Visualisierungsinstitut der Universität Stuttgart
Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Studienarbeit Nr. 2308

Globale Volumenbeleuchtung mit Photon Mapping und Path Tracing

Zeno-Oliver Groß

Studiengang:	Informatik
Prüfer:	Prof. Daniel Weiskopf
Betreuer:	Dipl.-Inf. Marco Ament
begonnen am:	1. Dezember 2010
beendet am:	1. Juni 2011
CR-Klassifikation:	I.3.3, I.3.7

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

In der fotorealistischen Bildsynthese kann nicht auf globale Beleuchtung verzichtet werden. Das gilt für die Beleuchtung von Oberflächen wie auch für volumetrische Daten. Ersteres ist heute sehr gut mit den verschiedensten Verfahren berechenbar. Letzteres hingegen ist immer noch sehr aufwändig, wenn neben der einfachen Streuung von Licht auch die mehrfache Streuung simuliert werden soll.

Methoden zur Simulation des Lichtransportes in partizipierenden Medien gibt es in großer Zahl. Einige diskretisieren das Medium und berechnen wie in [Rus94] die Isotrope Streuung in Anlehnung an das Radiosity Verfahren als Austausch von Radiance unter den Volumen- und Oberflächenelementen. Andere Verfahren modellieren anisotrope Streuung mit Hilfe von Spherical Harmonics, wie z. B. in [KH84]. Eine weitere Anwendung von Spherical Harmonics ist die Repräsentation der Beleuchtung je Volumenelement aus [BT92]. Die Unterteilung des Volumens funktioniert gut, solange es keine schärferen Kanten gibt. An dieser muss weiter diskretisiert werden, was die Komplexität deutlich erhöht.

Zwei Verfahren basierend auf der Monte Carlo Integration für die globale Beleuchtung von Oberflächen sind zum einen das Photon Mapping und zum anderen Path Tracing. Durch seine Geschwindigkeit und die Fähigkeit Kaustiken zu berechnen ist Photon Mapping meist die erste Wahl. Jedoch bleibt Path Tracing - was die Genauigkeit des Ergebnisses angeht - das überlegenere Verfahren. Diese Verfahren sollen in dieser Arbeit vorgestellt und je eine Erweiterung gezeigt werden, die den Lichttransport mit Mehrfachstreuung innerhalb eines Medium simulieren kann. Dazu wurden mehrere wichtige Quellen mit [JC98], [LW96] und [PKK00] herangezogen, die diese oder ähnliche Erweiterungen eingeführt haben. Zusätzlich dazu wird mit Hilfe von Perlin Noise [Per85] eine Methode vorgestellt, die der niedrig aufgelöste volumetrische Datensätze verfeinert werden können. Diese Verfahren wurden im Rahmen dieser Arbeit in das Renderingframework „pbrt“ [PH10a] implementiert. Die Resultate aus den jeweiligen Berechnungen mit pbrt sollen hier gezeigt und die Verfahren miteinander verglichen werden. Abschließend werden noch mögliche Punkte gezeigt an denen angeknüpft werden kann.

Gliederung

Die Arbeit ist in folgender Weise gegliedert:

Kapitel 2 – pbrt stellt kurz das zu Grunde liegende Framework, pbrt vor.

Kapitel 3 – Streuung an Volumina steckt die Grundlagen ab.

Kapitel 4 – Path Tracing beschreibt das Verfahren für die Beleuchtung von Oberflächen und stellt anschließend die Erweiterung und Aspekte der Implementierung vor.

Kapitel 5 – Photon Mapping erläutert ebenso zuerst das Basisverfahren und geht schließlich auf die Erweiterung und dessen Implementierung ein.

Kapitel 6 – Perlin Noise beschreibt eine Möglichkeit einen Volumendatensatz zu verfeinern

Kapitel 7 – Ergebnisse zeigt Renderings einiger Beispielscene und vergleicht die vorgestellten Verfahren miteinander

Kapitel 8 – Zusammenfassung und Ausblick fasst die Ergebnisse der Arbeit zusammen und stellt Anknüpfungspunkte vor.

2 pbrt

Mit pbrt haben M. Pharr G. Humphreys einen Renderer entwickelt, der möglichst akuell, vielseitig und erweiterbar sein soll. Die Implementierung basiert zum Zeitpunkt der Arbeit auf der zweiten Ausgabe des Buches [PH10b], in dem sie dokumentiert ist. Zwei wichtige Verfahren für die Berechnung der globalen Beleuchtung, das Photon Mapping und das Path Tracing, haben auch ihren Platz in pbrt und bieten damit die Basis für diese Studienarbeit. Der Ausgangspunkt für die Implementierung der Erweiterungen war der Quellcode vom 18.11.2010 und wurde im Verlauf der Arbeit aktualisiert.

2.1 Renderingarchitektur

Das Design von pbrt ist weitgehend modular gehalten. Jeder Bestandteil ist somit beinahe beliebig austauschbar ohne die anderen Funktionen zu beeinträchtigen. So gibt es `Film`-Klassen für die Ausgabe der Berechungen. Wobei hier nicht nur das Schreiben in Dateien gemeint ist sondern auch eine andere Art der Ausgabe denkbar sein könnte: z. B. über ein Netzwerk verschicken, in den Framebuffer der Grafikkarte schreiben und vieles mehr.

Die Informationen werden der `Film`-Klasse in der Regel durch die `Renderer`-Klasse übergeben. Auch hier ist das Design sehr allgemein gehalten, denn es wird dem `Renderer` überlassen wie und welche Beleuchtungsinformationen erzeugt werden. Beispielsweise berechnet die Klasse `MetropolisRenderer` alle Informationen selbst, die Klasse `SamplerRenderer` verteilt Aufgaben für die Oberflächen- und Volumenbeleuchtung und der Renderer der Klasse `CreateRadianceProbes` ermittelt die einfallende Beleuchtung für eine Menge von Punkten, umgeht die `Film`-Klasse und schreibt diese Informationen in eine Textdatei zur weiteren Verarbeitung.

Für die meisten Anwendungen, wie auch das Generieren von Bildern, wird die `SamplerRenderer`-Klasse verwendet. Bei dieser sind die Berechungen der Radiance aus Oberflächen- und Volumeninteraktionen jeweils auf einen `SurfaceIntegrator` und einen `VolumeIntegrator` verteilt. Das bietet die Möglichkeit verschiedene Verfahren miteinander zu kombinieren und pbrt um neue Verfahren zu erweitern, wie bei dieser Studienarbeit für die Erweiterung des Photon Mapping auf partizipierende Medien.

Der `SmaplerRenderer` verschießt beim Rendering mehrere Strahlen von der Kamera durch einen Bildpunkt, die durch eine (wieder beliebig austauschbare) `Sampler`-Klasse generiert sind. Der ermittelte Schnittpunkt wird einem `SurfaceIntegrator` übergeben, der für diesen Punkt die Beleuchtung berechnen soll. Im Anschluß soll der `VolumeIntegrator` den Beitrag

der Beleuchtung entlang des Strahles ermitteln. Für die Erweiterung des Path Tracing wird es jedoch nicht möglich sein dieses Konzept so weiterzuverwenden.

Mit pbrt können außerdem auch Effekte wie Tiefenunschärfe und Motion Blur, viele einstellbare Arten von Materialeigenschaften und neben der Beleuchtung mit den gängigsten Lichtquellen auch durch Environment Maps benutzt werden. Da die Berechnung der Schnittpunkte von Strahlen mit Oberflächenprimitiven sehr häufig vorkommt sind auch zwei Beschleunigungsstrukturen mit der Binary-Volume-Hierarchy und den *k*d-Bäumen implementiert.

Volumina sind immer in einem Quader einbeschrieben, können aber beliebig miteinander kombiniert werden. Volumendaten können als Homogener Quader, in der Dichte exponentiell abnehmend oder voxelisiert auftreten. An dieser Stelle wurde pbrt auch um prozedural erzeugbare Volumendaten mit Hilfe von Perlin Noise erweitert, welches in Kapitel 6 weiter ausgeführt wird.

3 Streuung an Volumina

Dieses Kapitel möchte zuerst die Grundlage für weitere Betrachtungen schaffen indem die grundlegenden Größen angerissen werden und anschließend die Verteilung von Radiance (Strahldichte) in einer Umgebung mit partizipierenden Medien betrachtet werden. Ein umgebendes Medium beeinflusst die Radiance entlang eines Strahles in dreierlei Weise:

Absorption Wird Licht in eine andere Form der Energie umgewandelt (z.B. in Wärme) reduziert sich dessen Intensität.

Emission Im Gegensatz zur Absorption, wird hier Energie in Licht umgewandelt und an die Umgebung abgestrahlt.

Streuung Beeinflusst die Ablenkung des Lichtes hervorgerufen durch Interaktion mit den Partikeln des Mediums.

3.1 Radiance und andere grundlegende Größen

Bevor weiter auf die Interaktionen mit partizipierenden Medien eingegangen wird, soll der nun folgende Abschnitt die grundlegenden Größen, die in dieser Studienarbeit benötigt werden, nennen. Die Notationen sind zum großen Teil aus [PH10b] oder [Kol05] übernommen und dort weiter ausgeführt.

Sei $A \subset \mathbb{R}^3$ die Gesamtheit aller Oberflächen einer betrachteten Szene und $V := \mathbb{R}^3 \setminus A$, wobei $\mathrm{d}A(x)$ bzw. $\mathrm{d}V(x)$ eine differentiell kleine Fläche bzw. Volumen sind. Nun lässt sich ein differentieller Raumwinkel $\mathrm{d}\omega$ in Richtung ω mit $\omega := \frac{x-y}{|x-y|}$ schreiben als

$$\mathrm{d}\omega = \frac{|\omega \cdot n_x|}{|x-y|^2} \mathrm{d}A(x) \quad , \tag{3.1.I}$$

eine Projektion einer differentiell kleinen Fläche bei x (mit Normale n_x) auf die Einheitskugel bei y. Analog dazu ist ein differentiell kleines Volumen $dV(x)$ bestimmt durch den differentiellen Raumwinkel $\mathrm{d}\omega$ von y aus durch

$$\mathrm{d}V = |x-y|^2 \, \mathrm{d}\omega \mathrm{d}r \quad , \tag{3.1.II}$$

wobei $\mathrm{d}r$ die differentielle Dicke (bzw. Länge) des Volumenelementes ist. Dies wird später für diverse Umrechnungen benötigt.

Zuletzt bestimmen wir mit der Strahldichte (im weiteren Verlauf „Radiance“) L die Grundlegende Größe für die Beleuchtungsberechnungen. Sie gibt an wieviel Strahlungsleistung

(bzw. -fluss), $\mathrm{d}\Phi(x,\omega)$ pro projizierte Flächeneinheit $\mathrm{d}A(x)$ und Raumwinkelelement $\mathrm{d}\omega$ von x abgestrahlt wird:

$$L(x,\omega) = \frac{\mathrm{d}\Phi(x,\omega)}{|\omega \cdot n_x| \, \mathrm{d}A(x)\mathrm{d}\omega}$$

3.2 Absorption und heraus gestreute Radiance

Nun gehen wir von einem differentiell kleinen Volumenelement der Länge $\mathrm{d}r$ aus, das entlang eines Strahles in Richtung ω sitzt. Für die resultierende Radiance nach einer Absorption ergibt sich nach [PH10b],

$$\mathrm{d}L(x,\omega) = -\sigma_a(x,\omega)L(x,-\omega)\mathrm{d}r.$$

σ_a, der Absorptionskoeffizient, gibt an zu welchen Teilen die einfallende Radiance absorbiert wird. Analog dazu verhält es sich mit der heraus gestreuten Radiance:

$$\mathrm{d}L(x,\omega) = -\sigma_s(x,\omega)L(x,-\omega)\mathrm{d}r$$

Ebenso wie σ_a gibt σ_s, der Streukoeffizient, an zu welchen Teilen die einfallende Radiance heraus gestreut wird.

Um die gesamte Reduktion der Radiance zu berechnen bilden wir die Summe der beiden Koeffizienten und erhalten mit σ_t, den Extinktionskoeffizient aus

$$\sigma_t(x,\omega) = \sigma_a(x,\omega) + \sigma_s(x,\omega).$$

Sind σ_a und σ_s ortsunabhängig so ist das Medium homogen bzw. uniform.

Schließlich ergibt sich mit σ_t eine Differentialgleichung, die die gesamte Abschwächung beschreibt:

$$\mathrm{d}L(x,\omega) = -\sigma_t(x,\omega)L(x,-\omega)\mathrm{d}r \tag{3.2.I}$$

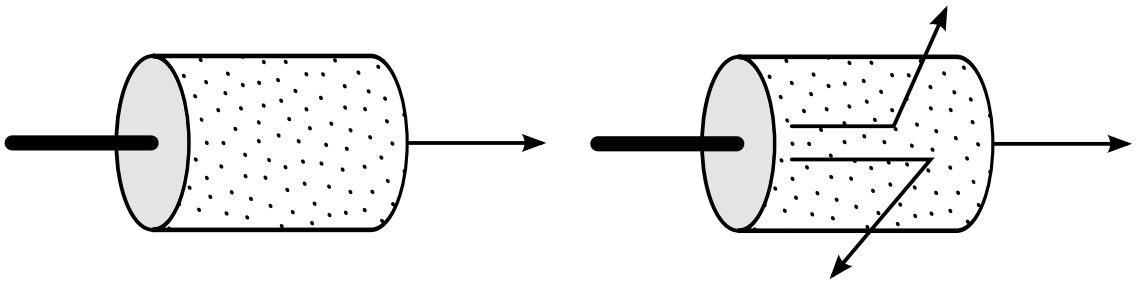

Abbildung 3.1: Die resultierenden Radiance wird im Volumenelement durch Absorption (links) und Herausstreuung (rechts) abgeschwächt.

Eine Lösung für diese Gleichung ist der Transmissionskoeffizient, τ. Das ist der Anteil an Radiance, der zwischen zwei Punkten eines Strahls transportiert wird:

$$\tau(x', x) = e^{-\int_{x'}^{x} \sigma_t(\xi,\omega)\,\mathrm{d}\xi} \tag{3.2.II}$$

mit ω, dem normalisierten Richtungsvektor von x' nach x.

3.3 Emission und eingestreute Radiance

Für ein differentiell kleines Volumenelement der Länge dr berechnet sich die Änderung der Radiance bei der Emission aus

$$\mathrm{d}L(x, \omega) = L_{e,V}(x, \omega)\mathrm{d}r.$$

$L_{e,V}$ ist hierbei die „volume emmitance"-Funktion des Volumens V zur Beschreibung von volumetrischen Lichtquellen wie z. B. Feuer oder Plasma.

Genauso wie bei der Emission verstärkt die eingestreute Radiance das Resultat. Deshalb lassen sich die Anteile im „source term", S, vereinen:

$$\mathrm{d}L(x, \omega) = S(x, \omega)\mathrm{d}t \tag{3.3.I}$$

mit

$$S(x, \omega) = L_{e,V}(x, \omega) + \sigma_s(x, \omega) \int_{S^2} f_p(\omega, x, \omega')L(x, \omega')\,\mathrm{d}\omega'. \tag{3.3.II}$$

Der hintere Teil der Gleichung (3.3.II) beschreibt den Anteil an akkumulierter Radiance am Punkt x über alle Richtungen der Einheitskugel S^2, gewichtet mit der Phasenfunktion f_p.

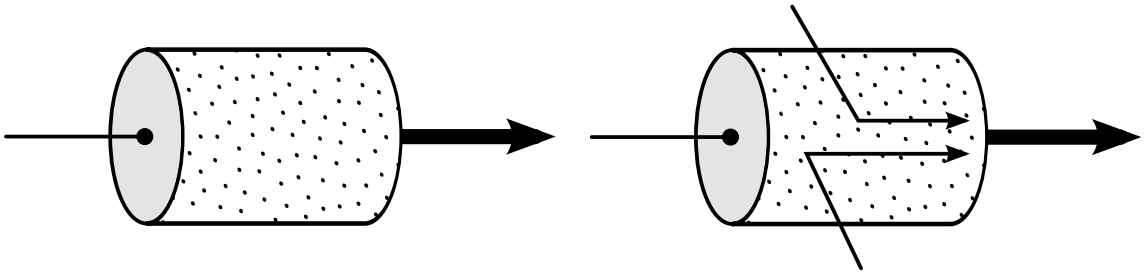

Abbildung 3.2: Die resultierenden Radiance wird im Volumenelement durch Emission (links) und Einstreuung (rechts) verstärkt.

3.4 Die Phasenfunktion

Die Phasenfunktion f_p gibt an zu welchen Teilen die Radiance bei der Interaktion mit dem Medium in eine andere Richtung gestreut wird. Ist sie richtungsunabhängig so spricht man von isotroper Streuung und ansonsten von anisotroper Streuung. Darüber hinaus gilt für eine Phasenfunktion

$$\forall x \forall \omega \quad \int_{S^2} f_p(\omega, x, \omega') \, \mathrm{d}\omega' = 1$$

wodurch die Energieerhaltung gewährleistet ist.

In der Computergrafik wird sehr oft die Phasenfunktion von Henyey und Greenstein, die 1941 in [HG41] eingeführt wurde, eingesetzt. Sie wurde so konzipiert, damit sie möglichst einfach an gemessene Streudaten angepasst werden kann und ist abhängig von nur einem Parameter, dem Anisotropieparameter g. Durch diesen beeinflusst man wie viel Radiance in oder entgegengesetzt der Einstreuungsrichtung gestreut wird:

$$f_{p,HG}(\omega, x, \omega') = \frac{1}{4\pi} frac1 - g^2(1 + g^2 - 2g(\omega \cdot \omega'))^{3/2)}$$

Der Anisotropieparameter kann darüber hinaus für eine willkürliche Phasenfunktion nach [PH10b] bestimmt werden aus dem Intergral

$$g = \int_{S^2} f_p(\omega, x, \omega')(\omega \cdot \omega') \, \mathrm{d}\omega' \quad .$$

Jedoch reicht im Allgemeinen ein Parameter nicht aus um eine beliebige Phasenfunktion auszudrücken, wodurch das Einsetzten des ermittelten Anisotropieparameters in $f_H G$ die ursprüngliche Funktion nicht exakt widerspiegelt.

Möchte man aber mit einer beliebigen Phasenfunktion, wie sie in pbrt konzipiert ist, Importance Sampling betreiben ist dies im Allgemeinen sehr aufwändig. Hier könnte man sich die Einfachheit der Henyey-Greenstain Phasenfunktion zu Nutze machen: Zuerst wird der Anisotropieparameter angenähert und anschließend damit eine ausgehende Richtung abgetastet[1].

3.5 Lichttransport bei Oberflächen

Auf den Oberflächen beschreibt nach [PKK00] die lokale Streuungsgleichung den Transport von Radiance. Diese ist eine Erweiterung der Renderinggleichung aus [Kaj86], die nur die Reflektion von Radiance betrachtet, auf Streuungseffekte durch Lichtbrechung:

$$L(x, \omega) = L_{e,A} + \int_{S^2} f_s(\omega, x, \omega') L(x, \omega') \langle n_x, \omega' \rangle \, \mathrm{d}\omega' \tag{3.5.I}$$

[1] Dieses Verfahren wird für beide Erweiterungen in pbrt genutzt.

Hier ist $L_{e,A}$ die emittierte Radiance der Oberfläche. S^2 ist auch hier die Einheitskugel, ω' eine Richtung daraus und $L(x, \omega')$ die einfallende Radiance aus ω'. Diese ist gewichtet mit dem Skalarprodukt aus der Oberflächennormale n_x und ω' und der „bidirectional scattering distribution function“ (BSDF), f_s.

3.6 Lichttransport bei partizipierenden Medien

Innerhalb des Mediums soll die nachfolgende Gleichung aus [Cha50] der Ausgangspunkt sein. Diese beschreibt die Änderung der Radiance in Richtung ω und beinhaltet die Beiträge aus emmitierter, eingestreuter, absorbierter und heraus gestreuter Radiance:

$$\begin{aligned} \omega \cdot \nabla L(x, \omega) = L_{e,V}(x, \omega) + \sigma_s(x, \omega) \int_{S^2} f_p(\omega, x, \omega') L(x', \omega') \, \mathrm{d}\omega' \\ - \sigma_t(x, \omega) L(x, \omega) \end{aligned} \tag{3.6.I}$$

Hier können Teile von (3.2.I) und (3.3.I) wiedererkannt werden.

Integriert man nun auf beiden Seiten entlang ω von x_A bis x, so ergibt sich nach [PKK00] mit der Randbedingung (3.5.I) die folgende Gleichung:

$$\begin{aligned} L(x, \omega) = \int_{x_A}^{x} \tau(x', x) \left[L_{e,V}(x', \omega) + \sigma_s(x') \int_{S^2} f_p(\omega, x', \omega') L(x', \omega') \, \mathrm{d}\omega' \right] \mathrm{d}x' \\ + \tau(x_A, x) L(x_A, \omega) \end{aligned} \tag{3.6.II}$$

Hier sind $x \in V$ und $x_A \in A$. x_A lässt sich berechnen mit dem „ray tracing“-Operator $h(x, \phi)$, der den nächsten Oberflächenpunkt in Richtung ϕ liefert.

3.7 Streuereignis entlang eines Strahles

Zwischen zwei Punkten p_1 und p_2 bestimmt man den Punkt $p_s = p_1 + d \cdot \omega$ für das Streuereignis mit Hilfe des Transmissionskoeffizienten τ nach [PKK00] und [JC98] ein wenig umgestellt aus

$$\ln(1 - \xi) = \int_0^d \sigma_t(p_1 + t\omega, \omega) \, \mathrm{d}t \quad .$$

ξ sei eine uniforme Zufallsvariable aus $[0, 1)$. d wird mit Ray Marching entlang des Strahles in Richtung ω durch p_1 und p_2 berechnet. Hierbei wird der Strahl in Stücke aufgeteilt, welche jeweils als homogen angesehen werden, bei denen sich also das lokale σ_t nicht ändert. Nun werden die jeweiligen σ_t so lange aufsummiert bis $\ln(1 - \xi)$ erreicht ist. Wurde ein d gefunden, so ist ein Streuereignis eingetreten und der Wert der PDF für dieses Ereignis ist nach [LW96] $p_{medium} = \sigma_t(p_s, \omega) \cdot \tau(p_1, p_s)$.

Wurde kein d gefunden, so ist auch kein Streuereignis eingetreten, es gab demnach eine Interaktion mit einer Oberfläche und der Wert der PDF dafür ist nach [LW96] die a posteriori Wahrscheinlichkeit für eine Oberflächeninteraktion, $P_{surface} = \tau(p_1, p_s)$.

Mit dieser Grundlage können nun die beiden Verfahren, Photon Mapping und Path Tracing, so erweitert werden, um damit Szenen mit partizipierenden Medien berechnen zu können.

4 Path Tracing

Das Path Tracing ist in seiner ursprünglichen Form ein grundlegendes Verfahren für das Rendering von Computergraphiken. Es wurde von J. Kajiya in [Kaj86] mit der Renderinggleichung als eine Lösung dessen eingeführt. Dieses Verfahren gilt als eine Referenz, da es die globale Beleuchtung vollständig berechnen kann [PH10b]. Es verfolgt Pfade ausgehend von der Kamera zurück zu den Lichtquellen. Da pro Bildpunkt sehr viele Pfade möglich sind, ist in der Regel das Konvergenzverhalten dieses Verfahrens schlecht. Jedoch ist die Konvergenz zum korrekten Ergebnis garantiert.

4.1 Die Renderinggleichung

Grundlage für das Path Tracing bildet die Renderinggleichung bzw. die „light transport equation“ (LTE). Diese beschreibt die Verteilung von Radiance in der Szene, welche sich im Energiegleichgewicht befindet. Sie gibt die abgestrahlte Radiance L in Richtung ω für einen Punkt p an (siehe: (3.5.I)).

$$L(p,\omega) = L_e(p,\omega) + \int_{S^2} f_s(\omega,p,\omega')L(p,\omega')\langle n_p,\omega'\rangle \,\mathrm{d}\omega'$$

Hier wird angenommen, dass die Radiance entlang eines Strahls konstant bleibt, d.h. partizipierende Medien werden nicht mit einbezogen. Als nächstes betrachten wir mit $h(p,\omega)$ den Raytracing-Operator, der den nächsten Oberflächenpunkt der Szene in Richtung ω liefert. Nun können wir L unter dem Integral der LTE in

$$L(p,\omega') = L(h(p,\omega'),-\omega') \tag{4.1.I}$$

umschreiben. Auf der rechten Seite von (4.1.I) steht nun die am Punkt x' in Richtung $-\omega$ abgestrahlte Radiance. Wenn $p_{i+1} = h(p_i,\omega)$ lässt sich also vereinfacht $L(p_i,\omega) = L(p_{i+1},-\omega) = L(p_{i+1} \to p_i)$ und analog dazu $f_s(p_{i+1} \to p_i \to p_{i-1})$ schreiben.

Analog zu der obigen Form der LTE über Richtungen ist es möglich diese über alle Oberflächen der Szene zu formulieren:

$$L(p_1 \to p_0) = L_e(p_1 \to p_0) + \int_A f_s(p_2 \to p_1 \to p_0)L(p_2 \to p_1)G(p_2 \leftrightarrow p_1)\,\mathrm{d}A(p_2) \tag{4.1.II}$$

G ist der Geometrieterm, in dem implizit der Raytracing Operator und die Umrechnung von differentiellen Richtungen auf differentielle Flächen beinhaltet:

$$G(p \leftrightarrow p') = V(p \leftrightarrow p')\frac{|\cos\theta|\,|\cos\theta'|}{\|p-p'\|^2}$$

Hier ergibt $V = 1$, wenn p' von p aus direkt sichtbar ist und 0 sonst.

4.2 Integrale über Pfade

Nach dieser Grundlage kann man auf eine zweckmäßigere Form übergehen: eine Summe über Pfade, die Licht (zur Kamera) transportieren. Dadurch ist es schließlich möglich mit Hilfe von Monte-Carlo-Integration den Path-Tracing-Algorithmus zu formulieren.

Sei p_0 der Bezugspunkt der Kamera und p_1 ein direkt sichtbarer Schnittpunkt mit einer Oberfläche der Szene, so ergibt sich nach [PH10b] durch ineinander Einsetzen von (4.1.II) in sich selbst:

$$\begin{aligned}
L(p_1 \to p_0) = {} & L_e(p_1 \to p_0) \\
& + \int_A L_e(p_2 \to p_1) f_s(p_2 \to p_1 \to p_0) G(p_2 \leftrightarrow p_1) \, \mathrm{d}A(p_2) \\
& + \int_A \int_A L_e(p_3 \to p_2) \\
& \quad \cdot f_s(p_3 \to p_2 \to p_1) f_s(p_2 \to p_1 \to p_0) \\
& \quad \cdot G(p_3 \leftrightarrow p_2) G(p_2 \leftrightarrow p_1) \, \mathrm{d}A(p_3) \, \mathrm{d}A(p_2) \\
& + \int_A \int_A \int_A L_e(p_4 \to p_3) \\
& \quad \cdot f_s(p_4 \to p_3 \to p_2) f_s(p_3 \to p_2 \to p_1) f_s(p_2 \to p_1 \to p_0) \\
& \quad \cdot G(p_4 \leftrightarrow p_3) G(p_3 \leftrightarrow p_2) G(p_2 \leftrightarrow p_1) \, \mathrm{d}A(p_3) \, \mathrm{d}A(p_2) \, \mathrm{d}A(p_2) \\
& + \cdots
\end{aligned}$$

Jeder Summand auf der rechten Seite repräsentiert den gesamten Beitrag aller Pfade der jeweiligen Länge: der erste Summand Pfade der Länge null, der zweite Pfade der Länge eins, der dritte Pfade der Länge zwei und so weiter.

Dies lässt sich auch in kompakter Form schreiben:

$$L(p_1 \to p_0) = \sum_{n=1}^{\infty} P(\bar{p}_n) \tag{4.2.I}$$

$P(\bar{p}_n)$ steht für die die Radiance über die Pfade $\bar{p}_n = p_0, p_1, p_2, ..., p_n$ mit $n+1$ Punkten und ist

$$\begin{aligned}
P(\bar{p}_n) = {} & \underbrace{\int_A \int_A \cdots \int_A}_{n-1} L_e(p_n \to p_{n-1}) \\
& \cdot \left(\prod_{i=1}^{n-1} f_s(p_{i+1} \to p_i \to p_{i-1}) G(p_{i+1} \leftrightarrow p_i) \right) \mathrm{d}A(p_2) \cdots \mathrm{d}A(p_n).
\end{aligned} \tag{4.2.II}$$

Für diese Summe kann nun eine Näherung mit der Monte-Carlo Integration berechnet werden: Für eine vorgegebene Länge n können je (mehrdimensionalem) Integral Pfade abgetastet werden und daraus mit (4.2.II) deren Beiträge ermittelt werden, um schließlich durch (4.2.I) die gesamte Radiance zu berechnen.

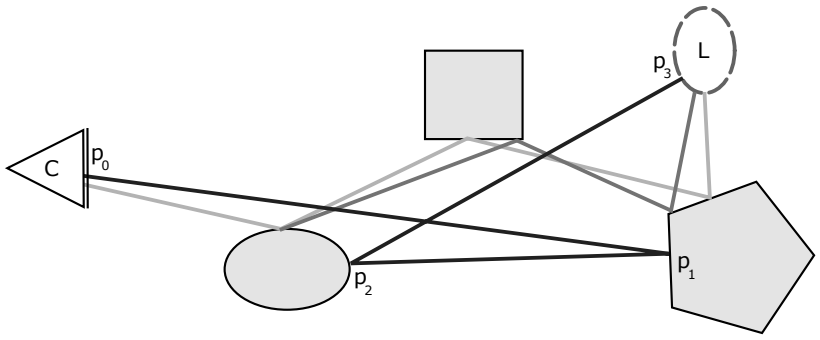

Abbildung 4.1: So könnten einzelne Pfade von der Kamera zu einer Lichtquelle aussehen. Dabei ist zu beachten, um das korrekte Ergebnis zu erhalten, müssten je Abschnitt unendlich viele neue Pfade generiert werden. Um die Varianz, die sich durch Rauschen widerspiegelt, zu verringern ist der letzte Punkt eines hier erzeugten Pfades stets der einer Lichtquelle.

4.3 Verfahren für Oberflächen

Als erstes muss eine Methode gefunden werden, mit der die Summe in (4.2.I) berechnet werden kann. Es ist natürlich nicht möglich den Wert dieser unendlichen Summe vollständig zu ermitteln. Daher findet die Berechnung wie in [PH10b] bis zu einer festen Pfadlänge statt und bei jedem weiteren Summand i bricht diese mit einer Wahrscheinlichkeit q_i ab.

Dies entspricht im Grunde der Idee des russischen Roulettes, bei dem ein Integrand für bestimmte Samples mit einer Wahrscheinlichkeit q nicht berechnet sondern durch eine Konstante (meist 0) ersetzt wird. Mit der Wahrscheinlichkeit $1 - q$ wird der Integrand ausgerechnet, jedoch mit $1/(1-q)$ gewichtet. Dadurch können Berechnungen eingespart werden ohne, dass sich der Erwartungswert insgesamt ändert [PH10b]. Deshalb bleibt in diesem Falle die Näherung der unendlichen Summe im Mittel korrekt.

Die Näherung ist dann für 3 feste Iterationen:

$$\begin{aligned} L(p_1 \to p_0) = {} & P(\bar{p}_1) + P(\bar{p}_2) + P(\bar{p}_3) \\ & + \frac{1}{1-q_4}\left(P(\bar{p}_4) + \frac{1}{1-q_5}\left(P(\bar{p}_5) + \frac{1}{1-q_6}\left(P(\bar{p}_6) + \cdots\right.\right.\right. \end{aligned} \tag{4.3.I}$$

Als nächstes müssen die Punkte eines Pfades ermittelt werden. Für ein korrektes Ergebnis müssen diese gleich verteilt aus allen Oberflächen zufällig gewählt sein. Unter der Annahme es gäbe ein Verfahren diese zu bestimmen ist ein Punkt p_i mit der Wahrscheinlichkeit

$$P_A(p_i) = \frac{1}{\sum_j A_j}$$

gewählt, wobei $\sum_j A_j$ der gesamten Fläche der Szene entspricht. Dies kommt daher, dass die Wahrscheinlichkeit das k-te Objekt mit der Fläche A_k zu wählen gerade $\frac{A_k}{\sum_j A_j}$ ist und die Wahrscheinlichkeit auf einer Fläche k einen Punkt zu wählen $1/A_k$ ist.

Nun ist es aber oft schwierig einen Pfad auf uniform gewählten Punkten zu generieren, der dann auch noch einen signifikanten Beitrag liefert, denn gerade wenn zwei Punkte nicht gegenseitig sichtbar sind ist dieser 0. Deshalb wird der Pfad nun inkrementell aufgebaut indem durch den Raytracing-Operator der nächste Punkt ermittelt wird und so ist der Sichtbarkeitsterm V immer eins. Dies hat große Auswirkung auf das Endergebnis, denn die Varianz der Lösung, also das auftretende Rauschen im Bild, wird so stark verringert. Um die Varianz der Lösung weiter zu verringern wird der letzte Punkt eines Pfades auf eine Lichtquelle gesetzt. Dies ist insofern sinnvoll, dass vor allem bei Lichtquellen mit kleiner Ausdehnung es sehr unwahrscheinlich ist diese zu treffen.

Die Richtung, in der der nächste Punkt des Pfades liegen soll, kann man beispielsweise durch die BSDF oder auch dem Skalarprodukt (also dem Kosinus aus Oberflächennormale und der Ausgangsrichtung) unter dem Integral bestimmen. Da nun Importance-Sampling angewandt wird, muss der Pfad jeweils mit der reziproken PDF gewichtet sein. Die Oberflächenform dafür ist P_A. Dies kann man nach [PH10b] für Raumwinkel umformuliert in:

$$P_A = P_\omega \frac{|\cos\theta_i|}{\|p_i - p_{i+1}\|^2} \tag{4.3.II}$$

sodass sich insgesamt für den Beitrag eines Pfades der Länge n aus 4.2.II ergibt:

$$\frac{L_e(p_n \to p_{n-1}) f_s(p_n \to p_{n-1} \to p_{n-2}) \left|cos\theta_{n-1}\right|}{P_{L_e}(p_n)} \left(\prod_{j=i}^{i-2} \frac{f_s(p_{j+1} \to p_j \to p_{j-1}) \left|cos\theta_j\right|}{P_\omega(p_{j+1} - p_j)} \right) \tag{4.3.III}$$

mit P_L der Entsprechung von P_A für die Oberflächen der Lichtquellen.

Bei diesem Verfahren wird schließlich für jeden Pixel die Summe aus (4.3.I) und jeder Summand mit Hilfe des obigen Produktes berechnet. Es müssen aber ausreichend viele Pfade der jeweiligen Länge generiert werden, um ein Ergebnis mit wenig Varianz zu erhalten.

4.4 Erweiterung auf partizipierende Medien

Nachdem die Methode für das Rendering von Oberflächen vorgestellt ist, kommen wir nun zu einer Erweiterung um auch die Streuung an partizipierenden Medien korrekt berechnen können. Hier betrachten wir alle Streueffekte aus dem Kapitel 3, die auch mehrfach auftreten können, also Emission, Abschwächung und herein- und heraus gestreute Radiance.

Grundlage hierfür ist Gleichung (3.6.II) aus Kapitel 3. Diese beschreibt die Radiance in Richtung ω als eine Summe bestehend aus der eingestreuten bzw. emittierten Radiance

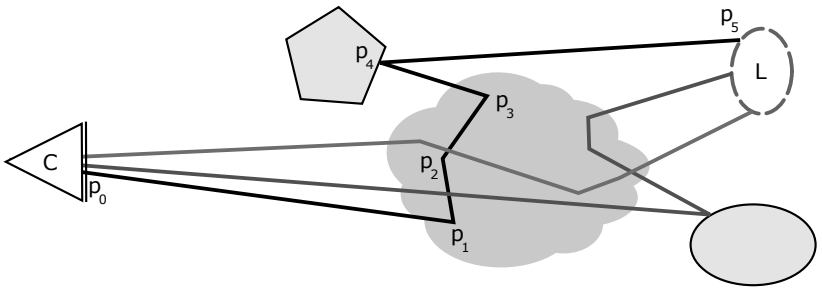

Abbildung 4.2: Auch hier sind einzelne Pfade von der Kamera zur Lichtquelle zu sehen, die jedoch durch Interaktion mit dem Medium (mehrfach) abgelenkt wurden. Auch bei der Erweiterung lässt man die Pfade stets an der Lichtquelle enden um die Varianz zu verringern.

entlang eines Strahles zwischen dem Oberflächenpunkt x_A und x und der ankommenden Radiance L_A von x_A:

$$L(p,\omega) = \int_{p_A}^{p} \tau(p',p)\left[L_{e,V}(p',\omega) + \sigma_s(p')\int_{S^2} f_p(\omega,p',\omega')L(p',\omega')\,\mathrm{d}\omega'\right]\mathrm{d}p' + \tau(p_A,p)L_A(p_A,\omega) \tag{4.4.I}$$

mit dem Transmissionskoeffizient τ, die sich durch (3.2.II) berechnet, L_{e,V^0}, der emittierten Radiance des Mediums, σ_s, dem Streukoeffizienten und f_p, der Phasenfunktion. Nun lässt sich die Radiance definieren, die an einem Punkt des Volumens p_1 in Richtung des Punktes p_0 gestreut oder dort hin emittiert wurde [Kolo5], als

$$L(p_1 \to p_0) = \begin{cases} L_{e,A}(p_1,\overrightarrow{p_1p_0}) + \int_{S^2} f_s(\overrightarrow{p_1p_0},p_1,\omega')L(p_1,\omega')\left|\omega'\cdot n_{p_1}\right|\,\mathrm{d}\omega' & p_1 \in A \\ L_{e,V}(p_1,\overrightarrow{p_1p_0}) + \int_{S^2} f_p(\overrightarrow{p_1p_0},p_1,\omega')L(p_1,\omega')\,\mathrm{d}\omega' & p_1 \in V \end{cases} \tag{4.4.II}$$

mit $\overrightarrow{p_1p_0} = \frac{p_0-p_1}{|p_0-p_1|}$.

Darüber hinaus erhält man aus (4.4.I) indem $p_2 = p_A = h(p_1,\omega)$ und $\tau(p',p) = \tau(p' \leftrightarrow p)$ gesetzt wird mit $p_2 = p_1(r) = p_1 - r\omega$

$$L(p_1,\omega) = \begin{cases} \int_0^r \tau(p_1(r') \leftrightarrow p_1)L(p_1(r') \to p_1)\,\mathrm{d}r' + \tau(p_2 \leftrightarrow p_1)L(p_2 \to p_1) & r < \infty \\ \int_0^\infty \tau(p_1(r') \leftrightarrow p_1)L(p_1(r') \to p_1)\,\mathrm{d}r' & \textit{sonst} \end{cases}.$$

Wird (4.4.II) in die obere Gleichung eingesetzt und die Bereiche der Integrale über alle Richtungen ω' und alle Entfernungen r' durch $\mathbb{R}^3$ ersetzt erhält man mit (3.1.I) und (3.1.II) die „drei Punkte Form“, die es anschließend möglich macht wieder ein Integral über Pfade zu formulieren, welches jetzt auch Streueung an partizipierenden Medien miteinbezieht:

$$L(p_1 \to p_0) = L_e(p_1 \to p_0) + \int_{\mathbb{R}} f(p_2 \to p_1 \to p_0)L(p_2 \to p_1)G(p_2 \leftrightarrow p_1)\,\mathrm{d}\lambda(p_2) \tag{4.4.III}$$

mit

$$\mathrm{d}\lambda(p) = \begin{cases} \mathrm{d}A & p \in A \\ \mathrm{d}V & p \in V \end{cases},$$

der zusammengefassten emitterten Radiance

$$L_e(p_1 \to p_0) := \begin{cases} L_{e,A}(p_1, \overrightarrow{p_1 p_0}) & p_1 \in A \\ L_{e,V}(p_1, \overrightarrow{p_1 p_0}) & p_1 \in V \end{cases},$$

der Streuungsfunktion

$$f(p_2 \to p_1 \to p_0) := \begin{cases} f_s(p_2 \to p_1 \to p_0) & p_1 \in A \\ \sigma_s(p_1) \cdot f_p(\overrightarrow{p_1 p_0}, p_1, \overrightarrow{p_2 p_1}) & p_1 \in V \end{cases}$$

und dem generalisierten Geometrieterm aus [PKKoo]

$$G(p \leftrightarrow p') = V(p \leftrightarrow p') \frac{D(p, p') \cdot D(p', p)}{|p - p'|^2} \tau(p \leftrightarrow p')$$

mit

$$D(p, p') = \begin{cases} \left| \overrightarrow{p\,p'} \cdot n_p \right| & p \in A \\ 1 & \textit{sonst} \end{cases}.$$

Analog zum Verfahren für Oberflächen erhält man, wenn (4.4.III) in sich selbst eingesetzt wird, wieder die unendliche Summe (4.2.I), nun aber über Pfade mit Punkten aus ganz $\mathbb{R}^3$. Dadurch bekommt $P(\bar{p}_n)$ nun die Form

$$\begin{aligned} P(\bar{p}_n) = \underbrace{\int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \cdots \int_{\mathbb{R}^3}}_{n-1} & L_e(p_n \to p_{n-1}) \\ & \cdot \left(\prod_{i=1}^{n-1} f(p_{i+1} \to p_i \to p_{i-1}) G(p_{i+1} \leftrightarrow p_i) \right) \mathrm{d}\lambda(p_2) \cdots \mathrm{d}\lambda(p_n). \end{aligned} \tag{4.4.IV}$$

Genauso wie bei dem einfachen Verfahren für die Oberflächenbeleuchtung sollen auch hier die Pfade inkrementell erzeugt werden und währenddessen die jeweiligen $P(\bar{p}_n)$ berechnet werden. Im weiteren Verlauf wird bei der Beschaffenheit des Mediums vom allgemeinsten Fall ausgegangen, das Medium ist inhomogen und die Phasenfunktion ist nicht die von Henyey und Greenstein.

Zusätzlich zum einfachen Path Tracing wird nun entlang eines Strahles durch ein Medium ein Punkt p_s gesucht, bei dem ein Streuereignis eintritt. Wird p_s mit dem Verfahren aus Abschnitt 3.7 gefunden, dann wird der Strahl gemäß der Phasenfunktion mit Importance Sampling abgelenkt. Ebenso analog zum Verfahren für Oberflächen wird während der

Konstruktion des Pfades bei jedem Streuereignis (Volumen- und Oberflächenstreuung) die direkte Beleuchtung mit einberechnet. Dadurch spart man sich auch hier Rechenzeit, denn sonst würden zusätzlich kürzere Pfade erzeugt werden müssen.

Wie eben beschrieben muss nicht nur die neue Richtung mit der Phasenfunktion sondern auch die Punkte der Streuereignisse mit Importance Sampling bestimmt werden. Für die Normierung sind die Angaben zu den PDF bzw. deren Werte aus [LW96] übernommen und an die Formeln hier angepasst. Die PDF für eine Interaktion mit dem Medium ist

$$P_{V,p,\omega}(p_s) = \sigma_t p_s, \omega\tau(p, p_s)$$

und die a posteriori Wahrscheinlichkeit für die Oberflächenstreuung ist

$$P_{A,p,\omega}(p_A) = \tau(p, p_A) \quad .$$

Die PDF für das Importance Sampling einer Phasenfunktion ist nach [Kolo5] die Phasenfunktion selbst.

Bei der inkrementellen Erzeugung des Pfades ist es nun wünschenswert eine Formel für dessen Beitrag an Radiance analog zu (4.3.III) zu erhalten. Und in Anlehnung an die inkrementell berechnete PDF des gesamten Pfades aus [LW96] findet sich für diesen Fall auch eine Umrechnung von der PDF eines Raumwinkels P_ω zu einer PDF eines Volumenelemts bzw. Oberflächenelements P_λ:

$$P_\lambda = P_\omega \cdot \frac{D(p_i, p_{i+1})}{|p_i - p_{i+1}|^2} \cdot P_{scatter}(p_i, p_{i+1})$$

mit

$$P_{scatter}(p_i, p_{i+1}) = \begin{cases} \tau(p_i \leftrightarrow p_{i+1}) & p_i \in A \\ \sigma_t(p_i)\tau(p_i \leftrightarrow p_{i+1}) & p_i \in V \end{cases}$$

Ist p_i nun ein Oberflächenpunkt entspricht dies bis auf $\tau(p_i, p_{i+1})$ der Gleichung (4.3.II).

Nun lässt sich auch für diese Erweiterung der Beitrag eines Pfades der Länge n, ausgehend von der Gleichung (4.4.IV), formulieren zu

$$\frac{L_e(p_n \to p_{n-1}) f(p_n \to p_{n-1} \to p_{n-2}) D(p_{n-1}, p_n) \tau(p_{n-1}, p_n)}{P_{L_e}(p_n)} \cdot \left(\prod_{j=i}^{i-2} \frac{f(p_{j+1} \to p_j \to p_{j-1}) D(p_j, p_{j+1}) P_{scatter}(p_i, p_{i+1})}{P_\omega(p_{j+1} - p_j)} \right)$$

Wie auch im Verfahren für Oberflächen beschreibt das hintere Produkt den „Durchsatz“ des Pfades.

Der letzte Punkt des Pfades ist hier wiederum der auf einer Lichtquelle. Da ein Medium auch Radiance emittiert und bei diesem Verfahren nur einen Strahl (dessen Richtung ein Sample von f war) durch das Medium abgetastet wird, ist P_{L_e} folgendermaßen definiert:

Im Falle einer Oberfläche ist es die PDF für einen Ort aus allen Flächen der Lichtquellen. Bei einem Volumenelement steht es für das Produkt aus f (der PDF der Ereignisse für eine Ablenkung des Strahles) mit $\sigma_t \cdot \tau$ (der PDF der Streuereignisse entlang eines Strahles), denn die beiden Ereignisse sind nach [LW96] voneinander unabhängig.

Darauf aufbauend ist die Implementierung der Erweiterung von pbrt für das Rendering von partizipierenden Medien mit globaler Beleuchtung entstanden, die im nächsten Abschnitt erläutert wird.

4.5 Implementierung

Die bestehende Implementierung des Path Tracing in pbrt nutzt das modulare Modell des `SamplerRenderer`, der in Kapitel 2 vorgestellt wurde. Das bedeutet, dass für die Oberflächenbeleuchtung ein `SurfaceIntegrator` und für die Volumenbeleuchtung ein `VolumeIntegrator` eingesetzt wird. Darüber hinaus berechnet der `SamplerRenderer` vorab den ersten Schnittpunkt mit den Objekten der Szene und übergibt diesen an den `SurfaceIntegrator`. Dieses Konzept passt nicht ganz auf das erweiterte Verfahren wie es im vorherigen Abschnitt eingeführt wurde.

Deshalb wird ein `VolumePathRenderer` eingeführt, dessen Implementierung zu großen Teilen von dem `SamplerRenderer` übernommen ist. Dieser neue Renderer benutzt einen eigenen `VolumePathIntegrator` für die Beleuchtungsberechnung. Diese Klasse ist der einfachen Implementierung halber vom `SurfaceIntegrator` abgeleitet. Korrekter wäre es direkt von der `Integrator`-Klasse abzuleiten. Um den Sampler aber weiterhin modular zu halten würde dies auch eine Umschreibung dessen bedeuten, worauf im Rahmen dieser Arbeit verzichtet wurde. Bei der Implementierung wurde darauf geachtet alle anderen Features des `SamplerRenderer` (bis eben auf die Angabe des Integrator) erhalten bleiben.

Der Algorithmus für die Erweiterung des Path Tracer auf partizipierende Medien gliedert sich in folgende Abschnitte:

1. Bestimmen eines Punktes bei dem ein Streuereignis eintritt
2. Berechnen des jeweiligen Beitrages und Akkumulieren der direkten Beleuchtung entweder im Falle von Oberflächeninteraktion oder der Interaktion mit einem Medium
3. Abbruch der Berechnung per russischem Roulette oder sobald die maximale Pfadlänge erreicht ist
4. Bestimmen der Richtung des neuen Strahles ausgehend von der aktuellen Position gemäß der Phasenfunktion bei Volumenstreuung bzw. der BSDF bei Oberflächenstreuung

Der Beitrag aus der Oberflächeninteraktion wurde aus der Implementiereung des `PathIntegrator` übernommen. Bei der Interaktion mit dem Medium ist der Beitrag zur

Gesamtbeleuchtung gegliedert in Emission durch das Volumen und der direkten Beleuchtung einer Lichtquelle. Der Beitrag aus der Emission ist

$$L_{e+} = T_p(x_{i-1}) \cdot \frac{1}{\sigma_t(x_i, -\omega_{ray})} L_{e,V}(x_i, -\omega_{ray})$$

und aus der direkten Beleuchtung ist

$$L_{d+} = T_p(x_i) \cdot L_{e,A}(x_L \to x_i)\tau(x_L \leftrightarrow x_i)\frac{f_p(x_L \to x_i \to x_{i-1})}{P_{L_e}(x_L)}$$

wobei ω_{ray} die Richtung des Strahles, x_L ein Ortssample einer Lichtquelle und $T_p(x_i)$ der Pfaddurchsatz ist mit

$$T_p(x_i) = \frac{\sigma_s(x_i, -\omega_{ray})}{\sigma_t(x_i, -\omega_{ray})} T_p(x_{i-1}) \quad .$$

Die Abbruchbedingung ist ebenso direkt aus dem `PathIntegrator` übernommen. Die nächste Richtung wird mit Importance Sampling der Phasenfunktion bestimmt. Wie schon in Abschnitt 3.4 beschrieben ist das Sampling einer allgemeinen Phasenfunktion sehr aufwändig. Stattdessen wird hier der Anisotropieparameter für die Henyey-Greenstein-Phasenfunktion bestimmt und diese dann abgetastet.

5 Photon Mapping

Photon Mapping wurde ursprünglich durch H. W. Jensen in [Jen96] für die globale Beleuchtung von Oberflächen eingeführt. Dieses Verfahren basiert auf der Idee der Simulation von Lichtpartikeln, durch diese kann die globale Beleuchtung simuliert werden. Mit Photon Mapping kann in der Regel in relativ kurzer Zeit ein gutes Resultat erzeugt werden. Im Allgemeinen ist es ein Verfahren mit Bias, d.h. das korrekte Ergebnis ist dadurch im Allgemeinen nicht berechenbar. Jedoch wird gerade durch seine kurze Laufzeit und durch seine Stärke bei der Berechnung von Kaustiken sehr oft eingesetzt.

Ein Bild wird nun in zwei Pässen berechnet: Im ersten Pass werden, wie oben genannt, Wege von Partikeln von den Lichtquellen aus simuliert. Diese Partikel, im weiteren Verlauf auch „Photonen“, hinterlassen dann auf den jeweiligen Oberflächen ihre Spuren, welche in einer Datenstruktur, der Photon Map, gespeichert sind. Der zweiten Pass ist im Grunde das Rendering eines angepassten Ray Traycers, der zusätzlich mit Hilfe der Photon Maps die indirekte Beleuchtung und Kaustiken berechnet.

5.1 Näherung der Radiance

Zunächst soll auch hier die Annahme sein, dass sich Radiance in der Szene im Energiegleichgewicht befindet und entlang eines Strahles nicht ändert. Daher bietet die Renderinggleichung erneut die Basis für dieses Verfahren. Sie wurde im Kapitel 3 bereits vorgestellt und später mit dem Path Tracing in Kapitel 4 aufgegriffen:

$$L(p,\omega) = L_e(p,\omega) + \underbrace{\int_{S^2} f_s(\omega,p,\omega')L(p,\omega')\left|n_p\cdot\omega'\right|\,\mathrm{d}\omega'}_{=L_r}$$

Um das Integral auf der rechten Seite zu lösen benötigt man die Information über die Radiance aus der jeweiligen Richtung. Die Idee ist nun, dass eine Photon Map den ankommenden Strahlungsfluss für eine Paar aus Punkt und Richtung, (x,ω), beschreibt. Deshalb lässt sich nach [Jeno1] ein Integral wie L_r mit

$$L_i(p,\omega') = \frac{\mathrm{d}^2\Phi(p,\omega')}{\left|n_p\cdot\omega'\right|\,\mathrm{d}\omega'\mathrm{d}A(p)}$$

aus Abschnitt 3.1 umschreiben in

$$\begin{aligned} L_r &= \int_{S^2} f_s(\omega, p, \omega') \frac{\mathrm{d}^2\Phi(p,\omega')}{\left|n_p \cdot \omega'\right| \mathrm{d}\omega' \mathrm{d}A(p)} \left|n_p \cdot \omega'\right| \mathrm{d}\omega' \\ &= \int_{S^2} f_s(\omega, p, \omega') \frac{\mathrm{d}^2\Phi(p,\omega')}{\mathrm{d}\omega' \mathrm{d}A(p)} \mathrm{d}\omega' \end{aligned}$$

Dieses Integral wurde in [Jen96] durch die Summe

$$L_r \approx \sum_{i=1}^{N} f_s(\omega, p, \omega_i) \frac{\Delta\Phi_i(p,\omega_i)}{\Delta A(p)} \tag{5.1.I}$$

mit $\Delta A(p) = \pi r_p^2$ approximiert.

Dieser Betrachtungsweise sehr ähnlich ist die Theorie des „Particle Tracing“ aus [Vea97], Anhang 4.A. Sie ist jedoch etwas allgemeiner gehalten und passt auch besser auf die Erweiterung des Photon Mapping, denn sie ist die Grundlage der Implementierung in pbrt.

Diese Theorie interpretiert die „Spuren“, die ein Photon auf den Oberflächen hinterlässt, als abgetastete Werte („Samples“) der Verteilung von Radiance in der Szene. Je Sample wird für die einfallende Beleuchtung α_p aus Richtung ω_p ein Tupel (ω_p, α_p) gespeichert.

Beim Rendering eines Bildes wird für jeden Pixel j ein Maß I_j berechnet. Mit der Importancefunktion $W_e(p,\omega)$, welche die einfallende Radiance für einen Bildpunkt je nach Richtung und (kontinuierlichem) Ort p_{Bild} gewichtet, ist I_j in [PH10b]

$$I_j = \int_{A_{Bild}} \int_{S^2} W_e(p_{Bild}, \omega) L_i(p,\omega) \,\mathrm{d}\omega \,\mathrm{d}A(p)$$

Dies wird in [Vea97] für ein beliebiges Maß verallgemeinert und es wurde gezeigt, dass

$$E\left[\frac{1}{N}\sum_{j=i}^{N} \alpha_j W_e(p_j, \omega_j)\right] = \int_A \int_{S^2} W_e(p,\omega) L_i(p,\omega) \,\mathrm{d}\omega \,\mathrm{d}A(p)$$

für ein gegebnes W_e gilt. Das bedeutet der Erwartungswert eines Maßes lässt sich direkt aus den Samples berechnen.

So kann man zeigen, dass auch das Maß der eingestreuten Radiance, L_r sich mit Hilfe der Samples ermitteln lässt. L_r wird dabei umformuliert in

$$L_r = \int_A \int_{S^2} \delta(p - p') f_s(\omega, p', \omega') \left|n_{p'} \cdot \omega'\right| L(p', \omega') \,\mathrm{d}\omega' \,\mathrm{d}A(p')$$

mit $W_e(p', \omega) = \delta(p - p') f_s(\omega, p', \omega') \left|n_{p'} \cdot \omega'\right|$. Beim Photon Mapping wird L_r nicht nur von den Samples genau an der Stelle p berechnet, da in der Regel nicht genügend vorhanden

sind. Deshalb approximiert man die Summe aus den $n < N$ nächsten Samples in einer Umgebung von p und nach [PH10b] kann man die Näherung

$$L_r(p,\omega) \approx \sum_{j=i}^{N} \hat{p}'(p_j)\alpha_j f_s(\omega, p, \omega_j) \tag{5.1.II}$$

ableiten. Der zusätzliche Faktor $\hat{p}'(p_j)$ filtert die Samples außerhalb der Umgebung heraus und gewichtet die innerhalb abhängig von der Entfernung zu p. Diese Art der Approximation ähnelt der aus Gleichung (5.1.I). Für den Strahlungsfluss $\Delta\Phi$ steht α und die Gewichtung ist nicht uniform sondern abhängig von der Entfernung zu p und mit N sind alle Samples gemeint.

Diese Form der Näherung mit $\hat{p}'$ ist aus einer Kerndichteschätzung nach [PH10b] entstanden. Dies ist eine Methode zur Schätzung einer Wahrscheinlichkeitsverteilung aus einer gegebenen Stichprobe an Punkten. In dieser Anwendung ist es die PDF von L_r aus der Stichprobe der Photonen in der Umgebung. Diese Art der Abschätzung hat den Vorteil, dass sie eine stetige Schätzung einer unbekannten Verteilung liefert. Hierfür wird eine Kernfunktion, $k(x)$ verwendet, die zu 1 integriert:

$$\int_{-\infty}^{\infty} k(x)\,\mathrm{d}x = 1$$

Der Kerndichteschätzer für N Samples an den Stellen x_i ist

$$\hat{p}(x) = \frac{1}{Nh}\sum_{i=1}^{N} k\left(\frac{x - x_i}{h}\right) \quad .$$

h bezeichnet die Bandbreite und bestimmt die Qualität der Schätzung. Um die Bandbreite erfolgversprechend zu wählen bedient man sich der Nearest-Neighbour-Schätzung, bei der die n nächsten Punkte zu x gewählt werden und für h die Distanz zum am weitesten entfernten Punkt x_m benutzt wird. Das ganze ergibt dann für d Dimensionen

$$\hat{p}(x) = \frac{1}{N\left|x - x_m\right|^d}\sum_{i=1}^{N} k\left(\frac{x - x_i}{|x - x_m|}\right) \quad .$$

Als Kernfunktion wird in [PH10b] der Simpson'sche Kern verwendet mit

$$k(x) = \begin{cases} \frac{3}{\pi}\left(1 - |x|^2\right)^2 & |x| < 1 \\ 0 & \text{sonst} \end{cases} \tag{5.1.III}$$

und $\hat{p}'$ hat die Form

$$\frac{1}{N\left|p - p_m\right|^d} \cdot k\left(\frac{p - p_i}{|p - p_m|}\right)$$

Hiermit werden die Photonen, die weiter von p entfernt sind geringer gewichtet wie die näher dran.

Diese Approximation von L_r hat Verwendung beim Verfahren für Oberflächen mit dem 2-dimensionalen Nearest-Neighbour-Schätzer und wird analog dazu bei der Erweiterung auf partizipierende Medien auch in 3 Dimensionen eingesetzt.

5.2 Separation der Renderinggleichung

Das Integral aus L_r auf kann man analog zu [Jen96] je nach Art von eingestreuter Radiance und dem dazugehörigen Teil der BSDF separieren. Die Aufteilung hier ist so gewählt wie in [PH10b], da der Photon Mapper aus pbrt als Grundlage für diese Arbeit genutzt wurde. Die BSDF ist aufgeteilt in

- $f_{s,\Delta}$, dem spekularen Anteil bei spiegelnden Material oder transluzentem Material
- $f_{s,\neg\Delta}$, dem nicht spekularen Anteil beispielsweise bei allem leicht spiegelnden oder diffus streuenden Material

und die eingestreute Radiance in

- L_d aus direkter Beleuchtung: Radiance, die nicht gestreut wurde
- L_i aus indirekter Beleuchtung: Radiance, die mindestens einmal diffus gestreut wurde
- L_c aus Kaustiken: Radiance, die durch spekularer Reflektion oder Lichtbrechung gestreut wurde

Daraus ergibt sich die Eingestreute Radiance als Summe

$$\begin{aligned} L_r(p,\omega) = &\int_{S^2} f_{s,\Delta}(\omega,p,\omega')L(p,\omega')\langle n_p,\omega'\rangle \,\mathrm{d}\omega' \\ &+ \int_{S^2} f_{s,\neg\Delta}(\omega,p,\omega')L_d(p,\omega')\langle n_p,\omega'\rangle \,\mathrm{d}\omega' \\ &+ \int_{S^2} f_{s,\neg\Delta}(\omega,p,\omega')L_c(p,\omega')\langle n_p,\omega'\rangle \,\mathrm{d}\omega' \\ &+ \int_{S^2} f_{s,\neg\Delta}(\omega,p,\omega')L_i(p,\omega')\langle n_p,\omega'\rangle \,\mathrm{d}\omega' \end{aligned} \tag{5.2.I}$$

mit

$$f_s = f_{s,\Delta} + f_{s,\neg\Delta} \quad \text{und} \quad L = L_d + L_c + L_i \quad .$$

In pbrt wird der erste Summand, die eingestreute Radiance bei perfekt spekularen Materialien (wie z.B. Spiegel oder Glas), mit rekursivem Ray Tracing berechnet. Der zweite Summand wird aus dem Standardverfahren zu direkter Beleuchtung ermittelt. Für die Kaustiken und die indirekte Beleuchtung werden im folgenden Abschnitt die Photon Maps benötigt.

5.3 Verfahren für Oberflächen

Dieses Verfahren benötigt zwei separate Pässe für das Rendering eines Bildes. Der erste Pass ist die Simulation der Wege von Photonen und im Zweiten Pass wird mit Hilfe der Photon Maps das Bild erzeugt. Bevor es an die Erweiterung auf partizipierende Medien geht sollen die nächsten zwei Abschnitte das Verfahren erläutern wie es unter anderem in pbrt implementiert ist.

5.3.1 Pass 1: Photonen verschießen

Mit den Lichtquellen als Ausgangspunkt werden die Photonen in die Szene verschossen und ihr Pfad simuliert. Beim Auftreffen auf eine Oberfläche werden sie dann entweder absorbiert oder an dieser gestreut und weiter verfolgt. Bei fast jeder Interaktion wird für dessen Ort p in einer Photon Map das Tupel (ω, α) gespeichert. ω zeigt in die Richtung, aus der das Partikel stammt, und α ist dessen Gewichtung.

Wird nun ein Pfad eines Partikels simuliert, so ist dessen Anfangsgewicht in Anlehnung an [Vea97] durch die abgestrahlte Radiance der jeweiligen Lichtquelle gegeben zu

$$\alpha_0 = \frac{L_e(p_0, \omega_0) \left| n_{p_0} \cdot \omega_0 \right|}{P(p_0, \omega_0)}$$

mit $\left| n_{p_0} \cdot \omega_0 \right|$, der Projektion mit der Normalen, $P(p_0, \omega_0)$, dem Produkt aus der Wahrscheinlichkeit eben diese Lichtquelle zu wählen und der Wahrscheinlichkeit des Partikels von p_0 aus in Richtung ω_0 abgestrahlt zu werden. Bei einer Interaktion wird die Gewichtung angepasst zu

$$\alpha_{j+1} = \alpha_j \frac{1}{q_{j+1}} \frac{f_s(\omega, p_j, \omega') \left| n_p \cdot \omega' \right|}{p(\omega')} \quad .$$

Die neue Richtung ω' entsteht durch Sampling der BSDF an der Stelle p. Der Pfad wird genauso wie es Path Tracing der Fall war mit russischen Roulette zufällig abgebrochen und so bezeichnet q_{j+1} die Wahrscheinlichkeit für die nächste Interaktion, also mit der Berechnung fortzufahren. Insgesammt hat die Gewichtung für einen Pfad der Länge n die Form von (4.3.III), den Beitrag eines Pfades aus dem Path Tracing.

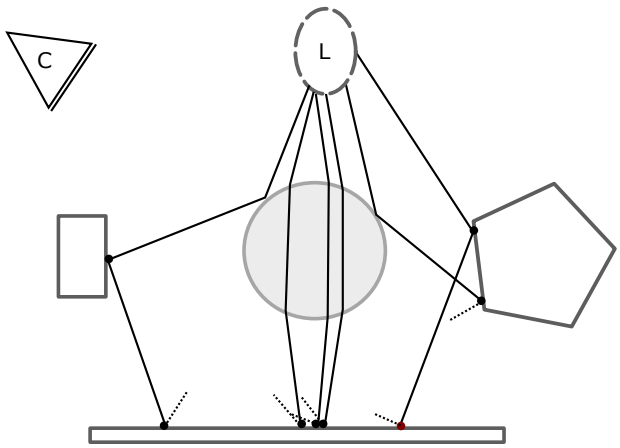

Abbildung 5.1: Der erste Pass: Hier werden die Photonen von den Lichtquellen aus in die Szene verschossen. Einträge in die jeweilige Photon Map werden nur an solchen Oberflächen angelegt, die nicht spiegelnd oder transparent sind.

Als Datenstruktur für die Photon Map wird in der Regel ein *k*d-Baum verwendet. Dieser wurde in [Ben75] eingeführt und ist ein unbalancierter Suchbaum für Punkte in k Dimensionen. Damit wird im weiteren Verlauf die Suche von Photonen in einer Umgebung beschleunigt.

Ausgehend von den Vorbemerkungen werden mehrere Photon Maps eingesetzt um die Interaktionen zu speichern. Dadurch verschafft man sich nicht nur viel kleinere Datenstrukturen und eine geringere Suchzeit, sondern kann vor allem die Dichte der Photonen für bestimmte Effekte beeinflussen. Werden Photonen verschossen, so treten in der Photon Map dort Häufungen auf, wo die Wahrscheinlichkeit für eine Interaktion mit der Oberfläche am höchsten ist. Kaustiken sind zwar auch Häufungen von Radiance, diese sind aber im Vergleich zur übrigen Beleuchtung sehr selten. Um diesen Effekt der Kaustiken zu unterstützen, können beispielsweise hochaufgelöste Photon Maps für deren Berechnung erzeugt werden indem eine Mindestanzahl der Photonen für die jeweilige Map vorgegeben wird.

Eine Mögliche Art der Aufteilung findet sich in der Implementierung von pbrt, bei welcher die Photonen auf 3 Photon Maps verteilt werden:

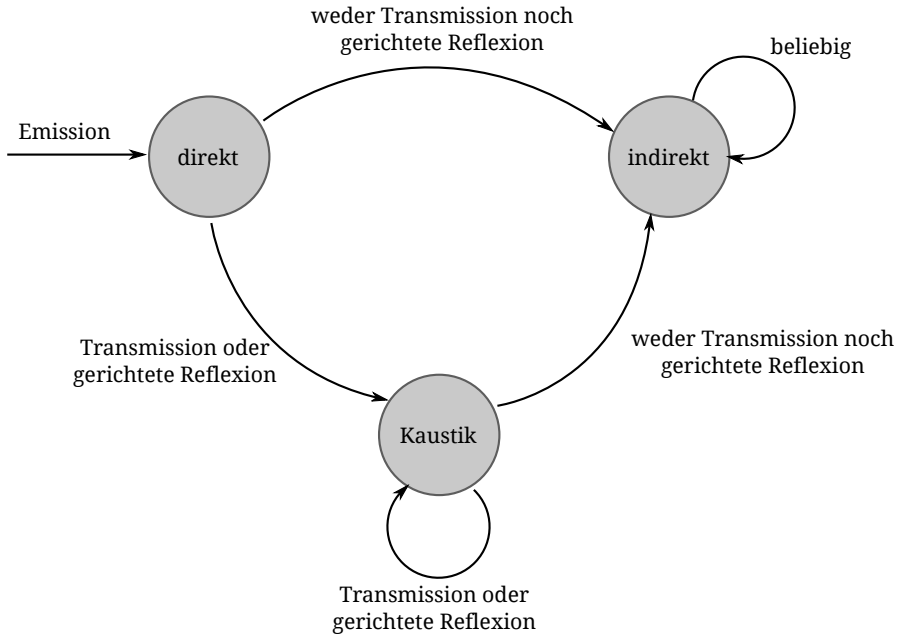

Abbildung 5.2: Ähnlich wie der Zustandsautomat aus [PH10b] soll dieser hier die Ablage der Daten nach einer entsprechenden Interaktion mit der Oberfläche in die jeweilige Photon Map darstellen. Die Zustände entsprechen den Photon Maps und zu beachten ist, dass je Zustand nur dann Photonen gespeichert werden, wenn das aktuelle Material weder spiegelnde noch transparente Anteile besitzt.

direkte Beleuchtung bei einer Oberflächeninteraktion direkt nach der Emission (Diese Map wird im späteren Verlauf auch für die Berechnung der indirekten Beleuchtung genutzt.)

indirekte Beleuchtung bei mindestens einer Oberflächeninteraktion

Kausiken bei mindestens einer Oberflächeninteraktion mit einem spekularen Material

Eine Übersicht welche Photon Map nach welcher Oberflächeninteraktion benutzt wird liefert der Zustandsautomat in Abbildung 5.2.

5.3.2 Pass 2: Rendering mit den Photon Maps

Sobald die Photonen verschossen sind, beginnt mit dem 2. Pass ist Berechnung des Bildes wie die eines rekursiven Ray Tracers. Es werden nun Strahlen von der Kamera aus in die Szene verschossen, die dann an einem Punkt p die Oberflächen schneiden. Für diesen Punkt wird eine Näherung an die Lösung Renderinggleichung mit Hilfe der Integrale aus (5.2.I) bestimmt.

Wie in den Vorbemerkungen ausgeführt, wird die direkte Beleuchtung mit dem Standardverfahren aus dem Ray Tracing ermittelt. Die Kaustiken werden direkt mit der jeweiligen Photon Map berechnet indem in einer Umgebung von p nach den nächsten n Photonen gesucht und mit der Gleichung (5.1.II) die resultierende Radiance bestimmt wird.

Die indirekte Beleuchtung kann nun auf zweierlei Weise bestimmt werden. Ebenso wie für die Kaustiken kann bei p die Photon Map direkt benutzt werden. Dies hat jedoch den Nachteil, dass für manche Szenen die Ergebnisse sehr unscharfe Stellen aufweisen. Aus diesem Grund bedient man sich des „Final Gatherings“. Bei diesem Schritt gibt es verschiedene Ansätze:

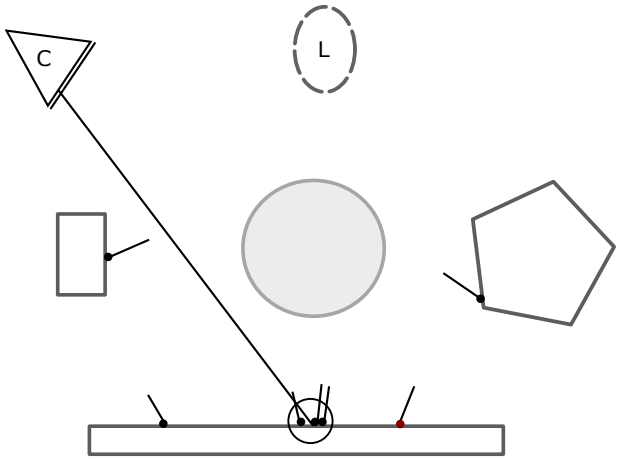

Abbildung 5.3: Der zweite Pass: Die Indirekte Beleuchtung wird hier mit den Photon Maps angenähert.

- Nach [Jeno1] können die Photonen als Lichtquellen angesehen und zusätzlich zu den Lichtquellen der Szene in die Berechnung der einfallenden Radiance in p miteinbezogen werden. Dies wäre dann vergleichbar mit einem Spezialfall des Bidirektional Path Tracing [LW93], bei dem die Lichtpfade die Länge n und die Kamerapfade die Länge 1 haben.
- In der Implementierung von pbrt und in [PH10b] von p aus zusätzlich Strahlen gemäß der BSDF an dieser Stelle verschossen und an den jeweiligen neuen Schnittpunkten p' wird nun die abgestrahlte Radiance mit der Hilfe jeweils einem vorausberechneten „Radiance Photon“ ermittelt. Im ersten Pass wird nun zusätzlich für einen Teil der Photonen die abgestrahlte Radiance vorberechnet und in einer Radiance Photon Map abgelegt. Aus dieser wird im zweiten Pass das am nächsten gelegene Radiance Photon gesucht und für die indirekte Beleuchtung von p' aus verwendet.

Wenn das Material bei p spekular (spiegelnd oder transparent) ist, wird diese Berechung schließlich rekursiv wiederholt.

5.4 Erweiterung auf partizipierende Medien

Die Erweiterung auf partzipierende Medien basiert auf der Aberbeit von H. W. Jensen und P. H. Christensen aus [JC98]. In diesem Paper ist die Transfergleichung aus [Sie92] die Grundlage. Sie unterscheidet sich von der Gleichung (3.6.I) dadurch, dass die Emittierte Radiance zusätzlich mit σ_a, dem Absorptionskoeffizient, multipliziert ist. Um mit dem Path Tracing konsistent zu bleiben, wird im Folgenden Gleichung (3.6.I) als Ausgangspunkt verwendet und an den jeweiligen Stellen aus [JC98] das σ_a weggelassen.

Die Idee für die Erweiterung ist es Photonen nicht nur auf den Oberflächen abzulegen sondern auch in einem partizipierenden Medium.

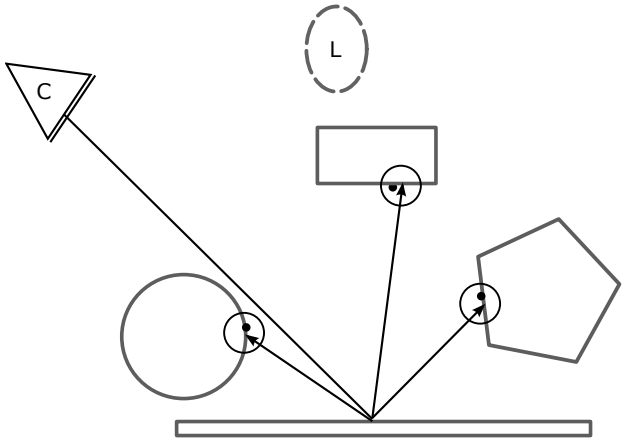

Abbildung 5.4: Näherung der Indirekten Beleuchtung mit final Gathering

In [JC98] wird eine Volume Photon Map eingeführt. Interagiert ein Lichtpartikel entlang eines Strahles mit einem Medium, so wird es entweder absorbiert oder gestreut. Für jedes dieser Ereignisse wird in diesem Verfahren ein Eintrag in die Volume Photon Map angelegt. Dies ist wieder ein Tupel aus der Gewichtung α und der negativen Strahlrichtung ω. Hier werden die Photonen innerhalb des Mediums abgelegt. Dies hat nach [JC98] folgende Vorteile:

- Die globale Beleuchtung des Mediums ist unabhängig von der Kameraposition. Legt man die Photonen nur auf den Rand ab so kann zwar von außerhalb des Mediums die globale Beleuchtung berechnen. Verschiebt man die Kamera jedoch ins Innere, so gelingt dies nicht mehr. Außerdem wären so die Berechnung von Kaustiken im Medium nicht möglich.
- Die Berechnungen sind unabhängig von der Art des Mediums. Es können ohne Einschränkung voxelisierte oder prozedural erzeugte Medien verwendet werden.
- Im Gegensatz zu Verfahren, die das Medium diskretisieren und je Volumenelement die Radiance berechnen, ergeben sich bei dieser Methode keine aliasing Probleme.

Der Strahlungstransport aus [Cha50] ist definiert durch Gleichung (3.6.I) und ist in der Integralform aus den Grundlagen in Kapitel 3:

$$L(x,\omega) = \int_{x_A}^{x} \tau(x',x)\left[L_{e,V}(x',\omega) + \sigma_s(x')L_i(x',\omega)\right]\,\mathrm{d}x' + \tau(x_A,x)L(x_A,\omega) \tag{5.4.I}$$

mit

$$L_i(x,\omega) = \int_{S^2} f_p(\omega,x,\omega')L(x,\omega')\,\mathrm{d}\omega' \tag{5.4.II}$$

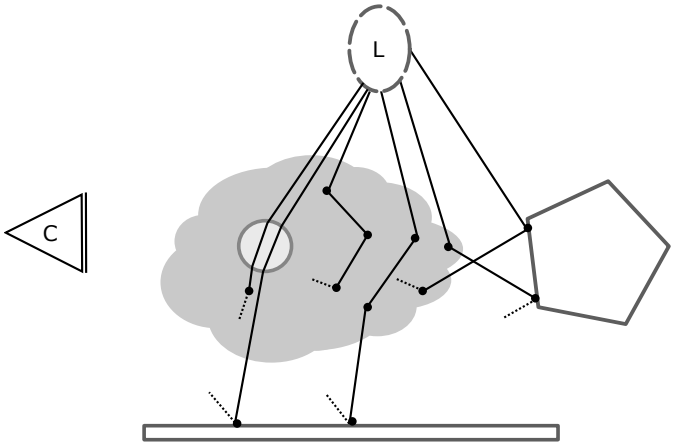

Abbildung 5.5: In dieser Erweiterung werden im ersten Pass Photonen am Medium gestreut oder davon absorbiert.

In diesem Verfahren werden erneut Photonen verwendet um die eingestreute Radiance zu berechnen. Diese werden dann analog zum Verfahren für Oberflächen in einem Vorverarbeitungsschritt verschossen und in die Volume Photon Map abgelegt. Die Wahrscheinlichkeit, dass ein Streuereignis stattfindet ist gegeben durch die Streualbedo, definiert als der Quotient

$$\Lambda(x,\omega) = \frac{\sigma_s(x,\omega)}{\sigma_t(x,\omega)} \quad .$$

Basierend auf Λ kann nun bei einem Streuereignis ein neues Photon erzeugt werden, dessen Gewicht mit Λ skaliert wird. Dies ist jedoch nach [Jeno1] nicht ganz zielführend, denn so entstünden viele Einträge in der Photon Map mit sehr kleinen Gewichten. Besser sei es da mit russischem Roulette zu entscheiden ob ein Photon absorbiert oder gestreut wird und die Skalierung weg zu lassen. Die neue Richtung bei einem Streuereignis wird durch Importance Sampling der Phasenfunktion entschieden.

Ebenso wie im Fall der direkten Beleuchtung für Oberflächen ist es für die Berechnung der Einfachstreuung nicht nötig Photon Maps zu benutzen. Deshalb werden in dieser Erweiterung nur Spuren derjenigen Partikel gespeichert, die mindestens einmal gestreut wurden. Eine mögliche Erweiterung des Zustandautomaten aus Abbildung 5.2 für die Verteilung auf die verschiedenen Photon Maps ist in Abbildung 5.7 zu sehen und die eingeführten Photon Maps werden im Abschnitt 5.5 genauer erläutert.

Bei der Einfachstreuung wird nur die Radiance betrachtet, die direkt von den Lichtquellen her eingestreut wird. Wird dieser Anteil abgespalten, lässt sich Gleichung (5.4.II) umformulieren in

$$L_i(x,\omega) = \underbrace{\int_{S^2} f_p(\omega,x,\omega')\tau(x_{L_e},x)L(x_{L_e},\omega')\,\mathrm{d}\omega'}_{=L_{i,d}} + \underbrace{\int_{S^2} f_p(\omega,x,\omega')L(x,\omega')\,\mathrm{d}\omega'}_{=L_{i,i}} \quad .$$

$L_{i,d}$ kann, wie in [PH10b] beschrieben, durch Abtasten der Lichtquellen berechnet werden und $L_{i,i}$ beschreibt den Anteil der Mehrfachstreuung, dieser wird als einziger im Weiteren Verlauf betrachtet.

In [JC98] und [Jeno1] sind der Streukoeffizient und der Absorptionskoeffizient (und die daraus resultierenden Größen) nicht richtungsabhängig. In pbrt und in [PH10b] sind beide jedoch sehr allgemein gehalten. Um mit dem Design von pbrt konform zu bleiben, wird im weiteren Verlauf auf die Richtungsabhängigkeit nicht verzichtet und alle Formulierungen sind entsprechend angepasst.

Wie in [JC98] wird der Zusammenhang von Radiance in partizipierenden Medien und dem gestreuten Fluss Φ aus [Sie92] benutzt:

$$L(x,\omega) = \frac{\mathrm{d}^2\Phi(x,\omega)}{\sigma_s(x,\omega)\mathrm{d}\omega\mathrm{d}V}$$

Setzt man das nun für die einfallende Radiance aus der Mehrfachstreuung ein, erhält man

$$L_{i,i}(x,\omega) = \int_{S^2} f_p(\omega,x,\omega')\frac{\mathrm{d}^2\Phi(x,\omega')}{\sigma_s(x,\omega')\mathrm{d}\omega'\mathrm{d}V}\,\mathrm{d}\omega'$$

Wird $L_{i,i}$ analog zum Verfahren für die Oberflächen durch eine Photon Map angenähert und sind n Photonen in der Umgebung von x gefunden, ist Approximation gegeben durch

$$L_{i,i}(x,\omega) \approx \sum_{j=1}^{n} f_p(\omega, x, \omega_j) \frac{\Delta\Phi_i(x,\omega_j)}{\sigma_s(x,\omega_j)\Delta V} \tag{5.4.III}$$

mit $\Delta V = \frac{4}{3}\pi r^3$ und r, den Abstand zum am weitesten entfernten Photon der Umgebung.

Ebenso analog zum Basis-Verfahren könnte man hier nicht über den Fluss Φ argumentieren sondern einen allgemeineren Ansatz aufbauend auf [Vea97] verfolgen. Wurden $n < N$ Samples in der Umgebung eines Oberflächenpunktes p gefunden, so konnte die Radiance L_r mit Hilfe eines Kerndichteschätzers aus den Gewichten α_j approximiert werden. Ähnliches soll nun unter der Annahme, dass Lichtpartikel ebenso an Teilchen eines Mediums gestreut werden für $L_{i,i}$ geschehen.

Die Umformulierung mit einem Kerndichteschätzer entspricht im Grunde einer anderen Gewichtung der Samples, der Einträge in der Volumen Photon Map. Hier wird nun die PDF der Radiance $L_{i,i}$ aus dem umgebenden Medium abgeschätzt. Daher lässt sich (5.4.III) umformulieren zu

$$L_{i,i}(p,\omega) \approx \sum_{j=1}^{n} \hat{p}'(p_j) f_p(\omega, p, \omega_j) \frac{\alpha_j}{\sigma_s(p,\omega_j)}$$

mit

$$\hat{p}'(p_j) = \frac{1}{N\,|p-p_m|^d} \cdot k\left(\frac{p-p_j}{|p-p_m|}\right) \quad .$$

Da die Radiance in drei Dimensionen abgeschätzt wird, ist $d = 3$. Hier ist p_m wieder der Sample mit der größten Distanz zu p, k ist die Kernfunktion aus (5.1.III) und N bezeichnet die Gesamtzahl Samples im Medium.

Aus dem Ergebnis von [JC98] für die gesamte gestreute Radiance von x in Richtung ω, ergeben sich mit der Einführung der Richtungabhängigkeit;

$$L_i(x,\omega) = L_{i,d}(x,\omega) + L_{i,i}(x,\omega)$$

und für $L_{i,i}$

$$L_{i,i}(p,\omega) \approx \sum_{j=1}^{n} \hat{p}'(p_j) f_p(\omega, p, \omega_j) \frac{\alpha_j}{\sigma_s(p,\omega_j)} \cdot \frac{\sigma_s(p,\omega_j)}{\sigma_t(p,\omega_j)} \quad .$$

Da bei jeder Art der Interaktion mit dem Medium ein Eintrag in der Photon Map erstellt wird muss nach [JC98] $L_{i,i}$ mit der Streualbedo gewichtet werden. Dieser ist in dieser Formulierung aber abhängig von der Richtung der gewählten Photonen und muss deshalb unter der Summe stehen.

Das Integral von x_A bis x aus Gleichung (5.4.I) wird mit dem Ray-Marching-Verfahren berechnet. Dieses unterteilt den Integrationsbereich entlang ω in finite Teilstücke. Für jedes

dieser Teile wird nun angenommen, dass sich die Radiance nicht verändert. Das Verfahren kann rekursiv formuliert werden und ist für dieses Integral im k-ten Teilstück

$$L(p_k, \omega) = L_{e,V}(p_k, \omega)\Delta p_k + \sigma_s(x_k, \omega)L_i(p_k, \omega)\Delta p_k + e^{-\sigma_t(p_k,\omega)\Delta p_k} L(p_{k-1}, \omega)$$

mit $\Delta p_k = |p_k - p_{k-1}|$ und p_0 ist der Schnittpunkt mit der Oberfläche im Medium oder der Rand des Mediums.

5.5 Implementierung

Für die Implementierung dieses Verfahrens wurden - genauso wie beim Path Tracing - viele Teile des besthenden Quelltextes aus pbrt wiederverwendet. Um die bestehende Implementierung nicht zu ersetzen, wurden in einem eigenen Namensraum (`extensions`) die Klasse für die Berechnung der Oberflächenbeleuchtung, `PhotonIntegrator`, übernommen und eine für die Beleuchtung der Volumendaten, `PhotonVolumeIntegrator`, ergänzt. Da beim verschießen der Photonen Samples für beide `Integrator`-Klassen erzeugt werden, wurde die entsprechende Funktionalität in eine gemeinsam verwendbare `PhotonMap`-Klasse ausgelagert. Diese übernimmt nun das Füllen der Photon Maps und die Vorausberechnungsschritte für das Final Gathering. Die `PhotonIntegrator`-Klasse der Erweiterung wurde sonst nicht weiter angetastet.

Es ist möglich die beiden `Integrator`-Klassen unabhängig voneinander einzusetzen, da beide, sofern noch nicht geschehen, die Berechnung der Photon Maps anstoßen. So kann die `PhotonVolumeIntegrator`-Klasse für Szenen verwendet werden, die fast vollständig aus Volumendatensätzen oder nur aus spiegelnde bzw. transparenten Oberflächen aufgebaut sind.

In der `PhotonVolumeIntegrator`-Klasse wurde in das Verfahren aus Abschnitt 5.4 implementiert. Das wichtigste Eingabedatum ist ein Strahl von der Kamera aus, für den die

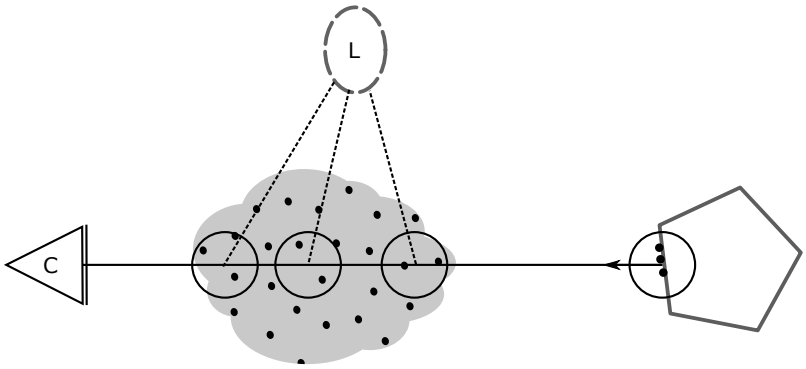

Abbildung 5.6: Die Radiance entlang eines Strahles durch ein Medium ist mit Ray Marching angenähert.

eingestreute Radiance berechnet werden soll. Diese geschieht durch ein leicht modifiziertes Ray Marching: Die Länge jedes Teilstück wird zufällig so variiert, dass sie bei einer festen Schrittgröße l zwischen $l/2$ und $l + l/2$ betragen kann. Zusätzlich wird in jedem Schritt k das Zwischenergebnis für L mit dem vorherigen verglichen. Hat sich die ermittelte Radiance verdoppelt, so wird die Schrittlänge halbiert und der k-te Schritt erneut berechnet. Dadurch soll nach möglichkeit verhindert werden, dass kleinere Details herausgemittelt werden. Im Gegensatz zu [JC98] geschieht die Berechnung nicht rekursiv. Es wird zwar das Aktuelle Teilstück erneut berechnet, jedoch wird die Schrittlänge zufällig (mitbestimmt durch das Verhältnis der aktuellen und der anfangs vorgegebenen Länge) wieder verdoppelt. Dadurch erspart man sich den Speicherverbrauch im Aufrufstack und die Ergebnisse werden dadurch nicht beeinträchtigt.

Das Verfahren aus [JC98] sieht eine Photon Map für die Samples aus den Interaktionen mit dem Medium vor, die „Volume Photon Map“. Um die Intensität von Kaustiken im Medium zu erhöhen wird in dieser Implementierung (ähnlich zum Verfahren für Oberflächen) eine „Volume Caustic Photon Map“ eingeführt in der alle Samples gespeichert werden, die nach einer Transmission oder gerichteten Reflektion durch eine Fläche mit dem Medium interagieren. Wird hier eine Mindestanzahl an Einträgen vorgegeben, steigt dadurch die Rechenzeit des beim Verschießen, doch die berechneten Kaustiken werden dadurch deutlich schärfer. Die Ergebnisse aus den Berechnungen sind im Anschluß an das Kapitel über Perlin Noise zu finden.

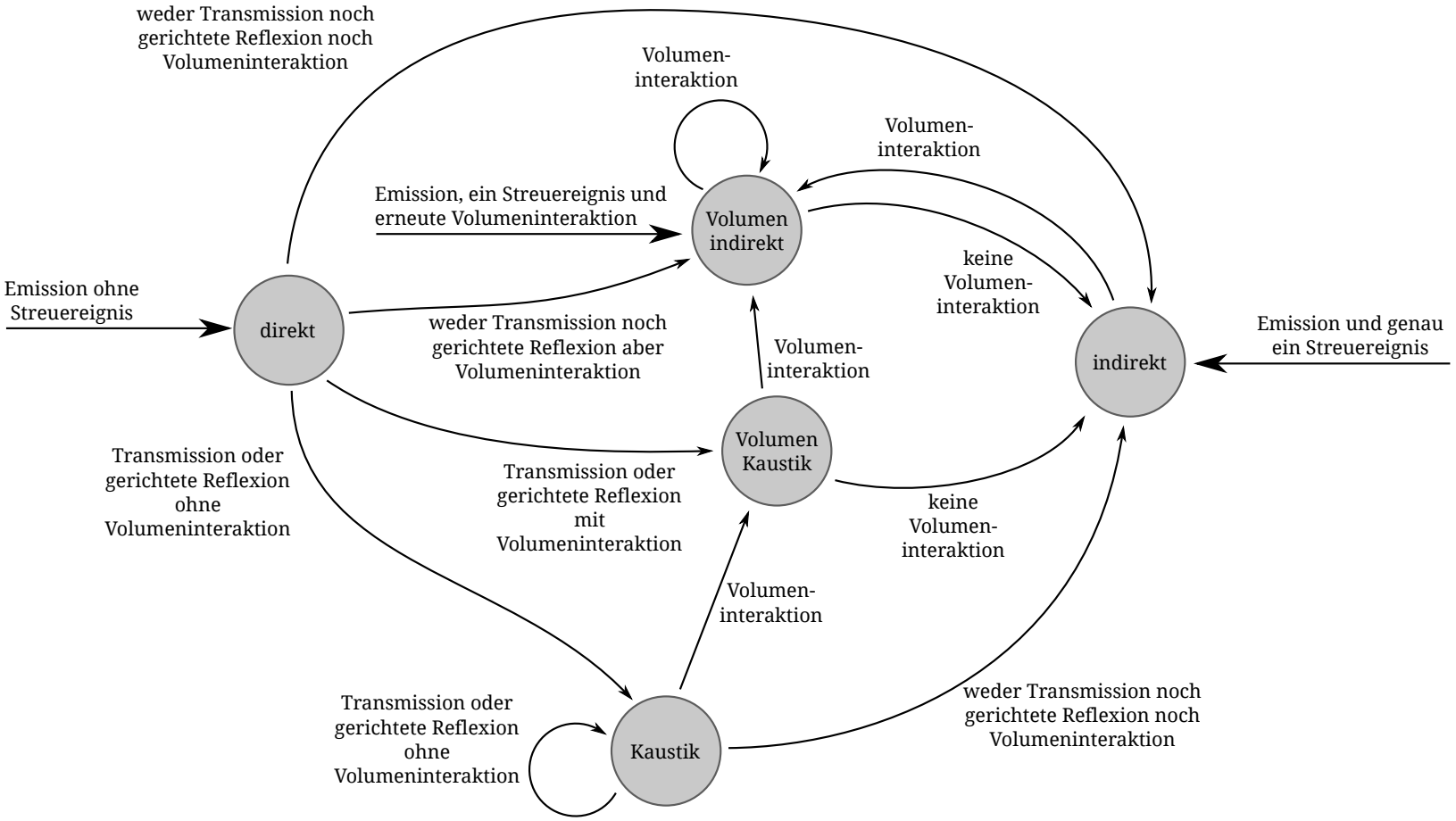

Abbildung 5.7: Eine erweitere Variante des Zustandsautomaten aus Abbildung 5.2 um eine Photon Map für die indirekte Einstreuung von Radiance in einem Medium (Volumen indirekt) und eine weitere für „Volumenkaustiken“ (Volumen Kaustik).

6 Perlin Noise

Mit der noise-Funktion, hat Ken Perlin 1985 in [Per85], eine vielseitiges Werkzeug eingeführt. Diese Funktion liefert für einen N-Dimensionalen Punkt einen Zufallswert, wobei meistens $N \leq 3$ gewählt wird. Werden noise-Funktionen miteinander kombiniert, so können damit z.B. Texturen mit den verschiedensten zufälligen Strukturen synthetisiert werden. Diese können dann als Höhenkarte für zufällig generierte Landschaften, für Wasser, Felsgestein und viele andere Anwendungen genutzt werden.

In dieser Arbeit wurde Perlin Noise dazu verwendet, um in niedrig aufgelösten voxelisierten Dichteinformationen mehr Details zu generieren. Darüber hinaus wurde auch ein Klasse aus prozedural erzeugbaren Dichtedaten implementiert, um hochaufgelöste inhomogene Medien zu erzeugen.

6.1 Die noise-Funktion

noise soll Zufallsfunktion sein, die von $\mathbb{R}^N$ nach $\mathbb{R}$ abbildet. Diese ist wiederum gestimmt durch eine diskrete Folge von zufälligen Werten. Für die Anwendungen hier wird $N = 3$ für den weiteren Verlauf vorgegeben und entspricht der Implementierung in pbrt.

Vorgaben an die noise-Funktion sind nach [Per85]:

- Sie muss statstisch invariant zu Rotationen und Translationen sein, also das gleiche statistische Verhalten für eine beliebeige Rotation oder Translation des Koordinatensystems aufweisen.
- Sie muss bandbegrenzt sein mit einer maximalen Frequenz, meist 1.

Die vorgegebenen zufälligen Werte $g(x, y, z)$ sind aus dem Intervall $[-1, 1]$. In [Per85] sind diese Werte der Gradient an den diskreten Punkten für eine vorgegebene Richtung. Im Gegensatz zu „Value Noise“ bei dem die Werte von g denen der noise-Funktion an den ganzzahlingen Stellen entsprechen, spricht man hier von „Gradient Noise“. An diesen Stellen, $p \in \mathbb{Z}^3$, ist außerdem der Wert der noise-Funktion mit 0 vorgegeben:

$$noise(p) = 0 \quad \text{für} \quad \text{und} \quad noise'(p) = g(p_x, p_y, p_z)$$

Um die noise-Funktion schließlich zu erhalten, werden die Gradienten an den jeweiligen Eckenpunkten diskreten Gitters interpoliert. Diese geben dann das Verhalten des noise-Funktion innerhalb einer Gitterzelle vor.

6.2 Kombination von Rauschfunktionen

Durch die Eigenschaft der Bandbegrenzung können höherfrequente noise-Funktionen durch skalierung der Koordinaten erzeugt werden. So liefert beispielsweise $noise(2 \cdot p)$ eine Funktion mit doppelt so hoher Frequenz wie $noise(p)$. Mit dieser Eigenschaft können prozedurale Texturen aus einer Summe von noise-Funktionen unterschiedlicher Frequenz und Gewichtung generiert werden. Typischerweise werden die Frequenzen mit jedem summanden verdoppelt und die Gewichte halbiert, sodass sich eine zusammengesetzte Funktion f_s ergibt mit

$$f_s(x) = \sum_i w_i noise(s_i \cdot x) \quad ,$$

$s_i = 2s_{i-1}$ und $w_i = w_{i-1}/2$. Je höher die Frequenz desto niedriger ist der Beitrag des Summanden bzw. der Oktave, da die Frequenz jeweils verdoppelt wird.

Nach [PH10b] wird diese zusammengesetzte Funktion oft als „Gebrochene Brownsche Bewegung“ (engl.: fractional Borwnian motion - kurz: FBm) genannt. Diese ist in pbrt für die Synthetisierung von Texturen implementiert. In pbrt geben Parameter die Anzahl der Oktaven und den Skalierungsfaktor der Gewichte w_i an.

Eine Variante davon wurde in [Per85] mit der „Turbulence“ eingeführt. Bei dieser wird von jedem Summand der Betrag addiert:

$$f_{s,turbulence}(x) = \sum_i w_i \,|(|\, noise(s_i \cdot x))$$

Diese Funktion besitzt zwar stellenweise Unstetigkeiten und die damit synthetisierten Texturen wirken rauer als die durch die FBm-Funktion erzeugten. Die Parameter entsprechen denen der FBm-Funktion.

6.3 Prozedurale Medien und Verfeinerung von Volumendatensätzen

Mit der Turbulence- und der FBm-Funktion lassen sich nun die Dichtedaten eines voxelisierten Datensatzes anpassen oder eigenständige Funktionen generieren, um prozedurale Medien zu erstellen.

In der Implementiereung dieser Arbeit wurde an der entsprechden Stelle in pbrt zu der interpolierten Dichte $D(p)$ eine entsprechend parametrisierte FBm-Funktion gewichtet mit der Dichte dazu addiert. Zusätzlich zu den oben genannten Paramertern wurde hier ein zusätzliches Gewicht für das Resultat, eine Skalierung und Verschiebung des Koordinatensystems angefügt. Die Ergebnisse, welche im nächsten Kapitel zu sehen sind, fielen sehr unterschiedlich aus und teilweise sind die Verbesserungen sehr subtil.

In [JC98] wurden auch Ergebnisse vorgestellt, bei denen das partizipierende Medium während der Berechnungen prozedural generiert wird. Ein erster Versuch ist auch im Rahmen

dieser Studienarbeit gelungen, bei dem die Turbulence- Funktion direkt als Maß für die Dichte des Mediums benutzt wurde. Auch hier wird auf die Ergebnisse im nächsten Kapitel verwiesen.

7 Ergebnisse

Dieses Kapitel soll die Ergebnisse aus den Berechnungen mit den vorgestellten Verfahren präsentieren. Darüber hinaus werden hier auch das Photon Mapping (incl. final gathering) mit dem Path Tracing verglichen. Path Tracing gilt hier als das Referenzverfahren. Jedoch dauert die Erzeugung akkurater Bilder bei komplexeren Szenen sehr lange, sodass hier einige kleinere Beispielszenen verwendet wurden. Als Rechenmaschine diente ein PC mit einem AMD Phenom II 1055T 2.8GHz als CPU und 4GB Arbeitsspeicher.

7.1 Homogene Medien

Bei dem linken Bild aus Abbildung 7.1 steht die Kaustik, erzeugt durch die abgebildete Glaskugel, im Vordergrund. Das Medium streut durch einen Anisotropieparameter von $g = -0.7$ in der Henyey-Greenstein Phasenfunktion nach hinten. Diese Szene ist ein Beispiel dafür den `PhotonVolumeIntegrator` ohne den dazugehörigen `PhotonIntegrator` für Oberflächen einzusetzen, da es hier nur die Glaskugel gibt, auf der keine Photonen verteilt werden können. Um den Effekt der Kaustik zu Stärken wurden für diese Szene 2.5 Millionen

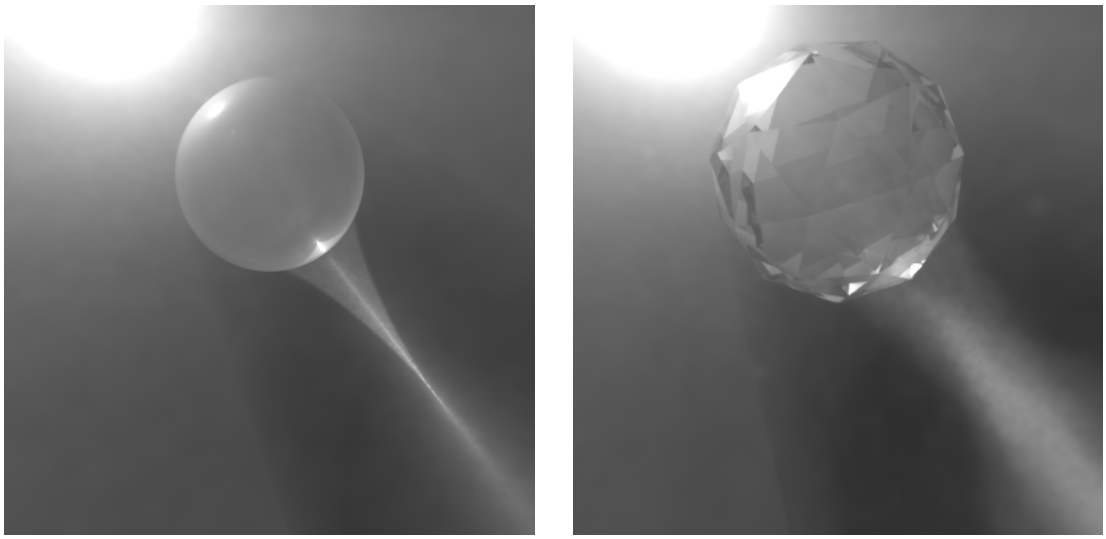

Abbildung 7.1: Volumenkaustik mit Photon Mapping - Kugel und regulärer Polyeder

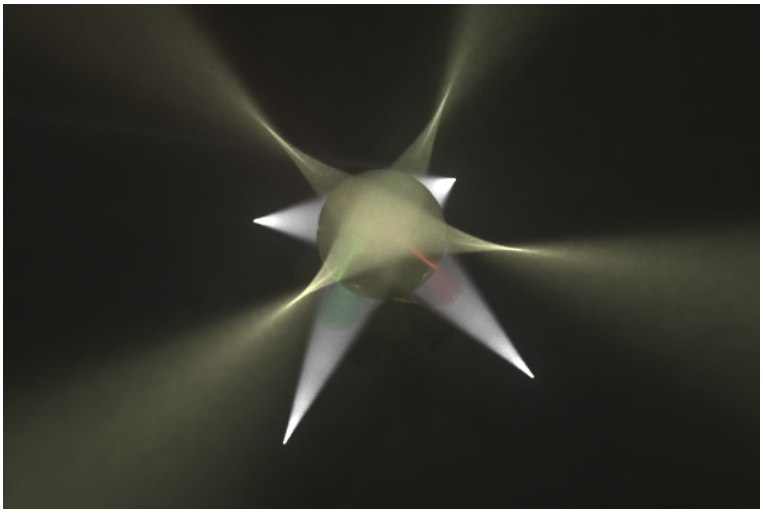

Abbildung 7.2: Mehrfache Volumenkaustik mit Photon Mapping

Photonen verschossen und je 200 Stück für die Beleuchtung verwendet. Das Erzeugen des Bildes dauerte 5min und die Photonen wurden in 50s verschossen.

Eine Variation davonist das Rechte Bild aus Abbildung 7.1. Hier wurde die gleiche Anzahl Photonen verwendet und bei gleicher Rechenzeit steht anstelle der Kugel ein gläserner Polyeder. Dieser erzeugt keine so scharf gebündelte Kaustik wie die Kugel, soll aber dennoch ein etwas komplexeres Model zeigen.

Als Abschluß der Szenen mit homogenem Medium soll die Abbildung 7.2 sein. In dieser Szene sind auf einem runden Podest drei verschiedenfarbige Kugeln und darauf eine weitere Kugel aus gelben Glas, welche von vier Strahlern beleuchtet wird. Hier sollen vor allem die scharfen, gelben Kaustiken hervorgehoben sein. Dieses Bild wurde mit 3 Millionen Photonen berechnet, die vor allem auf die Kaustiken verteilt wurden.

7.2 Inhomogene Medien

Als inhomogenes Medium wurde in erste Linie in Anlehnung an die Beispielszene in pbrt ein Datensatz von Duc Nguyen und Ron Fedkiw benutzt.

In der Abbildung 7.3 ist je Bild der Datensatz in einer einfachen Szene mit einer grauem Fläche als Untergrund und eine hellblauen Fläche als Hintergrund zu sehen. Das Linke Bild entstand mit dem Photon Mapping. Hier wurden 40000 Photonen verwendet für die indirekte Oberflächenbeleuchtung und 60000 für die Volumenbeleuchtung in der Szene verteilt. Jeweils 100 Partikel wurden für die Beleuchtung verwendet. Die Laufzeit hier betrug 83s für den ersten Pass und 160s für den zweiten. Das Path Tracing wurde mit 1024 Abtastungen je Pixel aufgerufen und hat nach 14min ein sehr ähnliches Bild erzeugt. Die Unterschiede sind in der Regel nicht wahrzunehmen.

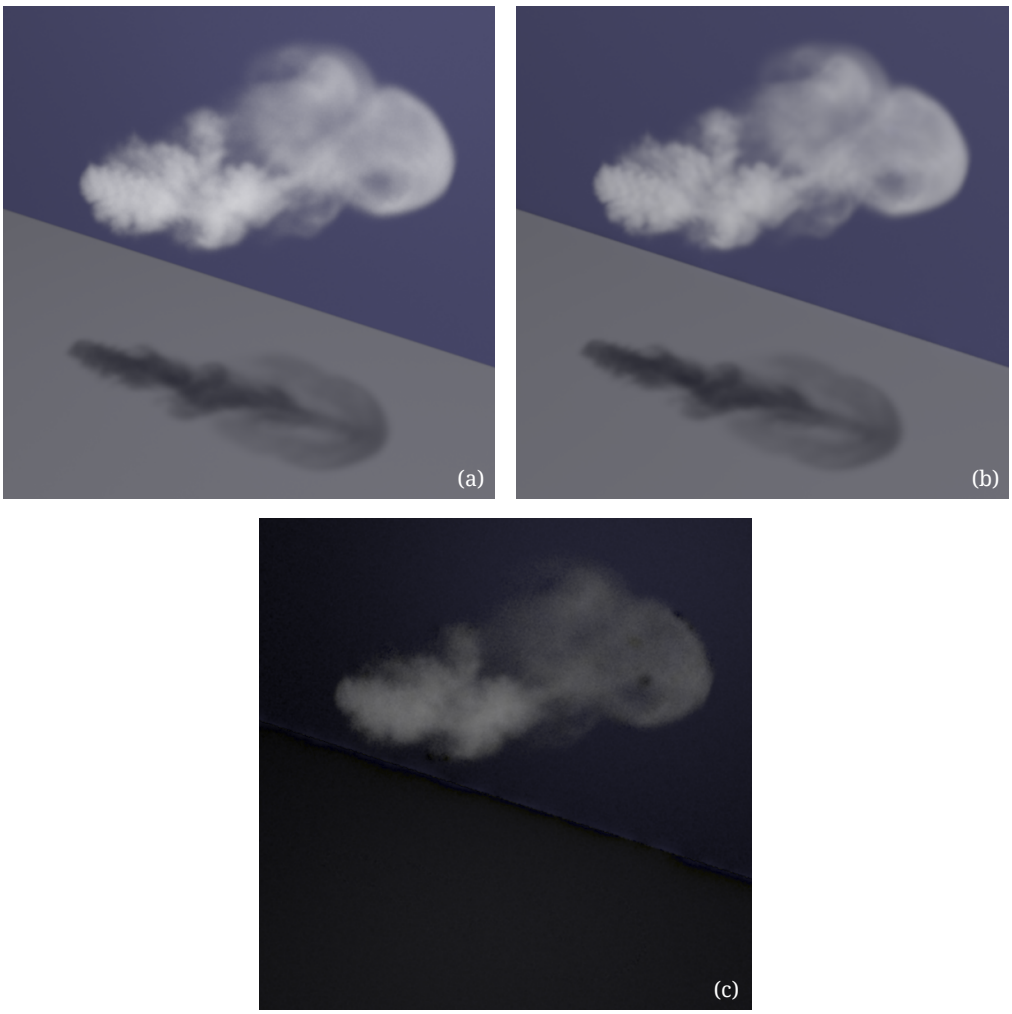

Abbildung 7.3: (a) Path Tracing; (b) Photon Mapping; (c) Differenz

Das Differenzbild zeigt jedoch wie stark ausgeprägt diese sein können. Das Medium in dieser Szene emittiert Radiance und da dieser Effekt nur schlecht von dem Photon Mapping berücksichtigt wird ist das Bild etwas dunkler. Darüber hinaus sind einige zusätzliche Details im Medium entstanden oder wurden heraus gemittelt, die nicht den Dichtedaten entsprechen.

Als nächstes wird der Effekt der emittierten Radiance eines Medium und der Verfeinerung durch Perlin Noise in Abbildung 7.4 gezeigt. Hier wurden bei dem Path Tracing 4096 Strahlen je Pixel ausgesandt und die Rechenzeit betrug etwa 2 Stunden. In dieser Szene gibt es außer dem Medium keine andere Lichtquelle, wodurch ein Rendering mit dem Photon Mapping ein beinahe schwarzes Bild liefern würde.

Das Bild wurde einmal ohne und einmal mit einer Verfeinerung durch Perlin Noise berechnet. Durch die noise-Funktion treten an manchen Stellen Details hervor und an anderen ver-

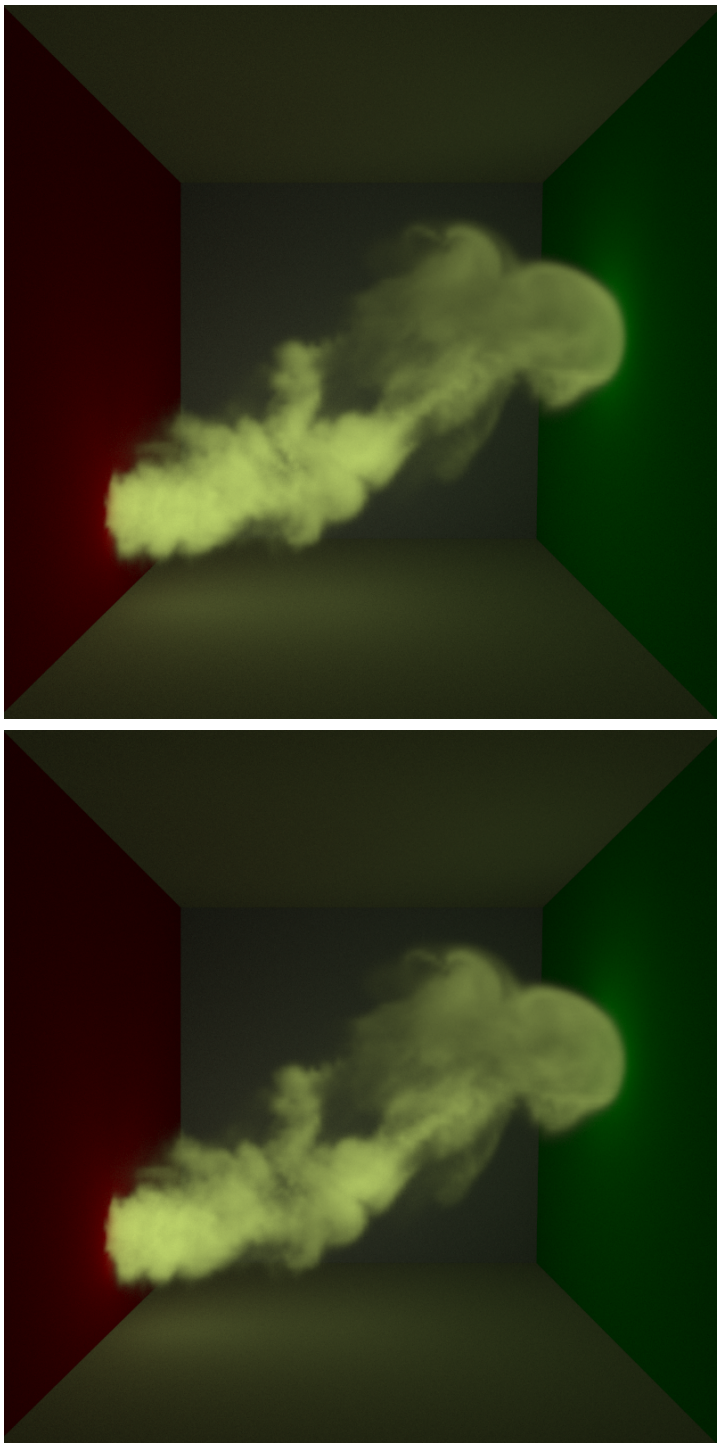

Abbildung 7.4: Oben: ohne Perlin Noise - Unten: mit Perlin Noise

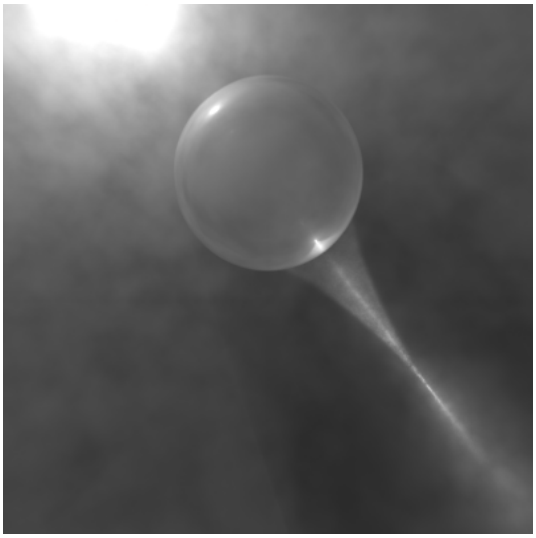

Abbildung 7.5: inhomogenes Medium mit Turbulence-Funktion prozedural erzeugt

schwinden sie. Die Wahl der Parameter ist hier entscheidend von dem Dichtedaten abhängig. Jedoch zeigt das Ergebnis an vielen Stellen die gewünschte Verbesserung.

Abbildung 7.5 zeigt eine ähnliche Szene wie aus Abbildung 7.1. Hier wurde das homogene Medium durch ein prozedural erzeugtes, inhomogenes Medium ersetzt. Das Bild wurde mit dem Photon Mapping unter denselben Bedingungen erzeugt, dauerte jedoch mit 10min für den ersten Pass und 30min für den zweiten deutlich länger.

Schließlich wurden in Abbildung 7.6 ein homogenes und ein inhomogenes Medium miteinander kombiniert. Das Bild, das mit Path Tracing erzeugt wurde (links), benötigte bei 16384 Pixelsamples etwa 20 Stunden Berechungszeit. Im Vergleich dazu das Bild, das mit Photon Mapping generiert wurde (rechts), benötigte 3min für das Verteilen der Photonen und 2 Stunden für die Berechmung des Bildes. Hier zeigt sich das sehr langsame Konvergenzverhalten des Path Tracing. Im Gegensatz dazu ist das Photon Mapping mit deutlich Abstrichen jedoch schneller. Auffällige Unterschiede sind der fehlende Schatten der Lichtquelle, die blaue Reflektion unterhalb spiegelnden Kugel und die allgemeine Helligkeit.

7.3 Verschiedenes

Nun sind auf den Bildern der Abbildungen 7.7 und 7.8 komplexere Szenen zu sehen. Beide sind an die Beispiele aus pbrt angelehnt. Die Landschaftsszene („plants“) mit leichter Tiefenunschärfe und homogenem Medium wurde mit 1024 Strahlen je Pixel in 3 Stunden generiert. Die Sponza-Szene wurde mit 15 Millionen Photonen berechnet, jedoch dauerte es nur 1 Stunde 40 Minuten. Die Dichte des Mediums hier nimmt exponentiell mit der Höhe ab.

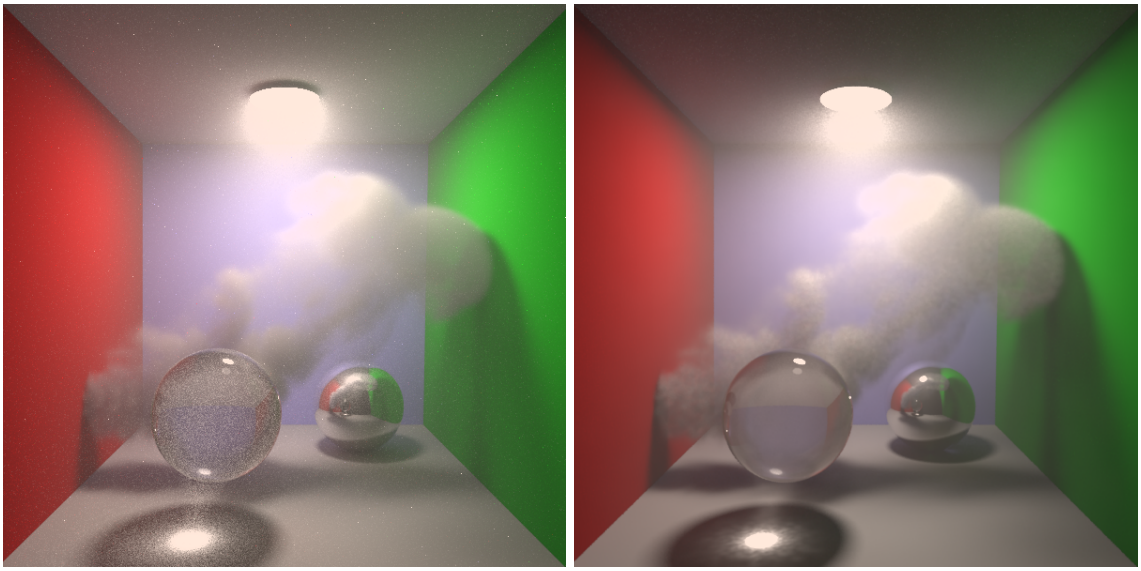

Abbildung 7.6: Inhomogener Volumendatensatz innerhalb eines homogenen Mediums - links: Path Tracing; rechts: Photon Mapping

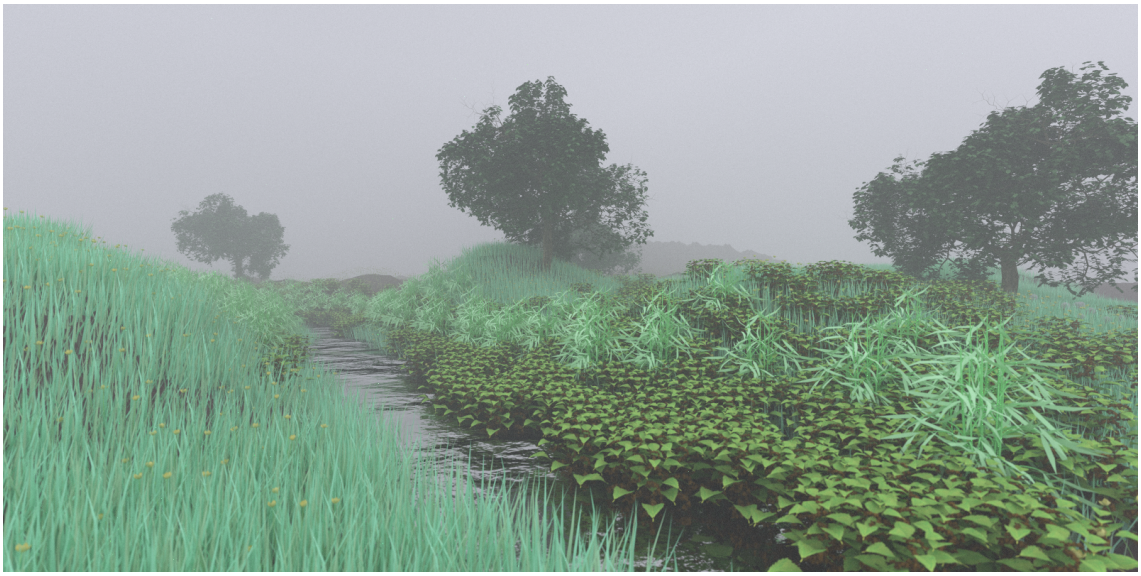

Abbildung 7.7: Die Landschatfsszene „plants“ wurde mit Path Tracing generiert.

Abbildung 7.8: Die Sponza-Szene wurde mit Photon Mapping berechnet.

8 Zusammenfassung und Ausblick

In dieser Arbeit wurden zwei Verfahren vorgestellt um die globale Beleuchtung mit partizipierenden Medien zu berechnen.

Zuerst wurde das Renderingframework, pbrt, vorgestellt. Eine umfangreiche Sammlung an Verfahren zur Bildsynthese ist in diesem Programm zu finden. Photon Mapping und Path Tracing sind hier bereits implementiert. Jedoch fehlte die Berechnung von Mehrfachstreuung in partizipierenden Medien. Dies wurde mit dieser Studienarbeit für die genannten Verfahren nachgeholt.

Als nächstes wurden die wichtigsten Grundlagen, wie die Transportgleichung von Radiance auf Oberflächen und anschließend in Medien, erläutert. Im nächsten Kapitel wurde mit dem Path Tracing ein Verfahren für die globale Beleuchtung von Oberflächen vorgestellt. Idee dabei war es Pfade zurückzuverfolgen, auf denen Radiance durch die Bildebene zur Kamera gelangen kann. Diese Idee wurde in der Erweiterung aufgegriffen, die Verallgemeinerung auf partizipierende Medien vorgestellt und schließlich wichtige Aspekte der Implementierung beschrieben.

Das nächste Thema war das Photon Mapping. Dieses Verfahren approximiert die globale Beleuchtung mit Hilfe der Simulation der Wege von Lichtpartikeln. Ein Bild wird hier in zwei Pässen generiert. Im Ersten werden die Wege simuliert und je Oberflächeninteraktion Spuren in eine Photon Map abgelegt. Mit dieser werden im zweiten Pass die indirekte Beleuchtung und Kaustiken berechnet. Auch hier wurde das Verfahren für die Beleuchtung von Oberflächen vorgestellt und in der Erweiterung auf Volumina generalisiert. Dort wird auf ähnliche Weise die Mehrfachstreuung berechnet und mit Hilfe des Ray Marching die gesamte Beleuchtunginformation entlang eines Strahles ermittelt. Auch hier wurden abschließend die wichtigsten Punkte die Implementierung betreffend erläutert.

Im Kapitel über Perlin Noise wurde eine Methode vorgestellt um anpassbares Rauschen zu erzeugen. Damit können Texturen synthetisiert werden oder wie in diesem Fall niedrig aufgelöse Volumendatensätze um Details ergänzt werden. Darüber hinaus wurde auch eine sehr einfache Art prozedurale Medien zu erstellen gezeigt, welche in pbrt implementiert wurde.

Schließlich wurden Ergebnisse aus den Berechnungsläufen gezeigt. Bei beiden Verfahren wurde Stärken und Schwächen gezeigt: Obwohl es die globale Beleuchtung nicht immer akkurat berechnet, überzeugt Photon Mapping durch dessen Geschwindigkeit und der Fähigkeit Kaustiken hervorzuheben. Während Path Tracing als Refenzverfahren die Korrektheit des Ergebnisses garantiert, benötigt es jedoch sehr viel Rechenzeit.

Ausblick

Neben dem Path Tracing in der hier vorgestellten Form gibt es mit dem Bidirektional Path Tracing oder dem Metropolis Light Transport effizientere Verfahren, die ebenso die Korrektheit des Ergebnises sicherstellen. In [LW96] wird Bidirectional Path Tracing um die Beleuchtung mit partzipierenden Medien erweitert, jedoch ohne die Berücksichtigung von emittierenden Volumina. Bidirectional Path Tracing bedeuted auch Pfade von den Lichtquellen aus zu verfolgen. Hier könnte der im Rahmen dieser Studienarbeit implementierte Algorithmus erweitert werden um ein Ergebnis effizienter zu berechnen. In pbrt ist bereits Metropolis Light Transport implementiert. Hier wäre es denkbar diesen Algorithmus wie in [PKK00] beschrieben zu erweitern.

Das hier implementierte Photon Mapping unterstützt emittierende Medien nur bedingt. Ist keine Lichtquelle vorhanden so bleibt das Bild zum größten Teil schwarz. An dieser Stelle könnte man, wie in [Jen01] angedacht, Photonen auch aus dem Medium verschießen und so Bilder mit leuchtendem Plasma oder Feuer akkurater zu berechnen.

Literaturverzeichnis

[Ben75] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, (18(9)):509–517, 1975. (Zitiert auf Seite 32)

[BT92] N. Bhate, A. Tokuta. Photorealistic Volume Rendering of Media with Directional Scattering. In *Proc. Eurographics Workshop on Rendering Techniques*, pp. 227–245. 1992. (Zitiert auf Seite 7)

[Cha50] S. Chandrasekhar. *Radiative Transfer*. Clarendon Press, Oxford, UK, 1950. (Zitiert auf den Seiten 15 und 35)

[HG41] L. G. Henyey, J. L. Greenstein. Diffuse radiation in the galaxy. *Astrophysical Journal*, (93):70–83, 1941. (Zitiert auf Seite 14)

[JC98] H. W. Jensen, P. H. Christensen. Efficient Simulation of Light Transport in Scenes with Participating Media using Photon Maps. In *SIGGRAPH 98 Conference Proceedings*, pp. 311–320. 1998. (Zitiert auf den Seiten 7, 15, 34, 35, 36, 37, 39 und 42)

[Jen96] H. W. Jensen. Global Illumination using Photon Maps. In *Proc. Eurographics Workshop on Rendering Techniques*, pp. 21–30. 1996. (Zitiert auf den Seiten 27, 28 und 30)

[Jen01] H. W. Jensen. *Realistic Image Synthesis Using Photon Mapping*. 2001. (Zitiert auf den Seiten 27, 34, 36 und 54)

[Kaj86] J. Kajiya. The Rendering Equation. In *Computer Graphics, vol 20 (SIGGRAPH 86 Conference Proceedings)*, pp. 143–150. 1986. (Zitiert auf den Seiten 14 und 17)

[KH84] J. Kajiya, B. V. Herzen. Ray Tracing Volume Desities. In *Computer Graphics, vol 18 (SIGGRAPH 84 Conference Proceedings)*, pp. 165–174. 1984. (Zitiert auf Seite 7)

[Kol05] T. Kollig. *Efficient Sampling and Robust Algorithms for Photorealistic Image Synthesis*. Ph.D. thesis, Technische Universität Kaiserslautern, 2005. (Zitiert auf den Seiten 11, 21 und 23)

[LW93] E. P. Lafortune, Y. D. Willems. Bidirectional Path Tracing. In *Proc. of CompuGraphics*, pp. 145–153. 1993. (Zitiert auf Seite 34)

[LW96] E. P. Lafortune, Y. D. Willems. Rendering Participating Media with Bidirectional Path Tracing. In *Proc. Eurographics Workshop on Rendering Techniques*, pp. 91–100. 1996. (Zitiert auf den Seiten 7, 15, 23, 24 und 54)

[Per85] K. Perlin. An Image Synthesizer. In *ACM SIGGRAPH Computer Graphics*, pp. 19(3): 287–296. 1985. (Zitiert auf den Seiten 7, 41 und 42)

[PH10a] M. Pharr, G. Humphreys. PBRT Home, 2010. URL http://www.pbrt.org. (Zitiert auf Seite 7)

[PH10b] M. Pharr, G. Humphreys. *Physically Based Rendering, From Theory to Implementation, Second Edition*. Morgan Kaufmann, 2010. (Zitiert auf den Seiten 9, 11, 12, 14, 17, 18, 19, 20, 28, 29, 30, 32, 34, 36 und 42)

[PKK00] M. Pauly, T. Kollig, A. Keller. Metropolis Light Transport for Participating Media. In *Proc. Eurographics Workshop on Rendering Techniques*, pp. 11–22. 2000. (Zitiert auf den Seiten 7, 14, 15, 22 und 54)

[Rus94] H. Rushmeier. Rendering Participating Media: Problem and Solutions from Application Area. In *Proc. Eurographics Workshop on Rendering Techniques*, pp. 35–36. 1994. (Zitiert auf Seite 7)

[Sie92] R. Siegel. *Thermal Radiation Heat Transfer, 3rd Edition*. Hemisphere Publishing Corporation, New York, 1992. (Zitiert auf den Seiten 34 und 36)

[Vea97] E. Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. Ph.D. thesis, Stanford University, 1997. (Zitiert auf den Seiten 28, 31 und 37)

Alle URLs wurden zuletzt am 30.06.2011 geprüft.

Erklärung

Hiermit versichere ich, diese Arbeit selbständig verfasst und nur die angegebenen Quellen benutzt zu haben.

(Zeno-Oliver Groß)