

# Diplomarbeit

## Fotorealistische Beleuchtungs- und Rendertechniken

Am Beispiel von Mental Ray und Maxwell Render

Ausgeführt zum Zweck der Erlangung des akademischen Grades eines

**Dipl.-Ing. (FH) Telekommunikation und Medien**

am Fachhochschul-Diplomstudiengang Telekommunikation und Medien St. Pölten

unter der Erstbetreuung von  
Mag. Franz Schubert

Zweitbegutachtung von  
Mag. Peter Adametz

Ausgeführt von  
Florian Schödl  
tm0210038105

St. Pölten, am

Unterschrift:

# Ehrenwörtliche Erklärung

Ich versichere, dass

- ich diese Diplomarbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.
- ich dieses Diplomarbeitsthema bisher weder im Inland noch im Ausland einem Begutachter/ einer Begutachterin zur Beurteilung oder in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

Diese Arbeit stimmt mit der vom Begutachter beurteilten Arbeit überein.

St. Pölten, am

Unterschrift

## **Danksagung**

An dieser Stelle möchte ich mich bei allen bedanken, die mich bei der Anfertigung dieser Arbeit unterstützt haben.

Ich danke meinen Eltern, dass sie mir in meinen Ausbildungs- und Berufswünschen stets vollkommen freie Wahl gelassen haben und mir das Studium an der Fachhochschule St. Pölten ermöglicht und finanziert haben.

Besonderer Dank gilt meiner Freundin, Nora Schmid, für ihre Unterstützung und Geduld in allen Bereichen.

## **Kurzfassung**

Fotorealistische Bilder zu erzeugen ist noch immer eine große Herausforderung, und ist bis heute eine treibende Kraft für die Weiterentwicklung in der Computerindustrie in den letzten Jahrzehnten. Es wird erwartet, dass der Trend hin zur Erstellung absolut fotorealistischer Bilder weiter anhält, da diese Art des Renderings, durch die stetig steigende Leistung heutiger Computerhardware, immer mehr Anwendungsgebiete findet.

Diese Diplomarbeit erläutert die Schwierigkeiten, die sich bei der Berechnung von fotorealistischen Bildern ergeben und gibt einen Überblick über die verschiedenen Renderalgorithmen.

Im letzten Teil der Diplomarbeit werden die wichtigsten Renderverfahren anhand eines praktischen Beispiels verglichen.

## **Abstract**

Generating photorealistic pictures is a very ambitious goal, and it has been one of the major driving forces in computer graphics over the last decades. Visual realism has always been a strong motivation for research in his field, and it is a selling point for many graphics-related, commercially available products. It is expected that this trend will continue in the coming years, and that photorealism will remain one of the core fields in rendering.

This diploma thesis illustrates the difficulties, which arise when rendering photorealistic pictures and gives a review of the different render algorithms.

In the last part of the diploma thesis the most important render algorithms are compared by means of a practical example.

# Inhaltsverzeichnis

<b>1. EINLEITUNG .....</b>	<b>7</b>
<b>2. BEGRIFFSDEFINITIONEN.....</b>	<b>8</b>
2.1. PHOTOREALISTIC RENDERING .....	8
2.2. GLOBALE BELEUCHTUNG .....	8
2.3. ALGORITHMUS .....	9
<b>3. ANWENDUNGSGEBIETE VON FOTOREALISTISCHEM RENDERING.....</b>	<b>10</b>
3.1. FILM UND VISUAL EFFECTS .....	10
3.2. ARCHITEKTUR.....	10
3.3. COMPUTERSPIELE .....	11
3.4. WERBUNG UND PRODUKTVISUALISIERUNGEN .....	11
<b>4. PHYSIKALISCHE GRUNDLAGEN DER LICHTAUSBREITUNG.....</b>	<b>12</b>
4.1. LICHTMODELLE.....	12
4.1.1. <i>Quantenoptik</i> .....	12
4.1.2. <i>Wellenmodel</i> .....	12
4.1.3. <i>Geometrische Optik</i> .....	13
4.2. RADIOMETRIE .....	13
4.2.1. <i>Radiometrische Größen</i> .....	14
4.3. LICHTEMISSION.....	15
4.4. INTERAKTION VON LICHT MIT OBERFLÄCHEN .....	16
4.4.1. <i>BRDF (Bidirectional Reflectance Distribution Function)</i> .....	16
4.4.2. <i>BRDF Arten</i> .....	17
4.5. RENDERGLEICHUNG.....	18
<b>5. ENTWICKLUNG DES FOTOREALISTISCHEN RENDERING.....</b>	<b>21</b>
5.1. SHADING .....	21
5.1.1. <i>Flat Shading (Constant Shading)</i> .....	21
5.1.2. <i>Gouraud Shading</i> .....	22
5.1.3. <i>Phong Shading</i> .....	23
5.2. RAY TRACING .....	24
5.2.1. <i>Verdeckungsberechnung</i> .....	25
5.2.2. <i>Shading</i> .....	25
5.2.3. <i>Schatten</i> .....	26
5.2.4. <i>Rekursives Raytracing</i> .....	26

5.2.5.	<i>Diffuses Raytracing</i> .....	26
5.3.	RADIOSITY .....	27
5.4.	MULTIPASS METHODEN.....	29
<b>6.</b>	<b>SAMPLING UND ANTIALIASING .....</b>	<b>29</b>
6.1.	SAMPLING THEORIE.....	29
6.2.	ALIASING .....	30
6.3.	ANTIALIASING .....	32
6.3.1.	<i>Nonuniform Sampling</i> .....	32
6.3.2.	<i>Adaptive Sampling</i> .....	32
6.3.3.	<i>Prefiltering</i> .....	33
<b>7.</b>	<b>MONTE CARLO METHODEN .....</b>	<b>34</b>
7.1.	GRUNDIDEE.....	34
7.2.	STRATIFIED SAMPLING .....	35
7.3.	IMPORTANCE SAMPLING.....	36
<b>8.</b>	<b>MONTE CARLO ALGORITHMEN.....</b>	<b>36</b>
8.1.	PATHTRACING.....	36
8.2.	BIDIRECTIONAL PATHTRACING .....	39
8.3.	IRRADIANCE CACHING.....	41
<b>9.</b>	<b>METROPOLIS LIGHT TRANSPORT.....</b>	<b>45</b>
9.1.	GRUNDIDEE.....	45
9.2.	METROPOLIS SAMPLING .....	46
9.2.1.	<i>Detailed Balance</i> .....	46
9.2.2.	<i>Annahmewahrscheinlichkeit</i> .....	47
9.2.3.	<i>Initialisierung und Start-Up Bias</i> .....	47
9.3.	MUTATIONSVERFAHREN.....	48
9.3.1.	<i>Bidirektionale Mutationen</i> .....	49
9.3.2.	<i>Perturbations</i> .....	51
9.3.3.	<i>Lens Subpath Mutationen</i> .....	53
9.4.	VERBESSERUNGEN AM ALGORITHMUS.....	53
9.4.1.	<i>Direkte Beleuchtung</i> .....	53
9.4.2.	<i>Importance Sampling an Mutationen</i> .....	53
9.5.	ERGEBNISSE .....	54
<b>10.</b>	<b>PHOTON MAPPING .....</b>	<b>57</b>

10.1.	GRUNDIDEE.....	57
10.1.1.	<i>Photonenaussendung</i> .....	58
10.1.2.	<i>Projektionskarten (Projection Maps)</i> .....	59
10.2.	PHOTONENVERFOLGUNG DURCH DIE SZENE .....	60
10.2.1.	<i>Reflexion, Transmission oder Absorption</i> .....	60
10.3.	SPEICHERUNG DER PHOTONEN .....	61
10.3.1.	<i>Arten der Photon Maps</i> .....	62
10.3.2.	<i>Der kd-Baum</i> .....	63
10.4.	BERECHNUNG DER STRAHLDICHTEN .....	64
10.5.	RENDERING .....	66
10.5.1.	<i>Direkte Beleuchtung</i> .....	67
10.5.2.	<i>Spiegelnde Reflexion</i> .....	68
10.5.3.	<i>Kaustiken</i> .....	68
10.5.4.	<i>Diffuse Reflexion</i> .....	69
10.6.	FILTER.....	69
10.7.	ERGEBNISSE .....	70
<b>11.</b>	<b>PRAKTISCHE ANWENDUNG IN MAXWELL<sup>TM</sup> UND MENTAL RAY<sup>®</sup> ...</b>	<b>72</b>
11.1.	FORSCHUNGSLEITENDE FRAGESTELLUNG.....	72
11.2.	HYPOTHESEN .....	72
11.3.	OPERATIONALISIERUNG VON VARIABLEN.....	72
11.3.1.	<i>Abhängige Variablen</i> .....	72
11.3.2.	<i>Unabhängige Variablen</i> .....	73
11.3.3.	<i>Intervenierende Variablen</i> .....	73
11.4.	AUFBAU DES EXPERIMENTS.....	74
11.4.1.	<i>Referenzfoto</i> .....	74
11.4.2.	<i>Aufbau der Szene in Maxwell Render (3D Studio Max)</i> .....	75
11.4.3.	<i>Aufbau der Szene in Mental Ray (Softimage XSI)</i> .....	77
11.5.	FAZIT.....	79
11.5.1.	<i>Beantwortung der gestellten Forschungsfragen</i> .....	79
11.5.2.	<i>Überprüfung der aufgestellten Hypothese</i> .....	82
<b>12.</b>	<b>ZUSAMMENFASSUNG .....</b>	<b>83</b>
<b>13.</b>	<b>GLOSSAR .....</b>	<b>84</b>
<b>14.</b>	<b>QUELLENVERZEICHNIS .....</b>	<b>87</b>
<b>15.</b>	<b>ABBILDUNGSVERZEICHNIS .....</b>	<b>89</b>

## 1. Einleitung

Das Erzeugen eines zweidimensionalen Bildes aus einer dreidimensionalen Szene, auch genannt Rendering, ist ein fundamentaler Bestandteil der Computergrafik. In vielen Einsatzgebieten spielt Realismus eine große Rolle und ist ein wichtiges Verkaufsargument für viele kommerzielle Produkte im Grafiksektor. Mit Hilfe modernster Rendertechniken ist es mittlerweile möglich digitale Bilder zu erzeugen, die der Betrachter nicht mehr von einer Fotografie unterscheiden kann. Durch die schnelle Weiterentwicklung und die stetig steigende Leistung heutiger Computerhardware findet diese Art des Renderings immer mehr Anwendungsgebiete.

Aus der fundamentalen *Rendergleichung*, einer Integralgleichung, die die Energieerhaltung bei der Ausbreitung von Lichtstrahlen beschreibt, haben sich verschiedenste Lösungsansätze zur Berechnung globaler Beleuchtung entwickelt. Unterschiedliche 3D Programme verwenden unterschiedliche Algorithmen, die globale Beleuchtung simulieren können.

Diese Diplomarbeit soll einen Überblick über die verschiedenen Methoden geben. Anschließend wird anhand der Render-Software *Mental Ray* und *Maxwell Render*, die sehr unterschiedliche Algorithmen verwenden, versucht, ein fotorealistisches Bild zu erzeugen. Hierbei soll nicht nur auf physikalisch korrekte Lichtberechnung geachtet werden, sondern auch auf die Flexibilität der Render-Software.

Da Fotorealismus ein sehr subjektives Empfinden ist, wird ein reales Foto, der in 3D nachgestellten Szene, als Vergleichswert herangezogen.

## 2. Begriffsdefinitionen

In diesem Kapitel werden einige Begriffe definiert, die in der weiteren Arbeit häufig vorkommen und deren Verständnis von großer Bedeutung sind.

### 2.1. *Photorealistic Rendering*

Photorealistic setzt sich aus den Worten „*Photo*“ (griechisch für Lichtbild<sup>1</sup>) und „*Realsitik*“ (lateinisch für Wirklichkeitsnähe<sup>2</sup>) zusammen. *Rendern* (von engl. *to render*: wiedergeben, vortragen) bedeutet im weiteren Sinne: die grafische Darstellung eines beschriebenen Objekts; im engeren Sinne: die Umsetzung von dreidimensionalen Modellen in eine zweidimensionale Darstellung. Mit Hilfe verschiedener Verfahren werden im Computer generierte Projektionen von Objekten durch Nachbearbeitung (Licht, Schatten, Farbverläufe) in realistisch wirkende dreidimensionale Körper umgewandelt.<sup>3</sup>

Beim fotorealistischen Rendern wird also versucht ein Bild zu erzeugen, das von einer echten Fotografie nicht mehr unterschieden werden kann. Dies erfordert eine physikalisch korrekte Berechnung von Licht und Materialien.

Das Streben nach Fotorealismus gibt es nicht nur in der 3D Grafik, sondern war schon immer ein Teil der Kunstgeschichte.

### 2.2. *Globale Beleuchtung*

Globale Beleuchtung (engl. Global Illumination, abgekürzt GI) bezeichnet in der 3D-Computergrafik die Simulation aller Möglichkeiten der Ausbreitung von Lichtstrahlen in einer 3D-Szene. Dadurch werden die Gesetze der geometrischen

---

<sup>1</sup> vgl. Wissen.de Online Lexikon, <http://www.wissen.de/wde/generator/wissen/ressorts/bildung/index?page=1212736.html>

<sup>2</sup> vgl. Wissen.de Online Lexikon, <http://www.wissen.de/wde/generator/wissen/ressorts/bildung/index?page=1222296.html>

<sup>3</sup> vgl. Wissen.de Online Lexikon, <http://www.wissen.de/wde/generator/wissen/ressorts/technik/computer/index?page=1224188.html>

Optik sowie der Energieerhaltung vollständig erfüllt und ein realistisches Bild erzeugt. Es gibt dabei 2 Ansätze: *Faked GI* und *Real GI*.

Beim ersten Ansatz wird innerhalb der 3D-Software durch kluge Platzierung von diversen Lichtquellen und durch Anwendung von verschiedenartigen Tricks die GI „erschwindelt“. In der Realität würden die Lichtpartikel von allen Objekten abgelenkt, reflektiert oder sogar beides. Bei der „gefakten“ GI platziert man die Lichter so, dass sie als Ersatz für abgelenkte und/oder reflektierte Lichtpartikel agieren. Üblich wären beispielsweise ein Licht als Ersatz der primären Lichtquelle (zum Beispiel Sonne) und weitere Lichter, die alle extrem dunklen Bereiche aufhellen, die nicht von der primären Lichtquelle erleuchtet werden.

Bei der „echten“ GI werden spezielle Renderalgorithmen verwendet, die tatsächlich physikalisch korrekte Beleuchtung simulieren können. Diese Art der GI gewinnt durch die stetig steigende Leistungsfähigkeit der Hardware in der Computergrafik immer mehr an Bedeutung.<sup>4</sup>

In dieser Diplomarbeit werden ausschließlich Verfahren zur Berechnung physikalisch korrekter GI erläutert.

### **2.3. Algorithmus**

Unter einem Algorithmus versteht man allgemein eine genau definierte Handlungsvorschrift zur Lösung eines Problems oder einer bestimmten Art von Problemen. In der Computergrafik erfolgt die Berechnung eines fotorealistischen Bildes immer durch einen Renderalgorithmus zur Lösung der Rendergleichung.

---

<sup>4</sup> Vgl. Wikipedia, [http://de.wikipedia.org/wiki/Globale\\_Beleuchtung](http://de.wikipedia.org/wiki/Globale_Beleuchtung)

### **3. Anwendungsgebiete von Fotorealistischem Rendering**

Das folgende Kapitel gibt einen Überblick über die verschiedenen Anwendungsgebiete von fotorealistischem Rendering.

#### **3.1. Film und Visual Effects**

Visual Effects sind schon immer eine große Antriebskraft für Entwicklungen in der Computergrafik gewesen. Eine besondere Rolle spielen hier Filme, bei denen reale Bilder mit computergenerierten Bildern in Berührung kommen. In diesem Fall ist eine einheitliche Beleuchtung von Filmmaterial und 3D Elementen absolut notwendig. Global Illumination spielt hierbei eine große Rolle. Der Einsatz von fotorealistischem Rendern geht von der Integration einzelner Objekte (Dinosaurier in Jurassic Park), über die Erzeugung virtueller Hintergründe, bis hin zu Szenen, die vollständig in 3D erstellt werden (Matrix Reloaded).

Der Einsatz von digitalen Doubles für besonders gefährliche Stuntszenen wäre ohne fotorealistisches Ergebnis nicht denkbar.

Durch den Einsatz von Computergrafik in Filmen sind dem Regisseur keine gestalterischen Grenzen mehr gesetzt. Oft können beim Einsatz von 3D Grafik in Filmen Kosten gesenkt werden, da digitale Sets billiger und schneller gebaut, und Änderungen schneller umgesetzt werden können.

#### **3.2. Architektur**

Architekturdesign ist wohl die vorteilhafteste Anwendung von fotorealistischem Rendering. Man kann Visualisierungen von Gebäuden erstellen, die noch nicht gebaut wurden. Der große Vorteil gegenüber einem Miniaturmodell besteht in der Interaktivität, die mit einer 3D Szene erreicht werden kann. Bei Architekturvisualisierungen kommen oft so genannte „Fly-Throughs“ zum Einsatz. „Fly-Throughs“ sind virtuelle Rundgänge durch die 3D Szene, bei der nur die Kamera in Bewegung ist und weder Lichtquellen, noch Geometrie animiert wurde. Diesen Umstand können einige Renderalgorithmen nutzen, um die Renderzeit beträchtlich zu verringern. Solche Algorithmen können die Lichtverteilung unabhängig vom Standpunkt berechnen. Dadurch muss ausschließlich beim

ersten Frame eine Berechnung stattfinden, auf die dann folgende Frames zurückgreifen können. Bei Architekturvisualisierungen kommen neben Innenaufnahmen auch Außenaufnahmen von Gebäuden zum Einsatz, da man die Möglichkeit hat verschiedene Tages- und Jahreszeiten zu simulieren. Neben dem puren Design kann mittels Global Illumination auch bestimmt werden, ob gewisse Beleuchtungsrichtlinien eingehalten werden. So kann man zum Beispiel bei Bürogebäuden schnelle Änderungen an der Positionierung der Fenster oder der Farbe der Wände durchführen.

### **3.3. Computerspiele**

Computerspiele haben die Aufgabe, uns in eine fiktive Welt eintauchen zu lassen. Um diese Illusion möglichst gut aufrecht zu erhalten ist eine fotorealistische Grafik wichtig. Da bei Computerspielen jedoch eine hohe Interaktivität gefordert wird, kann Fotorealismus noch nicht vollständig erreicht werden, da dies nur mit enorm leistungsfähiger Hardware umsetzbar wäre. Jedoch bedienen sich Spieleentwickler immer mehr jener Techniken, die ihren Ursprung beim Fotorealistischen Rendering haben. So kommt bei fast jedem neu entwickelten Spiel „HDRI-Based-Lighting“ zum Einsatz.

### **3.4. Werbung und Produktvisualisierungen**

Ein fotorealistisches Bild eines Produktes zu präsentieren, ohne dieses kostenaufwändig herstellen zu müssen, birgt viele Vorteile in sich. So kann man schon vor der Entwicklung sehen, wie der Kunde auf das Produkt reagieren wird. Außerdem spielt auch in der Werbung die Kosten- und Zeitreduzierung durch den Einsatz von 3D Grafik eine große Rolle. Im speziellen Autowerbungen kommen nicht mehr ohne Computergrafik aus, da viele Szenen durch den Einsatz eines 3D Modells schneller und günstiger produziert werden können.

## 4. Physikalische Grundlagen der Lichtausbreitung

In folgendem Kapitel werden die Grundlagen der Lichtausbreitung und die dabei entstehenden Probleme, die vom Renderalgorithmus gelöst werden müssen, beschrieben.

### 4.1. Lichtmodelle

In der Geschichte der Physik wurde Licht teils auf Grund experimenteller Befunde als Wellenerscheinung und teils (auf Grund anderer experimenteller Befunde) als Strom von Teilchen angesehen. Einige Phänomene, wie zum Beispiel Strahlenbeugung und Interferenz, können durch ein wellenartiges Model besser beschrieben werden, während andere, wie der Photoelektrische Effekt, besser unter der Annahme, dass Licht aus einem Strom von Teilchen besteht, erklärt werden können.<sup>5</sup>

#### 4.1.1. Quantenoptik

Quantenoptik ist das grundlegendste Model, das den Welle-Teilchen-Dualismus von Licht beschreibt. Die Wechselwirkung von Licht mit Materie wird auf atomarer Ebene beschrieben. Es muss besonders berücksichtigt werden, dass Licht nur in unteilbaren, also gequantelten Energieportionen, den Photonen, absorbiert und emittiert werden kann.<sup>6</sup> Dieses Model wird aber im Allgemeinen für die Erstellung von Computergrafik meist als zu detailliert angesehen und wird üblicherweise nicht verwendet.

#### 4.1.2. Wellenmodel

Das Wellenmodel ist eine Vereinfachung des Quantenmodells und wird mit Hilfe der Maxwell-Gleichung beschrieben. Dieses Model beinhaltet die Berechnung von Phänomenen wie Beugung, Interferenz und Polarisierung, welche auftreten, wenn Licht mit Objekten interagiert, die der Größe der Wellenlänge des Lichts entsprechen.<sup>7</sup> Im täglichen Leben tritt kann man dies zum Beispiel in den hellen

---

<sup>5</sup> Vgl. Dutre/Bekaert/Bala, Advanced Global Illumination (2003), Seite 18

<sup>6</sup> Vgl. Wikipedia, <http://de.wikipedia.org/wiki/Quantenoptik>

<sup>7</sup> Vgl. Dutre/Bekaert/Bala, Advanced Global Illumination (2003), Seite 18

Farben von Ölflecken erkennen. Im Allgemeinen wird aber auch der Ansatz des Wellenmodells, auf Grund zu hoher Komplexität, in der Computergrafik vernachlässigt.

### **4.1.3. Geometrische Optik**

Dieses Model ist das einfachste und somit in der Computergrafik am weitesten verbreitetste. Es wird angenommen, dass die Wellenlänge des Lichts wesentlich kleiner ist als die Größe des Objekts, mit welchem das Licht interagiert. Dieses Model nimmt an, dass Licht emittiert, reflektiert und weitergeleitet wird.<sup>8</sup> Dabei werden einige Annahmen getroffen:

- Licht bewegt sich in geraden Linien.<sup>9</sup> Phänomene die Beugung, bei denen sich Licht um ein Objekt „biegt“, werden vernachlässigt.
- Licht bewegt sich unverzögert durch ein Medium.<sup>10</sup> Diese Annahme setzt voraus, dass sich Licht unrealistischer Weise mit unbegrenzter Geschwindigkeit ausbreitet. Dies führt dazu, dass der Renderalgorithmus einen Dauerzustand der Lichtenergie einer Szene berechnet.
- Licht wird nicht durch äußere Faktoren wie Gravitation oder magnetische Felder beeinflusst.<sup>11</sup>

## **4.2. Radiometrie**

Radiometrie ist die Wissenschaft von der Messung elektromagnetischer Strahlung.<sup>12</sup> Dieses Kapitel soll einen kurzen Überblick über Einheiten der Radiometrie geben, die bei Renderalgorithmen für globale Beleuchtung von Bedeutung sind. Dabei sollte vor allem der Zusammenhang von Radiometrie und Fotometrie beachtet werden. Die Fotometrie setzt die von einem Fotometer gemessene elektromagnetische Strahlungsleistung in Beziehung zum

---

<sup>8</sup> Vgl. Dutre/Bekaert/Bala, *Advanced Global Illumination* (2003), Seite 18

<sup>9</sup> Vgl. Dutre/Bekaert/Bala, *Advanced Global Illumination* (2003), Seite 18

<sup>10</sup> Vgl. Dutre/Bekaert/Bala, *Advanced Global Illumination* (2003), Seite 18

<sup>11</sup> Vgl. Dutre/Bekaert/Bala, *Advanced Global Illumination* (2003), Seite 18

<sup>12</sup> Vgl. Wikipedia, <http://de.wikipedia.org/wiki/Radiometrie>

Sinneseindruck durch das menschliche Auge.<sup>13</sup> Das menschliche Auge kann Licht im Bereich von 380 Nanometer bis 780 Nanometer wahrnehmen.<sup>14</sup> Da fotometrische Werte von entsprechenden radiometrischen Werten abgeleitet werden können, arbeiten Renderalgorithmen ausschließlich mit radiometrischen.

#### 4.2.1. Radiometrische Größen

##### Strahlungsfluss (Strahlungsleistung)

Die grundlegendste radiometrische Größe ist der Strahlungsfluss. Der Strahlungsfluss ist die Strahlungsenergie  $dQ$ , die pro Zeiteinheit  $dt$  von elektromagnetischen Wellen transportiert wird. Sie wird in der SI-Einheit Watt angegeben.

$$\Phi = \frac{dQ}{dt}$$

Man kann also zu Beispiel sagen, dass eine Lichtquelle 50 Watt Strahlungsleistung ausstrahlt, oder dass 20 Watt Strahlungsleistung auf einem Objekt einfallen. Dabei wird weder die Größe der Lichtquelle, oder des Objekts beschrieben, noch wird Aufschluss über die Entfernung zwischen Lichtquelle und Empfänger gegeben.<sup>15</sup>

##### Bestrahlungsstärke

Die Bestrahlungsstärke ( $E$ ) ist für ein kollimiertes Strahlungsfeld definiert als der Strahlungsfluss  $d\Phi$  pro Fläche  $dA$ , wobei der Strahl einen Winkel  $\alpha$  zur Flächennormalen einnimmt.

$$E = \frac{d\Phi}{dA} \cos \alpha$$

Wenn zum Beispiel 50 Watt Strahlungsleistung an einer  $1,25\text{m}^2$  Oberfläche einfallen, beträgt die Bestrahlungsstärke an jedem Punkt der Oberfläche  $40 \text{ Watt/m}^2$ .<sup>16</sup>

---

<sup>13</sup> Vgl. Wikipedia, <http://de.wikipedia.org/wiki/Fotometrie>

<sup>14</sup> Vgl. Dutre/Bekaert/Bala, Advanced Global Illumination (2003), Seite 19

<sup>15</sup> Vgl. Dutre/Bekaert/Bala, Advanced Global Illumination (2003), Seite 19

<sup>16</sup> Vgl. Dutre/Bekaert/Bala, Advanced Global Illumination (2003), Seite 19

### Strahldichte (Radiance)

Die Strahldichte ist die wohl bedeutendste Größe in der globalen Beleuchtungsrechnung. Sie beschreibt den Fluss pro projizierter Einheitsfläche und pro Einheitsraumwinkel, der auf eine Fläche auftrifft oder diese verläßt.

$$L = \frac{1}{\cos \theta} \frac{d^2 \Phi}{dA d\Omega}$$

Die wichtigste Eigenschaft der Radiance ist, dass ihre Größe entlang der Strahlungsrichtung nicht variiert, wenn man von einem perfekten Vakuum ausgeht. Aus der Strahldichte können andere oben genannte radiometrische Größen von abgeleitet werden.<sup>17</sup>

### 4.3. Lichtemission

Licht ist elektromagnetische Strahlung verursacht durch das Beschleunigen von Ladung.<sup>18</sup> Licht kann durch mehrere Wege erzeugt werden; zum Beispiel durch Wärmequellen wie der Sonne, oder durch Quantenphänomene wie Fluoreszenz, wobei Materialien Energie mit einer bestimmten Wellenlänge absorbieren und mit einer unterschiedlichen Wellenlänge wieder ausstrahlen. In der Computergrafik wird auf ein detailliertes quantenmechanisches Model verzichtet. Die meisten Renderalgorithmen nehmen lediglich an, dass Licht von einer Lichtquelle mit einer bestimmten Wellenlänge und Intensität ausgestrahlt wird.<sup>19</sup>

Eine exakte Berechnung von globaler Beleuchtung benötigt die Beschreibung des Streuverhaltens jeder Lichtquelle in der Szene: räumliche, gerichtete und spektrale Streuung. Bei der idealisierten räumlichen Streuung nimmt man an, dass alle Lichtquellen Punktlichter (Point Lights) sind; wirklichkeitsnäher werden Lichter als Flächenlichter (Area Lights) betrachtet. Die gerichtete Streuung einer typischen Lichtquelle wird durch die Form des dazugehörigen Beleuchtungskörpers bestimmt. Obwohl die spektrale Streuung von Licht ebenfalls exakt simuliert werden könnte, verwenden die meisten Renderalgorithmen dafür aus

---

<sup>17</sup> Vgl. Dutre/Bekaert/Bala, Advanced Global Illumination (2003), Seite 20

<sup>18</sup> Vgl. Dutre/Bekaert/Bala, Advanced Global Illumination (2003), Seite 30

<sup>19</sup> Vgl. Dutre/Bekaert/Bala, Advanced Global Illumination (2003), Seite 31

Effizienzgründen RGB Werte. Jede Art des Streuverhaltens kann entweder als Funktion oder als Tabelle beschrieben werden.<sup>20</sup>

#### **4.4. Interaktion von Licht mit Oberflächen**

Licht interagiert in einer Szene mit den verschiedenen Objekten. Dabei wird es an der Oberfläche reflektiert oder weitergeleitet. Ein Teil der Lichtenergie kann auch von der Oberfläche absorbiert werden und in Hitze umgewandelt werden. Dieses Phänomen wird aber normalerweise von den Renderalgorithmen nicht explizit berechnet.<sup>21</sup>

##### **4.4.1. BRDF (Bidirectional Reflectance Distribution Function)**

Materialien interagieren auf verschiedene Weisen mit Licht. Einige treten als spiegelnde Oberflächen auf, andere als diffuse. Die Reflexionseigenschaften eines Objekts beeinflussen das Erscheinungsbild. In den meisten Fällen trifft das Licht am Punkt  $P$ , unter dem Einfallswinkel  $\Theta_i$  auf die Oberfläche und verlässt diese an einem anderen Punkt  $Q$ , unter dem Ausfallswinkel  $\Theta_r$ . Die Funktion, die die Relation von Einfallswinkel und Ausfallswinkel beschreibt nennt sich „*Bidirectional Scattering Surface Reflectance Distribution Function*“ (BSSRDF).<sup>22</sup> In der Computergrafik wird diese Funktion vereinfacht, indem man annimmt, dass der Eintrittspunkt identisch mit dem Austrittspunkt ist. Mit dieser Annahme kann zum Beispiel Volumenstreuung (Sub Surface Scattering), bei der sich Ein- und Austrittspunkt unterscheiden, nicht physikalisch korrekt berechnet werden. Die in der Computergrafik vereinfachte Funktion zur Beschreibung der Reflexionseigenschaften eines Objekts nennt sich „*Bidirectional Reflectance Distribution Function*“ (BRDF).<sup>23</sup> Die *BRDF* ermöglicht eine allgemeine Beschreibung des Reflexionsverhaltens einer Oberfläche. Sie ist definiert als der Anteil der auf einen Punkt  $\vec{x}$  aus einer Richtung  $\vec{\omega}_i$  auftreffenden Radiance, die in eine Richtung  $\vec{\omega}_r$  reflektiert wird.

---

<sup>20</sup> Vgl. Dutre/Bekaert/Bala, Advanced Global Illumination (2003), Seite 31

<sup>21</sup> Vgl. Dutre/Bekaert/Bala, Advanced Global Illumination (2003), Seite 31

<sup>22</sup> Vgl. Dutre/Bekaert/Bala, Advanced Global Illumination (2003), Seite 31

<sup>23</sup> Vgl. Dutre/Bekaert/Bala, Advanced Global Illumination (2003), Seite 31

$$f_r(\vec{x}, \vec{\omega}_i, \vec{\omega}_r) = \frac{L_r(\vec{x}, \vec{\omega}_i)}{E(\vec{x}, \vec{\omega}_r)} = \frac{L_r(\vec{x}, \vec{\omega}_r)}{L_i(x, \vec{\omega}_i)(\vec{N}_{\vec{x}} \vec{\omega}_i) d\vec{\omega}_i}$$

Eine physikalisch plausible *BDRF* besitzt folgende Eigenschaften:

- Wechselwirkung: Ein Vertauschen von Einfalls- und Reflexionsrichtung ändert den Wert der *BRDF* nicht.
- Energieerhaltung: Die Summe, der in alle Richtungen reflektierten Energie ist maximal so groß, wie die auf den Punkt auftreffende Energie.<sup>24</sup>

#### 4.4.2. *BRDF* Arten

Abhängig von den Eigenschaften der *BRDF* kann ein Material als diffuse Oberfläche, als Spiegel, oder als glänzende Oberfläche auftreten.

##### Diffuse Oberfläche (Diffuse Surface)

Bei diffusen Oberflächen wird das einfallende Licht gleichmäßig in den Halbraum vor der reflektierenden Fläche zurückgeworfen. Bei solchen Materialien ist der Wert der *BRDF* konstant für alle Werte des Einfallwinkels<sup>25</sup>, und sieht für den Beobachter von allen möglichen Richtungen gleich aus.

##### Spiegelnde Oberfläche (Specular Surface)

Auf einer perfekten spiegelnden Oberfläche werden alle einfallenden Lichtstrahlen ausschließlich in eine bestimmte Richtung reflektiert.<sup>26</sup>

- Reflexion: Das Reflexionsgesetz besagt, dass der Winkel des einfallenden Strahls zum Lot, gleich dem Winkel des ausfallenden Strahls zum Lot ist.<sup>27</sup> Ein perfekter Reflektor hat nur eine Ausfallsrichtung. In der Natur gibt es keinen absolut idealen Reflektor.
- Brechung: Das snelliussche Brechungsgesetz besagt, dass ein Lichtstrahl seine Richtung ändert, wenn er von einem transparenten Medium in ein anderes transparentes Medium übergeht.<sup>28</sup>

<sup>24</sup> Vgl. Pharr/Humphreys, Physically Based Rendering (2004), Seite 249

<sup>25</sup> Vgl. Dutre/Bekaert/Bala, Advanced Global Illumination (2003), Seite 35

<sup>26</sup> Vgl. Birn, Digital Lighting & Rendering (2006), Seite 33

<sup>27</sup> Wikipedia, [http://de.wikipedia.org/wiki/Reflexion\\_%28Physik%29](http://de.wikipedia.org/wiki/Reflexion_%28Physik%29)

<sup>28</sup> Wikipedia, [http://de.wikipedia.org/wiki/Snelliussches\\_Brechungsgesetz](http://de.wikipedia.org/wiki/Snelliussches_Brechungsgesetz)

### Glänzende Oberflächen (Glossy Surface)

Die meisten Oberflächen sind weder absolut diffus, noch absolut spiegelnd, sondern weisen eine Kombination beider Reflexionsarten auf.

Echte Materialien können sehr komplexe *BRDFs* aufweisen. Zur Umsetzung in der Computergrafik wurden mehrere Modelle entwickelt. So gibt es neben den populären empirischen Modellen wie *Gouraud*, *Phong* oder *Blinn* auch physikalisch fundierte Modelle wie das *Cook-Torrance-Beleuchtungsmodell* oder das *Ward-Beleuchtungsmodell*.<sup>29</sup>



Abbildung 1: Unterschiedliche Reflexionsverhalten

Entnommen aus Jeremy Birn, Digital Lighting & Rendering, Seite 248



Abbildung 2: Gerendertes Bild mit diffuser Reflexion (links), glänzender Reflexion (Mitte) und spiegelnder Reflexion (rechts).

Entnommen aus Jeremy Birn, Digital Lighting & Rendering, Seite 249

## 4.5. Rendergleichung

Bei der *Rendergleichung*, auch Rendering-Gleichung genannt, handelt es sich um eine Integralgleichung, welche die Energieerhaltung bei der Ausbreitung von Lichtstrahlen beschreibt und somit die mathematische Basis für alle Algorithmen zur globalen Beleuchtung bildet. Im Prinzip war alles, was zur Berechnung dreidimensionaler Bilder nötig war, schon lange vor der *Rendergleichung* vorhanden: Die Maxwell'schen Gleichungen, die spezielle Relativitätstheorie und

<sup>29</sup> Vgl. Dutre/Bekaert/Bala, Advanced Global Illumination (2003), Seite 38

die Quantenmechanik erklären die Interaktion von Licht und Materie so exakt, dass mit ihnen theoretisch die Berechnung beliebig realistischer Bilder möglich wäre. Doch wie in Kapitel 4.1 erwähnt, ist es in der Computergrafik völlig unpraktikabel mit solch detaillierten Modellen zu arbeiten, da der Rechenaufwand selbst mit heutiger Hardware nicht zu bewältigen wäre. Es zeigte sich jedoch auch, dass es nicht nötig war, mit derart exakten Modellen zu arbeiten: Die Quantenmechanik erklärt Effekte im Kleinen, die im menschlichen Alltag nicht wahrnehmbar sind und selbst einige Auswirkungen der Maxwell'schen Gleichungen sind für die Praxis der Computergrafik oft bedeutungslos. In der Folge arbeitete man, wie ebenfalls in Kapitel 4.1 erwähnt, mit der geometrischen Optik, die das Verhalten des Lichts im Großen – also unter Vernachlässigung seiner Welleneigenschaften – beschreibt. So entstanden die bekannten Ansätze des *Raytracing* und *Radiosity*, die statt komplexen Wellen einfache Lichtstrahlen durch die Modelle verfolgen.<sup>30</sup>

In diese Entwicklung hinein veröffentlichte James T. Kajiya 1986 die *Rendergleichung*. Kajiya zeigte, dass alle bis dato verbreiteten Rendertechniken direkt aus der *Rendergleichung* hergeleitet werden können. Damit gab es erstmals einen gemeinsamen mathematischen Unterbau, auf dem die Techniken verglichen werden konnten. Die *Rendergleichung* führte in der Folge nicht nur zu einer Systematisierung des Wissensgebietes sondern inspirierte auch zahlreiche Weiterentwicklungen. Sie gilt heute als dermaßen fundamental, dass viele fälschlicherweise annehmen, *Raytracing* sei aus der *Rendergleichung* entstanden oder die vorher entstandene Radiosity-Gleichung sei als Umformung aus ihr hervorgegangen.

### Formel

Die *Rendergleichung* lautet:

$$L(x, x') = g(x, x') \cdot \left( L_e(x, x') + \int_S b(x, x', x'') L(x', x'') dx'' \right)$$

Sie beschreibt, wie viel Licht einen Oberflächenpunkt  $x$  von einem anderen Oberflächenpunkt  $x'$  aus erreicht. Dabei wird ein dritter Oberflächenpunkt  $x''$

---

<sup>30</sup> Vgl. Computerbase, <http://www.computerbase.de/lexikon/Rendergleichung>

berücksichtigt, dessen Licht zunächst auf  $x'$  trifft und von dort aus nach  $x$  reflektiert wird.<sup>31</sup> Die einzelnen Teile haben die folgende Bedeutung:

- $L(x, x')$  gibt an, wie viel Licht  $x$  von  $x'$  aus auf direktem Weg erreicht. Es handelt sich um eine Strahldichte mit der Einheit  $[J / (m^4s)]$ .
- $g(x, x')$  beschreibt die gegenseitige Lage der Punkte in der Szene. Normalerweise hat der Term den Wert  $1/r^2$ , wobei  $r$  die Entfernung von  $x$  und  $x'$  ist. Er gibt dann an, wie viel des von  $x'$  ausgehenden Lichtes  $x$  tatsächlich exakt trifft. Liegt zwischen  $x$  und  $x'$  eine weitere Oberfläche, so ist der Term jedoch 0, das heißt bei  $x$  kommt kein Licht von  $x'$  auf direktem Weg an. Dies gilt auch, falls die dazwischen liegende Oberfläche völlig durchsichtig ist; in diesem Fall nimmt die Oberfläche das Licht auf der einen Seite auf und strahlt es auf der anderen Seite neu aus.
- $L_e(x, x')$  gibt an, wie viel Licht von  $x$  aus nach  $x'$  abgestrahlt wird. Dies ist wiederum eine Strahldichte mit der Einheit  $[J / (m^4s)]$ .
- $b(x, x', x'')$  gibt an, welcher Anteil des Lichts, das  $x'$  von  $x''$  aus erreicht, in Richtung  $x$  reflektiert wird. Es handelt sich hierbei um eine BRDF.
- $S$  ist die Gesamtheit aller Flächen in der Szene.

### Äquivalente Darstellung

Äquivalente Darstellungsformen der *Rendergleichung* werden gewählt, um andere Anwendungsfälle anschaulicher zu beschreiben. Verbreitet ist die folgende Darstellung, die beschreibt, wie viel Licht vom Oberflächenpunkt  $x$  aus in Richtung des Vektors  $\vec{\omega}$  abgestrahlt wird<sup>32</sup>:

$$L(x, \vec{\omega}) = L_e(x, \vec{\omega}) + \int_{\Omega} b(\vec{\omega}', x, \vec{\omega}) L(x, \vec{\omega}') (\vec{\omega}' \cdot \vec{n}) d\vec{\omega}'$$

Die einzelnen Teile haben im Wesentlichen dieselbe Bedeutung wie in der anderen Darstellung, sind jedoch in Abhängigkeit von der Richtung statt eines zweiten oder dritten Punktes gegeben:

<sup>31</sup> Vgl. <http://www.cs.princeton.edu/courses/archive/fall02/cs526/papers/kajiya86.pdf>

<sup>32</sup> Vgl. <http://www.cs.princeton.edu/courses/archive/fall02/cs526/papers/kajiya86.pdf>

- $L(x, \vec{\omega})$  gibt an, wie viel Licht von  $x$  aus in Richtung  $\vec{\omega}$  abgestrahlt wird; es handelt sich auch hier um die Strahldichte. Analoges gilt für den Term  $L(x, \vec{\omega}')$
- $L_e(x, \vec{\omega})$  gibt an, wie viel Licht von  $x$  aus in Richtung  $\vec{\omega}$  ausgestrahlt wird.
- $b(\vec{\omega}', x, \vec{\omega})$  ist eine *BRDF* und gibt an, wie viel Licht im Punkt  $x$  aus der Richtung  $\vec{\omega}'$  in die Richtung  $\vec{\omega}$  reflektiert wird.
- $\vec{\pi}$  ist die Normale der Oberfläche im Punkt  $x$ .
- $\Omega$  ist die Gesamtheit aller Winkel der Hemisphäre über der Oberfläche.

Das Ziel aller globalen Beleuchtungsverfahren ist die Lösung der *Rendergleichung*. Die Gleichung kann im Allgemeinen nicht analytisch gelöst werden. Stattdessen müssen numerische Verfahren zur Lösung des Integrals eingesetzt werden.

## 5. Entwicklung des fotorealistischen Rendering

Im folgenden Kapitel werden die wichtigsten Entwicklungen in der Computergrafik bis zu heutigen Renderingmethoden beschrieben.

### 5.1. Shading

*Shading* bezeichnet in der 3D-Computergrafik im allgemeinen Sinne die Simulation der Oberfläche eines Objekts.<sup>33</sup> Die genauen Eigenschaften der *BDRF* werden beim *Shading* umgesetzt. Den größten Durchbruch auf dem Gebiet des *Shading* schafften Henri Gouraud und Bui-Tong Phong, die beide Interpolationsverfahren anwendeten, um höheren Realismus zu erreichen als mittels *Flat-Shading*, dem einfachsten *Shading* Modell, möglich war.

#### 5.1.1. Flat Shading (Constant Shading)

*Flat Shading*, auch *Constant Shading*, ist das einfachste Schattierungsverfahren. Mit dieser Methode erhält jeder Pixel eines Polygons an Hand der Flächennormale

---

<sup>33</sup> Vgl. Kerlow, *The Art Of 3D Computer Animation and Effects* (2004), Seite 227

die gleiche Farbe und den gleichen Lichtwert. Dies hat eine abgestufte, eckige und unrealistische Erscheinung der Objekte besonders bei gekrümmten Oberflächen zur Folge. Um die facettenartige Darstellung zu vermindern, müssten die Polygone verkleinert werden, was einen erhöhten Rechenaufwand zur Folge hätte. *Flat Shading* findet daher besonders bei Objekten mit ebenen Flächen Anwendung. Polygone, die der Lichtquelle zugewandt sind, werden heller dargestellt als solche, die der Beleuchtungsquelle abgewandt sind. Eine entfernungsabhängige Lichtabnahme ist nicht darstellbar.<sup>34</sup>

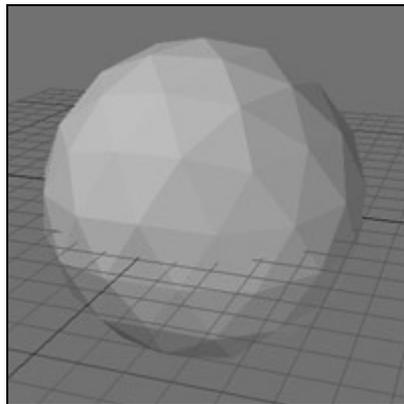


Abbildung 3: Flat Shading

Entnommen aus Wikipedia [http://de.wikipedia.org/wiki/Flat\\_Shading](http://de.wikipedia.org/wiki/Flat_Shading)

### 5.1.2. Gouraud Shading

Beim *Gouraud Shading*, benannt nach dessen Entwickler Henri Gouraud, werden zunächst die Farben des darzustellenden Polygons an seinen Eckpunkten berechnet und die Eckpunkte auf die Bildebene projiziert. Das so entstandene zweidimensionale Abbild des Polygons wird anschließend zeilenweise abgearbeitet. Dabei werden die Farben an den Schnittpunkten der Kanten mit der Abtastzeile aus den Farben der Eckpunkte interpoliert. Die Farbwerte der Bildpunkte der Abtastzeile werden wiederum aus den Farben der Kanten interpoliert.

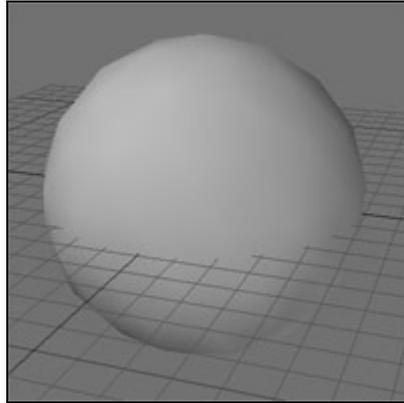
Beim Beleuchtungsverfahren wird beim *Gouraud Shading* lediglich diffuse Reflexion verwendet. Zur Berechnung der Farbwerte eines Vertex werden die Flächennormalen der anliegenden Flächen gemittelt. Die von der Fläche wiedergegebene Farbe wird aus dem diffusen Reflexionskoeffizienten, aus der

---

<sup>34</sup> Vgl. Wikipedia, [http://de.wikipedia.org/wiki/Flat\\_Shading](http://de.wikipedia.org/wiki/Flat_Shading)

Lichtquelle sowie aus dem Winkel zwischen Flächennormale und Lichtstrahl zur Lichtquelle gebildet.

Durch die Interpolation der Flächennormalen erscheinen so facettierte Oberflächen eines dargestellten Objekts nicht kantig wie beim *Flat Shading*, sondern weich. Das *Gouraud Shading* ist eines der schnellsten Verfahren in der 3D-Computergrafik zur Darstellung räumlicher Objekte.<sup>35</sup>



**Abbildung 4: Gouraud Shading**

Entnommen aus Wikipedia [http://de.wikipedia.org/wiki/Gouraud\\_Shading](http://de.wikipedia.org/wiki/Gouraud_Shading)

### **5.1.3. Phong Shading**

*Phong-Shading* wurde 1975 von Bui-Tuong-Phong erstmals vorgestellt und blieb bis heute eines der meist verwendeten Shading-Modelle in der Computergrafik. Es ist besonders zur Beschreibung von glatten, plastikähnlichen Oberflächen geeignet. Auch das Phong-Modell basiert auf keiner physikalischen Grundlage, sondern ist ein vollständig empirisches Modell. So erfüllt es nicht den Energieerhaltungssatz, der besagt, dass es innerhalb eines abgeschlossenen Systems nicht möglich ist, Energie zu erzeugen oder zu vernichten.<sup>36</sup>

Im Phong-Modell wird die Reflexion von Licht als Kombination aus ambienter, ideal diffuser und ideal spiegelnder Reflexion beschrieben.<sup>37</sup>

---

<sup>35</sup> Vgl. Wikipedia, [http://de.wikipedia.org/wiki/Gouraud\\_Shading](http://de.wikipedia.org/wiki/Gouraud_Shading)

<sup>36</sup> Vgl. Wikipedia, <http://de.wikipedia.org/wiki/Energieerhaltungssatz>

<sup>37</sup> Vgl. Kerlow, *The Art Of 3D Computer Animation and Effects* (2004), Seite 228

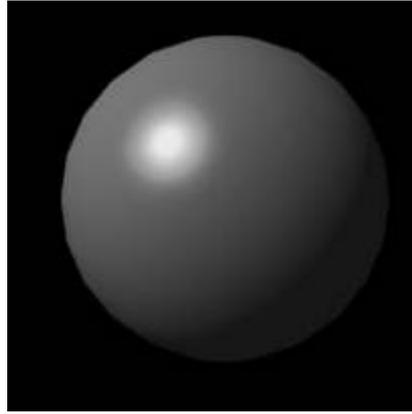


Abbildung 5: Phong Shading

Entnommen aus Wikipedia <http://en.wikipedia.org/wiki/Image:Phong-shading-sample.jpg>

Ein weiterer großer Schritt in Richtung Realität war die Verwendung von Texturen. Durch den Einsatz von Texturen konnten sämtliche Eigenschaften eines Objekts gezielt verändert werden. Neben dem herkömmlichen *Texture Mapping*, welches die Diffuse Farbe eines Objekts beschreibt, kamen *Bump-Mapping*, *Displacement Mapping*, *Specular Mapping*, etc. dazu.

## 5.2. Ray Tracing

Mit *Ray Tracing* wurde Ende der 1960er Jahre von Appel, Goldstein und Nagel der wohl berühmteste Renderalgorithmus eingeführt. *Raytracing* basiert auf der geometrischen Optik, bei der das Licht als eine Gruppe von Strahlen verstanden wird. *Raytracing* war der erste Algorithmus auf diesem Gebiet, der einen gewissen physikalischen Sinn ergab, konnte allerdings nur die direkte Beleuchtung simulieren. Folgende Daten sind zur Berechnung eines Bildes aus einer 3D-Szene notwendig.

- Die Position aller Objekte in der Szene
- Die lokalen Beleuchtungsmodelle (Shader) und deren Parameter
- Die Lichtquellen in der Szene

Außerdem wird beim *Raytracing* auch die Position eines Augpunkts sowie einer Bildebene angegeben. Der Augpunkt entspricht dabei der Position der virtuellen

Kamera. Die Bildebene befindet sich in einiger Entfernung zum Augpunkt und wird in Quadrate unterteilt, die den Pixeln des zu erzeugenden Bildes entsprechen.<sup>38</sup>

### 5.2.1. Verdeckungsrechnung

Als Verdeckungsrechnung wird das Vorgehen bezeichnet, mit dem nicht sichtbare Oberflächen erkannt und aussortiert werden. Die Verdeckungsrechnung ist zum korrekten Rendern einer 3D-Szene notwendig, weil Oberflächen, die vom Betrachter nicht sichtbar sind, auch nicht dargestellt werden sollten. Weiters wird das Renderverfahren dadurch beschleunigt, dass nicht sichtbare Oberflächen von der weiteren Verarbeitung durch den Algorithmus ausgeschlossen werden können.<sup>39</sup>

*Raytracing* arbeitet dabei mit Strahlen (vom englischen „Ray“), die durch die Szene verfolgt werden (vom englischen „Trace“). Es wird für jedes Pixel die Richtung des Strahls berechnet, der vom Augpunkt aus zum entsprechenden Pixel der Bildebene weist. Für jedes Objekt wird dabei der eventuelle Schnittpunkt berechnet, bei dem der Strahl auf das Objekt trifft und die daraus resultierende Entfernung zum Augpunkt. Da die Strahlen nicht wie in der Natur von den Lichtquellen, sondern vom Augpunkt ausgehen, spricht man von „*Backward Ray Tracing*“.<sup>40</sup>

### 5.2.2. Shading

Neben dem Schnittpunkt und der Entfernung des Objekts wird auch die Normale des Objekts am Schnittpunkt ermittelt. Dadurch kann die genaue, reflektierte Lichtstärke am Objekt berechnet werden. In diese Berechnung fließen neben den Eigenschaften der Lichtquelle, auch die Materialeigenschaften (*BRDF* simuliert durch Shader) des Objekts ein.<sup>41</sup>

---

<sup>38</sup> Vgl. Computerbase Lexikon, <http://www.computerbase.de/lexikon/Raytracing>

<sup>39</sup> Vgl. Computerbase Lexikon, <http://www.computerbase.de/lexikon/Sichtbarkeitsproblem>

<sup>40</sup> Vgl. Computerbase Lexikon, <http://www.computerbase.de/lexikon/Raytracing>

<sup>41</sup> Vgl. Computerbase Lexikon, <http://www.computerbase.de/lexikon/Raytracing>

### **5.2.3. Schatten**

Eine Weiterentwicklung von Appel im Jahre 1968 machte es möglich, Lichtstrahlen nicht nur vom Augpunkt, sondern auch von beliebigen anderen Punkten der Szene auszusenden. Dies führte dazu, dass auch Schatten simuliert werden konnten. Vom Schnittpunkt eines Objekts mit dem ersten Strahl wird nun ein weiterer „Schattenstrahl“ in Richtung der Lichtquelle ausgesendet. Trifft dieser „Schattenstrahl“ auf ein weiteres Objekt bevor er die Lichtquelle erreicht, so liegt der erste Schnittpunkt im Schatten.<sup>42</sup>

### **5.2.4. Rekursives Raytracing**

Eine weitere Weiterentwicklung ist das *rekursive Raytracing*, welches die Berechnung von spiegelnden und durchsichtigen Objekten ermöglicht. Dabei werden weitere Lichtstrahlen von den Schnittpunkten ausgesendet.

Bei spiegelnden Oberflächen wird ein „Reflexionsstrahl“ errechnet. Die Richtung dieses Strahls wird durch das Reflexionsgesetz (Einfallswinkel ist gleich Reflexionswinkel) bestimmt.

Bei durchsichtigen Objekten wird ein Strahl ins Innere des betreffenden Objekts ausgesendet. Dabei wird das Brechungsgesetz berücksichtigt.

Da die Sekundärstrahlen auch auf weitere Objekte fallen können, wird der Algorithmus rekursiv aufgerufen, um mehrfache Spiegelungen und Lichtbrechungen zu ermöglichen. Die hierarchische Gesamtheit der Aufrufe wird auch „Renderbaum“ genannt.<sup>43</sup>

### **5.2.5. Diffuses Raytracing**

Beim *diffusen Raytracing* können erstmals weiche Schatten simuliert werden. Statt einem Strahl werden mehrere Strahlen ausgesendet und aus den errechneten Farbwerten wird anschließend der Mittelwert gebildet. Neben weichen Schatten können dadurch auch glänzende Reflexionen und Antialiasing (siehe Kapitel 6.3) erreicht werden. Der Nachteil dabei ist, dass bei diesem Verfahren Bildrauschen

---

<sup>42</sup> Vgl. Computerbase Lexikon, <http://www.computerbase.de/lexikon/Raytracing>

<sup>43</sup> Vgl. Computerbase Lexikon, <http://www.computerbase.de/lexikon/Raytracing>

entsteht, wenn zu wenig Strahlen verwendet werden. Durch Sampling (siehe Kapitel 6) kann dieses Rauschen jedoch wieder reduziert werden.<sup>44</sup>

Obwohl man durch *diffuses Raytracing* bereits komplexe Phänomene wie Spiegelungen und Lichtbrechung simulieren konnte, stellte das Problem der globalen Beleuchtung immer noch eine unlösbare Aufgabe dar. Diffuse Reflexion und Kaustiken können mit den Basisvarianten des *Raytracing* nicht berechnet werden, da lediglich bei spiegelnden Oberflächen Sekundärstrahlen ausgesendet werden, jedoch nicht bei diffusen Oberflächen.

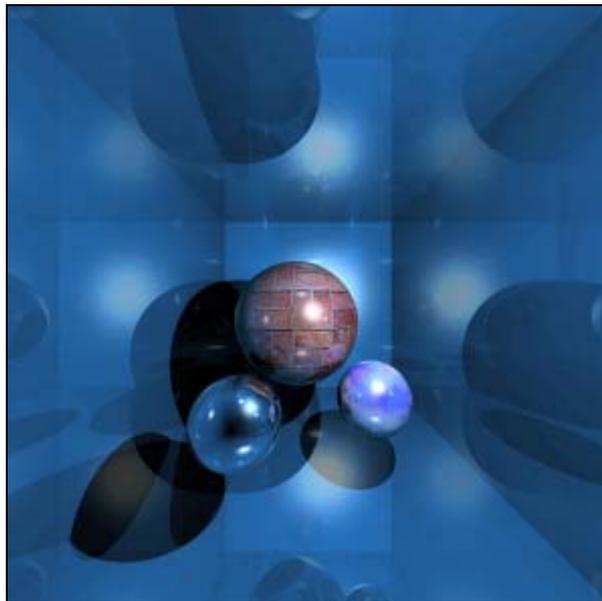


Abbildung 6: Eine typische Raytracing Szene mit vielen Reflexionen

Entnommen aus Wikipedia, [http://en.wikipedia.org/wiki/Ray\\_tracing](http://en.wikipedia.org/wiki/Ray_tracing)

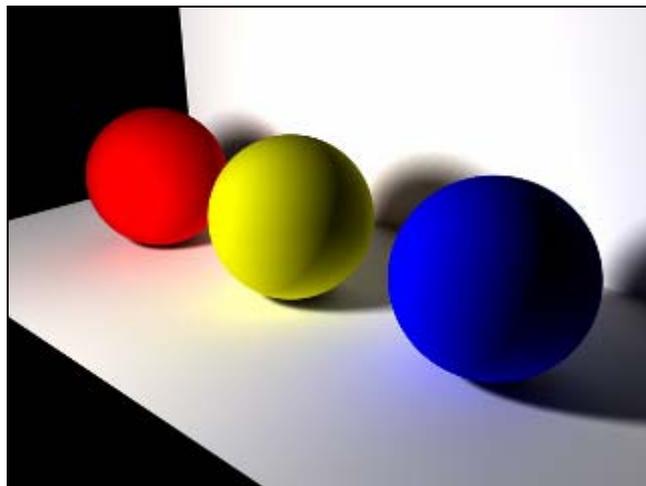
### 5.3. Radiosity

Obwohl *Raytracing* in den frühen 80er sehr populär war, stieß man schnell auf schwerwiegende Grenzen. Effekte wie Color-Bleeding oder diffuse Reflexionen, die durch indirekte Beleuchtung zu Stande kommen, konnten nicht simuliert werden. Man suchte also nach einer weitere Lösung um fotorealistische Bilder erstellen zu können. Die Antwort fand man in einer Finite-Element-Methode mit dem Namen *Radiosity*. Der große Unterschied zum *Raytracing* war, dass *Radiosity* nicht vom Standpunkt und Blickwinkel des Betrachters, sondern von der

---

<sup>44</sup> Vgl. Computerbase Lexikon, <http://www.computerbase.de/lexikon/Raytracing>

ganzen Szene abhängig war. Um die Lichtverteilung beim *Radiosity*-Verfahren zu berechnen wird die gesamte Szene in Flächen aufgeteilt. Für jede Fläche wird eine Gleichung aufgestellt, die das emittierte Licht aus dem von den anderen Flächen empfangenen Licht und eventuell ihrer eigenen Leuchtkraft bestimmt. Insgesamt ergibt sich damit ein Gleichungssystem, dessen Lösung die Helligkeit jeder einzelnen Fläche angibt.<sup>45</sup> Da die Lichtverteilung für die Szene nur einmal berechnet werden muss, stellte dieses Verfahren eine optimale Lösung für interaktive Anwendungen wie virtuelle Rundgänge dar. Die große Stärke von *Radiosity* lag in der diffusen Lichtreflexion. Die Helligkeit und Farbe einer Fläche wurde nicht allein aufgrund der direkten Beleuchtung einer Lichtquelle, sondern auch durch diffus reflektiertes Licht anderer Flächen bestimmt. Lediglich spiegelnde und transparente Objekte konnten mit *Radiosity* nicht dargestellt werden.



**Abbildung 7: Indirekte Beleuchtung; man sieht deutlich, dass das Licht von den Kugeln auf den Boden reflektiert wird**

Entnommen aus Wikipedia [http://de.wikipedia.org/wiki/Radiosity\\_%28Computergrafik%29](http://de.wikipedia.org/wiki/Radiosity_%28Computergrafik%29)

Da alle modernen Renderalgorithmen auf *Raytracing* basieren und *Radiosity* nur mehr in besonderen Einzelfällen, wie Architekturvisualisierungen oder Szenen mit sehr hoch aufgelöster Geometrie, zum Einsatz kommt, wird in dieser Diplomarbeit nicht näher auf diese Methode eingegangen.

---

<sup>45</sup> Vgl. Wikipedia, [http://de.wikipedia.org/wiki/Radiosity\\_%28Computergrafik%29](http://de.wikipedia.org/wiki/Radiosity_%28Computergrafik%29)

## 5.4. Multipass Methoden

Ende der 80er gab es zwei dominierende Arten von Renderalgorithmen: welche, die auf *Raytracing* basierten und welche, die auf *Radiosity* aufbauten. Auf der einen Seite konnten Spiegelungen und Transparenz simuliert werden, auf der anderen Seite wurde indirekte diffuse Beleuchtung berechnet und eine interaktive Manipulation war möglich. Es war daher unvermeidbar, dass Algorithmen entwickelt wurden, die die Vorteile und Charakteristika von *Raytracing* und *Radiosity* vereinen konnten. Diese Algorithmen bestanden üblicherweise aus mehreren Passes (englisch für Durchgang). Eine Multipass Methode besteht meistens aus einem *Radiosity* Pass, welcher die indirekte diffuse Beleuchtung berechnet, gefolgt von einem *Raytracing* Pass, der Spiegelungen und Transparenz berechnet, die auf den Werten des ersten Passes aufbauen.<sup>46</sup> Es sind auch mehrere Durchgänge möglich, wobei jeder Durchgang für einen bestimmten Aspekt der vollständigen Lichtverteilung bestimmt ist. Mittels Multipass Methoden konnte also zum ersten Mal globale Beleuchtung effizient und exakt berechnet werden.

## 6. Sampling und Antialiasing

Das finale Ergebnis eines Renderalgorithmus ist ein zwei-dimensionales Bild, bestehend aus einzelnen Pixel. Die einfallende Strahlung wird aber über die *Rendergleichung*, einer stufenlosen Funktion, definiert. Der Vorgang zur Berechnung von diskreten Pixelwerten (Samples) aus dieser Funktion trägt wesentlich zur Qualität des finalen Bildes bei und kann bei schlechter Umsetzung zu Artefakten führen. Dieses Kapitel soll Aufschluss über die gängigsten Sampling-Methoden, die in der Computergrafik verwendet werden, geben.

### 6.1. Sampling Theorie

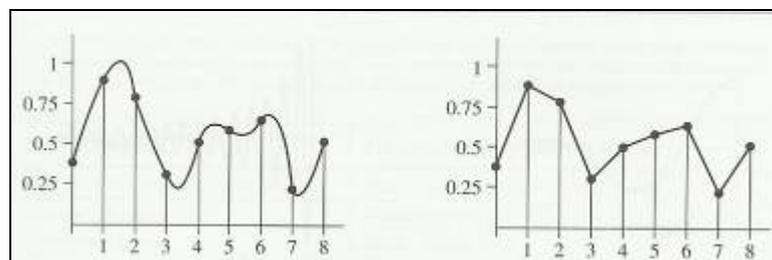
Ein digitales Bild besteht aus mehreren Pixel-Werten. Wird ein Bild, zum Beispiel, auf einem Bildschirm dargestellt, bestimmen diese Werte die Farbe und Helligkeit der Pixel auf dem Display. Dabei muss man zwischen Bildpixel und Displaypixel unterscheiden. Bildschirme verwenden die Werte der Bildpixel um eine neue

---

<sup>46</sup> Vgl. Dutre/Bekaert/Bala, *Advanced Global Illumination* (2003), Seite 8

Funktion zur Darstellung auf dem Display zu berechnen. Der Vorgang, bei dem eine Sammlung von Samplewerten genommen und anschließend wieder zurück zu einer Funktion umgewandelt wird, nennt sich „Rekonstruktion“.<sup>47</sup>

Um die diskreten Pixelwerte des digitalen Bildes zu berechnen, ist es notwendig die stufenlose Funktion abzutasten. Um dabei das bestmögliche Ergebnis zu bekommen, muss eine hohe Abtastrate verwendet werden. Je mehr Samples genommen werden, desto weniger unterscheidet sich die rekonstruierte Funktion von der originalen.



**Abbildung 8: Links die Originalfunktion, Rechts die rekonstruierte Funktion**  
Entnommen aus Pharr/Humphreys, Physically Based Rendering (2004), Seite 280

Da auch bei einer hohen Abtastrate nie die gesamte Information der Funktion gewonnen werden kann, bleibt das Ergebnis immer nur eine Annäherung. Dadurch kommt es beim Rekonstruktionsvorgang zu Fehlern, die sich in der Computergrafik, zum Beispiel, in Form von gezackten Kanten zeigen und „*Aliasing*“ genannt werden, kommen. Um die abzutastende Funktion besser analysieren zu können, wird diese durch eine Fourier-Transformation erst in ihr Frequenzspektrum transformiert.<sup>48</sup>

## 6.2. Aliasing

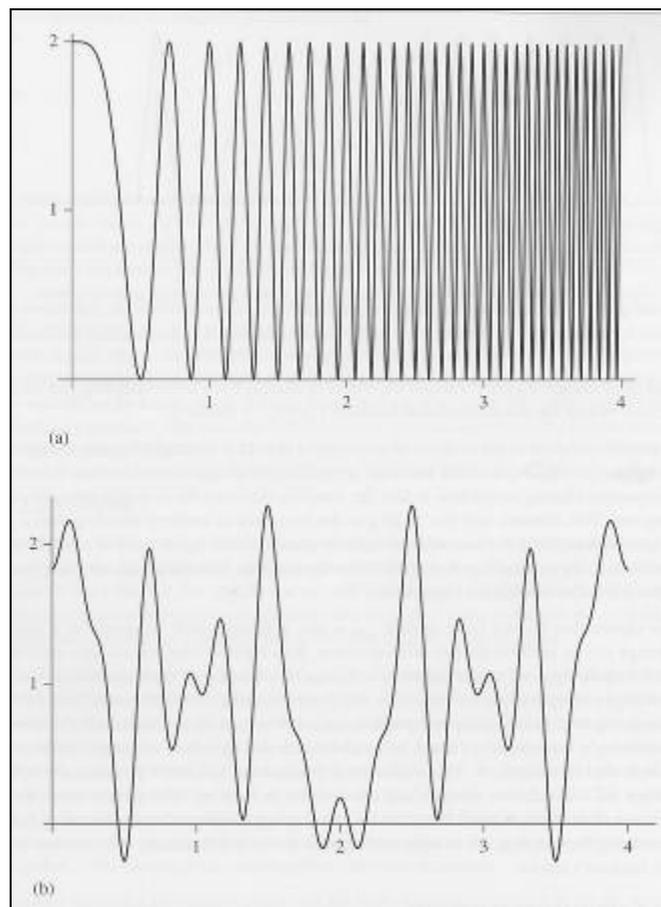
*Aliasing* tritt dann auf, wenn beim Abtasten der Funktion das Abtasttheorem nicht berücksichtigt wird. Jenes besagt, dass ein kontinuierliches, bandbegrenzte Signal mit einer Minimalfrequenz von 0 Hz und einer Maximalfrequenz  $f_{\max}$  mit einer Frequenz größer als  $2 * f_{\max}$  abgetastet werden muss, damit man aus dem

<sup>47</sup> Vgl. Pharr/Humphreys, Physically Based Rendering (2004), Seite 280

<sup>48</sup> Vgl. Pharr/Humphreys, Physically Based Rendering (2004), Seite 280

so erhaltenen zeitdiskreten Signal das Ursprungssignal ohne Informationsverlust (aber mit unendlich großem Aufwand) rekonstruieren bzw. (mit endlichem Aufwand) beliebig genau approximieren kann.<sup>49</sup> Die minimale Abtastfrequenz nennt sich Nyquist-Frequenz.

Liegen also im Eingangssignal Frequenzanteile, die größer sind als die halbe Abtastfrequenz, so werden diese beim rekonstruierten Signal fälschlicher Weise als niedrige Frequenzen interpretiert.



**Abbildung 9: Die Originalfunktion (oben) und die rekonstruierte Funktion (unten).**

**Durch Aliasing gehen die Informationen der hohen Frequenzen verloren und werden fälschlicherweise als niedrige Frequenzen wiedergegeben.**

Entnommen aus Pharr/Humphreys, *Physically Based Rendering* (2004), Seite 290

<sup>49</sup> Vgl. Computerbase Lexikon, <http://www.computerbase.de/lexikon/Nyquist-Shannon-Abtasttheorem>

Dies kann mit einer hohen Abtastrate umgangen werden. Da aber ab einem gewissen Grad der Rechenaufwand dafür zu groß wäre, wurden Methoden entwickelt, um *Alias*-Fehler zu minimieren.

Objektkanten, kleine Geometrie und harte Schatten sind die häufigsten Ursachen für *Aliasing* in einem gerenderten Bild.

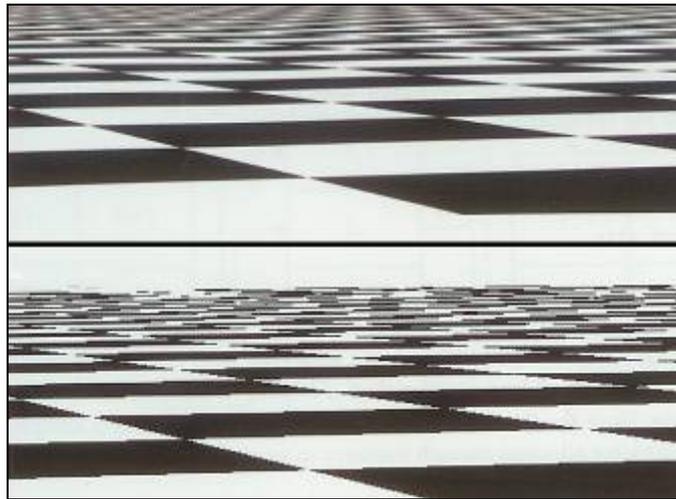


Abbildung 10: Checkerboard mit hoher Abtastrate (oben) und mit geringer (unten).

Die Treppenbildung bei unzureichenden Samples ist klar zu erkennen.

Entnommen aus Pharr/Humphreys, Physically Based Rendering (2004), Seite 305

## 6.3. *Antialiasing*

### 6.3.1. *Nonuniform Sampling*

Beim „*Nonuniform Sampling*“ versucht man die Sichtbarkeit der, durch eine zu niedrige Abtastfrequenz entstandene, Fehler zu reduzieren. Dies wird durch eine ungleiche Verteilung der Abtastpunkte erreicht. Die rekonstruierte Funktion weicht zwar auch bei diesem Verfahren von der Originalfunktion ab, der sichtbare Effekt durch *Aliasing* zeigt sich jedoch als Rauschen, welches für das menschliche Auge weniger ablenkend ist.<sup>50</sup>

### 6.3.2. *Adaptive Sampling*

Ein weiterer Ansatz um *Aliasing* zu minimieren ist „*Adaptive Supersampling*“. Dabei werden jene Bereiche des Signals ermittelt, bei denen die Frequenz höher

---

<sup>50</sup> Vgl. Pharr/Humphreys, Physically Based Rendering (2004), Seite 291

als das Nyquist Limit ist. In diesen Bereichen wird die Abtastfrequenz erhöht, ohne diese in den anderen Bereichen zu verändern.

Unglücklicherweise bringt dieser Ansatz in der Praxis nur bedingt gute Ergebnisse, da es schwer ist die Bereiche, an denen Supersampling notwendig wäre, herauszufiltern. Die meisten Techniken vergleichen angrenzende Samplingwerte miteinander. Bei einer hohen Differenz dieser Werte wird angenommen, dass das Signal in diesem Bereich eine hohe Frequenz aufweist. Im Allgemeinen muss aber eine hohe Differenz der Werte nicht zwangsläufig eine hohe Signalfrequenz zur Folge haben. Selbst wenn die Werte zweier angrenzender Abtastpunkte ident sind, gibt das keinen Aufschluss über den Bereich dazwischen.

„*Adaptive Sampling*“ kann daher nur verwendet werden, um eine hohe Anzahl von *Alias*-Fehlern in einem Bild zu reduzieren, ist aber üblicherweise nicht im Stande ein fehlerloses Bild effizienter zu erstellen, als mit einer Erhöhung der allgemeinen Abtastfrequenz möglich wäre; im Speziellen bei komplexen Szenen.<sup>51</sup>

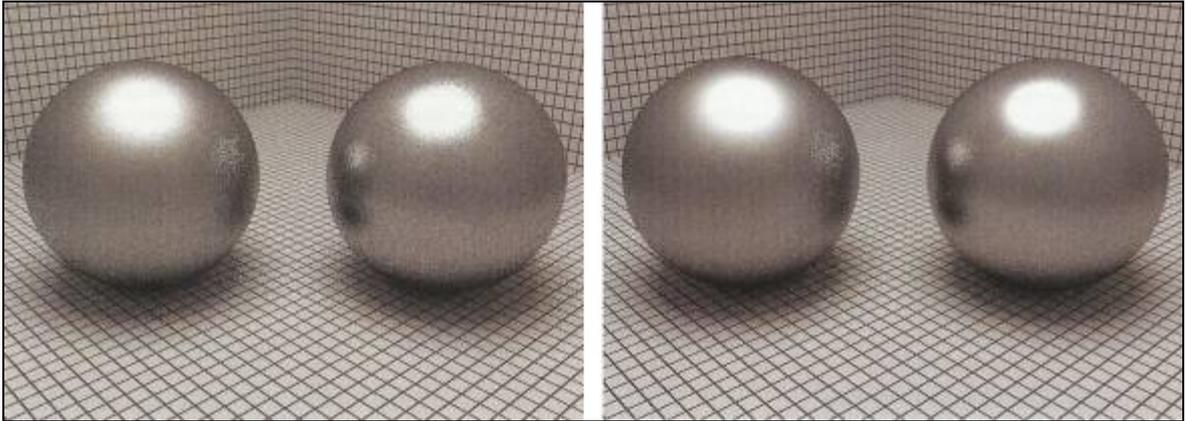
### **6.3.3. Prefiltering**

Beim *Prefiltering* versucht man *Aliasing* zu reduzieren, indem man die originale Funktion so filtert, dass im Signal keine Frequenzanteile, die bei der gewählten Abtastfrequenz zu Fehler führen würden, mehr vorhanden sind. Obwohl bei diesem Verfahren die Charakteristik der Originalfunktion, durch das Entfernen von Information, verändert wird, ist das Ergebnis weniger störend als *Alias*-Fehler.<sup>52</sup>

---

<sup>51</sup> Vgl. Pharr/Humphreys, *Physically Based Rendering* (2004), Seite 291

<sup>52</sup> Vgl. Pharr/Humphreys, *Physically Based Rendering* (2004), Seite 292



**Abbildung 11: Szene einmal mit einem ineffektiven Sampler (links) und einem verbesserten Sampler (rechts) gerendert. Bei beiden wurde die gleiche Anzahl von Samples verwendet. Der Qualitätsunterschied zeigt sich vor allem an den Kanten der Glanzpunkte.**  
Entnommen aus Pharr/Humphreys, *Physically Based Rendering* (2004), Seite 297

## 7. Monte Carlo Methoden

Zum Lösen der *Rendergleichung* werden im Wesentlichen zwei unterschiedliche Techniken eingesetzt. Ein Ansatz besteht darin, Finite-Elemente-Methoden anzuwenden, ein zweiter Ansatz besteht in der Anwendung von Punkt-Sampling-Techniken (Monte-Carlo). Monte Carlo Techniken werden häufig zur Approximation eines Integrals eingesetzt, wenn die Berechnung der genauen Lösung zu aufwendig oder nicht realisierbar ist. In der Computergraphik betrifft dies den Teil der *Rendergleichung*, der die Ermittlung der auf einen Punkt, aus allen Richtungen des Raumes auftreffenden Radiance beschreibt.

### 7.1. Grundidee

Die Grundidee der Monte-Carlo-Integration besteht darin, dass ein Integral

$$I := \int_0^1 f(x) dx$$

beliebig genau approximiert werden kann, indem man eine große Zahl von Positionen im Definitionsbereich der Funktion stochastisch bestimmt und die zugeordneten Funktionswerte gewichtet aufsummiert:

$$\langle I \rangle_N := \frac{1}{N} \sum_{i=1}^N f(\xi_i)$$

Der dabei erwartete Wert errechnet sich aus:

$$E(\langle I \rangle_N) = \int_0^1 \cdots \int_0^1 \frac{1}{N} \sum_{i=1}^N f(x_i) dx_1 \cdots dx_N = I$$

Als Varianz von  $\langle I \rangle_N$  ergibt sich:

$$\begin{aligned} V(\langle I \rangle_N) &= \int_0^1 \cdots \int_0^1 \left[ \frac{1}{N} \sum_{i=1}^N f(x_i) \right]^2 dx_1 \cdots dx_n - I^2 \\ &= \frac{1}{N} \int_0^1 f^2(x) dx - \frac{1}{N} I^2 \\ &= \frac{1}{N} V(\langle I \rangle_1) \quad . \end{aligned}$$

Varianz äußert sich in der Computergrafik als Bildrauschen. Ein großer Nachteil von Monte-Carlo-Methoden ist die langsame Annäherungsrate von  $\frac{1}{\sqrt{N}}$ , wobei  $N$  die Anzahl der verwendeten Samples angibt. Um also beispielsweise den Standardfehler zu halbieren, muss man die Zahl der Samples vervierfachen. Aus diesem Grund versucht man, vorhandene Informationen über die Gestalt des Integranden zur Reduktion der Varianz einzusetzen.<sup>53</sup>

## 7.2. Stratified Sampling

Liegen keine näheren Informationen über den Integranden vor, sollten die Sample-Positionen möglichst gleichmäßig verteilt werden, um sicherzustellen, dass alle Bereiche des Definitionsbereichs adäquat berücksichtigt werden. Um eine zufällige lokale Häufung von Abtastpunkten zu vermeiden wird beim *Stratified Sampling* der Integrationsbereich in  $N$ , sich nicht überschneidende, Teilbereiche  $A_i$  ( $i = 1, \dots, N$ ) aufgeteilt. Die zufällige Auswahl der Abtastpunkte wird auf je eine dieser Bereiche beschränkt. Jeder dieser Teilbereiche, auch Strata genannt, wird dabei unabhängig betrachtet. Das ursprüngliche Integral wird dabei durch die Summe von Integralen über die Teilbereiche ersetzt:

$$I = \int_0^1 f(x) dx = \sum_{i=1}^N \int_{A_i} f(x) dx = \sum_{i=1}^N I_i$$

Durch dieses Vorgehen erreicht man ein gleichmäßigeres Abtasten der Funktion und eine Reduzierung der Standardabweichung.<sup>54</sup>

<sup>53</sup> Vgl. Pharr/Humphreys, Physically Based Rendering (2004), Seite 636

<sup>54</sup> Vgl. Pharr/Humphreys, Physically Based Rendering (2004), Seite 666

### 7.3. Importance Sampling

Ein weiteres Verfahren zur Reduzierung der Varianz ist das *Importance Sampling*. Dabei werden die Samples nicht völlig zufällig im Integrationsbereich, sondern gemäß einer *Probability-Density-Function (PDF)*, verteilt, die möglichst ähnliches Aussehen wie die zu integrierende Funktion haben sollte. Dabei werden Samples in Bereichen konzentriert, von denen angenommen wird, dass sie auf das Ergebnis der Berechnung einen vergleichsweise größeren Einfluss haben.

$$I = \int_0^1 f(x)dx = \int_0^1 \frac{f(x)}{p(x)} p(x)dx$$

Eine gut gewählte Verteilung kann zu einer erheblich schnelleren Konvergenz des Verfahrens führen. Der optimale Fall, einer zur Funktion proportionalen *PDF*, lässt sich in der Praxis nicht erreichen, da in diesen Fall der Integrand selbst schon vollständig bekannt sein müsste. Eine schlecht gewählte *PDF* kann die Varianz allerdings auch vergrößern, so dass die Effizienz des Verfahrens unter die einer vollkommen zufälligen Verteilung sinken kann.<sup>55</sup>

## 8. Monte Carlo Algorithmen zur Beleuchtungsberechnung

### 8.1. Pathtracing

*Pathtracing* war der erste Monte-Carlo-Algorithmus in der Computergrafik, der zur Lösung der *Rendergleichung* verwendet wurde. Er wurde von J. T. Kajiya zeitgleich mit dessen *Rendergleichung* vorgestellt. Die auf jeden Pixel der Bildebene einfallende Lichtenergie wird dabei mit Hilfe von „Lichttransportpfaden“ ermittelt. Diese Pfade gehen, wie beim *Raytracing*, vom Auge des Betrachters aus und enden an den Lichtquellen der Szene. Im Unterschied zu klassischem *Raytracing* werden aber nicht an jedem Schnittpunkt des Pfades mit der Geometrie ein perfekter spiegelnder Reflexions- und ein Brechungsstrahl weiterverfolgt, sondern nur genau ein Strahl, dessen Richtung durch *Importance-Sampling* auf der Objektoberfläche ermittelt wird. Dieser Sample-Strahl dient der Bestimmung der von allen anderen Raumpunkten auf den Oberflächenpunkt einfallenden

---

<sup>55</sup> Vgl. Pharr/Humphreys, *Physically Based Rendering* (2004), Seite 673

Lichtenergie. Zusätzlich werden ausgehend von jedem Schnittpunkt, ähnlich wie beim *Raytracing*, Schattenstrahlen zur Lichtquelle berechnet, die anschließend in die Berechnung miteinbezogen werden.<sup>56</sup>

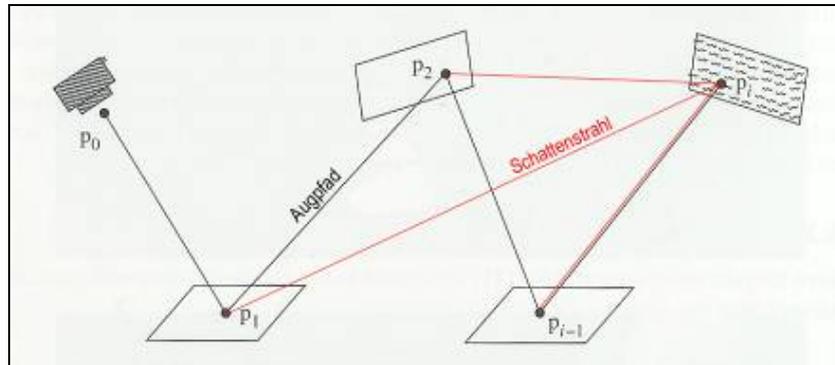


Abbildung 12: Ein Pfad wird von der Kamera  $P_0$  über mehrere Oberflächen  $P_1$ - $P_{i-1}$  bis zur Lichtquelle  $P_i$  verfolgt.

Um die Varianz des Verfahrens zu reduzieren wird *Importance Sampling* eingesetzt. Dabei sollte die *PDF*, zur Bestimmung der verfolgten Strahlen möglichst gleiche Gestalt wie der Integrand haben. Dieser ist im Wesentlichen von der einfallenden Strahlung sowie der *BRDF* der Objektoberfläche abhängig. Um die direkte Beleuchtung zu berücksichtigen, wird jeweils eine der Lichtquellen ausgewählt und für diese der direkte Lichteinfall berechnet. Dies ist eigentlich nicht Teil des reinen *Pathtracing*-Algorithmus, da man die Lichtquellen theoretisch auch ausschließlich über das rekursive Verfolgen des Pfades treffen kann. Allerdings ist das direkte Sampling von Lichtquellen, aufgrund deren verhältnismäßig geringen Größe, in der Praxis notwendig, da die Wahrscheinlichkeit zufällig kleine Lichtquellen zu treffen sehr gering ist. Somit würde nur eine geringe Anzahl von ausgesendeten Pfaden zur Berechnung der Lichtenergie beitragen, was zu einem hohen Rauschen im berechneten Bild führen würde. Das Sampling der direkten Beleuchtung führte zum „*Bidirectional Pathtracing*“ (siehe Kapitel 8.2), einer Weiterentwicklung des *Pathtracing*-Algorithmus.<sup>57</sup>

Um die indirekte Beleuchtung zu berücksichtigen, wird gemäß der Oberflächen-*BRDF* ein Reflexionspfad generiert und rekursiv weiterverfolgt. Theoretisch wird das Licht innerhalb einer Szene unendlich oft reflektiert. Um eine endlose

<sup>56</sup> Vgl. Pharr/Humphreys, *Physically Based Rendering* (2004), Seite 743f

<sup>57</sup> Vgl. Dutre/Bekaert/Bala, *Advanced Global Illumination* (2003), Seite 109

Verfolgung der Pfade zu verhindern, muss eine Abbruchbedingung eingebaut werden. Die einfachste Methode dazu wäre, die Rekursion nach einer bestimmten Anzahl von Reflexionen abubrechen. Die gewählte Tiefe der Rekursion hängt dabei von der Art der Szene ab. Szenen mit vielen spiegelnden Oberflächen benötigen eine lange Verfolgung des Pfades, während bei Szenen mit überwiegend diffusen Oberflächen eine kurze Verfolgung ausreichend ist. Da jedoch durch diesen Abbruch der Erwartungswert des Verfahrens verändert wird, steuert man die Tiefe der Rekursion über ein Verfahren, welches „Russisches Roulette“ genannt wird. Dabei wird die Wahrscheinlichkeit für einen Abbruch der Rekursion über die, approximativ bestimmte, Absorptionswahrscheinlichkeit auf der jeweils betrachteten Objektoberfläche ermittelt.<sup>58</sup>

Der *Pathtracing*-Algorithmus konvergiert für eine steigende Anzahl betrachteter Pfade. Allerdings ist der Nachteil eine sehr geringe Konvergenzgeschwindigkeit, wodurch sehr viele Samples benötigt werden, um akzeptable, rauschfreie Bilder zu generieren. Besondere Schwächen hat der Algorithmus bei der Berechnung von Szenen, in denen indirekte Beleuchtung dominiert. Hierbei kann es beliebig unwahrscheinlich sein, zufällig Oberflächen auf dem Pfad zu treffen, über die viel Licht reflektiert wird, die aber selber nicht emissiv sind.<sup>59</sup>



Abbildung 13: Sponza Atrium Model ausschliesslich mit direktem Licht gerendert.

---

<sup>58</sup> Vgl. Dutre/Bekaert/Bala, *Advanced Global Illumination* (2003), Seite 109

<sup>59</sup> Vgl. Dutre/Bekaert/Bala, *Advanced Global Illumination* (2003), Seite 113



Abbildung 14: Sponza Atrium Model gerendert mit 8 Samples pro Pixel.



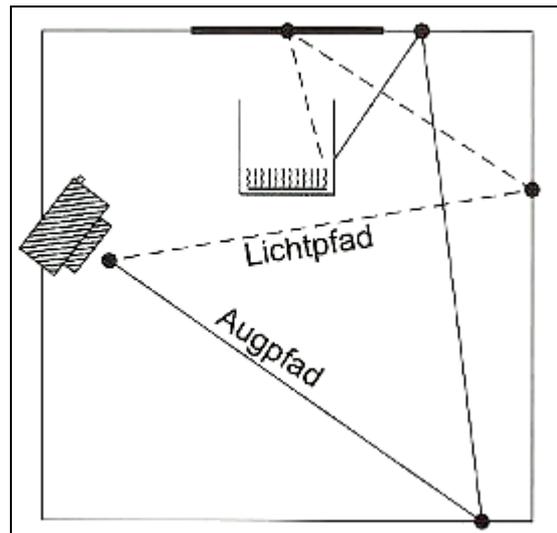
Abbildung 15: Sponza Atrium Model gerendert mit 1024 Samples pro Pixel.  
Entnommen aus Pharr/Humphreys, Physically Based Rendering (2004), Seite 744

## 8.2. Bidirectional Pathtracing

Obwohl *Pathtracing* globale Beleuchtung physikalisch korrekt berechnet, ist eine hohe Anzahl von Samples, und somit eine hohe Rechenzeit notwendig, um speziell in Szenen mit überwiegender indirekter Beleuchtung, rauscharme Bilder zu berechnen. Abbildung 16 zeigt eine typische Situation, welche mit *Pathtracing* schwer zu berechnen ist. Eine Lichtquelle beleuchtet lediglich einen kleinen Bereich der Decke. Der Rest des Raumes wird ausschließlich über Lichtreflexion von der Decke beleuchtet. Eine große Anzahl der, von der virtuellen Kamera ausgehenden, Pfade wird den kleinen Bereich an der Decke nicht antreffen und somit nicht in die Beleuchtungsberechnung einbezogen werden, was zu einer hohen Varianz im generierten Bild führt.<sup>60</sup>

---

<sup>60</sup> Vgl. Pharr/Humphreys, Physically Based Rendering (2004), Seite 753



**Abbildung 16: Die gezeigte Lichtsituation kann mit herkömmlichen Pathtracing nur schwer berechnet werden.**

Entnommen aus Pharr/Humphreys, Physically Based Rendering (2004), Seite 753

Beim *Bidirectional Pathtracing* werden nicht nur Pfade ausgehend vom Auge des Betrachters, sondern zusätzliche Pfade von der Lichtquelle aus verfolgt. Ein vollständiger Lichttransportpfad entsteht dabei durch Kombination beider Pfadtypen. Eine hohe Reduzierung der Varianz erzielt man durch weitere Kombinationen von verschiedenen Aug- und Lichtpfaden, ohne diese erneut berechnen zu müssen. Für jeden vollständigen Pfad der Länge  $n$  existieren dabei  $n + 1$  Möglichkeiten, diesen durch Kombination von Aug- und Lichtpfad zu generieren. Jede der  $n + 1$  Kombinationsmöglichkeiten kann als separates Verfahren zum Schätzen des Lichttransports zum Auge interpretiert werden, wobei alle Verfahren denselben Erwartungswert aufweisen, allerdings mit zum Teil deutlich unterschiedlicher Varianz.<sup>61</sup>

<sup>61</sup> Vgl. Pharr/Humphreys, Physically Based Rendering (2004), Seite 754



Abbildung 17: Reines Path Tracing erzeugt hohes Rauschen.



Abbildung 18: Optimiertes Bild gerendert mit Bidirectional Pathtracing.

Entnommen aus <http://graphics.stanford.edu/courses/cs348b-02/lectures/lecture15/walk037.html>

### **8.3. Irradiance Caching**

Renderlösungen, die auf reinen Monte-Carlo-Verfahren basieren, benötigen eine lange Berechnungszeit um artefaktfreie Bilder generieren zu können. Ein approximativer Lösungsansatz zur effizienten Berechnung der auf einen Oberflächenpunkt einfallenden Strahlung wurde 1988 von G.J. Ward, F.M. Rubinstein und R.D. Clear im SIGGRAPH Paper „A Ray Tracing Solution for

Diffuse Interreflektion“ vorgestellt. *Irradiance Caching* basiert auf der Beobachtung, dass indirektes Licht, im Gegensatz zum direkten Licht, welches sich oftmals von Punkt zu Punkt verändert (zum Beispiel harte Schatten), zwischen benachbarten Raumpunkten relativ gering variiert. Die Grundidee besteht darin, die indirekte Beleuchtung nur für wenige Raumpunkte in hoher Präzision zu berechnen und für alle anderen Raumpunkte zwischen diesen Ergebnissen zu interpolieren.<sup>62</sup>

Zur Berechnung der indirekten Irradiance für eine Punkt  $P$  wird die Hemisphäre um den Punkt  $P$  herum in  $M \times N$  Sektoren unterteilt. Durch Anwendung von stratifiziertem Monte-Carlo-Sampling der Hemisphäre ergibt sich die indirekte Beleuchtung durch:

$$E = \frac{\pi}{M * N} \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} L_{j,k}$$

Die resultierende Irradiance wird zusammen mit dem Oberflächenpunkt  $P$  und der Oberflächennormalen für diesen Punkt in einem Cache gespeichert, der intern eine Octree-Datenstruktur nutzt, um später effizient nützliche Informationen innerhalb des Caches zu lokalisieren.<sup>63</sup>

Immer, wenn im Laufe einer Beleuchtungsrechnung die indirekte Beleuchtung für einen gegebenen Raumpunkt benötigt wird, wird zunächst der Cache nach exemplarischen Daten befragt, die sich für diesen Oberflächenpunkt zur Interpolation von Irradiance-Werten nutzen lassen. Da sich die indirekte Beleuchtung in der Regel nicht abrupt ändert, führt die Interpolation nur zu einem geringen Fehler in der Berechnung. Ein neuer Wert wird lediglich generiert, wenn nicht ausreichend viele Daten für die Interpolation zur Verfügung stehen. Der Fehler bei der Berechnung der Irradiance im Oberflächenpunkt  $P$  bei Approximation durch einen im Cache gespeicherten Oberflächenpunkt  $P_i$  lässt sich durch folgende Gleichung abschätzen:

---

<sup>62</sup> Vgl. Pharr/Humphreys, *Physically Based Rendering* (2004), Seite 755f

<sup>63</sup> Vgl. Pharr/Humphreys, *Physically Based Rendering* (2004), Seite 756

$$\varepsilon_i(P) = \frac{\|P - P_i\|}{R_i} + \sqrt{1 - N_P * N_{P_i}}$$

wobei

- $R_i$  der mittlere Abstand aller Oberflächenpunkte zum Samplepunkt  $P_i$  ist;
- $N_P$  und  $N_{P_i}$  die Oberflächennormalen in den Punkten  $P$  und  $P_i$  sind.<sup>64</sup>

Die Irradiance am Punkt  $P$  wird anschließend durch Interpolation von benachbarten Samplewerten gewichtet berechnet. Punkte mit großen Fehlern werden weniger stark berücksichtigt als Punkte mit geringer Abweichung. Dadurch ergibt sich eine interpolierte Irradiance von

$$E(P) = \frac{w_i(P)E_i(P)}{\sum w_i(P)}$$

wobei

$$w_i(P) = \frac{1}{\varepsilon_i(P)} .^{65}$$

Bei jeder neuen Berechnung der Irradiance an einem Punkt, wird der Cache nach benachbarten Samples durchsucht. Eine benutzerdefinierte Variable  $a$  regelt dabei den Radius des Bereiches, in dem nach vorhandenen Samples gesucht wird. Werden passende Samples gefunden, wird ein neuer Irradiancewert interpoliert. Können keine passenden Werte im Cache gefunden werden, wird durch *Pathtracing* ein neuer Wert am Punkt  $P$  berechnet. Dieser wird wiederum im Cache gespeichert.<sup>66</sup>

Mittels *Irradiance Caching* lassen sich Bilder generieren, die keine typischen Varianzartefakte eines reinen Monte-Carlo-Verfahrens aufweisen. Die verkürzte Berechnungszeit durch Interpolation bringt aber auch Nachteile mit sich. Kann man sich bei einem, durch reine Monte-Carlo-Methoden berechneten, rauschfreien Bild sicher sein, dass eine physikalisch korrekte Szenenbeleuchtung wiedergegeben wird, ist dies beim *Irradiance Caching* nicht mehr der Fall. Dieser

---

<sup>64</sup> Vgl. Dutre/Bekaert/Bala, *Advanced Global Illumination* (2003), Seite 220

<sup>65</sup> Vgl. Dutre/Bekaert/Bala, *Advanced Global Illumination* (2003), Seite 221

<sup>66</sup> Vgl. Dutre/Bekaert/Bala, *Advanced Global Illumination* (2003), Seite 222

Fehler in der Berechnung wird im Englischen „*Bias*“ genannt. Reine Monte-Carlo-Verfahren werden hingegen als „*unbiased*“ bezeichnet.

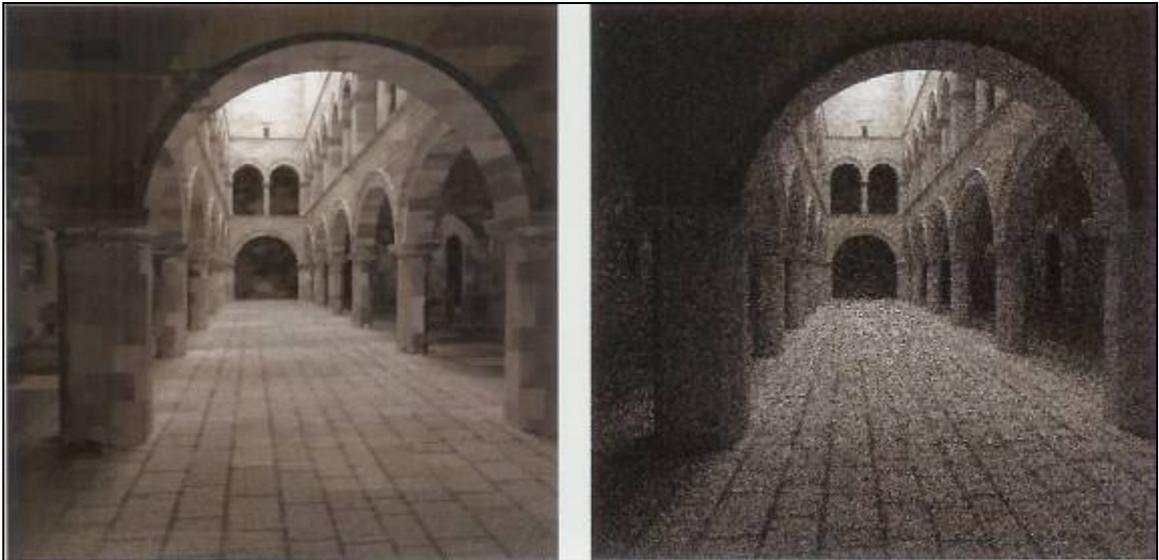


Abbildung 19: Sponza Atrium Szene gerendert mit Irradiance Caching (links) und Path Tracing (rechts). Beide Berechnungen wurden nach gleicher Zeit gestoppt. Path Tracing produziert hochfrequentes Rauschen, während die Artefakte beim Irradiance Caching kreisförmigen Flecken ähneln.



Abbildung 20: Die Position, an denen Samplewerte für den Cache berechnet wurden; hauptsächlich an Ecken mit hoher Variation der indirekten Beleuchtung.

Entnommen aus Pharr/Humphreys, Physically Based Rendering (2004), Seite 759 und 760

## 9. Metropolis Light Transport

### 9.1. Grundidee

1997 stellten Eric Veach und Leonidas J. Guibas einen neuen Renderalgorithmus vor, der auf den Grundsätzen des *Metropolis Sampling* basierte. Um ein Bild zu generieren, werden, wie beim *Bidirectional Pathtracing* (siehe Kapitel 8.2), einzelne Pfade von einer Lichtquelle zur Kamera verfolgt und abgetastet. Beim *Metropolis Light Transport* erfolgt, im Unterschied zu den bisher genannten Algorithmen, allerdings eine Gewichtung der Pfade. So werden Pfadkombinationen, über welche viel Energie übertragen wird, nicht sofort wieder verworfen, sondern weiterverwendet. Jeder Pfad  $\bar{x}$  besteht dabei aus einer Sequenz von Punkten  $x_0 x_1 \dots x_k$  auf den jeweiligen Objektoberflächen, wobei  $k \geq 1$  die Länge des Pfades angibt. Die Nummerierung der Punkte entlang des Pfades folgt der Richtung des Lichtflusses.<sup>67</sup>

Es wird eine Funktion  $f$  definiert, welche die Strahlungsleistung einer Lichtquelle entlang eines Pfades bis zur Bildebene beschreibt. Diese Funktion wird auch „*Image Contribution Function*“ genannt.

Die einzelnen Pfade werden dabei je nach deren Beitrag zum finalen Bild, proportional zu  $f$  gewichtet. Um dies zu erreichen, wird eine Reihe von Pfaden  $\bar{X}_0, \bar{X}_1, \dots, \bar{X}_N$  erzeugt, wobei  $\bar{X}_i$  aus einer zufälligen Mutation von  $\bar{X}_{i-1}$  hervorgeht. Jede Mutation kann dabei abgelehnt werden, was zu  $\bar{X}_i = \bar{X}_{i-1}$  führt.<sup>68</sup>

Nach jedem gesampelten Pfad, wird das generierte Bild (welches im Speicher als zwei-dimensionales Array abgelegt ist) aktualisiert. Dazu wird der Wert des, dem Pfad  $\bar{X}_i$  entsprechenden, Bildpunktes  $(u, v)$  erneuert, wobei alle Samples gleich gewichtet werden. Der *Metropolis Light Transport Algorithmus* könnte als Pseudocode folgendermaßen beschrieben werden<sup>69</sup>:

---

<sup>67</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 2

<sup>68</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 2

<sup>69</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 2

```

 $\bar{x} \leftarrow \text{INITIALPATH}()$ 
image  $\leftarrow$  {array of zeros}
for i  $\leftarrow$  1 to N
   $\bar{y} \leftarrow \text{MUTATE}(\bar{x})$ 
  a  $\leftarrow \text{ACCEPTPROB}(\bar{y} | \bar{x})$ 
  if  $\text{RANDOM}() < a$ 
    then  $\bar{x} \leftarrow \bar{y}$ 
  RECORDSAMPLE(image,  $\bar{x}$ )
return image

```

## 9.2. Metropolis Sampling

Im Jahre 1953 wurde von Metropolis, Rosenbluth, Rosenbluth, Teller und Teller ein Algorithmus zur Lösung von schwierigen Abtastproblemen vorgestellt.

Man nehme einen definierten Raum  $\Omega$ , eine nicht-negative Funktion  $f : \Omega \rightarrow R^+$  und einen Ausgangszustand  $\bar{X}_0 \in \Omega$ . Das Ziel ist einen zufälligen Durchgang  $\bar{X}_0, \bar{X}_1, \dots$  zu generieren, so dass  $\bar{X}_i$  letztendlich, unabhängig vom Ausgangszustand  $\bar{X}_0$ , proportional zu  $f$  gewichtet ist.

Jedes Sample  $\bar{X}_i$  wird dabei aus einer zufälligen Veränderung von  $\bar{X}_{i-1}$  gewonnen. Die Wahrscheinlichkeit eines Wechsels von Zustand  $\bar{x}$  nach Zustand  $\bar{y}$  wird dabei durch die „Transition Function“  $K(\bar{y} | \bar{x})$  angegeben, wobei für alle  $\bar{x} \in \Omega$  gilt, dass  $\int_{\Omega} K(\bar{y} | \bar{x}) d\mu(\bar{y}) = 1$ .<sup>70</sup>

### 9.2.1. Detailed Balance

Innerhalb eines physikalischen Systems wird die „Transition Function“  $K$  üblicherweise durch die physikalischen Gesetze des gleichen Systems bestimmt. Beim Metropolis Algorithmus wird  $K$  so gewählt, dass die Gewichtung  $p_i$  proportional zur gegebenen Funktion  $f$  ist und sich dieser möglichst schnell annähert.<sup>71</sup>

<sup>70</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 2

<sup>71</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 3

Aus  $\bar{X}_{i-1}$  ergibt sich  $\bar{X}_i$  folgendermaßen: Es wird ein vorläufiges Sample  $\bar{X}'_i$  gewählt, welches durch eine provisorische „*Transition Function*“  $T$  definiert ist. Dieser vorläufige Abtastwert wird, gemäß einer Annahmewahrscheinlichkeit  $a(\bar{y} | \bar{x})$ , entweder angenommen oder abgelehnt. Um  $a(\bar{y} | \bar{x})$  korrekt zu definieren, wird angenommen, dass  $p_i$  bereits proportional zu  $f$  ist. Um diesen Zustand zu halten, muss die Wahrscheinlichkeitsdichte eines Wechsels von  $\bar{x}$  nach  $\bar{y}$  gleich groß mit der von  $\bar{y}$  nach  $\bar{x}$  sein:

$$f(\bar{x})T(\bar{y} | \bar{x})a(\bar{y} | \bar{x}) = f(\bar{y})T(\bar{x} | \bar{y})a(\bar{x} | \bar{y})$$

Dieser Zustand wird „Detailed Balance“ genannt.<sup>72</sup>

### 9.2.2. Annahmewahrscheinlichkeit

Die Annahmewahrscheinlichkeit  $a(\bar{y} | \bar{x})$  einer Mutation wird folgendermaßen definiert:

$$a(\bar{y} | \bar{x}) = \min \left\{ 1, \frac{f(\bar{y})T(\bar{x} | \bar{y})}{f(\bar{x})T(\bar{y} | \bar{x})} \right\}.$$

Die „*Image Contribution Function*“  $f$  ist proportional zum relativen Einfluss eines Pfades  $\bar{x}$  und ist lediglich bei Pfaden, welche an einer Lichtquelle starten, auf die Bildebene treffen und an der Kamera enden, ungleich 0. Im Falle von  $f(\bar{y}) = 0$  gilt  $a(\bar{y} | \bar{x}) = 0$ . Somit werden Pfade ohne Einfluss immer abgelehnt.<sup>73</sup>

### 9.2.3. Initialisierung und Start-Up Bias

Die Wahl des ersten Pfades beeinflusst den gesamten *Metropolis* Algorithmus. Sollte der Anfangspfad keinen Einfluss auf das finale Bild haben, so können durch Mutation keine weiteren sinnvollen Pfade gefunden werden, da  $a(\bar{y} | \bar{x})$  in diesem Fall noch nicht definiert ist. Dies würde zu einem vollkommen schwarzen Bild führen. Dieser Fehler wird im Englischen „*Start-Up Bias*“ genannt. Um diesen Fehler zu korrigieren, wird folgender Lösungsansatz angewendet.

Der erste Pfad wird mit dem Faktor  $W = \frac{f(\bar{x})}{p_0(\bar{x})}$  gewichtet. Diese Gewichtung wird

bei allen abgetasteten Samples mit einberechnet und wird auch durch Mutationen

<sup>72</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 3

<sup>73</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 3

des Pfades nicht verändert. Dabei ergeben sich aber folgende Probleme, da die Gewichtung des ersten Pfades auf einer Zufallszahl basiert: Im Falle einer niedrigen Gewichtung des ersten Pfades ist eine hohe Anzahl von Iterationen notwendig, um ein qualitativ hochwertiges Bild zu generieren. Eine hohe Gewichtung führt hingegen zu einer erhöhten Streuung. Um den Einfluss des ersten Pfades auf den gesamten Algorithmus zu minimieren, werden mehrere Instanzen des Algorithmus, jede davon berechnet einen unterschiedlichen ersten Pfad, zeitgleich zur Berechnung herangezogen. Aus den daraus resultierenden Pfaden, werden alle Pfade mit gleicher Gewichtung als Basis für weitere Mutationen herangezogen.<sup>74</sup>

### 9.3. Mutationsverfahren

Ein sorgfältig ausgewähltes Mutationsverfahren ist entscheidend, um ein rauschfreies Bild erzeugen zu können. Ein gutes Mutationsverfahren sollte daher folgende Eigenschaften bei den Mutationen aufweisen können:

#### Hohe Annahmewahrscheinlichkeit

Eine durchschnittlich niedrige Annahmewahrscheinlichkeit  $a(\bar{y} | \bar{x})$  führt zu langen Pfadsequenzen  $\bar{x}, \bar{x}, \dots, \bar{x}$ . Dies führt zu einer gehäuften Ansammlung von Samples in einem kleinen Teilbereich des Bildes und führt zu Rauschen.<sup>75</sup>

#### Große Veränderungen

Kleine Veränderungen am Pfad führen zwar zu einer hohen Annahmewahrscheinlichkeit, die daraus gewonnenen Samples ähneln dabei allerdings stark denen des vorausgegangenen Pfades. Dies führt wiederum zu einer ungewollten Gruppierung der Samples in einem kleinen Teilbereich. Es ist daher wichtig, umfangreiche Veränderungen am Pfad durchzuführen. Dies können zum Beispiel die Erweiterung der Pfadlänge, oder der Austausch eines spiegelnden Pfades durch einen diffusen Pfad, sein.<sup>76</sup>

---

<sup>74</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 5

<sup>75</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 6

<sup>76</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 6

**Ergodizität**

Um ein unverfälschtes Bild zu generieren ist es wichtig, dass jeder mögliche Pfad mit einer Wahrscheinlichkeit von  $p > 0$  verfolgt wird.<sup>77</sup>

**Wechsel des Bildbereichs**

Um ein vollständiges Bild zu rendern, müssen an allen Bildpixel Berechnung durchgeführt werden. Um Übereinstimmungen verschiedener Abtastwerte am selben Bildpunkt zu vermeiden, sollten Mutationen die Stelle des Pfades auf der Bildebene verändern.<sup>78</sup>

**Stratifizierung**

Stratifizierung ist eine wichtige und einfache Methode um die Varianz im finalen Bilde zu reduzieren. Dabei wird der Integrationsbereich in mehrere kleinere Teilbereiche aufgeteilt. Jeder dieser Teilbereiche wird mit einer zur Größe des Teilbereichs proportionalen Anzahl von Samples abgetastet.<sup>79</sup>

**Geringer Rechenaufwand**

Es ist erstrebenswert, dass Mutationen einen geringen zusätzlichen Rechenaufwand erfordern. Im Allgemeinen bringen zusätzliche Raytracing-Berechnungen den größten Aufwand mit sich.<sup>80</sup>

**9.3.1. Bidirektionale Mutationen**

Bidirektionale Mutationen bilden die Grundlage des *Metropolis Light Transport* Algorithmus. Die Idee dahinter ist simpel. Man wählt einen Teilbereich des aktuellen Pfades  $\bar{x}$ , und ersetzt diesen durch einen neuen Teilbereich. Dieser Vorgang wird in mehreren Schritten durchgeführt.<sup>81</sup>

Als erstes wird der zu löschende Teilbereich ausgewählt. Dabei wird jeder mögliche Teilbereich des Pfades  $\bar{x} = x_0 \dots x_k$  berücksichtigt. Für jeden Teilbereich

---

<sup>77</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 6

<sup>78</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 6

<sup>79</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 6

<sup>80</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 6

<sup>81</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 6

$x_s \dots x_t$  wird dabei eine Wahrscheinlichkeit  $p_d[s, t]$  berechnet, die sich aus zwei Faktoren zusammensetzt. Der erste Faktor  $p_{d,1}$  hängt lediglich von der Pfadlänge ab und dient dazu die Entfernung von kurzen Teilbereichen zu favorisieren. Kurze Teilbereiche führen zu geringerem Rechenaufwand und Mutationen mit größerer Annahmewahrscheinlichkeit. Die Aufgabe des zweiten Faktors ist es Mutationen mit geringer Annahmewahrscheinlichkeit zu vermeiden (siehe Kapitel 9.4.2).<sup>82</sup>

Durch die Entfernung des Teilstücks besteht der originale Pfad  $\bar{x}$  nun aus zwei Teilen  $x_0 \dots x_s$  und  $x_t \dots x_k$ . Um die Mutation zu vervollständigen wird zuerst die Länge des neuen Teilbereichs festgelegt. Um eine hohe Annahmewahrscheinlichkeit zu erzielen, sollten dabei die Längen des gelöschten und des neu addierten Teilbereichs gleich groß sein. Danach werden vom Ende der zwei originalen Teilpfade nacheinander neue Punkte addiert, bis die festgelegte Länge erreicht wurde. Für jeden neuen Punkt wird dabei ein Strahl vom Ende des aktuellen Teilpfades ausgesendet. Die Richtung dieses Strahls wird durch die BRDF des aktuellen Punktes bestimmt. Trifft der Strahl auf einen Oberflächenpunkt, so wird dieser zum neuen Endpunkt des Pfades. Sollte der Strahl keine Geometrie antreffen, so wird die Mutation auf der Stelle abgelehnt. Es ist möglich, dass beim anfänglichen Entfernen des Teilbereichs auch der erste oder letzte Punkt des Pfades entfernt wurde. In diesem Fall wird der fehlende Endpunkt durch einen zufällig ausgewählten Punkt an der Lichtquelle oder der Kamera ersetzt.<sup>83</sup>

Schlussendlich müssen die zwei Hälften des Pfades miteinander verbunden werden. Dazu wird wieder ein Strahl zwischen den beiden Endpunkten ausgesendet und die Mutation ist beendet. Sollte der Strahl dabei auf eine Oberfläche treffen, wird die Mutation abgelehnt.<sup>84</sup>

---

<sup>82</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 6

<sup>83</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 6

<sup>84</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 6

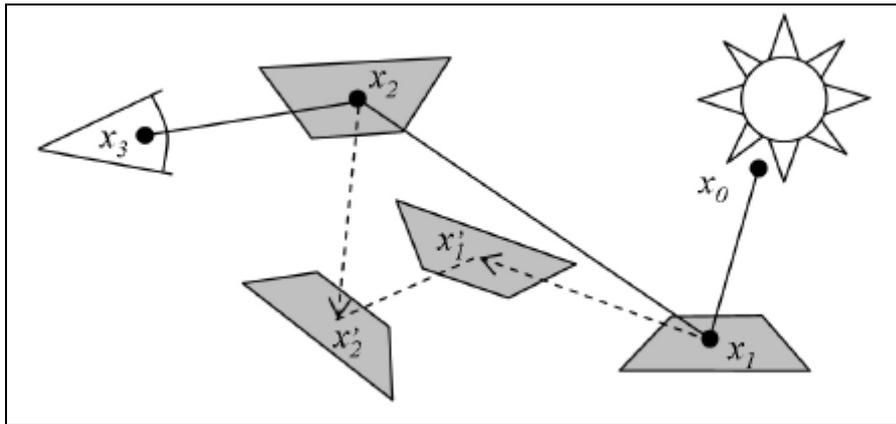


Abbildung 21: Ein Pfad der Länge 3 wird zu einem Pfad der Länge 5 mutiert. Der Teilbereich  $x_1, x_2$  wird entfernt. Die Punkte  $x'_1$  und  $x'_2$  werden durch die Aussendung von Strahlen ab den Punkten  $x_1$  und  $x_2$  gewonnen. Der neue Pfad wird durch den Strahl  $x'_1$  nach  $x'_2$  komplettiert.

### 9.3.2. Perturbations

Bidirektionale Mutationen basieren auf großen Änderungen des Pfades. Bei speziellen Lichtsituationen wie Kaustiken oder kleinen Details in der Geometrie, führen grobe Veränderungen am Pfad allerdings zu einer niedrigen Annahmewahrscheinlichkeit. Für diese speziellen Situationen wurde ein weiteres Mutationsverfahren entwickelt. Das Prinzip der *Perturbations* baut auf kleinen Veränderungen des Pfades, durch eine geringe lokale Veränderung der Pfadpunkte, auf. Dabei gibt es mehrere spezielle Arten der *Perturbations*.<sup>85</sup>

#### Lens Perturbations

Dabei wird ein Teilbereich  $x_1 \dots x_k$  des Pfades gelöscht, der in direkter Verbindung zur virtuellen Kamera steht. Danach wird der Punkt auf der Bildebene um eine zufällige Entfernung  $R$  und einen zufällig gewählten Winkel  $\Phi$  verändert. Von diesem neuen Punkt auf der Bildebene wird ein Strahl in die Szene ausgesendet. Trifft der Strahl auf eine Oberfläche wird von diesem Punkt ein weiterer Strahl ausgesendet. Die Anzahl und die Art der Reflexionen müssen dabei an jedem Punkt mit denen des originalen Teilpfades übereinstimmen. Trifft ein Strahl auf eine spiegelnde Oberfläche obwohl eine diffuse erwartet wird, wird die Mutation

<sup>85</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 7

abgelehnt. Sollte dieser Strahl auf keine Geometrie treffen, so wird die ebenfalls Mutation abgelehnt.<sup>86</sup>

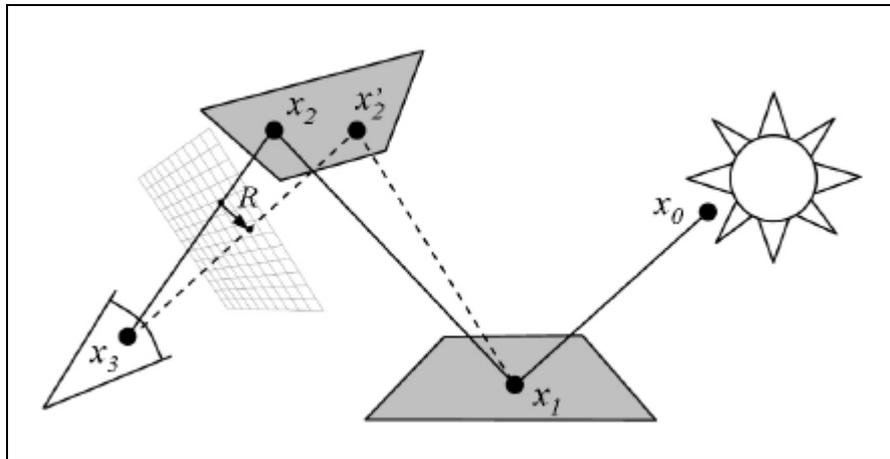


Abbildung 22: Punkt  $x_2$  ändert durch Lens Perturbation seine Position nach  $x'_2$ . Die gestrichelte Linie zeigt den mutierten Pfad.

### Caustics Perturbation

Im Gegensatz zur *Lens Perturbation* wird bei dieser Methode der neue Strahl von der Lichtquelle, oder einer diffusen Oberfläche, die Strahlung zur ersten spiegelnden Reflexion transportiert, ausgesendet. Die Richtung des Strahls zur spiegelnden Oberfläche wird abermals durch die Zufallsparameter  $\Phi$  und  $\theta$  verändert.<sup>87</sup>

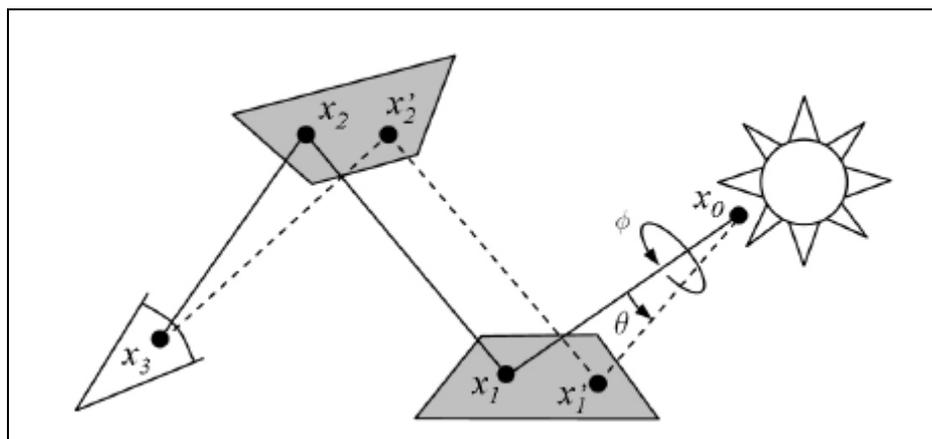


Abbildung 23: Ein Pfad wird durch eine Caustic Perturbation verändert. Die gestrichelte Linie zeigt den mutierten Pfad.  $x_1$  ist eine spiegelnde Oberfläche und wirft Kaustiken auf die diffuse Oberfläche  $x_2$ .

<sup>86</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 7

<sup>87</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 8

### 9.3.3. *Lens Subpath Mutationen*

*Lens Subpath* Mutationen dienen dazu, die Abtastpunkte gleichmäßig auf der Bildebene zu verteilen. Dabei wird wiederum ein Teilbereich des Pfades ausgewählt, der in direkter Verbindung zur virtuellen Kamera steht. Dieser Teilpfad  $\bar{x}_e$  wird im Speicher abgelegt. Bei der nächsten Mutation ersetzt  $\bar{x}_e$  den Teilpfad, des zu mutierenden Pfades. Jeder  $\bar{x}_e$  wird bei einer festgelegten Anzahl von Mutationen verwendet. Wird diese Anzahl überschritten, wird ein neuer Teilpfad an einer anderen Stelle der Bildebene generiert.<sup>88</sup>

*Lens Subpath* Mutationen bringen zwei wichtige Vorteile mit sich. Erstens wird sichergestellt, dass der gesamte Bildbereich abgearbeitet wird und somit jedes Pixel einen Abtastwert zugewiesen bekommt. Zweitens kann durch das Wiederverwerten von Teilpfaden an mehreren Bildpunkten Rechenzeit gespart werden.<sup>89</sup>

## 9.4. *Verbesserungen am Algorithmus*

Veach und Guibas stellten zusätzlich zum grundlegenden *Metropolis* Algorithmus einige Verbesserungen zur Steigerung der Effizienz vor.

### 9.4.1. *Direkte Beleuchtung*

Zur Berechnung des direkten Lichts in der Szene werden, anstatt des komplexen *Metropolis* Algorithmus, lediglich Standardtechniken (*Raytracing*) verwendet. In den meisten Fällen kann man dadurch ein besseres Ergebnis bei geringerem Rechenaufwand erreichen. Die *Metropolis*-Berechnung beschränkt sich ausschließlich auf die indirekte Beleuchtung.<sup>90</sup>

### 9.4.2. *Importance Sampling an Mutationen*

Durch eine Erhöhung der durchschnittlichen Annahmewahrscheinlichkeit der Mutationen kann die Effizienz des Algorithmus entscheidend verbessert werden. Dies wird durch eine Gewichtung sämtlicher möglicher Mutationen erreicht. Je

---

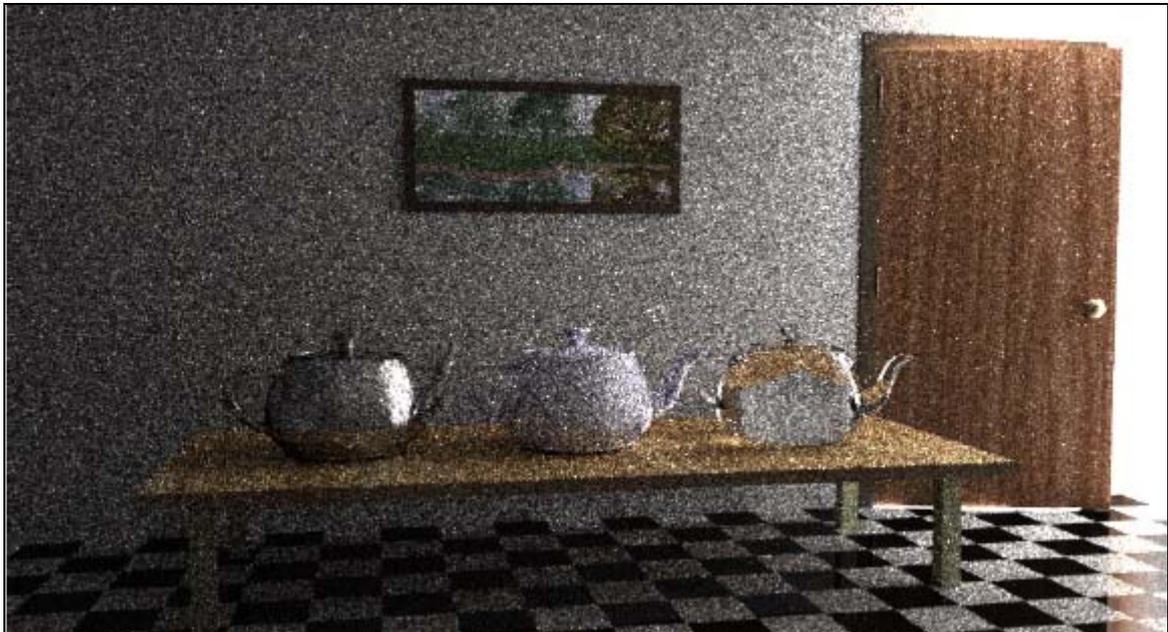
<sup>88</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 6

<sup>89</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 8

<sup>90</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 8

höher die Wahrscheinlichkeit ist, dass der neu generierte Teilpfad angenommen wird, desto höher fällt die Gewichtung der gesamten Mutation aus.<sup>91</sup>

## 9.5. Ergebnisse



**Abbildung 24: Bidirektionales Pathtracing**

Entnommen aus Veach/Guibas, Siggraph Paper Metropolis Light Transport



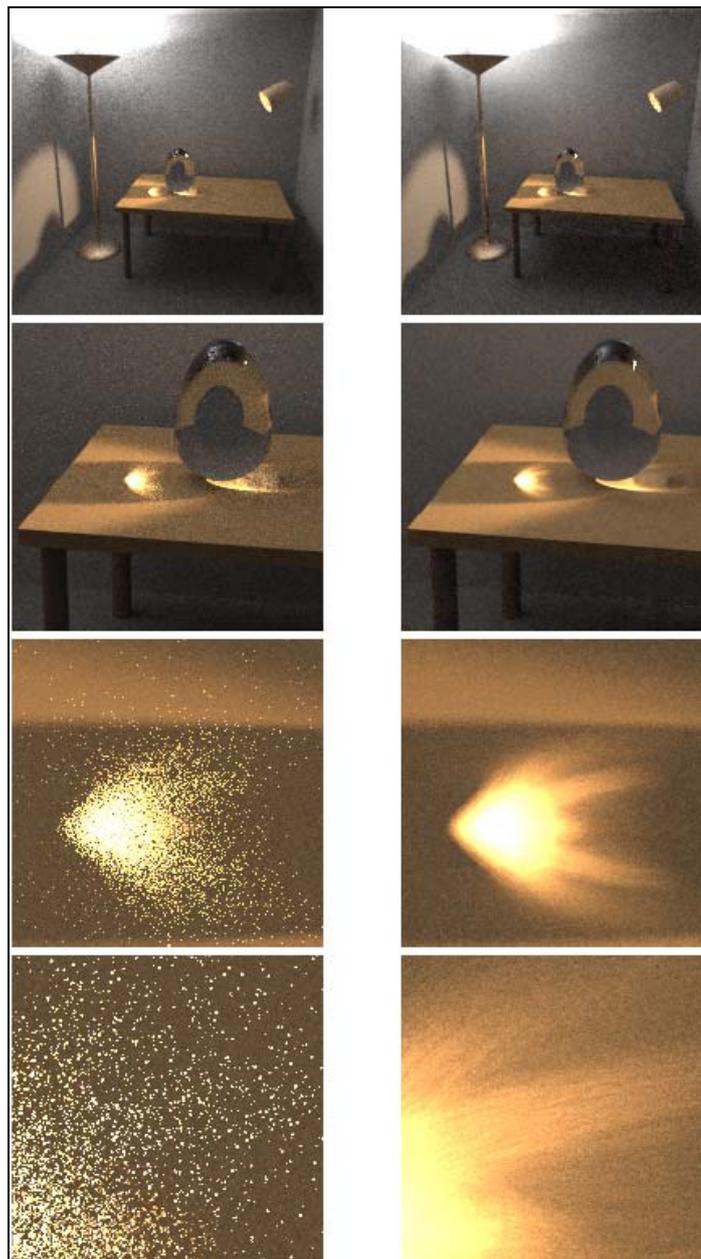
**Abbildung 25: Metropolis Light Transport**

Entnommen aus Veach/Guibas, Siggraph Paper Metropolis Light Transport

---

<sup>91</sup> Vgl. Veach/Guibas, Siggraph Paper Metropolis Light Transport, Seite 8

Abbildung 24 und 25 zeigen eine Testszene, welche überwiegend durch indirektes Licht beleuchtet wird. Bei gleicher Berechnungszeit liefert *Metropolis Light Transport* ein weitaus besseres Ergebnis als *Bidirektionales Pathtracing*. Einen besonderen Vorteil zieht der *Metropolis* Algorithmus dabei aus der Wiederverwendung der Teilpfade, die Licht durch die Tür in den Raum transportieren. Diese können die Basis vieler wichtiger Mutationen bilden.



**Abbildung 26: Bidirektionales Pathtracing (links) und Metropolis Light Transport (rechts).**

Entnommen aus Veach/Guibas, Siggraph Paper Metropolis Light Transport

Abbildung 26 stellt ebenfalls eine Szene mit indirektem Licht und Kaustiken dar. Während in der ersten Reihe die Unterschiede noch recht gering ausfallen, wird der Qualitätsunterschied bei Vergrößerung der Kaustiken sehr deutlich. Der *Metropolis* Algorithmus profitiert in dieser Szene von der Möglichkeit weitere Pfade durch *Perturbations* in wichtigen Bereichen zu generieren.

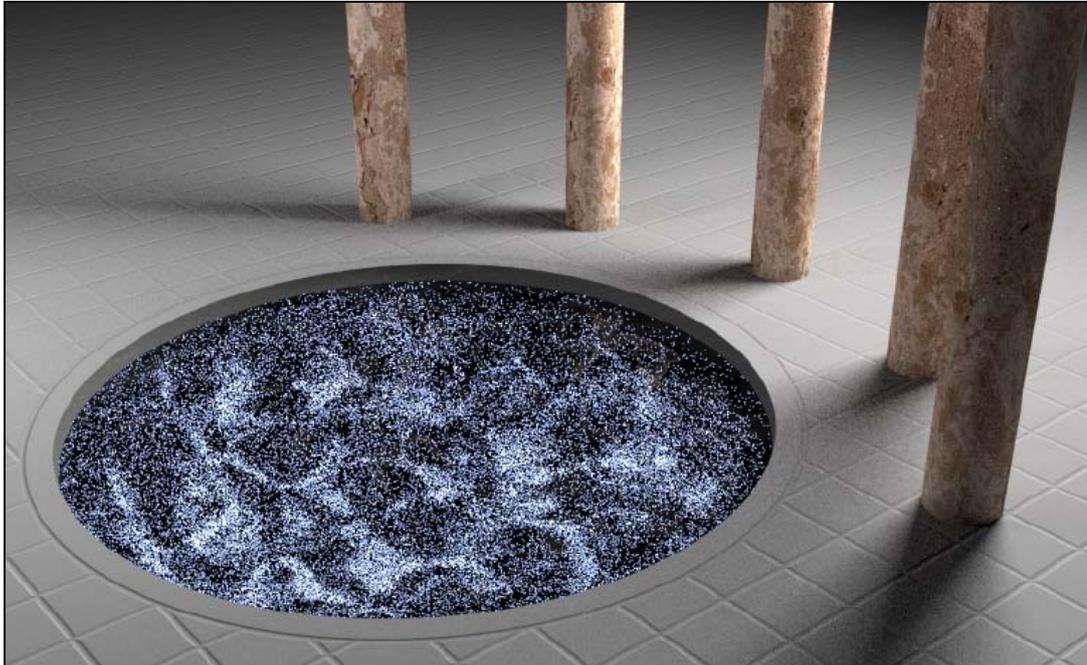


Abbildung 27: Pathtracing

Entnommen aus Veach/Guibas, Siggraph Paper Metropolis Light Transport

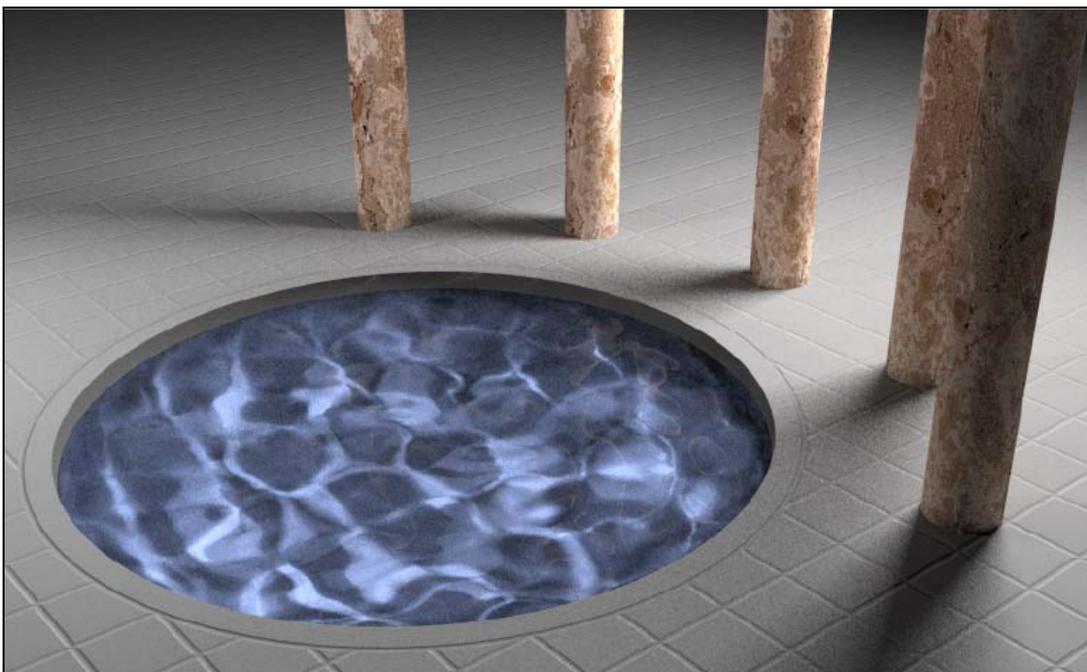


Abbildung 28: Metropolis Light Transport

Entnommen aus Veach/Guibas, Siggraph Paper Metropolis Light Transport

Abbildung 28 zeigt Kaustiken am Grund eines Swimmingpools. *Pathtracing* kann bei dieser Lichtsituation schwer wichtige Transportpfade finden, da die Pfade von der virtuellen Kamera ausgesendet werden. Die Chance vom Pool aus auf die Lichtquelle zu treffen, ist äußerst gering. Der *Metropolis* Algorithmus kann wichtige Pfade durch *Perturbations* abtasten und so ein rauschfreies Bild erzeugen.

## 10. Photon Mapping

### 10.1. Grundidee

*Photon Mapping* ist ein von Henrik Wann Jensen im Jahre 1995 veröffentlichter Algorithmus der Bildsynthese, der vornehmlich als Erweiterung von Raytracing-basierten Verfahren genutzt wird. Das Ziel von Photon Mapping ist es, die globale Beleuchtung einer Szene effizient zu ermitteln und somit realistische Bilder bei geringem Zeitaufwand zu erzeugen.<sup>92</sup>

Der Algorithmus arbeitet dabei in zwei Phasen. Im ersten Schritt wird durch die Aussendung von Photonen die Ausbreitung des Lichts simuliert. Die Photonen werden durch die Szene verfolgt und können an den Oberflächen der Szenenobjekte reflektiert oder transmittiert werden. Für jede Interaktion mit einer diffusen Oberfläche wird ein Eintrag in einer *Photonmap* gespeichert. Aus Effizienzgründen wird dabei zwischen einer globalen und einer Kaustik-Photonmap unterschieden.<sup>93</sup>

Im zweiten Teil des Verfahrens wird durch Auswertung, der aus der Versendung der Photonen gewonnenen Daten, die Beleuchtungssituation errechnet. Die Photonmap ist dabei unabhängig von der Komplexität der Szenengeometrie, wodurch dieser Algorithmus auch bei Szenen, welche hoch aufgelöste Geometrie enthalten, sehr effizient eingesetzt werden kann.<sup>94</sup>

---

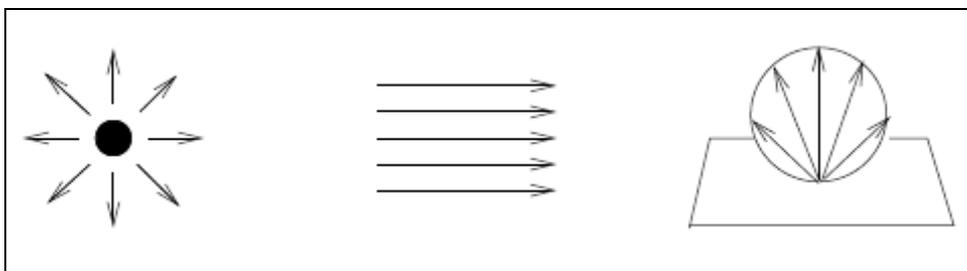
<sup>92</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 12

<sup>93</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 12

<sup>94</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 12

### 10.1.1. Photonenaussendung

Das Verfahren beginnt mit der Aussendung der Energie der Lichtquellen in Form kleiner Energiepakete, den Photonen. Bei einem diffusen Punktlicht werden die Photonen in zufällige Richtungen, gleichmäßig verteilt, ausgesendet. Photonen einer gerichteten Lichtquelle werden alle mit derselben Richtung ausgesendet, haben ihren Ursprung jedoch außerhalb der Szene. Von einer flächenförmigen Lichtquelle werden die Photonen an zufälligen Positionen auf der Fläche ausgesendet. Dadurch ist auch die Modellierung komplexer Lichtquellen, beispielsweise einer Glühbirne, möglich.<sup>95</sup>



**Abbildung 29: Ausstrahlung verschiedener Lichtquellen. Punktlicht (links), gerichtetes Licht (mitte) und Flächenlicht (rechts).**

Entnommen aus Jensen, A Practical Guide To Global Illumination Using Photon Maps

Die Energie einer Lichtquelle wird gleichmäßig auf die von ihr ausgesandten Photonen aufgeteilt. Die von einem einzelnen Photon transportierte Energie ergibt sich demnach aus:

$$L_{\text{Photon}} = \frac{L_{\text{Lichtquelle}}}{\text{Anzahl der Photonen}}$$

Existieren in der Szene verschiedene Lichtquellen, so ist die Anzahl der auszusendenden Photonen entsprechend der Energie und Größe auf diese zu verteilen. Eine Lichtquelle mit doppelter Intensität sollte demnach auch doppelt so viele Photonen aussenden. Die Auswahl der Lichtquelle erfolgt mittels eines Zufallswertes, wobei die Wahrscheinlichkeit, eine bestimmte Lichtquelle zu selektieren, proportional zu der von ihr abgestrahlten Energie ist.<sup>96</sup>

<sup>95</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 16

<sup>96</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 17

### 10.1.2. Projektionskarten (Projection Maps)

Vor allem in Szenen mit wenigen Objekten werden viele Photonen unnötig ausgesendet, da sie nie auf eine Oberfläche treffen. Ein von Jensen vorgestelltes Verfahren zur Reduzierung dieses Problems ist die Verwendung von Projektionskarten.<sup>97</sup>

Eine Projektionskarte beschreibt die Sichtbarkeit der Objekte von der Lichtquelle aus betrachtet in Form eines Rasterbildes. Dieses Rasterbild besteht aus mehreren kleinen Zellen. Eine Zelle wird „markiert“, wenn in diese Richtung Geometrie vorhanden ist. Zur Vereinfachung werden dabei nur die umhüllenden Körper (Bounding Box) der Objekte oder ganze Objektgruppen berücksichtigt. Die Aussendung der Photonen beschränkt sich dann nur noch auf die Richtungen, die durch die markierten Zellen vorgegeben sind. Dabei ist jedoch zu berücksichtigen, dass durch die Einschränkung der Abstrahlrichtungen auch nur ein Teil der Energie der Lichtquelle betrachtet wird. Jensen schlägt aus diesem Grund vor, die von den Photonen transportierte Energie wie folgt zu skalieren:

$$L_{\text{Photon}} = \frac{L_{\text{Lichtquelle}}}{\text{Anzahl der Photonen}} \frac{\text{Anzahl der besetzten Zellen}}{\text{Gesamtzahl der Zellen}}$$

In Szenen mit mehreren Lichtquellen führt dies jedoch dazu, dass die Energie der Photonen je nach Lichtquelle variiert. Um dies zu vermeiden, werden vor dem Aussenden die Projektionskarten für alle Lichtquellen erzeugt. Anschließend kann die Auswahl der Lichtquelle mit einer Wahrscheinlichkeit proportional zum abstrahlenden Energieanteil vorgenommen werden. Von einer Lichtquelle mit stark markierter Projektionskarte werden dann mehr Photonen versendet, als von einer Lichtquelle gleicher Helligkeit und leicht markierter Projektionskarte.<sup>98</sup>

Projektionskarten lassen sich für spezielle Anwendungsfälle weiter vertiefen. Für die Aussendung von Photonen zur Erzeugung der Kaustik-Photonmap, werden lediglich Objekte mit entsprechenden Materialeigenschaften für die Markierung der

---

<sup>97</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 18

<sup>98</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 18

Projektionskarte benutzt, da hierbei nur Photonen von Interesse sind, die sich über spiegelnde Reflexionen oder Transmissionen durch die Szene bewegen.<sup>99</sup>

## 10.2. Photonenverfolgung durch die Szene

Nachdem ein Photon ausgesendet wurde, wird es durch die gesamte Szene verfolgt. *Photon Mapping* arbeitet dabei genauso wie *Raytracing* (siehe Kapitel 5.2), abgesehen davon, dass Photonen, im Gegensatz zu Strahlen (Rays) nicht Strahldichte, sondern Strahlungsleistung transportieren. Diese Unterscheidung ist wichtig, da die Interaktion eines Photons mit einem Material, sehr von der eines Strahles abweichen kann. Für die Interaktion eines Photons mit einer Objektoberfläche ist das Verhalten des Materials in korrekter Weise nachzubilden. Da sich das Licht nach der Interaktion meist in mehr als einer Richtung ausbreitet, liegen die Aufspaltung der vom Photon transportierten Energie und die Verfolgung der resultierenden Anteile nahe. Dies würde jedoch zu einem exponentiellem Wachstum der zu verfolgenden und zu speichernden Photonen führen. Stattdessen wird, entsprechend der Materialparameter, nur über das Ende oder die Art der Interaktion entschieden.<sup>100</sup>

### 10.2.1. Reflexion, Transmission oder Absorption

Zur besseren Erläuterung wird von einer monochromatischen Situation ausgegangen. Für eine Oberfläche werden die diffuse und die spiegelnde Reflexion durch die Parameter  $d$  und  $s$  beschrieben ( $d + s \leq 1$ ). Nach der Auswahl einer gleich verteilten Zufallsvariablen  $\xi$  ergibt sich:

- $\xi \in [0, d]$  .....diffuse Reflexion
- $\xi \in [d, s + d]$  .....spiegelnde Reflexion
- $\xi \in [s + d, 1]$  .....Absorption

Wird die vom Photon transportierte Energie in mehr als einem Spektralbereich repräsentiert (RGB), so werden die Wahrscheinlichkeiten der Reflexionen aus dem Maximum der einzelnen Komponenten gebildet:

- Diffuse Reflexion:  $P_d = \frac{\max(d_r P_r, d_g P_g, d_b P_b)}{\max(P_r, P_g, P_b)}$

<sup>99</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 18

<sup>100</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 19

- Spiegelnde Reflexion:  $P_s = \frac{\max(s_r P_r, s_g P_g, s_b P_b)}{\max(P_r, P_g, P_b)}$

wobei  $P_r, P_g, P_b$  die Stärke der einfallenden Photonen in den verschiedenen Farbbereichen angeben. Über das Ende bzw. die Art der Fortführung des Photonenpfades wird wieder mit einer gleich verteilten Zufallsvariable entschieden:

- $\xi \in [0, P_d]$  .....diffuse Reflexion
- $\xi \in [P_d, P_s + P_d]$  .....spiegelnde Reflexion
- $\xi \in [P_s + P_d, 1]$  .....Absorption

Da die Wahrscheinlichkeiten aus den Farbkomponenten gebildet wurden, ist bei einer Weiterverfolgung der reflektierte Strahlungsfluss entsprechend der Anteile anzupassen. Wurde beispielsweise die spiegelnde Reflexion gewählt, so ergibt sich der Fluss des Photons nach der Interaktion aus:

$$P_{refl,r} = \frac{P_{inc,r} S_r}{P_s}, P_{refl,g} = \frac{P_{inc,g} S_g}{P_s}, P_{refl,b} = \frac{P_{inc,b} S_b}{P_s}.^{101}$$

### 10.3. Speicherung der Photonen

Beim *Photon Mapping* werden lediglich Photonen gespeichert, die auf eine diffuse Oberfläche treffen. Spiegelnde Reflexionen können mittels Standard *Raytracing* effizienter gerechnet werden. Alle weiteren Interaktionen der Photonen mit nicht spiegelnden Oberflächen werden in der so genannten *Photon Map* gespeichert. Dabei kann jedes ausgesendete Photon mehrmals entlang seines Pfades gespeichert werden. Für jede Interaktion werden die genaue Position, die anfallende Photon-Stärke und der Einfallswinkel gespeichert. Die Wahl der Datenstruktur zur Speicherung der Photonen hat entscheidenden Einfluss auf die Geschwindigkeit des Verfahrens. Die Anforderungen sind eine kompakte Speicherung, sowie die Möglichkeit, sehr schnell die in der Umgebung eines Punktes liegenden Photonen ermitteln zu können.<sup>102</sup>

<sup>101</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 20/21

<sup>102</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 22

### 10.3.1. Arten der Photon Maps

Zur weiteren Steigerung der Effizienz werden mehrere spezifische *Photon Maps* angelegt.

#### Kaustik Photon Map

Beinhaltet alle Photonen, die mindestens eine Interaktion mit einer spiegelnden Oberfläche aufweisen, bevor sie auf eine diffuse Oberfläche treffen.

#### Globale Photon Map

Stellt eine annähernde Repräsentation der globalen Beleuchtung aller diffusen Oberflächen in der Szene dar.

#### Volumetrische Photon Map

Dabei wird die indirekte Beleuchtung eines beteiligten Mediums (z.B. Nebel) gespeichert.

Die Aufteilung auf mehrere spezifische *Photon Maps* ermöglicht eine unterschiedliche Anzahl an ausgesendeten Photonen. Um dies zu erreichen werden die verschiedenen *Photon Maps* in mehreren Durchgängen erstellt. Dabei benötigt die *Kaustik Map* zum Beispiel eine höhere Anzahl an ausgesendeten Photonen, als die *globale Photon Map*.<sup>103</sup>

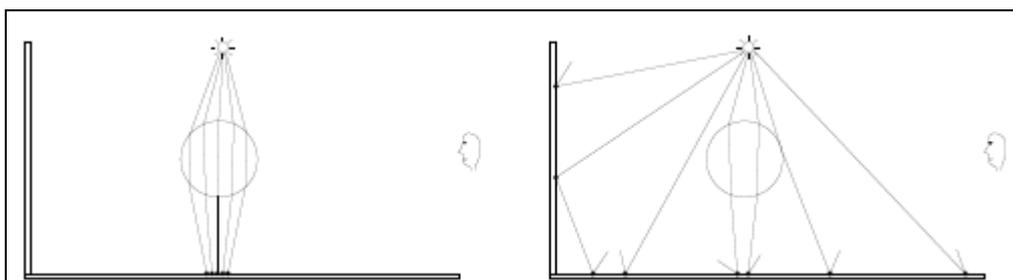


Abbildung 30: Die Erzeugung einer Kaustik Map (links) und einer globalen Photon Map (rechts).

Entnommen aus Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 27

Um beim anschließenden Renderingvorgang möglichst effizient zu jedem Punkt die benachbarten Photonen in der *Photon Map* lokalisieren zu können, wurde eine spezielle Datenstruktur entwickelt.

<sup>103</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 26

### 10.3.2. Der kd-Baum

Für die Speicherung der Photon Maps wird eine, von Jon Louis Bentley vorgestellte, spezielle Form eines kd-Baums verwendet, welche die Nachbarschafts- und Bereichsuche für mehrdimensionale Daten sehr effizient unterstützt. Ein kd-Baum ist eine Raum unterteilende Datenstruktur, wobei kd hierbei für k-dimensional steht. Als erstes wird der Ausgangsraum definiert. Dies ist beim *Photon Mapping* der kleinste, achsenparallele Quader (Bounding Box), welcher alle Photonen enthält. Danach wird der Raum durch jeden Knoten des kd-Baumes an einer Koordinatenachse geteilt. Pro Knoten wird dabei die Dimension und Position der Teilung gespeichert. Für eine effiziente Suche im Baum ist es entscheidend, dass jeder Knoten möglichst gleich ausgeprägte Unterbäume enthält. Im optimalen Fall spricht man von einem balancierten Baum.<sup>104</sup>

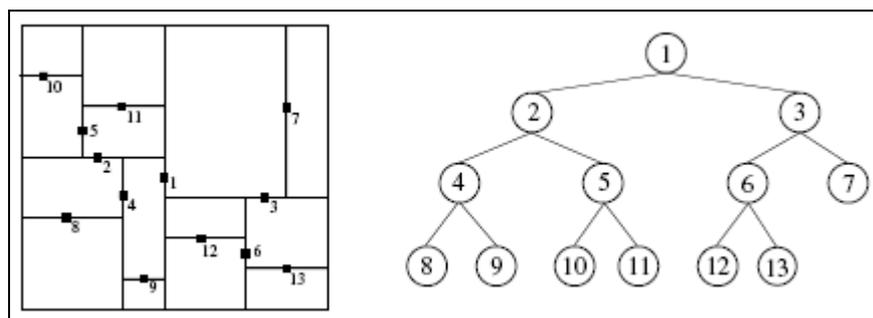


Abbildung 31: kd-Baum

Entnommen aus Jensen, A Practical Guide To Global Illumination Using Photon Maps

Der Algorithmus geht beim *Photon Mapping* dabei folgendermaßen vor: Für alle noch nicht definierten Photonen wird zuerst die Bounding Box berechnet. Danach wird der Median ermittelt. Das gefundene Photon stellt, zusammen mit der Dimensionsangabe der Teilung, einen neuen Knoten im Raum dar. Photonen mit einer Position vor dem Median bilden die Elemente des linken, die mit einer Position hinter dem Median die Elemente des rechten Unterbaumes. Die direkten Kinder des Knoten werden durch einen rekursiven Aufruf der Funktion mit diesen beiden Teilmengen ermittelt.<sup>105</sup>

Diesen Vorgang könnte man als Pseudocode wie folgt beschreiben:

<sup>104</sup> Vgl. Bentley, Multidimensional Binary Search Trees Used for Associative Searching, Seite 2

<sup>105</sup> Vgl. Bentley, Multidimensional Binary Search Trees Used for Associative Searching, Seite 2

```

kdtree * balance (point) {
    Find the cube surrounding the points
    Select dimension dim in which the cube is largest
    Find median of the points in dim
    s1 = all points below median
    s2 = all points above median
    node = median
    node.left = balance (s1)
    node.right = balance (s2)
    return node
}

```

#### 10.4. Berechnung der Strahldichte

Eine besondere Herausforderung bei der Lösung der Renderinggleichung stellt die von einem Punkt reflektierte Strahldichte dar. Diese wird im zweiten Teil der Renderinggleichung beschrieben:

$$L_r(\vec{x}, \vec{\omega}_r) = \int_{\Omega} f_r(\vec{x}, \vec{\omega}_i, \vec{\omega}_r) L_i(\vec{x}, \vec{\omega}_i) |(\vec{N}_{\vec{x}} * \vec{\omega}_i)| d\vec{\omega}_i$$

wobei  $L_r$  die reflektierte Strahldichte am Punkt  $x$  in Richtung  $\vec{\omega}_r$  ist. Durch  $\Omega$  wird die Einfallsrichtung angegeben,  $f_r$  beschreibt die BRDF am Punkt  $x$  und  $L_i$  stellt die einfallende Strahldichte dar.<sup>106</sup>

An diffusen Flächen repräsentieren die Einträge der *Photon Map* einen Teil des auftreffenden Energieflusses. Die von einem Oberflächenpunkt empfangene Radiance ergibt sich aus:<sup>107</sup>

$$L_i(\vec{x}, \vec{\omega}_r) = \frac{d^2\Phi_i}{dA^\Phi d\vec{\omega}_i} = \frac{d^2\Phi_i}{|(\vec{N}_{\vec{x}} * \vec{\omega}_i)| dA d\vec{\omega}_i}$$

Dadurch lässt sich die reflektierte Strahldichte in Abhängigkeit des Flusses beschreiben:

$$L_r(\vec{x}, \vec{\omega}_r) = \int_{\Omega} f_r(\vec{x}, \vec{\omega}_i, \vec{\omega}_r) \frac{d^2\Phi_i}{dA}$$

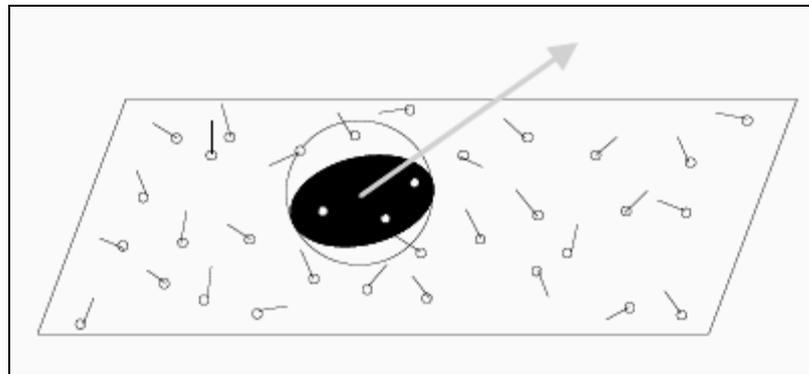
<sup>106</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 29

<sup>107</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 29

Da die gespeicherten Photonen den auftreffenden Fluss an unterschiedlichen Punkten beschreiben und auch nur einen Teil der möglichen Einfallsrichtungen  $\Omega$  abdecken, wird zur Auswertung der *Photon Map* folgende Näherung verwendet:<sup>108</sup>

$$L_r(\vec{x}, \vec{\omega}_r) \approx \sum_{p=1}^n f_r(\vec{x}, \vec{\omega}_i, \vec{\omega}_r) \frac{\Delta\Phi_p}{\Delta A}$$

Für die Berechnung der Strahldichte werden die in unmittelbarer Nähe des Punktes liegenden Photonen benutzt. Dabei nimmt man an, dass sich der Fluss innerhalb der Umgebung eines Punktes nicht wesentlich ändert. Um jeden Punkt  $x$  wird eine expandierende Kugel gebildet, welche so lange vergrößert wird, bis sie eine vorgegebene Anzahl  $n$  an Photonen beinhaltet. Diese Photonen werden anschließend zur Berechnung der Strahldichte herangezogen.<sup>109</sup>



**Abbildung 32: An jedem Punkt werden die umliegenden Photonen ermittelt.**

Entnommen aus Jensen, A Practical Guide To Global Illumination Using Photon Maps

Man geht davon aus, dass sich die Photonen auf einer ebenen Fläche um den Punkt  $x$  befinden. Durch die Projektion der Kugel auf die Oberfläche ergibt sich  $\Delta A = \pi \cdot r^2$ , wobei  $r$  den Abstand des am weitesten von  $x$  entfernten Photons der Umgebung bezeichnet. Daraus folgt:<sup>110</sup>

$$L_r(\vec{x}, \vec{\omega}_r) \approx \frac{1}{\pi \cdot r^2} \sum_{p=1}^n f_r(\vec{x}, \vec{\omega}_i, \vec{\omega}_r) \Delta\Phi_p$$

Durch die Verwendung einer Kugel besteht die Möglichkeit, dass auch Photonen, welche weit vom Punkt  $x$  entfernt sind, in die Berechnung mit eingehen. Dies

<sup>108</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 29

<sup>109</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 30

<sup>110</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 30

kann vor allem an Raumecken und harten Kanten von Objekten der Fall sein und führt zu sichtbaren Artefakten im finalen Bild. Eine höhere Genauigkeit bei der Auswahl der Photonen kann durch eine Stauchung der Kugel in eine Richtung erreicht werden. Durch eine scheibenartige Form werden weniger „falsche“ Photonen für die Berechnung ausgewählt.<sup>111</sup>

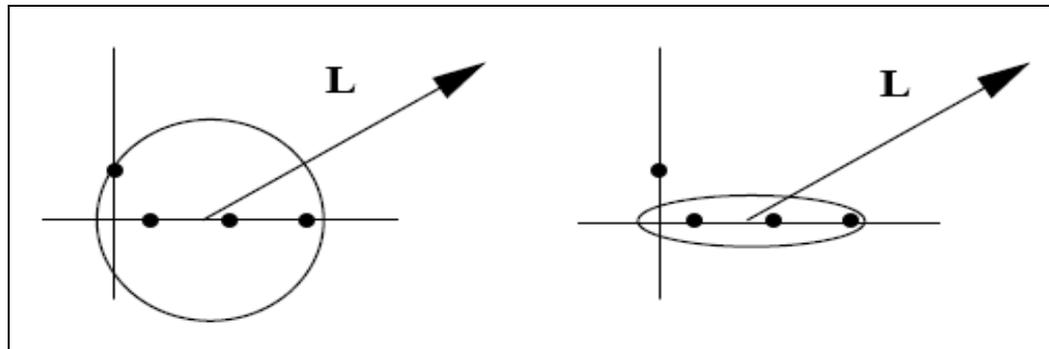


Abbildung 33: Durch eine Stauchung der Kugel werden weniger "falsche" Photonen ausgewählt.

## 10.5. Rendering

Mittels der *Photon Map* und der daraus gewonnenen Informationen über die Strahldichte, wird im anschließenden Durchgang das finale Bild gerendert. Dabei unterscheidet man bei der auf einen Punkt auftreffenden Strahldichte zwischen drei Anteilen:

$$L_i(\vec{x}, \vec{\omega}_i) = L_{i,l}(\vec{x}, \vec{\omega}_i) + L_{i,c}(\vec{x}, \vec{\omega}_i) + L_{i,d}(\vec{x}, \vec{\omega}_i)$$

- $L_{i,l}(\vec{x}, \vec{\omega}_i)$  beschreibt die direkte Beleuchtung des Punktes durch die Lichtquellen in der Szene
- $L_{i,c}(\vec{x}, \vec{\omega}_i)$  stellt die indirekte Beleuchtung durch spiegelnde Reflexionen oder Transmissionen dar.
- $L_{i,d}(\vec{x}, \vec{\omega}_i)$  beschreibt die indirekte Beleuchtung durch diffuse Reflexion<sup>112</sup>

Um zwischen einer diffusen und einer spiegelnden Reflexion unterscheiden zu können, wird das lokale Beleuchtungsmodell (BRDF) ebenfalls als Summe eines diffusen und eines spiegelnden Anteils beschrieben<sup>113</sup>:

<sup>111</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 30

<sup>112</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 38

<sup>113</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 38

$$f_r(\vec{x}, \vec{\omega}_i, \vec{\omega}_r) = f_{r,d}(\vec{x}, \vec{\omega}_i, \vec{\omega}_r) + f_{r,s}(\vec{x}, \vec{\omega}_i, \vec{\omega}_r)$$

Durch diese Zerlegungen kann letztendlich die von einem Punkt reflektierte Radiance in folgenden Teilen berechnet werden<sup>114</sup>:

$$\begin{aligned} L_r(\vec{x}, \vec{\omega}_r) &= \int_{\Omega} f_r(\vec{x}, \vec{\omega}_i, \vec{\omega}_r) L_{i,l}(\vec{x}, \vec{\omega}_i) |(\vec{N}_{\vec{x}} \vec{\omega}_i)| d\vec{\omega}_i \\ &\dots\dots\dots + \int_{\Omega} f_{r,s}(\vec{x}, \vec{\omega}_i, \vec{\omega}_r) (L_{i,c}(\vec{x}, \vec{\omega}_i) + L_{i,d}(\vec{x}, \vec{\omega}_i)) |(\vec{N}_{\vec{x}} \vec{\omega}_i)| d\vec{\omega}_i \\ &\dots\dots\dots + \int_{\Omega} f_{r,d}(\vec{x}, \vec{\omega}_i, \vec{\omega}_r) L_{i,c}(\vec{x}, \vec{\omega}_i) |(\vec{N}_{\vec{x}} \vec{\omega}_i)| d\vec{\omega}_i \\ &\dots\dots\dots + \int_{\Omega} f_{r,d}(\vec{x}, \vec{\omega}_i, \vec{\omega}_r) L_{i,d}(\vec{x}, \vec{\omega}_i) |(\vec{N}_{\vec{x}} \vec{\omega}_i)| d\vec{\omega}_i \end{aligned}$$

Zur Lösung dieser Gleichung werden, ähnlich wie beim *Raytracing*, Strahlen von der Kamera aus durch die Szene verfolgt. Trifft ein Strahl auf eine Oberfläche, so werden erneut Strahlen in die Umgebung ausgesendet, oder die in den *Photon Maps* gespeicherten Informationen ausgewertet. Bei jeder Interaktion eines Strahls mit einer Oberfläche werden die einzelnen Teile der Gleichung berechnet. Dabei wird zwischen einer exakten und einer approximativen Variante unterschieden. Eine exakte Lösung wird ermittelt, wenn der Punkt direkt oder über nur wenige spiegelnde Reflexionen von der virtuellen Kamera gesehen wird. Eine exakte Berechnung wird ebenfalls verwendet, sobald der Abstand von Strahlursprung und auszuwertendem Punkt eine Mindestdistanz unterschreitet. Ab einer definierten Rekursionstiefe oder diffuser Reflexion wird der approximative Ansatz verwendet.<sup>115</sup>

In den nächsten Kapiteln wird näher auf die einzelnen Teile der zu lösenden Gleichung eingegangen.

### 10.5.1. Direkte Beleuchtung

Die direkte Beleuchtung wird durch den Term

$$\int_{\Omega} f_r(\vec{x}, \vec{\omega}_i, \vec{\omega}_r) L_{i,l}(\vec{x}, \vec{\omega}_i) |(\vec{N}_{\vec{x}} \vec{\omega}_i)| d\vec{\omega}_i$$

beschrieben und stellt die reflektierte Strahldichte eines Punktes durch direkte Beleuchtung einer Lichtquelle dar. Dieser Teil der Gleichung ist oftmals der

<sup>114</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 39

<sup>115</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 39

wichtigste und bedarf einer sehr genauen Berechnung. Dazu werden in Richtung der Lichtquellen Strahlen (Shadow Rays) ausgesandt, um eine mögliche Verschattung durch ein zwischen Punkt und Lichtquelle liegendes Objekt zu testen. Bei flächenförmigen Lichtquellen werden mehrere dieser Schattenstrahlen ausgesendet um einen weichen Schattenübergang zu ermöglichen. Dieses Vorgehen kann bei großen Lichtquellen jedoch sehr zeitintensiv werden, da eine hohe Anzahl von Shadow Rays notwendig ist. Jensen stellte dafür ein Konzept vor, welches eine weitere *Photon Map* verwendet, in der Photonen als Träger von Verschattungsinformationen gespeichert werden. Dies kann einerseits zu einer hohen Reduktion der Berechnungszeit, andererseits aber auch zu Artefakten bei der Berechnung von Schatten an kleiner Szenegeometrie, führen. Die approximative Lösung der direkten Beleuchtung ergibt sich aus der *globalen Photon Map*.<sup>116</sup>

### 10.5.2. Spiegelnde Reflexion

Der zweite Term

$$\int_{\Omega} f_{r,s}(\vec{x}, \vec{\omega}_i, \vec{\omega}_r) (L_{i,c}(\vec{x}, \vec{\omega}_i) + L_{i,d}(\vec{x}, \vec{\omega}_i)) |(\vec{N}_{\vec{x}} \vec{\omega}_i)| d\vec{\omega}_i$$

beschreibt die durch spiegelnde oder glänzende Reflexion ausgesendete Strahldichte. Die Berechnung erfolgt dabei durch reines *Monte Carlo Raytracing* und ein auf  $f_{r,s}$  basierendes *Importance Sampling*. Die Verwendung von Informationen aus einer *Photon Map* ist dabei nicht sinnvoll, da der maximale Beitrag aus der perfekten Reflexionsrichtung zu erwarten ist und die Wahrscheinlichkeit, dass ein Photon mit dieser Einfallsrichtung gefunden werden kann, sehr gering ist.

Bei der approximativen Lösung werden weniger Strahlen verwendet.<sup>117</sup>

### 10.5.3. Kaustiken

Kaustiken werden durch den Term

$$\int_{\Omega} f_{r,d}(\vec{x}, \vec{\omega}_i, \vec{\omega}_r) L_{i,c}(\vec{x}, \vec{\omega}_i) |(\vec{N}_{\vec{x}} \vec{\omega}_i)| d\vec{\omega}_i$$

<sup>116</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 40

<sup>117</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 41

definiert. Bei einer exakten Lösung wird das Integral durch die Auswertung der eigens dafür angelegten *Kaustik Photon Map* berechnet. Durch die hohe Photonendichte an den wichtigen Stellen, kann dabei ein sehr gutes Ergebnis erreicht werden.

Zur Approximation wird die *globale Photon Map* zu Berechnung herangezogen.<sup>118</sup>

#### 10.5.4. Diffuse Reflexion

Der letzte Term

$$\int_{\Omega} f_{r,d}(\vec{x}, \vec{\omega}_i, \vec{\omega}_r) L_{i,d}(\vec{x}, \vec{\omega}_i) |(\vec{N}_{\vec{x}} \vec{\omega}_i)| d\vec{\omega}_i$$

beschreibt das durch diffuse Reflexion einfallende Licht. Für eine exakte Berechnung werden auch hier Strahlen in die Szene ausgesendet und die diffus einfallende Strahldichte mittels *Monte Carlo Raytracing* berechnet. Eine wichtige Optimierung stellt dabei die Verwendung eines *Irradiance-Cache* dar.<sup>119</sup> Dabei wird die indirekte Beleuchtung lediglich stellenweise exakt berechnet. Für alle dazwischen liegenden Positionen wird während des Renderings die diffuse Beleuchtung durch Interpolation ermittelt.<sup>120</sup>

Für die approximative Lösung wird abermals eine Näherung mittels der *globalen Photon Map* verwendet.

#### 10.6. Filter

Eine geringe Anzahl von Photonen in der *Photon Map*, führt zu einer Interpolation der Werte und somit zu verwischten Kanten. Dies kann zwar bei diffuser Beleuchtung von Vorteil sein, da es zu einer Glättung im hochfrequenten Bereich kommt. Bei Kaustiken jedoch führt das Verwischen zu einer Minderung der visuellen Qualität. Kaustiken werden durch scharfe Kanten definiert und können durch eine Interpolation nicht mehr korrekt dargestellt werden. Um eine verbesserte Qualität bei einer möglichst geringen Anzahl von Photonen zu erreichen, kommen Filter zu Einsatz. Die Grundidee dieser Filter ist eine höhere Gewichtung der Photonen mit geringem Abstand zu  $x$ . Die Gewichtung hat ihr

<sup>118</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 41/42

<sup>119</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 42

<sup>120</sup> Vgl. Ward/Rubinstein/Clear, A Raytracing Solution for Diffuse Interreflection

Maximum am Punkt  $x$  und verkleinert sich mit der Entfernung zu diesem Punkt. Dabei gibt es zwei Arten von Filtern: Kegelfilter und Gauss-Filter.<sup>121</sup>

### 10.7. Ergebnisse

*Photon Mapping* fasst einige frühere Konzepte wie Strahlverfolgung und Monte Carlo Verfahren zu einem leistungsstarken Algorithmus zusammen. Daraus resultiert ein Verfahren, welches zeitintensive Berechnungen, wie indirekte Beleuchtung und Kaustiken, wesentlich effizienter als reine *Pathtracing* Verfahren durchführen kann. Dennoch basiert *Photon Mapping* lediglich auf Näherungen und Interpolationen und führt keine exakten Berechnungen durch. Man spricht dabei von einem so genannten „*biased*“ (engl. für verzerrt) Algorithmus.

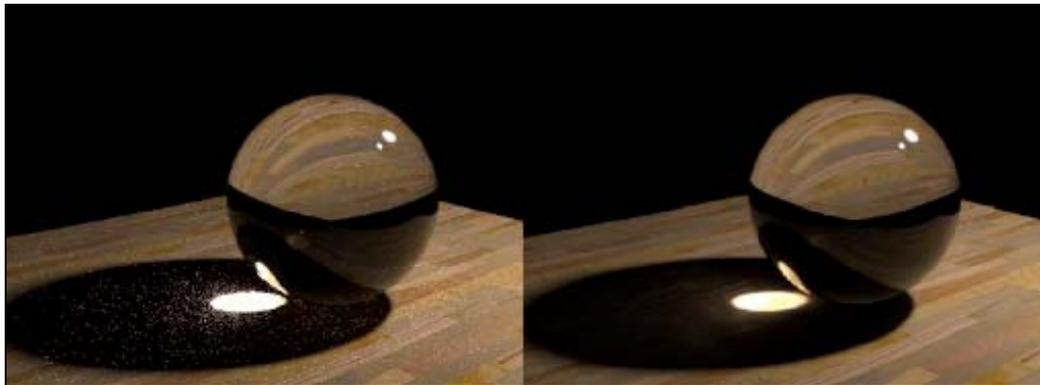


Abbildung 34: Kaustiken gerendert mit Pathtracing (links) und mit Photon Mapping (rechts).  
Entnommen aus [http://www.cs.sfu.ca/~torsten/Teaching/Cmpt461/LectureNotes/PDF/08\\_LTE\\_biased.pdf](http://www.cs.sfu.ca/~torsten/Teaching/Cmpt461/LectureNotes/PDF/08_LTE_biased.pdf)

Abbildung 34 zeigt einen direkten Vergleich zwischen *Pathtracing* und *Photon Mapping* beim Rendering von Kaustiken. Photon Mapping kann durch die angelegte *Kaustik Photon Map* die Lichtbündelungen wesentlich effizienter und rauschfreier berechnen als dies mit *Pathtracing* möglich ist.

Abbildung 35 und 36 zeigen weitere Vergleiche dieser zwei Verfahren. Während die, durch *Pathtracing* gerenderten, Bilder leichtes Rauschen enthalten, erzeugt *Photon Mapping* durch Interpolation fleckenartige Artefakte.

<sup>121</sup> Vgl. Jensen, A Practical Guide To Global Illumination Using Photon Maps, Seite 33

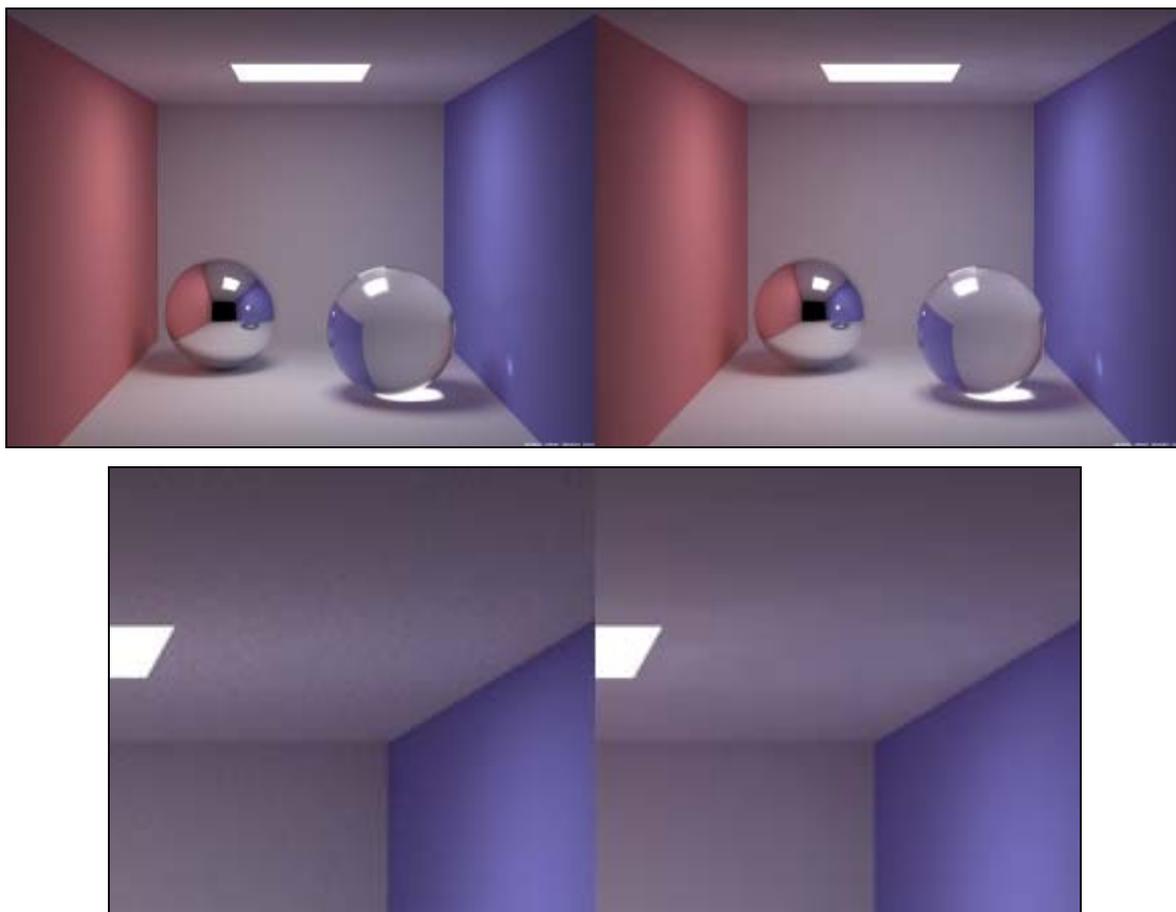


Abbildung 35: Cornell Box gerendert mit Pathtracing (links) und Photon Mapping (rechts).  
Entnommen aus <http://graphics.ucsd.edu/~henrik/images/cbox.html>



Abbildung 36: Sponza Atrium Szene gerendert mit Photon Mapping (links) und Pathtracing (rechts).  
Entnommen aus Pharr/Humphreys, Physically Based Rendering (2004), Seite 753

## 11. Praktische Anwendung in Maxwell Render™ und Mental Ray®

Im letzten Teil dieser Diplomarbeit wird anhand eines praktischen Beispiels die Rendersoftware *Maxwell Render* von Next Limit Technologies mit der, von Mental Images entwickelten, Rendersoftware *Mental Ray* verglichen. Als Testszene wird eine Cornell Box<sup>122</sup> nachgestellt. Die Ergebnisse werden anschließend mit einer Referenzfotografie einer echten Cornell Box verglichen.

### 11.1. Forschungsleitende Fragestellung

- Welche Beleuchtungs- und Rendertechniken bringen ein fotorealistisches Ergebnis und in wieweit unterscheidet sich die Berechnungszeit?
- Kann man die Einstellungen so weit optimieren, dass fotorealistisches Rendering auch für Animationen von Bedeutung ist?

### 11.2. Hypothesen

- Je fotorealistischer das digitale Bild sein muss, desto eher muss die Berechnung auf physikalisch korrekten Verfahren basieren.

### 11.3. Operationalisierung von Variablen

#### 11.3.1. Abhängige Variablen

##### Zeit

Es wird die Zeit für die Berechnung eines einzelnen Frames herangezogen. Da bei *Maxwell Render* ein „unbiased“ Algorithmus zum Einsatz kommt und die Berechnung des Bildes beliebig lange andauert, wird der Rendervorgang zu dem Zeitpunkt abgebrochen, an dem kein sichtbares Rauschen mehr im digitalen Bild erkennbar ist.

##### Realismus

Es wird versucht die Ergebnisse anhand des Referenzfotos in Bezug auf Realismus zu vergleichen. Dabei muss jedoch beachtet werden, dass die

---

<sup>122</sup> <http://www.graphics.cornell.edu/online/box/>

Messung des Realismus lediglich auf subjektiven persönlichen Meinungen basiert und nicht mit Werten gemessen werden kann.

### **11.3.2. Unabhängige Variablen**

#### **Verwendete Software**

Für das Experiment werden die Produkte *Maxwell Render* und *Mental Ray* herangezogen. *Maxwell Render* wird in der Version 1.1 als Plugin für 3D Studio Max verwendet und basiert auf einem *Metropolis Light Transport Algorithmus*.<sup>123</sup> *Mental Ray 3.4* kommt in der 3D Software Softimage|XSI zum Einsatz. Die Berechnung von globaler Beleuchtung wird dabei mittels *Photon Mapping* durchgeführt.<sup>124</sup>

#### **Aufbau der Szene**

Die Geometrie der 3D Szene wird exakt nach den Vorgaben der realen Cornell Box nachgebaut. Diese dient als Grundlage für die Berechnung. Die verwendeten Shader sind dabei softwarespezifisch. Es wird versucht möglichst ähnliche Shadereinstellungen zu verwenden.

### **11.3.3. Intervenierende Variablen**

#### **Computersystem**

Da die Berechnungszeit direkt von der Leistungsfähigkeit der verwendeten Computerhardware abhängt werden hier die wichtigsten Eckdaten angegeben.

- CPU: AMD Athlon 64 3700+ 2,2 GHz
- Arbeitsspeicher: 1024 MB RAM
- Grafikkarte: Geforce 7800 GTX 256 MB RAM
- Betriebssystem: Windows XP, Service Pack 2

---

<sup>123</sup> <http://www.maxwellrender.com/>

<sup>124</sup> [http://www.mentalimages.com/2\\_1\\_0\\_mentalray/index.html](http://www.mentalimages.com/2_1_0_mentalray/index.html)

## 11.4. Aufbau des Experiments

### 11.4.1. Referenzfoto

Als Referenzfoto wurde ein Foto einer echten Cornell Box herangezogen. Dazu wurde von A-Team Laboratories ein Modell einer Cornell Box gebaut und fotografiert. Dabei wurden sämtliche geometrischen Daten der Box, alle Einstellungen der verwendeten Kamera und die Eigenschaften der Glühbirne notiert.<sup>125</sup>



Abbildung 37: Modell der echten Cornell Box

Entnommen aus <http://www.maxwellrender.com/forum/viewtopic.php?t=14254>

### Cornell Box Daten

- Cornell Box Abmessungen: 50x50x50cm
- Kamera: ISO 200, Shutter Speed 90, fStop 8
- Lampe: 27W, 1700 Lumen, Farbe 6500k

---

<sup>125</sup> <http://www.maxwellrender.com/forum/viewtopic.php?t=14254>



Abbildung 38: Referenzfoto der echten Cornell Box

Entnommen aus <http://www.maxwellrender.com/forum/viewtopic.php?t=14254>

#### 11.4.2. Aufbau der Szene in Maxwell Render (3D Studio Max)

Die Szenengeometrie wurde mittels Polygonobjekten in 3D Studio Max maßstabsgetreu nachgebildet. Als virtuelle Kamera wurde eine „Maxwell Camera“ gewählt und so positioniert, dass der Szenenausschnitt exakt der Fotografie entspricht. Die Kameraeinstellungen wurden dabei an die Einstellungen der Originalkamera angepasst (ISO 200, fStop 8, Shutter Speed 90).

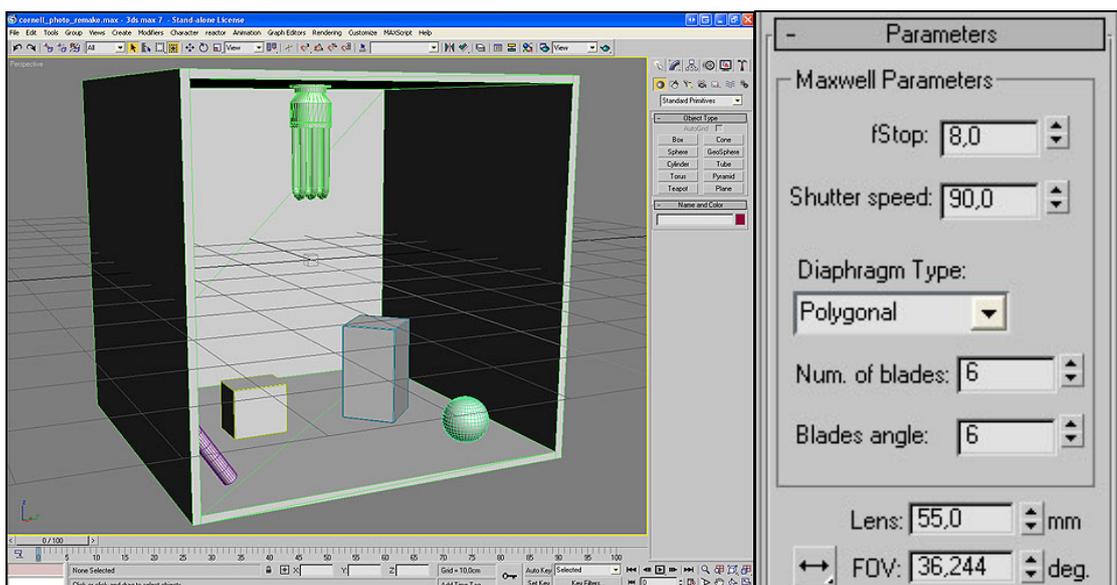


Abbildung 39: Aufbau der Szene und Kameraeinstellungen in 3D Studio Max

## Materialien

*Maxwell* verzichtet auf weit verbreitete Shadermodelle wie Phong oder Lambert und arbeitet mit vollkommen neu entwickelten Materialien, die durchgehend auf physikalisch korrekten Oberflächeneigenschaften echter Materialien aufbauen.<sup>126</sup>

In der Testszene werden 2 verschiedene Materialien verwendet:

- Wand- und Würfelmaterial: Für die Wände der Cornell Box und die beiden Würfel wurde das *Maxwell* Basismaterial „Diffuse“ verwendet.
- Glasmaterial: Der Kugel und dem Zylinder wurden ein vollkommen durchsichtiges „Dielectric“ Material mit einem „Index of Refraction“ von 1,5 zugewiesen.

## Lichtquelle

In *Maxwell* gibt es keine vordefinierten Lichtquellen. Stattdessen kann jedes beliebige Objekt eine Lichtquelle darstellen. Dadurch lassen sich komplexe Lichtquellen wie eine Glühbirne realistisch darstellen. Um aus einem Polygonobjekt eine Lichtquelle zu machen, muss dem Objekt lediglich ein so genanntes „Emitter-Material“ zugewiesen werden. Die Einstellungen dieses Materials basieren ebenfalls auf physikalischen Werten. So werden die Farbtemperatur und die Lichtstärke der Lichtquelle nicht in RGB Werten, sondern in Kelvin bzw. Watt angegeben. Diese Werte wurden von der echten Cornell Box übernommen.

## Rendereinstellungen

In *Maxwell* werden Phänomene wie globale Beleuchtung oder Kaustiken nicht als Feature ein- und ausgeschaltet, sondern sind immer aktiv.<sup>127</sup> Da *Maxwell* auf einem unbiased *Metropolis Light Transport* Algorithmus basiert, muss der Benutzer einen Zeitpunkt definieren, an dem die Berechnung abgebrochen wird. Dies kann einerseits über eine festgelegte Berechnungszeit oder durch das Erreichen eines festgelegten Samplinglevels geschehen. Außerdem kann der Benutzer den Renderalgorithmus jederzeit händisch abbrechen.

---

<sup>126</sup> Vgl. Maxwell Render Handbuch, Seite 8

<sup>127</sup> Vgl. Maxwell Render Handbuch, Seite 8

## Ergebnis

Der Rendervorgang wurde nach 6h 01m 20sec bei einem Samplinglevel von 16 durch den Benutzer abgeschlossen, da keine sichtbaren Artefakte mehr im Bild zu erkennen waren.



Abbildung 40: Cornell Box gerendert mit Maxwell Render 1.1

### 11.4.3. Aufbau der Szene in Mental Ray (Softimage XSI)

Die Geometrie wurde direkt aus 3D Studio Max exportiert und in Softimage XSI importiert. Die Kameraeinstellung wurde wieder so gewählt, dass diese der Originalfotografie entspricht. Im Gegensatz zur *Maxwell* Kamera gibt es standardmäßig in Softimage keine Parameter für Shutter Speed und fStop.

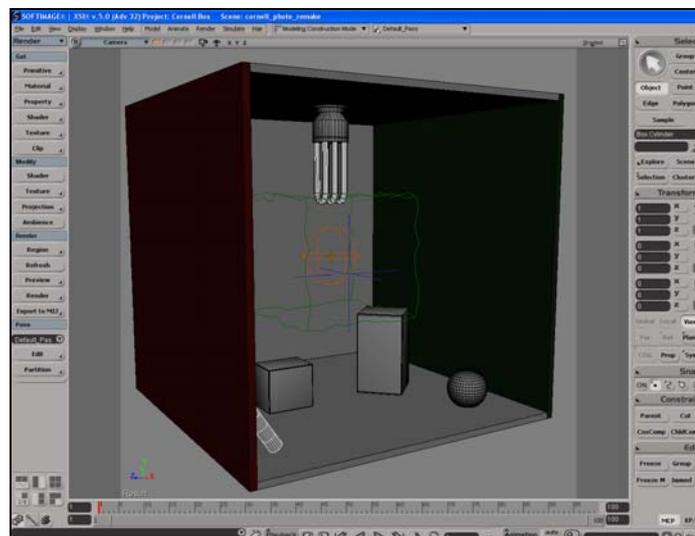


Abbildung 41: Aufbau der Szene in Softimage XSI

## Materialien

Den Wänden der Cornell Box, so wie den 2 Würfeln wurde ein einfacher Lambert Shader zugewiesen. Da es in der Natur kein hundertprozentig diffuses Material gibt, wurde die Reflectivity des Shaders auf einen Wert von 0,01 gesetzt. Die Kugel und der Zylinder verwenden ein Dielectric *Mental Ray* Material. Dieses Material ist standardmäßig nicht in Softimage integriert, sondern muss nachträglich als Addon installiert werden. Diese spezielle *Mental Ray* Material erlaubt eine physikalisch korrekte Simulation von dielektrischen Oberflächen.<sup>128</sup> Der „Index of Refraction“ wurde dabei auf 1,5 gesetzt.

## Lichtquelle

Softimage unterstützt keine Geometrie als Lichtquellen. Um die Glühbirne in der Cornell Box korrekt zu simulieren musste daher auf ein weiteres Addon zurückgegriffen werden. „Ctrl\_Geolight“ ermöglicht es jede Szenengeometrie als Lichtquelle zu verwenden. Weiters wurde als Shader für die Lichtquelle nicht der Standard-Softimage Shader, sondern der *Mental Ray* Shader „Physical\_Light“ verwendet. Dieser ermöglicht einen physikalisch korrekten Lichtabfall.<sup>129</sup> Die Farbtemperatur wird in Kelvin eingestellt, während die Lichtstärke durch einen abstrakten Zahlenwert bestimmt wird. Bei dieser Szene wurde letztendlich eine Lichtstärke von 200 gewählt. Dieser Wert wurde lediglich durch Testrenderings gefunden und steht in keiner Verbindung zur Lichtstärke der echten Glühbirne. Um in *Mental Ray* globale Beleuchtung zu simulieren, muss jede Lichtquelle als so genannter Photonenemitter aktiviert werden. Dabei wird auch die Anzahl der von der Lichtquelle ausgesendeten Photonen festgelegt. Ein passender Wert dafür kann wiederum lediglich durch Testrenderings gefunden werden und von keiner physikalischen Größe abgeleitet werden. In dieser Szene wurde die Anzahl der Photonen auf 1000000 gesetzt.

## Rendereinstellungen

In *Mental Ray* müssen globale Beleuchtung und Kaustiken extra aktiviert werden. Dies geschieht in den Renderoptionen unter dem Reiter „Photon“. Die Werte

---

<sup>128</sup> Vgl. Driemeyer, Rendering with Mental Ray, Seite 570

<sup>129</sup> Vgl. Driemeyer, Rendering with Mental Ray, Seite 568

„Accuracy“ und „Radius“ definieren die Anzahl bzw. den Radius der Photonen aus der Photon Map, die zur Berechnung eines Punktes herangezogen werden.<sup>130</sup> Bei dieser Szene wurden die Werte 250 für „Accuracy“ und 1 für „Radius“ gewählt.

### Ergebnis

Der Rendervorgang war nach 22m 01sec abgeschlossen. Davon wurden 16m 25sec zur Berechnung der Photon Map und 05m 26sec zur Berechnung des Bildes benötigt. Die Photon Map kann gespeichert werden und muss bei einem weiteren Rendervorgang nicht mehr berechnet werden.

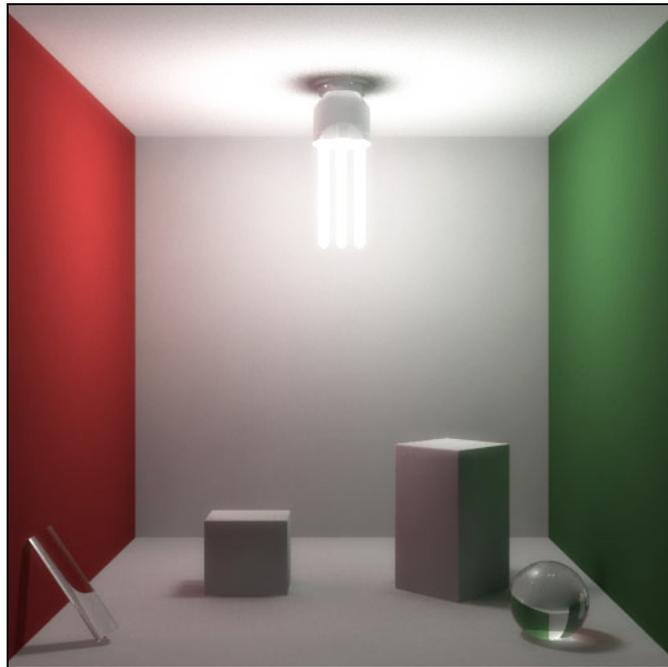


Abbildung 42: Cornell Box gerendert mit Mental Ray 3.4

## 11.5. Fazit

### 11.5.1. Beantwortung der gestellten Forschungsfragen

*Welche Beleuchtungs- und Rendertechniken bringen ein fotorealistic Ergebnis und in wieweit unterscheidet sich die Berechnungszeit?*

Um diese Frage beantworten zu können müssen 2 Fälle unterschieden werden:

- Das gerenderte Bild steht in direktem Vergleich zur Fotografie
- Das gerenderte Bild wird für sich alleine betrachtet

---

<sup>130</sup> XSI Documentation, Global Illumination

Wird das gerenderte Bild direkt mit einer Fotografie derselben Szene verglichen, so wird der Betrachter auf jeden Fall Unterschiede feststellen können. Das gerenderte Bild kann demnach nicht fotorealistic im Sinne einer Pixel-für-Pixel-Übereinstimmung sein. Nach dieser Definition kann mit heutigen Techniken kein fotorealistic Bild erstellt werden, da jeder Algorithmus lediglich auf Annäherungen basiert.

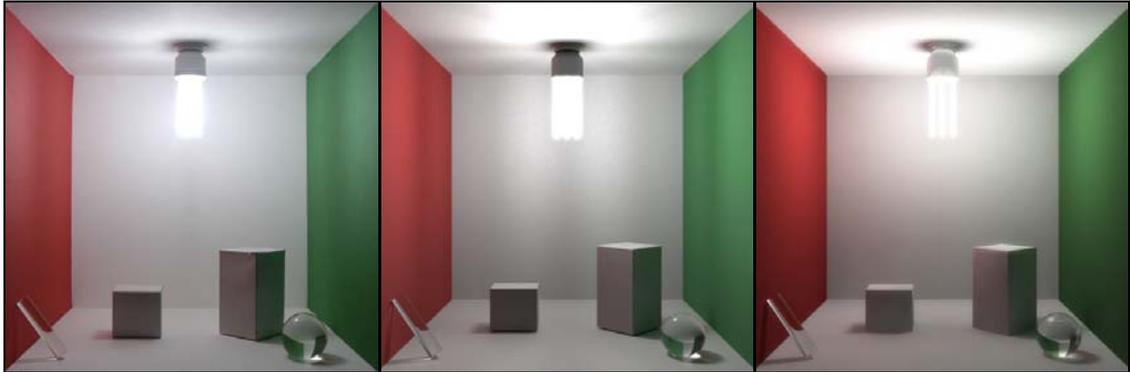


Abbildung 43: Vergleich der echten Cornell Box (links) mit der Maxwell Version (Mitte) und der Mental Ray Version (rechts).

Wird das gerenderte Bild für sich alleine betrachtet, so hat der Betrachter keine direkte Vergleichsmöglichkeit. In diesem Fall entscheidet weniger die absolute, physikalische Korrektheit, sondern die Glaubwürdigkeit des digitalen Bildes. In diesem Fall können sowohl *Mental Ray* als auch *Maxwell Render* fotorealistic Bilder berechnen. Eine größere physikalische Korrektheit, besonders bei Glasmaterialien und Kaustiken, wird mit *Maxwell Render* erreicht. Die Berechnungszeit liegt dabei aber bei 6h 01m 20sec. Ein verfrühter Abbruch des Rendervorgangs würde dabei zu einem hohen Rauschanteil im gerenderten Bild führen.



Abbildung 44: Abbruch des Rendervorgangs in Maxwell bei einem Samplinglevel von 5.

*Mental Ray* berechnet das Bild in 22m 01sec.

*Kann man die Einstellungen so weit optimieren, dass fotorealistisches Rendering auch für Animationen von Bedeutung ist?*

*Maxwell* gibt dem Benutzer keinerlei Optimierungsmöglichkeiten. Phänomene wie globale Beleuchtung, Kaustiken oder Tiefenunschärfe werden immer berechnet und können durch keine Parameter optimiert werden. Eine Berechnungszeit von 6h 01m 20sec pro Bild machen den Einsatz bei Animationen unmöglich.

*Mental Ray* berechnet das Bild in 22m 01sec und gibt dem Benutzer eine große Anzahl an Optimierungsmöglichkeiten. So kann die Berechnungszeit der *Photon Map* durch einfache Veränderungen der Parameter für Genauigkeit (Accuracy) und Radius weiter gesenkt werden. Bei Animationen sollten außerdem die Werte des Anti-Aliasing weiter gesenkt werden, da bei bewegten Bildern in den meisten Fällen eine geringere Kantenglättung ausreichend ist. Eine weitere Option ist das Speichern der *Photon Map*. Bei so genannten "Fly-Throughs" muss diese bei weiteren Bildern einer Animation nicht mehr neu berechnet werden.

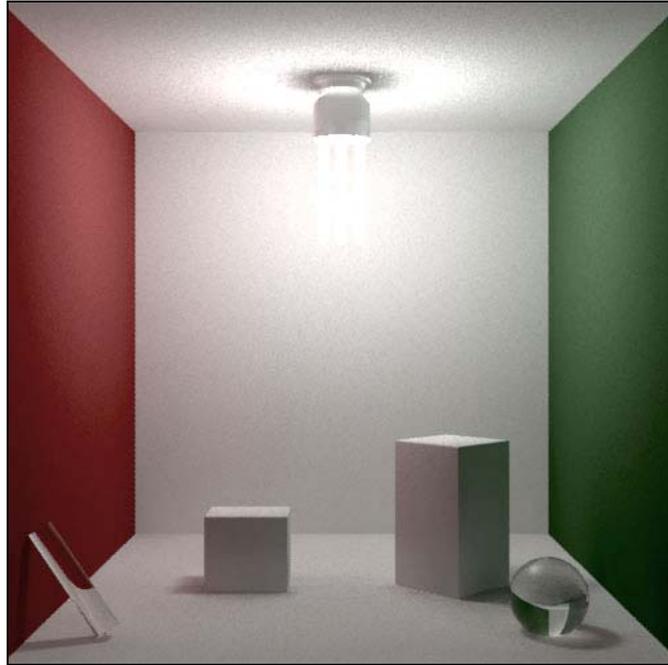


Abbildung 45: Cornell Box gerendert mit Mental Ray. Accuracy wurde auf 100 reduziert, Radius auf 5 erhöht. Die Berechnungszeit konnte dabei auf 3m 15sec reduziert werden.

### **11.5.2. Überprüfung der aufgestellten Hypothese**

*Je fotorealistischer das digitale Bild sein muss, desto eher muss die Berechnung auf physikalisch korrekten Verfahren basieren.*

Diese Hypothese wurde durch das praktische Beispiel bestätigt. Besonders die korrekte Berechnung von Lichtquellen und indirekter Beleuchtung hat großen Einfluss auf das Ergebnis. Gleichzeitig zeigt sich aber auch, dass die Berechnungszeit mit einer physikalisch korrekten Berechnung steigt. Dabei ist vor allem die Komplexität der Szene entscheidend. Szenen mit überwiegend direkter Beleuchtung und diffusen Oberflächen können wesentlich schneller berechnet werden als Szenen mit indirekter Beleuchtung und mehreren spiegelnden und durchsichtigen Objekten. Eine physikalisch korrekte Berechnung bei Animationen ist mit heutiger Technik und Hardware auf Grund der viel zu langen Rendervorgänge nicht sinnvoll. Auch wenn es absehbar ist, dass durch Weiterentwicklungen die Berechnungszeit in *Maxwell* verringert werden kann, wird sich das Einsatzgebiet vorwiegend auf Architektur- und Produktvisualisierungen beschränken.

## 12. Zusammenfassung

In dieser Diplomarbeit wurde ein Überblick über verschiedene Renderverfahren zur Berechnung von globaler Beleuchtung gegeben. Dabei wurden zunächst die Grundlagen der Lichtausbreitung erklärt und anschließend die Schwierigkeiten, die sich bei einer physikalisch korrekten Lichtberechnung ergeben, erläutert.

Unterschiedliche Ansätze werden dabei zur Lösung der Rendergleichung angewendet. Monte-Carlo-Verfahren ermöglichen ein äußerst fotorealistisches Ergebnis, in dem sie die Rendergleichung annähernd lösen. Die Rendersoftware *Maxwell Render* von Next Limit Technologies, welche auf einem *Metropolis Light Transport* Algorithmus basiert, ermöglicht es Phänomene wie globale Beleuchtung, Kaustiken oder Tiefenunschärfe physikalisch korrekt darzustellen und legt dabei auf eine möglichst einfache Bedienung Wert. Dies ermöglicht auch relativ unerfahrenen Benutzern fotorealistische Bilder zu generieren. Eine sehr hohe Berechnungszeit verhindert dabei allerdings den Einsatz bei Animationen. Im Gegensatz dazu ermöglicht die Rendersoftware *Mental Ray* die Berechnung von globaler Beleuchtung in einem Bruchteil der Zeit und bietet eine große Anzahl an Optimierungsmöglichkeiten. Dadurch ist auch die Berechnung von realistischen Animationen möglich, allerdings auf Kosten einer absolut physikalisch korrekten Lichtberechnung.

Obwohl es mit heutigen Techniken möglich ist ein digitales Bild zu generieren, das von einer Fotografie nicht mehr unterschieden werden kann, gibt es immer noch viele Fragen, die gelöst werden müssen. Einige Renderalgorithmen arbeiten in speziellen Situationen effizienter als andere. Ziel sollte ein Algorithmus sein, der sich szenenabhängig an unterschiedliche Lichtsituationen anpasst und somit eine effizientere Berechnung von globaler Beleuchtung ermöglicht. Außerdem sollten eine physikalisch korrekte Berechnung und gleichzeitig eine große Anzahl an Parametern zur Optimierung der Renderzeit in eine Rendersoftware integriert werden. Je nach Anwendungsgebiet könnte vom Benutzer der gewünschte Grad an Realismus eingestellt werden. Weiters wird es in Zukunft möglich sein, Oberflächeneigenschaften von realen Materialien zu messen und daraus Modelle für den Einsatz in der Computergrafik zu entwickeln. Durch die rasch steigende Leistung der Computerhardware wird fotorealistisches Rendering immer mehr Anwendungsgebiete finden.

## 13. Glossar

### **Algorithmus**

Unter einem Algorithmus versteht man allgemein eine genau definierte Handlungsvorschrift zur Lösung eines Problems oder einer bestimmten Art von Problemen.

### **Artefakt**

Artefakte sind Fehler in computergenerierten Bildern, die aufgrund von Rundungsfehlern und mathematischen Annäherungen entstehen.

### **Beugung**

Die Beugung oder Diffraktion ist die "Ablenkung" von Wellen an einem Hindernis.

### **Computergrafik**

Die Computergrafik ist ein Teilbereich der Informatik, der sich mit der künstlichen Erzeugung von Bildern mit Hilfe von Computern beschäftigt.

### **Cornell Box**

Die Cornell Box ist ein Test, um die Genauigkeit einer Rendersoftware zu ermitteln. Dabei wird zuerst eine reale Box gebaut und mit einer CCD Kamera fotografiert. Alle wichtigen Daten der Lichtquelle, sowie sämtliche Abmessungen der Box werden notiert. Auf Basis dieser Daten wird anschließend eine 3D Szene nachgestellt und ein digitales Bild berechnet. Dieses wird mit der Originalfotografie der echten Cornell Box verglichen.

### **CPU**

Der Hauptprozessor (CPU) wird im allgemeinen Sprachgebrauch oft auch als Prozessor bezeichnet. Die CPU ist der zentrale Prozessor eines Computers, der alle anderen Bestandteile steuert.

### **Finite-Element-Methode**

Die Finite-Elemente-Methode ist ein numerisches Verfahren zur näherungsweise Lösung von Differentialgleichungen.

**Fly-Throughs**

Fly-Throughs sind virtuelle Rundgänge durch eine 3D-Szene, bei der nur die Kamera in Bewegung ist.

**Frame**

Ein Frame ist ein Einzelbild einer Animation. Eine Sekunde besteht in dem PALSystem aus 25 Frames.

**Globale Beleuchtung**

Globale Beleuchtung (engl. Global Illumination) bezeichnet in der 3D-Computergrafik die Simulation aller Möglichkeiten der Ausbreitung von Lichtstrahlen in einer 3D-Szene.

**Index of Refraction**

Der Index of Refraction kennzeichnet die Brechung des Lichts beim Übergang in ein transparentes Material.

**Interferenz**

Interferenz beschreibt die Überlagerung von zwei oder mehr Wellen.

**Kaustiken**

Als Kaustik bezeichnet man ein geometrisches Helligkeitsmuster, das durch Intensitätsüberhöhung an der Einhüllenden eines breiten Strahlenbündels auftritt.

**Pixel**

Pixel bezeichnet sowohl die kleinste Einheit einer digitalen Rastergrafik als auch deren Darstellung auf einem Bildschirm mit Rasteransteuerung.

**Polygon**

Ein Polygon erhält man, indem man mindestens drei voneinander verschiedene Punkte in einer Zeichenebene durch Strecken so miteinander verbindet, dass eine geschlossene Figur entsteht. Dreiecke, Vierecke und Sechsecke sind aus dem Alltag bekannte Beispiele für Polygone.

### **Postproduktion**

Postproduktion umfasst sämtliche Aktionen, die nötig sind, um einen bereits abgedrehten Film fertig zustellen. Es ist der allgemeine Begriff für alle Stadien der Produktion zwischen Rohmaterial und fertigen Filmmaterial.

### **Radiosity**

Radiosity ist ein Verfahren zur Berechnung der Lichtverteilung in der 3D-Computergrafik, das auf dem Energieerhaltungssatz beruht: alles Licht, das eine Fläche empfängt und nicht absorbiert, muss sie wieder reflektieren.

### **Raytracing**

Raytracing ist ein auf der Aussendung von Strahlen basierender Algorithmus zur Ermittlung der Sichtbarkeit von dreidimensionalen Objekten ab einem bestimmten Punkt im Raum.

### **Rendern**

Umsetzung von dreidimensionalen Modellen in eine zweidimensionale Darstellung.

### **Sampling**

Sampling ist die Registrierung von Messwerten zu diskreten Zeitpunkten. Aus einem zeitkontinuierlichen Signal wird so ein zeitdiskretes Signal gewonnen.

### **Shading**

Shading bezeichnet in der 3D-Computergrafik im allgemeinen Sinne die Simulation der Oberfläche eines Objekts.

### **Textur**

Textur bezeichnet ein Bild, welches auf der Oberfläche eines virtuellen Körpers dargestellt wird.

### **Visual Effects**

Als Visuelle Effekte bzw. Visual Effects bezeichnet man Effekte in Filmen, die im Gegensatz zu Spezialeffekten nicht in Camera also am Set gedreht, sondern künstlich, heutzutage meist mit Computern, in der Postproduktion erzeugt werden.

## 14. Quellenverzeichnis

- [Dutre, P., Bekaert, P., Bala, K.] „Advanced Global Illumination“, 2003
- [Pharr, M., Humphreys, G.] „Physically Based Rendering“, 2004
- [Kajiya, J. T.] „The Rendering Equation“, 1986
- [Veach, E., Guibas L. J.] „Metropolis Light Transport“, 1997
- [Jensen, H. W.] „A Practical Guide To Global Illumination Using Photon Maps“, 2000
- [Bentley, J. L.] „Multidimensional Binary Search Trees Used for Associative Searching“, 1975
- [Ward, G. J., Rubinstein, F. M., Clear, R. D.] „A Ray Tracing Solution For Diffuse Interreflection, 1988
- [Driemeyer, T.] „Rendering With Mental Ray“, 2005
- [Birn, J.] „Digital Lighting & Rendering“, 2006
- [Kerlow, I. V.] „The Art Of 3D Computer Animation and Effects“, 2004
- Maxwell Render Handbuch
- Softimage XSI Handbuch
- [Wissen.de], <http://www.wissen.de/wde/generator/wissen/ressorts/bildung/index?page=1212736.html>
- [Wissen.de], <http://www.wissen.de/wde/generator/wissen/ressorts/bildung/index?page=1222296.html>
- [Wissen.de], <http://www.wissen.de/wde/generator/wissen/ressorts/technik/computer/index?page=1224188.html>
- [Wikipedia Online], [http://de.wikipedia.org/wiki/Globale\\_Beleuchtung](http://de.wikipedia.org/wiki/Globale_Beleuchtung)
- [Wikipedia Online], <http://de.wikipedia.org/wiki/Quantenoptik>
- [Wikipedia Online], <http://de.wikipedia.org/wiki/Radiometrie>
- [Wikipedia Online], <http://de.wikipedia.org/wiki/Fotometrie>
- [Wikipedia Online], [http://de.wikipedia.org/wiki/Reflexion\\_%28Physik%29](http://de.wikipedia.org/wiki/Reflexion_%28Physik%29)
- [Wikipedia Online], [http://de.wikipedia.org/wiki/Snelliussches\\_Brechungsgesetz](http://de.wikipedia.org/wiki/Snelliussches_Brechungsgesetz)
- [Wikipedia Online], <http://de.wikipedia.org/wiki/Shading>
- [Wikipedia Online], [http://de.wikipedia.org/wiki/Flat\\_Shading](http://de.wikipedia.org/wiki/Flat_Shading)
- [Wikipedia Online], [http://de.wikipedia.org/wiki/Gouraud\\_Shading](http://de.wikipedia.org/wiki/Gouraud_Shading)
- [Wikipedia Online], <http://de.wikipedia.org/wiki/Energieerhaltungssatz>

- [Wikipedia Online], [http://de.wikipedia.org/wiki/Radiosity\\_%28Computergrafik%29](http://de.wikipedia.org/wiki/Radiosity_%28Computergrafik%29)
- [Computerbase], [http://www.\[Computerbase\].de/lexikon/Rendergleichung](http://www.[Computerbase].de/lexikon/Rendergleichung)
- [Computerbase], [http://www.\[Computerbase\].de/lexikon/Raytracing](http://www.[Computerbase].de/lexikon/Raytracing)
- [Computerbase], [http://www.\[Computerbase\].de/lexikon/Sichtbarkeitsproblem](http://www.[Computerbase].de/lexikon/Sichtbarkeitsproblem)
- [Computerbase], [http://www.\[Computerbase\].de/lexikon/Nyquist-Shannon-Abtasttheorem](http://www.[Computerbase].de/lexikon/Nyquist-Shannon-Abtasttheorem)
- [Cornell University Program of Computer Graphics], <http://www.graphics.cornell.edu/online/box/>
- [Maxwell Render], <http://www.maxwellrender.com/>
- [Maxwell Render Forum], <http://www.maxwellrender.com/forum/viewtopic.php?t=14254>
- [Mental Images], [http://www.mentalimages.com/2\\_1\\_0\\_mentalray/index.html](http://www.mentalimages.com/2_1_0_mentalray/index.html)

## 15. Abbildungsverzeichnis

Abbildung 1: Unterschiedliche Reflexionsverhalten.....	18
Abbildung 2: Gerendertes Bild mit diffuser, glänzender und spiegelnder Reflexion .....	18
Abbildung 3: Flat Shading.....	22
Abbildung 4: Gouraud Shading.....	23
Abbildung 5: Phong Shading.....	24
Abbildung 6: Eine typische Raytracing Szene mit vielen Reflexionen.....	27
Abbildung 7: Indirekte Beleuchtung .....	28
Abbildung 8: Abtasten einer Funktion.....	30
Abbildung 9: Aliaseffekte bei hochfrequenten Funktionen.....	31
Abbildung 10: Checkerboard mit verschiedenen Abtastraten.....	32
Abbildung 11: Szene mit unterschiedlichen Samplern gerendert.....	34
Abbildung 12: Pfadverfolgung beim Pathtracing.....	37
Abbildung 13: Sponza Atrium Model ausschliesslich mit direktem Licht gerendert. ....	38
Abbildung 14: Sponza Atrium Model gerendert mit 8 Samples pro Pixel.....	39
Abbildung 15: Sponza Atrium Model gerendert mit 1024 Samples pro Pixel. ....	39
Abbildung 16: Lichtsituation, die mit Pathtracing nur schwer berechnet werden kann .....	40
Abbildung 17: Reines Path Tracing erzeugt hohes Rauschen.....	41
Abbildung 18: Optimiertes Bild gerendert mit Bidirectional Pathtracing. ....	41
Abbildung 19: Sponza Atrium Szene gerendert mit Irradiance Caching und Path Tracing .....	44
Abbildung 20: Position der gespeicherten Samplewerte.....	44
Abbildung 21: Pfadmutationen.....	51
Abbildung 22: Lens Perturbations. ....	52
Abbildung 23: Caustic Perturbations. ....	52
Abbildung 24: Beispiel Bidirektionales Pathtracing.....	54
Abbildung 25: Beispiel Metropolis Light Transport .....	54
Abbildung 26: Vergleich Bidirektionales Pathtracing und Metropolis Light Transport. ....	55
Abbildung 27: Beispiel Pathtracing.....	56
Abbildung 28: Beispiel Metropolis Light Transport.....	56
Abbildung 29: Ausstrahlung verschiedener Lichtquellen .....	58
Abbildung 30: Die Erzeugung einer Kaustik Map und einer globalen Photon Map.....	62
Abbildung 31: Kd-Baum .....	63
Abbildung 32: Auswahl der umliegenden Photonen .....	65
Abbildung 33: Verbesserung der Auswahl der Photonen.....	66
Abbildung 34: Kaustiken gerendert mit Pathtracing und mit Photon Mapping.....	70
Abbildung 35: Cornell Box gerendert mit Pathtracing und Photon Mapping.....	71
Abbildung 36: Sponza Atrium Szene gerendert mit Photon Mapping und Pathtracing.....	71
Abbildung 37: Modell der echten Cornell Box .....	74
Abbildung 38: Referenzfoto der echten Cornell Box .....	75

Abbildung 39: Aufbau der Szene und Kameraeinstellungen in 3D Studio Max .....	75
Abbildung 40: Cornell Box gerendert mit Maxwell Render 1.1 .....	77
Abbildung 41: Aufbau der Szene in Softimage XSI.....	77
Abbildung 42: Cornell Box gerendert mit Mental Ray 3.4 .....	79
Abbildung 43: Vergleich der Ergebnisse .....	80
Abbildung 44: Cornell Box gerendert mit Maxwell – abgebrochen bei Samplinglevel 5.....	81
Abbildung 45: Cornell Box gerendert mit Mental Ray mit optimierten Einstellungen .....	82