(i+1)\%n}$ ist. Ferner normalisiert $\overline{X}$ den Vektor X.

Da durch die endliche Größe der Einheitsfläche die Summe aller Formfaktoren immer zu klein ist, wird jeder Wert wieder durch einen Korrekturfaktor K aufgehellt:

$$K = \pi - \sum_{\mathbf{S}} F,$$

wobei $\mathbf{S}$ die Menge aller verschickten Strahlen über die Einheitsebene ist.

Die Anpassung der Berechnung der Gradienten auf diese Ebenenabtastung wurde nicht mehr beendet. Da es aber das Ziel dieses Ansatzes der Ebenenabtastung ist, hier nicht mehr die Gradienten, sondern die gesamten Umgebungsbilder an den betreffenden Stellen zu interpolieren, ist die Gradientenberechnung für dieses Vorhaben auch nicht notwendig.
Für den Algorithmus der Interpolation auf Gradientenbasis stellt die Verwendung einer Ebenenparametrisierung alleine keine Verbesserung, sondern eher eine Verschlechterung des Verfahrens da. Deshalb ist es hier nicht sinnvoll, diese Ebenenparametrisierung direkt auf den Irradiance Gradients-Algorithmus anzuwenden.

Kapitel 5

Visualisierung der Werte

Zur besseren Bewertung und Verifizierung der Ergebnisse der Verfahren wurde ein zusätzliches Programm entwickelt, daß die berechneten Irradiance-Objekte mit verschiedenen Methoden graphisch darstellen kann.

5.1 Verteilung der Irradiance-Werte

Die räumliche Position der berechneten Irradiance-Objekte sollte stark von der Szenengeometrie abhängig sein. Bei vielen Objekte nahe beisammen sollten mehr Objekte berechnet werden, bei isolierten Flächen hingegen möglichst wenige.
Mit diesem Verfahren kann man die räumliche Lage der Irradiance-Objekte in der Szene graphisch in die bestehende Rib-Szene einfügen. Dazu werden die Objekte an den betreffenden Stellen als Kugeln eingeblendet. Hierdurch kann die Verteilung der Irradiance-Objekte verifiziert werden.

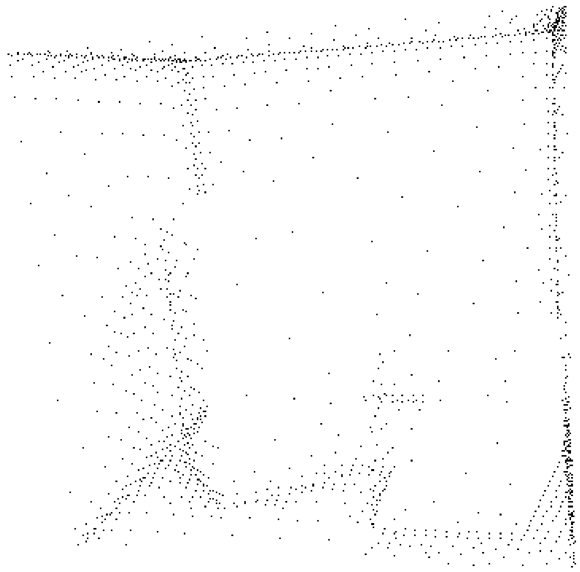

Abbildung 5.1: Irradiance-Werte der ersten Rekursionsstufe

Die Position der Irradiance-Werte der ersten Rekursionsebene, welche in Abbildung 5.1 separat zu sehen sind, wurden in Bild B.10 durch dieses Verfahren in die ursprüngliche Szene als Halbkugeln eingefügt. Dies sind die Werte, die dann auch tatsächlich für die endgültige Interpolation verwendet werden.
Man kann hier erkennen, daß an den Kanten, aber auch an Stellen, an denen die zwei Kisten eng beieinander sind, mehr Irradiance-Werte berechnet wurden, als in der Mitte von ebenen Flächen.

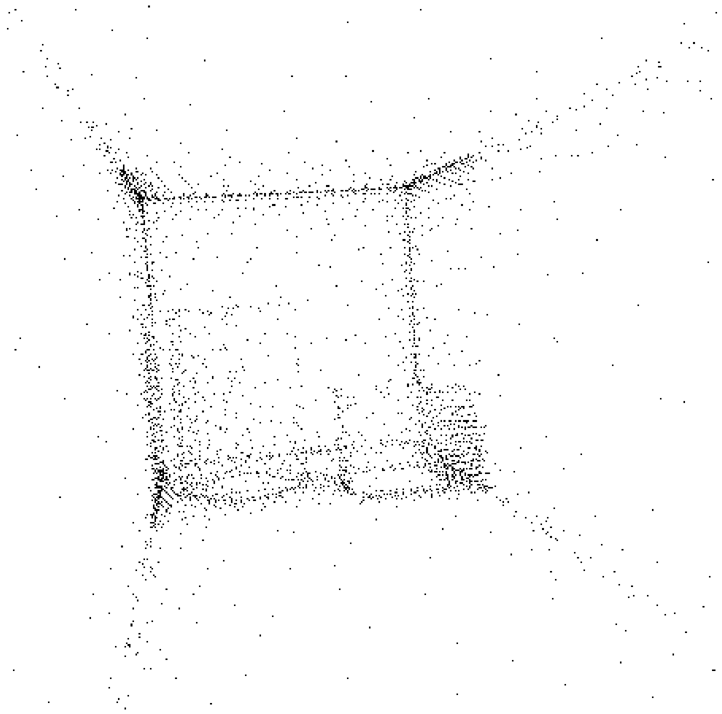

Abbildung 5.2: Irradiance-Werte aller Rekursionsstufen

Die Abbildung 5.2, und das zugehörige Bild B.11 zeigen nun die berechneten Irradiance-Werte aus allen vorhandenen Rekursionstiefen.
Da die äußere Box nach hinten verlängert ist, kann man hier gut erkennen, daß auch Irradiance-Werte an Stellen berechnet wurden, die nicht im vom Augpunkt aus sichtbar sind. Dies sind zwangläufig Werte aus tieferen Rekursionsstufen. Ferner kommt es vor, daß mehrere Irradiance-Werte an Stellen berechnet worden sind, an denen eigentlich ein einzelner Wert genügen müßte. Dies kommt dadurch zustande, daß Werte aus tieferen Rekursionsstufen in höheren Rekursionsstufen nicht verwendet werden dürfen. Deshalb muß dann in der höheren Rekursionsstufe an dieser Stelle erneut ein Irradiance-Wert berechnet werden.

5.2 Bild der Umgebung

Um die Berechnung der Gradienten zu verifizieren, zeigt dieses Verfahren die Umgebung, wie sie von einem Punkt, an dem ein neuer Irradiance-Wert berechnet wurde, gesehen wird. Diese Darstellung wurde verwendet, um mögliche Erweiterungen des Algorithmus zu finden. So lassen sich auf den Umgebungsbildern sehr gut Schattengrenzen erkennen, die durch das in Sektion 2.5.1 beschriebene Auswahlverfahren zur Bestimmung der Gültigkeit der Irradiance-Werte nur mäßig gut erkannt werden (siehe Sektion 4.4). Eine geplante Verbesserung stellt die Verwendung des gesamten Bildes der Umgebung zur Interpolation, wie in Sektion 4.6 beschrieben, dar. Als Beispiel wurden hier zwei Auflösungen berechnet, Bild B.12 und B.13 mit ca. 120 Unterteilungen und Bild B.14 mit ca. 2000 Unterteilungen. Bild B.12 stellt einen realistischen Wert dar, mit dem typischerweise die Umgebung abgetastet wird.

In Bild B.14 kann man hingegen gut erkennen, wie durch das Abtasten schon ein deutliches Abbild der Umgebung entsteht.

5.3 Darstellung der Gradienten

Hier wurden die erzeugten Gradienten mit Hilfe von Open Inventor graphisch an ihrer jeweiligen Position dargestellt, und zwar in den Abbildungen 5.3 und 5.5 die Translationsgradienten und 5.4 und 5.6 die Rotationsgradienten. Dadurch kann man sehr gut die Verteilung und die speziellen Eigenschaften der Gradienten erkennen. Die Größe der einzelnen Gradienten sollte von der räumlichen Verteilung und Helligkeit der Flächen abhängen, die von Berechnungspunkt sichtbar sind. Bei starken Änderungen innerhalb der Umgebung sollte der Gradient groß sein.

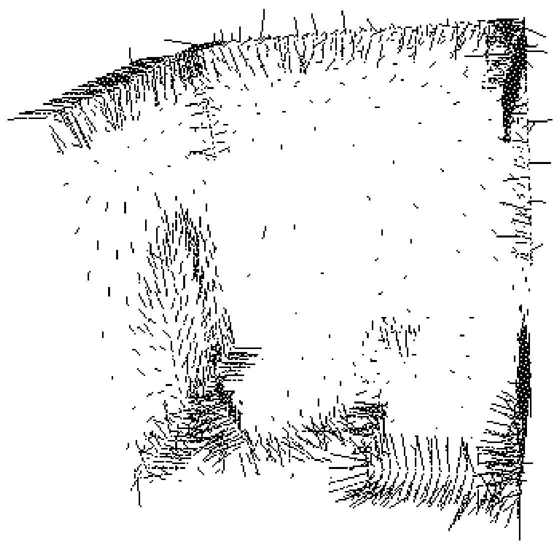

Abbildung 5.3: berechnete Translationsgradienten, nur erste Rekursionsstufe

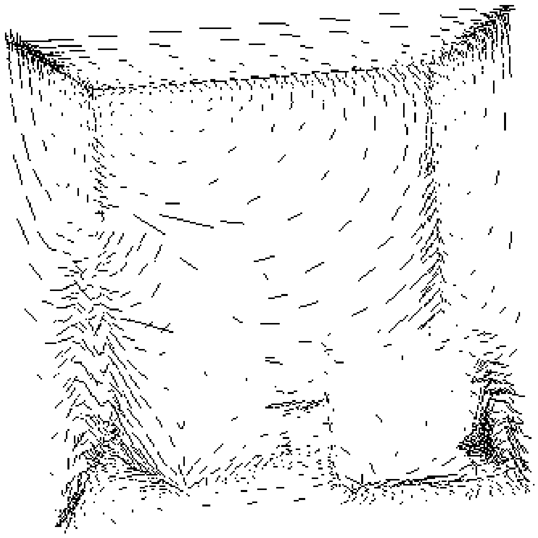

Abbildung 5.4: berechnete Rotationsgradienten, nur erste Rekursionsstufe

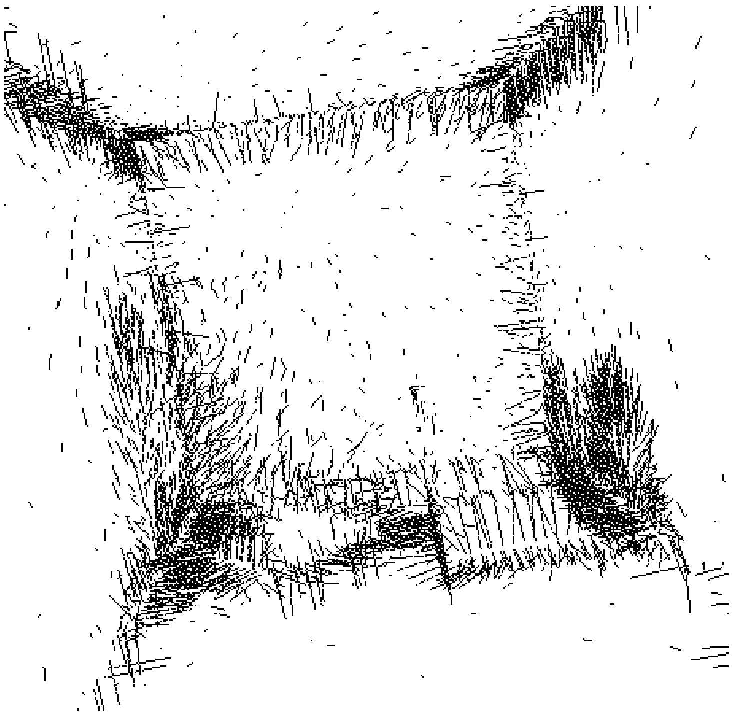

Abbildung 5.5: berechnete Translationsgradienten aller Rekursionsstufen

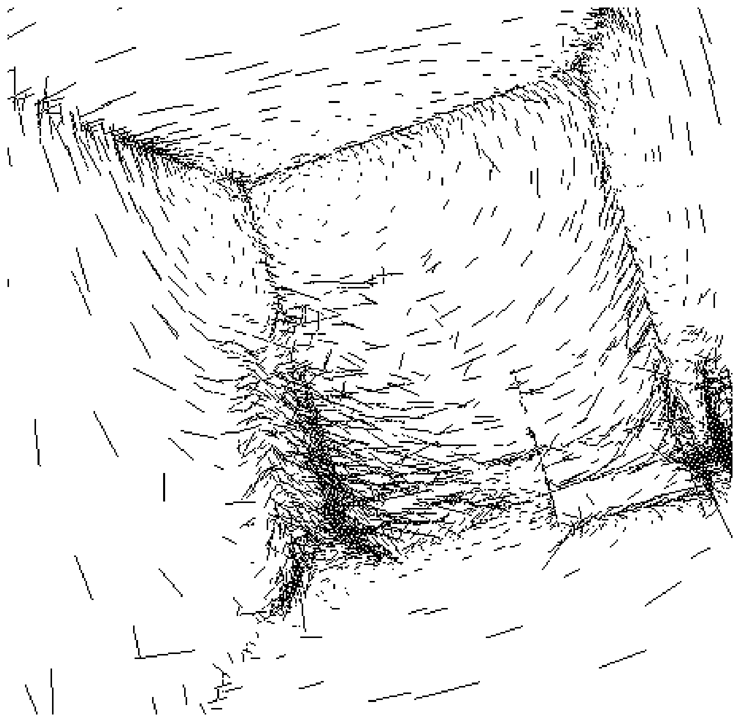

Abbildung 5.6: berechnete Rotationsgradienten aller Rekursionsstufen

Kapitel 6

Bewertung

Der hier vorgestellte und implementierte Algorithmus stellt eine effiziente Methode dar, diffuse Beleuchtung auch mit einem Raytracing-Ansatz zu berechnen. Er hat zwar durchaus seine Beschränkungen, welche aber zum großen Teil durch einige kleinere Erweiterungen behoben werden konnten. So ist die Verwendung eines auf Irradiance Gradients abgestimmten Tracer essentiell, auch mit Vorabtasten des Bildes und Supersampling an kritischen Stellen ließ sich das Berechnungsergebnis stark verbessern.
Die Hauptschwäche des Algorithmus, diffuse Lichtquellen und deren Schattenlinien nicht richtig zu erkennen, und deshalb die durch diese entstehenden harten Schattengrenzen zu verwaschen, kann aber mit einzelnen kleineren Verbesserungen nicht gelöst werden.
Hierzu wird in der Folgezeit versucht, den hier schon kurz angerissenen Ansatz der Interpolation der Umgebungsbilder weiter zu entwickeln und dadurch den Irradiance Gradients-Algorithmus auch in dieser Richtung zu verbessern. Für weitere Informationen siehe die dazugehörigen Online Seiten unter **http://www9.informatik.uni-erlangen.de/~msott**.

Anhang A

Parameter

Die Parameter, die zur Feineinstellung des Verfahrens dienen, sind in Übereinstimmung mit dem RenderMan-Standard als Optionen auf Datei-Basis implementiert worden, wobei außerdem die Möglichkeit der Einstellung dieser zur Laufzeit unterstützt wird.
Die einzelnen Parameter, mit denen der Benutzer das Verhalten des Verfahrens steuern kann, werden hier beschrieben.

A.1 Art des Speichers

Man kann hier den Typ des Speichers angeben, in dem bereits berechnete Irradiance-Werte abgelegt werden.
Für eine genauere Beschreibung siehe Sektion 3.3.2.
Option: `store` $\mapsto$ `none, liststore, octreestore, octreewardstore`

A.2 Art des Tracers

Hiermit kann man den Algorithmus bestimmen, mit dem die Strahlen in der Szene verschossen werden.
Für eine genauere Beschreibung siehe Sektion 2.4.2 und 4.6.
Option: `sampler` $\mapsto$ `rad_hemisphere, mc_hemisphere, plane`

A.3 Unterteilung

Hier wird die Anzahl der Strahlen angeben, die in der ersten Rekursionstiefe über die Halbkugel verschickt werden. Der Fehler der Monte Carlo Berechnung der indirekten Beleuchtung ist invers proportional zu der Wurzel dieser Zahl [10].
Man beachte, daß die Anzahl N der Strahlen über die Rekursion nach der folgenden Formel abnimmt:

$$N = N_i \cdot 0.5^R$$

N_i ist die anfängliche Anzahl der Strahlen
R ist die Rekursionsstufe, wobei bei 0 zu zählen angefangen wird.

Da das Gewicht der einzelnen Strahlen mit steigender Rekursionstiefe abnimmt, steigt der Fehler, den ein einzelner Strahl verursachen kann, durch die Verringerung der Strahlen über die Rekursion nicht an.
Für eine genauere Beschreibung siehe Sektion 2.4.2.
Option: `division` $\mapsto$ n

A.4 Rekursion

Dies setzt die Anzahl der Rekursionsschritte, die maximal berechnet werden. Da die Anzahl der Strahlen aber mit der Rekursionstiefe abnimmt, muß die maximale Rekursionstiefe nicht unbedingt erreicht werden. Ein Wert von Null schaltet die Berechnung der indirekten Beleuchtung ab.
Für eine genauere Beschreibung siehe Sektion 2.4.4.
Option: `recursion` $\mapsto$ n

A.5 Ambiente Genauigkeit

Dieser Wert gibt die Genauigkeit der Berechnung an. Man kann damit ungefähr den Fehler, der bei der Interpolation der Irradiance-Werte maximal auftreten kann, angeben. Ein Wert von Null bedeutet keine Interpolation.
Je kleiner man den Wert wählt, desto präziser wird die Berechnung, es werden dann aber auch mehr Irradiance-Objekte erzeugt und die Berechnung des Bildes dauert länger.
Option: `ambacc` $\mapsto$ n

A.6 Ambiente Auflösung

Dies gibt die ambiente Auflösung an, was bedeutet, daß der Fehler größer wird auf Flächen, deren Abstand voneinander geringer als folgender Wert ist:

$$\frac{\texttt{Größe der Szene}}{\texttt{ambiente Auflösung}}$$

Die maximale Dichte der Irradiance-Objekte wird durch folgende Formel bestimmt:

$$\frac{\texttt{Größe der Szene} * \texttt{ambiente Genauigkeit}}{\texttt{ambiente Auflösung}}$$

Ein Wert von Null bedeutet hier unbegrenzte Auflösung. Die ambiente Auflösung wird wie folgt berechnet:

$$\texttt{minimaler ambienter Radius} = \frac{\texttt{Szenengröße}}{\texttt{ambiente Auflösung}}$$

$$\texttt{maximaler ambienter Radius} = 64 * \texttt{minimaler ambienter Radius}$$

Option: `ambres` $\mapsto$ n

A.7 Ambiente Farbe

Hiermit wird die Farbe, die bei einem Rekursionsabbruch anstatt einer weiteren Berechnung zurückgeliefert wird, gesetzt.
Für eine genauere Beschreibung siehe Sektion 4.2.
Option: `amb_color` $\mapsto$ `[`n n n`]`
Option: `amb_weight` $\mapsto$ n

A.8 Anzahl der Supersamples

Man kann hiermit die maximale Anzahl der Supersamples angeben, die berechnet werden, wenn benachbarte Strahlen einen zu großen Unterschied aufweisen.
Für eine genauere Beschreibung siehe Sektion 4.1.
Option: `ssamples` $\mapsto$ n

A.9 Name der Ausgabedatei

Hiermit kann man den Namen der Ausgabedatei bestimmen, in die die Irradiance-Objekte nach der Berechnung der Szene gespeichert werden.
Für eine genauere Beschreibung siehe Sektion 3.3.3.
Option: `filename` $\mapsto$ *name*

Anhang B

Bildteil

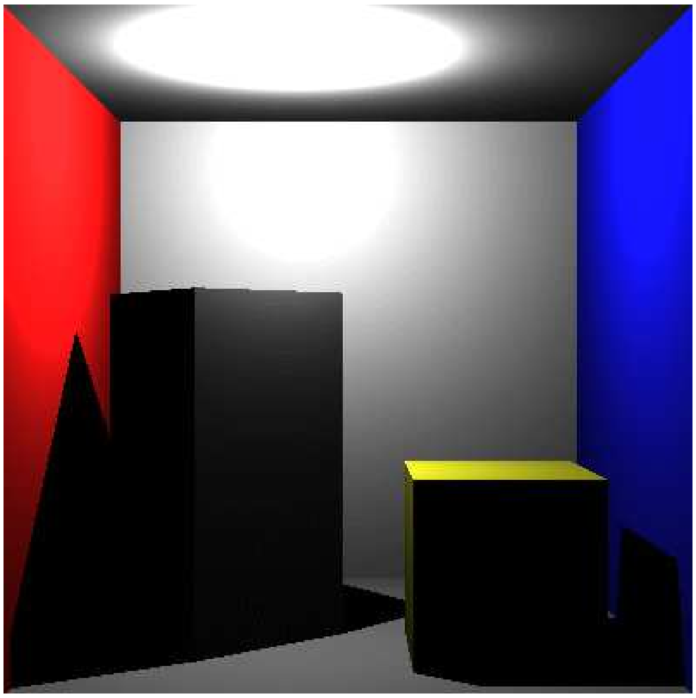

Abbildung B.1: Bild ohne indirekte Beleuchtung berechnet

Abbildung B.2: Bild mit direkter und indirekter Beleuchtung berechnet

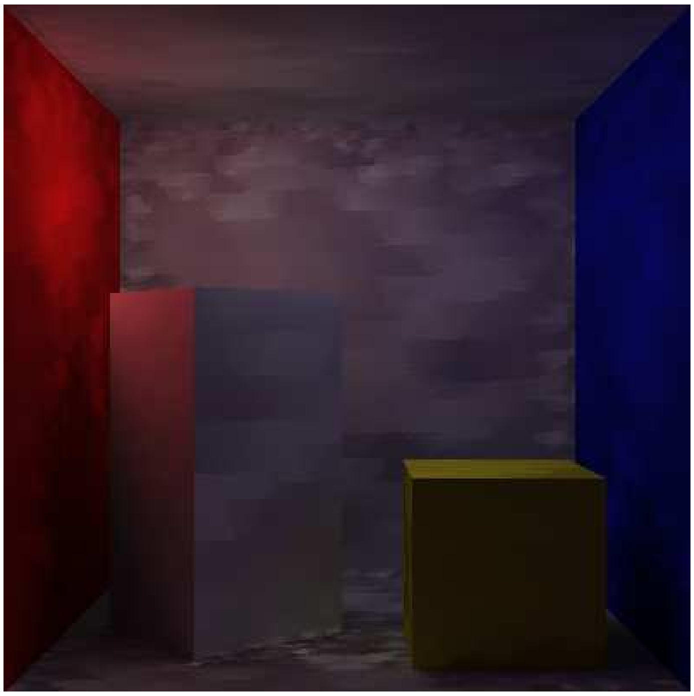

Abbildung B.3: Bild mit Standard-Tracer berechnet, nur diffuse Beleuchtung, man sieht Fehler durch schlechte Extrapolation.

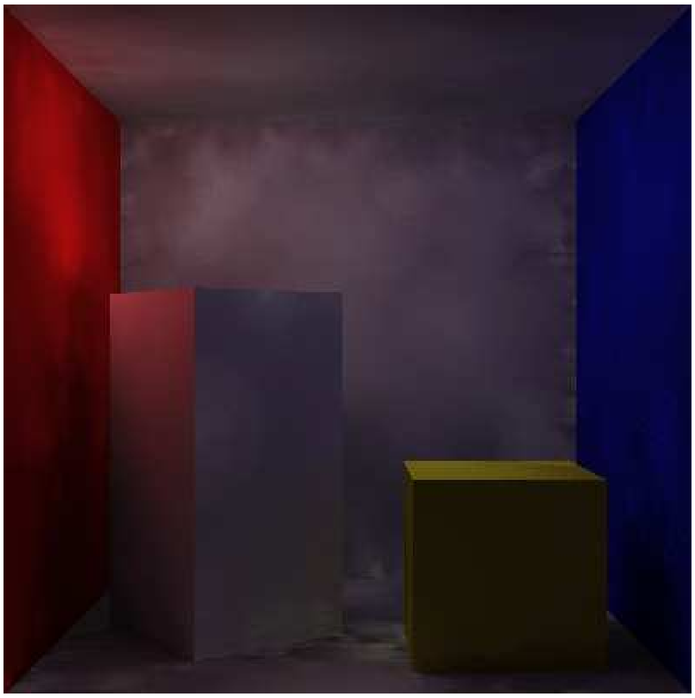

Abbildung B.4: Bild mit Interlace-Tracer berechnet, nur diffuse Beleuchtung, es sind weniger sichtbare Extrapolationsfehler gegenüber dem Standard-Tracer vorhanden.

Abbildung B.5: Bild mit Zufalls-Tracer berechnet, nur diffuse Beleuchtung, die Extrapolationsfehler sind fast verschwunden.

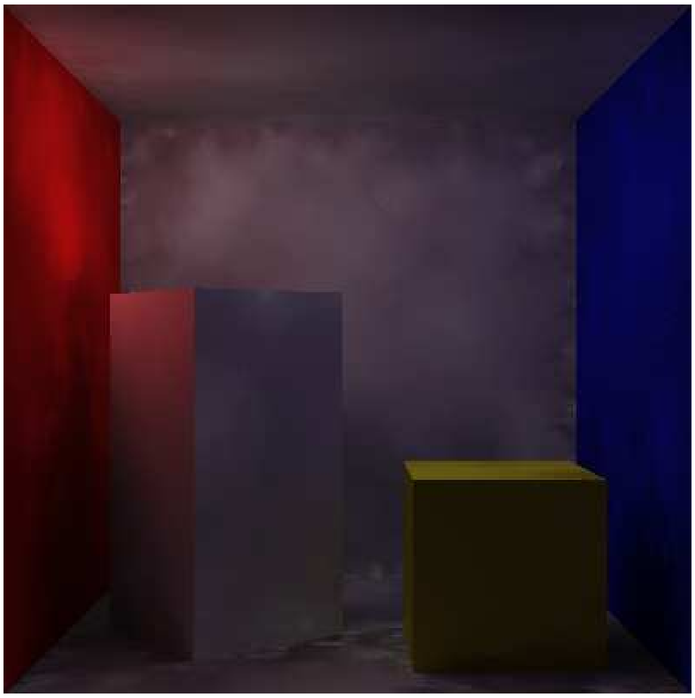

Abbildung B.6: Bild mit Interlace-Tracer und Vorabtasten berechnet, nur diffuse Beleuchtung, die Fehler durch Extrapolation sind verschwunden.

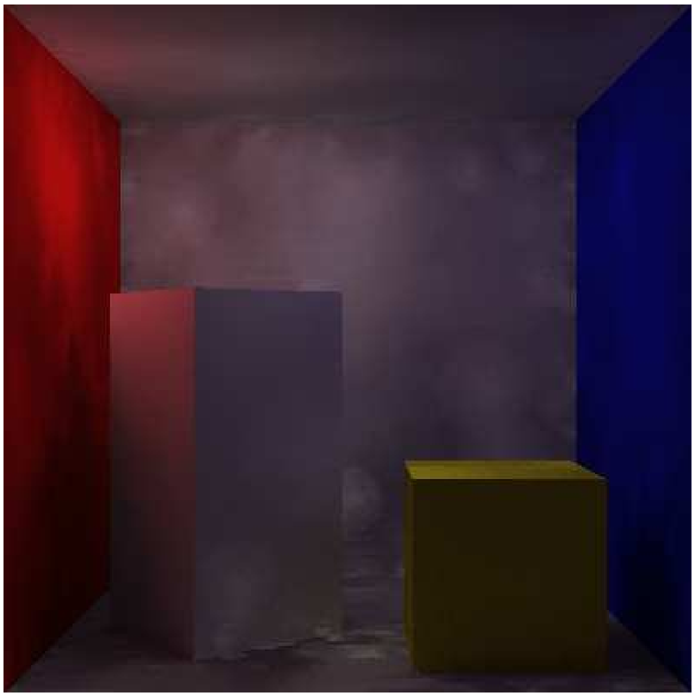

Abbildung B.7: Bild mit schwarzer initialer Grundhelligkeit und Mittelungsalgorithmus berechnet

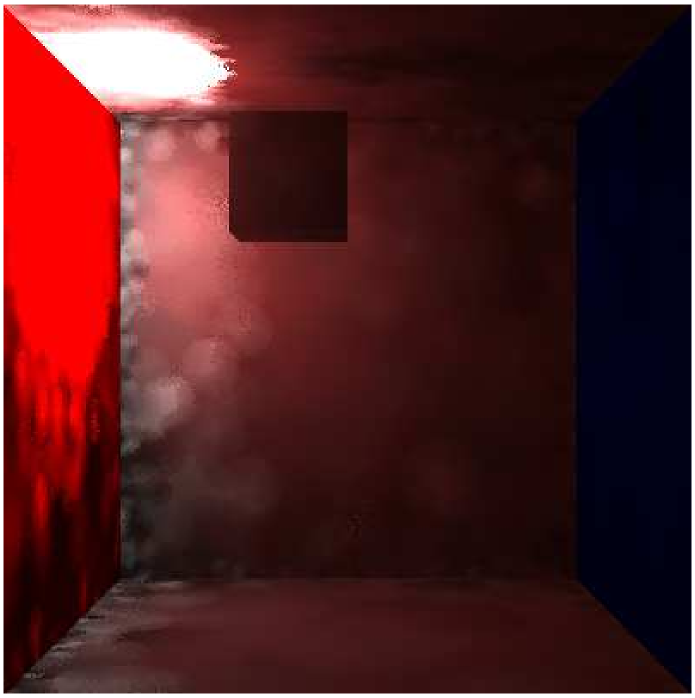

Abbildung B.8: indirekte Beleuchtung, circa 200 Strahlen, maximal 2 Rekursionsebenen

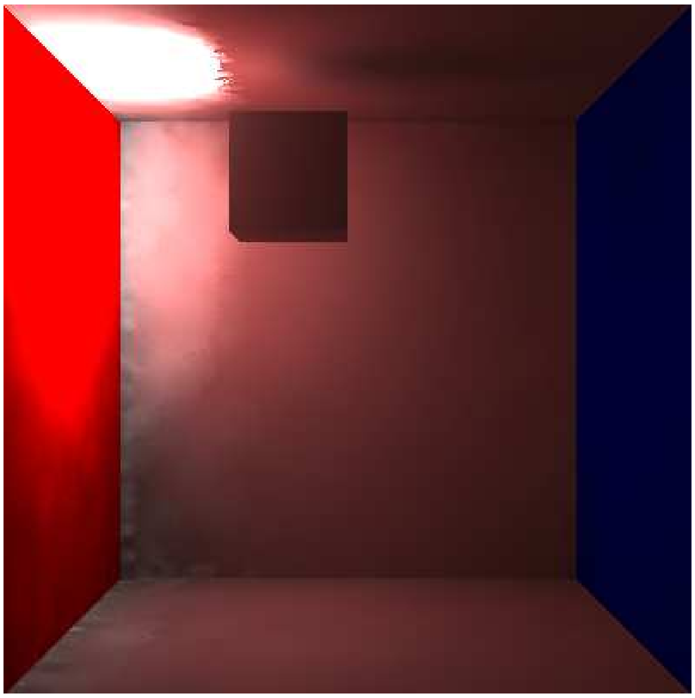

Abbildung B.9: indirekte Beleuchtung, circa 1000 Strahlen, maximal 6 Rekursionsebenen

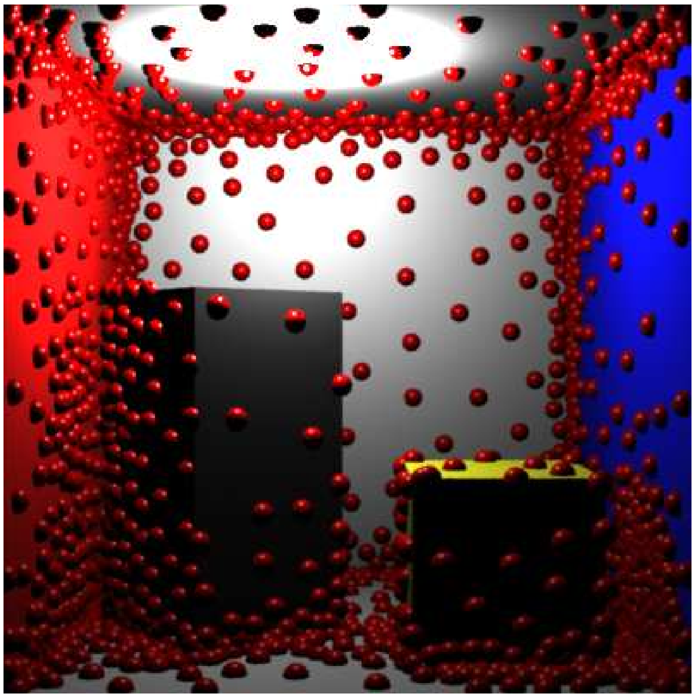

Abbildung B.10: Die Lage der Irradiance-Werte der ersten Rekursionsebene (es wurde hier kein Schattenwurf berechnet)

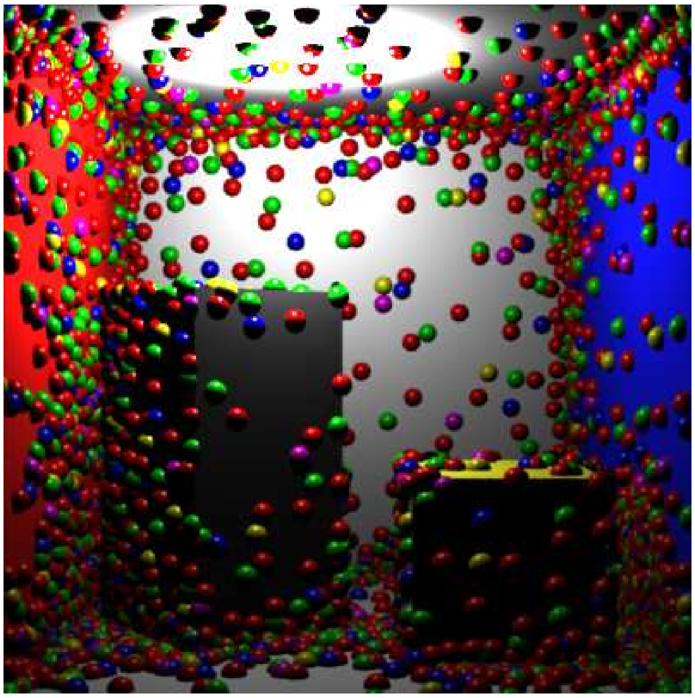

Abbildung B.11: Die Lage der Irradiance-Werte der ersten 5 Rekursionsebenen (es wurde hier kein Schattenwurf berechnet)

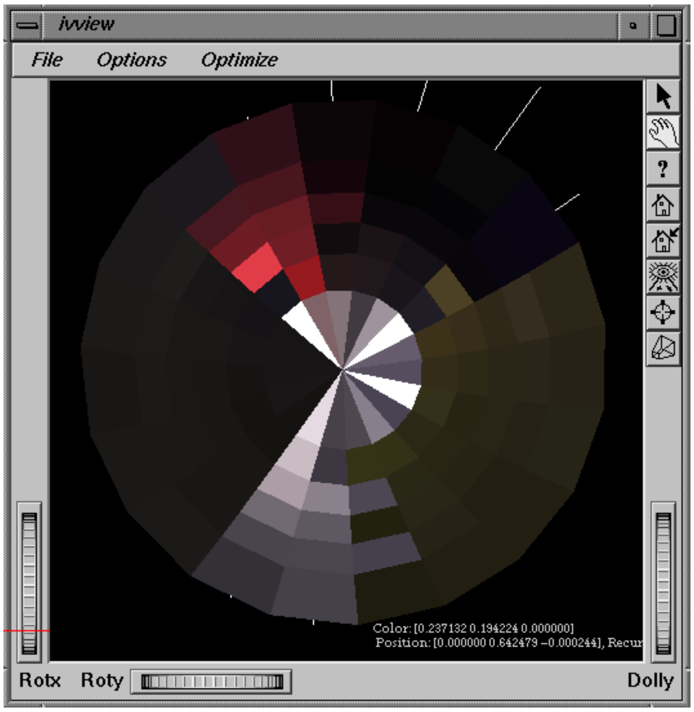

Abbildung B.12: Bild durch Halbkugel-Sampling mit circa 120 Strahlen erzeugt

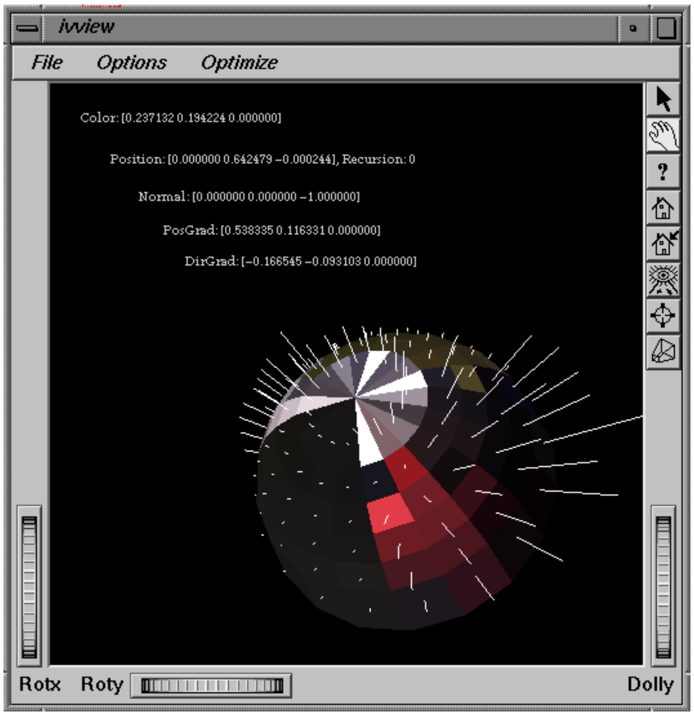

Abbildung B.13: Bild durch Halbkugel-Sampling mit circa 120 Strahlen erzeugt

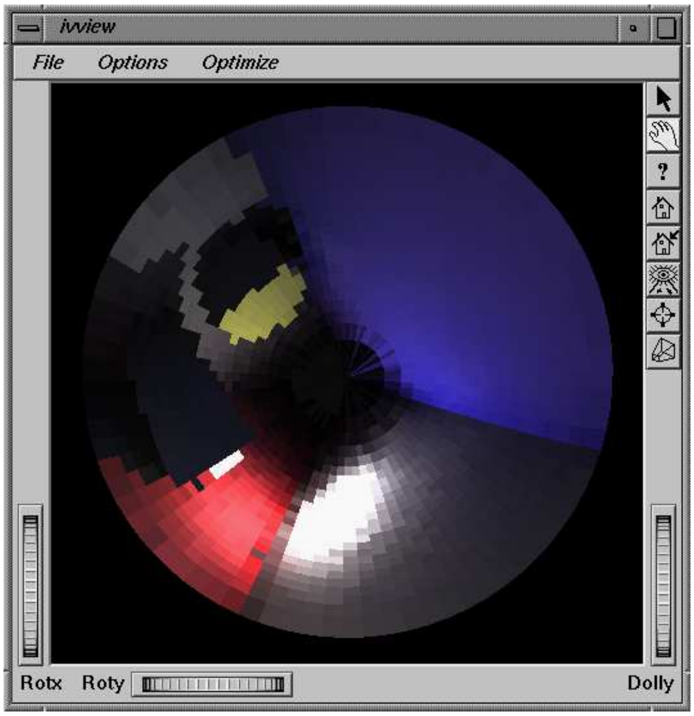

Abbildung B.14: Bild durch Halbkugel-Sampling mit circa 2000 Strahlen erzeugt

Abbildung B.15: Szene mit Radiance (direkte und diffuse Beleuchtung) berechnet.

Abbildung B.16: Ausschnitt nur mit diffuser Beleuchtung von Radiance berechnet, es kommt zu Fehlern durch numerische Probleme des Raytracers.

Abbildung B.17: Szene von Vision mit direkter und diffuser Beleuchtung berechnet.

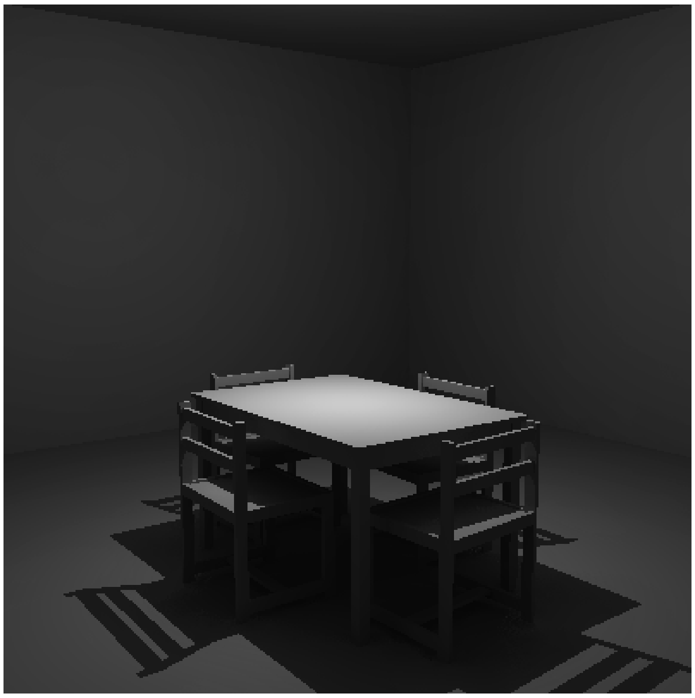

Abbildung B.18: Szene von Vision mit direkter und diffuser Beleuchtung berechnet.

Literaturverzeichnis

[1] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 279–288, August 1993.

[2] Michael F. Cohen and John R. Wallace. *Radiosity and Realistic Image Synthesis.* Academic Press, Cambridge, 1993.

[3] Robert L. Cook, Thomas Porter, and Loren Carpenter. Distributed ray tracing. In *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 137–45, July 1984. Monte Carlo distribution of rays to get gloss, translucency, penumbras, depth of field, motion blur.

[4] James T. Kajiya. The rendering equation. In David C. Evans and Russell J. Athay, editors, *Computer Graphics (SIGGRAPH '86 Proceedings)*, volume 20, pages 143–150, August 1986.

[5] John Kaufmann. *IES Lighting Handbook, Reference Volume.* IESNA, New York, 1981.

[6] Radiance. http://radsite.lbl.gov/radiance/home.html.

[7] P. Slusallek and H.-P. Siedel. Vision - an architecture for global illumination calculations. *IEEE Transactions on Visualization and Computer Graphics*, 1(1):77–96, March 1995. ISSN 1077-2626.

[8] Philipp Slusallek. Vision: An architecture for physically-based rendering. Technical report, Universiät Erlangen-Nürnberg, Erlangen, Germany, 1995.

[9] Gregory J. Ward and Paul Heckbert. Irradiance gradients. *Third Eurographics Workshop on Rendering*, pages 85–98, May 1992.

[10] Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear. A ray tracing solution for diffuse interreflection. In John Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 85–92, August 1988.

Erklärung

Ich versichere, daß ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und daß die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Ich bin damit einverstanden, daß die Arbeit veröffentlicht wird und daß in wissenschaftlichen Veröffentlichungen auf sie Bezug genommen wird.

Der Universität Erlangen-Nürnberg, vertreten durch den Lehrstuhl für Graphische Datenverarbeitung, wird ein (nicht ausschließliches) Nutzungsrecht an dieser Arbeit sowie an den im Zusammenhang mit ihr erstellten Programmen eingeräumt.

Erlangen, den 10. November 1997

(Matthias Ott)