

# Atlas-basierte globale Beleuchtung im Bildraum

## Studienarbeit

im Studiengang Computervisualistik

vorgelegt von  
Matthias Merz

Betreuer: Prof. Dr.-Ing. Stefan Müller, Dipl.-Inform. Niklas Henrich  
Institut für Computervisualistik, AG Computergraphik

Koblenz, im März 2010

## Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ja    Nein

Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.       

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.       

.....  
(Ort, Datum)

.....  
(Unterschrift)

Aufgabenstellung für die Studienarbeit  
Matthias Merz (Matr.-Nr. 205210144)

**Thema: Atlas-basierte globale Beleuchtung im Bildraum**

Es hat sich gezeigt, dass eine globale Beleuchtungssimulation im Bildraum möglich ist. Viele dieser Verfahren haben den Nachteil, dass nicht sichtbare Teile der Szene keinen Einfluss auf die Beleuchtungssituation haben. Dadurch können visuelle Artefakte entstehen, die vor allem bei einer Kamerabewegung störend auffallen.

Innerhalb dieser Studienarbeit soll untersucht werden, wie diese Artefakte durch den Einsatz eines Textur-Atlanten vermieden bzw. eingeschränkt werden können. Die Beleuchtungssimulation soll nicht nur anhand der im Bild sichtbaren Flächen durchgeführt werden, sondern es soll zusätzlich die Information im Textur-Atlas genutzt werden. Da in einem Textur-Atlas in der Regel die gesamte Szene abgespeichert ist, könnten die zuvor beschriebenen Artefakte vermieden werden.

Zunächst sollen die Möglichkeiten untersucht werden, die ein Atlas zur Speicherung von Szeneninformationen bereithält. Anschließend soll der Atlas in die bildbasierte Beleuchtung mit integriert werden. Anhand der daraus gewonnenen Erkenntnisse sollen Sampling-Strategien für den Atlas entwickelt werden, die eine Kamerabewegung ohne störende Artefakte ermöglicht. Das Thema Sichtbarkeit zwischen Sender- und Empfängerfläche beim Austausch des direkten sowie indirekten Lichtes soll zunächst vernachlässigt werden.

Schwerpunkte dieser Arbeit sind:

- Einarbeitung in die GPU-Programmierung
- Erforschung der Möglichkeiten von Textur-Atlanten zur Speicherung von Szeneninformationen
- Minimierung von visuellen Artefakten bei Kamerabewegungen
- Bewertung des entwickelten Verfahrens
- Dokumentation der Ergebnisse

Koblenz, den 01. Oktober 2009

- Prof. Dr. Stefan Müller-

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
<b>2</b>	<b>Related Work</b>	<b>2</b>
<b>3</b>	<b>Grundlagen</b>	<b>4</b>
3.1	Idee . . . . .	4
3.2	Lokale Beleuchtungsmodelle . . . . .	5
3.3	Globale Beleuchtungsmodelle . . . . .	6
3.4	Globale Beleuchtung im Bildraum . . . . .	7
3.5	Textur-Atlas . . . . .	8
3.6	Schatten - Shadow Mapping . . . . .	11
3.7	Rendering Equation . . . . .	11
3.8	Bidirektionale Reflexionsverteilungsfunktion . . . . .	12
3.9	Hammersley-Sequenz . . . . .	14
3.10	Monte-Carlo-Integration . . . . .	14
<b>4</b>	<b>Implementierung</b>	<b>16</b>
4.1	Überblick . . . . .	16
4.2	Zusammengefasster Ablauf . . . . .	16
4.3	Geometrie und Textur-Atlas erstellen . . . . .	20
4.4	G-Buffer Textur-Atlas erzeugen . . . . .	20
4.5	Erzeugen einer Samplettextur mit der Hammersley-Sequenz	22
4.6	Berechnung des indirekten Lichts . . . . .	22
4.7	Darstellung der Szene mit der berechneten Beleuchtung . . .	26
<b>5</b>	<b>Ergebnisse und Auswertung</b>	<b>27</b>
5.1	Qualität . . . . .	27
5.2	Test einer komplexeren Szene . . . . .	28
5.3	Performance . . . . .	29
5.4	Erweiterungsmöglichkeiten . . . . .	29
5.5	Fazit . . . . .	30

# 1 Einführung

Mit globalen Beleuchtungssimulationen wird versucht, die physikalischen Eigenschaften von Licht und dessen Ausbreitung möglichst korrekt zu berechnen. Dabei werden diese üblicherweise im dreidimensionalen Objektraum berechnet und sind deshalb sehr rechenintensiv und von der Anzahl der Polygone der Szene abhängig.

Objektraum-basierte Verfahren ermöglichen durch komplexe Berechnungen allerdings auch gute Annäherungen physikalisch korrekter Beleuchtungen. Die Beleuchtungsberechnung in den Bildraum zu verlagern hat aber den großen Vorteil, dass die Berechnung hier unabhängig von der Größe und Komplexität der Szene durchführbar ist. Einzig die Auflösung entscheidet über den Rechenaufwand. Dieser Geschwindigkeitsvorteil beinhaltet jedoch einen Kompromiss, was die physikalische Korrektheit der Beleuchtungssimulation angeht. Bei bisherigen Bildraum-Verfahren war es nicht möglich, für die Kamera nicht sichtbare Teile der Szene und deren Beleuchtungsinformationen, in die Berechnung mit einzubeziehen. Dies ist der Fall, da über Geometrie die aus Sicht der Kamera nicht sichtbar ist, im Bildraum zu nächst keinerlei Informationen zur Verfügung stehen.

Ein bekanntes Beispiel zur Annäherung einer globalen Beleuchtung im Bildraum ist das Verfahren Screen Space Ambient Occlusion. Dieses liefert zwar sehr gut Selbstverschattungen und wirkt dadurch realitätsnah, erzeugt aber keinen korrekten Lichtaustausch. Da die Beleuchtung von dem Inhalt des aktuellen Bildausschnitts abhängig ist, entstehen visuelle Artefakte, die vor allem bei Kamerabewegungen störend auffallen.

Ziel der Studienarbeit ist es daher diese Artefakte durch die Verwendungen von Textur-Atlanten zu vermeiden. Dies wird durch eine Bildraum-basierte Beleuchtungssimulation ermöglicht, die die gesamte Szene zur Berechnung der Beleuchtung miteinbezieht und nicht nur die Sicht der Kamera. Dabei wird in einem Textur-Atlas die gesamte Szene gespeichert.

## 2 Related Work

Es gibt bereits einige Bildraum-basierte Renderingverfahren. Eines der prominentesten Beispiele ist das Scene Space Ambient Occlusion Verfahren, dass von Crytek entwickelt wurde [1]. Dieses Verfahren liefert für jedes Pixel den Verdeckungsgrad und damit, wie viel Licht jedes Pixel erreicht, wie in Abbildung 1 zu sehen ist. Als zusätzliche Information, wird lediglich der Tiefenpuffer der Szene benötigt. Da es sich um ein Bildraum Verfahren handelt, ist der Berechnungsaufwand unabhängig von der Komplexität der Szene. Die Anzahl der Polygone, die für die Modellierung von Objekten verwendet wurden, hat somit keine Auswirkung auf die Dauer der Berechnung. Einzig die Auflösung des Bildes entscheidet über die Berechnungsdauer.

Alle Ambient Occlusion Verfahren können als Annäherung einer globalen Beleuchtung angesehen werden. Sie erhöhen zwar den Realitätsgrad des erzeugten Bildes enorm, können aber nicht alle Effekte einer globalen Beleuchtung liefern. Für interaktive Echtzeitanwendungen, wie zum Beispiel Computerspiele, liefern sie jedoch eine passende Mischung aus glaubhafter Darstellung und Geschwindigkeit. Für eine möglichst korrekte Simulation von Licht eignet sich ein Ambient Occlusion Verfahren grundsätzlich nicht, da zur Berechnung des Umgebungslichts, ein konstantes, diffuses Umgebungslicht angenommen wird. Da dies in den meisten Fällen nicht zutrifft, lässt sich hiermit keine korrekte Simulation erzielen.

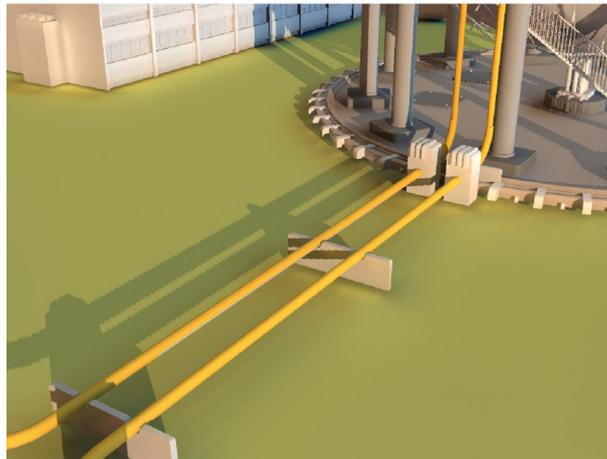
Bildraum Verfahren können auch in Kombination mit anderen Räumen benutzt werden, wie unter anderem Reinbothe et al. mit ihrer Hybrid Ambient Occlusion Methode gezeigt haben [2]. Dabei verwenden sie zusätzlich zum Bildraum auch den voxelbasierten Objektraum.



**Abbildung 1:** SSAO, Bild aus [1]

Eine Weiterentwicklung von Screen Space Ambient Occlusion, die weitere Effekte einer globalen Beleuchtung abdeckt, stellt die Arbeit von Rit-

schel et al. dar [3]. Ihr Screen Space Directional Occlusion nähert die direkte Beleuchtung und indirekte Beleuchtung im Bildraum an. Dadurch werden farbige Schatten und Color Bleeding ermöglicht. Dabei wird der Farbaustausch durch Licht zwischen benachbarten Flächen simuliert, wie in Abbildung 5 zu sehen ist. So ist zu sehen, dass die farbigen Blöcke den Untergrund mit ihrer Farbe anstrahlen.



**Abbildung 2:** Screen Space Directional Occlusion, Bild aus [3]

Neben den Ambient Occlusion Methoden gibt es noch weitere Versuche, globale Beleuchtung im Bildraum zu berechnen. McGuire et al. haben mit Image Space Photon Mapping eine echtzeitfähige Photon Mapping Variante entwickelt, die die kostenintensivsten Teile des Algorithmus auf den Bildraum verlagert [4].

Ein weiteres Verfahren zur Berechnung des indirekten Lichts sind Reflective Shadow Maps [5]. Dabei wird angenommen, dass jeder Pixel der Shadow Map eine indirekte Lichtquelle ist. Das indirekte Licht wird mittels adaptivem Sampling im Fragment Shader berechnet und im Bildraum interpoliert um interaktive Bildwiederholraten zu erzeugen. Das Ergebnis ist auch hier eine Annäherung der indirekten Beleuchtung, die aber plausible Bilder liefert.

## 3 Grundlagen

In diesem Abschnitt wird zunächst die Idee der Ausarbeitung näher erläutert. Anschließend werden die, zur Umsetzung dieser Idee benötigten Grundlagen erklärt.

Zur Umsetzung der Atlas-basierten globalen Beleuchtung im Bildraum wird zuerst ein grundlegendes lokales Beleuchtungsmodell benötigt. Mit Hilfe der so berechneten direkten Beleuchtung kann dann eine globale Beleuchtung ermittelt werden. Dazu werden die Grundlagen der globalen Beleuchtung im Allgemeinen und im Bildraum vorgestellt. Auch das Prinzip eines Textur-Atlas und die daraus entstehenden Möglichkeiten werden vorgestellt. Desweiteren wird Shadow-Mapping zur Berechnung von Schatten in einer Szene kurz erläutert.

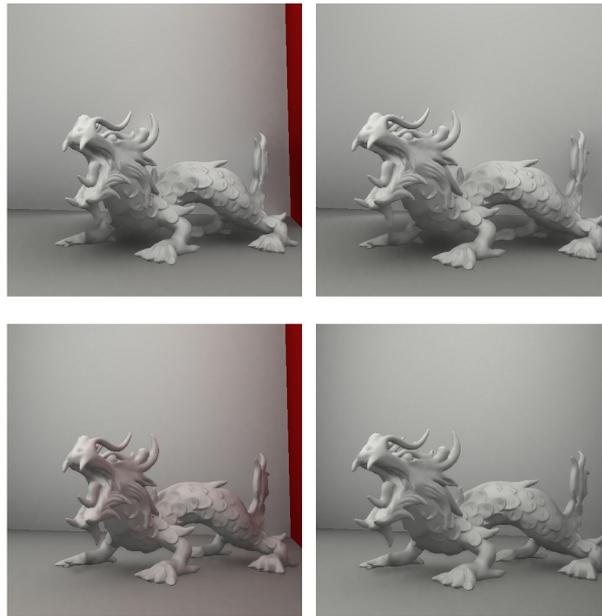
Die Rendering Equation ist in der Computergraphik die grundlegende Formel zur Simulation von Licht. Auch die Berechnung des hier vorgestellten Verfahrens lässt sich aus der Rendering Equation ableiten. Ein wichtiger Teil dieser Gleichung ist die bidirektionale Reflexionsverteilungsfunktion, die das Verhalten von Materialien bei der Interaktion mit Licht beschreibt. Die Monte-Carlo-Integration zeigt wie es möglich ist, Integrale mit Hilfe von Zufallszahlen anzunähern.

### 3.1 Idee

Eine wesentliche Einschränkung fast aller Bildraum-basierten Verfahren ist, dass die Berechnungen nur auf Grundlage der Informationen, die im aktuellen Bildausschnitt zu sehen sind, durchgeführt werden können. Geometrie außerhalb dieses Bereichs kann bei solchen Screen Space Methoden nicht berücksichtigt werden. Das ist besonders für die Berechnung der globalen Beleuchtung problematisch, da so bei Kamerabewegungen störende visuelle Artefakte entstehen können. Sie entstehen durch das plötzliche Auftauchen von Geometrie, die zunächst außerhalb des sichtbaren Bereichs liegen. Wenn die Kamera dann eine Bewegung ausführt und neue Teile der Szene ins Bild hinzukommen oder andere verschwinden, können Beleuchtungseffekte, die durch diese Teile verursacht werden, ebenso auftauchen oder verschwinden. Dadurch wird der Realitätsgrad der Bilder deutlich verringert und die visuellen Artefakte können störend auffallen.

Ein Beispiel für einen solchen Effekt lässt sich anhand einer Cornell-Box erklären. Diese besteht aus fünf Flächen, wobei der Boden, die Decke und die Rückwand weiße Flächen sind. Die linke Wand der Box ist rot und die rechte grün. Eine Lichtquelle ist in der Mitte der Box platziert. Die Box wird von vorne betrachtet und die linke, rote Wand ist sichtbar. Bei einer Berechnung der indirekten Beleuchtung im Bildraum ist nun der bereits beschriebene Effekt des Color Bleeding zu sehen und die rote Wand würde die drei benachbarten weißen Wände leicht anstrahlen. Diese wären vor al-

lem an den zur roten Fläche angrenzenden Rändern leicht rot. Dasselbe gilt auch für die grüne Wand. Wenn die Kamera nun weiterbewegt würde und die rote Wand aus dem sichtbaren Bereich verschwindet, würde auch das Color Bleeding schlagartig verschwinden, obwohl sich die Beleuchtungssituation nicht verändert hat, wie in Abbildung 3 zu sehen ist.



**Abbildung 3:** Besonders im unteren Bild fällt auf, dass die rote Beleuchtung verloren geht. Bild aus [6]

Diese visuellen Artefakte könnten durch den Einsatz eines Textur-Atlas verringert oder gar vermieden werden. Ein Textur-Atlas bietet die Möglichkeit, die gesamte Szene in einer Textur zu speichern. Dabei werden allerdings nicht nur alle Informationen aus einer totalen Kamerasicht gespeichert, sondern die komplette Geometrie der Szene. Anstatt auf einem Bild aus Kamerasicht, können die Berechnungen nun auf dem Textur-Atlas vorgenommen werden. Somit ist es möglich, auch Teile der Szene, die nicht im letztendlichen Ausgabebild enthalten sind, bei der Berechnung der Beleuchtung mit einfließen zu lassen.

### 3.2 Lokale Beleuchtungsmodelle

Bevor die indirekte Beleuchtung berechnet werden kann, benötigt man als Grundlage zunächst die direkte Beleuchtung. Als direkte Beleuchtung wird das Licht, das direkt von einer Lichtquelle auf Objekte ausgestrahlt wird, bezeichnet.

Lokale Beleuchtungsmodelle simulieren somit das Verhalten von Licht wenn es auf Oberflächen trifft. Es wird die Helligkeit und Farbe berech-

net, die bei der Interaktion des Materials mit Licht entsteht. Dabei sind die Eigenschaften des Materials entscheidend, welcher Anteil des Lichts reflektiert, absorbiert oder transmittiert werden. Für die Berechnung benötigt man die Blickrichtung, den Lichteinfallswinkel und die Materialeigenschaften, sowie die Lichteigenschaften. Die Blickrichtung spielt allerdings für vollkommen diffuse Materialien keine Rolle, da diese auf jeder Richtung gleich hell erscheinen. Bei der direkten Beleuchtung werden ausschließlich definierte Lichtquellen berücksichtigt. Dabei werden Lichtreflektionen von angestrahlten Objekten nicht berechnet, wodurch die Beleuchtung weitaus weniger realistisch wirkt. Die Lichteigenschaften, wie Farbe und Position, werden dabei gesetzt.

Eigenschaften von Materialien können mit der bidirektionalen Reflexionsverteilungsfunktion (Bidirectional Reflectance Distribution Function, kurz BRDF) modelliert werden. Diese Funktion beschreibt das Verhalten von Materialien bei der Interaktion mit Licht. Es wird für jeden Einfallswinkel beziehungsweise Ausfallswinkel angegeben, wie viel der eingehenden Beleuchtungsstärke als Leuchtdichte reflektiert wird. Die BRDF kann dabei in diskreter Form, als eine Vielzahl an Messwerten vorliegen, oder aber als mathematische Funktion.

Eine vereinfachte Form der BRDF kann für ideal diffuse Materialien angenommen werden. Dabei ist die BRDF konstant. Sie beschreibt somit den Reflexionsgrad des Materials unter der Annahme, dass für jede Lichteinfallrichtung die gleiche Leuchtdichte in alle Richtungen reflektiert wird. Dabei wird das Material als diffuser Lamberstrahler angenommen.

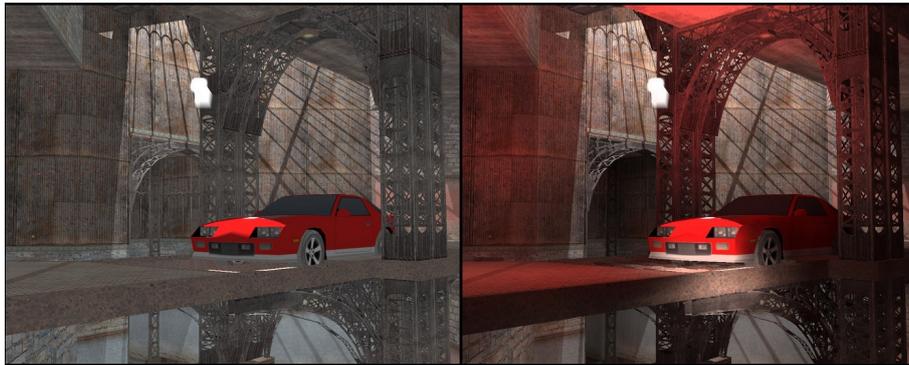
### 3.3 Globale Beleuchtungsmodelle

Um realitätsnahe Ergebnisse erzielen zu können, benötigt man weitere Berechnungen auf Basis der direkten Beleuchtung. Die direkte Beleuchtung berechnet lediglich den Weg des Lichts bis zum ersten Auftreffen. Die Auswirkungen des reflektierten Lichts auf andere Teile der Szene wurden dabei bisher nicht simuliert. Dieses reflektierte, oder auch indirekte Licht, welches eben nicht direkt von der Lichtquelle aus auf ein Objekt trifft, trägt aber wesentlich zur Gesamtbeleuchtung bei.

Bei der globalen Beleuchtung werden alle Möglichkeiten der Lichtausbreitung abgedeckt, nicht nur die direkte Ausbreitung. Vor allem Teile einer Szene, die wenig oder gar kein direktes Licht empfangen, werden ohne Berücksichtigung des indirekten Lichts zu dunkel dargestellt. In der Realität werden diese dunkleren Bereiche durch reflektiertes Licht aus ihrer Umgebung beleuchtet. Der Lichtaustausch zwischen einzelnen Objekten hat dabei nicht nur Einfluss auf deren Helligkeit, sondern auch auf die Farbe.

Wie bereits am Beispiel der Cornell-Box gezeigt wurde, strahlt eine farbige Fläche, die direkt beleuchtet wird, farbiges Licht auf andere Flächen in der Umgebung und lässt diese somit auch leicht farbig erscheinen. Um

übermäßig dunkle Stellen zu vermeiden, kann auf die gesamte Szene ein zusätzlicher Helligkeitswert addiert werden. Dieser sogenannte ambiante Term hat allerdings den Nachteil, dass die Szene damit nicht besonders realistisch wirkt, wie in Abbildung 4 zu sehen ist.



**Abbildung 4:** links ambianter Term, rechts Image Space Photon Mapping ,Bild aus [4]

Eine Referenzlösung für globale Beleuchtung liefern dagegen Path Tracing Algorithmen [7]. Bei diesem Verfahren wird versucht die Rendering Equation näherungsweise zu lösen. Diese Integralgleichung beschreibt die Energieerhaltung bei der Ausbreitung von Licht. Dabei werden im Vergleich zum herkömmlichen Ray-Tracing, viele Strahlen pro Pixel statt nur einem Strahl betrachtet. Jeder dieser Strahlen wird nach dem Auftreffen auf Geometrie in eine zufällige Richtung weiterverfolgt. Bei einer sehr hohen Anzahl von Strahlen nähert man sich einem idealen Bild an. Der Rechenaufwand steigt allerdings mit der Anzahl der Strahlen und ist dadurch sehr hoch. Daher eignet sich Path Tracing nicht für Echtzeitleösungen.

### 3.4 Globale Beleuchtung im Bildraum

Eine genaue Berechnung der globalen Beleuchtung im Bildraum ist ohne die Verwendung zusätzlicher Informationen nicht möglich, denn dazu benötigt man die gesamte Szene als Berechnungsgrundlage. Bei vielen Bildraum-basierten Methoden stehen jedoch zunächst nur Informationen über den aktuellen Kameraausschnitt bereit. Diese werden als Screen Space Verfahren bezeichnet. Auf einem solchen Ausschnitt der Szene ist es lediglich möglich, eine Annäherung der globalen Beleuchtung zu berechnen.

Bei Screen Space Ambient Occlusion Verfahren wird beispielsweise die Umgebungsverdeckung bei einem angenommenen konstanten Umgebungslicht berechnet. Dieses Umgebungslicht ist jedoch unabhängig von der tatsächlichen Beleuchtungssituation der Szene und somit liefert das Verfahren nur eine grobe Annäherung.

Das Screen Space Directional Occlusion (SSDO) Verfahren von Ritschel et al. deckt bereits größere Teile der globalen Beleuchtung ab [3]. So werden hierbei auch Effekte wie gerichtete Schatten und indirekter Lichtaustausch berechnet, wodurch die Ergebnisse an Realismus gewinnen. Allerdings gilt auch für SSDO die Einschränkung, dass die Berechnung ausschließlich auf der aktuellen Kamerasicht durchgeführt werden können und somit nicht immer alle Teile der Szene in die Berechnung mit einfließen können. Der dabei auftretende Effekt ist sehr deutlich in Abbildung 5 zu erkennen.



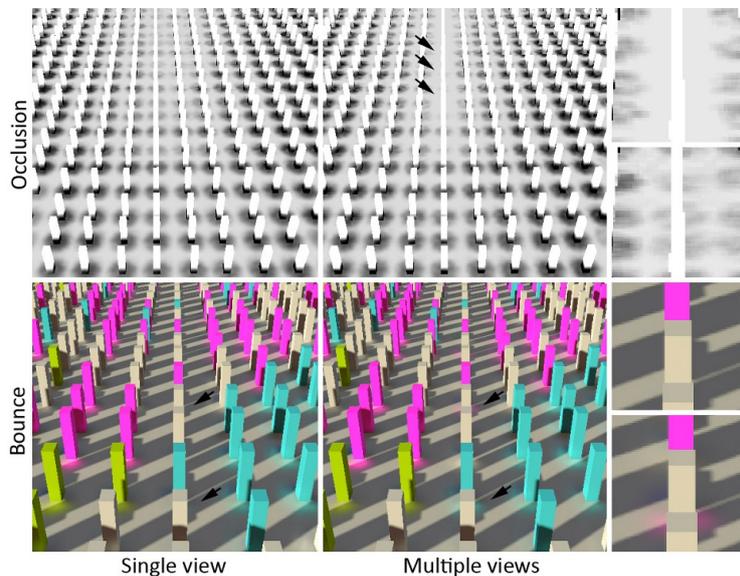
**Abbildung 5:** Je weniger von der Fläche der gelben Box zu sehen ist, desto weniger ist auch der indirekte Beleuchtungseffekt auf der Bodenfläche zu sehen, Bild aus [3]

Je weniger der gelben Fläche im Bild zu sehen ist, desto schwächer wird auch der mit einem Pfeil markierte Beleuchtungseffekt. Dieser unerwünschte Effekt kann durch die Verwendung zusätzlicher Kameras verhindert werden. Diese Methode benötigt jedoch relativ aufwendige Berechnungen und kann auf Grund von Verdeckungen nicht immer zuverlässige Ergebnisse liefern [3]. Objekte, die aus Sicht des Betrachters zu sehen sind, können aus einer weiteren Kameraperspektive verdeckt sein und somit können keine weiteren Informationen dazugewonnen werden. Wie in Abbildung 6 zu sehen ist, lassen sich damit jedoch für die meisten Szenarien gute Ergebnisse erzielen.

Als alternative Möglichkeit zu zusätzlichen Kameras, stellen Ritschel et al. noch die Verwendung von Depth Peeling [8] vor. Im Vergleich zur normalen Renderingmethode werden dabei  $n$  Tiefenwerte pro Pixel durch  $n$  Rendering-Durchläufe, statt nur einem Tiefenwert pro Pixel erzeugt. Mit diesen zusätzlich gewonnenen Informationen lässt sich nun für jede Schicht die direkte Beleuchtung berechnen und Effekte wie Color Bleeding werden auch für abgewandte und verdeckte Flächen beziehungsweise Objekte korrekt berechnet. Allerdings werden Beleuchtungseffekte von Flächen außerhalb des aktuellen Bildes nicht berücksichtigt wie zum Beispiel in Abbildung 3 zu sehen ist.

### 3.5 Textur-Atlas

Ein Textur-Atlas ist ein großes Bild, das viele kleinere Sub-Bilder enthält. Jedes der Sub-Bilder ist dabei die Textur eines Objekts. Vor allem wenn für



**Abbildung 6:** Oben ist die Verdeckung zu sehen und unten die Beleuchtung, dabei fällt auf, dass durch zusätzliche Kameras einige Effekte verdeckter Teile berechnet werden können, Bild aus [3]

eine Szene eine Vielzahl an kleinen Texturen benötigt wird, lohnt es sich einen Textur-Atlas zu verwenden. Durch die Zusammenfassung von vielen Texturen zu einer großen Textur, ist es möglich ohne immer wieder eine neue Textur binden zu müssen, auf viele verschiedene Texturen Zugriff zu erhalten. So kann die Performance einer Szene, die viele kleine Texturen benötigt, deutlich gesteigert werden. Die einzigen Limitierungen, die dabei zu beachten sind, sind die maximale unterstützte Texturauflösung der verwendeten Grafikhardware und dass ein Textur-Atlas auf Auflösungen von Zweierpotenzen beschränkt ist.

Ein interessanter Aspekt an Textur-Atlanten ist, dass durch ihre Verwendung alle Oberflächenteile eines dreidimensionalen Objektes in einer zweidimensionalen Ebene ausgebreitet werden können.

Wie in Abbildung 8 zu sehen ist, sind alle Oberflächen des dreidimensionalen Objekts in einer zweidimensionalen Textur ausgebreitet. Auch Flächen, die dem Betrachter der dreidimensionalen Version abgewandt sind, werden im Textur-Atlas sichtbar. Außerdem ist zu sehen, dass nicht jedes Teilobjekt etwa einen eigenen Textur-Atlas besitzt. Alle Objekte, die zu der Szene der Abbildung gehören, sind auf einem Textur-Atlas enthalten. Auch Flächen, wie zum Beispiel die Rückseite des Quaders, werden im Atlas repräsentiert.

Ein Textur-Atlas eignet sich somit auch zur Repräsentation einer ganzen Szene. Mit Hilfe dieser Informationen kann dann die indirekte Beleuch-



Abbildung 7: Beispiel für die Verwendung eines Textur-Atlas, Bild aus [9]

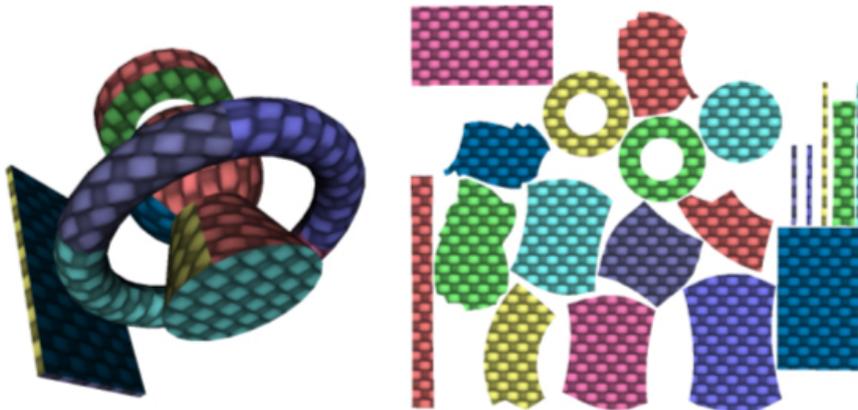


Abbildung 8: Links Objekt in 3d-Sicht, rechts als Textur-Atlas, Bild aus [10]



Abbildung 9: Wirkung von Schatten, Bild aus [11]

tung für die gesamte Szene berechnet werden.

### 3.6 Schatten - Shadow Mapping

Schatten spielen bei der menschlichen Wahrnehmung eine sehr wichtige Rolle. Durch Schatten wird die Erkennung der Position und Größe eines Gegenstands erleichtert beziehungsweise auf Bildern überhaupt erst ermöglicht, wie in Abbildung 9 zu sehen ist. Shadow Mapping ist eine einfache Methode um mit geringem zusätzlichem Aufwand Schatten zu erzeugen. Die wesentliche Annahme beim Shadow Mapping ist, dass Punkte der Szene, die direkt von der Lichtquelle angestrahlt werden, für eine Kamera mit derselben Position und Blickrichtung wie die Lichtquelle, sichtbar sind. Falls ein Punkt im Schatten liegt, ist er für die Kamera aus dieser Perspektive nicht sichtbar, da er von einem anderen Punkt verdeckt wird. Die Szene wird zunächst aus Sicht der Lichtquelle gerendert. Dabei werden die Tiefeninformationen der einzelnen Pixel in eine Textur, die sogenannte Shadow Map gespeichert. In dieser Textur sind nun die Tiefenwerte aus Sicht der Lichtquelle gespeichert, der Abstand der Szene wird für jeden einzelnen Pixel der Shadow Map abgetastet.

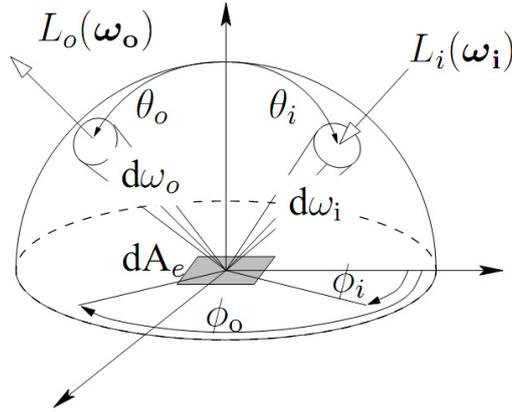
Danach wird die Szene aus der normalen Kamerasicht erneut gerendert. Die Werte aus der Shadow Map werden im nächsten Schritt mit den Entfernungen, der für die Kamera sichtbaren Punkte zur Lichtquelle, verglichen. Durch diesen Vergleich wird bestimmt, ob ein Punkt im Schatten liegt oder nicht.

### 3.7 Rendering Equation

In der Computergraphik basiert das Bildentstehungsmodell auf der Simulation von Licht. Man versucht die Lichtausbreitung mit strahlungsphysikalischen Gleichungen möglichst realistisch zu beschreiben. Die grundlegende und wichtigste Gleichung ist dabei die von Kajiya veröffentlichte Integralgleichung, die Rendering-Equation [7].

Die Rendering-Equation:

$$L_o(dA_e, d\omega_o) = L_e(dA_e, d\vec{\omega}_o) + \int_{2\pi} f_r(dA_e, d\vec{\omega}_o, d\vec{\omega}_i) \cdot L_i(dA_e, d\vec{\omega}_i) \cdot \cos \theta_i \cdot d\omega_i \quad (1)$$



**Abbildung 10:** Die verwendeten Basisgrößen, Bild aus [12]

Vereinfacht gesagt liefert die Rendering-Equation das Licht  $L_o$ , das von einem beliebigen Oberflächenelement  $dA_e$  in eine beliebige Richtung  $\omega_o$  ausgestrahlt wird. Dieses Licht setzt sich unter anderem aus dem Licht, das von dem Oberflächenelement in Richtung  $\omega_o$  emittiert wird, zusammen. Hinzu kommt das Licht  $L_i$ , das aus allen Einfallsrichtungen  $\omega_i$  einfällt und in Richtung  $\omega_o$  reflektiert wird.

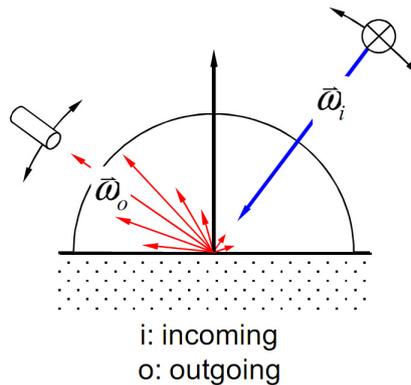
Genauer betrachtet, beschreibt  $L_o$  die Leuchtdichte, die von einem infinitesimalen Flächenelement  $dA_e$  in eine beliebige Richtung  $d\omega_o$  ausgestrahlt wird. Diese Leuchtdichte ergibt sich aus der Leuchtdichte  $L_e$ , die das Flächenelement in diese Richtung emittiert. Hinzu kommen alle Leuchtdichten  $L_i$ , die von allen Raumwinkeln  $d\omega_i$  des vorderen Halbraums einstrahlt und in die Richtung  $d\omega_o$  reflektiert werden. Um transparente Materialien darzustellen, müsste man analog dazu den hinteren Halbraum betrachten. Der Term  $f_r$  ist die sechsdimensionale bidirektionale Reflexionsverteilungsfunktion. Diese Funktion beschreibt für jedes Raumwinkelpaar den Anteil der ausfallenden Leuchtdichte  $dL_o$  zur einfallenden Beleuchtungsstärke  $dE_i$  und bildet den Kern der Rendering Equation.

### 3.8 Bidirektionale Reflexionsverteilungsfunktion

Die bidirektionale Reflexionsverteilungsfunktion:

$$f_r(dA_e, d\omega_o, d\omega_i) = \frac{dL_o(dA_e, d\omega_o)}{dE_i(dA_e, d\omega_i)} \quad (2)$$

Die BRDF beschreibt das Reflexionsverhalten von Materialien. Um dieses zu messen, geht man wie in Abbildung 11 zu sehen ist, vor. Für das von rechts kommende, einfallende Licht wird die Beleuchtungsstärke gemessen, da eine andere Größe nur sehr schwer messbar wäre. Für das ausfallende, reflektierte Licht misst man die Leuchtdichte.



**Abbildung 11:** Einfache Darstellung der Messung, Bild aus [13]

Eine Messung des gesamten Halbraums in  $5^\circ$  Schritten würde dabei ca.  $1300 \times 1300$  Werte erzeugen. Das Resultat wäre eine Annäherung der BRDF. In dieser Form ist die BRDF eine sechsdimensionale Funktion, wenn die Wellenlänge noch hinzu kommt eine siebendimensionale. Die BRDF kann Werte aus dem Bereich  $(0, \infty)$  annehmen und hat dabei die Einheit  $1/\text{sr}$ . Damit eine BRDF physikalisch plausibel ist, muss sie folgende Kriterien erfüllen [13]:

- Sie darf nicht negativ sein.
- Sie muss den Energieerhaltungssatz erfüllen.
- Sie muss die Helmholtz Reziprozität erfüllen.

Der Energieerhaltungssatz ist erfüllt, wenn kein zusätzliches Licht entsteht. Da Licht nur reflektiert wird und nicht erzeugt, muss diese Bedingung erfüllt werden. Die Helmholtz Reziprozität besagt, dass der Lichtstrom in eine Richtung für Licht in die umgekehrte Richtung identisch sein muss. Sie ergibt sich aus den Maxwell'schen Gleichungen. Damit der Energieerhaltungssatz erfüllt ist, muss der Reflexionsgrad  $\rho$  im Bereich  $(0, 1)$  liegen. Ein Wert von genau eins entspricht dabei einer perfekten Reflexion, vollkommen ohne Absorption. Solche Materialien kommen allerdings in der Natur nicht vor. Der Reflexionsgrad ist als der gesamte ausfallende Lichtstrom im Verhältnis zum gesamten eingestrahnten Lichtstrom definiert.

Formel Reflexionsgrad:

$$\rho = \frac{\Phi_o}{\Phi_i} = \frac{\int_{2\pi} L_o(d\omega_o) \cdot \cos \theta_o \cdot d\omega_o}{\int_{2\pi} L_i(d\omega_i) \cdot \cos \theta_i \cdot d\omega_i} \quad (3)$$

Da sowohl die bidirektionale Reflexionsverteilungsfunktion, als auch die gesamte „Rendering-Equation“ aufgrund ihrer Komplexität kaum zu lösen sind, werden in der Computergraphik Modelle zur Annäherung benutzt.

### 3.9 Hammersley-Sequenz

Um Samples auf einer Fläche zu erzeugen werden üblicherweise Zufallszahlen verwendet. Dabei sind diese idealerweise so zwischen den Werten Null und Eins verteilt, dass jeder Wert mit gleicher Wahrscheinlichkeit angenommen wird. Für diesen Zweck eignen sich die sogenannten Low Discrepancy Sequenzen wie die Hammersley-Sequenz und die Halton-Sequenz. Diese Sequenzen weisen eine sehr geringe Diskrepanz auf, was bedeutet, dass die erzeugten Zahlen sehr gleichmäßig verteilt sind [14].

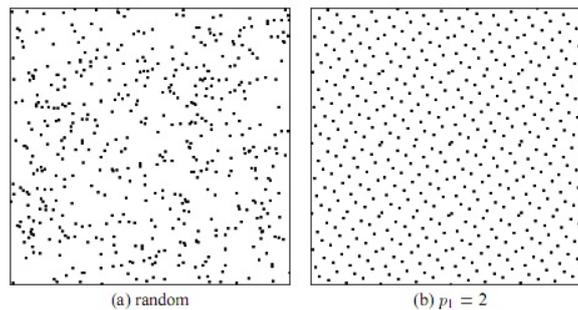


Abbildung 12: Links mit Randomfunktion verteilte Punkt, rechts mit Hammersley, Bild aus [14]

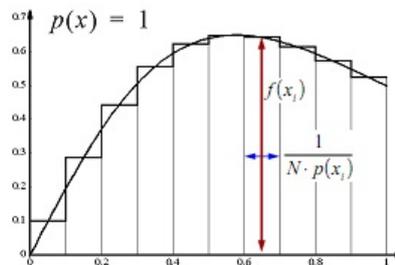
### 3.10 Monte-Carlo-Integration

Als Monte-Carlo-Integration wird die Bestimmung von Integralen durch Zufallszahlen bezeichnet. Der Wert des Integrals wird durch den Erwartungswert einer Zufallsvariable dargestellt und durch Stichproben angenähert.

$$E(g(x)) = \int_a^b g(x)p(x)dx \approx \frac{1}{N} \sum_{i=1}^N g(x_i) \quad (4)$$

Gesucht ist das Integral die Fläche unter der Kurve. Durch eine Unterteilung dieser Fläche in Rechtecke, wie in Abbildung 13 zu sehen ist,

$$\int_a^b f(x) dx \approx \frac{1}{N} \sum_{i=1}^N \frac{f(x_i)}{p(x_i)} \quad \blacksquare$$



**Abbildung 13:** Monte-Carlo Integration, Bild aus [15]

kann diese Fläche durch Rechtecke angenähert werden. Je geringer dabei die Breite der einzelnen Rechtecke gewählt, desto genauer passt sich der Umriss, der durch die Rechtecke entstanden ist, der tatsächlichen Form der Kurve an. Damit erhöht sich die Genauigkeit der Annäherung. Die Breite der Rechtecke wird dabei durch die Anzahl der Samples bestimmt. So lässt sich nachvollziehen, dass durch eine höhere Anzahl an Samples, ein genaueres Ergebnis erzielt werden kann [15].

## 4 Implementierung

Im folgenden Teil wird die Umsetzung der Atlas-basierten globalen Beleuchtung beschrieben. Dazu wird zunächst ein Überblick über die einzelnen Teile der Implementierung gegeben. Dann wird der Ablauf zunächst grob zusammengefasst. Danach werden die einzelnen Teile genauer erläutert. Die Abbildung „Programmablauf“ bildet die Funktionsweise des Programms zusammengefasst ab.

### 4.1 Überblick

In dieser Studienarbeit wurde ein GPU-basiertes Verfahren zur Berechnung der globalen Beleuchtung im Bildraum mit Hilfe eines Textur-Atlas erarbeitet und realisiert. Die Anwendung besteht aus folgenden Teilen:

- Geometrie erzeugen und durch setzen der Texturkoordinaten auf einen Textur-Atlas abbilden
- Shadow Map berechnen
- G-Buffer Textur-Atlas erzeugen: direktes Licht, Material, Normalen, Welt-Koordinaten
- Erzeugen einer Samplettextur mit der Hammerley-Sequenz
- Berechnung des indirekten Lichts
- Kombination des direkten und indirekten Lichts
- Textur-Atlas mit kombinierter Beleuchtung auf Szenen-Geometrie abbilden

Die Anwendung wurde mit der Entwicklungsumgebung Visual Studio 2008 in C++, OpenGL (Open Graphics Library), sowie GLSL (OpenGL Shading Language) umgesetzt. Mit Hilfe des Wavefront-OBJ-Loaders GLM können 3D-Modelle geladen werden.

### 4.2 Zusammengefasster Ablauf

Zunächst wird die Geometrie der Szene erzeugt und eventuell ein 3D-Modell geladen. Dabei muss beachtet werden, dass die Textur-Koordinaten speziell angepasst sein müssen. Während üblicherweise die Textur-Koordinaten lediglich dazu verwendet werden, einer Fläche der Szene eine bestimmte Textur zuzuordnen, müssen die Textur-Koordinaten nun so gesetzt werden, dass sie die gesamte Geometrie im Textur-Atlas repräsentieren. Am einfachsten lässt sich dies realisieren, indem die gesamte Szene als 3D-Modell geladen würde. Bereits bei der Modellierung werden dabei für

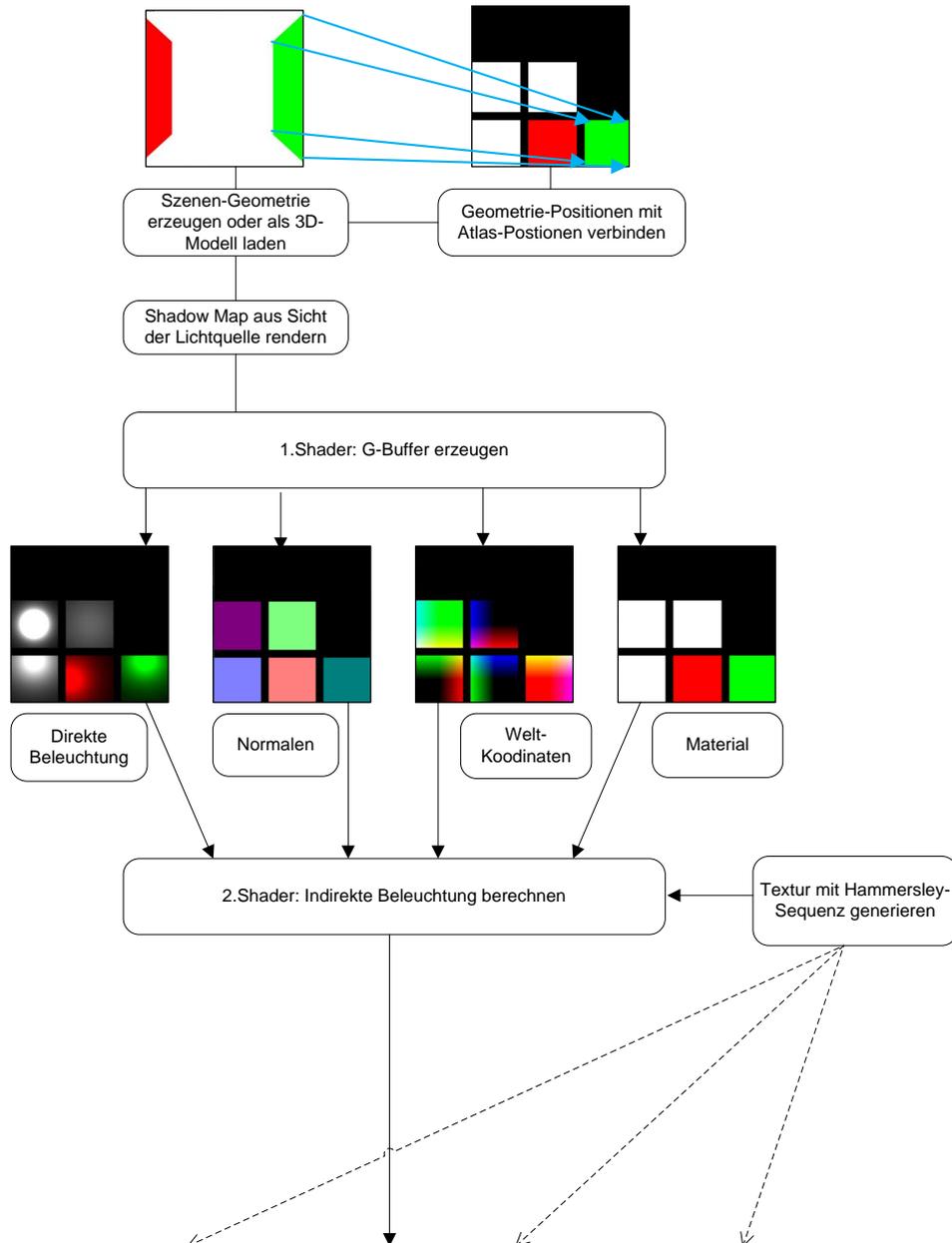
alle Teile der Szene Texturkoordinaten festgelegt und somit die Positionen der einzelnen Teile im Atlas bestimmt.

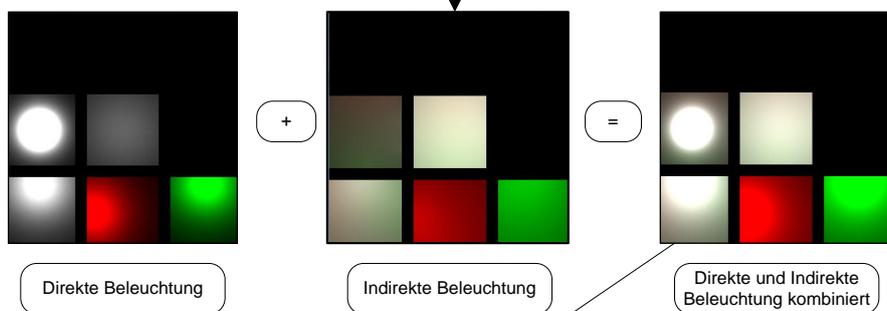
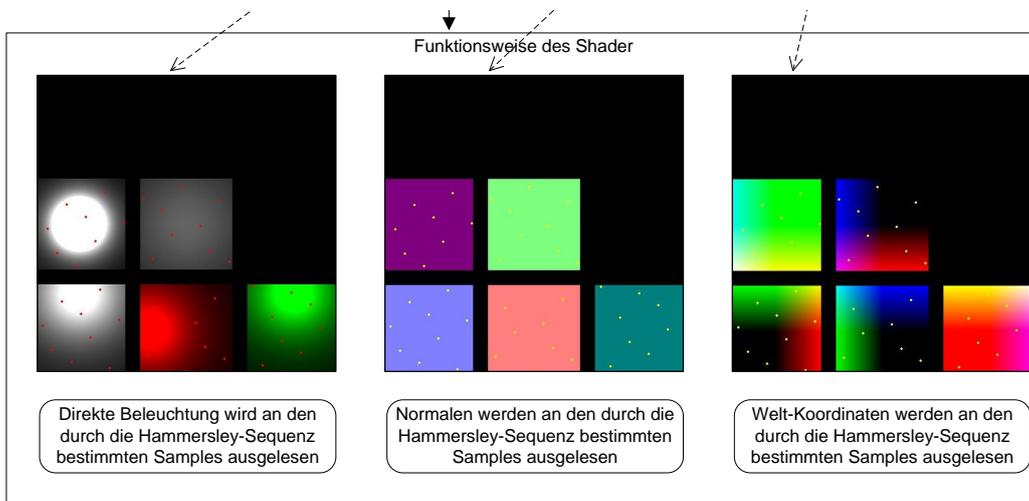
Nachdem die Geometrie erstellt wurde, wird die Szene aus Sicht der Lichtquelle gerendert um die Shadow Map zu generieren. Nun werden die Shadow Map und die Geometrie mit den Verbindungen zu den Textur-Atlas-Koordinaten an den ersten Shader übergeben. In diesem Schritt wird die direkte Beleuchtung berechnet. Außerdem werden auch die Normalen, die Welt-Koordinaten und das Material jeweils in einen Textur-Atlas geschrieben.

Für die Berechnung des indirekten Lichts wird die Hammersley-Sequenz benötigt. Diese wird auf der CPU berechnet und in Form einer Textur an den zweiten Shader übergeben. Der zweite Shader bekommt neben der Textur, welche die Hammerley-Sequenz enthält, auch alle Textur-Atlanten die im ersten Shader erstellt wurden als Eingabe-Texturen. Diese werden als Grundlage zur Berechnung des indirekten Lichts benötigt. Das indirekte Licht wird, kombiniert mit dem direkten Licht, ebenfalls in Form eines Textur-Atlas gespeichert.

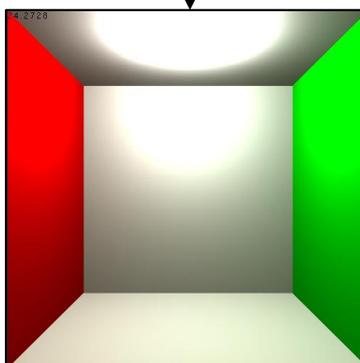
Schließlich wird der letzte Shader aufgerufen, in welchem der Textur-Atlas mit der kombinierten Beleuchtung auf die Geometrie der Szene abgebildet wird um diese wieder dreidimensional darzustellen.

# Programmablauf





3.Shader: Abbilden der kombinierten direkten und indirekten Beleuchtung aus dem Textur-Atlas auf die Szenen-Geometrie





**Abbildung 14:** Material der Conference Room Szene in 3d

### 4.3 Geometrie und Textur-Atlas erstellen

Für die Testszene wurden die Flächen der Cornell-Box manuell erzeugt und auch mit Textur-Koordinaten versehen. Diese Lösung bietet eine einfache und effiziente Möglichkeit um die Funktionsweise der Atlas-basierten globalen Beleuchtung zu testen. Da das Verfahren im Bildraum arbeitet, ist die Komplexität der Szene auch nicht ausschlaggebend für die Dauer der Berechnung. Allerdings werden komplexe Szenen nicht direkt in OpenGL-Code geschrieben, sondern mit Modellierungsprogrammen erstellt.

Um auch komplexere Modelle darstellen zu können, wurde der Wavefront-OBJ-Loaders GLM in das Programm eingebunden. Genauso wie die Modellierung komplexer Objekte oder Szenen kann auch die Erstellung eines Textur-Atlas mit Hilfe eines Modellierungsprogrammes vorgenommen werden. Die automatische Generierung eines Textur-Atlas liefert dabei gute Ergebnisse, wie in Abbildung 15 zu sehen ist. Es werden alle Teile der Szene korrekt abgebildet und auch die Größenverhältnisse bleiben erhalten.

### 4.4 G-Buffer Textur-Atlas erzeugen

Der erste Shader benötigt die Szenen-Geometrie mit Textur-Koordinaten und die Shadow Map. Im Vertex Shader wird nun jedem Vertex, die durch die Textur-Koordinate angegebene Position, zugewiesen. Dieser Schritt wird in der Abbildung „Programmablauf“ durch die erste Grafik verdeutlicht. Für die abgebildete Szene der Cornell-Box werden dabei für jeden Eckpunkt die Textur-Koordinaten verwendet, um die Eckpunkte auf den Atlas

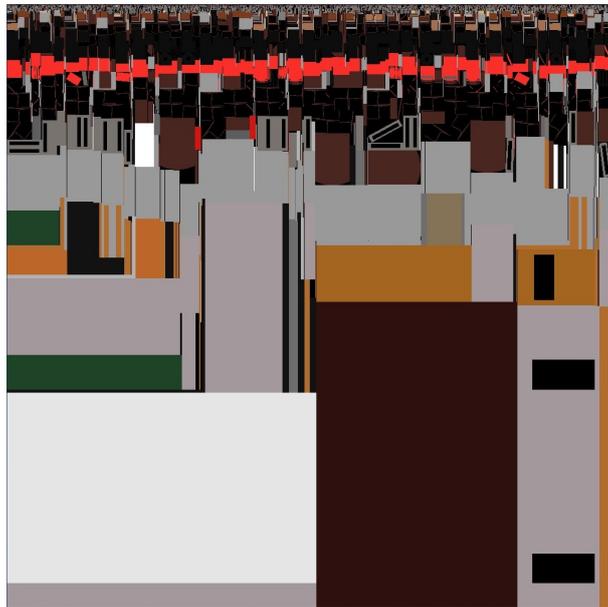


Abbildung 15: Material der Conference Room Szene im Atlas

abzubilden. Im Fragment Shader wird nun die direkte Beleuchtung berechnet. Dabei wird die Leuchtdichte  $L$  berechnet mit:

$$L = f(r, d) * E \quad (5)$$

Die direkte Beleuchtung wird anhand des Lambert-Beleuchtungsmodells berechnet. Die photometrische Formel der Beleuchtungsstärke  $E$  ist dabei:

$$E = \frac{I * \cos \theta}{d^2} \quad (6)$$

Die Lichtstärke  $I$  definiert wie sehr die Lichtquelle strahlt. Der Abstand des zu beleuchtenden Punktes zur Lichtquelle ist  $d$ . Der Winkel  $\theta$  liegt zwischen der Oberflächennormale des zu beleuchtenden Punktes und dem Vektor von diesem Punkt zur Lichtquelle, dessen Länge  $d$  ist.

Als Ergebnis wird die Leuchtdichte  $L$  in einen Textur-Atlas gerendert. Für das Rendern wird ein Framebuffer-Objekt verwendet um mehrere Ausgabe-Texturen zu ermöglichen. Mit Hilfe des Framebuffer-Objekts werden vier Textur-Atlanten erstellt und mit Informationen beschrieben. Die erste Textur enthält dann die direkte Beleuchtung, die nach der beschriebenen Methode berechnet wird. Die zweite Textur enthält die Oberflächennormalen (Darstellung der Normalen: Die Normalen wurden zur Darstellung in den Bereich von Null bis Eins abgebildet, da negative Farbwerte sonst als schwarze Flächen nicht sichtbar wären), die dritte Textur die Welt-Koordinaten und die vierte Textur das Material der Geometrie. Alle vier

Texturen werden mit den Textur-Koordinaten der Szene hergestellt und enthalten somit alle Oberflächen der Szene.

#### 4.5 Erzeugen einer Sampletextrur mit der Hammersley-Sequenz

Um die indirekte Beleuchtung zu berechnen, müssen Samples, also Stichproben, der direkten Beleuchtung ausgewählt werden. Die Farbe dieser Samples ergibt gewichtet die indirekte Beleuchtung für den jeweiligen Punkt. Da die Samplepunkte auf dem Textur-Atlas ausgewählt werden müssen, benötigt man dazu Werte zwischen Null und Eins, da zweidimensionale Textur-Koordinaten in diesem Bereich benötigt werden.

Eine Möglichkeit wäre es, Zufallszahlen zwischen Null und Eins zu generieren. Diese würden allerdings eine sehr schlechte Verteilung aufweisen. Das bedeutet konkret, dass an einigen Stellen viele Samples erzeugt würden und an anderen wenige. Da die Geometrie über den gesamten Atlas verteilt ist, müssen auch die Samples möglichst gleichmäßig über den Textur-Atlas verteilt sein. Die Low Discrepancy Sequenzen, wie die Halton-Sequenz oder die Hammersley-Sequenz, liefern deutlich besser verteilte Samplepunkte. Im Vergleich in Abbildung 12 wird der Unterschied zwischen zufälligen Werten und der Hammersley-Sequenz deutlich.

Die Hammersley-Sequenz wird auf der CPU erzeugt und in eine Textur gespeichert, um diese an den Shader übermitteln zu können. Dabei wird pro Farbkanal eine Komponente der Hammersley-Sequenz geschrieben. So wird die x-Textur-Koordinate in den Rot-Kanal geschrieben und die y-Textur-Koordinate in den Grün-Kanal. Bei der Hammersley-Sequenz muss die benötigte Anzahl an Samples bereits vor der Berechnung der Sequenz bekannt sein. Bei dieser Anwendung hat diese Einschränkung allerdings keine Auswirkungen. Die Elemente der zweidimensionalen Hammersley-Sequenz werden dabei durch  $X_{(i)} = (\frac{i}{N}, \Phi_{(2)}(i))$  berechnet.  $\Phi$  ist dabei die radikal inverse Funktion wobei  $i$  eine natürliche Zahl sein muss und der Basis 2. Mit der radikal inverse Funktion werden die Zahlen  $i$  aus dem Dezimalsystem in das Binärzahlensystem überführt. Dieses Ergebnis wird dann am Komma gespiegelt und wieder ins Dezimalsystem überführt. Dadurch entstehen Zahlen zwischen Null und Eins. Die andere Komponente der Hammersley-Sequenz ist mit  $\frac{i}{N}$  in gleichmäßigen Abständen über den Bereich von Null bis Eins verteilt. Schließlich wird die fertige Textur mit den Werten an den Shader übergeben, um die Textur-Koordinaten dort wieder auszulesen.

#### 4.6 Berechnung des indirekten Lichts

Das indirekte Licht wird in einem zweiten Shader berechnet, welcher als Eingabe die vier Textur-Atlanten mit:

- der direkten Beleuchtung,

- den Normalen,
- den Welt-Koordinaten
- und dem Material

bekommt. Außerdem wird auch die Textur, welche die Hammersley-Sequenz enthält, an den Shader übergeben. Mit Hilfe der Informationen aus diesen fünf Texturen kann nun das indirekte Licht für jedes Fragment berechnet werden. Bevor konkret die einzelnen Schritte der Berechnung erläutert werden, wird zunächst das Modell der Berechnung aus den Grundlagen abgeleitet.

Das Ziel der Berechnung ist es, für jedes Fragment die Rendering Equation zu lösen. Da die Berechnung des Integrals für jedes Fragment allerdings zu aufwändig ist, müssen einige Vereinfachungen angenommen werden. Die erste vereinfachende Annahme ist, dass alle Flächen vollkommen diffus sind. Somit wird die ausfallende Leuchtdichte des Fragments  $A_e$  für alle Winkel als konstant betrachtet. Auch die BRDF wird für diffuse Flächen als konstant angenommen und durch  $\frac{\rho}{\pi}$  beschrieben.

Durch diese Annahme, kann diese aus dem Integral herausgezogen werden. Der Term  $L_e(dA_e, d\omega_o)$ , der die Eigenemission des Flächenelements  $A_e$  beschreibt, wird nun durch das direkte Licht am Punkt  $A_e$  beschrieben und ist unabhängig vom Winkel  $\omega_o$ . Ausgehend von der ursprünglichen Rendering Equation ergibt sich somit aus:

$$L_o(dA_e, d\omega_o) = L_e(dA_e, d\vec{\omega}_o) + \int_{2\pi} f_r(dA_e, d\vec{\omega}_o, d\vec{\omega}_i) \cdot L_i(dA_e, d\vec{\omega}_i) \cdot \cos \theta_i \cdot d\omega_i \quad (7)$$

die folgende 1. Vereinfachung:

$$L_o(dA_e) = L_e(dA_e) + \frac{\rho}{\pi} \int_{2\pi} L_i(dA_e, d\vec{\omega}_i) \cdot \cos \theta_i \cdot d\omega_i \quad (8)$$

Da die Berechnung des Integrals über den Halbraum zu aufwendig zu berechnen wäre, wird das Integral durch die Monte-Carlo-Integration gelöst. So kann das Integral über den Halbraum in eine Summe über alle Samples umgewandelt werden. Diese Summe muss noch durch die Anzahl der Samples geteilt werden. Es wird die Summe der jeweiligen Leuchtdichte am Samplepunkt gewichtet mit dem Einfallswinkel  $\theta$  berechnet.

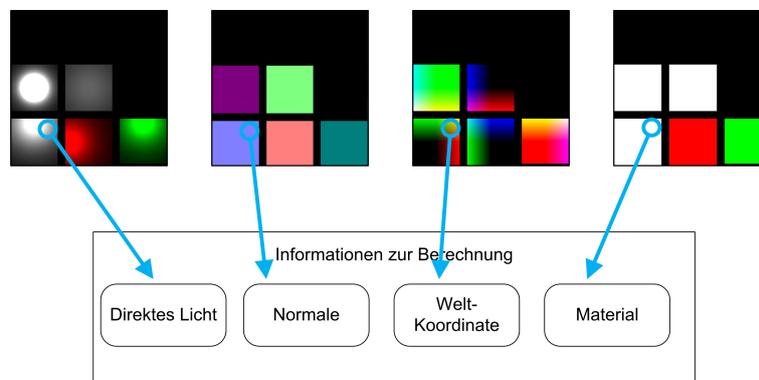
2. Vereinfachung der Rendering Equation

$$L_o(dA_e) = L_e(dA_e) + \frac{\rho}{\pi} \sum_{i=1}^N L_i(dA_e, d\vec{\omega}_i) \cdot \cos \theta_i \cdot 2\pi \quad (9)$$

Nun soll gezeigt werden, wie sich diese Berechnung konkret umsetzen lässt. Als Berechnungsgrundlage dienen die vorgestellte Formel, sowie

die Texturen. Der Fragment Shader wird pro Fragment, was in diesem Fall einem Pixel der Textur entspricht ausgeführt. Dabei wird die folgende Berechnung für jeden Pixel des Ergebnis-Textur-Atlas ausgeführt.

Zunächst wird für den aktuellen Pixel, also für den Pixel dessen Leuchtdichte in diesem Durchgang berechnet werden soll, die Position in Welt-Koordinaten bestimmt. Dazu wird auf den Textur-Atlas, der die Welt-Koordinaten enthält, an der Stelle zugegriffen, an der sich der aktuelle Pixel befindet. Ebenso werden die Normale, das Material und das direkte Licht an dieser Stelle benötigt. Dieser Vorgang ist in Abbildung 16 dargestellt. Auf alle Texturen wird an der entsprechenden Stelle zugegriffen und der Wert ausgelesen.

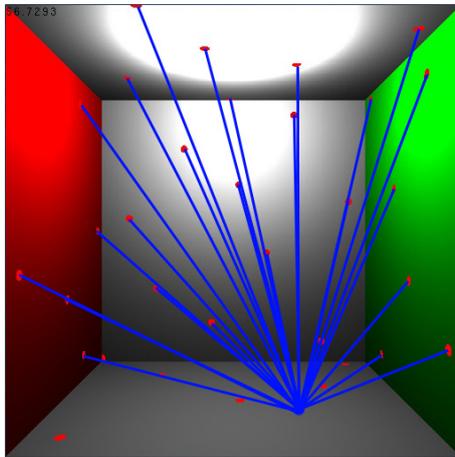


**Abbildung 16:** Information werden aus dem jeweiligen Atlas auslesen

Nun werden für jeden Pixel des Textur-Atlas  $N$  Samples der direkten Beleuchtung betrachtet, um das indirekte Licht für diese Stelle zu berechnen. Dazu werden an  $N$  Samplepunkten, die mit der Hammersley-Sequenz ausgewählt wurden, die direkten Beleuchtungsinformationen abgefragt. Aus der Textur, welche die Hammersley-Sequenz enthält, werden nacheinander in einer Schleife alle Samplepunkte ermittelt, die zur Berechnung des indirekten Lichts herangezogen werden. Innerhalb dieser Schleife über die Anzahl der Samples wird die folgende Berechnung ausgeführt.

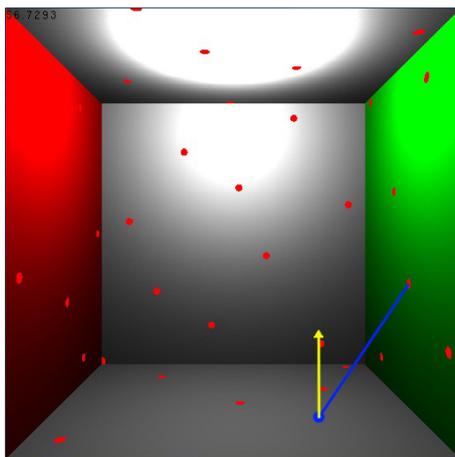
Zunächst wird überprüft, ob die Textur-Koordinate, die durch die Hammersley-Sequenz erzeugt wurde auf dem Textur-Atlas auf eine korrekte Stelle trifft. Das bedeutet es wird überprüft, ob an dieser Stelle im Textur-Atlas tatsächlich Geometrie abgebildet wird oder ob auf eine Lücke im Atlas getroffen wurde. Dazu wird der Alpha-Wert an dieser Stelle des Atlas überprüft. Falls dieser Null ist, wurde dort kein Wert zur Repräsentation der Geometrie gespeichert und es handelt sich um eine Lücke im Atlas. Damit wird dieser Samplepunkt nicht berücksichtigt und der nächste Punkt wird getestet.

Für einen gültigen Samplepunkt wird dann überprüft, ob der Samplepunkt und der zu beleuchtende Punkt einander zugewandt sind. Die Sicht-



**Abbildung 17:** Die blauen Linien führen jeweils zu einem Samplepunkt und zu dem Punkt dessen Beleuchtung grade berechnet wird

barkeit, also die eventuelle Verdeckung durch andere Objekte, wird dabei in dieser Arbeit vernachlässigt. Falls die Punkte zugewandt sind, handelt es sich um ein gültiges Sample für die Berechnung des indirekten Lichts. Nun wird der Winkel  $\theta$  zwischen der Normale des zu beleuchtenden Punktes und der Verbindung von diesem Punkt zum Samplepunkt berechnet (siehe Abbildung 18). Dafür wird die Welt-Koordinate des Samplepunktes benötigt, die wiederum aus dem Textur-Atlas an der entsprechenden Stelle ausgelesen werden kann. Zur indirekten Beleuchtung wird nun die Farbe des Samplepunktes aus dem Textur-Atlas, der die direkte Beleuchtung enthält, ausgelesen und mit  $\cos \theta$  gewichtet.



**Abbildung 18:** Gelb: die Normale an der aktuellen Stelle, blau: die Verbindung zum Samplepunkt, dazwischen liegt der Winkel  $\theta$

Als Dichtefunktion für die Monte-Carlo Integration wird die konstante Dichte  $\frac{1}{2\pi}$  verwendet, da die erzeugten Samples gleichmäßig verteilt sein sollen. Deshalb wird die Farbe des Samplepunkts für die indirekte Beleuchtung noch zusätzlich mit  $2\pi$  multipliziert. Die so berechnete Farbe wird dann mit den jeweiligen Farbwerten der anderen Samples aufaddiert und letztlich durch die Anzahl der Samples geteilt, um den Wert für die indirekte Beleuchtung zu erhalten. Um das Endergebnis zu erhalten wird der Wert noch mit der diffusen BRDF  $\frac{\rho}{\pi}$  gewichtet. Für die Darstellung werden das direkte Licht und das indirekte Licht kombiniert in einen Textur-Atlas gespeichert. Dazu werden die beiden Werte addiert und das Ergebnis in die Ausgabe-Textur des Shaders geschrieben.

#### **4.7 Darstellung der Szene mit der berechneten Beleuchtung**

Als Ergebnis liefert der 2. Shader einen Textur-Atlas mit der kombinierten Beleuchtung. Um die Szene nun dreidimensional mit diesen Beleuchtungsinformationen darstellen zu können, wird die Szene zunächst gezeichnet und dann wird der 3. Shader aufgerufen. Dieser bekommt den Textur-Atlas mit der Beleuchtung übergeben und legt diesen als Textur auf die gesamte Szene.

## 5 Ergebnisse und Auswertung

Im folgenden Teil sollen die Ergebnisse in Hinsicht auf Qualität und Performance bewertet werden. Dazu wurden verschiedene Testszenen mit verschiedenen Sampleanzahlen getestet. Besonders die Entstehung von Artefakten bei Kamerabewegung soll an einigen Beispielen gezeigt werden, beziehungsweise wie deren Entstehung durch den Einsatz eines Textur-Atlas verhindert werden kann.

### 5.1 Qualität

Das Verfahren zur Atlas-basierten globalen Beleuchtung im Bildraum wurde wie beschrieben umgesetzt und liefert die folgenden Ergebnisse. Vor allem die Auswirkungen auf die Beleuchtungssimulation von Objekten, die im aktuellen Bildausschnitt nicht zu sehen sind, wurde realisiert. Speziell in dieser Hinsicht liefert das Verfahren gute Ergebnisse.

Das folgende Beispiel soll zeigen, dass auch Teile der Szene, die momentan nicht zu sehen sind, Einfluss auf die Beleuchtung haben. In Abbildung 19 ist deutlich zu erkennen, dass die beiden Kisten in der Mitte des Raums auch rot angestrahlt werden, wenn die linke, rote Wand nicht mehr zu sehen ist. Dabei ist die eigentliche Farbe der beiden Kisten weiß. Ebenso ist auch in Abbildung 20 zu erkennen wie die beiden Kisten von der anderen Seite durch indirektes Licht der rechten Wand, grün beleuchtet werden. Dadurch wird auch deutlich, dass Flächen nur indirektes Licht von anderen zugewandten Flächen erhalten, auch wenn dabei die Sichtbarkeit nicht berücksichtigt wird. Da die indirekte Beleuchtung auf dem Textur-Atlas durchgeführt wird, bleibt die Beleuchtungssituation auch bei Kamerabewegungen und beim Auftauchen oder Verschwinden von einzelnen Objekten aus dem Sichtfeld der Kamera konstant.

Die Anzahl der Samples bestimmt maßgeblich die Qualität der Ergebnisse. Wie in Abbildung 22 zu sehen ist kann beispielsweise mit zehn Samples, selbst bei einer einfachen Szene, noch kein ausreichendes Ergebnis erzielt werden. So ist zu sehen, dass die rote Fläche deutlich mehr Einfluss auf die angrenzenden Flächen bekommt als die grüne Fläche, obwohl beide Fläche die gleiche Größe haben, die gleiche direkte Beleuchtung bekommen und auch die Winkel zu den angrenzenden Flächen identisch sind. Da jedoch mehr Samplepunkte auf der roten Fläche erstellt wurden, wird diese stärker gewichtet. Bereits mit 25 Samples lässt sich für diese einfache Szene ein recht gutes Ergebnis erzielen. Die rote und die grüne Fläche fließen nun ungefähr gleich stark in die indirekte Beleuchtung ein.

Bei dem Testbild mit 100 Samples ist zu bemerken, dass die gesamte indirekte Beleuchtung etwas heller ausfällt als bei allen anderen Sampleanzahlen. Das ist dadurch zu erklären, dass bei 100 ausgewählten Samples, zufällig mehr Samples auf hellere Stellen der Textur mit der direkten Be-

leuchtung gefallen sind. Zwischen 200 und 400 Samples ist kein wesentlicher Unterschied mehr festzustellen.

Auch die Verteilung der Samples ist wichtig. Werden die Samples beispielsweise nur über einen Teil des Atlas verteilt, so werden ausschließlich Flächen, die dort liegen zur indirekten Beleuchtung herangezogen. Das Ergebnis ist Abbildung 21 zu sehen. Die hintere Wand leuchtet sehr viel stärker rot und grün als es bei einer gleichmäßigen Verteilung der Samples der Fall sein dürfte.

Um möglichst gute Ergebnisse zu erhalten, ist es außerdem notwendig, dass die relativen Größenverhältnisse der verschiedenen Flächen auch so im Textur-Atlas wiedergespiegelt werden. Konkret bedeutet das, dass Flächen die in der Szene sehr klein sind auch im Textur-Atlas klein sein müssen. Bei der automatischen Generierung eines Textur-Atlas stellt dies allerdings kein Problem dar. Wie zum Beispiel in Abbildung 15 zu erkennen ist, werden kleinere Details wie die Stühle auch im Atlas sehr viel kleiner abgebildet als die großen Flächen der Szene.

Ein Problem des Verfahrens sind numerische Ungenauigkeiten bei Texturzugriffen. So können an Flächengrenzen störende Artefakte entstehen, wie in Abbildung 19 unten rechts, zwischen der hinteren grünen und der weißen Flächen zu sehen ist. Dort entsteht eine flackernde, schwarze Kante. Vor allem bei komplexeren Szenen mit vielen Details wie dem Conference Room wird dieses Problem deutlich. Die Stühle bekommen beispielsweise störende, schwarz flackernde Ränder. Das Problem könnte durch höhere Textur-Atlas-Auflösungen verringert werden, allerdings sind kleine Details dennoch schwierig.

## 5.2 Test einer komplexeren Szene

Neben der Cornell-Box wurde auch eine komplexe Szene getestet. Die Conference Room Szene besteht aus 167000 Dreiecken und stellt so vor allem für die Erstellung des Textur-Atlas eine Herausforderung dar. Wie in Abbildung 15 zu sehen ist, konnte jedoch automatisch ein Textur-Atlas erstellt werden, der die gesamte Szene abbildet.

Allerdings enthält die Szene einige Flächen mit falschen Normalen, die so keine korrekte, direkte Beleuchtung zulassen (siehe Abbildung 26). Deshalb sind die Ergebnisse der indirekten Beleuchtung, also der Atlas-basierten globalen Beleuchtung, für diese Szene nicht aussagekräftig. Es konnte gezeigt werden, dass auch komplexe Szenen in einem Textur-Atlas dargestellt werden können. Für eine korrekte Berechnung müsste die Szene allerdings fehlerfrei sein.

### 5.3 Performance

Um die Geschwindigkeit der Atlas-basierten globalen Beleuchtung zu testen, wurden die beiden bereits beschriebenen Testszenen, die Cornell-Box und der Conference Room mit verschiedenen Sampleanzahlen gerendert. Die Geschwindigkeit des Verfahrens hängt dabei zum einen von der verwendeten Textur-Atlas-Auflösung ab, zum anderen von der verwendeten Sampleanzahl. Pro Sample werden insgesamt fünf Texturzugriffe benötigt, was bei einer Auflösung von 512x512 und einer Sampleanzahl von 100 zu einer Summe von 1310720 Textur-Zugriffen pro Frame führt. Da das Verfahren im Bildraum arbeitet, sollte die Komplexität der Szene, also die Anzahl der Dreiecke die zur Modellierung verwendet wurden, keine Auswirkung auf die Geschwindigkeit haben. Da allerdings der verwendete Wavefront-OBJ-Loader GLM bei aufwendigen Szenen Geschwindigkeitseinbrüche verursacht, lagen die gemessenen Bildwiederholungsraten für die Conference Room Szene deutlich unter denen der einfachen Szenen, wie in Tabelle 1 zu sehen ist.

Szene	Anzahl der Samples	Anzahl der Dreiecke	FPS	Auflösung
Cornell-Box	25	16	57	512x512
Cornell-Box	50	16	48	512x512
Cornell-Box	100	16	27	512x512
Cornell-Box	200	16	14	512x512
Cornell-Box	400	16	7	512x512
Cornell-Box	10	16	48	1024x1024
Cornell-Box	25	16	24	1024x1024
Cornell-Box	50	16	14	1024x1024
Cornell-Box	100	16	7	1024x1024
Cornell-Box	200	16	4	1024x1024
Cornell-Box	400	16	2	1024x1024
Conference Room	10	167000	14	1024x1024
Conference Room	50	167000	7	1024x1024
Conference Room	100	167000	4	1024x1024
Conference Room	200	167000	2	1024x1024

**Tabelle 1:** Übersicht Performance. Verwendetes System: Intel Core 2 Duo E8200 2,66 GHz, Zwei GB Arbeitsspeicher, nVidia Geforce 8800GT

### 5.4 Erweiterungsmöglichkeiten

Es konnte gezeigt werden, dass eine Atlas-basierte globale Beleuchtung interessante Ansätze bietet. Eine mögliche Erweiterung, die das Verfahren verbessern würde, wäre die Einbeziehung der Sichtbarkeit. Somit könnte auch für komplexe Szenen eine korrekte globale Beleuchtung berechnet werden. Dadurch würden Objekte die verdeckt sind, keine falschen Beleuchtungsinformationen erzeugen. Besonders interessant wäre dabei ein Sichtbarkeitstest im Bildraum, um diesen eventuell auch mit Hilfe des Tex-

tur-Atlas zu berechnen.

Zusätzlich könnte man Geschwindigkeitsoptimierungen erforschen um die Performance zu steigern. Dabei wäre zum Beispiel für statische Szenen sehr einfach möglich, die gesamte Beleuchtungsberechnung lediglich einmal durchzuführen. Da die Beleuchtung auf dem Atlas berechnet wird, wird in einem Schritt die Beleuchtung der gesamten Szene berechnet. Die Szene könnte dann ohne weitere Berechnungen aus allen möglichen Perspektiven betrachtet werden. Für diese Arbeit war dieser Ansatz jedoch nicht relevant, da auch die Performance des Verfahrens getestet werden sollte.

## 5.5 Fazit

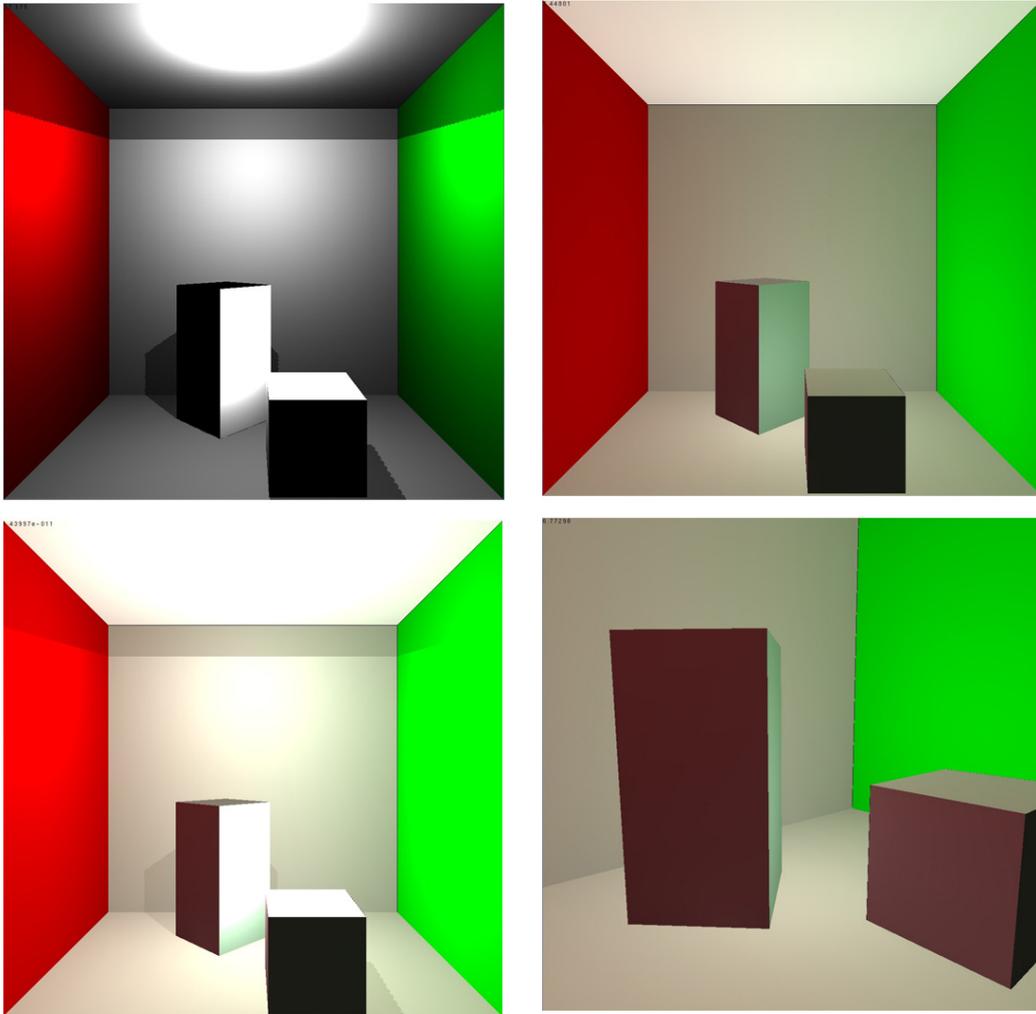
In dieser Studienarbeit wurde untersucht, ob es möglich ist eine Atlasbasierte globale Beleuchtungssimulation im Bildraum auf der GPU zu implementieren. Dazu wurden zunächst die Möglichkeiten, welche ein Textur-Atlas bietet, untersucht. Es konnte gezeigt werden, dass ein Textur-Atlas sowohl zur Repräsentation einfacher, als auch komplexer Szenen verwendet werden kann.

Aufbauend auf diesen Erkenntnissen wurde ein Verfahren entwickelt um eine globale Beleuchtungssimulation mit Hilfe eines Textur-Atlas zu ermöglichen. Dazu wurden zunächst die Möglichkeiten der GPU-Programmierung in Bezug auf die Verwendung eines Atlas untersucht. Durch die Verwendung mehrerer Textur-Atlanten, konnte eine effiziente Speicher methode für die, zur Berechnung der indirekten Beleuchtung benötigten Informationen, gefunden werden. Die zur Berechnung verwendete Formel konnte mit den angenommenen Vereinfachungen plausibel aus der Rendering Equation abgeleitet werden. Dabei wird versucht das Integral der Rendering Equation durch Monte-Carlo-Integration zu lösen.

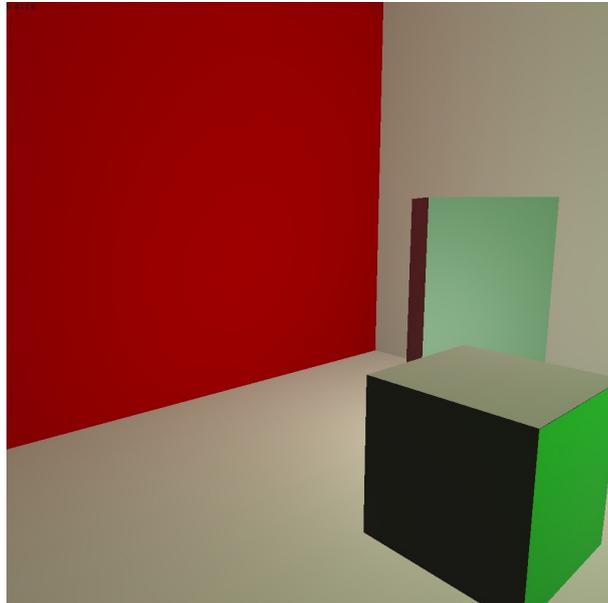
Das Verfahren liefert eine gute Annäherung an eine globale Beleuchtung, dabei muss jedoch berücksichtigt werden, dass ausschließlich diffuse Materialien simuliert werden und lediglich eine Lichtindirektion verfolgt wird. Der Lichtaustausch zwischen verschiedenen Flächen kann mit diesem Verfahren gut simuliert werden.

Für die Darstellung von komplexen Szenen wäre eine Sichtbarkeitsprüfung sehr von Vorteil. Außerdem bereitet der Modell-Loader GLM einige Probleme beim Laden großer Szenen.

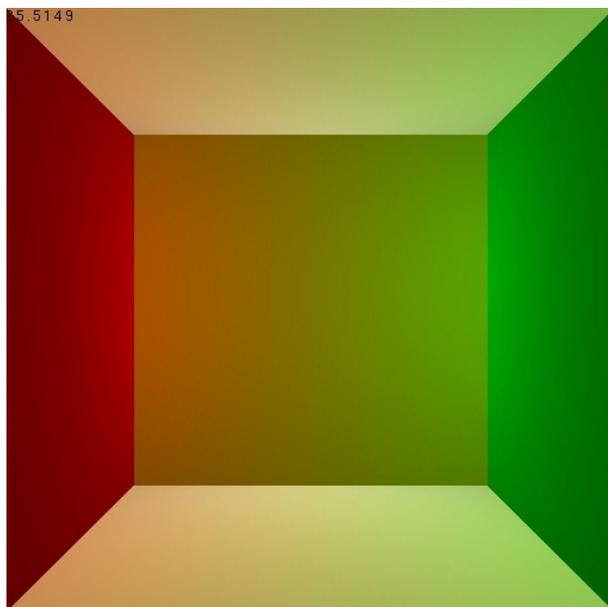
Das Ziel, Artefakte bei Kamerabewegungen zu vermeiden, wurde erreicht. Falsche Beleuchtungseffekte, die durch das Auftauchen oder Verschwinden einzelner Objekte im aktuell für die Kamera sichtbaren Bereich entstehen können, werden durch die Verwendung eines Textur-Atlas umgangen.



**Abbildung 19:** Testszene, links oben: direkte Beleuchtung, rechts oben: indirekte Beleuchtung, links unten: direkte und indirekte kombiniert, rechts unten: Kisten von der Seite mit rotem, indirektem Licht



**Abbildung 20:** Testszene, indirekte Beleuchtung, Kisten von der anderen Seite mit grünem, indirektem Licht



**Abbildung 21:** Testszene, indirekte Beleuchtung, mit ungleichmäßig verteilten Samples

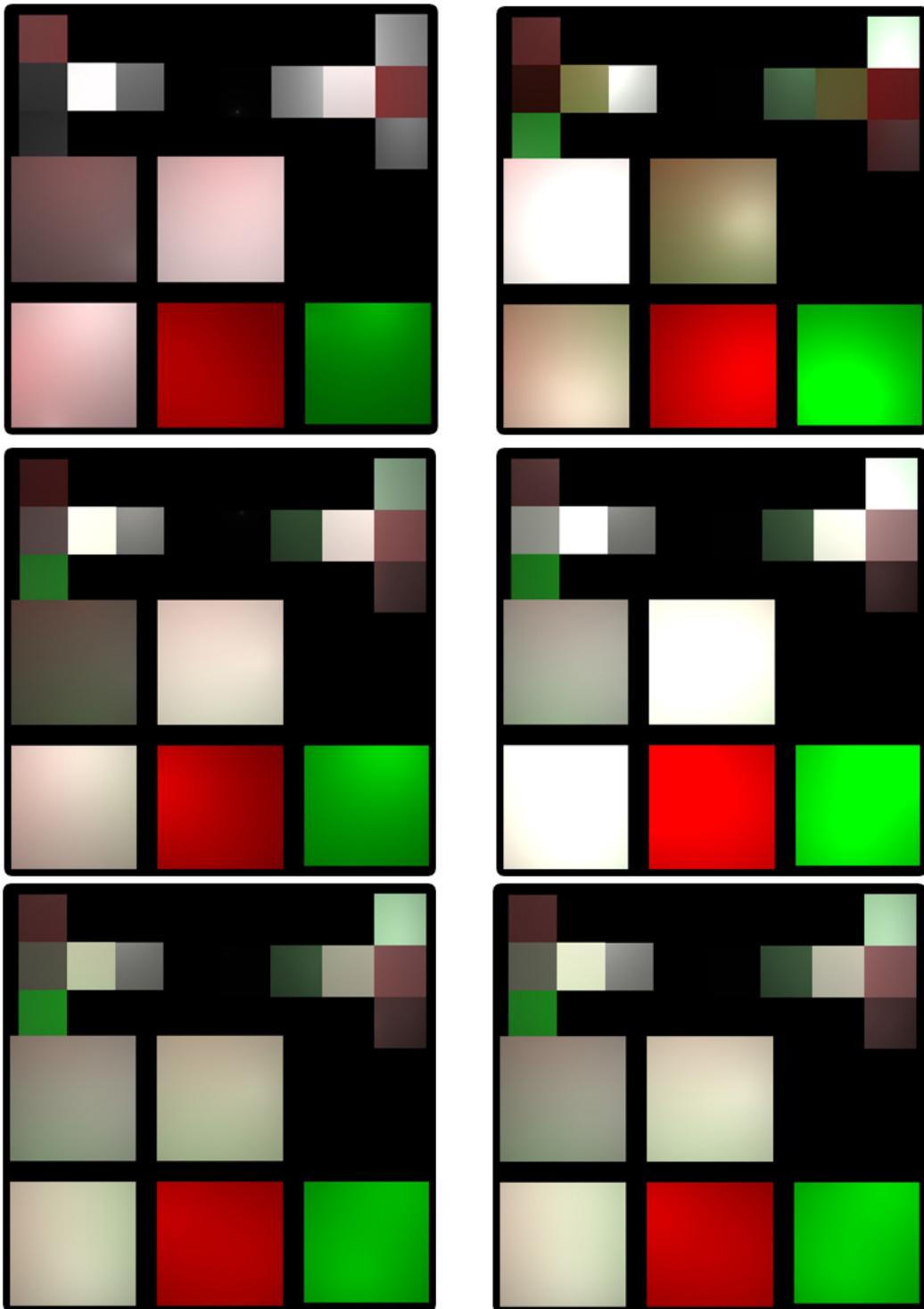


Abbildung 22: Testszene, indirekte Beleuchtung, oben links: 10 Samples, oben rechts: 25 Samples, mitte links: 50 Samples, mitte rechts: 100 Samples, unten links: 200 Samples, unten rechts 400 Samples



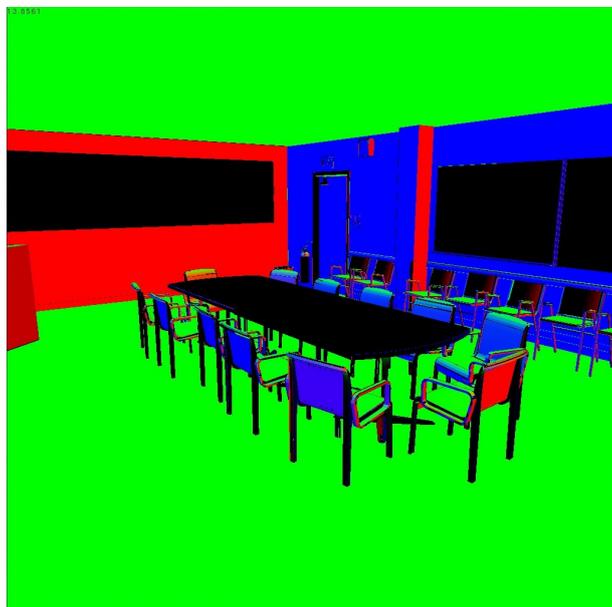
**Abbildung 23:** Testszene Conference Room, kombinierte direkte und indirekte Beleuchtung, 100 Samples, Auflösung 1024x1024



**Abbildung 24:** Testszene Conference Room, indirekte Beleuchtung, 100 Samples, Auflösung 1024x1024



**Abbildung 25:** Testszene Conference Room, direkte Beleuchtung, 100 Samples, Auflösung 1024x1024



**Abbildung 26:** Testszene Conference Room, Normalen, die Tafeln, der Tisch und die Decke haben falsche Normalen, Auflösung 1024x1024

## Literatur

- [1] Mittring, M.: Finding next gen: Cryengine 2. In: SIGGRAPH '07: ACM SIGGRAPH 2007 courses, New York, NY, USA, ACM (2007) 97–121
- [2] Reinbothe, C., Boubekur, T., Alexa, M.: Hybrid ambient occlusion. EUROGRAPHICS 2009 Areas Papers (2009)
- [3] Ritschel, T., Grosch, T., Seidel, H.P.: Approximating dynamic global illumination in image space. In: I3D '09: Proceedings of the 2009 symposium on Interactive 3D graphics and games, New York, NY, USA, ACM (2009) 75–82
- [4] McGuire, M., Luebke, D.: Hardware-accelerated global illumination by image space photon mapping. In: Proceedings of the 2009 ACM SIGGRAPH/EuroGraphics conference on High Performance Graphics, New York, NY, USA, ACM (August 2009)
- [5] Dachsbacher, C., Stamminger, M.: Reflective shadow maps. In: I3D '05: Proceedings of the 2005 symposium on Interactive 3D graphics and games, New York, NY, USA, ACM (2005) 203–231
- [6] Thiedemann, S.: Globale Beleuchtung im Bildraum unter besonderer Berücksichtigung der Sichtbarkeitsbestimmung (2009)
- [7] Kajiya, J.T.: The rendering equation. In: 13th annual conference on Computer graphics and interactive techniques (SIGGRAPH). (1986) 143–150
- [8] Everitt, C.: Interactive order-independent transparency (2001)
- [9] [http://www.gamasutra.com/features/20060126/ivanov\\_01.shtml](http://www.gamasutra.com/features/20060126/ivanov_01.shtml) (2009) [Online; Stand 27. März 2010].
- [10] <http://msdn.microsoft.com/en-us/library/ee422528.aspx> (2009) [Online; Stand 27. März 2010].
- [11] Hasenfratz, J.M., Lapierre, M., Holzschuch, N., Sillion, F.: A survey of real-time soft shadows algorithms (2003)
- [12] Hans, W., Grosch, T., Feldmann, T., Paulus, D., Müller, S.: Modell der Bildentstehung mit HDR-Kameras. In: 12. Workshop Farbbildverarbeitung. (10 2006) 97–108
- [13] Müller, S.: Vorlesungsmaterial computergraphik 2 (2008)

- [14] Wong, T.T., Luk, W.S., Heng, P.A.: Sampling with hammersley and halton points. *J. Graph. Tools* **2**(2) (1997) 9–24
- [15] Müller, S.: Vorlesungsmaterial photorealistische computergraphik (2008)