



UNIVERSITÄT
KOBLENZ · LANDAU

Fachbereich 4: Informatik

Volumetrische Beleuchtung

Bachelorarbeit

zur Erlangung des Grades Bachelor of Science (B.Sc.)
im Studiengang Computervisualistik

vorgelegt von
Michael Kingston

Erstgutachter: Prof. Dr.-Ing. Stefan Müller
(Institut für Computervisualistik, AG Computergraphik)
Zweitgutachter: Kevin Keul, M.Sc.

Koblenz, im April 2018

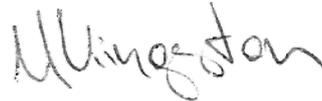
Erklärung

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

Ja Nein

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

Koblenz, 19.04.2018



.....
(Ort, Datum)

.....
(Unterschrift)



Aufgabenstellung für die Bachelorarbeit

Michael Kingston

(Matr.-Nr. 215 100 808)

Thema: Volumetrische Beleuchtung

In keinem Bereich der Informatik hat sich die Hardware so rasant entwickelt, wie im Bereich der Computergraphik. So ist es heute möglich, sehr komplexe Szene in sehr hoher grafischer Qualität und mit sehr hohen Bildwiederholraten zu rendern und zu raytracen. Ein interessantes Beleuchtungsphänomen ist dabei die volumetrische Beleuchtung. Darunter versteht man in der Regel Streupartikel im lichtübertragenden Medium, so dass die Leuchtdichte entlang eines Strahl durch Streuung oder Absorption abnimmt.

Ziel dieser Arbeit ist es, sich in bestehende Verfahren zum Thema Volumetrische Beleuchtung einzuarbeiten. Ziel ist es, mindestens ein Verfahren auszuwählen, umzusetzen und auf seine Grenzen und Möglichkeiten zu untersuchen. Alternativ können auch mehrere Verfahren umgesetzt und gegeneinander verglichen werden. Im Vordergrund steht dabei die Implementierung einer Anwendung, welche die volumetrische Beleuchtung gut in Szene setzt.

Schwerpunkte dieser Arbeit sind:

1. Analyse und Recherche verschiedener Verfahren
2. Einarbeitung in relevante Technologien und Frameworks
3. Konzeption der Umsetzung
4. Implementierung
5. Bewertung und Dokumentation der Ergebnisse

Koblenz, den 23.10.2017

- Prof. Dr. Stefan Müller -

- Michael Kingston -

Kurzfassung

Volumetrische Beleuchtung ist ein verbreitetes Beleuchtungsphänomen in der Natur und trägt in weiten Teilen zu dem realistischen Erscheinen von computergenerierten Bildern bei. Diese Arbeit befasst sich mit den physikalischen Hintergründen dieses Phänomens, listet bekannte Modelle zur Visualisierung von volumetrischer Beleuchtung in der Computergrafik auf und vergleicht anschließend drei Verfahren, mit denen eine volumetrische Beleuchtung mit heutiger Grafikhardware visualisiert werden kann. Diese Verfahren werden anhand einer Implementation in OpenGL hinsichtlich ihrer Möglichkeiten und Einschränkungen, sowie ihres Rechenaufwands miteinander verglichen.

Abstract

Volumetric lighting is a common lighting phenomena in nature and carries an important role in the realistic appearance of computergenerated images. This thesis explains the physical background of this phenomena and lists common modells for visualising volumetric lighting in Computergraphics. Following this, this thesis compares three methods for visualising volumetric lighting with modern graphics hardware and compares these according to their abilities, restrictions and performance in an OpenGL implementation.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Anwendungsbereiche	2
2	Grundlagen	4
2.1	Licht	4
2.1.1	Messung von Licht	6
2.2	Rendering Equation	6
2.3	Participating Media	7
2.3.1	Absorption	8
2.3.2	Emission	9
2.3.3	Streuung	9
2.3.4	Phasenfunktionen	10
2.4	Radiance Transport Equation	12
3	Streuungsmodelle	13
3.1	Analytische Modelle	16
3.2	Deterministische Modelle	17
3.3	Stochastische Modelle	18
3.4	Multiple Scattering	18
3.5	Fake Media	19
4	Implementation	21
4.1	Raymarching	22
4.1.1	Grundlegendes Prinzip	22
4.1.2	Umsetzung in OpenGL	23
4.2	Analytisches Modell	24
4.2.1	Formulierung eines zu verarbeitenden Terms aus der RTE	25
4.2.2	Herleitung eines analytisch vereinfachten Terms	25
4.2.3	Umsetzung in OpenGL	29
4.3	Shadow Volumes	29
4.3.1	Generieren der Shadow Volumes	29
4.3.2	Reduktion des abzutastenden Bereiches	30
4.3.3	Umsetzung in OpenGL	31
5	Auswertung	32
5.1	Raymarching	32
5.1.1	Notwendigkeit einer Rauschtextur	32
5.1.2	Rechenaufwand	32
5.2	Analytisches Modell	35
5.2.1	Rechenaufwand	35
5.2.2	Probleme mit der Textur	35

5.2.3	Nachteile des analytischen Modells	36
5.3	Shadow Volumes	37
5.3.1	Vorteile gegenüber dem Raymarching	37
5.3.2	Rechenaufwand	38
6	Fazit	40

1 Einleitung

Eines der größten Ziele der Computergrafik ist es eine möglichst realistische Nachbildung der realen Welt zu erschaffen. Es ist jedoch üblich, dass Anwendungen der Computergrafik lediglich starke Vereinfachungen der echten Welt darstellen, da unter Beachtung der vollen Komplexität der Realität selbst die stärksten Prozessoren den enormen Rechenanforderungen nicht gewachsen sind.

Um realitätsgetreue Bilder zu generieren müssen eine Unmenge an Faktoren in Betracht gezogen werden. Diese Faktoren tragen zur physikalischen Korrektheit der Bilder bei und können ein breites Spektrum an Komplexität besitzen. So ist es bspw. sehr speicheraufwändig die genauen Materialeigenschaften eines Objektes, unter Berücksichtigung aller Variablen, zu erfassen. In den meisten Fällen sind es jedoch nicht die Anforderungen an den Speicher, sondern die Anforderungen an die Rechenleistung, die die tatsächliche Hürde des physikalisch korrekten Renderings darstellen. Das Rendern eines einzelnen Bildes einer photorealistischen Szene kann Stunden dauern, an eine flüssige Bildwiederholrate ist in diesem Bereich der Computergrafik nicht zu denken. Deshalb ist es üblich in interaktiven Anwendungen, bspw. Videospielen, die Welt zu vereinfachen.

Diese Vereinfachungen beinhalten einerseits die Reduktion von Objekten auf eine handhabbare Anzahl an Eckpunkten und sinnvoll speicherbare Texturen. Andererseits werden hier jedoch am meisten die Shader der Anwendungen vereinfacht. Da diese den größten Einfluss auf den Rechenaufwand besitzen, werden sie auf größt mögliche Effizienz getrimmt, wobei es ersteinmal unerheblich ist, ob das resultierende Bild physikalisch korrekt ist oder nicht. In vergangenen Jahren war es irrelevant, ob ein Videospiel einem Photo der realen Welt zum Verwechseln ähnlich sieht. Mit wachsender Hardwarekapazität und wachsender Erwartungshaltung der Spieler jedoch, setzen immer mehr Entwickler auf besseres visuelles Auftreten ihrer Spiele und integrieren allerlei besondere Effekte in ihre Shader.

Ein solcher, sehr gefragter Effekt, ist die volumetrische Beleuchtung. Dabei handelt es sich um die Darstellung von sichtbaren Lichtstrahlen in einem transparenten Medium, welche dem Bild eine spezielle Atmosphäre hinzufügen und für eine realistische, physikalisch korrekte Darstellung in bestimmten Szenen unerlässlich ist.

Grundsätzlich werden in den meisten Modellen für die Interaktion des Lichts lediglich die Oberflächen von Objekten betrachtet. Das Medium durch welches das Licht von der Quelle an die Oberfläche übertragen wird, wird jedoch meist vollständig ignoriert. Dabei ist es aufgrund von Verunreinigungen in diesen Medien falsch anzunehmen, dass Lichtstrahlen nicht mit diesem interagieren. Verunreinigungen sind meist mikroskopisch kleine Partikel, wie Staub oder Nebel, welche im einzelnen für das menschliche Auge unsichtbar sind, jedoch durch die große Anzahl und die Volumen-



(a) Lichtstrahlen im Wald



(b) Sonnenstrahlen mit Wolken

Abbildung 1: Beispiele volumetrischer Beleuchtung in der Natur

abdeckende Verteilung einen sichtbaren Effekt auf das Gesamtergebnis haben können. Es sind solche Imperfektionen der Natur, welche es der Computergrafik sehr schwer machen realistische Bilder der Realität zu generieren.

Durch den stetigen Wachstum an Rechenleistung moderner Prozessoren, rückt die Darstellung solcher Phänomene des Lichts immer weiter in den Vordergrund und in den vergangenen Jahrzehnten haben sich immer bessere Algorithmen zur effizienten und korrekten Darstellung von volumetrischer Beleuchtung entwickelt. Während noch vor zehn Jahren die Rechenleistung für viele Algorithmen nur knapp ausreichte, liefern heutige Grafikkarten mehr als genug Rechenleistung und auch Speicher, um noch vor Jahren undenkbare Effekte in hoher Auflösung und mit flüssigen Bildwiederholraten zu rendern.

1.1 Anwendungsbereiche

Die Anwendungsbereiche für volumetrische Beleuchtung sind vielfältig. Für die realistische und physikalisch korrekte Darstellung einer Szene ist sie unerlässlich, jedoch birgt die Visualisierung von Lichtstrahlen in einem Medium nicht nur ästhetische Funktionen.

Wie bereits genannt, legt die Spieleindustrie immer mehr Wert auf die Integration von herausstechenden Effekten in ihre Spiele, darunter fällt selbstverständlich auch die volumetrische Beleuchtung, welche in diesem Bereich vermutlich besser unter dem Namen Godrays bekannt ist.

Die Visualisierung von volumetrischem Licht findet aber ebenfalls in vielen anderen Bereichen Anwendung. Oftmals geht es nicht um eine interaktive Visualisierung, sondern um eine physikalisch korrekte Berechnung unter Berücksichtigung verschiedener Parameter, um die Auswirkungen von Nebel oder ähnlichem in der realen Welt besser zu verstehen und Produkte an die erschwerten Sichtbedingungen anzupassen. Dies kann eine Simulation von Autoscheinwerfern im Nebel oder Sichtbarkeit von Not-

ausgängen bei dichten Rauch beinhalten. Ebenfalls müssen Simulationen für die Ausbildung von Piloten, Nebel und Wolken korrekt darstellen können, um die Piloten auf erschwerte Sichtbedingungen vorzubereiten. Letztendlich sind auch das Militär, sowie die Industrie daran interessiert, Sensoren für allerlei Anwendungen auf Effekte der volumetrischen Beleuchtung vorzubereiten, sodass Systeme korrekt und störungsfrei arbeiten können.

Einen gesonderten Bereich stellt die medizinische Visualisierung dar. Mit moderner Technik können Aufnahmen des Körperinneren produziert werden. Um den drei-dimensionalen Datensatz lesbar darzustellen, wird oftmals volumetrisches Rendering genutzt. Dieses Verfahren weist Parallelen zu volumetrischer Beleuchtung auf, wird aber sehr unterschiedlich gehandhabt, da ein medizinischer Datensatz in den meisten Fällen einen sehr hohen Speicherbedarf besitzt.

2 Grundlagen

2.1 Licht

Um ein besseres Verständnis über die volumetrische Beleuchtung zu erlangen, ist es notwendig sich erst einmal mit den physikalischen Eigenschaften von Licht im Allgemeinen zu befassen. Hierbei muss verstanden werden, woher das Licht kommt und wie es mit anderen Einheiten und somit auch Nebel, Rauch oder ähnlichem in der Welt interagiert.

Elektromagnetische Strahlung Elektromagnetische Strahlung ist ein alltägliches Phänomen, welches sich auf ein breites Spektrum verteilt, mit welchem wir auf unterschiedliche Weisen interagieren. Der größte Teil dieses elektromagnetischen Spektrums ist für uns nicht von Relevanz. Das sichtbare Licht bildet nur einen kleinen Teil des Spektrums für welches die Evolution das menschliche Auge empfindlich gemacht hat.

Abhängig davon auf welcher Ebene man elektromagnetische Strahlung untersucht, zählt sie zu einem der verwirrendsten Themenbereichen in der Physik, trotz dass sie uns alltäglich umgibt. Der Grund dafür liegt darin, dass sie auf einer Ebene von kleinsten Teilchen in Quantenphysik und Relativitätstheorie übergeht und die Gesetze der klassischen Physik, wie der Mensch sie spürt nicht mehr ausreichen, um Licht zu beschreiben [1].

Elektromagnetische Strahlung wirkt auf einer makroskopischen Ebene, wie ein konstanter, unmittelbarer Fluss von Teilchen, welche sich nur in einer Summe messen lassen. Die Radiometrie befasst sich mit der Messung von elektromagnetischer Strahlung und ist die Erweiterung der Photometrie auf das gesamte elektromagnetische Spektrum [2].

Photonen Schauen wir uns Licht jedoch auf der Ebene der kleinsten Teilchen an, stellen wir fest, dass es ein minimales Bündel an Licht gibt, wir bezeichnen dieses kleinste Teilchen als Photon. Jedes Photon besitzt eine bestimmte Menge an Energie. Diese Energie bestimmt mit welcher Frequenz dieses Photon schwingt und folglich auch die Wellenlänge des resultierenden Lichts. Die Energie E bestimmt die Frequenz f und Wellenlänge λ durch

$$E = hf = h * c/\lambda \quad (1)$$

wobei $h = 6.626 * 10^{-34} kg * m^2/s$ das planksche Wirkungsquantum und $c = 2.996 * 10^8 m/s$ die Geschwindigkeit von elektromagnetischer Strahlung in einem Vakuum beschreiben.

Das elektromagnetische Spektrum Das elektromagnetische Spektrum ist durch die möglichen Energieniveaus der Strahlung in verschiedene Bereiche unterteilt. Diese Bereiche werden üblicherweise durch die Wellen-

länge der Strahlung beschrieben, wobei lange Wellenlängen mit niedriger Energie einhergehen. Die niedrigsten Energien besitzen Radiowellen, gefolgt von Mikrowellen. Danach findet sich Infrarotstrahlung, das sichtbare Licht und ultraviolette Strahlung. Mit höheren Energien folgen Röntgenstrahlen, Gammastrahlen und schließlich kosmische Strahlung. Uns interessiert vor allem das sichtbare Licht. Ab einer Wellenlänge von ca. 380 nm können wir violettes Licht sehen, bei 450 nm blaues, bei 530 nm grünes, bei 600 nm gelbes, bei 670 nm rotes Licht und bei ca. 780 nm endet das sichtbare Spektrum.

Interaktion von Licht mit Atomen Sichtbares Licht ist also nur ein Teil des Spektrums, aber woher genau kommt das Licht und wie interagiert es mit einem Medium? Um diese Frage zu beantworten muss man sich ansehen, wie Licht mit Atomen interagiert. Atome bestehen aus einem Nukleus und umkreisenden Elektronen, welche um den Nukleus herum in Orbitalen schwingen. Damit Elektronen in diesem Orbital bleiben, benötigen sie eine bestimmte Menge an Energie. Es ist jedoch auch möglich, dass ein Elektron durch die Aufnahme oder Abgabe von Energie auf ein höher oder niederenergetisches Orbital springt. Wenn ein Elektron hinunterfällt wird typischerweise ein Photon entsprechend der Energiedifferenz abgegeben, genauso ist es möglich, dass ein Atom ein Photon absorbiert. In diesem Fall muss jedoch das Photon genau die richtige Menge an Energie tragen, sodass das Elektron auf ein höheres Orbital springen kann. Für gewöhnlich kommt ein Atom jedoch nicht einzeln, sondern in einem dichten Paket, sodass die Energieniveaus zwischen den Orbitalen sich auf ein breites Band von Energien verteilen. Verschiedene Materialien besitzen verschiedene Energiebänder, wodurch nur bestimmte Wellenlängen aufgenommen und abgegeben werden und Materialien Farben verliehen wird.

Durch diese Interaktion von Licht mit Atomen entstehen auch virtuelle Übergänge, bei denen Photonen in einem nicht messbaren Zustand aufgenommen und sofort wieder abgegeben werden, welche Einfluss auf die Ausbreitungsgeschwindigkeit von Licht nehmen und somit für Brechungen an Materialübergängen verantwortlich sind. Außerdem entstehen Effekte wie Fluoreszenz und Phosphoreszenz. In Metallen finden sich oft frei bewegliche Elektronen, weshalb die Bandbreite an Energien sehr hoch ist und die meisten Metalle reflektiv sind. In anderen Materialien finden sich ebenfalls freie Elektronen, die sich jedoch nicht so ungehindert bewegen können. Diese interagieren stark mit den restlichen Atomen und verursachen somit Vibrationen bzw. Wärme. Dieser Prozess ist ebenfalls umkehrbar, Licht und Wärme lassen sich in einander umwandeln. Dies funktioniert sehr gut mit Infrarotstrahlung [1].

Der letzte Prozess findet mit jedem Objekt statt, welches eine Temperatur über dem absoluten Nullpunkt besitzt, jedoch entsteht meistens nicht

sichtbare Strahlung. So sehen wir im Normalfall nur die Strahlung von besonders stark erhitzten Objekten, wie bspw. glühend heißen Metallen. Ein sogenannter Schwarzer Körper (engl. Black Body Radiator) dient als Referenz für die durch Wärme abgegebene Strahlung [3].

2.1.1 Messung von Licht

Wie bereits erwähnt besteht Licht aus Photonen, doch auf Grund der großen Anzahl und ihrer unterschiedlichen Energiemengen ist es nicht sinnvoll Photonen zu zählen. Licht wird daher als kontinuierlicher Fluss modelliert. Die Wellenlänge des Lichtes bestimmt dessen Energie und diese Energien lassen sich zu einer Gesamtenergie aufaddieren, welche sich über einen festgelegten Zeitraum messen lässt.

Man nennt diese Energie die Strahlungsleistung, auch genannt Strahlungsfluss (engl. Radiant Flux). Sie beschreibt die Energiemenge, welche über einen Zeitraum dt von Licht transportiert wird und wird in Watt $W = kg * m^2/s^3 = J/s$ gemessen. Aus der Strahlungsleistung lassen sich die Bestrahlungstärke und spezifische Ausstrahlung ableiten. Diese geben an wie viel Strahlungsleistung an einer gegebenen Fläche ankommt bzw. von einer solchen Fläche abgegeben werden. Diese werden in W/m^2 gemessen. Interessanter für den Bereich der Computergrafik ist jedoch die Strahldichte (engl. Radiance). Diese misst die Bestrahlungstärke bzw. die spezifische Ausstrahlung des Lichtes an einem Punkt in einer bestimmten Richtung und wird gemessen in $W/(m^2 * sr)$. Ein Steradian sr ist eine Maßeinheit für Raumwinkel wobei eine Kugel mit Radius $r = 1m$ eine Oberfläche von $4 * PI * sr$ hat. Nützlich kann außerdem auch die Strahlintensität sein, welche die ausgehende Leistung einer Lichtquelle in eine bestimmte Richtung beschreibt. Sie wird gemessen in W/sr [2].

2.2 Rendering Equation

Die Rendering Equation ist ein Gleichungssystem, welches dazu genutzt werden kann, um die genaue Beleuchtung eines jeden Oberflächenpunktes in einem abgeschlossenen System zu beschreiben. Sie beschreibt für jeden Oberflächenpunkt x die Radianz in eine bestimmte Richtung ω durch die eigene Ausstrahlung des Punktes, zuzüglich einer Streuung der gesamten Bestrahlungsstärke aus möglichen einstrahlenden Richtungen ω' . Dabei ist ω' durch die Sichtbarkeit von anderen Oberflächenpunkten y limitiert.

Die allgemeine Formel für die Rendering Equation ist definiert durch

$$L(x, \omega) = L_e(x, \omega) + \int_{\Omega'} L(y, \omega') f_r(\omega', x, \omega) \cos^+ \theta'_x d\omega' \quad (2)$$

wobei $L(x, \omega)$ die Strahldichte L eines Oberflächenpunktes x in eine ausgehende Richtung ω bezeichnet. Diese setzt sich aus der emittierten Strahl-

dichte $L_e(x, \omega)$ von x in Richtung ω und einem gestreuten Anteil des Lichtes aus einfallenden Richtungen $\omega' \in \Omega'$. Die Menge Ω' umfasst alle Richtungen ω' von denen x von einem Oberflächenpunkt y aus sichtbar ist. $f_r(\omega', x, \omega)$ ist eine BRDF, passend zu den Eigenschaften der Oberfläche und $\cos^+ \theta'_x$ gleicht dem Minimum des Kosinus Wertes des Winkels θ'_x zwischen der Oberflächennormale von x und $-\omega'$ und null [4].

Die Rendering Equation ist nur mit sehr hohen Rechenaufwand lösbar. Lokale Beleuchtungsmodelle gehen daher von starken Vereinfachungen der Rendering Equation aus. Sie reduzieren meistens den Anteil des gestreuten Lichtes auf einen Term welcher lediglich die ankommende Strahlung der Lichtquelle selbst in Betracht zieht und somit die spezifische Ausstrahlung von sichtbaren Oberflächenpunkten durch Streuung von Licht vernachlässigt. Die auf einen Punkt oder eine Richtung reduzierten Lichtquellen lassen ihren Anteil in der Rendering Equation sehr einfach bestimmen. Somit vereinfacht sich die Gleichung zu

$$L(x, \omega) = L_e(x, \omega) + \sum_l (L_l^{in}(x) f_r(\omega'_l, x, \omega) \cos^+ \theta'_l) \quad (3)$$

wobei l die Menge der Lichtquellen darstellt und $L_l^{in}(x)$ die einkommende Strahlung der Lichtquelle an dem Punkt x definiert.

Globale Beleuchtungsmodelle vereinfachen die Rendering Equation nicht so drastisch, wie das vorherige lokale Modell. Anstatt, dass die Menge an Punkten Ω' durch die Menge an Lichtquellen l ersetzt wird, wird der gestreute Anteil des Lichtes auf eine geringe, feste Anzahl an Iterationen bzw. Lichtstreuungen reduziert, da der Anteil des gestreuten Lichtes nach einigen Iterationen zu vernachlässigen ist. Das heißt, dass die Rendering Equation initial nur mit emittierter Strahlendichte berechnet wird und anschließend mehrfach der gestreute Anteil neu errechnet wird, sodass über viele Iterationen die tatsächliche Lösung der Rendering Equation approximiert wird.

2.3 Participating Media

Während die Rendering Equation die Strahlung von allen Oberflächen in Betracht zieht, vernachlässigt sie eine wichtige Komponente. Es ist äußerst untypisch, dass Licht nicht mit Medium interagiert, durch welches es wandert. Licht bewegt sich ohne äußere Einflüsse zwar nur geradlinig, jedoch gibt es im übertragenden Medium nahezu immer Faktoren, die dazu beitragen, dass die Lichtstrahlen von ihrem gedachten Pfad abkommen, auch wenn diese in vielen Fällen nicht nennenswert zu dem resultierenden Bild beitragen. Ursprünglich stammt die Berechnung von Licht in teilnehmenden Medien aus der atmosphärischen Optik und Astronomie, jedoch hat die Interaktion von Licht mit dem übertragenden Medium sich über die

letzten Jahrzehnte immer mehr in der Computergrafik zurecht gefunden [5].

Die zuvor genannte Rendering Equation umfasst lediglich die Beleuchtung von allen Oberflächenpunkten und vernachlässigt dabei die Interaktion mit dem Medium. In vielen Fällen ist dies eine sinnvolle Vereinfachung der tatsächlich korrekten Beleuchtung, da das Medium Luft bei Tageslicht die Beleuchtung nur sehr minimal verändert und nur über große Distanzen eine Rolle spielt. Wichtig wird die Einbeziehung des übertragenden Mediums, sobald starke Verunreinigungen auftreten. Mit Verunreinigungen sind kleine Partikel in dem Medium gemeint, welche einzeln betrachtet zu klein sind, um einen sichtbaren Effekt zu haben, jedoch auf Grund ihrer großen Anzahl in manchen Fällen eine wichtige Rolle für der korrekte Beleuchtung tragen. Ein geringer Anteil des Lichtes interagiert in diesen Fällen mit den winzigen Partikeln im Medium.

Medien für welche die Betrachtung der Interaktion mit Licht unerlässlich ist, sind vor allem Wolken, Nebel und Dunst, staubige Luft, Qualm und Rauch, sowie getrübbtes oder gefärbtes Wasser. Wenn ein Medium nur äußerst wenige dieser Partikel aufweist, wie zum Beispiel ein Vakuum, dann ist es eine sehr valide Approximation, anzunehmen, dass das Licht sich geradlinig ohne Beeinflussung fortbewegt.

Im Allgemeinen lassen sich Interaktionen zwischen Licht und dem teilnehmenden Medium auf drei Vorgänge zurückführen. Diese sind Absorption, Emission und Streuung. Diese Vorgänge können über das gesamte Volumen des teilnehmenden Mediums homogen oder auch inhomogen verteilt sein, d.h. dass die Wahrscheinlichkeiten für das Auftreten dieser Vorgänge überall gleich groß sein oder auch stark variieren können. In der Realität sind die folgenden Gleichungen zusätzlich zu ihren genannten Parametern ebenfalls von der Wellenlänge λ des Lichtes abhängig, der Einfachheit halber wird dies vernachlässigt.

2.3.1 Absorption

Die Absorption beschreibt eine Reduktion der Strahldichte durch Konvertierung von Licht in eine andere Form von Energie z.B. Wärme. Es besteht eine Wahrscheinlichkeit, dass das Licht auf seiner Bahn auf ein Partikel trifft und absorbiert wird. Bspw. ist die Wahrscheinlichkeit, dass ein Lichtstrahl ungehindert durch dichten, schwarzen Rauch kommt sehr gering, wohingegen die Wahrscheinlichkeit für Absorption in einem Vakuum gleich null ist. Der Rauch verhindert somit, dass das Licht das Ziel erreicht. Die Stärke der Absorption wird beschrieben durch den Absorptionswirkungsquerschnitt des Mediums σ_a , welche der Wahrscheinlichkeit entspricht, dass das Licht auf einer bestimmten Strecke innerhalb des Mediums absorbiert wird und in $1/m$ gemessen wird. Dieser Faktor ist in einfachen Fällen für gewöhnlich abhängig von der Position innerhalb des Me-

diums, er kann jedoch ebenfalls von der Strahlungsrichtung als auch von der Wellenlänge der Strahlung abhängen. Das Licht verliert einen Bruchteil der Strahldichte abhängig von der Distanz:

$$\frac{dL}{L} = -\sigma_a dx \quad (4)$$

Nach einer Distanz x beträgt die Strahldichte also:

$$L(x) = L_0 * e^{-\int_0^x \sigma_a(x') dx'} \quad (5)$$

bzw. für ein konstantes σ_a :

$$L(x) = L_0 * e^{-\sigma_a * x} \quad (6)$$

Im Falle verschiedener Absorptionen von verschiedenen Partikeln lassen sich die Absorptionsfaktoren aufsummieren. Ein Medium mit starker Absorption kann zudem durch die starke Reduktion der Strahldichte von der Lichtquelle zu Schattenbildung auf einer dahinter liegenden Oberfläche führen [6].

2.3.2 Emission

Die Emission entspricht der Energie, die der Umgebung durch die Photonenabgabe von Partikeln hinzugefügt wird. Sie ist das Gegenstück zur Absorption und entsprechend erhöht sie die Strahldichte eines Lichtstrahls während es durch ein emittierendes Medium wandert. Für gewöhnlich ist emittiertes Licht nicht von der ursprünglichen Strahldichte abhängig. Meist ist sie richtungsunabhängig aber dafür stark an bestimmte Wellenlängen gebunden. Sie ist anhand des Mediums definiert durch $L_{ve}(x, \omega)$ [6].

2.3.3 Streuung

Streuung beschreibt wie Licht, welches aus einer Richtung kommt, durch Kollision mit Partikeln in andere Richtungen gestreut wird. Die Streuung ist der Absorption ähnlich, jedoch wird das Licht nicht in eine andere Form von Energie verwandelt, sondern erfährt durch Interaktion mit dem Partikel eine Richtungsänderung. Es gibt zwei verschiedene Arten von Streuung, sogenanntes In-Scattering und Out-Scattering [6].

Out-Scattering Das Out-Scattering beschreibt den Verlust an Strahldichte, da Lichtstrahlen von ihrem ursprünglichen Pfad weg gestreut werden. Sie lässt sich genauso berechnen, wie die Absorption, besitzt jedoch ihren eigenen Streukoeffizienten σ_s . Absorptionskoeffizient und Streukoeffizient lassen sich zusammenfassen zu einem einzigen Dämpfungskoeffizienten

$$\sigma_t = \sigma_a + \sigma_s \quad (7)$$

Aus dem Dämpfungskoeffizienten lässt sich für eine gegebene Distanz x die optische Dichte

$$\tau = \int_0^x \sigma_t(z) dz \quad (8)$$

ermitteln. Diese findet weiterhin in der Transmission T_r Verwendung.

$$T_r = e^{-\tau} = e^{-\int_0^x \sigma_t(z) dz} \quad (9)$$

Das Verhältnis von Streuung zu Absorption wird Albedo genannt und wird durch folgende Gleichung bestimmt.

$$\beta = \sigma_s / \sigma_t \quad (10)$$

In-Scattering Das In-Scattering hingegen beschreibt den Anteil von Licht aus anderen Richtungen, welches in den betrachteten Lichtstrahl hinein gestreut wird und mit diesem weiter wandert. Somit steigt die betrachtete Strahldichte, weshalb es notwendig ist die Streuung genauer zu definieren als für das Out-Scattering. Anstatt, dass ein bestimmter Anteil abgezogen wird, wird nun abhängig von den Eigenschaften des Mediums Licht hinzu gefügt. Während es einfach ist herauszufinden, wie viel Licht gestreut wird, ist es nicht ganz so trivial zu sagen, wohin dieses Licht gestreut wird. Zur Lösung dieses Problems nutzt man eine sogenannte Phasenfunktion p , welche abhängig von eingehender und ausgehender Richtung für eine bestimmte Art von Partikeln einen Wert zurückgibt, der dem Anteil des gestreuten Lichtes entspricht. Die Strahldichte für einen Punkt x in eine Richtung ω errechnet sich aus

$$L_i(x, \omega) = \int_{\Omega_{4\pi}} p(x, \omega, \omega') * L(x, \omega') d\omega' \quad (11)$$

Der Term $L(x, \omega')$ bezeichnet die aus Richtung ω' eingehende Strahldichte an dem Punkt x [6].

2.3.4 Phasenfunktionen

Wie bereits genannt beschreibt die Phasenfunktion p den Anteil des gestreuten Lichtes abhängig von Position x , der ausgehenden Richtung ω und eingehenden Richtung ω' . Die Integration der Phasenfunktion über alle eingehenden Richtungen für eine bestimmte ausgehende Richtung muss gleich 1 sein [7]. Sie ist also normalisiert durch die Funktion

$$\int_{\Omega_{4\pi}} p(x, \omega, \omega') d\omega' = 1, \forall \omega \quad (12)$$

Die einfachste Phasenfunktion entspricht der isotropen Streuung mit

$$p_I = \frac{1}{4\pi} \quad (13)$$

Dies bedeutet, dass der gestreute Anteil des Lichtes unabhängig von der Richtung des ein- und ausgehenden Lichtes immer konstant bleibt.

Die Henyey-Greenstein Phasenfunktion [7] ist eine beliebte Phasenfunktion, da die Verteilung der Streuung über eine einzelne Konstante g gesteuert werden kann und sie somit an viele verschiedene Szenarien angepasst werden kann. Bisher war die Phasenfunktion von Position, sowie ein- und ausgehender Richtung des Lichtsstrahls abhängig. Im Folgenden werden diese auf einen Parameter θ vereinfacht. Dabei beschreibt θ den von ein- und ausgehender Richtung eingeschlossenen Winkel. Die Henyey-Greenstein Funktion ist definiert durch

$$p_{HG}(\theta) = \frac{1 - g^2}{4\pi(1 + g^2 - 2g \cos \theta)^{1.5}} \quad (14)$$

wobei der Wert null für die Konstante $g \in [-1, 1]$ der isotropen Phasenfunktion entspricht. Falls $g > 0$ gilt, dann beschreibt die Funktion Vorwärtsstreuung, falls $g < 0$ gilt, dann beschreibt die Funktion Rückwärtsstreuung.

Eine gute Annäherung an die Funktion von Henyey und Greenstein bietet die Schlick Phasenfunktion [7], welche effizienter zu berechnen ist. Sie ist definiert durch

$$p_s(\theta) = \frac{1 - k^2}{4\pi(1 + k \cos \theta)^2} \quad (15)$$

mit dem Wert von k angenähert durch

$$k = 1.55g - 0.55g^3. \quad (16)$$

Die Henyey-Greenstein oder Schlick Phasenfunktion werden für die Lorenz-Mie Streuung genutzt. Diese findet für gewöhnlich an Partikeln statt, welche ungefähr eine Größe besitzen die der Wellenlänge des Lichtes entspricht, d.h. Radius $r = \lambda$. Eigentlich ist Lorenz-Mie Scattering sehr komplex und von der genauen Wellenlänge des Lichtes abhängig, da jedoch die Größe der Mie-Partikel in den allermeisten Fällen variiert, mittelt sich die Wellenlängenabhängigkeit über das Medium aus. Typischerweise findet Lorenz-Mie Streuung in Wettereffekten Anwendung.

Für Streuung an Partikeln einer geringeren Größe mit $r < 0.1 * \lambda$ wird die Rayleigh Phasenfunktion [7] verwendet. Sie ist vereinfacht definiert durch die Funktion

$$p_r(\theta) = \frac{3}{16\pi}(1 + \cos^2 \theta) \quad (17)$$

Rayleigh Streuung ist stark von Wellenlängen abhängig und Grund für die blaue Farbe des Himmels und rote Sonnenuntergänge. Typischerweise wird sie also für Moleküle in der Luft genutzt.

2.4 Radiance Transport Equation

Da nun die Interaktionen von Licht und einem teilnehmenden Medium bekannt sind, ist es möglich daraus eine zusammenhängende Gleichung zu formulieren. Es handelt sich hierbei, um die Erweiterung der Rendering Equation auf das teilnehmende Medium. Somit fließen nun nicht mehr nur die Oberflächenpunkte in die Beleuchtung mit ein, sondern ebenfalls die Interaktionen mit dem Medium, welche für die volumetrische Beleuchtung benötigt werden. Das resultierende Gleichungssystem wird Radiance Transport Equation, auch Radiative Transfer Equation genannt, kurz RTE [6].

Die durch ein teilnehmendes Medium gegebene Änderung der Strahlendichte an einem Punkt x in Richtung ω ist gegeben durch

$$\begin{aligned}\frac{\partial L(x, \omega)}{\partial x} &= -\sigma_a(x)L(x, \omega) + \sigma_a(x)L_{ve}(x, \omega) - \sigma_s(x)L(x, \omega) + \sigma_s(x)L_i(x, \omega) \\ &= -\sigma_t(x)L(x, \omega) + \sigma_a(x)L_{ve}(x, \omega) + \sigma_s(x)L_i(x, \omega).\end{aligned}\quad (18)$$

Hieraus lässt sich die Gleichung für $L(x, \omega)$ ableiten als

$$\begin{aligned}L(x_a, \omega) &= T_r(x_a, x_b)L_b(x_b, \omega) \\ &\quad + \int_{x_a}^{x_b} (T_r(x_a, x)\sigma_a(x)L_{ve}(x, \omega))dx \\ &\quad + \int_{x_a}^{x_b} (T_r(x_a, x)\sigma_s(x)L_i(x, \omega))dx \\ &= e^{-\tau(x_a, x_b)}L_b(x_b, \omega) \\ &\quad + \int_{x_a}^{x_b} e^{-\tau(x_a, x)}(\sigma_a(x)L_{ve}(x, \omega) + \sigma_s(x)L_i(x, \omega))dx\end{aligned}\quad (19)$$

Dabei beschreibt $L_b(x, \omega)$ eine lokale Beleuchtung eines Oberflächenpunktes x durch:

$$L_b(x, \omega) = I_0 \frac{e^{-\sigma_t * d(x)}}{d(x)^2} f_r(\omega', x, \omega) \cos^+ \theta'_x \quad (20)$$

Für das Licht $L_i(x, \omega)$, welches an einem Punkt x den eingestreuten Anteil wiedergibt gilt:

$$L_i(x, \omega) = I_0 \frac{e^{-\sigma_t * d(x)}}{d(x)^2} * p(\theta) \quad (21)$$

Die Funktion $d(x)$ gibt die Distanz zwischen dem Punkt x und der Lichtquelle wieder, I_0 beschreibt die Helligkeit bzw. Strahlungsintensität dieser Lichtquelle.

3 Streuungsmodelle

Das Rendern von teilnehmenden Medien zieht aufgrund der hohen Komplexität der Radiance Transport Equation einen enormen Rechenaufwand mit sich. Um interaktive Bildwiederholraten in Anwendungen zu produzieren, müssen vereinfachte Modelle in Betracht gezogen werden. Obwohl diese Modelle demzufolge nicht physikalisch exakt sind, sollten sie zumindest eine gute Annäherung an die tatsächliche Lösung der Radiance Transport Equation bieten.

Im folgenden werden die grundlegenden Streuungsmöglichkeiten von Licht an einer Oberfläche und einem teilnehmenden Medium aufgelistet und beschrieben.

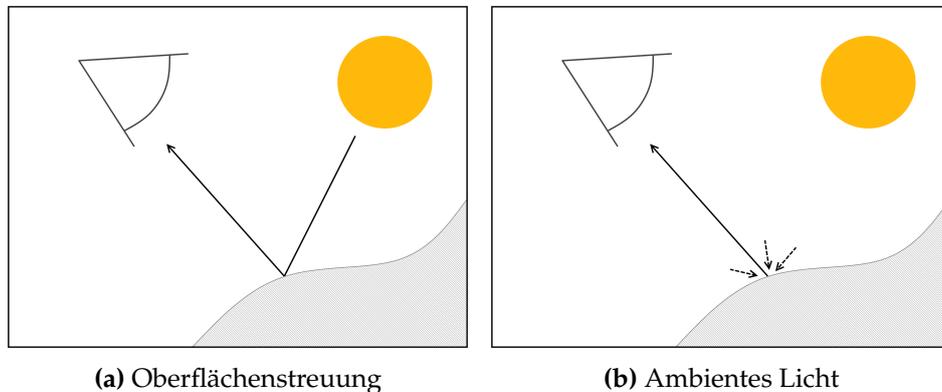


Abbildung 2: Bestandteile des Phong Modells

Das wohl bekannteste Beleuchtungsmodell ist das Phong Beleuchtungsmodell. Dieses entstammt der Klasse der lokalen Beleuchtungsmodelle, welche das teilnehmende Medium vollkommen außer Acht lassen. Phong Beleuchtung setzt sich aus einem diffusen und spekularen Anteil zusammen, welche sich entsprechend Abbildung 2a aus Ein- und Ausfallwinkel des Lichtes an der Oberfläche berechnen lassen. Zusätzlich wird ein ambienter Term entsprechend Abbildung 2b aufaddiert, um den Eindruck einer globalen Beleuchtung anzutäuschen.

Unter den Modellen, welche das teilnehmende Medium außer Acht lassen, findet sich das Modell, welches unabhängig von Lichtquellen, die Oberflächen von Objekten beleuchtet (Abbildung 3a). Diese Oberflächen besitzen somit ihre eigene Farbe, welche als einzige in das Endergebnis einfließt. Des weiteren befindet sich das globale Beleuchtungsmodell der Rendering Equation ebenfalls hier (Abbildung 3b).

Es gibt Modelle, welche für gewöhnlich separat von anderen Modellen betrachtet werden. Dazu zählen die Modelle, die direktes Licht (Abbildung 4a) von der Lichtquelle in die Bilder mit einbeziehen, sowie das Prinzip der Emission (Abbildung 4b). Direktes Licht von der Lichtquelle wird in

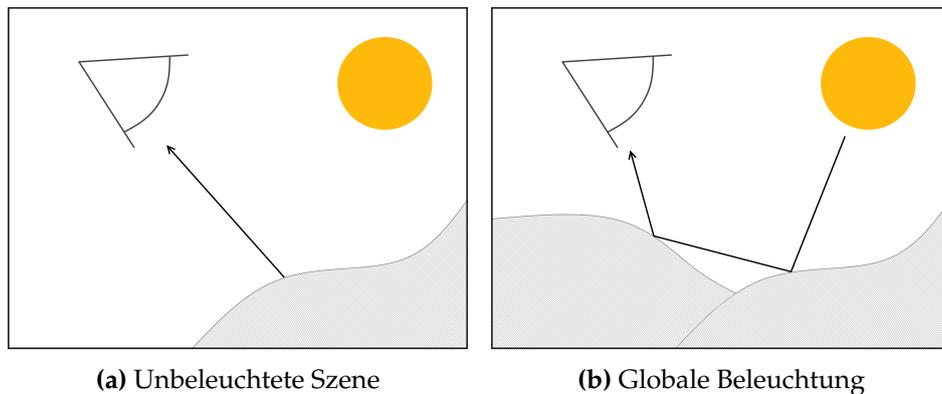


Abbildung 3: Beleuchtungsmodelle

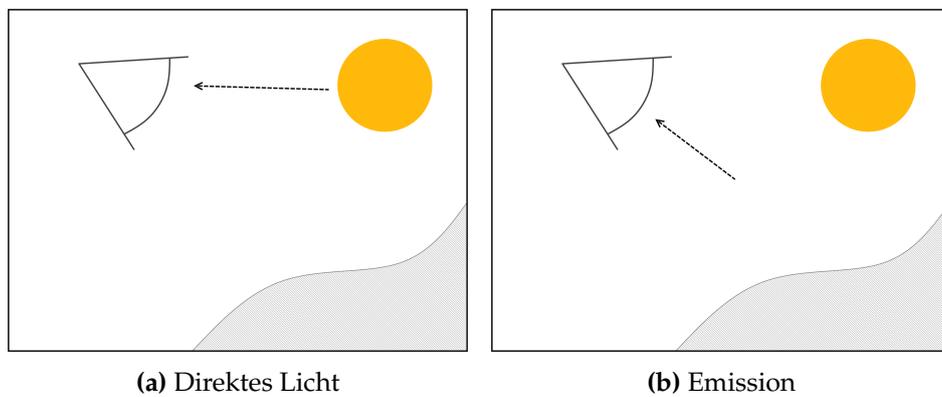


Abbildung 4: Separat behandeltes Licht

einfachen Modellen ignoriert, da die Lichtquelle selbst nicht als Oberflächenpunkt zählt und somit nicht gerendert wird. Schließlich werden nur gerenderte Objekte der Szene beleuchtet. Direktes Licht kann zu sichtbaren Effekten im Zusammenspiel mit der Kamera führen, wie bspw. Lens Flares. Da diese jedoch sehr speziell sind, werden sie oftmals in einem Postprocessing Schritt hinzugefügt und somit separat behandelt. Ein anderes Thema hingegen stellt die Emission dar. Diese ist abhängig von den Eigenschaften eines teilnehmenden Mediums, da sie aber für gewöhnlich zu schwach ist, um nach zusätzlichen Streuungen noch erkennbar einen Einfluss auf das Endergebnis zu haben, wird sie in den meisten Fällen gesondert gehandhabt.

Es folgt die erste Kategorie von Modellen, welche im Gegensatz zu den vorherigen Modellen, das teilnehmende Medium und dessen Eigenschaften in Betracht zieht und daraus die Beleuchtung von den dahinterliegenden Oberflächenpunkten verändert. Es handelt sich hierbei, um die Kategorie der Single Scattering Modelle (Abbildung 5). Sie zeichnen sich da-

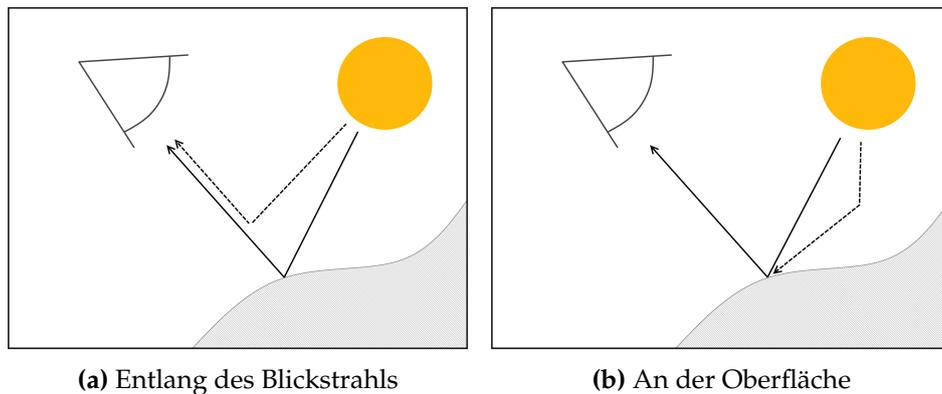


Abbildung 5: Single Scattering Modelle

durch aus, dass sie eine einzelne Streuung des Lichts in dem teilnehmenden Medium in die Beleuchtung mit einbeziehen. Sie machen somit einen Kompromiss zwischen dem Rechenaufwand und der physikalischen Korrektheit der RTE, sodass die grundlegenden Eigenschaften von teilnehmenden Medien gut erkennbar sind und die Echtzeitfähigkeit nicht zu stark beeinträchtigt wird.

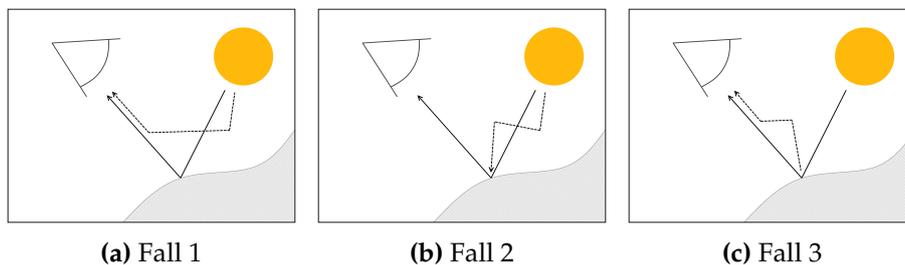


Abbildung 6: Multiple Scattering

Die letzte Kategorie von Modellen beinhaltet Multiple Scattering (Abbildung 6), welche der tatsächlichen Lösung der RTE am nächsten kommen. Allerdings sind dies ebenfalls die Modelle, welche am rechenaufwändigsten sind. Modelle für interaktive Anwendungen beschränken sich demzufolge meist auf das Single Scattering. Das heißt, dass nur eine einzige Streuung des Lichtes durch das teilnehmende Medium zwischen Lichtquelle und Oberfläche bzw. Kamera in Betracht gezogen wird. Diese Vereinfachung reduziert die RTE auf einen Term, welcher mit verschiedenen Methoden in angemessener Zeit lösbar ist. Im Allgemeinen ist dies auch nicht schlimm, da die Aspekte des Multiple Scatterings in der Regel nur bei einem hohen Wert für Albedo oder hoher optischer Dichte eine nennenswerte Rollen tragen.

Es sind über die Jahre viele Modelle entstanden, welche für ein spezifisches Problem, selbst nach Vereinfachungen und unter Vernachlässigung von physikalisch korrekter Streuung eine Lösung mit sehr geringer Fehlerabweichung von der RTE produzieren. Somit finden sich Dutzende von Verfahren zum Rendern von Licht in verschiedensten Medien, wie Staub, Rauch, Nebel, Wolken, Wasser und sogar der Atmosphäre von Planeten. Meist gelingt es diesen Verfahren, mittels eines Single Scattering Modells ein visuell ansprechendes Bild unter wenigen Rahmenbedingungen zu erstellen. Diese Rahmenbedingungen sind meistens die Forderung an die Homogenität des Mediums und in manchen, für gewöhnlich älteren Verfahren, die Position von Betrachter und/ oder Lichtquelle. Die nachfolgenden Abschnitte nehmen die Erfassung von Methoden für das Rendern von teilnehmenden Medien von Cerezo et al. [8] als Grundlage. Dabei begrenzen sich die genannten Methoden jedoch auf die relevantesten oder nennenswertesten Beispiele und wurden um aktuelle Methoden ergänzt.

3.1 Analytische Modelle

Vor mehr als 30 Jahren bestand bereits die Idee, den Einfluss von teilnehmenden Medien auf die Beleuchtung in Anwendungen mit einzubeziehen. Es fehlte zu dieser Zeit jedoch die notwendige Rechenleistung, um das Gleichungssystem der RTE selbst unter vereinfachten Bedingungen zu lösen. Aufgrund dieses Mangels bestand eine sehr hohe Nachfrage nach einer analytischen Lösung der RTE. Hierunter versteht sich die Reduktion der RTE auf eine Gleichung, welche ohne Abtasten von vielen Punkten entlang des Blickstrahls lösbar ist. Um die Gleichung jedoch auf einen analytisch lösbaren Term zu reduzieren müssen in der Regel einige starke Vereinfachungen über die Eigenschaften des teilnehmenden Mediums gemacht werden, sodass analytische Verfahren lediglich für einfache Szenarien geeignet sind. Ein großer Nachteil der rein analytischen Verfahren ist die vollständige Abwesenheit von Schattenwürfen in den teilnehmenden Medien. Somit werden in vielen modernen Verfahren analytische Erkenntnisse mit den Vorteilen der deterministischen Verfahren verknüpft.

Die erste Person, welche sich mit dem Einfluss von teilnehmenden Medien auf das visuelle Endergebnis beschäftigte war Blinn [9], welcher jedoch in seinen analytischen Lösungen für die RTE von einem im unendlichen platzierten Betrachter und Lichtquelle ausging. Obwohl seine Überlegung ebenfalls eine Vielzahl von anderen Themen deckte, stellen seine Ideen die ersten analytischen Gleichungen zu teilnehmenden Medien in der Computergrafik dar.

Erst mit der Arbeit von Sun et al. [10] wurden Gleichungen bekannt gemacht, welche die Lösung der RTE durch die Simplifikation des Terms lediglich unter der Bedingung von Homogenität des Mediums und isotroper Phasenfunktion erreichen. Hierbei entsteht zwar keine Gleichung, welche

mit üblichen Mitteln der Analysis lösbar ist, jedoch eine, dessen Lösung man numerisch approximieren und vorberechnen kann. Diese Gleichung wird im Laufe dieser Arbeit noch weiter im Detail erläutert.

Einige Jahre nach der Arbeit von Sun et al. wurde von Pegoraro und Parker [11] eine noch weiter geführte Form dieser analytischen Lösung vorgestellt. Durch ihre mathematischen Erkenntnisse gelang es ihnen die Bedingung der isotropen Phasenfunktion aufzuheben, durch welche analytische Verfahren in der Vergangenheit immer beschränkt waren.

3.2 Deterministische Modelle

Die deterministische Methode zur Lösung der RTE ist die flexibelste unter den Single Scattering Modellen, welche viele verschiedene Verfahren zusammenfasst. Im Allgemeinen versteht man unter diesen Ansätzen die Auswertung der RTE durch tatsächliches Auswerten der Gleichung an verschiedenen, systematisch gewählten Punkten im Raum. Aufgrund der Flexibilität der deterministischen Vorgehensweise wurden zahlreiche Ansätze entwickelt, welche in den verschiedensten Bereichen Anwendung finden.

Den Startpunkt der als deterministisch markierten Verfahren wurde von Kajiyama und Herzen gesetzt [12]. Sie bezogen teilnehmende Medien in das Raytracing mit ein, indem sie neue Algorithmen entwickelten, um Objekte in einem auf Voxeln basierendem Dichtesystem zu tracen. Hierzu zählten dementsprechend auch Wolken, Nebel und Partikel.

Max [13] stellte wenig später ein Scanline basiertes Verfahren vor, um die beleuchteten Regionen vor jeder gerenderten Oberfläche als Streuungskomponente mit einzubeziehen. Hierzu werden Informationen aus Shadowvolumen hinzugezogen, um Lichtschächte korrekt darzustellen.

Nishita et al. [14] greifen die Idee von Max auf und verwenden ebenfalls Shadow Volumes für die Bestimmung von beleuchteten und unbeleuchteten Regionen. Sie beschrieben die Beleuchtung durch das Medium als

$$L = e^{-\tau(x)}L_0 + \int_0^x \sigma_s e^{-\tau(z)}L_s(z)dz \quad (22)$$

welche heute besser unter dem Namen Airlight Formula oder Airlight Contribution bekannt ist. Ursprünglich wurde ihr Verfahren für gefärbte Spotlights genutzt, später wendeten sie es jedoch ebenfalls auf andere Medien an. Im Allgemeinen wurde ihr Modell für die Airlight Contribution als Grundlage für viele später folgende Verfahren verwendet.

Lecocq et al. [15] formulierten die Gleichungen von Nishita et al. neu und berechneten diese in Abhängigkeit von Winkeln anstatt Längen, um sie auf bewegende Lichtquellen einsetzen zu können. Durch die Neuformulierung ermöglichte es ihnen eine Vorabberechnung der Gleichung, durch welche sie zusätzlich eine starke Erhöhung der Performanz erreichten.

Biri et al. [16] nutzten in in ihrem Verfahren Shadow Volumes, um den Aufwand, welcher normalerweise durch Sampling besteht zu reduzieren. Durch die Vorberechnung des gestreuten Lichtes in Kombination mit dem Entfallen von Sampling formulierten sie somit ein besonders schnelles Verfahren, um volumetrische Schatten darzustellen.

Die genannten Verfahren repräsentieren nur einige wenige der vielen Verfahren, welche über die Jahre entwickelt wurden. Viele Verfahren, welche hier nicht genannt worden sind, sind ebenso in der Lage für ihr Anwendungsgebiet ein bestimmtes Problem zu lösen. Dabei produzieren sie sehr gute Ergebnisse für bspw. Turbulenzen in Rauch und Nebel von Stam und Fiume [17], oder die Berechnung von atmosphärischer Streuung von Klassen [18], Irwin [19], Dobashi et al. [20].

3.3 Stochastische Modelle

In der Regel unterscheiden sich die stochastischen Verfahren nicht besonders stark von den deterministischen Verfahren. Sie entsprechen vom Ablauf her grundsätzlich den deterministischen Verfahren, jedoch mit der Änderung, dass eine Form von stochastischen Sampling eingeführt wird, um die finite Anzahl an Punkten zu bestimmen, welche in die RTE einfließen.

Rushmeier [21] verwendet in ihrem Verfahren die sogenannte Zonal Method in Kombination mit Monte Carlo Methoden. Die Zonal Method erweitert das klassische Radiosity Verfahren auf das teilnehmende Medium durch die Verwendung von Voxeln. Durch die Kombination dieser zwei Methoden, gelingt es ihr im Detail bessere Bilder zu generieren, als es nur mit der Zonal Methode der Fall wäre.

Es können jedoch auch sehr einfache Verbesserungen, wie bspw. eine geschicktere Wahl der Sampling Punkte, einen stochastischen Ansatz darstellen. Kulla und Fajardo [22] verwenden in ihrem Verfahren eine Scanline basierte deterministische Methode, um den gestreuten Lichtanteil zu errechnen, formulieren jedoch die Sampling Punkte nicht anhand von Abständen sondern anhand von Winkeln, mit einem Resultat, dass die Verteilung der Punkte in relevanteren Regionen konzentriert ist. So gelingt es ihnen mit einer geringeren Gesamtzahl von Punkten ein besseres Endergebnis zu erzielen.

3.4 Multiple Scattering

Während die meisten der bisher genannten Verfahren sich auf Single Scattering beschränken, waren wenige von diesen in der Lage Rücksicht auf die Effekte des Multiple Scatterings zu nehmen. Ohne diese Betrachtung verlieren die Single Scattering Verfahren an Genauigkeit und weichen von der eigentlichen Lösung der RTE ab. Multiple Scattering Verfahren sind

hingegen in der Lage durch Mehraufwand bei der Berechnung der Beleuchtung detailreichere und physikalisch genauere Bilder zu generieren. Dieser Mehraufwand erfordert jedoch erheblich mehr Rechenleistung als ein Single Scattering Verfahren, sodass sich die Effekte des Multiple Scatterings aufgrund des geringen Anwendungsbereiches in erster Hinsicht nicht für die meisten Probleme lohnen. Die Effekte werden in erster Hinsicht nur sichtbar, falls es sich um ein optisch besonders dichtes Medium handelt oder der Wert für den Albedo des Mediums besonders hoch ist. Medien, welche diese Eigenschaften besitzen, sind bspw. Flüssigkeiten, wie Milch oder besonders dichter Nebel. Auch für das korrekte Visualisieren von Wolken müssen Multiple Scattering Effekte berechnet werden. Es haben sich einige verschiedene Ansätze zur Verringerung dieses enormen Rechenaufwands gebildet.

Ein Ansatz zur Lösung von Multiple Scattering sind Progressive Refinement Ansätze, in denen ein bestehendes Ergebnis immer weiter anhand des aktuellen Zustandes verfeinert und somit verbessert wird. Eines der bereits genannten Verfahren ist die Zonal Method welche Rushmeier [21] in Kombination mit Monte Carlo Methoden verwendet. Die Zonal Method erweitert das Prinzip des Radiosity Verfahrens auf Voxel und zieht somit das teilnehmende Medium mit in Betracht.

Premoze und Ashikhmin [23] stellten eine rechengünstige Methode vor, um mit Point Spread Functions die Effekte von Multiple Scattering einzufangen. Anhand von mathematischen Funktionen berechneten sie einen statistischen Anteil des durch mögliche Multiple Scattering Pfade einkommenden Lichts, um eine gute Approximation der Multiple Scattering Beleuchtung zu erhalten.

Erweiterte Verfahren des Bidirectional Path tracing nach Lafortune and Willems [24] oder auch Metropolis Light Transport nach Pauly et al. [25] sind ebenfalls in der Lage die Streuungseffekte einzufangen.

Virtual Point Lights nach Keller [26] sind eine weitere Methode, um Multiple Scattering Effekte darzustellen. Sie sind ähnlich zu dem Prinzip des Photon Mappings nach Jensen and Christensen [27]. So werden von der Lichtquelle aus Lichtstrahlen verschossen, welche innerhalb des Mediums streuen. Es werden nun eine finite Anzahl an Punkten generiert, an welche man sogenannte Virtual Point Lights setzt. Nach diesem Vorabschritt beginnt dann der eigentliche Rednerprozess, in welchem mit üblichen Methoden der Streuungsanteil entlang eines Blickstrahls von allen Virtual Point Lights berechnet wird. Virtual Ray Lights Novák et al. [28] erweitern das Prinzip der Virtual Point Lights.

3.5 Fake Media

Neben den auf physikalischen Prinzipien basierenden Verfahren, welche in den vorausgehenden Abschnitten genannt wurden, sind über die Jahre eben-

falls Verfahren und Algorithmen entstanden, welche ihren gesamten Fokus auf die Echtzeitfähigkeit, sowie visuell ansprechende Ergebnisse legen. Diese lassen oftmals die physikalischen Grundlagen teils oder sogar ganz außen vor. Einige dieser Modelle modellieren das Medium und berechnen deren Beleuchtung basierend auf Texturen und Alpha Blending, ignorieren jedoch deren Interaktion mit dem Rest der Umgebung. Andere Verfahren hingegen versuchen mit den verfügbaren Mitteln eine Interaktion so realistisch wie möglich vorzutäuschen.

Yeager [29] stellte unter anderem aus Mitteln des Partikel Renderings und Texture Mappings ein System zusammen, welches in der Lage ist, eine realistische Simulation der Atmosphäre des Jupiter zu rendern. Premoze und Ashikhmin [30] formulierten ein Modell, welches für verschiedene Parameter die Färbung und Beleuchtung von Gewässern visualisiert. Dabei reichen die Anwendungsbereiche ihres Verfahrens von Ozeanen über klare Seen bis zu stark getrübbten Küstengewässern. Biri et al. [31] stellten eine Methode vor, um Nebel anhand dessen Transmission in dem Kontext der Echtzeitfähigkeit zu visualisieren. Es gibt noch einige weitere Verfahren, welche jedoch nicht besonders nennenswert sind.

4 Implementation

Im vorherigen Kapitel wurden eine Reihe von Verfahren genannt, welche für die Visualisierung von volumetrischer Beleuchtung verwendet werden können. Dies stellt keine vollständige Liste aller Verfahren dar, sondern schneidet lediglich die relevantesten oder historisch ersten Vorkommnisse dieser Modelle aus.

Ziel der Arbeit ist es Verfahren der volumetrischen Beleuchtung auszuwählen und diese miteinander zu vergleichen und auszuwerten. Im Rahmen dieser Arbeit stellt sich nun die Frage, welche Modelle für einen Vergleich am besten geeignet wären. Aufgrund des technischen Aufwandes der Multiple Scattering Verfahren, werden nur Verfahren des Single Scatterings in Betracht gezogen.

Im Bereich der Single Scattering Verfahren gibt es eine Vielzahl von Modellen, welche auf bestimmte Szenarien und teilnehmende Medien abgestimmt sind und welche unter den korrekten Voraussetzungen hervorragende Bilder produzieren können. Interessant sind jedoch die Verfahren, welche auf ein breites Spektrum von Problemen angewendet werden können oder als Grundlage für Erweiterungen und Modifikationen dienen. Auch besonders effiziente Verfahren sind einen näheren Blick Wert, da diese vermutlich mit den größten Einschränkungen einher gehen.

Im folgenden werden das sogenannte Raymarching, ein analytisches Modell, sowie ein Modell, welches Shadow Volumes verwendet, erläutert. Hierbei wird das allgemeine Prinzip dieser Modelle erklärt und wie sie in OpenGL implementiert werden können.

Die Implementation dieser Verfahren baut auf der CVK2 Bibliothek der Computervisualistik der Universität Koblenz auf. Die Bibliothek wurde an manchen Stellen leicht verändert oder um relevante Klassen erweitert, um die Arbeit zu erleichtern. Die grundlegenden Abläufe der folgenden Implementationen sind sehr ähnlich zueinander. Zu Beginn der Anwendung wird OpenGL initialisiert. Dies beinhaltet das Erstellen eines Fensters, sowie das festlegen von Keybinds, um die Parameter für die volumetrische Beleuchtung zu verändern. Anschließend werden die Kamera und die Lichtquelle initialisiert und die Szene mit den dazugehörigen Materialien wird an einen globalen Szenenknoten angehängt. Anschließend werden die Shader geladen und Frame Buffer Objects generiert. In der Render-Schleife werden mehrere Schritte abgearbeitet. Zuerst wird die Szene über den Szenenknoten in das erste FBO gerendert. Hierbei gibt es separate Texturen für die Position, Normale und Farbe der Szene. Außerdem wird eine ShadowMap angelegt, welche später benötigt wird. Danach wird der zweite Shader gebunden, welcher für die Beleuchtung zuständig ist. Er rendert in ein weiteres FBO die beleuchtete Szene, als auch zu Veranschaulichungszwecken die separate Beleuchtung. Letzlich wird mittels eines dritten Sha-

ders eine ausgewählte Textur aus den FBOs, wie bspw. die beleuchtete Szene in das Fenster gerendert.

4.1 Raymarching

Raymarching beschreibt ein sehr einfaches Prinzip, volumetrische Beleuchtung zu berechnen, welches gleichzeitig eine Vielzahl an Möglichkeiten ermöglicht und nach Bedarf an bestimmte Probleme angepasst werden kann. Dieses Verfahren zählt zu den deterministischen Single Scattering Verfahren, kann jedoch durch Modifikationen auch zu einem stochastischen Verfahren erweitert werden. Raymarching entspricht der diskreten Implementation der Airlight Contribution, wodurch das Verfahren typischerweise mit hohem Rechnaufwand verbunden ist.

4.1.1 Grundlegendes Prinzip

Die Oberflächenbeleuchtung dieses Modells ergibt sich aus dem gedämpften Anteil des Lichtes, welches von der Lichtquelle auf die Oberfläche trifft. Hierbei wird die Strahldichte durch Absorption und Out-Scattering verringert. Diese Oberflächenbeleuchtung wird als Hintergrund verwendet auf den das Licht des In-Scatterings aufaddiert wird.

Die RTE legt folgende Gleichung für $L(x, \omega)$ fest:

$$L(x_a, \omega) = e^{-\tau(x_a, x_b)} L_b(x_b, \omega) + \int_{x_a}^{x_b} e^{-\tau(x_a, x)} (\sigma_a(x) L_{ve}(x, \omega) + \sigma_s(x) L_i(x, \omega)) dx \quad (23)$$

Dabei beschreibt $L_b(x, \omega)$ eine lokale Beleuchtung eines Oberflächenpunktes x durch:

$$L_b(x, \omega) = I_0 \frac{e^{-\sigma_t * d(x)}}{d(x)^2} f_r(\omega', x, \omega) \cos^+ \theta'_x \quad (24)$$

Für das Licht $L_i(x, \omega)$, welches an einem Punkt x den eingestreuten Anteil wiedergibt gilt:

$$L_i(x, \omega) = I_0 \frac{e^{-\sigma_t * d(x)}}{d(x)^2} * p(\theta) \quad (25)$$

Raymarching ist ein Begriff der verwendet wird, um zu beschreiben, dass ein Lichtstrahl in Intervallen abgetastet wird. In diesem Fall wird ein Strahl von der Kamera zu dem gerenderten Punkt gebildet. Entlang dieses Blickstrahls werden in wählbaren Intervallen Punkte gewählt, welche für die Berechnung der Beleuchtung verwendet werden.

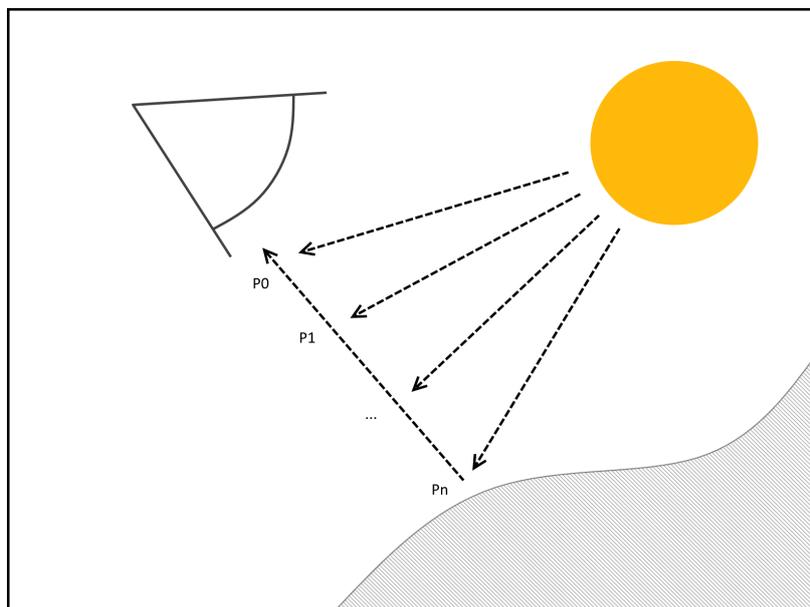


Abbildung 7: Raymarching

In der Regel wird keine Emission betrachtet. Das heißt, dass in der Gleichung 23 folgende Veränderungen stattfinden:

$$L(x_a, \omega) = e^{-\tau(x_a, x_b)} L_b(x_b, \omega) + \sum_P e^{-\tau(x_a, p_i)} \sigma_s(p_i) L_i(p_i, \omega) \quad (26)$$

P umfasst die Menge aller abzutastenden Punkte.

4.1.2 Umsetzung in OpenGL

Initialisierung Mittles Deferred Shading werden die Informationen der Szene dem Shader zur Verfügung gestellt. Dieser erhält Texturen im Kamerakoordinatensystem mit Position, Normale und Farbe. Zusätzlich sind dem Shader über Uniform Variablen die ViewMatrix, die Position des Lichtes und dessen Farbe, sowie Near und Far Werte der Kamera bekannt. Weitere Uniform Variablen bestimmen die Werte für I_0 , σ_s , σ_t und die Anzahl an Schritten mit denen die Strahlen abgetastet werden sollen. Zuletzt wird eine ShadowMap benötigt, aus welcher sich bestimmen lässt, ob ein betrachteter Punkt beleuchtet ist oder nicht.

Fragment Shader Der abzutastende Strahl wird anhand der Position der Camera und der Position der gerenderten Szene gebildet. Da sich die Position der gerenderten Szene bereits im Kamerakoordinatensystem gegeben

ist, liegt die Position der Kamera selbst im Ursprung. Nun wird eine Funktion aufgerufen welche anhand von Start und Endpunkt den gestreuten Anteil des Lichtes bestimmt.

raymarchViewSpaceAirlight Die Funktion `raymarchViewSpaceAirlight` berechnet linear anhand des Start- und Endpunktes und der Anzahl an Iterationen, die Punkte, welche für die Streuung betrachtet werden sollen. Für diese Punkte, welche im Medium liegen, gibt es zwei mögliche Szenarien. Sie können in einer beleuchteten oder einer unbeleuchteten Region liegen. Sollten sie in einer unbeleuchteten Region liegen, so sind sie nicht relevant und können ignoriert werden. Ist die Lichtquelle jedoch von dem Punkt aus sichtbar, so muss für diesen Punkt der Anteil des gestreuten Lichtes berechnet werden und zu der Summe hinzugefügt werden.

Algorithm 1 `raymarchViewSpaceAirlight(vec3 a, vec3 b, int iterations)`

```
vec3 step = (b - a) / (iterations + 1);
float sum = 0;
for i = 1; i < iterations; i ++ do
    vec3 point = a + i * step;
    if point is lit then
        sum += getInScattering(point) * length(step);
    else
        do nothing;
return sum;
```

Die Funktion `getInScattering` verwendet die Formel 26 zur Bestimmung des gestreuten Lichtes. Das gestreute Licht kann anschließend mit der Oberflächenbeleuchtung zusammengerechnet werden, um das Endresultat zu erzeugen.

4.2 Analytisches Modell

Das analytische Verfahren baut auf mathematischen Erkenntnissen und Vereinfachungen aus der RTE auf. Die neueren Verfahren schaffen es vereinfachte Formen der RTE auf einen analytisch geschlossenen Term herunter zu brechen, welcher sogar Rücksicht auf verschiedene Phasenfunktionen nehmen kann. Aufgrund der mathematischen Komplexität, wird hier jedoch lediglich ein etwas älteres Verfahren erläutert, welches zwar eine Funktion vorab berechnen und in einer Lookuptabelle speichern muss, dafür aber verständlicher und einfacher nachvollziehbar ist.

4.2.1 Formulierung eines zu verarbeitenden Terms aus der RTE

Aus der RTE ist folgende Gleichung unter der Vernachlässigung von Emission bekannt:

$$L(x_a, \omega) = e^{-\tau(x_a, x_b)} L_b(x_b, \omega) + \int_{x_a}^{x_b} e^{-\tau(x_a, x)} \sigma_s(x) L_i(x, \omega) dx \quad (27)$$

$$L_b(x, \omega) = I_0 \frac{e^{-\sigma_t * d(x)}}{d(x)^2} f_r(\omega', x, \omega) \cos^+ \theta'_x \quad (28)$$

$$L_i(x, \omega) = I_0 \frac{e^{-\sigma_t * d(x)}}{d(x)^2} p(\theta) \quad (29)$$

$$\tau(x_a, x_b) = \int_{x_a}^{x_b} \sigma_t(z) dz \quad (30)$$

Um aus der RTE einen analytisch lösbaren Term aufzustellen, werden einige vereinfachende Bedingungen gestellt. Offensichtlich fallen für σ_a , σ_s und σ_t allerlei Abhängigkeiten von der Position weg. Das heißt, dass das Medium homogen im Raum verteilt sein muss. Außerdem wird die isotrope Phasenfunktion p_I verwendet. Es muss also der folgende Term in eine analytisch lösbare Form gebracht werden.

$$L(x_a, \omega) = e^{-\sigma_t(x_b - x_a)} L_b(x_b, \omega) + \int_{x_a}^{x_b} e^{-\sigma_t(x - x_a)} \sigma_s I_0 \frac{e^{-\sigma_t * d(x)}}{d(x)^2} \frac{1}{4\pi} dx \quad (31)$$

Die Funktion $d(x)$ beschreibt den Abstand des Punktes x zu der Lichtquelle. Die Gleichung setzt sich aus der durch Absorption und Out-Scattering gedämpften Beleuchtung der Oberfläche und dem entlang des Blickstrahl akkumulierten In-Scatterings zusammen. Für das In-Scattering wird ein Single Scattering Ansatz verwendet. Interessant ist für die Vereinfachung nur der zweite Teil:

$$\int_{x_a}^{x_b} e^{-\sigma_t(x - x_a)} \sigma_s I_0 \frac{e^{-\sigma_t * d(x)}}{d(x)^2} \frac{1}{4\pi} dx \quad (32)$$

4.2.2 Herleitung eines analytisch vereinfachten Terms

Beschrieben von Sun et al. [10] setzt sich der Term für das In-Scattering aus einem direktem Transmissions-Term und einem Single Scattering Term zusammen. Interessant ist für eine analytische Herleitung der Single Scattering Term beschrieben durch

$$L_s(\gamma, d_{lv}, d_{vp}, \sigma_t, \sigma_s) = \int_0^{d_{vp}} \sigma_s p(\theta) \frac{I_0 * e^{-\sigma_t * d}}{d^2} e^{-\sigma_t x} dx \quad (33)$$

Diese Formel gleicht Formel 32, wobei der Punkt x_a , welcher dem Startpunkt für das Abtasten entspricht im Ursprung, also in der Kamera liegt. Die Vereinfachung dieser Gleichung nach Sun et al. wird im folgenden ausführlich aufgeführt. Eine stark gekürzte Fassung findet sich in [10].

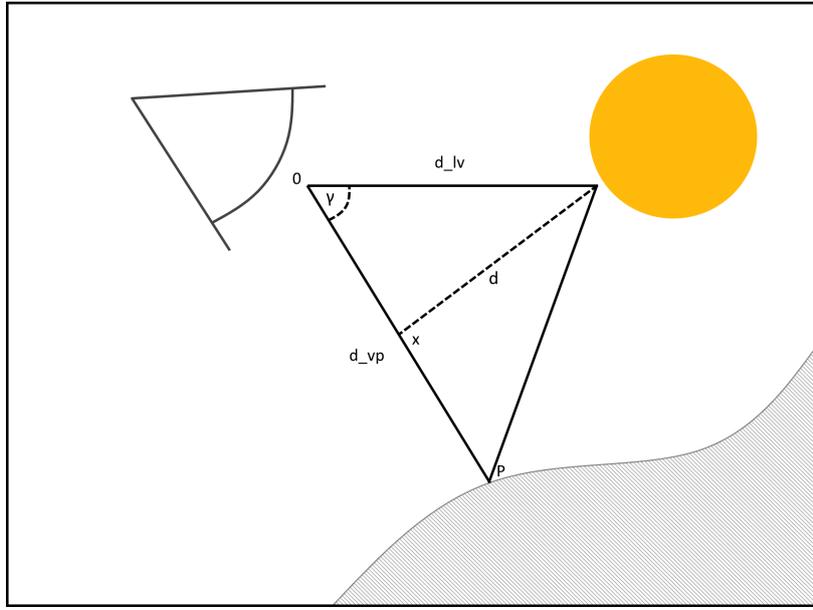


Abbildung 8: Skizze der Ausrichtungen

Durch den Kosinussatz $c^2 = a^2 + b^2 - 2ab \cos \gamma$ lässt sich der Abstand d zur Lichtquelle anders ausdrücken.

$$L_s(\gamma, d_{lv}, d_{vp}, \sigma_t, \sigma_s) = \frac{\sigma_s * I_0}{4\pi} * \int_0^{d_{vp}} \frac{e^{-\sigma_t * \sqrt{d_{lv}^2 + x^2 - 2x d_{lv} * \cos \gamma}}}{d_{lv}^2 + x^2 - 2x d_{lv} * \cos \gamma} * e^{-\sigma_t * x} dx \quad (34)$$

Als Erinnerung hier die Gleichung für Integration durch Substitution:

$$\int_a^b f(\varphi(u)) du = \int_{\varphi(a)}^{\varphi(b)} f(x) * \frac{1}{\varphi'(u)} dx \quad (35)$$

Es wird $T_{lv} = \sigma_t * d_{lv}$ und $T_{vp} = \sigma_t * d_{vp}$ definiert. Daraus lassen sich folgende Umformungen durchführen.

$$\begin{aligned} L_s(\gamma, d_{lv}, d_{vp}, \sigma_t, \sigma_s) &= \frac{\sigma_s * I_0}{4\pi} * \int_0^{d_{vp}} \frac{e^{-\sigma_t * \sqrt{d_{lv}^2 + x^2 - 2x d_{lv} * \cos \gamma}}}{d_{lv}^2 + x^2 - 2x d_{lv} * \cos \gamma} * e^{-\sigma_t * x} dx \\ &= \frac{\sigma_s * I_0}{4\pi} * \int_0^{d_{vp}} \frac{\sigma_t^2 e^{-\sqrt{\sigma_t^2 * (d_{lv}^2 + x^2 - 2x d_{lv} * \cos \gamma)}}}{\sigma_t^2 d_{lv}^2 + x^2 - 2x d_{lv} * \cos \gamma} * e^{-\sigma_t * x} dx \end{aligned}$$

Substitution: $t = \sigma_t * x$

$$\begin{aligned}
&= \frac{\sigma_s * I_0}{4\pi} * \int_0^{T_{vp}} \sigma_t^2 * \frac{e^{-\sqrt{T_{lv}^2 + t^2 - 2tT_{lv} * \cos \gamma}}}{T_{lv}^2 + t^2 - 2tT_{lv} * \cos \gamma} * e^{-t} * \frac{1}{\sigma_t} dt \\
&= \frac{\sigma_s * I_0}{4\pi} * \sigma_t * \int_0^{T_{vp}} \frac{e^{-\sqrt{T_{lv}^2 + t^2 - 2tT_{lv} * \cos \gamma}}}{T_{lv}^2 + t^2 - 2tT_{lv} * \cos \gamma} * e^{-t} dt \\
&= \frac{\sigma_s * I_0}{4\pi} * \sigma_t * \int_0^{T_{vp}} \frac{e^{-\sqrt{T_{lv}^2 (\cos^2 \gamma + \sin^2 \gamma) + t^2 - 2tT_{lv} * \cos \gamma}}}{T_{lv}^2 (\cos^2 \gamma + \sin^2 \gamma) + t^2 - 2tT_{lv} * \cos \gamma} * e^{-t} dt \\
&= \frac{\sigma_s * I_0}{4\pi} * \sigma_t * \int_0^{T_{vp}} \frac{e^{-\sqrt{(t - T_{lv} \cos \gamma)^2 + T_{lv}^2 \sin^2 \gamma}}}{(t - T_{lv} \cos \gamma)^2 + T_{lv}^2 \sin^2 \gamma} * e^{-t} dt
\end{aligned}$$

Substitution: $z = t - T_{lv} \cos \gamma$

$$\begin{aligned}
&= \frac{\sigma_s * I_0}{4\pi} * \sigma_t * \int_{-T_{lv} \cos \gamma}^{T_{vp} - T_{lv} \cos \gamma} \frac{e^{-\sqrt{z^2 + T_{lv}^2 \sin^2 \gamma}}}{z^2 + T_{lv}^2 \sin^2 \gamma} * e^{-(z + T_{lv} \cos \gamma)} dz \\
&= \frac{\sigma_s * I_0}{4\pi} * \sigma_t * e^{-T_{lv} \cos \gamma} * \int_{-T_{lv} \cos \gamma}^{T_{vp} - T_{lv} \cos \gamma} \frac{e^{-z - \sqrt{z^2 + T_{lv}^2 \sin^2 \gamma}}}{z^2 + T_{lv}^2 \sin^2 \gamma} dz
\end{aligned}$$

Substitution: $\eta = \arctan \frac{z}{T_{lv} \sin \gamma}$ bzw. $z = T_{lv} \sin \gamma \tan \eta$

Beachte: $\frac{\partial}{\partial z} \arctan \frac{z}{T_{lv} \sin \gamma} = \frac{T_{lv} \sin \gamma}{z^2 + T_{lv}^2 \sin^2 \gamma}$

$$\begin{aligned}
&= \frac{\sigma_s I_0 \sigma_t e^{-T_{lv} \cos \gamma}}{4\pi} \\
&\quad * \int_{\arctan \frac{-T_{lv} \cos \gamma}{T_{lv} \sin \gamma}}^{\arctan \frac{T_{vp} - T_{lv} \cos \gamma}{T_{lv} \sin \gamma}} \frac{e^{-(T_{lv} \sin \gamma \tan \eta) - \sqrt{(T_{lv} \sin \gamma \tan \eta)^2 + T_{lv}^2 \sin^2 \gamma}}}{z^2 + T_{lv}^2 \sin^2 \gamma} \frac{1}{\left(\frac{T_{lv} \sin \gamma}{z^2 + T_{lv}^2 \sin^2 \gamma}\right)} d\eta \\
&= \frac{\sigma_s I_0 \sigma_t e^{-T_{lv} \cos \gamma}}{4\pi} \int_{\gamma - \frac{\pi}{2}}^{\arctan \frac{T_{vp} - T_{lv} \cos \gamma}{T_{lv} \sin \gamma}} \frac{e^{-(T_{lv} \sin \gamma \tan \eta) - T_{lv} \sin \gamma \sqrt{\tan^2 \eta + 1}}}{T_{lv} \sin \gamma} d\eta \\
&= \frac{\sigma_s I_0 \sigma_t e^{-T_{lv} \cos \gamma}}{4\pi} \int_{\gamma - \frac{\pi}{2}}^{\arctan \frac{T_{vp} - T_{lv} \cos \gamma}{T_{lv} \sin \gamma}} \frac{e^{-(T_{lv} \sin \gamma \tan \eta) - T_{lv} \sin \gamma \frac{1}{\cos \eta}}}{T_{lv} \sin \gamma} d\eta \\
&= \frac{\sigma_s I_0 \sigma_t e^{-T_{lv} \cos \gamma}}{4\pi} \int_{\gamma - \frac{\pi}{2}}^{\arctan \frac{T_{vp} - T_{lv} \cos \gamma}{T_{lv} \sin \gamma}} \frac{e^{-T_{lv} \sin \gamma (\tan \eta + \frac{1}{\cos \eta})}}{T_{lv} \sin \gamma} d\eta \\
&= \frac{\sigma_s I_0 \sigma_t e^{-T_{lv} \cos \gamma}}{4\pi T_{lv} \sin \gamma} \int_{\gamma - \frac{\pi}{2}}^{\arctan \frac{T_{vp} - T_{lv} \cos \gamma}{T_{lv} \sin \gamma}} e^{-T_{lv} \sin \gamma \frac{1 + \sin \eta}{\cos \eta}} d\eta
\end{aligned}$$

Substitution: $\xi = \frac{\eta + \frac{\pi}{2}}{2}$

$$\begin{aligned}
&= \frac{\sigma_s I_0 \sigma_t e^{-T_{lv} \cos \gamma}}{4\pi T_{lv} \sin \gamma} \int_{\frac{\gamma}{2}}^{\frac{\pi}{4} + \frac{1}{2} \arctan \frac{T_{vp} - T_{lv} \cos \gamma}{T_{lv} \sin \gamma}} e^{-T_{lv} \sin \gamma \frac{1 + \sin(2\xi + \frac{\pi}{2})}{\cos(2\xi + \frac{\pi}{2})}} * 2d\xi \\
&= \frac{\sigma_s I_0 \sigma_t e^{-T_{lv} \cos \gamma}}{2\pi T_{lv} \sin \gamma} \int_{\frac{\gamma}{2}}^{\frac{\pi}{4} + \frac{1}{2} \arctan \frac{T_{vp} - T_{lv} \cos \gamma}{T_{lv} \sin \gamma}} e^{-T_{lv} \sin \gamma \tan \xi} d\xi \\
&= \frac{\sigma_s I_0 \sigma_t e^{-T_{lv} \cos \gamma}}{2\pi T_{lv} \sin \gamma} \int_{\frac{\gamma}{2}}^{\frac{\pi}{4} + \frac{1}{2} \arctan \frac{T_{vp} - T_{lv} \cos \gamma}{T_{lv} \sin \gamma}} \exp(-T_{lv} \sin \gamma \tan \xi) d\xi \\
&= A_0 * \int_{\frac{\gamma}{2}}^{A_2} \exp(A_1 \tan \xi) d\xi \\
&= A_0 * (F(A_1, A_2) - F(A_1, \frac{\gamma}{2}))
\end{aligned} \tag{36}$$

Hierbei sind A_0, A_1, A_2 und $F(u, v)$ gegeben durch:

$$A_0 = \frac{\sigma_s I_0 e^{-\sigma_t d_{lv} \cos \gamma}}{2\pi d_{lv} \sin \gamma} \tag{37}$$

$$A_1 = \sigma_t d_{lv} \sin \gamma \tag{38}$$

$$A_2 = \frac{\pi}{4} + \frac{1}{2} \arctan \frac{\sigma_t d_{vp} - \sigma_t d_{lv} \cos \gamma}{\sigma_t d_{lv} \sin \gamma} \tag{39}$$

$$F(u, v) = \int_0^v \exp(-u \tan \xi) d\xi \tag{40}$$

Es sollte angemerkt werden, dass sich die Funktion

$$F(u, v) = e^{-u * \tan v}$$

theoretisch integrieren lässt, da aber

$$\int e^{-u \tan x} dx = \frac{1}{2} i e^{2iu} (Ei(-u(\tan x + i)) - Ei(-u(\tan x - i))) + C$$

mit der Integraleponentialfunktion

$$Ei(x) = - \int_{-x}^{\infty} \frac{e^{-t}}{t} dt$$

sich nicht einfacher handhaben lässt als vorher auch, ist eine berechnete Methode die Funktion $F(u, v)$ als LookUpTabelle vorzuberechnen und in einer Textur an den Shader weiterzugeben. Tatsächlich verwenden Pegoraro und Parker [11] diese Integraleponentialfunktion, um weitere Vereinfachungen durchzuführen.

4.2.3 Umsetzung in OpenGL

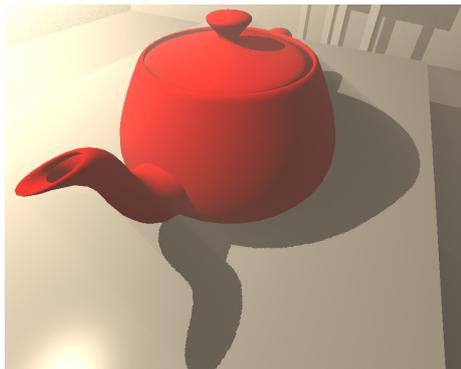
Nun gilt es die mathematischen Erkenntnisse des letzten Abschnittes in die Anwendung mit einzubinden. Hierzu muss zu Beginn der Anwendung eine Textur generiert werden, welche die Funktion 40 anhand der x und y Koordinate der Textur die korrekten Werte für u und v wählt und entsprechend vorberechnet. Diese Textur wird mit den restlichen Uniform Variablen an den Shader gegeben. Der Shader muss daraufhin nichts weiter tun als die Funktion 36 auswerten und auf die Hintergrundbeleuchtung des Oberflächenpunktes aufaddieren.

4.3 Shadow Volumes

Shadow Volumes werden verwendet, um den Rechenaufwand für interaktive Anwendungen zu verringern. Je nach Anwendung und Implementation können sie den Aufwand gegenüber eines gewöhnlichen Raymarching Verfahrens erheblich verringern und ressourcenschonend ebenso gute, wenn nicht sogar genauere Bilder produzieren.

4.3.1 Generieren der Shadow Volumes

Shadow Volumes helfen zwar bei der Berechnung von volumetrischer Beleuchtung, müssen jedoch vorher generiert werden, wodurch in vielen Fällen ein erheblicher Aufwand entstehen kann. Um nicht wieder an Geschwindigkeit zu verlieren, welche ansonsten durch die Verwendung von Shadow Volumes gewonnen wird, muss eine Methode gefunden werden, welche es erlaubt einfache Shadow Volumes für beliebige Objekte zu erstellen.



(a) Beleuchteter Teapot



(b) Shadow Volume des Teapots

Abbildung 9: Darstellung der Shadow Volumes

Es bietet sich das Verfahren von McCool an [32], welches anhand von einer ShadowMap die Tiefenwerte eines Terrains manipuliert. Da diese Aufgabe sehr einfach einem Compute Shader übergeben werden kann, ist die-

ses Verfahren in der Lage mit einem angemessenen aufgelösten Terrain eine gute Annäherung an die tatsächlichen Shadow Volumes zu erstellen, welche nun jedoch in Echtzeit neu berechnet werden können.

4.3.2 Reduktion des abzutastenden Bereiches

Der gesamte Sinn und Zweck von Shadow Volumes ist es die Anzahl der Sampling Punkte zu reduzieren und gleichzeitig bessere Schattenkonturen zu erhalten. Hierzu muss verstanden werden, welche Fälle für die Anordnung der Shadow Volumes und Szene eintreten können.

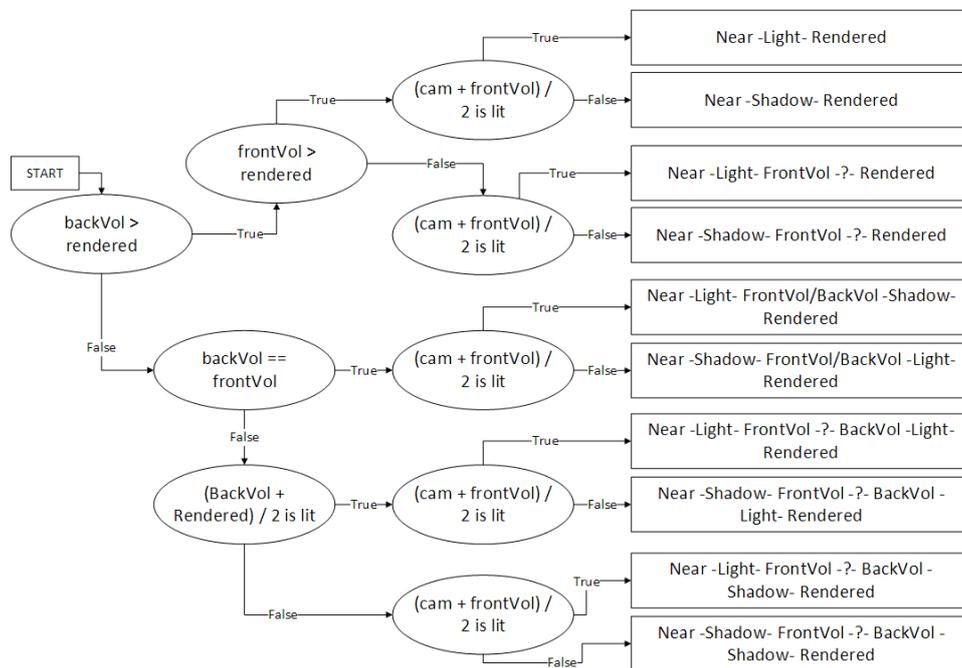


Abbildung 10: Shadow Volume Flow

Abbildung 10 zeigt die möglichen Anordnungen der Shadow Volumes in einer Szene. Dabei werden durch Vergleiche von Tiefenwerten die vorderste und hinterste Fläche des Shadow Volumes in Relation zur gerenderten Oberfläche der Szene gesetzt. Dabei wird zusätzlich bestimmt, welche der Zwischenregionen schattiert oder beleuchtet sind oder aufgrund von möglicher Mehrfachverschattung im Bereich zwischen vorderster und hinterster Fläche des Shadow Volumes eine Mischung von Schatten und Licht besitzen könnten.

4.3.3 Umsetzung in OpenGL

Für die Umsetzung dieses Verfahrens müssen vor der Berechnung der Beleuchtung im Shader einige zusätzliche Vorbereitungen getroffen werden. Dies beinhaltet vor allem das Generieren und Berechnen der Shadow Volumes sowie einen weiteren Zwischenschritt, um die Tiefenwerte der vordersten und hintersten Fläche in weitere FBOs zu rendern. Diese Tiefentexturen werden zusätzlich an den Shader gegeben, welcher dann gemäß Abbildung 10 die beleuchteten Regionen findet. Nun lassen sich genauso wie für das Raymarching Start- und Endpunkte finden, an denen nach Gleichung 26 der Anteil des gestreuten Lichtes in der Region berechnet werden kann. Dieser wird anschließend auf die Oberflächenbeleuchtung aufaddiert.

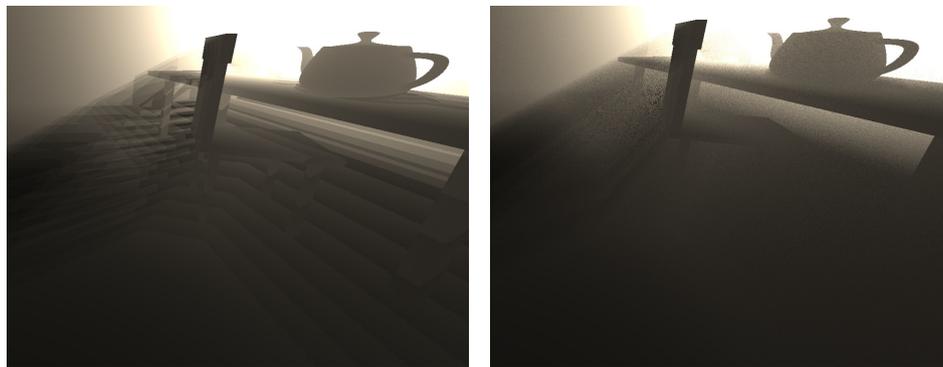
5 Auswertung

Für die Auswertung der drei Verfahren wurde ein Rechner mit einer Intel i5-4670 CPU und einer Nvidia GTX 1080 GPU verwendet. Der Rechenaufwand der Verfahren wurde anhand der Bildwiederholrate, sowie der Auslastung des Rechners bestimmt.

5.1 Raymarching

5.1.1 Notwendigkeit einer Rauschtextur

Bei der ersten Ausführung der Anwendung fällt eine störende Eigenschaft auf. Es sind sichtbare Lichtbänder zu erkennen, welche aufgrund der Start- und Endpunktwahl auftreten.



(a) Ohne Rausch Textur

(b) Mit Rausch Textur

Abbildung 11: Vergleich von Sampling Varianten

Um diesen Nebeneffekt der Diskretisierung der Abtastpunkte zu verschleiern, bietet es sich an den Start- und Endpunkt, welcher für das Raymarching verwendet wird ein wenig zu variieren. Dies gelingt mit einer einfachen Rauschtextur, welche vorab berechnet und innerhalb des Shaders verwendet wird, um einen variierenden Offset auf die Punkte aufzuzaddieren. Abbildung 11 zeigt die Unterschiede durch diese Rauschtextur.

5.1.2 Rechenaufwand

Die Messung des Rechenaufwandes des Raymarching Verfahrens wurde in zwei verschiedenen Auflösungen durchgeführt. Einmal in einer Auflösung von 720p und ein weiteres mal in einer Auflösung von 1080p. Die Anzahl an Iterationen beschreibt die Anzahl an Punkten, welche für die Berechnung des gestreuten Lichtes verwendet werden. Es wurde die Zeit für die

Ausführung des Shaders gemessen, welcher für die Beleuchtung zuständig ist, sowie die Anzahl an Bildern pro Sekunde, welche nach Messung einer gesamten Rendschleife möglich wäre. Diese theoretische Bildwiederholrate wurde durch den Swap Buffer und die Limitationen der Bildwiederholrate des Bildschirms beschränkt. Zuletzt wurde die Auslastung der Grafikkarte während der Ausführung gemessen.

Iterationen	Shader (ms)	FPS (theoretical)	FPS (after swap)	GPU (%)
5	1.12	~640	144	22
20	1.45	~460	144	26
100	2.38	~330	144	39
200	3.93	225	144	60
500	9.27	102	70	68

Abbildung 12: Raymarching bei einer Auflösung von 1280x720

Iterationen	Shader (ms)	FPS (theoretical)	FPS (after swap)	GPU (%)
5	1.92	~350	144	37
20	2.61	~290	144	47
100	4.79	175	144	73
200	8.75	103	79	65
500	19.98	48	36	73

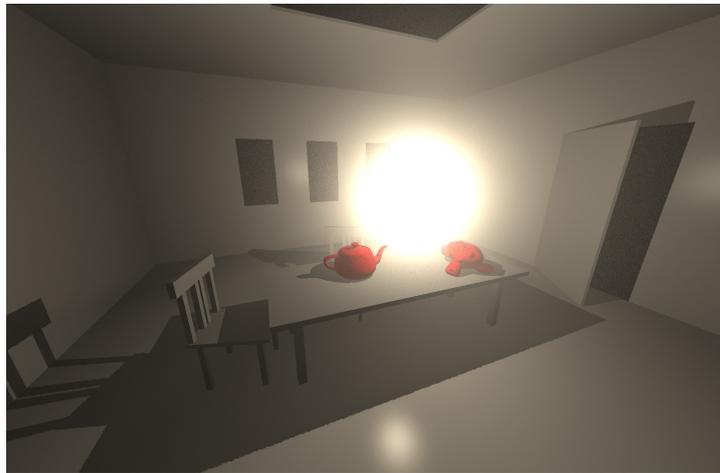
Abbildung 13: Raymarching bei einer Auflösung von 1920x1080

Offenbar besteht ein annähernd linearer Zusammenhang zwischen der Anzahl an Iterationen und der Rechendauer des Shaders bzw. der Auslastung der Grafikkarte. Somit ergibt sich ebenfalls eine entsprechende Verringerung der Bildwiederholrate. Es stellt sich nun also die Frage, wie viele Iterationen für ein gutes Bild tatsächlich notwendig sind und ab wann die zusätzlichen Abtastpunkte keinen erkennbaren Mehrwert tragen.

In Abbildung 14 zeigen sich bei 5 Iterationen deutliche Artefakte, wohingegen bei 20 Iterationen die Intensität des Rauschens deutlich abnimmt. Dieser Trend ist jedoch nicht linear und mit zunehmender Anzahl an Iterationen nimmt die Sichtbarkeit des Rauschens nur leicht ab. Bei einer Anzahl von 200 Iterationen lassen sich keine Hinweise auf das Rauschen mehr finden. Da bei einer Auflösung von 1080p dies nah an den gewünschten 60 Bildern pro Sekunde liegt, stellen 200 Iterationen die Obergrenze für diesen Rechner dar.



(a) 5 Sampling Punkte



(b) 20 Sampling Punkte



(c) 200 Sampling Punkte

Abbildung 14: Vergleich von Sampling Anzahlen

5.2 Analytisches Modell

5.2.1 Rechenaufwand

Entgegen dem Raymarching Verfahren besteht für das analytische Modell keinerlei Abhängigkeit von Iterationen innerhalb des Shaders. Stattdessen muss das Integral aus Gleichung 40 ausgewertet werden. Dies kann auf zwei Arten geschehen. Es kann innerhalb des Shaders berechnet werden und liefert somit mit etwas Rechenaufwand ein sehr genaues Ergebnis oder es kann vorberechnet und in einer Textur dem Shader übergeben werden. Letztere Variante spart eine ganze Menge Rechenaufwand, nimmt jedoch in Kauf, dass die Schrittweite der Parameter in der Textur eine Ungenauigkeit auf das Endergebnis hat.

Texture	Shader (ms)	FPS (theoretical)	FPS (after swap)	GPU (%)
No	3.41	~250	144	54
Yes	0.24	~1000	144	11

Abbildung 15: Analytisches Modell bei einer Auflösung von 1280x720

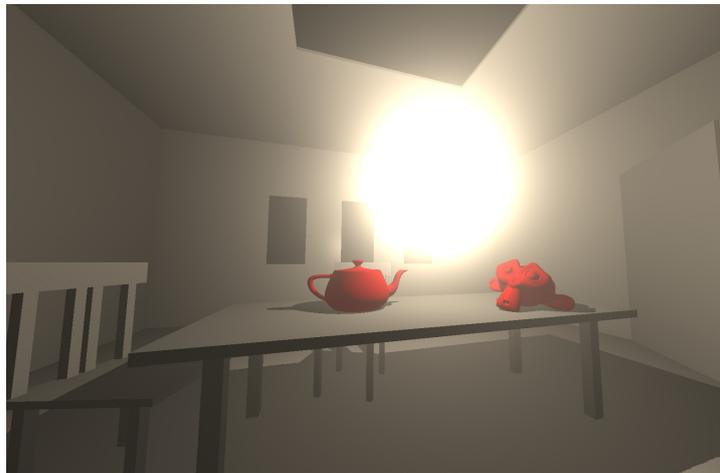
Texture	Shader (ms)	FPS (theoretical)	FPS (after swap)	GPU (%)
No	7.76	114	69	58
Yes	0.57	~600	144	22

Abbildung 16: Analytisches Modell bei einer Auflösung von 1920x1080

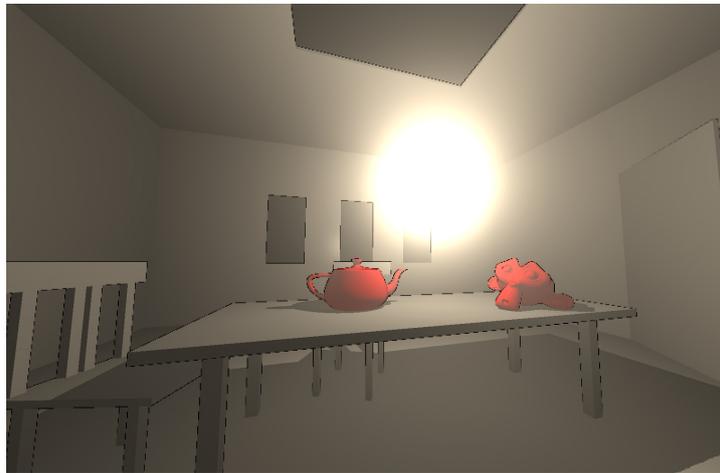
Während die Berechnung des gestreuten Lichtes gegenüber dem Raymarching sehr genau ist, da sie keine Diskretisierung durch die Einführung von Abtastpunkten besitzt, ist ihr Rechenaufwand für eine feste Auflösung konstant an die Genauigkeit des numerisch errechneten Integrals 40 gebunden. Findet diese Berechnung nun bereits zu Beginn der Anwendungsausführung statt, kann durch die Verwendung einer Textur als Look Up Tabelle, eine enorme Leistungssteigerung erreicht werden.

5.2.2 Probleme mit der Textur

Unglücklicherweise ist während der Umsetzung der Anwendung ist ein Fehler aufgetreten, welcher sich trotz Bemühungen nicht beheben ließ. Abbildung 17 zeigt den Fehler, welcher nur an starken Kanten der Szene und an diesen nur zur Hälfte der Kante auftritt.



(a) Berechnung im Shader



(b) Verwendung einer Textur

Abbildung 17: Vergleich des Analytischen Modells

Es ist noch ungeklärt, wieso dieser Fehler auftritt oder wie dieser zu beheben ist.

5.2.3 Nachteile des analytischen Modells

Letzlich sollte klar gemacht werden, welchen Nachteil das analytische Verfahren gegenüber dem Raymarching besitzt. Dieser findet sich in der vollständigen Abwesenheit von volumetrischen Schattenwürfen. Diese Eigenschaft ist in Abbildung 18 dargestellt. Somit verliert das Verfahren eine große Menge an realistischen Auftreten.



(a) Analytisches Modell

(b) Raymarching

Abbildung 18: Schattenwürfe der zwei Verfahren

5.3 Shadow Volumes

5.3.1 Vorteile gegenüber dem Raymarching

Der große Vorteil des auf Shadow Volumes basierten Modells liegt in der Fähigkeit die Anzahl der verwendeten Abtastpunkte für verschiedene Regionen separat voneinander zu betrachten. So gibt es entsprechend Abbildung 10 Bereiche, welche für die Berechnung des eingestreuten Lichtes relevanter sind oder vollkommen außer Acht gelassen werden können.



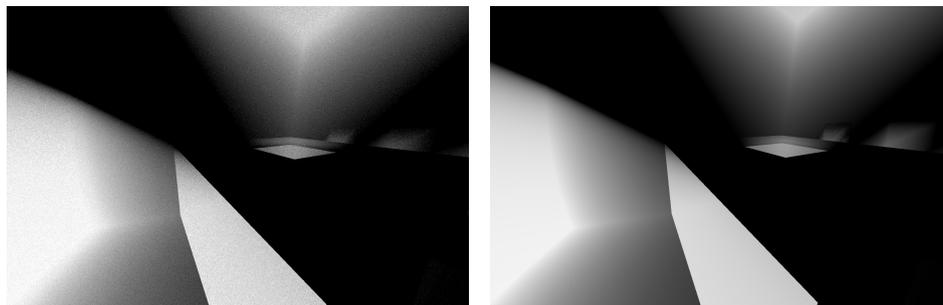
(a) Raymarching

(b) Shadow Volumes

Abbildung 19: Vergleich von Rauschen der zwei Verfahren

Ein gutes Beispiel findet sich in Abbildung 19. Es zeigt wie das Verfahren abhängig von dem betrachteten Szenario in der Lage ist trotz gleicher Anzahl von Abtastpunkten sehr genaue Schattenkonturen abzubilden. An manchen Stellen findet sich immer noch ein starkes Rauschen, welches jedoch niemals stärker sein wird als das Raymarching Verfahren bei einer gleichen Anzahl von Abtastpunkten.

Abbildung 20 zeigt eines der günstigsten Szenarien für das Verfahren. Es gibt einen klaren Start und Endpunkt, wodurch die Abtastpunkte innerhalb des beleuchteten Bereiches konzentriert werden können und die unbeleuchteten Regionen nicht beachtet werden müssen.



(a) Raymarching

(b) Shadow Volumes

Abbildung 20: Vergleich von Rauschen der zwei Verfahren

5.3.2 Rechenaufwand

Der Rechenaufwand dieses Verfahrens gegenüber dem Raymarching wirkt auf den ersten Blick unbeeindruckend. Es lassen sich klare Parallelen zwischen dem Aufwand der beiden Verfahren finden, da sie die selbe Anzahl an Abtastpunkten verwenden. Oftmals findet sich die Verbesserung darin, dass für das Shadow Volume Verfahren nicht so viele Iterationen notwendig sind, um ähnlich gute oder oftmals sogar bessere Bilder als das Raymarching zu erzeugen.

Iterationen	Shader (ms)	FPS (theoretical)	FPS (after swap)	GPU (%)
5	1.22	~300	144	43
20	1.58	~270	144	48
100	2.68	210	144	64
200	4.27	154	84	54
500	9.76	84	70	80

Abbildung 21: Shadow Volumes bei einer Auflösung von 1280x720

Iterationen	Shader (ms)	FPS (theoretical)	FPS (after swap)	GPU (%)
5	2.67	192	144	68
20	3.37	170	144	69
100	4.77	120	71	55
200	9.51	83	70	80
500	20.92	43	36	80

Abbildung 22: Shadow Volumes bei einer Auflösung von 1920x1080

Trotz sehr ähnlicher Rechenzeiten zu dem Raymarching innerhalb des Shaders liegt die Bildwiederholrate für das Shadow Volume Verfahren niedriger. Dies ist dem zusätzlichen Aufwand durch die Berechnung der Shadow Volumes geschuldet. Abbildung 23 zeigt wie lange der Compute Sha-

der benötigt, um verschieden aufgelöste Terrains in die Shadow Volumes umzuwandeln.

Auflösung Shadow Volume	Compute Shader (ms)	Volume Shader (ms)
32x32x6	0.37	0.26
64x64x6	0.38	0.29
128x128x6	0.39	0.37
256x256x6	0.44	0.59
512x512x6	0.57	1.28
1024x1024x6	1.03	4.17

Abbildung 23: Shadow Volume Berechnung

Es sollte angemerkt werden, dass mit wachsender Auflösung der Shadow Volumes ebenfalls die benötigte Renderzeit für den vordersten und hintersten Tiefenwert als auch die gesamte Auslastung der Grafikkarte steigen und somit weniger Zeit für die Berechnung der Beleuchtung bleibt.

6 Fazit

Es wurden drei Verfahren aus dem Bereich der Single Scattering Modelle für volumetrische Beleuchtung anhand einer OpenGL Implementation miteinander verglichen. Es wurden die Möglichkeiten und Einschränkungen dieser Verfahren aufgezeigt und es wurde deren Echtzeitfähigkeit bestätigt.

Während das analytische Modell stark auf mathematischen Erkenntnissen beruht ist es in der Lage sehr schnell Bilder zu generieren, welche dafür jedoch ohne Verschattungen daher kommen. Das Raymarching und das Shadow Volume Verfahren sind beide in der Lage auf solche Verschattungen Rücksicht zu nehmen, müssen dafür jedoch auf das Abtasten von Punkten in dem teilnehmenden Medium zurückgreifen, wodurch ihr Rechenaufwand erhöht wird. Das Shadow Volume Verfahren stellt trotz des zusätzlichen Aufwandes durch die Berechnung von Shadow Volumes eine Verbesserung zu dem Raymarching Verfahren dar, da es durch geschickte Fallunterscheidungen interessante Regionen von unwichtigen trennen kann.

Zum Schluss sollte erwähnt werden, dass viele moderne Verfahren nicht nur eines der drei Verfahren verwenden, sondern vielmehr auf die Erkenntnisse aller zurückgreifen und diese miteinander verknüpfen. So lassen sich bspw. durch Aufteilen eines Shadow Volumes in mehrere Volumen ohne Mehrfachverschattung, in Verknüpfung mit dem analytischen Modell, die Notwendigkeit von Abtastpunkten streichen.

Jedoch haben die implementierten Verfahren ebenfalls ihre Grenzen aufgezeigt. Um nach heutigen Maßstäben photorealistische Bilder zu generieren, müssen bspw. für Beleuchtung von Wolken Multiple Scattering Verfahren verwendet werden, da die Single Scattering Verfahren nicht in der Lage sind die komplexen Lichtstreuungen und feinen Details auszumachen.

Literatur

- [1] John F. Hughes, Andries van Dam, Morgan McGuire, David F. Sklar, James D. Foley, Steven K. Feiner, and Kurt Akeley. *Computer Graphics: Principles and Practice (3rd Edition)*. Addison-Wesley Professional, 2013.
- [2] Wikipedia. Radiometrie, 2017.
- [3] Wikipedia. Schwarzer Körper, 2018.
- [4] Laszlo Szirmay-Kalos. *GPU-based Techniques for Global Illumination Effects (Synthesis Lectures on Computer Graphics and Animation)*. Morgan and Claypool Publishers, 2008.
- [5] Srinivasa G. Narasimhan. *Models and Algorithms for Vision through the Atmosphere*. PhD thesis, Columbia University, 2003.
- [6] Wojciech Jarosz. *Efficient Monte Carlo Methods For Light Transport In Scattering Media*. PhD thesis, University of California, San Diego, 2008.
- [7] Matt Pharr and Greg Humphreys. *Physically Based Rendering, Second Edition: From Theory to Implementation*. Morgan Kaufmann, 2010.
- [8] Eva Cerezo, Frederic Pérez, Xavier Pueyo, Francisco J. Seron, and François X. Sillion. A survey on participating media rendering techniques. *The Visual Computer*, 21(5):303–328, jun 2005.
- [9] James F. Blinn. Light reflection functions for simulation of clouds and dusty surfaces. *ACM SIGGRAPH Computer Graphics*, 16(3):21–29, jul 1982.
- [10] Bo Sun, Ravi Ramamoorthi, Srinivasa G. Narasimhan, and Shree K. Nayar. A practical analytic single scattering model for real time rendering. *ACM Transactions on Graphics*, 24(3):1040, jul 2005.
- [11] Vincent Pegoraro and Steven G. Parker. An analytical solution to single scattering in homogeneous participating media. *Computer Graphics Forum*, 28(2):329–335, apr 2009.
- [12] James T. Kajiya and Brian P Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH Computer Graphics*, 18(3):165–174, jul 1984.
- [13] Nelson L. Max. Atmospheric illumination and shadows. *ACM SIGGRAPH Computer Graphics*, 20(4):117–124, aug 1986.
- [14] Tomoyuki Nishita, Yasuhiro Miyawaki, and Eihachiro Nakamae. A shading model for atmospheric scattering considering luminous intensity distribution of light sources. *ACM SIGGRAPH Computer Graphics*, 21(4):303–310, aug 1987.

- [15] P. Lecocq, S. Michelin, D. Arquès, and A. Kemeny. Simulation d'éclairage en présence de milieux participatifs: vers une solution temps-réel. 2000.
- [16] V. Biri, S. Michelin, and D. Arquès. Real-time single scattering with shadows. *Eurographics Symposium on Rendering*, pages 1 – 10, 2003.
- [17] Jos Stam and Eugene Fiume. Depicting fire and other gaseous phenomena using diffusion processes. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM Press, 1995.
- [18] R. Victor Klassen. Modeling the effect of the atmosphere on light. *ACM Transactions on Graphics*, 6(3):215–237, jul 1987.
- [19] John Irwin. Full-spectral rendering of the earth's atmosphere using a physical model of rayleigh scattering. In *In Proc. 14th Eurographics UK Conference*, pages 103–115, 1996.
- [20] Y. Dobashi, T. Yamamoto, and T. Nishita. Interactive rendering of atmospheric scattering effects using graphics hardware. 2002.
- [21] H. E. Rushmeier. Realistic image synthesis for scenes with radiatively participating media. 1988.
- [22] Christopher Kulla and Marcos Fajardo. Importance sampling techniques for path tracing in participating media. *Computer Graphics Forum*, 31(4):1519–1528, jun 2012.
- [23] Simon Premoze, Michael Ashikhmin, Jerry Tessendorf, Ravi Ramamoorthi, and Shree Nayar. Practical rendering of multiple scattering effects in participating media, 2004.
- [24] E. P. Lafortune and Y. D. Willems. Rendering participating media with bidirectional path tracing. pages 91–100, 1996.
- [25] M. Pauly, T. Kollig, and A. Keller. Metropolis light transport for participating media. *Rendering Techniques 2000 (Proceedings of the Eleventh Eurographics Workshop on Rendering)*, page 11–22, 2000.
- [26] Alexander Keller. Instant radiosity. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press, 1997.
- [27] H. W. Jensen and P. H. Christensen. Efficient simulation of light transport in scenes with participating media using photon maps. 1998.

- [28] J. Novak, D. Nowrouzezahrai, C. Dachsbacher, and W. Jarosz. Virtual ray lights for rendering scenes with participating media. *ACM Transactions on Graphics*, 31(4), July 2012.
- [29] Larry Yaeger, Craig Upson, and Robert Myers. Combining physical and visual simulation—creation of the planet jupiter for the film “2010”. *ACM SIGGRAPH Computer Graphics*, 20(4):85–93, aug 1986.
- [30] Simon Premoze and Michael Ashikhmin. Rendering natural waters. *Computer Graphics Forum*, 20(4):189–199, dec 2001.
- [31] Venceslas Biri and Sylvain Michelin. Real Time Animation of Realistic Fog. Poster session of 13th Eurographic Workshop on Rendering, June 2002. Poster.
- [32] Michael D. McCool. Shadow volume reconstruction from depth maps. *ACM Transactions on Graphics*, 19(1):1–26, jan 2000.