

# Implementierung eines Subsurface Scattering Shader Plugins für die Augenblick Raytracing Engine

## Diplomarbeit

zur Erlangung des Grades eines Diplom-Informatikers  
im Studiengang Computervisualistik

vorgelegt von  
Dennis Wadkien

Erstgutachter: Prof. Dr.-Ing. Stefan Müller  
(Institut für Computervisualistik, AG Computergraphik)  
Zweitgutachter: Dipl. Inf. Oliver Abert  
(Institut für Computervisualistik, AG Computergraphik)

Koblenz, im Oktober 2008



## 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)



**Here be Aufgabenstellung**



# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
<b>2</b>	<b>Grundlagen</b>	<b>5</b>
2.1	Das Phänomen des Subsurface Scattering . . . . .	5
2.2	Homogene und nicht-homogene Materialien . . . . .	6
2.3	Renderingverfahren . . . . .	8
2.3.1	Scanline Rendering . . . . .	8
2.3.2	Raytracing . . . . .	9
2.4	Beleuchtungsmodelle . . . . .	10
2.4.1	Die BRDF - Bidirectional Reflectance Distribution Function . . . . .	10
2.4.2	Das Lambert Beleuchtungsmodell . . . . .	12
2.4.3	Das Phong Beleuchtungsmodell . . . . .	13
2.5	Der Monte Carlo Pathtracer . . . . .	14
2.5.1	Die Monte Carlo Intergration . . . . .	14
2.5.2	Russisches Roulette . . . . .	16
2.5.3	Der Pathtracer . . . . .	17
2.6	Strahlverfolgung . . . . .	19
2.6.1	Snell's Brechungsgesetz . . . . .	19
2.6.2	Der Fresnel Term . . . . .	19
2.6.3	Rotationen mit Hilfe der Axis-Angle Rotation . . . . .	20
2.7	Raytracer Beschleunigungsverfahren . . . . .	21
2.7.1	SIMD - Single Instruction Multiple Data . . . . .	21
2.7.2	Multi-Threading . . . . .	21
2.7.3	Boundingvolumes . . . . .	21
2.8	Der Raytracer <i>Augenblick</i> . . . . .	22
2.8.1	Die Schnittstelle zum Raytracer <i>Augenblick</i> . . . . .	22
2.8.2	Plattform und Entwicklungsumgebung . . . . .	23
<b>3</b>	<b>Vorangegangene Arbeiten</b>	<b>25</b>
<b>4</b>	<b>Der Subsurface Scattering Shader</b>	<b>29</b>
4.1	Die Wahl des Pathtracers . . . . .	30
4.2	SIMD für Subsurface Scattering . . . . .	30
4.3	Simulation von Material . . . . .	30
4.4	Geeignete Modelle für das Subsurface Scattering . . . . .	31
4.5	Ablauf des Shaders im Detail: der einfache Pathtracer . . . . .	32
4.5.1	Diffus beleuchten . . . . .	32
4.5.2	Diffus weiterverfolgen . . . . .	33
4.5.3	Reflektieren . . . . .	33
4.6	Ablauf des Shaders im Detail: das Subsurface Scattering . . . . .	33
4.6.1	Refraktion am Übergang Luft-Material . . . . .	34

4.6.2	Reflektion und Refraktion im Material . . . . .	35
4.6.3	Refraktion und Reflektion am Übergang Material-Luft . . . . .	35
4.6.4	Refraktion am Übergang Material-Material . . . . .	36
4.7	Subsurface Phong Beleuchtung . . . . .	36
<b>5</b>	<b>Der Subsurface Scattering Shader:</b>	
	<b>Tests und Möglichkeiten</b>	<b>39</b>
5.1	Monte Carlo Raytracing für das Subsurface Scattering . . . . .	39
5.1.1	Raytracing ohne Monte Carlo . . . . .	39
5.1.2	Monte Carlo nur für Refraktions- und Reflexionstiefe . . . . .	40
5.1.3	Monte Carlo Dichteverteilungen innerhalb des Materials . . . . .	41
5.1.4	Objektabhängige Dichteverteilungen . . . . .	42
5.2	Beschleunigung des Subsurface Scattering Shaders . . . . .	43
5.2.1	Erweitern des Pathtracers zu einem Multi-Path Tracer . . . . .	44
5.2.2	Ersetzen des Monte Carlo Verfahrens durch feste Sample Muster . . . . .	45
<b>6</b>	<b>Subsurface Scattering in Echtzeit</b>	<b>47</b>
6.1	Subsurface Scattering durch Skalarwerte . . . . .	47
6.2	Subsurface Scattering durch Licht-Lot . . . . .	47
6.2.1	Lichttransport auf beleuchteten Flächen . . . . .	48
6.2.2	Das Licht-Lot . . . . .	49
6.2.3	Interpolation zwischen Front- und Backshader . . . . .	50
<b>7</b>	<b>Ergebnisse</b>	<b>53</b>
7.1	Subsurface Scattering mit Hilfe eines Monte Carlo Pathtracers . . . . .	54
7.2	Vereinfachtes Subsurface Scattering durch Licht-Lot . . . . .	54
7.3	Vereinfachtes Subsurface Scattering durch Skalarwerte . . . . .	55
<b>8</b>	<b>Diskussion</b>	<b>57</b>
8.1	Aufgabenschwerpunkte: Performanz versus korrekte Lösung . . . . .	57
8.2	BSSRDF für Orte im Material . . . . .	58
8.3	Nicht geschlossene Modelle . . . . .	58
<b>9</b>	<b>Vergleich und Auswertung</b>	<b>61</b>
9.1	Vergleich der Shader untereinander . . . . .	61
9.2	Vergleich der Renderingzeiten . . . . .	62
9.3	Vergleich mit <i>A practical model for subsurface light transport</i> . . . . .	63
<b>10</b>	<b>Ausblick</b>	<b>65</b>
	<b>Literatur</b>	<b>67</b>

## Abbildungsverzeichnis

1	Beispiel für Subsurface Scattering. Scattering-example.jpg entnommen aus <a href="http://en.wikipedia.org/wiki/Subsurface_scattering">http://en.wikipedia.org/wiki/Subsurface_scattering</a> am 14.10.2008 . . . . .	2
2	Vergleich zwischen einfacher Beleuchtung und Subsurface Scattering. (a) einfache Phong Beleuchtung ohne Subsurface Scattering, (b) Subsurface Scattering gerendert mit dem Monte Carlo Pathtracer dieser Diplomarbeit . . . . .	2
3	Materialmischungen in verschiedenen Zuständen: (a) granuläre Einschlüsse, (b) Einschlüsse teilweise vermischt zu Schlieren, (c) Homogene Mischung, (d) Schichten mit verschiedenen Eigenschaften . . . . .	7
4	(a) typische BRDF, (b) $\theta$ und $\phi$ für den Betrachter . . . . .	12
5	Annäherung eines Integrals durch die Monte Carlo Interation mit einer Dichtefunktion $p(x) \approx f(x)$ . Die Abweichung ist hier gleich Null. . . . .	15
6	Annäherung einer BRDF durch die Monte Carlo Integration . . . . .	16
7	Darstellung der möglichen Pfade eines Pathtracers . . . . .	18
8	Darstellung der V-förmigen Schluchten auf der Oberfläche eines Objekts. Entnommen aus [16]. . . . .	26
9	(a) Darstellung der Dipol-Lichtquelle, (b) vereinfachter Strahlengang. Entnommen aus [11]. . . . .	27
10	Die Büste der Diana, gerendert mit dem Shader von Henrik Wann Jensen et. al. [11]. . . . .	27
11	Darstellung der Subsurface Phong Beleuchtung . . . . .	37
12	Der Subsurface Scattering Shader ohne Monte Carlo Raytracing. (a) mit großer Eindringtiefe, (b) mit geringer Eindringtiefe . . . . .	40
13	Ein Terracotta Hase. Das Monte Carlo verfahren wurde hier nur auf die Eindringtiefe angewendet. . . . .	40
14	Lichttransport durch Material . . . . .	41
15	Ein VW Beetle. Die Wahrscheinlichkeit der Strahlen das Modell zu durchdringen ist auf Null gesetzt. . . . .	42
16	Ein Stanford Hase. Bei (a) hatten die Strahlen eine 90% Chance intern reflektiert zu werden, bei (b) eine 90% Chance weiter hinein refraktiert zu werden. . . . .	43
17	Das Modell eines Jade Drachens. Die Wahrscheinlichkeit der Strahlen das Modell zu durchdringen ist bei 70%. . . . .	44
18	Skalarwert $s$ abhängig vom Richtungsvektor . . . . .	49
19	Stanfordbunny mit Licht-Lot Shader . . . . .	50
20	Nähte zwischen den einzelnen Licht-Lot Shadern . . . . .	51

21	Die einzelnen Schritte des Subsurface Scattering Shaders in Bildern. (a) Die Szene mit Phong Beleuchtung, (b) der erste Rendschritt des Monte Carlo Pathracers und (c) der fertige Drache nach 1000 Durchläufen . . . . .	53
22	Ein Terracotta Hase und ein Buddha gerendert mit dem Licht-Lot Shader. . . . .	55
23	Ein Buddha gerendert mit dem Skalar Shader. (a) ohne den zusätzlichen Schattenfühler, (b) mit dem zusätzlichen Schattenfühler . . . . .	55
24	Direkter Vergleich der drei Shader. (a) Scalar Shader, (b) Licht-Lot Shader, (c) Monte Carlo Pathtracer . . . . .	61
25	Subsurface Scattering eines Lasers in einem Marmorblock. Entnommen aus [11] . . . . .	63
26	Ein Block gerendert mit (a) dem korrekten Monte Carlo Pathtracer und (b) dem Licht-Lot Shader . . . . .	64

## Abstract

In dieser Ausarbeitung werden drei Beleuchtungsverfahren und mögliche Implementierungen zur Realisierung eines Subsurface Scattering Shaders vorgestellt und diskutiert. Subsurface Scattering bezeichnet allgemein den Lichttransport in die Oberfläche von Objekten hinein und durch sie hindurch. Die korrekte Darstellung dieses Phänomens ist sehr komplex und ist nicht mittels einer einfachen BRDF und gängiger Beleuchtungsverfahren darstellbar. Die drei Verfahren sind:

- Physikalisch korrektes Subsurface Scattering durch das Monte Carlo Pathtracing
- Vereinfachtes Subsurface Scattering durch Nutzung eines Licht-Lots
- Stark vereinfachtes Subsurface Scattering durch ein normalenabhängiges Aufhellen der Kanten.

Durch die Nutzung des Monte Carlo Pathtracers können zudem Beleuchtungseffekte wie das sogenannte Colorbleeding, das heißt Lichttransport von einer farbigen Fläche auf eine andere mittels indirekter Beleuchtung, ermöglicht werden. Jedes Verfahren beinhaltet eine andere Kombination der bekanntlich gegenläufigen Eigenschaften Performanz und Korrektheit, je nachdem in welchem Rahmen und Aufgabenbereich Subsurface Scattering benötigt wird. Am Schluss der Arbeit werden Ergebnisse präsentiert, diskutiert und ein Ausblick auf weiterführende Arbeiten gegeben.

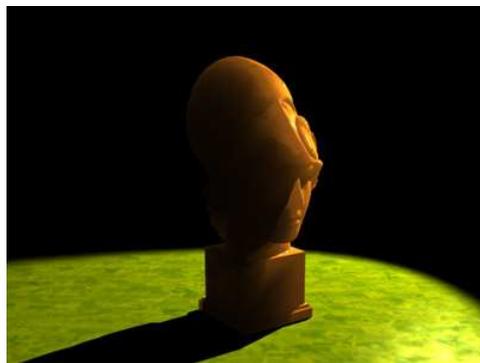
Alle drei Verfahren wurden als Plugin für den Raytracer *Augenblick* von Oliver Abert realisiert.

## 1 Einführung

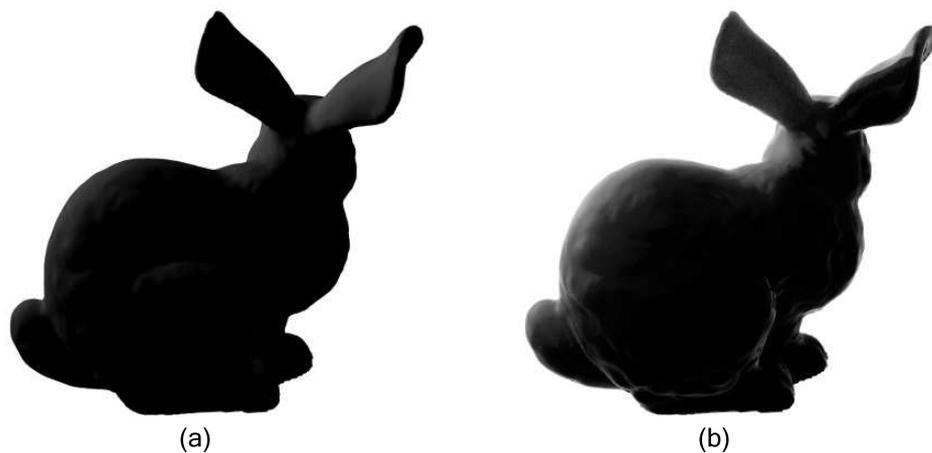
Subsurface Scattering ist eine Notwendigkeit für die realistische Darstellung von Beleuchtung. Die meist genutzte BRDF, die Bidirectional Reflectance Distribution Function, vorgestellt von Nicodemus [15], ist nur eine Vereinfachung der BSSRDF, der Bidirectional Scattering Surface Reflectance Distribution Function. Beide setzen die einfallende Beleuchtungsstärke  $E_i$  mit der ausfallenden Leuchtdichte  $L_o$  ins Verhältnis, doch im Falle der BRDF verlässt der Lichtstrahl die getroffene Fläche an derselben Stelle an der er auftraf. Es findet kein Lichttransport innerhalb der Oberfläche statt. Viele der heute genutzten Beleuchtungsmodelle nähern lediglich die BRDF an. Doch gerade transluzente Materialien, wie Plastik, Haut und Milch lassen sich damit nur schlecht darstellen, sie wirken unecht, künstlich und computergeneriert.

Subsurface Scattering bezeichnet die Streuung von Licht innerhalb einer Oberfläche beziehungsweise innerhalb eines Objektes. Das Licht dringt

durch die Oberfläche in das Material ein und wird innerhalb gebrochen und reflektiert. Dadurch kann ein Lichtstrahl das Objekt an einer anderen Stelle verlassen als er es betreten hat. Lichtstrahlen können je nach Material und Dicke des Objektes sogar komplett durch das Objekt hindurch scheinen. Abbildung 1 zeigt einen von hinten angeleuchteten transluzenten Kopf. Die Oberfläche wirkt weich und wachsartig, und es ist sehr gut zu erkennen wie das Licht hindurch scheint. Abbildung 2 zeigt einen mit Phong Beleuchtung gerenderten Stanford Hasen 2(a) und als Vergleich einen Hasen 2(b) mit Subsurface Scattering Effekten, gerendert mit dem Monte Carlo Pathtracer dieser Diplomarbeit.



**Abbildung 1:** Beispiel für Subsurface Scattering. Scattering-example.jpg entnommen aus [http://en.wikipedia.org/wiki/Subsurface\\_scattering](http://en.wikipedia.org/wiki/Subsurface_scattering) am 14.10.2008



**Abbildung 2:** Vergleich zwischen einfacher Beleuchtung und Subsurface Scattering. (a) einfache Phong Beleuchtung ohne Subsurface Scattering, (b) Subsurface Scattering gerendert mit dem Monte Carlo Pathtracer dieser Diplomarbeit

Es gab bereits einige Versuche die BSSRDF anzunähern und somit Subsurface Scattering darzustellen. Doch diese Versuche beschränkten sich bisher größtenteils auf medizinisch wissenschaftliche Zwecke, wie den Weg eines Lasers durch die Haut zu verfolgen. Solche Anwendungsgebiete benötigen keine hohe Performanz und legen großen Wert auf Korrektheit. Renderzeiten von mehreren Minuten bis Stunden sind akzeptabel.

Ziel dieser Diplomarbeit wurde es somit einen Shader zu entwerfen und zu implementieren, der Subsurface Scattering realisiert und dabei möglichst korrekte Ergebnisse in möglichst geringer Zeit liefert. Sollte dies nicht möglich sein, sollte er durch einen einfachen Benutzerparameter zwischen den beiden Extremen korrekte Darstellung und Performanz wählen können.



## 2 Grundlagen

In diesem Kapitel werden die Grundlagen, die zum Verständnis des Phänomens Subsurface Scattering und der verwendeten Verfahren des Raytracings benötigt werden, erklärt. Die Themen werden hier nur im benötigten Zusammenhang erläutert, die Erklärungen sind also nicht vollständig. Zur Vertiefung der Themen sollten die jeweils genannten Quellen zu Rate gezogen werden. Die behandelten Themen sind:

- **Das Phänomen des Subsurface Scattering**  
Wie der Effekt entsteht und wie er funktioniert.
- **Homogene und nicht-homogene Materialien**  
Welche Materialien sich für das Subsurface Scattering eignen.
- **Renderingverfahren**  
Erklärung der gängigsten Renderingverfahren und ihr Nutzen für den Subsurface Scattering Shader.
- **Beleuchtungsmodelle**  
Erklärung der gängigsten Beleuchtungsmodelle und bisher verwendeter Verfahren.
- **Der Monte Carlo Pathtracer**  
Aufbau und Funktion des Monte Carlo Pathtracers.
- **Strahlverfolgung**  
Die wichtigsten Gesetze der Strahlverfolgung.
- **Raytracer Beschleunigungsverfahren**  
Möglichkeiten einen Raytracer zu beschleunigen und ihre mögliche Anwendungen für den Subsurface Scattering Shader.
- **Der Raytracer Augenblick**  
Die verwendete Raytracer Engine.

### 2.1 Das Phänomen des Subsurface Scattering

Subsurface Scattering hat einen deutlichen größeren Einfluss auf die Wahrnehmung als der erste Blick vermuten lässt. Subsurface Scattering tritt sowohl bei Flüssigkeiten wie Milch auf, als auch bei farbigem Glas und lichtdurchlässigem Gestein wie Marmor und Jade. Doch auch die menschliche Haut und Autolack erlangen ihr Leuchten und ihre wahrgenommene Oberfläche durch dieses Phänomen.

Wie aus den oben genannten Beispielen bereits deutlich wird, tritt Subsurface Scattering auf, sobald Licht in die Oberfläche eines Materials eindringen kann. Solche lichtdurchlässige Materialien werden als transluzent

bezeichnet. Trifft ein Lichtstrahl auf eine lichtdurchlässige Oberfläche wird ein Teil des Lichtes absorbiert, reflektiert oder eben refraktiert, das heißt ein Teil des Lichtstrahls wird gebrochen und dringt weiter in das Material ein. Dort kann der refraktierte Lichtstrahl ebenfalls wieder gebrochen, reflektiert und refraktiert werden, bis er das Material wieder verlässt oder vollständig in dessen Innerem absorbiert wird.

Durch diese mehrfachen Reflexionen und Refraktionen verlässt der Lichtstrahl also das Material an einem anderen Ort als er es betreten hat. Dadurch kann Licht um Ecken und, je nach Dicke und Art des Materials, sogar durch das komplette Objekt transportiert werden. Seiten eines Modells, die andernfalls verschattet wären, können jetzt von innen heraus angeleuchtet werden. Direkt beleuchtete Seiten wirken weicher, da Schatten und Lichtpunkte durch die interne Streuung weichgezeichnet werden.

Subsurface Scattering tritt bei nahezu jedem Material auf. Abgesehen von Metallen verfügt jedes Material über eine Oberfläche, in die das Licht eindringen kann. Je tiefer das Licht vordringt, desto deutlicher wird der Effekt des Subsurface Scattering und desto schwieriger wird dessen Darstellung. Bei geringer Tiefe ist der Effekt kaum wahrnehmbar und die BRDF reicht aus um das Material darzustellen. Doch selbst viele solcher Materialien, die auf den ersten Blick nicht transluzent erscheinen, wirken mit der einfachen BRDF künstlich und falsch.

Bei transluzenten Materialien muss zwischen den homogenen und den nicht-homogenen Materialien unterschieden werden, da diese grundlegenden Unterschiede in der Darstellung und Realisierung des Subsurface Scattering besitzen.

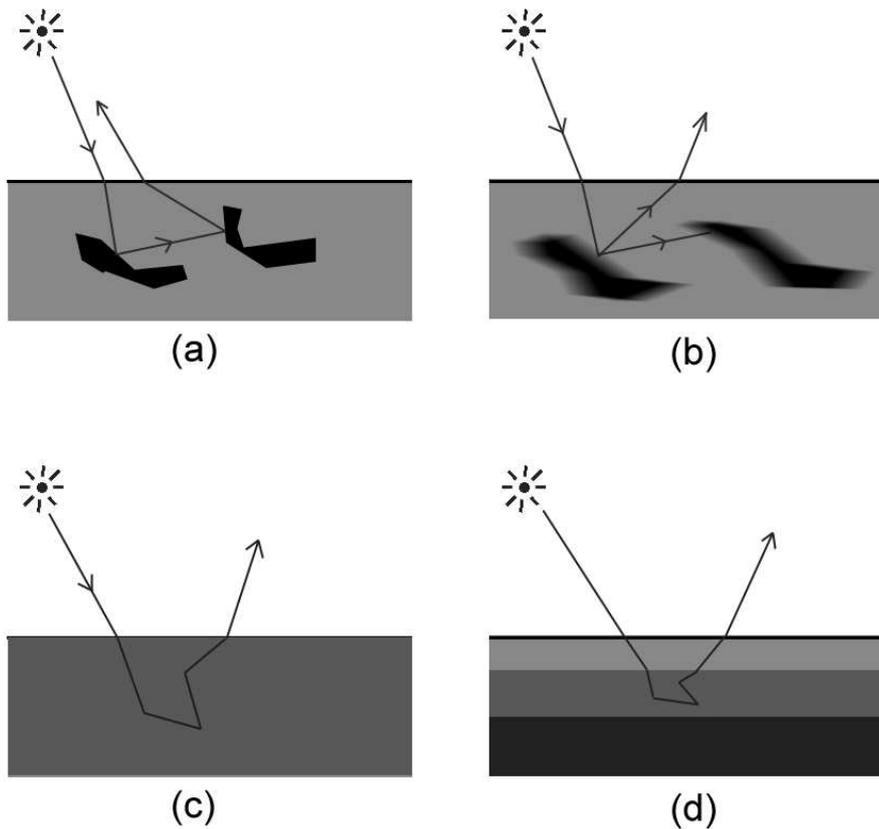
## 2.2 Homogene und nicht-homogene Materialien

Homogene Materialien sind solche deren Inneres homogen ist. Also jeder Ort innerhalb des Materials weist dieselben Eigenschaften auf, wie zum Beispiel Dichte und Brechzahl. Praktisch heißt das, dass die Stoffe aus denen das Material gegebenenfalls besteht, so vermischt sind, dass sie nicht mehr unterschieden werden können und als ein neues Ganzes anzusehen sind (siehe Abbildung 3(c)). Ein anschauliches Beispiel ist Kakao. Wenn das Kakaopulver vollständig in der Milch aufgelöst ist, kann man die zwei Stoffe nicht mehr voneinander unterscheiden.

Nicht-homogene Materialien sind also solche, die aus einer Mischung verschiedener Stoffe bestehen. Für das Subsurface Scattering muss hierbei zwischen zwei Arten der Mischung unterschieden werden:

- Unvollständige Vermischung. Die beiden Stoffe vermischen sich nicht komplett. Einer der beiden Stoffe hat sich nicht vollständig aufgelöst.

- Mischung auf granularer Ebene. Mindestens einer der beiden Stoffe besteht aus groben Teilchen die zwar untergemischt eine neue Menge ergeben, aber sich noch eben in diese Teilchen unterteilen lassen.



**Abbildung 3:** Materialmischungen in verschiedenen Zuständen: (a) granulare Einschlüsse, (b) Einschlüsse teilweise vermischt zu Schlieren, (c) Homogene Mischung, (d) Schichten mit verschiedenen Eigenschaften

Eine Mischung auf granularer Ebene ist zum Beispiel Lack mit glitzernen Splittern. Zwar lassen sich die Splitter bei genauem Hinsehen noch erkennen und auch von dem sie umgebenden Lack unterscheiden, doch der Lack erhält durch sie ein ganz anderes Erscheinungsbild (siehe Abbildung 3(a)). Das Licht wird innerhalb der Lackschicht zusätzlich durch diese Splitter gestreut, es findet ein anderer Lichttransport innerhalb der Lackschicht statt als er ohne die Splitter stattfinden würde.

Ein Beispiel für eine unvollständige Vermischung ist Marmor. Er besitzt die typischen grauen Schlieren, Einschlüsse in dem eigentlichen Gestein.

Diese Einschlüsse können nicht klar von dem eigentlichen Gestein getrennt werden (siehe Abbildung 3(b)).

An diesen Einschlüssen, beziehungsweise an den Übergängen zu solchen Einschlüssen, können einfallende Lichtstrahlen wieder reflektiert und absorbiert werden und beim Refraktieren sogar in ein anderes Medium mit vollkommen anderen Eigenschaften wechseln. Nicht-homogene Materialien im Zusammenhang des Subsurface Scattering sind auch solche, die aus mehreren Schichten bestehen (siehe Abbildung 3(d)). Diese Schichten verschiedener Materialien liegen so nahe beieinander, dass sie als Summe und somit scheinbar als ein Material wirken. Haut zum Beispiel besteht aus mehreren Schichten die sehr genau voneinander getrennt sind, vom menschlichen Auge aber als eine Schicht wahrgenommen werden. Ein Lichtstrahl, der in die Haut eindringt, wird in den oberen Schichten gebrochen, in den unteren färbt er sich durch die feinen Blutgefäße, die er durchdringt, rot und ergibt so das typische weiche Rosa.

Solche Materialien lassen sich nur durch die tatsächliche Modellierung von Schichten, beziehungsweise Einschlüssen durch homogene Materialien darstellen. Im Falle von unvollständiger Vermischung ist eine Modellierung nahezu unmöglich. Eine Lösung ist die Darstellung solcher Materialien mit Hilfe von Texturen. Die einzelne Textur enthält ein typisches Bild der Oberfläche mit den Einschlüssen. Diese wird über das Objekt gelegt und mit der normalen Beleuchtung für das Objekt verrechnet. Um Haut zu realisieren werden sogar Texturen für jede Schicht der Haut benötigt. Die Ergebnisse sind schon sehr überzeugend, aber natürlich nicht physikalisch korrekt.

Im Rahmen dieser Diplomarbeit werden nur homogene Materialien realisiert, doch dazu später mehr im Kapitel 4.3 *Simulation von Material*.

Um Subsurface Scattering zu realisieren, muss nun ein Blick auf die verschiedenen Arten der Darstellung geworfen werden, um zu entscheiden was möglich ist und was nicht.

## 2.3 Renderingverfahren

Als Rendering wird das Darstellen einer modellierten Szene auf dem Bildschirm bezeichnet. Es gibt zwei gebräuchliche Verfahren: das Scanline Rendering und das Raytracing. Jedes für sich besitzt verschiedene Vorteile, Nachteile und Möglichkeiten. Im Folgenden werden die Verfahren und ihr Nutzen für den Subsurface Scattering Shader behandelt.

### 2.3.1 Scanline Rendering

Das Scanline Rendering ist ein Verfahren zur Darstellung einer beliebigen Szene auf dem Bildschirm [3,23]. Objekte der Szene werden durch Projektion parallel auf eine virtuelle Bildebene projiziert, diese wird dann geclippt,

dass heißt überstehende Polygone werden zurechtgeschnitten und dann auf den Bildschirm übertragen. Im Falle einer perspektivischen Projektion werden alle Objekte durch eine Projektionsmatrix so verzerrt, dass sie aus Sicht der Kamera, und somit des Betrachters, perspektivisch verzerrt aussehen und dann auf die Bildebene projiziert. Dieses Verfahren ist sehr schnell und wird durch spezielle Hardware der Grafikkarten unterstützt. Dadurch findet Scanline Rendering seine Anwendung größtenteils in Bereichen in denen Echtzeit Rendering benötigt wird, wie zum Beispiel in der Spieleindustrie. Das Scanline Rendering kann jedoch viele Lichteffekte nicht realisieren oder nur über Umwege erreichen:

- Schatten können nicht direkt dargestellt werden. Es gibt jedoch Verfahren wie Shadowvolumes (vorgestellt von Franklin C. Crow, siehe [5]) und Shadowmapping (vorgestellt von Lance Williams, siehe [22]) die in der Lage sind Schatten zu simulieren. Jedes dieser Verfahren hat jedoch seine Nachteile und beide müssen zusätzlich das Anti-aliasing der Schattenkanten behandeln.
- Direkte Beleuchtung und speziell Reflexionen sind nicht möglich. Da es sich bei dem Scanline Rendering nur um eine Projektion handelt, kann der Blick in einen Spiegel nicht verfolgt werden. Es gibt einfache Lösungen, wie zum Beispiel das Erstellen einer spiegelverkehrten Szene hinter einer transparenten Wand, die dann als "Spiegel" dient. Doch bei komplexeren Szenen ist dieses Verfahren nicht anwendbar.
- Mehrfacher Lichttransport durch Brechungen und Reflexionen, sogenannte Kaustiken, sind nicht möglich. Kaustiken treten oft bei gläsernen oder hochgradig spiegelnden Objekten auf, wie zum Beispiel einem Cognac Glas.

All diese Effekte können im besten Fall angenähert werden, doch die korrekte Darstellung kann nur mittels eines Raytracers erfolgen. Die Verwendung eines Scanline Renderers kam somit für diese Diplomarbeit nicht in Frage.

### 2.3.2 Raytracing

Im Gegensatz zum einfachen Scanline Rendering, das die Szene auf den sichtbaren Bildschirm projiziert, werden beim Raytracing [8] Sichtstrahlen vom Betrachter aus durch das virtuelle Pixelraster des Bildschirms in die Szene verschossen. Diese Strahlen werden dann rekursiv in der Szene weiterverfolgt und können wie Licht gebrochen und absorbiert werden, bis ein bestimmtes Abbruchkriterium erreicht wird. Gängige Abbruchkriterien sind das Erreichen einer gegebenen Rekursionstiefe oder wenn der Anteil, den der Strahl noch zur Pixelfarbe beiträgt, unter einen festgelegten minimalen Sollwert sinkt. Ein Raytracer kann somit zum Beispiel vollständige

Spiegelungen und teilweise-Reflexionen darstellen, zu denen ein Scanline Renderer nicht in der Lage ist. Das Modell des Raytracers ist zudem deutlich einfacher zu verstehen und ermöglicht physikalisch korrekte Darstellung der Beleuchtung.

Der große Nachteil des Raytracings ist der deutlich höhere Rechenaufwand verglichen mit dem des Scanline Renderings. Mit steigender Anzahl an Dreiecken verschwindet der Nachteil des Rechenaufwands gegenüber dem Scanline Rendering jedoch komplett. Beim Raytracing wächst der Aufwand nicht linear mit der Anzahl der Dreiecke, wie im Falle des Scanline Renderings, sondern logarithmisch. Raytracing kann somit komplexe und größere Szenen schneller und realistischer darstellen.

Implementierungen und Verfahren des Raytracings wurden über die letzten Jahre stark verbessert und auch das Aufkommen erster dedizierter Raytracing Hardware macht Raytracing allmählich zu einem echten Konkurrenten für das Scanline Rendering. In Bereichen, in denen die Bilder nicht in Echtzeit benötigt werden, wie in der Autoindustrie, werden bereits größtenteils Raytracer eingesetzt.

Ein Raytracer ist zwar sehr mächtig, doch für das Darstellen von Subsurface Scattering Effekten konnte er in dieser Form nicht verwendet werden. Um das zu ermöglichen, musste der Raytracer zu einem Monte Carlo Pathtracer erweitert werden (siehe 4.1 *Die Wahl des Pathtracers*).

Die folgenden Kapitel sollen einen Überblick ermöglichen und ein Grundverständnis der verwendeten Verfahren aufbauen. Erklärt werden:

- die gängigen Beleuchtungsverfahren und Modelle
- die Monte Carlo Integration und ihr Einsatz im Monte Carlo Pathtracer
- die Gesetze der Strahlverfolgung

## 2.4 Beleuchtungsmodelle

Ein Beleuchtungsmodell beschreibt wie eine Fläche, beziehungsweise ein Punkt, innerhalb der Szene beleuchtet wird. Je nachdem welcher Grad an Genauigkeit und Realitätsnähe benötigt wird, können diese Modelle einfach bis sehr komplex werden. In den folgenden Unterkapiteln werden das korrekte Modell, die BRDF, und zwei ihrer Annäherungen, das Lambert und das Phong Beleuchtungsmodell, vorgestellt.

### 2.4.1 Die BRDF - Bidirectional Reflectance Distribution Function

Die BRDF wurde erstmals von Nicodemus [15] vorgestellt und beschreibt die Reflexionseigenschaften eines Materials. Sobald ein Sehstrahl einen Punkt auf der Oberfläche eines Objektes getroffen hat, muss dieser beleuchtet

werden. Die Frage, die es zu lösen gilt, ist: Wieviel Licht sieht der Betrachter an dieser Stelle? Bekannt sind die Blickrichtung, das Material das beleuchtet wird und die Lichtquellenposition sowie ihre Helligkeit. Was nun benötigt wird ist eine Funktion die abhängig von diesen bekannten Werten eine Helligkeit bestimmt.

Diese Funktion ist die sogenannte Bidirectional Reflectance Distribution Function, oder auch BRDF. Das einfallende Licht wird beschrieben durch die Beleuchtungsstärke  $E = \frac{d\Phi}{dA_e}$ , wobei  $\Phi_{in}$  der einfallende Lichtstrom ist und  $A_e$  die Empfängerfläche. Im Falle eines Punktes ist die Empfängerfläche infinitesimal klein. Das ausfallende Licht wird beschrieben durch die Radiosity  $B = \frac{d\Phi}{dA_s}$ . Hier ist  $\Phi_{out}$  der abgegebene Lichtstrom und  $A_s$  die Senderfläche. Die BRDF beschreibt das Verhältnis zwischen der Beleuchtungsstärke und der Radiosity. Abbildung 4(a) zeigt eine typische BRDF für die wahrgenommene Helligkeit.

Seien:  $\omega_{in}$  und  $\omega_{out}$  Ein- und Ausfallswinkel

Dann gilt:

$$f_r(d\vec{\omega}_{in}, d\vec{\omega}_{out}) = \frac{dR(d\vec{\omega}_{out})}{dB(d\vec{\omega}_{in})} \quad (1)$$

Abhängig von den gewünschten Eigenschaften besitzt sie verschiedene Dimensionen:

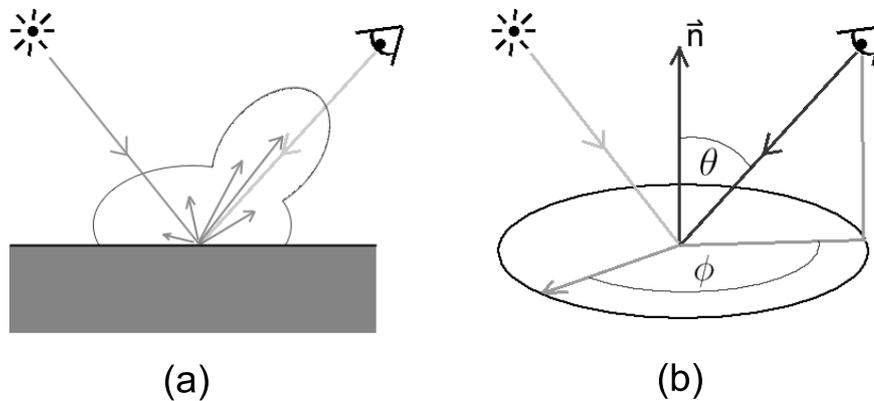
- 4 Dimensionen für alle Ein- und Ausfallswinkel
- 5 Dimensionen wenn sie zusätzlich für jede Wellenlänge gilt
- 7 Dimensionen für zusätzlich jeden Ort der Oberfläche
- 9 Dimensionen für zusätzlich Ein- und Austrittspunkte

Die BRDF ist rotationssymmetrisch für isotrope Materialien, solche, die unabhängig von der Richtung aus der sie betrachtet werden immer gleich aussehen. Das heißt relevant ist nur der Winkel  $\theta$  zwischen der Blickrichtung und der Flächennormalen.

Bei anisotropen Materialien, also solchen die abhängig von der Richtung aus der sie betrachtet werden andere Eigenschaften aufweisen, wie zum Beispiel Seide, ist die BRDF zusätzlich abhängig vom Winkel  $\phi$  (siehe Abbildung 4(b)).

Wenn Ein- und Austrittspunkt verschieden sind, spricht man auch von der Bidirectional Scattering Surface Reflectance Distribution Function. Diese wird für Subsurface Scattering Phänomene benötigt.

Die BRDF ist sehr komplex und kann mittels Messungen für ein Material bestimmt werden. Jedoch ist der Datensatz enorm groß und somit nur



**Abbildung 4:** (a) typische BRDF, (b)  $\theta$  und  $\phi$  für den Betrachter

schwer einsetzbar, das heißt es werden Annäherungen benötigt. Beispiele für solche Annäherungen sind zum Beispiel das Phong Beleuchtungsmodell (siehe 2.4.3 *Das Phong Beleuchtungsmodell*) und die Cook-Torrance BRDF. Die typische Annäherung einer BRDF beinhaltet drei Komponenten:

- Die **Distribution**, die Verteilung der Mikrofacetten auf der Oberfläche.
- Die **Geometric Attenuation**, die Selbstverschattung. Je nach Lage der Mikrofacetten und Beschaffenheit der Oberfläche kann eine Fläche sich selbst verschatten beziehungsweise beleuchten.
- Den **Fresnel Term**. Er gibt an welcher Anteil refraktiert und welcher Anteil reflektiert wird.

In den folgenden Unterkapiteln werden nun zwei der gängigen Beleuchtungsmodelle vorgestellt. Beide sind Versuche die BRDF mit jeweils verschiedener Genauigkeit anzunähern.

## 2.4.2 Das Lambert Beleuchtungsmodell

Das Lambert Beleuchtungsmodell ist ein einfaches Beleuchtungsmodell das eine Fläche abhängig von ihrer Neigung zur Lichtquelle beleuchtet. Hierzu wird der Skalarwert zwischen der Flächennormalen und der Verbindung des beleuchteten Punktes mit der Lichtquelle berechnet und als Cosinus interpretiert. Werte gleich oder kleiner Null bedeuten dass die Fläche senkrecht zur Lichtquelle steht beziehungsweise abgewandt ist, also die Fläche nicht beleuchtet wird. Werte größer Null bedeuten, dass die Materialfarbe

mit diesem Cosinuswert gewichtet wird. Eine senkrecht beleuchtete Fläche hat mit dem Faktor eins entsprechend ihre maximale Helligkeit.

Das Lambert Beleuchtungsmodell setzt die Annahme voraus, dass die Welt diffus ist. Ausserdem kann es keine direkte Beleuchtung darstellen, dazu wird die Erweiterung zum Phong Beleuchtungsmodell benötigt.

### 2.4.3 Das Phong Beleuchtungsmodell

Das Lambert Beleuchtungsmodell beinhaltet nur die einfache indirekte Beleuchtung. Das Licht, das abhängig von der Position des Betrachters ins Auge einfällt wird nicht berücksichtigt. Dieses Licht wird im Allgemeinen als spekulares Licht, auch Highlights genannt, wahrgenommen. Das Phong Beleuchtungsmodell, vorgestellt von Bui Tuong Phong [18], teilt das Licht in den diffusen und spekularen Anteil auf. Der diffuse Anteil wird entsprechend des Lambert Beleuchtungsmodells beleuchtet. Für den spekularen Anteil wird der ideal reflektierte Sichtstrahl berechnet und mit der Verbindung von Punkt zu Lichtquelle skalar verrechnet. Wie auch bei dem Lambert Beleuchtungsmodell wird der Skalarwert als Cosinus interpretiert und mit einer materialabhängigen Glanzzahl  $n$  potenziert. Die Glanzzahl beschreibt die Beschaffenheit der Oberfläche. Kleine Werte stehen für eine raue Oberfläche, Werte gegen unendlich beschreiben einen perfekten Spiegel. Auch hier gilt wieder: je mehr der ausfallende Sichtstrahl sich zur Lichtquelle neigt, desto heller wird der Punkt beleuchtet. Beide Farbwerte werden dann mit ihrem jeweiligen Anteil gewichtet und aufsummiert die Gesamthelligkeit des Punktes. Das Phong Beleuchtungsmodell ist nur für hohe Glanzzahlen physikalisch korrekt.

Sei:

- $I_a$  das ambiente Licht
- $I_{in}$  das direkte einfallende Licht
- $k_{ambient}$  der ambiente Term des Materials
- $k_{diffus}$  der diffuse Term des Materials
- $k_{specular}$  der reflektierende Term des Materials
- $n$  die Glanzzahl
- $\theta$  der Einfallswinkel zur Oberflächennormalen
- $\phi$  der Winkel zwischen idealer Reflexion der Blickrichtung des Betrachters und der direkten Verbindung von Punktlichtquelle zum beleuchteten Punkt

Dann gilt:

$$I_{out} = I_a * k_{ambient} + I_{in} * (k_{diffus} * \cos\theta + k_{specular} * \cos^n\phi) \quad (2)$$

mit:

$$k_{ambient} + k_{diffus} + k_{specular} = 1 \quad (3)$$

## 2.5 Der Monte Carlo Pathtracer

Wie auch die vorher genannten Verfahren dient der Monte Carlo Pathtracer zur Annäherung der BRDF und zur Darstellung der Szene. Er macht sich hierbei folgende Verfahren zu Nutze:

- Die Monte Carlo Integration, um die BRDF anzunähern.
- Den Pathtracer, um die Szene effizient rendern zu können.

Beide werden in den folgenden Unterkapiteln genauer erläutert.

### 2.5.1 Die Monte Carlo Intergration

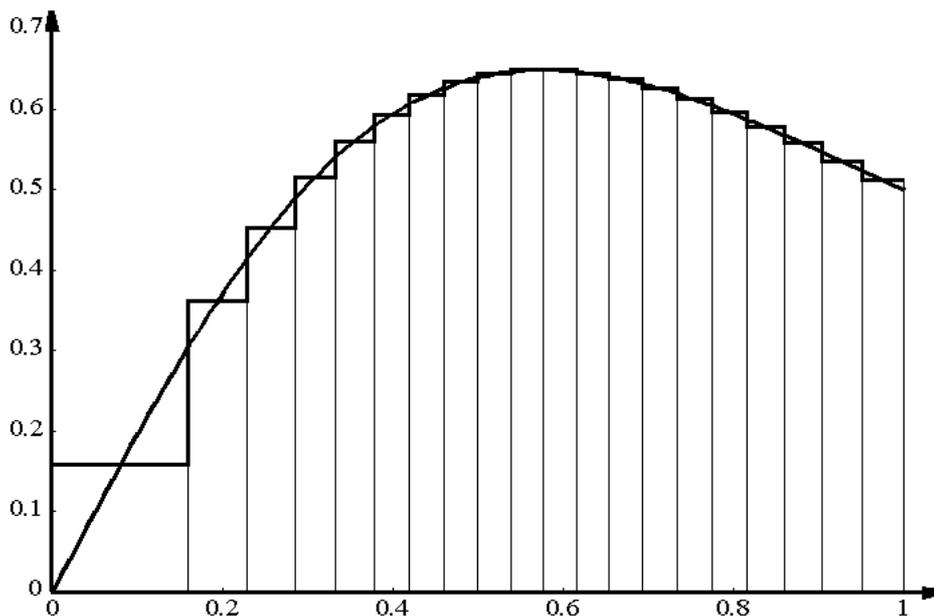
Das Integral einer beliebigen Funktion zu bestimmen kann mitunter sehr aufwendig und rechenintensiv sein. Die Idee der Monte Carlo Integration ist es den Wert des Integrals als Erwartungswert zu interpretieren und diesen Erwartungswert mittels Stichproben, auch Samples genannt, anzunähern (siehe Gleichung 4). Sei:

- $f(x)$  die Funktion die innerhalb des Intervalls von  $a$  bis  $b$  integriert werden soll
- $N$  die Anzahl der Stichproben
- $x_i$  eine Zufallszahl zwischen  $[0, 1]$
- $p_{[x]}$  die beliebige Dichtefunktion, die die Eigenschaften einer Dichte erfüllt

Dann gilt:

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

Es wird eine beliebige Anzahl von  $N$  Samples mit einer beliebigen Dichte  $p$  erzeugt und deren Ergebnisse aufsummiert und gemittelt. Einzige Bedingung ist, dass diese Dichte auch die Eigenschaften einer Dichte erfüllen muss, das heißt alle Werte der Dichte müssen positiv sein und kein Wert darf innerhalb des zu integrierenden Intervalls null sein. Je geschickter die Dichte gewählt ist, desto schneller konvergiert die Integration zu dem korrekten Wert hin. Gefunden werden kann diese Dichte durch das sogenannte Importance Sampling, das heißt in Bereichen mit hohen Werten werden mehr Samples erzeugt. Ideal ist eine Dichte ähnlich der Funktion von der das Integral bestimmt werden soll. Wichtig ist auch eine ausreichende Anzahl von Samples zu erzeugen. Ansonsten reichen sie unter Umständen nicht aus um das Integral anzunähern und die Ergebnisse werden stark verrauscht. Bemerkenswert ist, dass durch das Gesetz der großen Zahlen letztendlich jede gültige Dichte in der entsprechenden Zeit zum Ergebnis führt. Abbildung 5 zeigt ein Integral das mit einer Dichte  $p(x)$  angenähert wird. Die Abweichung vom korrekten Wert wäre in diesem Idealfall Null.

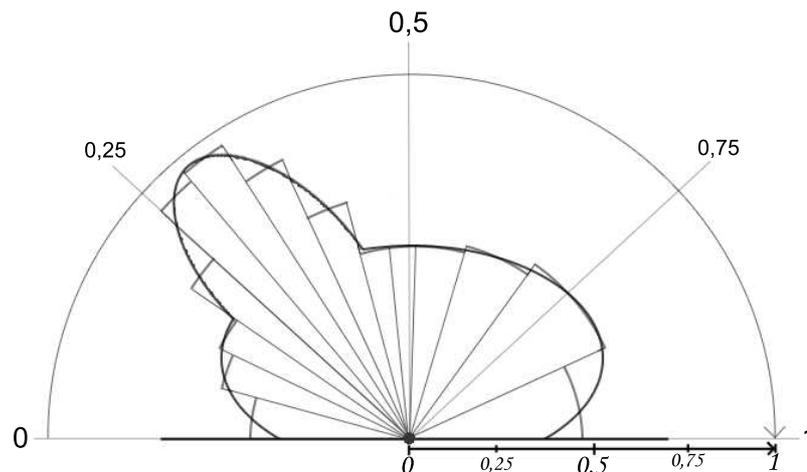


**Abbildung 5:** Annäherung eines Integrals durch die Monte Carlo Iteration mit einer Dichtefunktion  $p(x) \approx f(x)$ . Die Abweichung ist hier gleich Null.

Die Monte Carlo Integration wird in der Computergrafik benutzt um Formfaktoren und auch die BRDF zu bestimmen [13,14]. Man spricht dabei auch von stochastischem Raytracing. Das Integral, das es zu lösen gilt, ist

im Falle der BRDF die Beleuchtungskurve des jeweiligen Materials. Um es korrekt zu lösen müssten unendlich viele Strahlen in den oberen Halbraum des Materials verschossen werden damit jede Stelle des Integrals abgetastet wird. Bedenkt man zudem, dass diese Strahlen wieder gebrochen und gespiegelt werden können, wird klar, dass der Aufwand unverhältnismäßig und nicht praktikabel ist. Um das Problem dennoch in endlicher Zeit lösen zu können verwendet man die Monte Carlo Integration. Die Stichproben sind hierbei mit der entsprechenden Dichte  $p$  in den oberen Halbraum verschossene Strahlen. Ihr Ergebnis wird entsprechend aufaddiert und gemittelt und als Farbe eingetragen.

Abbildung 6 zeigt eine Skizze der Annäherung einer BRDF durch die Monte Carlo Integration. Die Dichtefunktion  $p(x)$  gibt an in welche Winkel die Samples bevorzugt verschossen werden. Der Winkel ergibt sich aus  $p(x) * 180$ . Die normierte Leuchtdichte ist kursiv angetragen und entspricht bei den Samples dem Radius des jeweiligen Bogenabschnitts. Auch hier werden anhand des Importance Samplings mehr Samples in Bereiche mit höheren Werten verschendet.



**Abbildung 6:** Annäherung einer BRDF durch die Monte Carlo Integration

Um die Monte Carlo Integration in einen Pathtracer zu integrieren wird jedoch ein weiteres Verfahren benötigt: Das sogenannte Russische Roulette.

### 2.5.2 Russisches Roulette

Das sogenannte Russische Roulette ist ein Verfahren, das bei Pathtracing verwendet wird. Mit Hilfe dieses Verfahrens wird entschieden welcher mögliche Pfad des Pathtracers zum Weiterverfolgen ausgewählt wird. Die Wahrscheinlichkeiten sind hierbei jedoch nicht gleichverteilt. Jeder Pfad hat sei-

ne eigene Wahrscheinlichkeit ausgewählt zu werden, die nicht unbedingt gleich der Wahrscheinlichkeit der anderen Pfade ist. Im Falle eines Pathtracers werden diese Wahrscheinlichkeiten durch das Material festgelegt.

Sei:

- $k_d$  der diffuse Anteil
- $k_r$  der reflektierte Anteil
- $k_t$  der transmissions Anteil

Dann gilt für die Zufallszahl  $\xi$ :

- $\xi \in [0, k_d] \rightarrow \text{diffus}$
- $\xi \in [k_d, k_d + k_r] \rightarrow \text{reflexion}$
- $\xi \in [k_d + k_r, k_d + k_r + k_t] \rightarrow \text{transmission}$

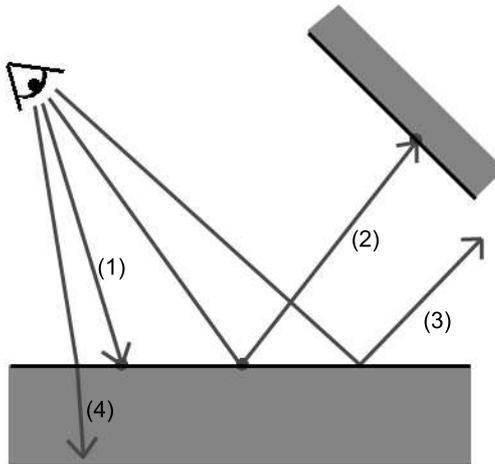
Wobei  $k_d + k_r + k_t \leq 1$  erfüllt sein muss.

### 2.5.3 Der Pathtracer

Raytracing ist speziell bei großen Rekursionstiefen mit einem hohen Rechen- und Speicheraufwand verbunden. Strahlen die auf eine transparente oder transluzente Oberfläche treffen werden entsprechend der Fresnel Gleichung (siehe Kapitel 2.6.2 *Der Fresnel Term*) sowohl zu einem Teil reflektiert und zu einem anderen Teil gebrochen. Je mehr gebrochene, beziehungsweise reflektierte, Strahlen erzeugt werden, desto mehr Strahlen müssen auch verfolgt und berechnet werden. Zudem ist die Chance auf diesem Weg vom Betrachter aus eine Lichtquelle zu treffen nahezu Null.

Ein Ansatz dieses Problem zu lösen ist das sogenannte Pathtracing und stammt von James T. Kajiya [12]. Die Idee ist mehrere Strahlen durch ein Pixel zu verschießen, aber immer nur einen Strahl der möglichen Nachfolgerstrahlen zu verfolgen und die Ergebnisbilder zu mitteln. Um sicher zu gehen, dass die Lichtquelle getroffen wird, versendet der Pathtracer einen Schattenfühler in die Richtung der Lichtquelle, der überprüft ob der getroffene Punkt von der Lichtquelle direkt beleuchtet wird oder verschattet ist. Die möglichen Pfade, englisch *path*, die ein Strahl nehmen kann sind:

- (1) Er wird nicht weiterverfolgt und der Punkt diffus beleuchtet.
- (2) Der Punkt wird beleuchtet und der Strahl diffus weiterverfolgt.
- (3) Er wird reflektiert.
- (4) Er wird refraktiert.



**Abbildung 7:** Darstellung der möglichen Pfade eines Pathtracers

Je nachdem welche Funktion der Pathtracer erfüllen soll, können diese Pfade erweitert werden (siehe 4.5 *Ablauf des Shaders im Detail*).

Welcher Strahl weiterverfolgt wird, wird per Russisch Roulette (siehe 2.5.2) entschieden und mit Hilfe des Monte Carlo zufällig in eine der Dichte entsprechenden Richtung verschossen. Die Chancen, dass ein Pfad gewählt wird, entspricht dem Anteil mit dem der entsprechende Strahl bei einem einfachen Raytracer eingerechnet werden würde. Ist ein Material zum Beispiel zu 50% reflektierend wird der Pfad "Reflektion" mit einer Wahrscheinlichkeit von 50% ausgewählt. Auf diese Weise werden die diffusen, reflektierten und refraktierten Anteile korrekt eingerechnet. Der Vorteil ist hier, dass man bei deutlich weniger Rechenaufwand schneller ein Bild erhält. Der Nachteil ist, dass die frühen Bilder stark verrauscht sind und sich erst nach recht vielen Durchläufen, etwa 10.000, ein rauschfreies Bild ergibt.

Durch das Monte Carlo Raytracing ist der Pathtracer jedoch in der Lage schneller ein Bild zu liefern, das deutlich näher am korrekten Ergebnis liegt als es ein einfacher Raytracer könnte. Mit der entsprechenden Dichte lässt sich somit die BRDF eines Materials sehr gut und schnell annähern.

Für den Subsurface Scattering Shader kam nur ein Pathtracer in Frage da durch die zahlreichen Brechungen und Reflexions Ereignisse sehr viele Strahlen verfolgt und berechnet werden müssen (siehe 4.1 Die Wahl des Pathtracers).

## 2.6 Strahlverfolgung

Die Sichtstrahlen eines Raytracers können exakt wie Lichtstrahlen innerhalb einer Szene an Oberflächen gebrochen und gespiegelt werden. Dabei folgen sie zwei großen Gesetzmäßigkeiten:

- Snell's Brechungsgesetz. Es beschreibt wie ein Strahl bei dem Übergang von einem Medium in ein anderes gebrochen wird.
- Dem Fresnel Term. Mit Hilfe des Fresnel Terms lässt sich bestimmen, welcher Anteil des Lichtstrahl gebrochen und welcher gespiegelt wird.

Beide Gesetzmäßigkeiten geben an, in welchem Winkel zum Lot der Strahl die Oberfläche wieder verlässt. Um diese Drehung des Strahls zu realisieren, wurde die Axis-Angle Rotation verwendet.

### 2.6.1 Snell's Brechungsgesetz

Trifft ein Lichtstrahl auf eine Oberfläche im Winkel  $\theta_1$  zum Lot der Oberfläche, wird er abhängig von der Brechzahl  $n_1$  des Mediums aus dem er kommt, und deren Brechzahl  $n_2$  in das er geht entsprechend Snell's Gesetz gebrochen und verlässt die Oberfläche auf der anderen Seite des Lots im Winkel  $\theta_2$ .

$$\frac{\sin(\theta_1)}{\sin(\theta_2)} = \frac{n_1}{n_2} = n \quad (5)$$

Im klassischen Modell von Snell befinden sich die gebrochenen beziehungsweise reflektierten Strahlen auf derselben Ebene wie das Lot der Oberfläche und der Eintrittsstrahl.

### 2.6.2 Der Fresnel Term

Trifft Licht auf eine Grenzfläche zwischen zwei Medien mit verschiedenen Brechzahlen so wird ein Teil des Lichts reflektiert und ein anderer Teil refraktiert. Man spricht hier von sogenannter partieller Reflexion. Der Fresnel Term gibt an, welcher Anteil des Lichts abhängig von den Brechzahlen der beiden Medien und dem Auftreffwinkel reflektiert, beziehungsweise refraktiert wird.

In dieser Diplomarbeit wird die Annäherung nach Schlick [20] verwendet. Sei  $\theta$  der Winkel in dem der Strahl die Oberfläche relativ zur Flächennormalen trifft, und  $n_1$  und  $n_2$  die Brechzahlen für die beiden Medien, dann

gilt:

$$f_r(\theta) \approx f_0 + (1 - f_0)(1 - \cos(\theta))^5 \quad (6)$$

mit:

$$f_0 = \left(\frac{n_1 - n_2}{n_1 + n_2}\right)^2 \quad (7)$$

### 2.6.3 Rotationen mit Hilfe der Axis-Angle Rotation

Da Raytracing allgemein und speziell für das Subsurface Scattering sehr viele Reflexionen und Brechungen handhaben muss, war es um so wichtiger das Drehen und neu Ausrichten der Strahlvektoren zum Weiterverfolgen genauer zu betrachten.

Es gibt zahlreiche Methoden einen Vektor im dreidimensionalen Raum zu rotieren. Die klassischste Darstellung ist wohl die Rotation um die drei Achsen des Karthesischen Koordinatensystems. Dieses lässt sich erweitern zu der sogenannten Axis-Angle Rotation. Gegeben ist ein beliebig im Raum liegender normierter Vektor, der als Rotationsachse dient und ein Winkel  $\phi$  um den der Vektor rotiert werden soll. Diese Art der Rotation hat den Vorteil dass sie nicht in den Gimbal-Lock geraten kann, das heißt, dass sich mehrere Rotationen gegenseitig aufheben und so ein Freiheitsgrad verloren gehen würde. In der Matrix-Form lässt sich die Axis-Angle Rotation sehr einfach und effizient auf die Vektoren anwenden. Sei:

- $c = \cos(\phi)$
- $s = \sin(\phi)$
- $t = 1 - c$
- $[x, y, z]$  die normierte Rotationsachse

Dann ergibt sich die Matrixform für die Axis-Angle Rotation:

$$[R] = \begin{pmatrix} t * x * x + c & t * x * y - z * s & t * x * z + y * s \\ t * x * y + z * s & t * y * y + c & t * y * z - x * s \\ t * x * z - y * s & t * y * z + x * s & t * z * z + c \end{pmatrix} \quad (8)$$

Da die Strahlen des Ratraycers bei Spiegelungen und Brechungen immer relativ zur Flächennormalen rotiert werden müssen war die Axis-Angle Rotation hier die ideale Wahl.

## 2.7 Raytracer Beschleunigungsverfahren

Ein Raytracer benötigt sehr viele Berechnungen für die Strahlen- und Schnitttests. Einfach programmierte Raytracer besitzen Renderzeiten von mehreren Minuten. Um dennoch auf gute Bildraten, größer als ein Frame pro Sekunde, zu kommen, ist es wichtig durch Beschleunigungsverfahren die Rechenzeit soweit wie möglich zu reduzieren. Abgesehen von effizienten Algorithmen für die Schnitttests lohnt sich vor allem die Parallelität sowohl von Daten als auch von Prozessen auszunutzen. Gerade bei hochgradig kohärenten Strahlen kann man davon ausgehen, dass sie dasselbe Objekt treffen werden und so Strahlen zu Bündeln zusammenfassen beziehungsweise mehrere Strahlenbündel parallel berechnen lassen.

### 2.7.1 SIMD - Single Instruction Multiple Data

Single Instruction Multiple Data ist ein Verfahren, bei dem eine Operation auf bis zu vier Datensätzen gleichzeitig ausgeführt wird. Im Falle des Raytracings wird die Kohärenz der Strahlen ausgenutzt. Werden viele Strahlen in dieselbe Richtung verschossen, ist die Wahrscheinlichkeit sehr hoch, dass viele bis alle von ihnen dasselbe Objekt treffen, beziehungsweise sie einen ähnlichen Weg durch die Szene nehmen werden. Durch SIMD können Berechnungen der Strahlvektoren einfach auf alle vier Strahlen angewendet und teure Materialabfragen reduziert werden. Im Idealfall erfolgt nur noch eine Abfrage der Objektdaten pro Strahlenbündel.

SIMD konnte für den eigentlichen Subsurface Scattering Shader nicht verwendet werden. (siehe 4.2 *SIMD für Subsurface Scattering*)

### 2.7.2 Multi-Threading

Als Multi-Threading bezeichnet man das gleichzeitige Ausführen mehrerer paralleler Prozesse beziehungsweise mehrerer Programme auf einem Rechner. Dies kann auch schon auf einem Prozessor geschehen, soweit möglich werden diese Prozesse jedoch auf verschiedene Prozessorkerne aufgeteilt. Jeder Kern bekommt dann einen gerade aktiven Prozess zugewiesen. Diese Parallelisierung ermöglicht eine große Beschleunigung bei Programmen, die viele kleinere Prozesse benötigen, beziehungsweise sich in viele Teilschritte aufteilen lassen.

### 2.7.3 Boundingvolumes

Ein weiterer Ansatz zum Beschleunigen eines Raytracers ist es, die zahlreichen Schnitttests zu reduzieren. Hierzu werden Objekte und Mengen von Objekten rekursiv zu sogenannten Boundingvolumes zusammengefasst und bekommen jeweils eine umschließende Geometrie zugewiesen. Schnitttests finden dann zuerst mit der Boundingvolume Hierarchie statt

und danach mit den beinhalteten Objekten. Der Aufbau der Hierarchie ist hierbei sehr entscheidend. Da der Strahl sich durch die Szene bewegt, sollten sich Boundingvolumes nach lokaler Nähe und Zugehörigkeit der Objekte richten. Wie bereits erwähnt stellen Schnitttests mit NURBS Flächen eine große Herausforderung dar, da der Schnittpunkt der Strahlen und der NURBS Kurven sehr viel Rechenaufwand kostet. Der in dieser Diplomarbeit verwendete Raytracer *Augenblick* unterteilt die NURBS Flächen ebenfalls wieder durch Boundingvolumes, um den Schnitttest auf Segmente der NURBS Flächen, jeweils definiert durch die zugehörigen baryzentrischen Koordinaten  $u$  und  $v$ , zu reduzieren.

## 2.8 Der Raytracer *Augenblick*

Der Raytracer *Augenblick* ist ein von Oliver Abert an der Universität Koblenz-Landau entwickelter Raytracer. Er erreicht mittels SIMD, Multi-Threading und spezieller adaptiver Bounding Boxes echtzeitfähige Frameraten. Der Raytracer *Augenblick* war anfangs darauf ausgelegt NURBS Modelle in Echtzeit darstellen zu können. Inzwischen ist der Raytracer *Augenblick* zu einem vollwertigen Raytracer erweitert worden. Er ermöglicht und unterstützt eine einfache Einbindung und Implementierung von Plugins.

Da der Aufwand einen eigenen Raytracer zusätzlich zu dem Subsurface Scattering Shader zu schreiben zu groß war, sollte das Shader Plugin für einen vorhandenen Raytracer geschrieben werden. Aufgrund der einfachen Anbindung und der vorort Unterstützung wurde der Raytracer *Augenblick* ausgewählt.

### 2.8.1 Die Schnittstelle zum Raytracer *Augenblick*

Es ist wichtig den Ablauf und die Schnittstelle zwischen dem Shader und dem Raytracer *Augenblick* zu kennen, um zu verstehen, wie das Subsurface Scattering arbeitet und über welche Möglichkeiten beziehungsweise welche Einschränkungen der Shader verfügt.

Der Raytracer *Augenblick* lädt die Szene und die beinhalteten Objekte. Jedes dieser Objekte bekommt einen Shader zugewiesen. Sobald ein Bild gerendert wird, verschickt der Raytracer Primärstrahlen vom Betrachter aus durch die Pixel in die Szene. Sobald ein solcher Primärstrahl ein Objekt trifft, wird der dem Objekt zugewiesene Shader aufgerufen um diesen Auftreffpunkt zu beleuchten. Aufgabe des Shaders ist es also nicht nur den Anteil des Subsurface Scattering darzustellen, sondern auch die normale Beleuchtung des Punktes. Der Subsurface Scattering Shader hat Zugriff auf die Szene und kann weitere Strahlen versenden. Nachdem er den Punkt komplett beleuchtet hat, gibt er den gefundenen Farbwert an den Raytracer *Augenblick* zurück, der ihn dann auf dem Bildschirm entsprechend darstellt.

Ein Pathtracer benötigt mehrere Rendering Durchläufe um ein rauschfreies Bild zu erzeugen. Da der Subsurface Scattering Shader nur pro Strahl aufgerufen wird, hat er weder Informationen über vorangegangene Ergebnisse für das aktuelle Pixel, noch im wievielten Durchlauf er sich befindet. Der Raytracer *Augenblick* verfügt jedoch über eine Funktion namens "use idle time of processor". Ist diese Funktion aktiv, werden weitere Renderschritte ausgeführt, wenn der Prozessor nicht vollständig ausgelastet ist. Die neuen Bilder dieser zusätzlichen Renderschritte werden dann mit dem vorhandenen gerade gezeigten Bild verrechnet und neu dargestellt. Der Subsurface Scattering Shader führt also immer nur einen Durchlauf des Pathtracers durch.

Daraus ergeben sich folgende Eigenschaften und Aufgaben für den Subsurface Scattering Shader:

- Der Shader muss den Punkt, für den er aufgerufen wird, vollständig beleuchten. Er kann sich nicht nur auf das Subsurface Scattering beschränken.
- Er hat Zugriff auf die Szene, kann Objekte und deren Eigenschaften abrufen. Sollte ein Strahl kein Objekt treffen, wird der entsprechende Farbwert aus der Environment Box zurückgegeben. Ist keine Box vorhanden, gilt der Default-Wert Weiß.
- Er kann neue Strahlen erzeugen und versenden, also selbst Funktionalitäten eines Raytracers übernehmen. Jeder Strahl beinhaltet einen Wert Lambda, der multipliziert mit dem Richtungsvektor und anschließend auf den Startpunktvektor aufaddiert den Auftreffpunkt des Strahls wiedergibt. Ist der Richtungsvektor normiert, gibt Lambda die Länge des Strahls wieder.
- Er hat kein Wissen über das dargestellte Bild, umliegende Strahlen und Pixel. Sein Wissen über den Rendervorgang beschränkt sich auf den Punkt und den zugehörigen Primärstrahl.
- Er hat vorerst keinen Einfluss auf das Verrechnen der einzelnen Renderframes.

### 2.8.2 Plattform und Entwicklungsumgebung

Vorgegeben durch den Raytracer *Augenblick* musste die Implementierung und die Tests auf einem Macintosh Rechner unter MAC OS X und der Verwendung der Programmiersprache C++ durchgeführt werden. Verwendet wurde ein Intel Xeon 2\*3GHz QuadCore mit 2048MB RAM um die Möglichkeiten des Raytracers *Augenblick* voll ausnutzen zu können. Als Entwicklungsumgebung wurde Xcode von Apple gewählt.



### 3 Vorangegangene Arbeiten

Nahezu jede BRDF ist letztendlich nur eine Annäherung an die jeweilige BSSRDF des Materials. Es gibt nur wenige Materialien in die das Licht nicht eindringen kann. Die BSSRDF eines Materials lässt sich korrekt und vollständig berechnen doch der Aufwand ist enorm. In der Vergangenheit gab es bereits einige vorangegangene Arbeiten, die das Phänomen des Sub-surface Scatterings und somit die BSSRDF annähernten.

Eine erste Annäherung war das Modell von Hanrahan und Kruger [9], das der BRDF einen analytischen Ausdruck für einen einfachen Streueffekt innerhalb eines homogenen gleichmäßig beleuchteten Quaders hinzufügte. Doch berücksichtigte dieses Modell, wie auch viele Folgende, nicht, dass das Licht das Material an anderer Stelle verlässt als es das Material betreten hat. Zwar lässt sich auch diese Tatsache mit berechnen und mittels der Strahlungsübertragungsgleichung von Chandrasekhar [4] komplett lösen, doch auch hier ist der Aufwand enorm und dementsprechend zeitintensiv.

Ein anderer Lösungsansatz war es die BSSRDF mit Hilfe der Monte Carlo Integration [13, 14] anzunähern und stammt von Matt Pharr und Pat Hanrahan [17]. Strahlen die in ein Material eindringen wurden entsprechend der gewichteten Zufallszahl der Monte Carlo Integration in Richtung Lichtquelle weiter versendet. Auch hier ergaben sich schon sehr gute Bilder, doch die Strahlen drangen nur in eine sehr geringe Tiefe ein und durchdrangen das Objekt nie.

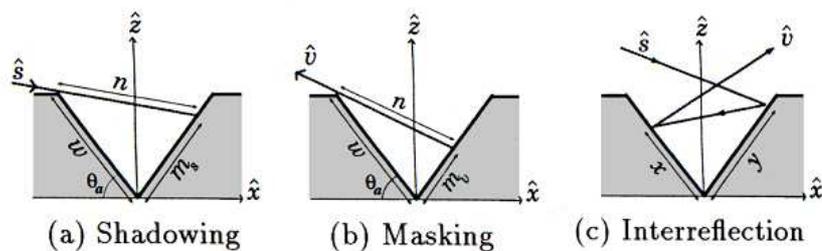
Ein weiterer Ansatz ist es Photonmapping zur Simulation des Lichttransportes in und durch Materialien zu benutzen, vorgestellt von Henrik Wann Jensen [10]. In einem ersten Durchlauf werden von der Lichtquelle aus Photonen entsprechend der Lichtverteilungskurve in die Szene versendet und in einer dreidimensionalen Photonmap gespeichert. Ein Raytracer kann in einem zweiten Durchlauf dann diese Photonen einsammeln und den Punkt mittels der im Material liegenden Photonen beleuchten. Dieses Verfahren liefert sehr gute Bilder und kann zusätzlich diffuse Interreflektionen, sogenanntes Colorbleeding, darstellen. Mit Hilfe einer Causticmap werden sogar Kaustiken, Lichttransport über mehrere spiegelnde Flächen, möglich. Nachteil ist hierbei, dass das Photonmapping einen szenenabhängigen zusätzlichen Speicherplatz für die Photonmap benötigt und sehr viele Photonen verschießen muss, um die Szene ausreichend auszuleuchten. Rendering in Echtzeit ist hier nicht möglich.

Stam [19] gelang es die Streuung innerhalb des Materials mittels einer Diffus-Funktion anzunähern. Durch Anwendung einer Multigrid konnte er diese Funktion annähern und so Wolken mit interner Lichtstreuung darstellen.

Dieses Verfahren erzielte zwar die gewünschten Effekte und Ergebnisse, benötigte aber auch deutlich mehr Rechenaufwand und Speicherplatz als es diffuse Lichtmodelle benötigen. Gerade bei hochgradig streuenden Materialien wie Milch, in denen das Licht oft mehrere hundert male gestreut wird ehe es das Material wieder verlassen kann, erwiesen sich derartige Verfahren als unpraktikabel.

Neben den Versuchen die komplette BSSRDF anzunähern, beschränkten sich einige Verfahren darauf lediglich die Effekte von transluzenten Oberflächen zu visualisieren. Durch diese Vereinfachungen konnten raue Oberflächen dargestellt werden [21] wie zum Beispiel Staub [1, 2]. Ein echter Lichttransport durch Objekte konnte hier jedoch nicht stattfinden.

Subsurface Scattering ist ein besonderes Problem wenn es um die Darstellung von Haut geht. Oren und Nayar [16] entwarfen eine BRDF für die Darstellung von Haut und fügten einen Term für die Selbstverschattung hinzu. Selbstverschattung entstand durch die raue Oberfläche des Materials. Oren und Nayar nahmen an, dass V-förmige Schluchten die Oberfläche durchzogen (siehe Abbildung 8). Diese V-Schluchten konnten sich nun selbst verschatten 8(a), das abgestrahlte Licht maskieren 8(b) und sich durch Interreflexionen selbst beleuchten 8(c). Sie benutzten einen Monte Carlo Pathtracer um ihren Shader umzusetzen. Dieses Modell lieferte schon sehr realistische Haut, war aber auf Texturen angewiesen um die einzelnen Hautschichten zu visualisieren.

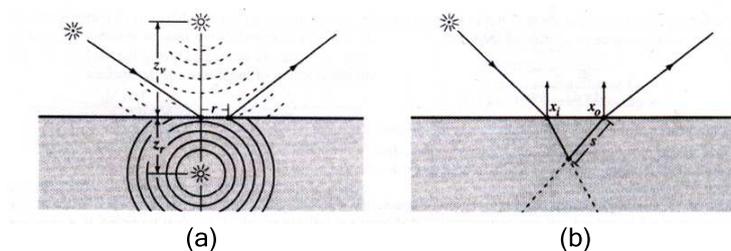


**Abbildung 8:** Darstellung der V-förmigen Schluchten auf der Oberfläche eines Objekts. Entnommen aus [16].

### A Practical Model for Subsurface Light Transport

Das von Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy und Pat Hanrahan vorgestellte "A Practical Model for Subsurface Light Transport" [11] sei hier gesondert erwähnt, da es im späteren Verlauf dieser Arbeit für Vergleiche und Beurteilung der Ergebnisse genutzt wird.

Henrik Wann Jensen et. al. benutzten die von Eason [6] und Farrel et al. [7] vorgestellte Idee einer Dipol Lichtquelle um den Lichttransport eines einfach gestreuten Strahls durch das Material anzunähern.



**Abbildung 9:** (a) Darstellung der Dipol-Lichtquelle, (b) vereinfachter Strahlengang. Entnommen aus [11].

Ein Strahl, der in das Material eindrang wurde in eine Dipol Lichtquelle umgewandelt um die Beleuchtung des Austrittspunktes anzunähern (siehe Abbildung 9). Sie gingen dabei davon aus das nur ein Streuungsereignis innerhalb des Materials zustande kommt und vereinfachten so den Weg des Strahls.

Die vorgestellten Ergebnisse erreichten bereits gute Renderzeiten und sehr realistische Bilder (siehe Abbildung 10). Dennoch wurde auch hier nicht der tatsächliche Weg der Lichtstrahlen durch das Material berücksichtigt.



**Abbildung 10:** Die Büste der Diana, gerendert mit dem Shader von Henrik Wann Jensen et. al. [11].



## 4 Der Subsurface Scattering Shader

Durch den Raytracer *Augenblick* war bereits ein Raytracer geschrieben der Objekte in Echtzeit rendern konnte. Dadurch ergaben sich zwei Ziele für diese Diplomarbeit:

- Den Subsurface Scattering Shader weg von den Vereinfachungen wieder hin zu mehr Realismus zu bringen. Die Strahlen sollten tatsächlich innerhalb des Materials in verschiedenen Tiefen gestreut werden können und so Effekte ermöglichen, die durch die Vereinfachungen nicht möglich waren. Die NURBS Objekte ermöglichen zudem ein korrektes Simulieren der Objektoberfläche, ohne dass die Aufteilung des Objekts in Dreiecke durch Interpolationen der Normalen die Werte verfälscht.
- Subsurface Scattering ist trotz aller Vereinfachungen immer noch aufwendig und physikalisch korrekte Bilder in Echtzeit sind immer noch undenkbar. Doch abgesehen von Anwendungsbereichen wie der Medizin benötigen Computerspiele zum Beispiel keine physikalische Korrektheit, es ist ausreichend, wenn der Effekt physikalisch plausibel ist und für das Auge realistisch erscheint. Da das Raytracing durch bessere Verfahren, Implementationen und in der Entwicklung befindliche dedizierte Hardware in den nächsten Jahren immer interessanter für Spielehersteller wird liegt es nahe eben einen solchen Raytracing Shader zu entwerfen der in Echtzeit plausible Subsurface Scattering Effekte darstellen kann.

Als Erstes wurde ein korrekter Subsurface Scattering Shader für den Raytracer *Augenblick* ohne Rücksicht auf Rendering Zeiten erstellt. Er sollte sowohl einen tieferen Blick in die Materie erlauben als auch Grundlage, Referenz und Vergleichsmöglichkeit bieten für später erstellte Beschleunigungs- und Vereinfachungsverfahren. Durch den Raytracer *Augenblick* konnte bereits Multi-Threading verwendet werden.

Die folgenden Kapitel geben einen genauen Einblick in die verwendeten Verfahren und den Aufbau des korrekten Subsurface Scattering Shaders:

- Die Wahl des Pathtracers
- SIMD für Subsurface Scattering
- Simulation von Material
- Geeignete Modelle für das Subsurface Scattering
- Ablauf des Shaders im Detail: der einfache Pathtracer

- Ablauf des Shaders im Detail: das Subsurface Scattering
- Subsurface Phong Beleuchtung

#### 4.1 Die Wahl des Pathtracers

Subsurface Scattering erfordert, dass Strahlen weit in ein Material hinein verfolgt werden und innerhalb des Materials gegebenenfalls neue Strahlen erzeugt werden müssen. Gerade in hochgradig streuenden Materialien wie Milch würde der Rechenaufwand sehr schnell überhand nehmen. Deshalb fiel die Wahl auf einen Pathtracer. Er ermöglicht die Darstellung selbst höchst komplexer Strahlverfolgungen und liefert sehr früh Ergebnisse. Während der Implementierung wurden verschiedene Versionen des Pathtracers entwickelt und getestet.

#### 4.2 SIMD für Subsurface Scattering

Der Raytracer *Augenblick* ist in der Lage mit Hilfe von SIMD Pakete von bis zu vier Strahlen durch die Szene zu verfolgen. Alle Primärstrahlen, die der *Augenblick* verschießt, sind in solchen vierer Strahlenpaketen zusammengefasst. Für die einzelnen Strahlen wird dann jeweils der Subsurface Scattering Shader aufgerufen. Die Verwendung von SIMD innerhalb des Subsurface Scattering Shaders war leider nicht möglich. Durch das Pathtracing wird immer nur ein Strahl weiterverfolgt und durch das Monte Carlo Raytracing in eine gewichtet zufällige Richtung verschossen. Die Sekundärstrahlen werden somit weit gestreut und verlieren jegliche Kohärenz der Primärstrahlen und der Einsatz von SIMD ist nicht mehr effizient möglich.

#### 4.3 Simulation von Material

Die große Herausforderung ist es, Material zu simulieren wo keines existiert. Modelle sind innen hohl. Ihnen kann zwar ein Material zugewiesen werden, aber ein Strahl kann nur anhand der Oberflächennormale erkennen, ob er ein Objekt betritt oder verlässt. Insofern das Modell korrekt modelliert wurde, gibt es im Inneren auch keinerlei Geometrie. Vorhandene Geometrie würde sogar Fehler hervorrufen, da die Strahlen einen Materialwechsel finden würden wo keiner sein dürfte. Ohne gegebene Geometrie ist es schwer Einschlüsse, wie sie bei Marmor vorhanden sind, zu simulieren. Sie sind konkrete Geometrie innerhalb des Materials das ausmodelliert werden müsste. Zudem besitzen die Einschlüsse nicht unbedingt eine scharfe Abgrenzung zum restlichen Material. Der Übergang kann auch verwischt sein und die Materialien sich innerhalb eines Bereichs halb durchdringen. Dies ist nur äußerst schwer zu modellieren. Benötigt würden Volumendatensätze, die den Rechenaufwand extrem erhöhen würden und bei

dem Raytracer *Augenblick* nicht unterstützt werden. Der Subsurface Scattering Shader in dieser Diplomarbeit unterstützt deshalb nur homogene Materialien.

#### 4.4 Geeignete Modelle für das Subsurface Scattering

Abgesehen von der Modellierung der Einschlüsse stellte sich ein weiteres Problem: Viele gängige Modelle sehen von außen plausibel aus, sind aber nicht real möglich. Ein Beispiel ist der Stanford Terracotta Hase. Das ursprüngliche Modell war ein hohler Terracotta Hase, das Modell ist jedoch nur eine Hülle. Diese Hülle besitzt Löcher an der Unterseite, die auch das Original besitzt, und unerwünschte Löcher am Brustbereich. Auf den ersten Blick mag es plausibel erscheinen, der Hase besitzt Aussparungen. Doch die Wände des Hasen haben keinerlei Dimension. Sie sind lediglich Dreiecke deren Normalen nach außen zeigen. Ein Blick in diese Löcher zeigt selbst bei einfachen Beleuchtungsmodellen Fehlbeleuchtungen, der Hase ist innen beleuchtet obwohl kein Licht in ihn dringt.

Für das Subsurface Scattering stellen solche Modelle ein Problem dar. Der Shader verfolgt Strahlen durch ein Objekt. Diese Strahlen werden innerhalb des Objektes gestreut und verlassen es gegebenenfalls wieder. Im Fall des Hasen ist das Modell für den Shader also innen massiv und ohne Hohlraum. Schießt ein Strahl jedoch durch eines der Löcher, lässt sich nicht mehr feststellen wann das Modell betreten beziehungsweise verlassen wurde.

Hieraus lassen sich folgende Voraussetzungen für Modelle schliessen, die für das Subsurface Scattering geeignet sind:

- Die Modelle dürfen keinerlei interne Geometrie besitzen, ansonsten werden Strahlen innerhalb falsch gebrochen.
- Die Modelle sollten geschlossen sein. Der Shader kann sonst nicht entscheiden, wann ein Strahl das Modell verlassen beziehungsweise betreten hat.

Bei dem Testen des Shader an verschiedenen Modellen ergab sich jedoch ein gewisser Toleranzbereich in dem der fertige korrekte Subsurface Scattering Shader nicht geschlossene Modelle ausreichend korrekt darstellen kann (siehe Kapitel 8 *Diskussion*).

- Die Modelle sollten mit der entsprechenden Genauigkeit modelliert worden sein. Die meisten Modelle sind durch eine endliche Anzahl von Dreiecken beschrieben. Je mehr Dreiecke verwendet wurden, desto besser sieht das Modell aus und desto genauer und realistischer sind die Ergebnisse des Subsurface Scatterings.

Diese Voraussetzungen sind wünschenswert und ein Modell, das sie erfüllt lässt sich fehlerlos beleuchten. In der Praxis genügt aber bei weitem nicht jedes Modell, sei es durch Fehler oder nicht entsprechend sorgfältiges Modellieren, diesen Voraussetzungen.

## 4.5 Ablauf des Shaders im Detail: der einfache Pathtracer

Der Raytracer *Augenblick* verschießt die Primärstrahlen in einem SIMD Paket aus vier Strahlen. Sobald einer dieser Strahlen auf ein Objekt trifft wird der Subsurface Scattering Shader aufgerufen. Dieser verwendet das Russisch Roulette um auszuwählen wie er weiter verfährt. Die Wahrscheinlichkeiten mit denen das Russisch Roulette einen Pfad auswählt sind abhängig von den Eigenschaften des getroffenen Materials und dem zugehörigen Fresnel Term. Die vier möglichen Pfade sind hier der Übersichtlichkeit halber noch einmal aufgezählt (siehe Abbildung 7):

- (1) Diffus beleuchten
- (2) Diffus weiterverfolgen
- (3) Reflektieren
- (4) Refraktieren

Die ersten drei Pfade sind unverändert die des einfachen Pathtracers und werden im Folgenden erläutert. Das Refraktieren wird separat erklärt, da hier das Subsurface Scattering angewendet wird, welches deutlich aufwendiger ist als die Standardrefraktion.

### 4.5.1 Diffus beleuchten

Der einfachste Fall: Der getroffene Punkt wird gemäß des Phong Beleuchtungsmodells beleuchtet und die Farbe zurückgegeben.

Die Formel für die Phongbeleuchtung wird erweitert um:

- $\vec{n}$  die Flächennormale
- $\vec{b}$  die Blickrichtung des Betrachters
- $\overrightarrow{b_{reflec}}$  die perfekte Reflexion der Blickrichtung des Betrachters
- $\vec{l}$  der Vektor vom Punkt zur Punktlichtquelle

Somit ergibt sich:

$$I_{out} = I_a * k_{ambient} + I_{in} * (k_{diffus} * (\vec{b} \cdot \vec{n}) + k_{specular} * (\overrightarrow{b_{reflec}} \cdot \vec{l})^n)$$

### 4.5.2 Diffus weiterverfolgen

Der getroffene Punkt wird gemäß des Phong Beleuchtungsmodells beleuchtet. Dann wird ein weiterer Strahl erzeugt, der reflektiert und rekursiv durch die Szene verfolgt wird. Die Rekursion endet sobald eine festgelegte Rekursionstiefe erreicht wurde oder der Anteil, den der Strahl noch beiträgt, unter einen festgelegten Schwellwert sinkt. Der Anteil, den ein Strahl zu dem ursprünglichen Punkt den der Primärstrahl getroffen hat beiträgt ist materialabhängig.

Dieses diffuse Weiterverfolgen erzeugt das sogenannte Color Bleeding. Werden zwei verschiedenfarbige Wände angeleuchtet, so beleuchten sie sich durch das indirekte Licht gegenseitig. Eine rote Wand würde also einen roten Schimmer auf die direkt anliegende Wand werfen. Die Farbe "blutet" auf die andere Wand.

### 4.5.3 Reflektieren

Der Strahl wird perfekt reflektiert, das heißt der Ausfallswinkel  $\theta_{outgoing}$ , ist gleich dem Einfallswinkel  $\theta_{incoming}$ , und weiter in die Szene verfolgt. Um die perfekte Reflexion zu erreichen wird die Richtung des Strahls invertiert und um  $180^\circ$  um die Normale des Auftreffpunktes rotiert. Die neu erzeugten reflektierten Strahlen werden rekursiv weiter durch die Szenen verfolgt bis eine festgelegte Rekursionstiefe erreicht ist. Diese Rekursionstiefe sollte Szenenabhängig gewählt werden. Bei komplexen Objekten mit sehr vielen spiegelnden Flächen muss die Rekursionstiefe sehr hoch gewählt werden da sehr viele Strahlen benötigt werden um komplett durch das Objekt zu reisen und eine diffuse Fläche zu treffen. Ein Beispiel für ein solches Objekt wäre ein Diamant. Wird die Rekursionstiefe zu gering gewählt bleiben einige Facetten des Diamants schwarz, da die Strahlen noch keine diffuse Fläche mit Farbinformationen gefunden haben.

## 4.6 Ablauf des Shaders im Detail: das Subsurface Scattering

Dieses Kapitel widmet sich dem genauen Aufbau und Ablauf des Subsurface Scattering Shaders. Die jeweils verwendeten Verfahren werden an geeigneter Stelle erläutert.

Die Refraktion eines Strahls bei einem Subsurface Scattering Shader benötigt deutlich mehr Aufwand als bei einem einfachen Pathtracer und wird deshalb hier in mehrere Schritte unterteilt und erklärt. Unterschieden wird zwischen den verschiedenen Reflektions- beziehungsweise Refraktionsergebnissen an den Übergängen zwischen den Medien und innerhalb des Objekts:

- Refraktion am Übergang Luft-Material
- Refraktion und Reflektion im Material
- Refraktion und Reflektion am Übergang Material-Luft
- Refraktion am Übergang Material-Material

#### 4.6.1 Refraktion am Übergang Luft-Material

Trifft ein Strahl auf ein Objekt in das er eindringen kann wird ein Teil des Strahls abhängig von dem jeweiligen Fresnel Term reflektiert. Der restliche Anteil des Strahls wird in das Material weiterverfolgt und zunächst an der Oberfläche des Materials refraktiert. Hierzu wird er anhand des Snelliusschen Brechungsgesetzes abhängig von den Brechzahlen  $n$  der Medien, durch die er wechselt, gebrochen. Diese Brechung findet immer relativ zum Lot, also der Senkrechten zur Fläche, statt.

$$\frac{\sin(\theta_1)}{\sin(\theta_2)} = \frac{n_1}{n_2}$$

Diese durch die Brechung hervorgerufene Rotation des Strahls erfolgt im Normalfall auf der Ebene, die durch Strahl und Flächennormale aufgespannt wird. Um diese Rotation im dreidimensionalen Raum zu realisieren wird das Kreuzprodukt des Strahlvektors und der Flächennormalen genommen und normalisiert als Rotationsachse interpretiert. Trifft ein Strahl senkrecht auf die Fläche ergibt das Kreuzprodukt kein sinnvolles Ergebnis. Da ein Strahl bei senkrechtem Einfall jedoch nicht gebrochen wird, wird dieser Fall separat abgefangen und der Strahl behält seine Ursprungsrichtung bei.

Die Flächen der Objekte sind jedoch nicht immer perfekt glatt. Sie können aufgeraut sein, das heißt über feinere Geometrie verfügen, die den Strahl in eine andere Richtung als vermutet bricht. Um dies zu simulieren wird der Strahl um einen zufälligen Winkel in eine zufällige Richtung von der idealen Brechung weggedreht. Dadurch kann es passieren, dass der Strahl das Material direkt wieder verlässt und stattdessen in eine zufällige Richtung reflektiert wird. Die Dichte der Zufallszahlen ist so gewählt, dass Strahlen bevorzugt der idealen Brechung folgen.

Mit dem ersten Eintauchen in das Material wird die Strecke gemessen, die der Strahl im Material zurücklegt. Mit zunehmender Strecke innerhalb des Materials verliert der Farbwert, den der Strahl sammelt, an Helligkeit da er vom Material geschluckt wird. Innerhalb eines Materials ist dieser Verlust logarithmisch. Die Dichte wird simuliert durch zwei Werte:

- Die maximale Eindringtiefe: Der Weg, den ein Strahl zurücklegen kann ehe er komplett absorbiert wird.
- Die maximale Refraktions- und Reflexionstiefe: Die Strecke innerhalb derer der Strahl auf jeden Fall refraktiert oder reflektiert wird. Mit dem Erreichen dieser Tiefe wird jeder Strahl gestreut. Die Strecke wird entlang des Strahls gemessen, nicht entlang des Lots.

Somit lässt sich das Verhalten der Strahlen innerhalb des Materials steuern. Genauer gesagt das Verhältnis der Länge von eintauchenden zu auftauchenden Strahlen. Dies ist wichtig für Materialien, die nicht oder nur selten durchdrungen werden. Je kleiner der Wert für Refraktion- und Reflexionstiefe wird, desto früher werden Strahlen wieder zur selben Seite zurückreflektiert. Die zurückgelegte Strecke ist kürzer und ihr Lichtanteil somit höher, der Pathtracer konvergiert schneller hin zur fertigen Farbe. Ist das Objekt an einer Stelle dünner wird es normal durchleuchtet.

#### 4.6.2 Reflektion und Refraktion im Material

Jedes Material weist eine bestimmte Dichte auf. Abhängig von dieser Dichte kann ein Strahl eine bestimmte Strecke in das Material eindringen und intern wieder reflektiert und refraktiert werden ehe er das Material verlässt oder vollständig absorbiert wird. Die Helligkeit des Lichtstrahls und somit der Beitrag des Sichtstrahls zu dem jeweiligen Punkt auf dem Objekt nehmen logarithmisch ab.

Dringt ein Lichtstrahl in ein transluzentes Medium ein, wird auf der ganzen Strecke, die er in das Medium eindringt, jeweils ein Teil durch interne Refraktion und Reflexion gestreut. Um zu simulieren, dass Strahlen nicht immer in derselben Tiefe gestreut werden, dringt ein Strahl eine zufällige Strecke in das Material ein. Sei  $s_{Strahl}$  die zufällig bestimmte Strecke, die der Strahl zurücklegt bevor er gestreut wird und  $m_{max}$  die maximale Refraktions- beziehungsweise Reflexionstiefe, so gilt:

$$0 \leq s_{Strahl} \leq m_{max}$$

#### 4.6.3 Refraktion und Reflexion am Übergang Material-Luft

Trifft ein von innerhalb des Materials kommender Strahl auf die Außenwand des Objektes, kann der Strahl wieder reflektiert, refraktiert oder absorbiert werden. Im Falle der Reflexion beziehungsweise Refraktion wird analog zum Übergang Luft-Material per Fresnel Term der jeweilige Anteil bestimmt und per Russisch Roulette ausgewählt.

Für den Fall, dass der Strahl das Objekt verlässt wird die Messung des Weges im Material hier ausgesetzt. Hier gilt die vereinfachende Annahme das Luft das Licht nicht abschwächt. Erst bei dem Eindringen in ein neues Objekt wird der Weg weiter aufaddiert und der Strahl entsprechend abgedunkelt.

Bei sehr großen Entfernungen entstehen hierbei Fehler, da Luft sehr wohl Einfluss auf Lichtstrahlen hat. Für gerenderte Szenen ist der Fehler jedoch vernachlässigbar da modellierte Szenen selten die entsprechenden Ausmaße annehmen um diesen Effekt sichtbar hervorzurufen. Meist wird eine Würfelgeometrie um die gesamte Szene herum generiert damit Strahlen des Raytracers nicht im Nichts enden können. Je nach Verwendung besitzt der Würfel dann eine Textur auf der Innenseite und dient als sogenannte Skybox, die die weit entfernte Umgebung simuliert. Eine Skybox kann zusammen auch mit einer Environmentmap verwendet werden. Trifft ein reflektierter oder refraktierter Strahl keines der modellierten Objekte, wird anhand der Betrachterposition und des Winkels zur getroffenen Fläche der Farbwert aus der Environmentmap ermittelt.

#### **4.6.4 Refraktion am Übergang Material-Material**

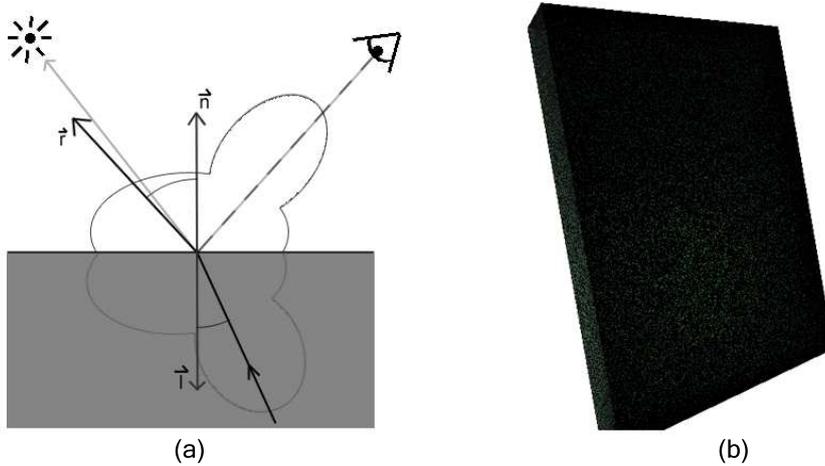
Da sich diese Diplomarbeit nur mit homogenen Materialien beschäftigt, wird der Übergang transluzentes Material in ein anders transluzentes Material nicht beachtet. Für nahe beieinander stehende transluzente Objekte benötigt das hier vorgestellte Verfahren also immer einen Mindestabstand zwischen den beiden, da die Brechungen sonst nicht korrekt ausgeführt werden können.

Dieser Fall kann jedoch auftreten wenn das Modell Löcher besitzt. Ein Strahl trifft das Objekt, dringt in das Material ein und verlässt das Objekt durch ein solches Loch. Jetzt kann er wieder eine Fläche treffen deren Normale zu ihm zeigt, also einen scheinbaren Einschluss im Objekt finden.

Bei Modellen mit Löchern in der Geometrie kann nicht zuverlässig entschieden werden wann ein Objekt betreten beziehungsweise verlassen wird. Eine Behandlung solcher Fälle ist jedoch für komplett geschlossene Modelle möglich (siehe *10 Ausblick*).

#### **4.7 Subsurface Phong Beleuchtung**

Nach eingehender Betrachtung stellte sich heraus, dass die Beleuchtung eines Punktes der Fläche, der von innen getroffen wurde, ebenfalls korrekt nach dem Phong Beleuchtungsmodell beleuchtet werden konnte. Der spekulare Wert ergibt sich hierbei einfach zwischen der gebrochenen Strahlrichtung und der Richtung vom Punkt zur jeweiligen Lichtquelle (Abbildung 11).



**Abbildung 11:** Darstellung der Subsurface Phong Beleuchtung

Der Effekt wird einleuchtend wenn ein entsprechender virtueller Betrachter auf der entgegengesetzten Seite der Oberfläche eingezeichnet wird. Der reflektierte Strahl des virtuellen Betrachters und der refraktierte Subsurface Strahl sind identisch. Wichtig ist hier, dass es sich bei diesem virtuellen Betrachter nicht um den an der Oberflächenlinie gespiegelten Subsurface Betrachter handelt. Durch die Refraktion sind für den Subsurface Betrachter die Winkel  $\theta_{incoming}$ , zwischen Blick und Lot  $\vec{l}$ , und  $\theta_{outgoing}$ , zwischen refraktiertem Vektor und Normale  $\vec{n}$ , nicht gleich. Auch die BRDF ist somit verzerrt (skizziert in Abbildung 11(a)).

Durch diesen Effekt ist es möglich eine Art Subsurface Highlights durch ein Material hindurch zu sehen. Dies wird besonders deutlich bei einem flachen Quader wie in Abbildung 11 (b) gezeigt. Die Lichtquelle befindet sich auf der anderen Seite des Quaders, dennoch ist in der Vorschau bereits ein deutliches Leuchten des Subsurface Highlights zu sehen.



## 5 Der Subsurface Scattering Shader: Tests und Möglichkeiten

Nachdem der Subsurface Scattering Shader geschrieben war stellte sich die Frage ob es möglich wäre ihn zu beschleunigen, beziehungsweise in seiner Funktionsweise zu vereinfachen. Zwar sollte er hauptsächlich als Referenzmodell dienen, doch wurden während der Implementierungsphase einige Ansätze für Varianten des Shaders und des Pathtracers aufgeworfen. Diese Ansätze wurden getestet und werden in den folgenden Kapiteln erläutert und bewertet:

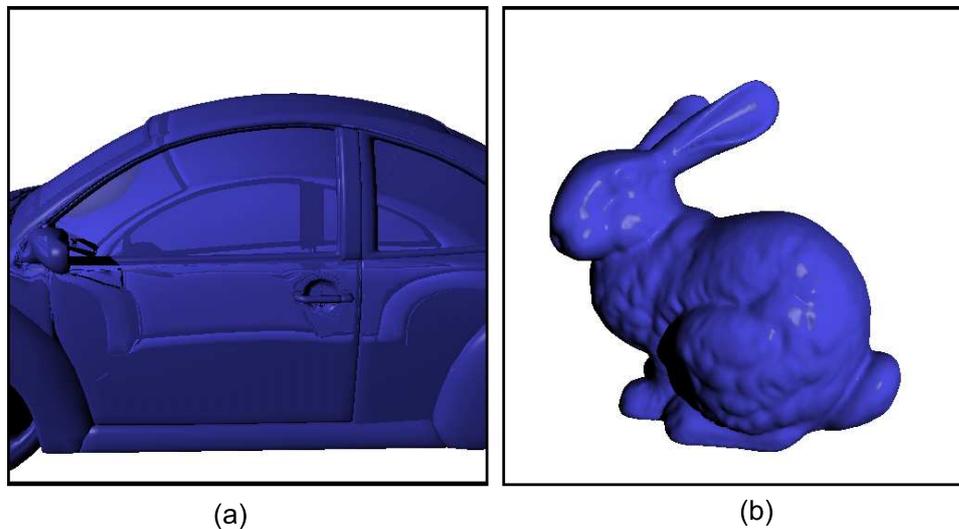
- **Monte Carlo Raytracing für das Subsurface Scattering**  
Das Monte Carlo Raytracing war ohne Frage immer noch sehr aufwendig. In diesem Abschnitt wird untersucht, ob Vereinfachungen des Monte Carlo Raytracing Verfahrens möglich wären.
- **Beschleunigung des Subsurface Scattering Shaders**  
Der Pathtracer benötigt immer eine feste Anzahl an Render-Durchläufen ehe er ein rauschfreies Bild liefern konnte. In diesem Abschnitt werden Ansätze zur Vereinfachung und Reduzierung dieser Durchläufe vorgestellt.

### 5.1 Monte Carlo Raytracing für das Subsurface Scattering

Anfangs wurde getestet wie der Pathtracer ohne das Monte Carlo Verfahren arbeitet und welchen Einfluss das Monte Carlo Verfahren auf das Raytracing hat. Dies diente dem Verständnis und sollte erste Möglichkeiten für Vereinfachungen und Beschleunigung aufwerfen.

#### 5.1.1 Raytracing ohne Monte Carlo

Komplett ohne das Monte Carlo Verfahren wurden die Strahlen ohne Abweichung gebrochen, drangen in das Material ein, wurden innerhalb des Materials reflektiert und lieferten die Helligkeit, die sie beim Austrittspunkt vorfanden. Wie in Abbildung 12 zu erkennen, waren die Strahlen so jedoch entschieden zu deterministisch und lieferten ein projiziertes Bild des Modells auf sich selbst, je nach Eindringtiefe entsprechend mehr verzerrt (Abbildung 12 (a)). Ohne zumindest ein Sampling mehrerer möglicher Richtungen war hier kein vernünftiges Bild möglich. Eine Anwendung bieten hier jedoch geringe Eindringtiefen. Das Modell wirkt, als läge ein dünner Film Wasser über ihm und spiegele das Modell in sich. Es sieht nass aus (siehe Abbildung 12 (b)).



**Abbildung 12:** Der Subsurface Scattering Shader ohne Monte Carlo Raytracing. (a) mit großer Eindringtiefe, (b) mit geringer Eindringtiefe

### 5.1.2 Monte Carlo nur für Refraktions- und Reflexionstiefe

Ohne Monte Carlo Raytracing ist Subsurface Scattering also nicht möglich. Die Lichtstrahlen dringen in das Material ein und werden entlang dieses Weges reflektiert und gebrochen. Das Monte Carlo Verfahren wurde verwendet um die Eindringtiefe zu variieren. Abbildung 13 zeigt ein Stanford Bunny das mit einem solchen Verfahren beleuchtet wurde. Der Lichttransport sieht besser aus, aber das Licht erinnert doch noch sehr an die Lichtbrechung von einfachem Glas und wirkt unecht.



**Abbildung 13:** Ein Terracotta Hase. Das Monte Carlo verfahren wurde hier nur auf die Eindringtiefe angewendet.

Erst wenn die Strahlen innerhalb des Modells zusätzlich in jede beliebige Richtung gebrochen werden können entstehen korrekte Subsurface Scattering Effekte. Hier ist zu beachten dass sich die refraktierten und reflektierten Strahlen innerhalb des Materials nicht mehr nur auf einer bevorzugten Ebene bewegen, wie zum Beispiel bei der Brechung am Übergang Luft-Material, sondern sich frei im Material verteilen können.

### 5.1.3 Monte Carlo Dichteverteilungen innerhalb des Materials

Der erste Ansatz war eine BRDF, beziehungsweise eine Dichteverteilung für das Innere eines Objektes zu finden. Es wurde mit verschiedenen Dichten getestet, ob ein Strahl bessere oder schlechtere Ergebnisse liefert und welche Zeit er für ein fertiges Bild benötigt. Die Tests ergaben jedoch, dass es gerade bei großen Eintauchtiefen zu viele Fälle gibt in denen vorgegebene Dichten falsche Ergebnisse liefern. Da die Umgebung unbekannt ist, kann der Subsurface Scattering Shader nicht entscheiden, wo er mehr Licht finden würde und wo nicht. Selbst das Wissen über Positionen der Lichtquellen helfen hier nicht weiter, da die Ausdehnung des Objektes unbekannt ist. Gerade durch das Subsurface Scattering kann das Licht für den Punkt im Material aus einer komplett anderen Richtung kommen.

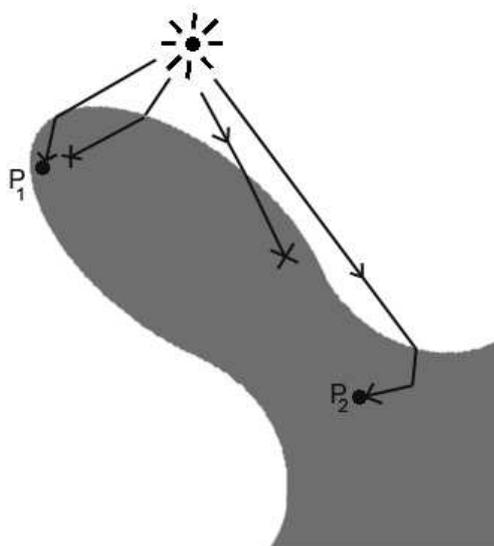
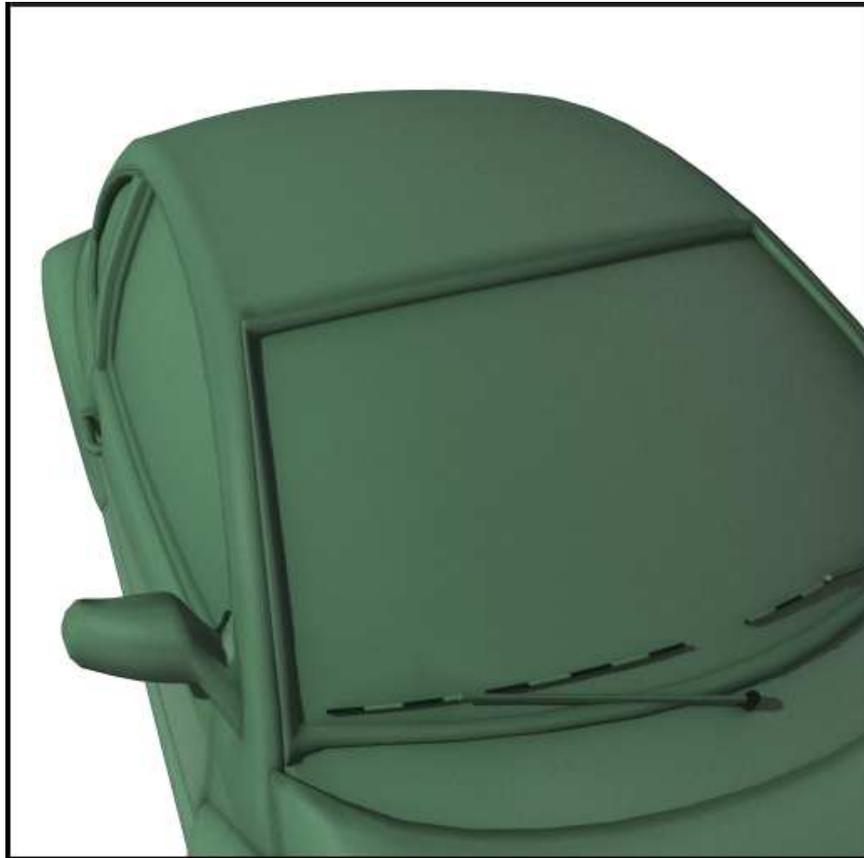


Abbildung 14: Lichttransport durch Material

Abbildung 14 zeigt für die Punkte  $P_1$  und  $P_2$ , dass der direkte Weg zum Licht nicht möglich ist. Die Punkte werden dennoch beleuchtet, da durch Subsurface Scattering Licht auf indirektem Weg zu ihnen gelangt.

### 5.1.4 Objektabhängige Dichteverteilungen

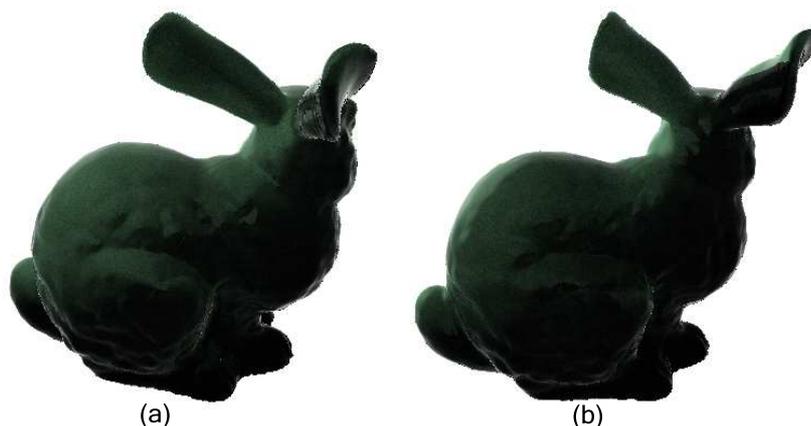


**Abbildung 15:** Ein VW Beetle. Die Wahrscheinlichkeit der Strahlen das Modell zu durchdringen ist auf Null gesetzt.

In dem fertigen Subsurface Scattering Shader wurde jedoch die Möglichkeit gegeben, Dichten für den oberen und unteren Halbraum vorzugeben. Dies ist dann von Nutzen, wenn vorher bekannt ist, dass zum Beispiel ein Strahl in einer bestimmten Szene nicht in der Lage ist ein Modell zu durchdringen. Dadurch wird der Hauptteil der Beleuchtung von den Strahlen gesammelt die zur selben Fläche wieder rausreflektiert werden. Analog existiert der Fall, dass Strahlen grösstenteils das Objekt durchdringen. Eine entsprechende Dichte liefert hier wesentlich schneller ein aussagekräftiges Bild.

Abbildung 15 zeigt das Model des VW Beetle. Bei einem massiven Gegenstand wie einem Auto wird nicht erwartet, dass das Licht durch das Model transportiert wird. Dementsprechend beschränkt sich das Subsurface Scattering hier auf geringe Tiefen innerhalb des Lacks oder auch Plastiks. Die Wahrscheinlichkeit für die Strahlen das Modell zu Durchdringen

wurde auf Null gesetzt. Die Strahlen können das Auto nicht mehr durchdringen und werden nur nach außen hin abgestrahlt. Der Beetle wird größtenteils also ganz normal entsprechend des Phong Beleuchtungsmodells beleuchtet. An der gebogenen Frontscheibe ist jedoch deutlich der weiche Übergang und ein leichtes Leuchten um die Rundungen erkennbar.



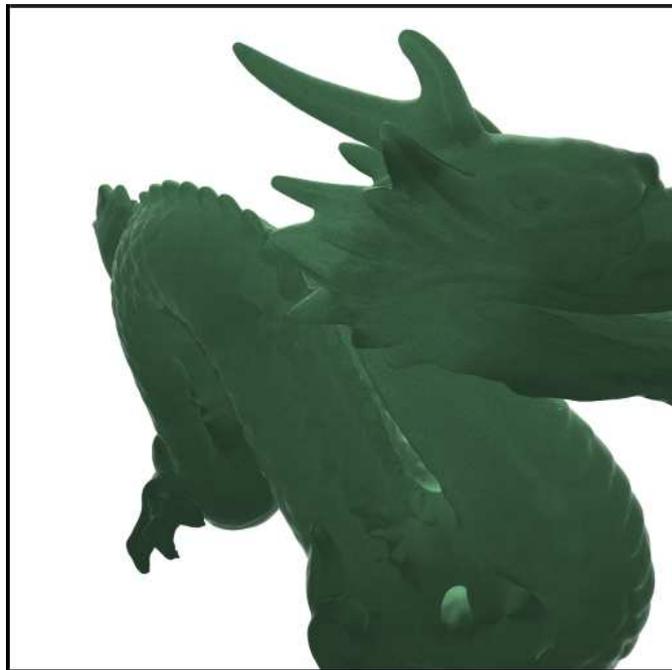
**Abbildung 16:** Ein Stanford Hase. Bei (a) hatten die Strahlen eine 90% Chance intern reflektiert zu werden, bei (b) eine 90% Chance weiter hinein refraktiert zu werden.

Abbildung 13 zeigt zwei Stanford Hasen. Beide wurden mit dem Monte Carlo Pathtracer in 30 Durchläufen gerendert, jedoch mit verschiedenen Wahrscheinlichkeiten für die Strahlen das Modell zu durchdringen. Im ersten Bild 13(a) war die Wahrscheinlichkeit der Strahlen bei 90% intern reflektiert zu werden, im zweiten 90% das Modell zu durchdringen. Sehr deutlich wird hier, dass das erste Bild sein Licht von der gegenüberliegenden oberen Seite des Hasen bezieht, also schneller aufhellt als das zweite Bild, dass sein Licht bevorzugt aus der Tiefe und dem Fuß des Modells zieht.

Abbildung 17 zeigt ein Modell eines Jade Drachen. Da hier ein besonders lichtdurchlässiges Material gewählt wurde resultiert der Großteil der Beleuchtung aus den das Modell durchdringenden Strahlen. Die Dichte der Strahlen ist so gewählt, dass sie vorzugsweise weiter in das Modell vordringen und es so mit größerer Wahrscheinlichkeit durchdringen.

## 5.2 Beschleunigung des Subsurface Scattering Shaders

Der korrekte Subsurface Scattering Shader lieferte sehr gute Bilder und eine Vorschau war dank des Pathtracings schon sehr früh zu sehen. Diese Vorschau ließ sich sogar mit akzeptablen Frameraten von 5-6 Frames interaktiv bewegen. Der nächstliegende Ansatz war natürlich zu versuchen dieses Verfahren zu beschleunigen ehe Vereinfachungen eingesetzt werden würden.



**Abbildung 17:** Das Modell eines Jade Drachens. Die Wahrscheinlichkeit der Strahlen das Modell zu durchdringen ist bei 70%.

### 5.2.1 Erweitern des Pathtracers zu einem Multi-Path Tracer

Erster Ansatz war, den Pathtracer in seiner Funktion zu verändern und mehr Strahlen pro Durchlauf zu verschießen. Somit würden die einzelnen Renderschritte schon bessere Ergebnisse liefern und insgesamt weniger Schritte benötigt werden.

Als erstes wurde das Russisch Roulette deaktiviert und alle möglichen vier Pfade (siehe 2.5.3 *Der Pathtracer*) in einem Durchgang verfolgt. Hier von profitierte vor allem die Vorschau. Der erste Renderframe ermöglichte bereits einen sehr guten Eindruck zu bekommen wie das Modell letztendlich beleuchtet werden würde.

Weiter wurden während eines Rendering Aufrufs mehrere Durchläufe des Subsurface Scattering Shaders ausgeführt. Die einzelnen Frames lieferten nun Ergebnisse, die sehr viel näher an dem gewünschten Endergebnis lagen, doch die Frameraten reduzierten sich enorm. Ein Vergleich der Renderzeiten ergab sogar, dass die insgesamt Renderzeit gleich blieb.

Der fertige Subsurface Scattering Shader, wie er im Rahmen dieser Diplomarbeit vorgelegt wird, hat weiterhin das Russisch Roulette deaktiviert um eine bessere Vorschau zu bieten. Aber wie der klassische Pathtracer verfolgt er nur einen Strahl für jeden der vier möglichen Pfade.

### 5.2.2 Ersetzen des Monte Carlo Verfahrens durch feste Sample Muster

Der Versuch das Monte Carlo Raytracing zu reduzieren war durch anfängliche Tests ausgeschlossen. Idee war es nun das Monte Carlo Verfahren durch ein festes Sample Muster zu ersetzen, das heißt eine feste Anzahl von Strahlen in festgelegte Richtungen zu versenden. Anfängliche Bilder lieferten akzeptable Ergebnisse. Doch sobald das Modell aus leicht anderen Blickwinkeln betrachtet wurde, wurde deutlich, dass das Muster einige Wege verfehlte. Letztendlich müssten sehr viele Strahlen versendet werden, damit das Bild realistische Ergebnisse liefern konnte. Um korrekte Ergebnisse, also eine korrekte Auswertung der BSSRDF innerhalb des Materials zu erhalten, wären sogar unendlich viele Strahlen notwendig. Ein Problem, das gerade durch Monte Carlo Integration behoben werden sollte. Ein Ersetzen des Monte Carlo Raytracings durch feste Sample Muster war also nicht möglich.



## 6 Subsurface Scattering in Echtzeit

Der physikalisch korrekte Subsurface Scattering Shader liefert zwar korrekte Bilder, braucht aber immer noch rund eintausend Durchläufe des Pathtracers um ein rauschfreies Bild zu liefern. Im Folgenden werden zwei Ansätze vorgestellt Subsurface Scattering in Echtzeit zu realisieren. Da Subsurface Scattering selbst stark vereinfacht noch sehr aufwendig ist, sind diese Ansätze lediglich darauf hin ausgelegt einen ähnlichen Effekt zu erzielen und in keinsten Weise physikalisch korrekt.

### 6.1 Subsurface Scattering durch Skalarwerte

Ein einfacher und recht früher Ansatz war es die Kanten aufzuhellen. Die Normale der Fläche und die Richtung zur Lichtquelle hin werden skalar verrechnet und je näher sie nahe Null sind einfach aufgehellt. Somit erhält man das für Subsurface Scattering typische leuchten der Kanten. Ein einfacher Schattenfühler, der mit einem gewissen objektabhängigen Wert entlang der Normalen von der Fläche abgehoben wird, kann sicherstellen, dass die beleuchtete Stelle auch wirklich Rand der Objektsilhouette liegt.

Dieses Verfahren liefert schon sehr ansehnliche Ergebnisse für Objekte mit vielen Rundungen und wenig bis keinen dünnen Stellen, wie dem Stanford Bunny und dem Buddha. Bei scharfen Kanten und großen ebenen Flächen versagt die Annäherung jedoch gänzlich. Auch das Durchleuchten dünner Objekte ist natürlich nicht möglich. Es ist jedoch mit 22 Frames das schnellste Verfahren.

### 6.2 Subsurface Scattering durch Licht-Lot

Subsurface Scattering bedeutet einfach beschrieben, dass das Licht in das Material eindringen kann und durch dieses hindurch leuchtet. Diese einfache Beschreibung entspricht dem, was der naive Betrachter erwartet. Es fehlen die Brechungen und Reflektionen, da auf den ersten Blick das Licht tatsächlich nur durchscheint. Ein solcher Shader ist natürlich nicht physikalisch korrekt, doch er ist vom Aufbau und Aufwand her deutlich einfacher und dadurch schon vom Entwurf her schneller als der korrekte Subsurface Scattering Shader.

Was bisher nicht beschrieben wurde, aber unbedingt auch in diesem Shader vorhanden sein muss, ist das Weichzeichnen der direkt beleuchteten Objektseiten. Dieses Weichzeichnen entsteht dadurch, dass Strahlen in die Oberfläche eindringen und an anderer Stelle dieselbe Fläche wieder verlassen. Somit geht die Umgebung, um den Auftreffpunkt herum gemittelt, mit in den Pixel ein. Naheliegender wäre hier einen Gaußfilter zu verwenden, der eben diesen Effekt erreicht. Der Subsurface Scattering Shader

hat aber keinen Zugriff und kein Wissen auf das zurzeit angezeigte Bild. Er wird immer nur pro Pixel aufgerufen. (siehe 2.8.1 *Der Raytracer Augenblick*)

Es mussten also zwei Shader realisiert werden:

- Ein Frontshader, der den Lichttransport in beleuchteten Flächen behandelt und so den typischen Weichzeichnungseffekt erreicht.
- Ein Backshader, der mit Hilfe eines Lot Strahles ermittelt, ob das Licht durch das Objekt zu dem gefundenen Punkt hin gelangt.

Auch dieses Verfahren gilt nur für homogene Materialien. Zudem kann es keine Effekte darstellen, die durch mehrfach hintereinander liegende transluzente Objekte entstehen.

### 6.2.1 Lichttransport auf beleuchteten Flächen

Trifft ein Strahl auf eine Oberfläche eines transluzenten Objektes, so wird dieser im einfachen Fall in das Material eindringen und nach mehrfachen Brechungen und Spiegelungen das Material verlassen. Vereinfacht gesagt fließt in den Punkt, den der Betrachter sieht, die Beleuchtung eines nahegelegenen Punktes ein. Um diesen komplexen Vorgang zu vereinfachen, wird davon ausgegangen, dass die getroffene Fläche eben ist und die Streuung dem Brechungsgesetz transparenter Objekte folgt, sprich der gebrochene Strahl befindet sich immer auf der von der Flächennormale und dem ankommende Strahl gebildeten Ebene. Dies entspricht der Vereinfachung im Strahlverlauf der auch im "A practical model for subsurface light transport" [11] verwendet wurde.

Die Beleuchtung des Punktes ist nur abhängig von der Flächennormalen und dem auftreffenden Sichtstrahl. Um den Lichttransport wie oben beschrieben durchzuführen reicht es also aus, den Eintrittspunkt zu verschieben, als Austrittspunkt zu beleuchten und beide Farbwerte entsprechend ihrer Anteile zu verrechnen.

Um den Punkt zu verschieben, wird das Kreuzprodukt zwischen der Flächennormalen und dem auftreffenden Strahl berechnet und das Ergebnis dann abermals per Kreuzprodukt mit der Flächennormalen. Der resultierende Vektor ist Teil der Objektfläche und befindet sich auf derselben Ebene wie die Flächennormale und der auftreffende Strahl. Er dient als Translationsvektor mit dem die Flächennormale transliert und somit der Austrittspunkt verschoben wird. Trifft der Strahl senkrecht auf, findet kein Transport statt, er verlässt das Material an derselben Stelle an der er es betreten hat. Dadurch wird automatisch vermieden, dass das Kreuzprodukt zwischen zwei parallelen Vektoren gebildet werden muss.

Die verschatteten Seiten kann dieser Shader jedoch nicht beleuchten. Der neue Austrittspunkt befindet sich auf derselben Ebene wie der Eintrittspunkt, ist er verschattet sind auch alle Austrittspunkte verschattet. Es

war also nötig einen zweiten Teil-Shader für die lichtabgewandten Seiten des Modells zu schreiben.

## 6.2.2 Das Licht-Lot

Um die unbeleuchteten Seiten des Modells zu beleuchten, wurde das Modell als transparent angesehen und die Brechungen komplett weg gelassen. Somit werden die zahlreichen Strahlen eingespart, die sonst nötig sind, um eine beleuchtete Seite zu finden.

Trifft ein Lichtstrahl auf die verschattete Seite eines transluzenten Objektes, wird ein Licht-Lot versendet. Dieses Lot ist ein Strahl, der vom Auftreffpunkt zur Lichtquelle geschossen wird und dessen Richtungsvektor normiert ist. Jedesmal wenn dieser Strahl eine Oberfläche trifft entscheidet der Shader anhand des Skalarwertes  $s$  seiner Richtung und der Flächennormalen, ob er ein Material betritt oder verlässt.

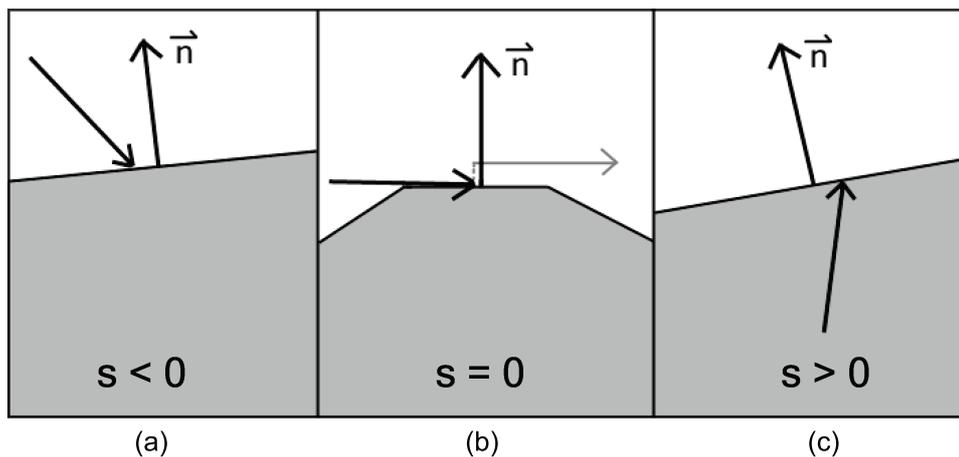


Abbildung 18: Skalarwert  $s$  abhängig vom Richtungsvektor

- Ist  $s$  kleiner Null, betritt der Strahl das Material (Abbildung 18 (a)).
- Ist  $s$  gleich Null, trifft der Strahl parallel zu der Fläche als Tangente auf, er wird um ein Epsilon von der Fläche abgehoben und weiterversendet (Abbildung 18 (b)). Der abgehobene Vektor ist hier grau dargestellt).
- Ist  $s$  größer Null, verlässt der Strahl das Material (Abbildung 18 (c)).

Dieses Lot wird genutzt, um die gesamte Strecke zu messen, die der Strahl innerhalb eines Objektes zurücklegen muss, um die Lichtquelle zu erreichen. Die Länge des Strahls lässt sich einfach aus dem zugehörigen Lambda-Wert auslesen, da der Richtungsvektor des Strahls normiert ist.

Sollte die gemessene Strecke kürzer sein als die maximale Strecke die ein Strahl in das Material eindringen kann, wird der letzte Austrittspunkt beleuchtet und entsprechend der durch die Strecke gegebenen Absorption abgedunkelt.

### 6.2.3 Interpolation zwischen Front- und Backshader

Diese beiden Teil-Shader ähnelten für sich genommen schon den jeweiligen Subsurface Scattering Effekten. Kanten zwischen den beiden Shadern wiesen jedoch noch große Unregelmäßigkeiten auf. Der Frontshader tendierte dazu gerade an scharfen Kanten diese sehr hell zu beleuchten und erschwerte einen weichen Übergang zum Backshader. Um dieses Problem zu lösen, wird der Anteil des Frontshaders abhängig von dem Skalarwert zwischen der Flächennormalen und der Richtung vom Punkt zum Licht abgedunkelt. Je näher der Punkt an einer Kante liegt, desto mehr wird der Anteil, den der Frontshader liefert vernachlässigt und der Farbwert der einfachen Phongbeleuchtung gewichtet. Der Anteil des Frontshaders kann jedoch einen gewissen Schwellwert nicht unterschreiten.

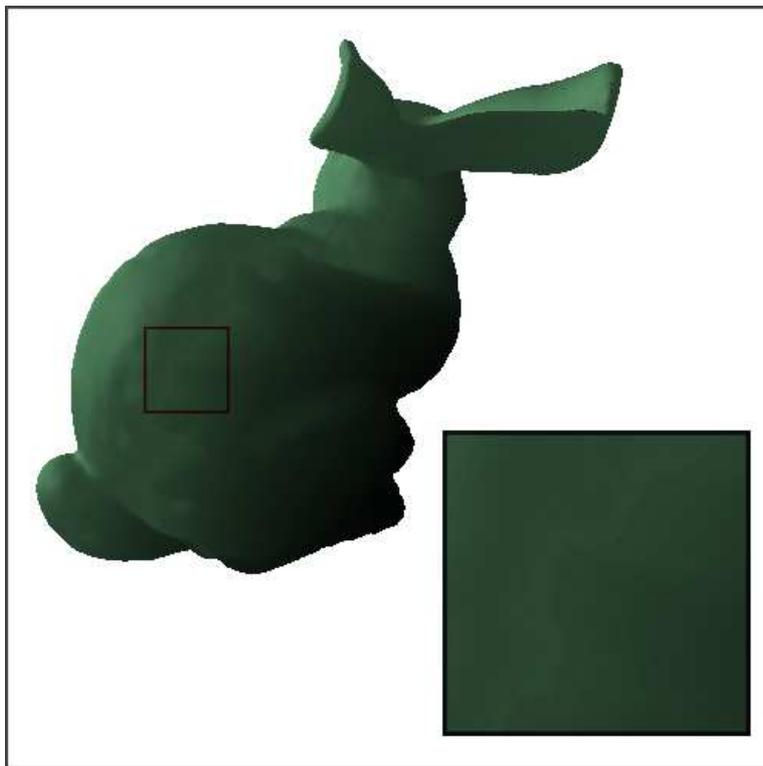


Abbildung 19: Stanfordbunny mit Licht-Lot Shader

Das Angleichen der beiden Shader ist nicht vollständig möglich. Jeder der zwei Shader nähert sich der Kante mit einem anderen Farbverlauf. Je nachdem wie abrupt dieser Farbverlauf verläuft, wird die Kante dennoch sichtbar. Besonders der Wert des Backshaders kann hier stark variieren. Der Fehler ist modellabhängig, verschwindet jedoch bei großen Modellen fast vollständig. Bei kantigen Modellen ist er gar nicht mehr wahrnehmbar.

Abbildung 19 zeigt ein Stanford Hasen. Der Betrachter sieht auf die dem Licht abgewandte Seite. Das Durchscheinen wirkt sehr realistisch, im Vergleich zu dem korrekten Subsurface Scattering Shader ist das Objekt jedoch heller.

Abbildung 20 zeigt ein Stanford Hasen bei dem der Übergang der beiden Shader sehr deutlich hervortritt. Die Wahl der Gewichtung der Shader im Vergleich zum skalarabhängigen Leuchten ist ausschlaggebend für die Sichtbarkeit der Nähte.

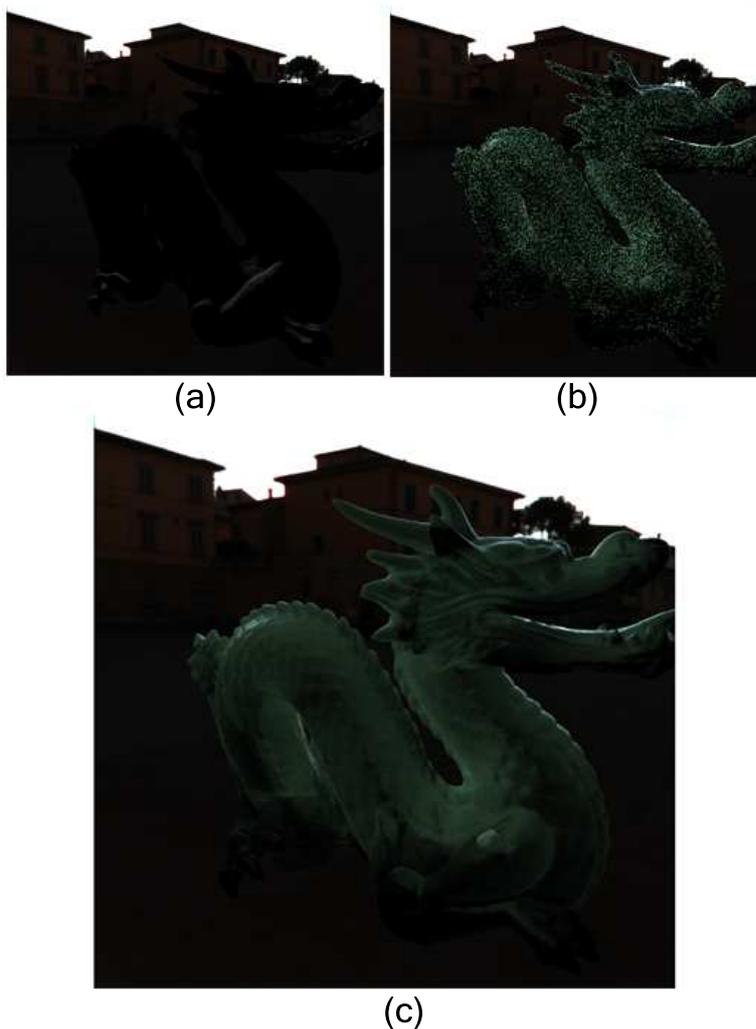


**Abbildung 20:** Nähte zwischen den einzelnen Licht-Lot Shadern



## 7 Ergebnisse

Durch die hohe Komplexität des Subsurface Scattering war es nicht möglich einen einzigen Shader zu schreiben der zwischen Korrektheit und Performanz wählen konnte. Um dennoch Lösungen für diese gegenläufigen Extrema zu finden, wurden drei Shader implementiert, die jeweils mit verschiedenen Graden der Vereinfachung Subsurface Scattering realisieren. Jeder verwendet dabei ein anderes Verfahren um den gewünschten Effekt zu erreichen.



**Abbildung 21:** Die einzelnen Schritte des Subsurface Scattering Shaders in Bildern. (a) Die Szene mit Phong Beleuchtung, (b) der erste Renderschritt des Monte Carlo Path Tracers und (c) der fertige Drache nach 1000 Durchläufen

## 7.1 Subsurface Scattering mit Hilfe eines Monte Carlo Pathtracers

Dieser Ansatz ist nicht vollständig physikalisch korrekt, da er immer noch von wenn auch geringen Vereinfachungen ausgeht. Die Ergebnisbilder sind jedoch hochgradig physikalisch plausibel und ähneln sehr realen Bildern.

Ein Beschleunigen war hier nicht möglich. Die Anzahl der Durchläufe des Pathtracers entsprechen denen vorangehender Verfahren und lässt sich nicht reduzieren. Zwar ist der Subsurface Scattering Shader dieser Diplomarbeit schneller, aber dies lässt sich auf die schnellere Hardware und die Beschleunigungsverfahren des Raytracers *Augenblick* zurückführen.

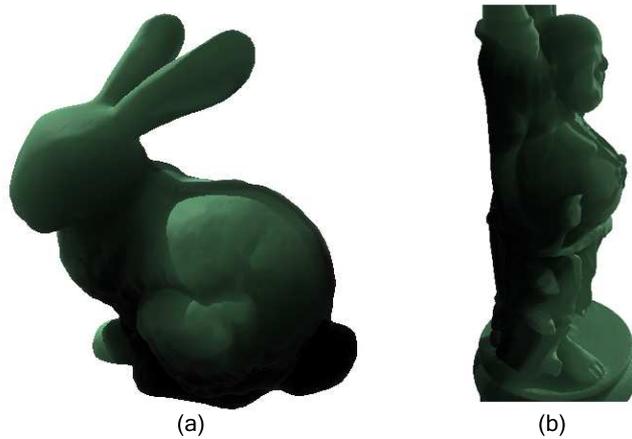
Durch das Erweitern des Pathtracers wurde es ermöglicht bereits im ersten Durchlauf einen ersten Eindruck des fertigen Bildes zu bekommen. Diese Vorschau lässt sich auch in interaktiven Frameraten bewegen.

Abbildung 21 zeigt einen Drachen. Um ihn herum ist die Skybox einer Environmentmap zu sehen.

- Das erste Bild 21 (a) zeigt die Szene mit einfacher Phong Beleuchtung. Da sich die Lichtquelle jenseits des Drachens befindet ist er für den Betrachter schwarz. Weder reflektiert er, noch wird diffuses Licht indirekt auf ihn übertragen.
- Abbildung 21 (b) zeigt den ersten Renderschritt des Monte Carlo Pathtracers und die Vorschau. Diese lässt sich in interaktiven Frameraten bewegen.
- Nach 1000 Durchläufen ist das Rauschen fast vollkommen verschwunden. Das Licht hat den Drachen durchleuchtet, an den Rändern, speziell der Zunge, erkennt man die Fresnel-abhängigen Spiegelungen und die Farben der Environmentmap leuchten schwach auf die Schuppen (Abbildung 21 (c)).

## 7.2 Vereinfachtes Subsurface Scattering durch Licht-Lot

Dieses Verfahren stellt die erste Vereinfachung im Rahmen dieser Diplomarbeit dar. Die Strahlen verhalten sich im weitesten Sinne noch ähnlich dem korrekten Verfahren und liefern immer noch physikalisch plausible Bilder. Dem direkten Vergleich mit den korrekten Bildern halten sie allerdings nicht stand. Ihre Handhabung ist jedoch deutlich einfacher und die interaktiven Frameraten machen dieses Verfahren attraktiv für Bereiche in denen die physikalische Korrektheit zweitrangig ist. Problematisch ist hier immer noch die Handhabung der Kanten zwischen den zwei Teil-Shadern.

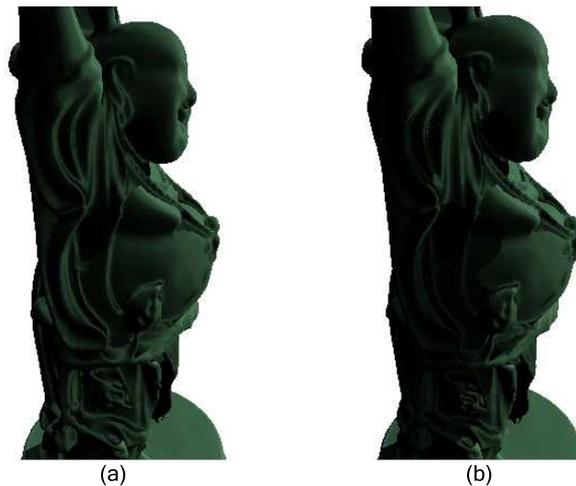


**Abbildung 22:** Ein Terracotta Hase und ein Buddha gerendert mit dem Licht-Lot Shader.

Abbildung 22 zeigt einen Terracotta Hasen 22(a) und einen Buddha 22(b) die jeweils mit dem Licht-Lot Shader gerendert wurden. Die Ergebnisse sehen schon sehr plausibel aus, erscheinen aber zu hell.

### 7.3 Vereinfachtes Subsurface Scattering durch Skalarwerte

Hier wurde auf jegliche physikalische Basis verzichtet. Die Ergebnisbilder sehen zwar nach Subsurface Scattering aus, liefern jedoch nicht für alle Modelle plausible Beleuchtungseffekte. Die Implementation ist allerdings sehr einfach und performant. Es ist vorstellbar für entfernte oder einfachere Objekte diesen Shader zu bevorzugen.



**Abbildung 23:** Ein Buddha gerendert mit dem Skalar Shader. (a) ohne den zusätzlichen Schattenfänger, (b) mit dem zusätzlichen Schattenfänger

Abbildung 23 zeigt den Skalar Shader mit und ohne den Kanten er-  
kennenden Schattenfühler. In Abbildung 23(a) ist deutlich zu erkennen wie  
Falten beleuchtet werden die in keinster Weise Licht bekommen können.  
Mit dem Schattenfühler jedoch fallen alle diese Falsch beleuchteten Fal-  
ten weg, wie in Abbildung 23(b) gezeigt. Der Offset für den Schattenfühler  
muss jedoch objektabhängig gewählt werden.

## 8 Diskussion

Während dieser Diplomarbeit wurden einige Themen aufgeworfen die nähere Betrachtungen verdienen, sei es dass sie eine besondere Schwierigkeit darstellen oder nicht gelöst werden konnten. Sie werden hier vorgestellt und, insofern möglich, anhand der durchgeführten Tests und Verfahren diskutiert.

### 8.1 Aufgabenschwerpunkte: Performanz versus korrekte Lösung

In der Aufgabenstellung war gefordert sowohl eine möglichst korrekte Lösung zu erzielen wie auch eine höchst mögliche Performanz. Sollte dies durch die hohe Komplexität nicht realisierbar sein, sollte ein möglichst einfacher Benutzerparameter einen fließenden Übergang zwischen den zwei Extrema ermöglichen.

Wie bereits unter 5.1 *Monte Carlo Raytracing für das Subsurface Scattering* besprochen, kann das Monte Carlo Raytracing weder weggelassen noch vereinfacht werden. Der Pathtracer benötigt zudem immer eine feste Anzahl an Durchläufen um ein korrektes Bild anzunähern. Auch hier erwiesen sich alle Versuche den gesamten Rendervorgang zu beschleunigen als nicht realisierbar (siehe 5.2 *Beschleunigung des Subsurface Scattering Shaders*). Der korrekte Subsurface Scattering Shader kann somit nicht auf echtzeitfähige Renderingzeiten kommen.

Einzige Möglichkeit Subsurface Scattering schneller zu rendern sind Vereinfachungen der Renderingverfahren. Diese haben dann jedoch keine Verbindung mehr zu dem korrekten Modell.

Ein fließender Übergang mittels eines Benutzerparameters war also nicht möglich. Stattdessen bietet diese Diplomarbeit drei Verfahren mit verschiedenen Kombinationen der gewünschten Kriterien Performanz und Korrektheit:

- Der korrekte Subsurface Scattering Shader. Er bietet eine korrekte Lösung, benötigt jedoch relativ lange Rendering Zeiten.
- Subsurface Scattering mit Hilfe eines Licht-Lots. Die Bilder sind ähnlich der korrekten Lösungen und ermöglichen bereits interaktive Frameraten.
- Subsurface Scattering mit Hilfe von Skalarwerten. Dieses Verfahren sieht nur noch nach Subsurface Scattering aus und hat mit dem korrekten Ablauf nichts mehr gemeinsam. Die Renderingzeiten und der Programmieraufwand sind jedoch deutlich geringer.

## 8.2 BSSRDF für Orte im Material

Wie bereits in 5.1.3 *Monte Carlo Dichteverteilungen innerhalb des Materials* beschrieben, ist es ohne erheblichen Aufwand nicht möglich eine allgemein gültige BSSRDF für Orte innerhalb eines Materials zu bestimmen. Es müssten alle Entfernungen zu allen Punkten auf den Wänden des Objektes bekannt sein und anhand dieser Werte und der Lichtquellenpositionen sollte dann die BSSRDF bestimmt werden.

Daraus folgt, dass keine allgemein gültigen Dichteverteilungen für den Monte Carlo Raytracer angegeben werden können. Im Falle einer Gleichverteilung der Wahrscheinlichkeiten kann der Monte Carlo Raytracer die BSSRDF innerhalb des Materials jedoch in keinster Weise besser oder schneller bestimmen. Wie auch ein einfacher Raytracer mit festem Verteilungsmuster bräuchte er nahezu unendlich viele Strahlen um den Ort innerhalb des Materials und damit den getroffenen Punkt an der Oberfläche korrekt zu beleuchten.

Doch der Monte Carlo Raytracer wird damit nicht überflüssig. Insofern Eigenschaften und Form des Objektes bekannt sind, können auch Dichteverteilungen angegeben werden, die die Bildgenerierung deutlich beschleunigen.

Abgesehen von Subsurface Scattering Effekten kann der korrekte Subsurface Scattering Shader natürlich nicht auf das Monte Carlo Raytracing zu Gunsten eines festen Musters verzichten, da der implementierte Pathtracer sonst an Funktionalität verlieren würde.

## 8.3 Nicht geschlossene Modelle

Leider sind nicht alle Modelle immer geschlossen. Viele besitzen Öffnungen und Löcher, sei es durch das Modellieren selbst oder das dass Original tatsächlich dort eine Öffnung hatte. Das Stanford Bunny ist das Abbild eines Terracota Hasen, der Öffnungen auf der Unterseite und an der unteren Brust besitzt. Seine Wände haben jedoch keine Dimension. Für einen Subsurface Scattering Shader heißt das, dass er das Material betreten kann und, wenn er das Loch trifft, scheinbar keinen Ausgang findet. Es fehlt die gegenüberliegende Wand des Materials. Um selbst bei Modellen mit Löchern noch brauchbare Ergebnisse liefern zu können, muss der Shader erkennen können wann er ein solches Loch im Modell gefunden hat.

Wird ein Strahl innerhalb eines Materials verschossen, wird immer überprüft, was er getroffen hat. Trifft er eine Geometrie verhält er sich entsprechend des Subsurface Shadings. Trifft er jedoch die umspannende Bounding Box oder eine Fläche deren Normale zu ihm zeigt, muss er das Modell verlassen haben und gibt seine Länge als unendlich an, also wird die Materialfarbe verwendet als habe der Strahl im Material geendet. Diese Lösung funktioniert nicht bei Löchern in sehr Nahe gelegenen Wänden. Die

umliegende Fläche wird durchscheinen während die Fläche, die dem Loch gegenüber liegt einen Schatten des Loches zeigt. Der Path Tracer ist in der Lage mithilfe des Monte Carlo Verfahrens diesen Fehler bei kleinen Löchern nahezu rauszurechnen, doch das Bild ist dann nur noch physikalisch plausibel. Physikalisch korrekte Bilder sind nur bei geschlossenen Modellen möglich.



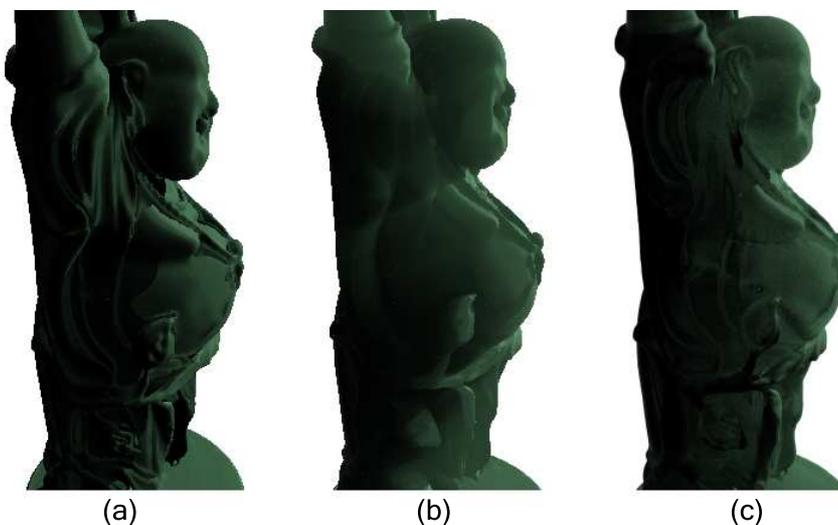
## 9 Vergleich und Auswertung

Dieses Kapitel vergleicht die einzelnen Verfahren dieser Diplomarbeit untereinander und bewertet die Ergebnisse. Vorgestellt und verglichen werden Bilder, die mit den drei Verfahren gerendert wurden. Alle Bilder wurden auf einem 2\*3GHz QuadCore Prozessor von Intel Xeon auf einem Macintosh unter MAC OS X gerendert und haben eine Höhe und Breite von 500 Pixeln.

Ein Vergleich mit realen Bildern konnte leider nicht durchgeführt werden. Es fehlten vergleichbare Modelle und Szenen. Zum Zeitpunkt dieser Diplomarbeit war es leider noch nicht möglich in dem Raytracer *Augenblick* die Lichtquellenposition festzulegen. Deshalb beschränkt sich die Bewertung der Shader lediglich auf die physikalische Plausibilität.

### 9.1 Vergleich der Shader untereinander

Mit zunehmender Vereinfachung des Subsurface Scattering verlieren die Bilder mehr und mehr Plausibilität. Der physikalisch korrekte Subsurface Scattering Shader dient in den folgenden Vergleichen als Referenzmodell.



**Abbildung 24:** Direkter Vergleich der drei Shader. (a) Scalar Shader, (b) Licht-Lot Shader, (c) Monte Carlo Pathtracer

Abbildung 24 zeigt die drei Shader anhand des Modells eines Buddhas im direkten Vergleich. Hier wird besonders deutlich wie von der einfachen Annäherung 24(a) zum vollen Monte Carlo Pathtracer 24(c) die physikalische Plausibilität zunimmt:

- Der einfache Skalar Shader ähnelt schon Subsurface Scattering Effekten, jedoch beleuchtet er eine Stellen falsch.
- Der Licht-Lot Shader sieht schon sehr plausibel aus. Nur im Vergleich mit dem Monte Carlo Pathtracer werden die fehlenden Brechungen und Lichttransporte deutlich. Da seine Strahlen immer den direkten weg zum Licht wählen ist er zudem heller als das korrekte Verfahren.
- Das korrekte Verfahren mit Hilfe des Monte Carlo Pathtracers liefert ein plausibles Bild und stellt Subsurface Scattering korrekt dar.

## 9.2 Vergleich der Renderingzeiten

Die folgenden zwei Tabellen präsentieren die verschiedenen Renderingzeiten anhand der zwei Modelle des Terracotta Hasen und des Buddhas. Die Vereinfachungen lieferten ihr Endergebnis nach jeweils einem Render-schritt, für das korrekte Verfahren wurden durch den Monte Carlo Pathtracer mehrere Durchläufe nötig. **Renderingzeiten für den Stanford Hasen**

Benutzer Shader	Anzahl der Renderparses	benötigte Zeit	Bilder pro Sekunde
Korrekte Shader	1	0,12 sec	8,22 fps
	100	4,66 sec	0,21 fps
	500	23,47 sec	0,04 fps
	1000	67,51 sec	0,01 fps
Licht-Lot Shader	1	0,11 sec	9,49 fps
Scalar Shader	1	0,03sec	39,11 fps

### Renderingzeiten für das Modell eines Buddhas

Benutzer Shader	Anzahl der Renderparses	benötigte Zeit	Bilder pro Sekunde
Korrekte Shader	1	0,17 sec	5,77 fps
	100	6,27 sec	0,16 fps
	500	29,61 sec	0,033 fps
Licht-Lot Shader	1	0,11 sec	9,43 fps
Scalar Shader	1	0,04sec	22,94 fps

Der korrekte Shader benötigt sehr viel Zeit um ein rauschfreies Bild zu liefern. Die Vorschau ist jedoch interaktiv bedienbar.

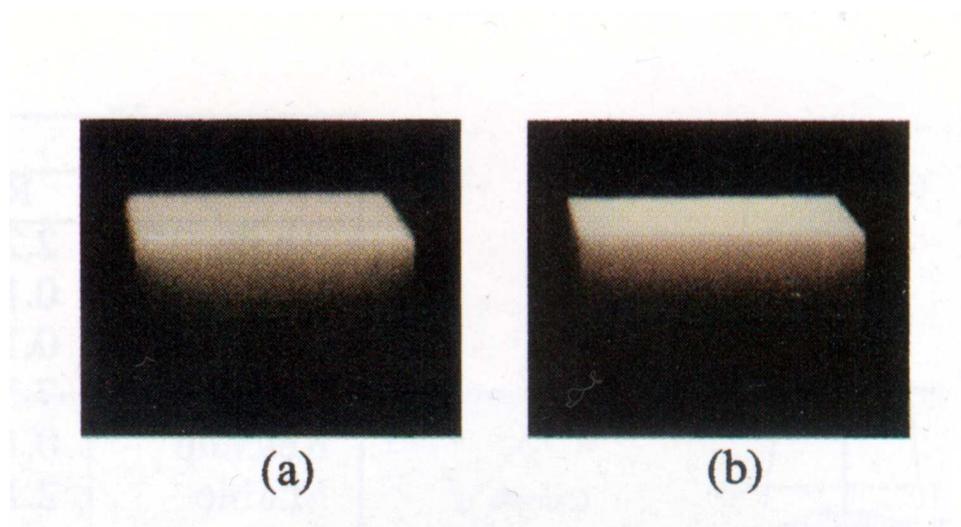
Die beiden anderen Verfahren sind deutlich schneller und bleiben, da sie keine mehrfachen Rendschritte benötigen, für beide Modelle interaktiv.

### 9.3 Vergleich mit *A practical model for subsurface light transport*

In dem Paper *A practical model for subsurface light transport* von Henrik Wann Jensen et. al. wurde eine sehr realistische Annäherung für Subsurface Scattering gefunden. In diesem Kapitel werden die Shader dieser Diplomarbeit mit eben jenem Shader verglichen.

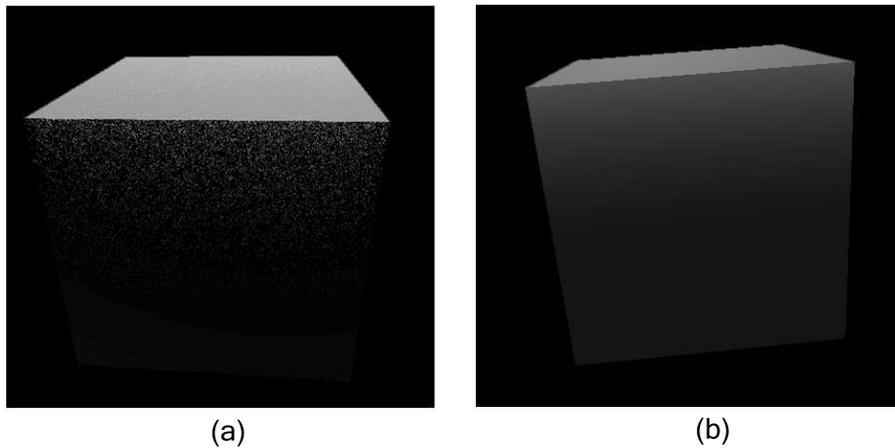
Zwar fehlten die Modelle und durch die Entwicklung in der Hardware fallen Vergleiche schwer, dennoch lassen sich Vorteile

Henrik Wann Jensen et. al. verifizierten ihr Ergebnis anhand eines Photos eines kleinen Marmorblocks.



**Abbildung 25:** Subsurface Scattering eines Lasers in einem Marmorblock. Entnommen aus [11]

Abbildung 25(a) zeigt das Foto des Marmorblocks, Abbildung 25(b) zeigt das gerenderte Bild mit dem Shader aus *A practical model for subsurface light transport*. Die Bilder ähneln sich sehr, jedoch ist erkennbar, dass der Lichtverlauf im realen Foto sich im Gegensatz zu dem gerenderten nach unten hin krümmt.



**Abbildung 26:** Ein Block gerendert mit (a) dem korrekten Monte Carlo Pathtracer und (b) dem Licht-Lot Shader

Der korrekte Monte Carlo Pathtracer dieser Diplomarbeit stellt diese Krümmung ähnlich dar, Abbildung 26(a). Leider kann hier nur die Vorschau gezeigt werden, da es zum Zeitpunkt dieser Ausarbeitung für den Shader nicht möglich war auf vorherige Frames des Raytracers *Augenblick* zu zugreifen. Durch diesen Nachteil wurden bei Quadern und ähnlichen Objekten die Subsurface Scattering Ergebnisse des Pathtracers abgedunkelt und nicht korrekt eingerechnet.

Auch der Licht-Lot Shader ergibt ein ähnliches Bild. Es wirkt jedoch im Vergleich künstlich und eindeutig computergeneriert.

## 10 Ausblick

Natürlich sind die drei vorgestellten Shader keine absolute Lösungen. Alle drei Verfahren lassen sich sicher noch verbessern und erweitern. Im Folgenden werden einige Möglichkeiten aufgezeigt, wie die Shader erweitert und verbessert werden können:

- **Performanz:** Durch geschicktes Programmieren können teure Algorithmen und Speicherzugriffe sicher noch weiter optimiert werden.
- **Subsurface Scattering für andere Raytracer:** Zwar sind alle drei Shader zur Zeit Plugins für den Raytracer *Augenblick*, doch lassen sie sich ohne großen Aufwand auch auf andere Raytracer umschreiben. Einzige Voraussetzungen für den Shader sind dabei:
  - Zugriff auf die Szenedaten
  - Die Möglichkeit Strahlen zu versenden
  - Die Möglichkeit die Szene mehrfach zu rendern und die Ergebnisbilder zu interpolieren.
- **Steuerung der Parameter:** Vor allem der korrekte Subsurface Scattering Shader verfügt über eine Unmenge an Parametern, die das Ergebnisbild beeinflussen. Im Rahmen dieser Diplomarbeit war es nicht mehr möglich diese Parameter in die GUI des Raytracers *Augenblick* zu integrieren.
- **Entfernungsabhängige Shaderwahl:** Der Aufbau der drei hier vorgestellten Shader legt nahe die verschiedenen Stufen der physikalischen Plausibilität und der entsprechenden Rechenzeiten zu nutzen um abhängig von der Entfernung zum Betrachter verschiedene Shader zu benutzen. Ein Shader mit hoher physikalischer Plausibilität kann nahe Objekte beleuchten, während ein weniger plausibler Shader weit entfernte Objekte beleuchtet. Bedingt durch ihre Performanz kommen hier nur die beiden Vereinfachungen in Frage.

Ein Wechsel der Shader eines Objektes abhängig von der Position und Entfernung des Betrachters kommt dabei weniger in Frage. Bei Szenen, in denen der Betrachter jedoch bestimmte Bereiche nicht erreichen kann, könnten entfernte Objekte direkt performantere Shader zugewiesen bekommen.
- **Modellieren von Einschlüssen:** Für Einschlüsse mit klar definierten Formen ließen sich der korrekte Shader und der Licht-Lot Shader erweitern um auch solche Materialien darstellen zu können. Zum Zeitpunkt der Diplomarbeit war es leider nicht möglich innerhalb des Raytracers *Augenblick* den Objekten der Szene verschiedene Eigenschaften zuzuweisen.

- **Volumendatensätze:** Es ist möglich den korrekten Shader zu erweitern und für Volumendatensätze zu implementieren. Hiermit wären dann auch Einschlüsse mit fließendem Übergang zum umliegenden Material möglich. Die Performanz würde jedoch enorme Einbußen hinnehmen müssen. Jedes Voxel müsste als eigenes Objekt betrachtet werden und es müsste ständig geprüft werden ob ein Material- oder Dichtewechsel stattgefunden hat. Zum Beispiel für medizinisch wissenschaftliche Zwecke wäre dieses Verfahren jedoch sehr interessant da die Strahlen sehr realitätsnah verfolgt werden können und Renderzeiten nicht so wichtig sind wie eine korrekte Auswertung selbiger.

## Literatur

- [1] Michael Ashikhmin, Simon Premoze, and Peter Shirley. A microfacet-based brdf generator. In *Computer Graphics Proceedings, Annual Conference Series*, pages 65–74. ACM SIGGRAPH, 2000.
- [2] James F. Blinn. Light reflection functions for simulation of clouds and dusty surfaces. In *Computer Graphics Proceedings, Annual Conference Series*, pages 21–29. ACM SIGGRAPH, 1982.
- [3] W.J. Bouknight. *An Improved Procedure for Generation of Half-tone Computer Graphics Representation*. UI, Coordinated Science Laboratory, 1969.
- [4] S. Chandrasekhar. *Radiative Transfer*. Oxford University Press, 1960.
- [5] Franklin C. Crow. Shadow algorithms for computer graphics. In *Computer Graphics Proceedings, Annual Conference Series*, pages 242–248. ACM SIGGRAPH, 1977.
- [6] G. Eason, A. Veitch, R. Nisbet, and F. Turnbull. *The theory of the backscattering of light by blood*. J.Physics, 1978.
- [7] T.J. Farrel, M.S. Patterson, and B. Wilson. *A diffusion theory model of spatially resolved, steady-state diffuse reflectance for the noninvasive determination of tissue optical properties in vivo*. Med.Phys., 1992.
- [8] Andrew Glassner. *An Introduction to Ray Tracing*. Academic Press, 1989.
- [9] Pat Hanrahan and W. Krueger. Reflection from layered surfaces due to subsurface scattering. In *Computer Graphics Proceedings, Annual Conference Series*, pages 165–174. ACM SIGGRAPH, August 1993.
- [10] Henrik Wann Jensen. *Realistic Image Synthesis Using Photon Mapping*. AK Peters, 2001.
- [11] Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. A practical model for subsurface light transport. In *Computer Graphics Proceedings, Annual Conference Series*, pages 511–518. ACM SIGGRAPH, 2001.
- [12] James T. Kajiya. The rendering equation. In *Computer Graphics Proceedings, Annual Conference Series*, pages 143–150. ACM SIGGRAPH, August 1986.
- [13] Malvin H. Kalos and Paula A. Whitelock. *Monte Carlo Methods: Volume 1: Basics*. John Wiley and Sons, 1986.

- [14] Eric Lafortune. Mathematic models and monte carlo algoritms for physically based rendering. In *Ph.D. thesis*. Katholieke Universiteit Leuven, 1986.
- [15] F.E. Nicodemus, J.C. Richmond, J.J. Hsia, and I.W. Ginsberg und T. Limperis. *Geometric considerations and nomenclature for reflectance*, volume Monograph 161. National Bureau of Standards(US), Oktober 1977.
- [16] Michael Oren and Shree K. Nayar. Generalization of lambert’s reflectance model. In *International Journal on Computer Vision*, pages 227–251, 1995.
- [17] Matt Pharr and Pat Hanrahan. Monte carlo evaluation of non-linear scattering equations for subsurface reflection. In *Computer Graphics Proceedings, Annual Conference Series*, pages 75–84. ACM SIGGRAPH, 2000.
- [18] Bui Tuong Phong. *Illumination for Computer Generated Pictures*. Comm. ACM, 1975.
- [19] L. Reynolds, C. Johnson, and A. Ishimaru. *Applied Optics*. 15:2059, August 1976.
- [20] Christophe Schlick. *An inexpensive BRDF model for physically-based rendering*. Computer Graphics Forum, 1994.
- [21] B. Walter. *Microfacet Models for Refraction through Rough Surfaces*. EGSR, 2007.
- [22] Lance Williams. Casting curved shadows on curved surfaces. In *Seminal graphics: pioneering efforts that shaped the field*, pages 51–55. ACM, 1998.
- [23] C. Wylie, G.W. Romney, D.C. Evans, and Erdahl. *Halftone Perspective Drawings by Computer*. Proc. AFIPS FJCC, 1967.