

TECHNISCHE UNIVERSITÄT DRESDEN

FAKULTÄT INFORMATIK

INSTITUT FÜR SOFTWARE- UND MULTIMEDIATECHNIK

PROFESSUR FÜR COMPUTERGRAPHIK UND VISUALISIERUNG

PROF. DR. STEFAN GUMHOLD

## Großer Beleg

Ground-Truth-Renderer für Partikelbasierte Daten

Josef Schulz

(Mat.-Nr.: 3658867)

Betreuer: Dipl-MedienInf. Joachim Staib

Dresden, 7. Juni 2015



---

## Aufgabenstellung

Die Darstellung von Partikeldaten mittels Kugelglyphen ist in der wissenschaftlichen Visualisierung inzwischen etabliert. Gerade bei dichten Datensätzen stellen kompakte Anordnungen von sehr vielen Kugeln jedoch ein Problem für die Erkennbarkeit der zu visualisierenden Vorgänge dar. Eine Möglichkeit, diesem Problem zu begegnen, ist es, über Blinn-Phong-Beleuchtung hinausgehende Effekte wie globale Schatten oder den Einsatz von Methoden aus dem Volume-Rendering zu integrieren. Durch deren Komplexität muss in Echtzeitvisualisierungen jedoch auf teilweise grobe Approximationen zurückgegriffen werden. Die Einschätzung der Approximationsqualität fällt häufig schwer, da keine Visualisierung des exakten Verfahrens verfügbar ist. Ziel dieser Belegarbeit ist die Umsetzung eines CPU-Renderers für Partikeldaten, der eine Reihe von erweiterten Visualisierungseffekten unterstützt. Er soll die Grundlage für Ground-Truth-Visualisierungen bieten. Zunächst soll eine geeignete Softwarearchitektur konzipiert und umgesetzt werden. Die Partikel sollen als mit lichtemittierendem und absorbierendem Gas gefüllte Kugeln interpretiert werden. Es sollen anschließend Methoden entwickelt werden, um einen physikalisch plausiblen globalen Schattenwurf und Lichttransport für eine beliebige Anzahl an Punkt- und Richtungslichtquellen zu ermöglichen. Die dafür notwendigen Gleichungen für Kugeln mit konstanter Dichte und Emission, sowie linearer Absorption, sollen soweit wie möglich analytisch bestimmt und, sobald nicht mehr möglich, mittels möglichst exakter numerischer Integratoren ausgewertet werden.

Die Teilaufgaben umfassen:

- Umfassende Literaturrecherche zur globalen Beleuchtungsrechnung in der Volumen Visualisierung
- Schrittweise Konzeption und Umsetzung einer erweiterbaren Architektur zum Erzeugen von Ground-Truth-Bildern:
  1. Zunächst als Raytracer für opake Kugeln, der globale Schatteneffekte von frei positionierbaren Punkt- und Richtungslichtquellen unterstützt
  2. Umsetzung eines Renderers, der Kugeln als Volumen nach dem Emissions-Absorptions-Modell rendert, dabei analytische Bestimmung des Volume-Rendering-Integrals, einschließlich Integration direkter Beleuchtung unverdeckter Lichtquellen
  3. Erweiterung zu verdeckten Lichtquellen und Bestimmung der Lichtstärke- und Farbe für Lichtstrahlen durch verdeckende Kugeln
- Unterstützung für ein Standardformat wie VRML
- Evaluation in Bezug auf Korrektheit, Bildartefakte und (numerische) Grenzfälle

Optional:

- Unterstützung für Refraktionseffekte
- Unterstützung komplexerer Materialtypen



---

# **Selbstständigkeitserklärung**

Hiermit erkläre ich, dass ich die von mir am heutigen Tag dem Prüfungsausschuss der Fakultät Informatik eingereichte Arbeit zum Thema:

*Ground-Truth-Renderer für Partikelbasierte Daten*

vollkommen selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Dresden, den 7. Juni 2015

Josef Schulz



# Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>3</b>
<b>2 Verwandte Arbeiten</b>	<b>5</b>
<b>3 Grundlagen</b>	<b>7</b>
3.1 Strahl und Kugelgleichung . . . . .	7
3.2 Die perspektivische Kamera . . . . .	8
<b>4 Rendergleichung</b>	<b>11</b>
4.1 Emission und Absorption . . . . .	12
4.2 Einfaches Beleuchtungs-Modell . . . . .	13
4.3 Erweitertes Beleuchtungs-Modell . . . . .	14
<b>5 Implementierung</b>	<b>19</b>
5.1 Szenenbeschreibung . . . . .	19
5.2 Details zum Algorithmus . . . . .	20
5.3 Aufbau des Programms . . . . .	21
<b>6 Auswertung und Fazit</b>	<b>23</b>
6.1 SSIM . . . . .	25
6.2 Quantitative Evaluation . . . . .	26
6.3 Fazit . . . . .	28
<b>Literaturverzeichnis</b>	<b>31</b>
<b>Abbildungsverzeichnis</b>	<b>33</b>
<b>Tabellenverzeichnis</b>	<b>35</b>



# 1 Einleitung

Ein Partikel ist ein beliebiges geometrisches Objekt. Diesem wird eine Position im Raum, als Attribut, zugeordnet. Zusätzliche weitere Attribute, wie zum Beispiel ein Geschwindigkeitsvektor sind optional. In Computersimulationen werden Partikel zur Beschreibung physikalischer Vorgänge verwendet. Innerhalb eines solchen Kontextes wechselwirken die Partikel und es ergeben sich analysierbare Strukturen. Neben Computersimulationen, der größten Quelle für Partikeldaten, sind es Messvorgänge, vorwiegend in der Biologie und Kernphysik, in denen Partikeldaten erhoben werden. Dabei können die Volumen der einzelnen Teilchen konkret erfasst werden.

Eine direkte und unverfälschte Darstellung ist für die Analyse und Auswertung notwendig. Visualisierungen sollen menschlichen Wahrnehmung bei dieser Aufgabe unterstützen. Eine bloße optische Verschönerung der Darstellung ist nicht das Ziel einer wissenschaftlichen Visualisierung, vielmehr geht es darum, bestimmte Eigenschaften in den Fokus des Betrachters zu rücken und Tendenzen hervorzuheben.

Die Abbildung von dreidimensionalen Partikeln auf eine zweidimensionale Ebene lässt sich nicht ohne Kompromisse bewerkstelligen. Die Fähigkeit des Menschen, die Tiefe von Objekten in einer Szene schätzen zu können, wird durch ein System ermöglicht, das den Wert für die Tiefe auf unterschiedlichen Wegen bestimmt. Ein Weg, die Tiefe zu ermitteln, besteht in der Schätzung der Epipolargeometrie. Bei der Betrachtung eines zweidimensionalen Bildes kann diese Variante lediglich die Bildschirmebene erfassen, jedoch nicht die Tiefe der abgebildeten Objekte. Die Objekttiefen in der Abbildung kann der Mensch anhand von Überdeckungen und durch eine perspektivische Transformation der Objekte abschätzen. Damit Überdeckungen erkannt und die abgebildeten Objekte von einander unterschieden werden können, müssen sich diese durch einen Kontrast voneinander abheben. Schattierungen lösen dieses Problem. Zudem wirkt ein Kugelglyph im Vergleich zur Darstellung durch einen einfarbigen Kreis plastischer. Überdeckungen verhindern allerdings, dass der Betrachter die dahinterliegenden Partikel sehen kann. Ganze Strukturen, zu denen beispielsweise Höhlen zählen, können deshalb nicht erfasst werden. Eine Möglichkeit, dieses Problem zu lösen, besteht darin, die Partikel transparent abzubilden, damit trotz der Überdeckung alle Strukturen sichtbar bleiben. Die zuvor erwähnten Schattierungen ergeben sich in der realen Welt aus der Wechselwirkung zwischen Licht und Materie. Lokale Beleuchtungsmodelle eignen sich zur Berechnung von Schattierung, unterstützen die Wahrnehmung der Tiefe jedoch nur bedingt. Eine weitere vom Menschen, zur Bestimmung von Objekttiefen genutzte Möglichkeit, ist die der Rekonstruktion von Lichtwegen. Zur Umsetzung wird eine globale Beleuchtung nötig, es existiert eine Reihe von Beleuchtungsmodellen, die die Wahrnehmungsunterstützung möglich machen. Existierende Algorithmen müssen auf Grund einer hohen Rechenkomplexität der globalen Beleuchtung an vielen Stellen approximativ vereinfachen. Insbesondere GPU-Implementierungen sind auf diese angewiesen, um die globale Beleuchtung in Echtzeit zur realisieren.

Ziel dieser Arbeit ist die Entwicklung eines CPU-Renderers für Partikel, bei dem der Fokus auf der Genauigkeit und nicht auf der Geschwindigkeit der Berechnung liegt. Die Partikel werden als mit Gas gefüllte Kugeln interpretiert und mit einem Verfahren der direkten Volumen-Darstellung gezeichnet. Direkte Verfahren zerlegen die Daten nicht in einem Vorverarbeitungsschritt in Netze aus Polygonen, sondern visualisieren die Daten in einem Schritt. Die Grundlage des Verfahrens stellt ein Volumenintegral dar, dessen Lösung mit Hilfe eines Raycast-Algorithmus dargestellt wird. Dabei handelt es sich um ein halb-analytisches Verfahren. Vorteile analytischer Lösungen liegen in der Genauigkeit und der Geschwindigkeit der Berechnung. Die erzeugten Bilder dienen als *Ground-Truth* Information zur Evaluation von approximativen Implementierungen.

Die Arbeit beginnt mit der Einführung der Grundlagen des Raycast-Verfahrens. Anschließend werden weitere Darstellungsalgorithmen und das Volumenintegral erläutert. Für dieses wird unter gewissen Einschränkungen, auf die später eingegangen wird, eine weitestgehend analytische Lösung hergeleitet. Nach dem Details zur eigentlichen Implementierung vorgestellt wurden, wird das Verfahren qualitativ und quantitativ evaluiert und im Schlusswort wird auf mögliche Erweiterungen eingegangen.

## 2 Verwandte Arbeiten

Die Arbeit [JSYR14] fasst verschiedene Algorithmen zur Beleuchtung und interaktiven Darstellung von Volumendaten in Echtzeit zusammen. Vorgestellte Algorithmen werden hinsichtlich ihrer technischen Umsetzung, Performance und ihrer Unterstützung für die menschliche Wahrnehmung klassifiziert. Diese Kriterien erleichtern die Auswahl des Verfahrens für den Entwickler in Abhängigkeit der Anwendung. Bei den Verfahren handelt es sich um approximative Varianten mit dem Ziel, interaktive Bildraten zu gewährleisten.

Die Grundlagen des Emission-Absorption-Modells werden von Nelson Max in der Arbeit [Max95] beschrieben. Das Volumenintegral des Emission-Absorption-Modells wird hergeleitet und Stück für Stück mit Effekten angereichert. Neben der einfachen Streuung und der Schattenberechnung wird auch die Mehrfachstreuung vorgestellt. Beendet wir die Arbeit mit der Vorstellung anwendbarer Lösungsverfahren.

In der Publikation [JPP98] wird ein zum Teil analytisches Verfahren vorgestellt. Der Algorithmus wird zur Darstellung von Voxelgittern eingesetzt. Hierbei wird jedem Voxel eine Dichte zu geordnet, aus welcher ein Wert für die Transparenz abgeleitet wird. Hinter dem Voxelgitter befindet sich eine Lichtquelle, deren Lichtintensität durch das Volumen unterschiedlich stark abgeschwächt wird. Die Integrale der Gleichungen werden streckenweise analytisch gelöst, wodurch ein Zuwachs an Genauigkeit erreicht wird.

In der Arbeit [ASW13] geht es um einen alternativen Ansatz, welcher nicht auf dem Emission- und Absorptionsmodell aufbaut, sondern von einem Punkt aus Stahlen in die umliegende Nachbarschaft verfolgt, um den Grad an Verdeckung an diesem Punkt zu bestimmen. Je mehr Segmente mit hoher Dichte in der Nachbarschaft liegen, umso geringer ist der Anteil des Lichtes, welches eben jenen Punkt erreicht. Ein Geschwindigkeitszuwachs wird durch eine im Vorfeld integrierte Transferfunktion erreicht. Ein Vorteil dieses Verfahrens stellt die Güte der erzeugten Schatten da.

Photonen-Mapping ist ein Verfahren zur Erzeugung von globalen Beleuchtungseffekten. Dieses kommt in der Arbeit [JKRY12] zur Beleuchtung von Volumendaten in Echtzeit zum Einsatz. Die dazu benötigte Photonenmap wird dabei mit Histogrammen realisiert. Mit deren Hilfe können Änderungen in interaktiven Bildwiederholungsraten ermöglicht werden, da bei Parameteränderungen nur die betroffenen Histogramme neu erzeugt werden müssen.

Die Monte Carlo-Methode ist ein Verfahren zur numerischen Approximation von Integralen. Diese kommt zur Lösung der Volumengleichung in der Arbeit [KPB12] zum Einsatz. Der Fokus der Arbeit liegt auf der physikalischen Grundlage dieses Verfahrens. Die Arbeit verspricht eine globale Beleuchtungsberechnung in Echtzeit und unterstützt unter anderem auch Mehrfachstreuungen.

Ambiente Verdeckung kommt in der Arbeit [KPH<sup>+</sup>03] zum Einsatz. Mit Hilfe des sogenannten *blurring* werden die Farbanteile des indirekten Lichtes in einem Buffer gehalten. Dieses Verfahren approximiert die Streuungseffekte, welche in Materialien auftreten können. Eine Erweiterung dieses Verfahrens stellt die Arbeit [SPH<sup>+</sup>09] dar, welche hier nur der Vollständigkeit halber erwähnt werden soll.



## 3 Grundlagen

Der konstruierte Algorithmus basiert auf dem Raycast-Verfahren. Grundlage dessen, ist das Modell einer Kamera. Diese Modell besitzt einen, in einzelne Bildpunkte aufgeteilten Sensor. Für jeden Pixel wird ein Strahl in die Szene geschossen und für jeden Strahl wird ein Kollisionstest mit den Objekten durchgeführt. Gibt es einen Schnitt von Objekt und Strahl, wird die Gleichung für das dazugehörige Stahlsegment gelöst. Das Ergebnis der Gleichung sind Lichtintensitäten, die sich auf der Sensorfläche aufsummieren.

In dieser Arbeit wird die Menge der Objekte auf Kugeln reduziert. Jede Kugel repräsentiert einen mit Gas gefüllten Partikel. Die Dichte im Inneren jedes Partikels wird als konstant angenommen und die Menge der Transferfunktionen auf Lineare beschränkt. Im Folgenden werden die Gleichungen für Strahlen, Kugeln und deren Schnittberechnung definiert. Das Kapitel wird mit der Herleitung der perspektivischen Kamera beendet.

Punkte werden in dieser Arbeit wie Vektoren behandelt. Sie unterscheiden sich von diesen in den Formeln dadurch, dass sie unterstrichen und nicht mit einem Pfeil gekennzeichnet sind:  $\underline{p}$  stellt einen Punkt und  $\vec{p}$  einen Vektor dar. Skalarprodukte werden mit Hilfe von spitzen Klammern  $\langle \vec{a}, \vec{b} \rangle$  repräsentiert.

### 3.1 Strahl und Kugelgleichung

Es wird mit der Definition des Strahls begonnen, der das zentrale Element des Algorithmus bildet. Die Formel beschreibt, wie jeder Punkt auf dem Strahl in Abhängigkeit eines skalaren Wertes  $t$ , eines Stützpunktes  $\underline{o}$  und eines Richtungsvektors  $r$  bestimmt werden kann. Die Gleichung definiert sich wie folgt:

$$\underline{p}(t) = \underline{o} + t \cdot \vec{r}, \text{ mit } t \in \mathbb{R} \quad (3.1)$$

Jeder Punkt auf dem Strahl  $\underline{p}(t)$ , ergibt sich aus der Addition eines Stützpunktes  $\underline{o}$  mit dem durch  $t$  skalierten Richtungsvektor  $\vec{r}$ . Der Vektor  $\vec{r}$  muss normiert sein, die Länge des Vektors muss genau 1 betragen:  $\|\vec{r}\| = 1$ . Ist der Wert von  $t < 0$  liegt der Punkt hinter dem Ausgangspunkt des Strahls, andernfalls davor oder im Fall von  $t = 0$  entspricht er eben diesem.

Neben dem Strahl spielt die Kugel, welche die geometrische Form der betrachteten Partikel darstellt, eine wichtige Rolle und soll ebenfalls definiert werden, damit die Schnittpunktberechnung durchgeführt werden kann.

Die Fläche einer Kugel wird im  $\mathbb{R}^3$  durch den Mittelpunkt  $\underline{m} = (x_0, y_0, z_0)$  und den Radius  $r$  parametrisiert. Jeder Punkt auf der Kugeloberfläche lässt sich durch den Abstand zum Mittelpunkt, dem Zentrum der Kugel, definieren. Die folgende Formel beschreibt diese Formulierung:

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2 \quad (3.2)$$

Jeder Punkt  $\underline{p} = (x, y, z)$ , der die Formel 3.2 erfüllt, liegt auf der Oberfläche der Kugel. Die Kugelgleichung 3.2 lässt sich auf beliebige  $\mathbb{R}^n$ , mit  $n \in \mathbb{N}_+$  erweitern. Die Beschreibung ist für alle  $n$  äquivalent:

$$\langle \underline{p} - \underline{m}, \underline{p} - \underline{m} \rangle = r^2 \quad (3.3)$$

Um den Schnittpunkt zwischen Kugel und Strahl zu berechnen, wird die Kugelgleichung 3.3 vereinfacht. Wird der Mittelpunkt der Kugel auf den Koordinatenursprung verschoben, liegen alle Punkte  $\underline{p}$  auf der Oberfläche  $O$ , wenn sie denn Abstand  $r$  zum Koordinatenursprung haben. Die zu erfüllende Bedingung hat demnach folgende Form:

$$\|\underline{p}\| = r \quad (3.4)$$

Nach Quadrieren der Gleichung 3.4 wird der Strahl anschließend in diese eingesetzt. Das Resultat ist eine quadratische Gleichung. Bis auf das Skalar  $t$  sind alle Werte bekannt. Die Gleichung

$$r^2 = \langle \underline{p}, \underline{p} \rangle = \langle r(t), r(t) \rangle = \langle \underline{p} + t \cdot \vec{r}, \underline{p} + t \cdot \vec{r} \rangle \quad (3.5)$$

kann nach  $t$  umgestellt und die Nullstellen berechnet werden. Es gibt entweder keine, eine oder zwei Lösungen für die Gleichung. Werden die bestimmten Schnittpunkte in die Gleichung des Strahls eingesetzt, können die Positionen der Schnittpositionen berechnet werden. Die komplette Lösungsformel definiert sich wie folgt:

$$t_{1,2} = \frac{-2 \cdot \langle \underline{p}, \vec{r} \rangle \pm \sqrt{4 \cdot \langle \underline{p}, \vec{r} \rangle^2 - 4 \cdot \langle \underline{p}, \underline{p} \rangle \cdot \langle \vec{r}, \vec{r} \rangle}}{2 \cdot \langle \vec{r}, \vec{r} \rangle}. \quad (3.6)$$

## 3.2 Die perspektivische Kamera

Um in der realen Welt Bilder aufzunehmen, wird eine Kamera benötigt. Hier wird eine Simulation durchgeführt und für diese wird das Modell einer Kamera benötigt, welches sich von der *camera obscura* ableitet. Dabei handelt es sich um das Modell einer Lochkamera, diese besteht aus einem Kasten. Auf der einen Seite befindet sich ein lichtempfindliches Material, welches den Sensor bildet, und auf der gegenüberliegenden Seite befindet sich ein schmales Loch. Das Licht wird von der Oberfläche der abzubildenden Objekte reflektiert und fällt durch das Loch auf den Sensor. Bei dem Vorgang der Abbildung findet eine vertikale und horizontale Spiegelung der Szene statt. Das führt dazu, dass das Bild der Szene spiegelverkehrt aufgenommen wird.

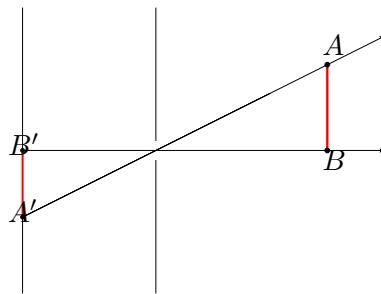


Abbildung 3.1: Schematische Darstellung des Abbildungsvorgangs der *camera obscura*

Die Abbildung 3.1 zeigt den Aufbau der *Camera obscura* schematisch. Auf der linken Seite der Abbildung befindet sich die Bildebene, auf welche die Szene abgebildet wird. In der Mitte der Abbildung 3.1 befindet sich die Wand mit dem kleinen Loch. Hier wird die erste Abstraktion durchgeführt, denn in der Realität kann das Loch eine gewisse Größe nicht unterschreiten, da andernfalls kein Licht mehr durch das Loch dringen würde. Auf Grund dieser minimalen Größe kommt es zu einer Glättung der Abbildung. Diese Einschränkung gilt für die Computersimulation nicht. Es wird angenommen, dass das Loch unendlich klein ist und deshalb die Szene unendlich genau abgebildet werden kann.

Auf der rechten Seite in der Abbildung 3.1 befindet sich ein Objekt in Form einer Strecke, welche durch die Punkte  $A$  und  $B$  begrenzt wird. Der Punkt  $B$  befindet sich mit der Sensormitte und dem Loch genau auf einer Ebene und wird auf den Punkt  $B'$  abgebildet. Der zweite Punkt  $A$  wird auf dem Punkt  $A'$  abgebildet. Genauer zeigt die Abbildung 3.1 den Strahlensatz, der in diesem Zusammenhang gilt.

Die Distanz zwischen dem Sensor und dem Loch der Kamera wird mit der Variable  $f$  bezeichnet. Da es sich- wie oben bereits erwähnt- um eine Computersimulation handelt, kann das Modell weiter abstrahiert werden. In der im Computer simulierten Welt kann der Sensor auch vor dem Loch positioniert werden, was zur Folge hat, dass die Abbildung entspiegelt wird.

Die Position der Kamera soll im Folgenden mit der Variable  $\underline{p}$  bezeichnet werden. Diese ist mit der Position des Lochs der Lochkamera identisch und wird in der Literatur auch als Augpunkt bezeichnet. Für jeden Pixel der Bildebene wird mindestens ein Strahl erzeugt, welcher als Stützpunkt die Position der Kamera erhält. Der Richtungsvektor ist der normierte Vektor von der Kameraposition zum Pixel. Zur Vereinfachung wird angenommen, dass die Bildebene parallel zur xy-Ebene des Koordinatensystems der Szenen ist und der Augpunkt genau im Ursprung von diesem liegt. Die Fläche des Sensors wird in  $W \times H$  Pixeln unterteilt. Für jeden Pixel  $(x, y)$  wird ein Strahl  $s$  in die Szene geschossen, dessen Stützpunkt der Augpunkt der Kamera ist. Der Richtungsvektor ist der normierte Vektor  $\vec{r}(x, y)$ , welcher sich wie folgt definiert.

$$\vec{r}(x, y) = \begin{pmatrix} (2\frac{x}{W} - 1) \cdot \frac{W}{H} \\ -2\frac{y}{H} + 1 \\ f \end{pmatrix} \quad (3.7)$$

Es ist zu beachten, dass der Richtungsvektor noch normiert werden muss:  $\vec{r}(x, y) = \frac{\vec{r}(x, y)}{\|\vec{r}(x, y)\|}$ . Jeder Kamerastrahl, im folgenden auch als Primärstrahl bezeichnet, hat die folgende Form:

$$\underline{s}(x, y) = \underline{p} + t \cdot \vec{r}(x, y) \quad (3.8)$$

Die Kamera kann an dieser Stelle nur entlang der z-Achse blicken. Eine Rotation im Raum lässt sich durch die Multiplikation einer Rotationsmatrix mit den Richtungsvektoren bewerkstelligen. Die Position des Kameramodells ist frei wählbar. Es genügt, den Augpunkt zu verschieben. Mit Hilfe der Distanz  $f$ , der fokalen Länge, kann ein Zoomeffekt der Kamera erzielt werden.



## 4 Rendergleichung

Die Algorithmen, zur Abbildung geometrischer Objekte auf die zweidimensionale Bildebene, lassen sich in zwei Gruppen unterteilen. Die erste Gruppe umfasst die Technik der Rasterisierung, die Objekte auf die Bildebene projiziert. Durch dieses Vorgehen entstehen Fragmente, deren Farbe nach einem beliebigen Vorgehen bestimmt werden kann. Geometrische Objekte werden in einem Vorverarbeitungsschritt, meist in polygonale Netze zerlegt. Ein großer Vorteil dieses Verfahrens ist die Geschwindigkeit, es lassen sich ohne weiteres echtzeitfähige Bildwiederholungsraten erreichen. Nachteilig ist die Zerlegung der Objekte in polygonale Strukturen, weil damit schon im Vorfeld die eigentliche Form approximiert wird.

Die zweite Gruppe bezeichnet die Vertreter der Raytracing-Algorithmen, Raycasting, Pathtracing oder Photonemapping sind Beispiele für spezielle Ausprägungen dieser Variante. Bei den meisten Vertretern wird ausgehend von der Bildebene, mit Hilfe von Strahlen die Entfernung zu den Objekten bestimmt. Diese Vorgehensweise ist zwar um ein Vielfaches langsamer, physikalisch jedoch plausibel. Bildwiederholungsraten in Echtzeit lassen sich mit diesen Algorithmen kaum erreichen. Algorithmen dieser Gruppe werden meistens zur Lösung der Rendergleichung verwendet. Unter diesem Namen wurde 1986 eine Integralgleichung in der Arbeit [Kaj86], von Jim Kajiya veröffentlicht.

Die Rendergleichung, in Form einer Integralgleichung, beschreibt die Energieerhaltung bei der Ausbreitung von Lichtstrahlen. Grundlage dieser Gleichung, sind physikalische Modelle, die die Wechselwirkungen von Licht mit Materie beschreiben. Eine sinnvolle Beschränkung ist die Interpretation des Lichtes mit Hilfe der Strahloptik, da zum Beispiel Beugungseffekte nur dann eine Rolle spielen, wenn die betrachteten Objekte nicht wesentlich größer sind als die Wellenlänge des Lichtes. Die Ausbreitung des Lichtes wird mit Lichtstrahlen beschrieben. Diese Lichtstrahlen können in der Strahloptik absorbiert, reflektiert, gestreut oder gebrochen werden. Grundlegend gilt, dass Strahlen ihre Ausbreitungsrichtung nur auf Grund der Wechselwirkung mit der Materie ändern. So können sich zwei Lichtstrahlen schneiden, ohne dass die Kollision ihre Energie oder ihren Weg beeinflusst. Eine weitere Eigenschaft der Strahloptik besteht in der Umkehrbarkeit der betrachteten Lichtwege. Die Gleichungen ändern sich nicht, wenn die Lichtrichtung umgekehrt wird.

Es existieren zwei Möglichkeiten, die Rendergleichung zu betrachten. Zum einen lässt sich sagen, dass diese angereichert mit allen Effekten aus der Natur heraus extrahiert und für die Verwendung vereinfacht wird. Zum anderen lässt sich die Rendergleichung von Grund auf herleiten und mit verschiedenen Effekten anreichern, so dass sie die Effekte aus der Natur abbildet. Nelson Max nutzt in seiner Arbeit [Max95] die zweite Variante. Er reichert die Gleichung Schritt für Schritt mit verschiedenen physikalischen Effekten an.

Neben der einfachen Emission- und Absorptionsgleichung, die eine starke Approximation der physikalischen Grundlagen darstellt, geht Nelson Max auf diverse Erweiterungen zur Berechnung von Streuungseffekten und Schatten ein. Der Begriff Albedo bezeichnet ein Maß für das Rückstrahlvermögen, das bedeutet, dass die Oberflächen das eingestrahlte Licht nur diffuse reflektieren. Es wird definiert, dass dieser Effekt für die betrachteten Volumen vernachlässigbar klein ist, und deshalb die Rendergleichung auf die Emission und Absorption beschränkt werden kann. Auf eine Phasenfunktion, beziehungsweise eine bidirektionale Reflektanzverteilungsfunktion, wird ebenfalls vernachlässigt.

## 4.1 Emission und Absorption

Ausgangspunkt ist das Emission-Absorption-Modell von Nelson Max. In diesem werden die Wechselwirkungen des Lichtes auf die Absorption und die Emission beschränkt. Die Streuung des Lichtes wird wie oben beschrieben vernachlässigt. Die Gleichung der Beleuchtungsberechnung unterteilt sich in zwei Summanden:

$$I = I_A + I_E \quad (4.1)$$

Die Variable  $I$  repräsentiert die von dem Sensor aufsummierte Intensität des Lichtes. Der Term  $I_A$  beschreibt die Abschwächung der Hintergrundbeleuchtung durch das mit Gas gefüllte Volumen und  $I_E$  die Emission des Gases. Die Abschwächung entspricht dem Produkt des Hintergrundlichtes mit einem Wert für die Transparenz, welcher in Abhängigkeit zur Streckenlänge  $D \in \mathbb{R}_+$ , die das Licht durch das Volumen zurücklegen muss, steht. Dieser Zusammenhang wird in der folgenden Gleichung dargestellt:

$$I_A = I_B \cdot T(D) \quad (4.2)$$

Die Transparenz wird dabei durch die Funktion  $T(D)$  ermittelt:

$$T(s) = \exp\left(-\int_0^s \tau(t)dt\right) \quad (4.3)$$

Der Grad der Abschwächung wird mit  $\tau(t) = \sigma(\nu(t))$  bestimmt. Die Funktion  $\tau(t)$  wird als Transferfunktion bezeichnet, die die Dichte des Volumens auf einen Wert für die Absorption abbildet. Die Menge der Transferfunktionen wird in dieser Arbeit auf lineare Funktionen beschränkt.  $\sigma(\nu(t)) = \lambda\nu$  skaliert mit einem konstanten Faktor  $\lambda \in (0, 1]$  den Wert für die Dichte des Volumens. Diese wird mit der Funktion  $\nu(t) = \kappa$  auf einen konstanten Faktor  $\kappa$  abgebildet. Zur Vereinfachung der Formeln wird eine weitere Funktion definiert, welche die Länge aus der Differenz einer hinteren und einer vorderen Position berechnet:

$$T'(t_n, t_f) = \exp\left(-\int_{t_n}^{t_f} (\lambda\kappa)dt\right) \quad (4.4)$$

In der Funktion 4.4 steht die Variable  $t_n$  für den Eintrittspunkt in das Volumen und  $t_f$  für den Austrittspunkt. Da das Produkt von  $\lambda$  und  $\kappa$  ein konstanter Wert ist, kann es aus dem Integral herausgezogen werden. Somit existiert eine analytisch Lösung:

$$T'(t_n, t_f) = \exp\left(-\lambda\kappa \cdot \int_{t_n}^{t_f} dt\right) = e^{-\lambda\kappa \cdot (t_f - t_n)} \quad (4.5)$$

Die Emission wird mit der Variable  $I_E$  bezeichnet und wird zunächst sehr allgemein definiert:

$$I_E = \int_0^D g(s) \cdot T'(s, D)ds \quad (4.6)$$

Die Funktion  $g(s)$  wird als Quellterm bezeichnet und stellt eine beliebige Funktion, in Abhängigkeit einer Position  $s$  dar. Das bei  $s$  emittierte Licht wird auch durch das das Volumen abgeschwächt. Die

Formeln für die Absorption und die Emission wird in die Gleichung 4.1 eingesetzt und es ergibt sich das komplette Modell für die Beleuchtungsberechnung in der Form:

$$I = I_B \cdot T(D) + \int_0^D g(s) \cdot T'(s, D) ds \quad (4.7)$$

Die Gleichung 4.7 ist noch sehr allgemein. Es wird beschrieben, wie sich das Hintergrundlicht durch das Volumen abschwächt. Unter der Beschränkung auf lineare Transferfunktion existiert eine analytische Lösung. Das Integral über den Quellterm wurde bisher noch nicht genauer definiert. In den nächsten beiden Absätzen werden zwei verschiedene Funktionen für  $g(s)$  definiert. Beide kommen im eigentlichen Algorithmus zum Einsatz.

## 4.2 Einfaches Beleuchtungs-Modell

Ein primärer Strahl, auch als Sichtstrahl bezeichnet, wird vom Sensor aus in die Szene geschossen. Die folgende Abbildung stellt diesen Vorgang schematisch dar.

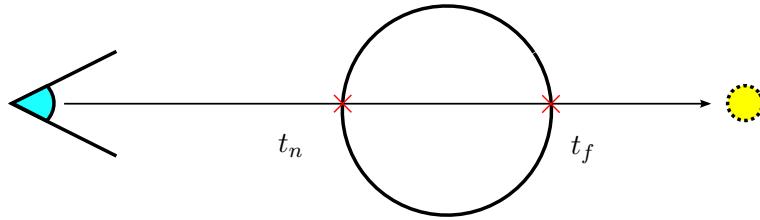


Abbildung 4.1: Die Abbildung zeigt einen primären Sichtstrahl, der ein Partikel schneidet. Auf der rechten Seite der Abbildung symbolisiert eine kleine Sonne die Hintergrundbeleuchtung und auf der linken ein Auge den Sensor.

Der Sichtstrahl schneidet ein Partikel an den Positionen  $t_n$  und  $t_f$ . Es wird eine Hintergrundbeleuchtung  $I_B$  mit einer konstanten Intensität definiert. Diese wird entlang des Strahls durch das Gas im Volumen mit der bereits ermittelten Formel  $I_A = I_B \cdot \exp(-\lambda\kappa(t_f - t_n))$  abgeschwächt. Die Transferfunktion für die Emission wird durch die Funktion  $c(\nu(t)) = \lambda I_c \cdot \nu(t)$  beschrieben, wobei  $I_c$  als konstante Lichtintensität gewählt wird und  $\nu(t) = \kappa$  die Dichte auf den konstanten Wert  $\kappa$  abbildet. Zunächst wird die Emissionsgleichung mit den Eintritts- und Austrittspunkten parametrisiert:

$$I_E = \int_{t_n}^{t_f} T'(t, t_f) \cdot g(t) dt \quad (4.8)$$

Die Quellfunktion wird in diesem Modell als  $g(t) = \lambda\kappa I_c$  definiert, so dass die Formel für die Emission ausgeschrieben der folgenden Form entspricht:

$$I_E = \int_{t_n}^{t_f} \exp(-\lambda\kappa(t_f - t)) \cdot \lambda\kappa I_c dt \quad (4.9)$$

Für das Integral 4.9 existiert eine analytische Lösung. Dazu werden im ersten Schritt die Konstanten aus dem Integral heraus gezogen

$$I_E = \lambda\kappa I_c \cdot \int_{t_n}^{t_f} \exp(-\lambda\kappa(t_f - t)) dt \quad (4.10)$$

und als nächstes wird die Klammer im Exponenten aus multipliziert:

$$I_E = \lambda\kappa I_c \cdot \int_{t_n}^{t_f} \exp(\lambda\kappa \cdot t - t_f \cdot \lambda\kappa) dt \quad (4.11)$$

Die Stammfunktion des Integranden  $e^x$  und dem dazugehörigen Differenzial  $dx$ , entspricht der Summe einer konstanten mit dem Integrand selbst:  $e^x + C$ . Der Integrand der Formel 4.11 entspricht dieser Form nicht, aufgrund der Subtraktion im Exponenten. Mit Hilfe einer geeigneten Substitution lässt sich die Form  $e^x$  erzeugen und das Integral lösen. Als erstes wird eine Variable  $u = \lambda\kappa \cdot t - t_f \cdot \lambda\kappa$  eingeführt und das Differenzial zu  $du = \lambda\kappa dt$  umgeschrieben. Jetzt müssen noch die obere und untere Schranke angepasst werden und die Substituierte Formel definiert sich wie folgt:

$$I_E = I_c \cdot \int_{\lambda\kappa(t_n - t_f)}^{\lambda\kappa \cdot (t_f - t_f)} e^u du \quad (4.12)$$

Es folgt die Auswertung des Integrals mit Hilfe der Stammfunktion  $e^u$ , die Lösung ergibt sich als Differenz der in die Stammfunktion eingesetzten Intervallgrenzen:

$$I_E = I_c \cdot (e^0 - e^{-\lambda\kappa(t_f - t_n)}) \quad (4.13)$$

Der Term  $e^0$  entspricht der Zahl 1 und der Zweite Term  $\exp(-\lambda\kappa(t_f - t_n))$  entspricht der Transparenz  $T'$ . Zum Schluss wird der gesamte Term durch das Formelzeichen  $\Theta$  ersetzt, damit die Formel kurz und übersichtlich bleibt.

$$\begin{aligned} I_E &= I_c \cdot (1 - e^{-\lambda\kappa(t_f - t_n)}) = I_c \cdot (1 - T'(t_n, t_f)) \\ I_E &= I_c \cdot \Theta(t_n, t_f) \end{aligned} \quad (4.14)$$

Für dieses Modell existiert somit eine analytische Lösung für den Absorptionsterm und für den Emissionsterm. Werden diese in die Gleichung 4.7 eingesetzt, dann ergibt sich die gesamte Gleichung zur Beleuchtung eines Partikels in einer kompakten Form wie folgt:

$$I = I_B \cdot T'(t_n, t_f) + I_c \cdot \Theta(t_n, t_f) \quad (4.15)$$

### 4.3 Erweitertes Beleuchtungs-Modell

Kern dieser Arbeit ist die Weiterentwicklung der eben eingeführten Volumengleichung. Die einfache Variante entstammt der Arbeit [SGG15], in welcher zusätzlich eine Implementierung der ambienten Verdeckung zum Einsatz kommt, um die globale Beleuchtung zu approximieren. Neben der Hintergrundbeleuchtung werden weitere Punkt- und Richtungslichtquellen zur Beleuchtung eingesetzt. Mit der Einführung dieser Lichtquellentypen gibt es einen Bruch mit der physikalischen Motivation der Gleichung.

Bei einer Punktlichtquelle handelt es sich um eine infinitesimale Kugellichtquelle, die im Gegensatz zu dieser kein Volumen und keine wirkliche Oberfläche besitzt. Daraus resultiert, dass die Wahrscheinlichkeit, dass ein Sekundärstrahl eine solche Lichtquelle treffen könnte, unendlich klein wird und schließlich gegen 0 geht. Eine Richtungslichtquelle dagegen kann als eine unendlich weit entfernte Kugellichtquelle interpretiert werden und entspricht damit schon eher einem physikalischen Äquivalent.

Zur Herleitung der Gleichung werden die Lichtquellen auf die Hintergrundbeleuchtung und eine Richtungslichtquelle beschränkt. Anschließend wird die gefundene Lösung wieder für beliebig viele Lichtquellen verallgemeinert. Die folgende Abbildung soll zunächst die Vereinfachung veranschaulichen.

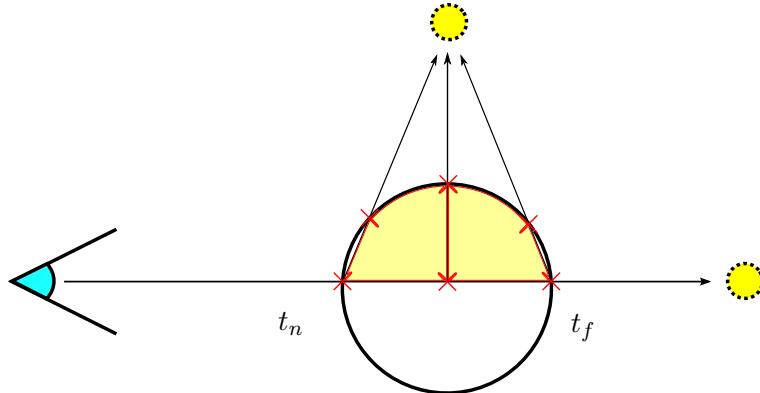


Abbildung 4.2: Die Notation ist dieselbe wie in der Abbildung 4.1. Diese Abbildung ergänzt das Modell um eine Punktlichtquelle und es werden drei sekundäre Strahlen zu dieser dargestellt.

Die Grundlage für das erweiterte Beleuchtungs-Modell ist dieselbe Gleichung 4.7, wie für das vorherige Modell. Diese wird zur besseren Übersicht an dieser Stelle mit den Notationen aus dem vorhergehenden Abschnitt wiederholt:

$$I = I_B \cdot T'(t_n, t_f) + \int_{t_n}^{t_f} g(t) \cdot T'(t, t_f) dt \quad (4.16)$$

Der erste Teil der Gleichung beschreibt die Absorption und bleibt unverändert. Die Modelle unterscheiden sich nur durch die Quellfunktion  $g(t)$ . Diese berechnet die Kugelfarbe an jedem Punkt  $t \in [t_n, t_f]$  entlang des Sichtstrahls und wird anschließend durch das Volumen abgeschwächt. Einfluss auf die Farbe der Kugeln hat die Intensität der Punktlichtquelle und die Emission des Gases. Die Funktion  $l(t)$  liefert die Schnittlänge des Sekundärstrahls mit dem Partikel. Ein Sekundärstrahl ist in diesem Zusammenhang ein Strahl, welcher ausgehend von der Position  $t$  in die Richtung der Richtungslichtquelle geschossen wird. Die Farbe an der Position  $t$  bestimmt sich mit der Hilfe der Länge  $l(t)$  und dem Modell aus dem vorherigen Abschnitt durch die folgende Rechenvorschrift:

$$g(t) = I_L \cdot T'(0, l(t)) + I_c \cdot \Theta(0, l(t)) \quad (4.17)$$

Die Variable  $I_L$  entspricht der Intensität der Richtlichtquelle. Punkt und Richtungslichtquellen unterscheiden sich in der Berechnung des Richtungsvektors für die Sekundärstrahlen. Der Richtungsvektor der Sekundärstrahlen für Punktlichtquellen bestimmt sich aus der normierten Differenz zwischen der Position der Lichtquelle und der Position auf dem Sichtstrahl in Abhängigkeit von  $t$ . Richtungslichtquellen werden bereits mit einem normierten Richtungsvektor parametrisiert. Die Invertierung von diesem entspricht dem Richtungsvektor des Sekundärstrahls. Wird die Funktion 4.17 in die Gleichung 4.16 eingesetzt, ergibt sich daraus ein Integral:

$$I = I_B \cdot T'(t_n, t_f) + \int_{t_n}^{t_f} (I_L \cdot T'(0, l(t)) + I_c \cdot \Theta(0, l(t))) \cdot T'(t, t_f) dt \quad (4.18)$$

Um zu überprüfen, ob sich diese Gleichung analytisch lösen lässt, wurde zu Beginn angenommen, dass nur eine Richtungslichtquelle vorhanden ist. Die gesamte Gleichung wird dadurch kompakter, da der Richtungsvektor  $\vec{d}$  der Richtungslichtquelle in der gesamten Szene konstant ist. Der Richtungsvektor  $\vec{d}$  wird benötigt, um die Länge der Sekundärstrahlen  $l(t)$  in der Gleichung 4.18 zu bestimmen. Dazu wird der Strahl aus der Gleichung 3.1 in die Lösungsformel 3.6 eingesetzt. Der Ursprung des Strahls liegt im Inneren der Kugel, deshalb wird nur der Schnittpunkt  $t_1$  benötigt und dessen Wert entspricht der gesuchten Länge  $l(t)$ . Diese lässt sich durch die folgende Vorschrift berechnen:

$$l(t) = \frac{-2 \cdot \langle \underline{p}(t), \vec{d} \rangle + \sqrt{4 \cdot \langle \underline{p}(t), \vec{d} \rangle^2 - 4 \cdot \langle \underline{p}(t), \underline{p}(t) \rangle \cdot \langle \vec{d}, \vec{d} \rangle}}{2 \cdot \langle \vec{d}, \vec{d} \rangle}. \quad (4.19)$$

Wird in die Gleichung 4.19 der Strahl 3.1 eingesetzt und die gesamte Gleichung aus multipliziert, dann lässt sich das Renderintegral für die Absorption durch die folgende Gleichung vereinfacht repräsentieren:

$$F(x) = \int I \cdot \exp(a \cdot x + b + \sqrt{c \cdot x^2 + d \cdot x + e}) dx \quad (4.20)$$

Dieses Integral lässt sich nicht elementar integrieren. Mehr Informationen zur Integrierbarkeit von Funktionen und warum sich einige Funktionen nicht elementar integrieren lassen, können der Arbeit von Martin Huber [Hub96] entnommen werden.

Da keine analytische Lösung für die Gleichung 4.18 existiert, muss an dieser Stelle nummerisch approximiert werden. Dazu wird das Integral in Riemann Summen zerlegt. Bevor die Lösung präsentiert und die Quellfunktion für mehrere Lichtquellen angepasst wird, soll die numerische Approximation anhand einer fiktiven Funktion  $f$  beschrieben werden. Dazu wird das Integral der Funktion  $f$  zu einer Summe von skalierten Funktionswerten überführt:

$$\int_0^D f(x) dx \approx \sum_{i=1}^n f(x_i) \Delta x \quad (4.21)$$

Das Intervall von 0 bis  $D$  wird in  $n$  kleinere Teile zerlegt, wobei jedes dieser Teilintervalle die selbe Länge  $\Delta x = \frac{D}{n}$  besitzt. Aus jedem Intervall  $i$  wird ein beliebiges Element  $x_i$  aus diesem Intervall gewählt und der Funktionswert  $f(x_i)$  mit  $\Delta x$  skaliert und auf das Ergebnis aufsummiert.

Demnach lässt sich die Formel 4.16 zu der folgenden Gleichung vereinfachen:

$$I = I_B \cdot T'(t_n, t_f) + \sum_{t=t_n}^{t_f} g(t) \cdot T'(t, t_f) \Delta t \quad (4.22)$$

Nun wird eine Anzahl von  $n$  Stützpunkten zur Regulierung der Güte der Approximation definiert. Generell gilt, dass je größer der Wert von  $n$  ist, um so besser wird das Integral approximiert. Der Wert für die Skalierung bestimmt sich als  $\Delta t = \frac{t_f - t_n}{n}$ .

Im Folgenden entspricht die Menge  $L = \{L_1, L_2, \dots, L_m\}$  der Menge der Lichtquellen. Jede von ihnen wird über den dazugehörigen Index identifiziert. Die Intensitäten der Lichtquellen werden mit dem For-

melzeichen  $I_{L_i}$  symbolisiert. Dabei entspricht  $i$  dem Index der dazugehörigen Lichtquelle. Damit kann der Quellterm erweitert werden, so dass die Formel für die Beleuchtungsberechnung lautet:

$$I = I_B \cdot T'(t_n, t_f) + \sum_{t=t_n}^{t_f} \left( \frac{1}{m} \cdot \sum_{i=1}^m I_{L_i} \cdot T'(0, l(t)) + I_c \cdot \Theta(0, l(t)) \right) \cdot T'(t, t_f) \Delta t \quad (4.23)$$

Die Gleichung 4.23 bildet die Grundlage der Beleuchtungsberechnung. Wie und wann diese Gleichung im Raycast-Algorithmus zum Einsatz kommt, wird im folgenden Kapitel beschrieben.



## 5 Implementierung

Die beschriebenen Beleuchtungsmodelle wurden für den Schnitt eines Strahls mit einem Partikel aufgestellt. In diesem Kapitel werden die Grundlagen und die beiden Beleuchtungsmodelle zusammengeführt. Es wird erläutert, wie die Bilder im Detail erzeugt werden und welche Vereinfachungen die Berechnung beschleunigen.

Die zu zeichnende Szene besteht aus einer Menge von Partikeln, dargestellt als Kugeln. Jede Kugel wird durch ihre Position und einen Radius parametrisiert. Zusätzlich wird für jede Kugel ein Material definiert, welches sich aus einer Farbe und ein Wert für die maximale Lichtundurchlässigkeit zusammensetzt. Für die Beleuchtungsberechnung, wird für jede Kugel der Parameter  $\kappa$  benötigt, denn dieser beschreibt die Lichtundurchlässigkeit und damit die Transparenz für die entsprechende Kugel. Dieser Parameter wird jedoch nicht direkt gesetzt, sondern berechnet sich aus der angegebenen maximalen Lichtundurchlässigkeit  $\Theta_{max} \in [0, 1]$  in Abhängigkeit des Radius  $r$ . Um den Wert von  $\kappa$  zu bestimmen, wird  $\kappa$  in die Gleichung für  $\Theta$  eingesetzt:

$$\Theta(t_n, t_f) = 1 - e^{-\lambda\kappa(t_f - t_n)} \quad (5.1)$$

Wird  $\Theta_{max}$  in die Formel 5.1 eingesetzt, dann lässt diese sich nach  $\kappa$  auflösen. Zusätzlich wird die größte mögliche Distanz  $2 \cdot r$  für die Länge  $t_f - t_n$  gesetzt und es ergibt sich folgende Lösung für  $\kappa$ :

$$\kappa = -\frac{1}{\lambda \cdot 2r} \ln(1 - \Theta_{max}) \quad (5.2)$$

Der Parameter  $\lambda$  wird für alle Kugeln global gewählt und liegt im Intervall  $(0, 1]$ , mit diesem Parameter kann die Transparenz der Kugeln global beeinflusst werden.

Das Programm unterstützt eine Vielzahl von Einstellungsmöglichkeiten, diese werden in einer externen Textdatei gesetzt. Deren Aufbau und mögliche Parameter werden im Folgenden Abschnitt erläutert.

### 5.1 Szenenbeschreibung

Wie oben beschrieben, lässt sich die Szene und die Programmparameter durch eine Textdatei definieren. Die Datei wird Zeilenweise eingelesen, jede Zeile bietet die Möglichkeit einen Befehl zu setzen. Für die Platzierung und Ausrichtung der Kamera existieren zum Beispiel drei verschiedene Varianten. In allen Varianten wird die Kamera durch die Angabe eines Vektors positioniert. Die Blickrichtung kann entweder mit zwei Drehwinkeln beeinflusst werden oder es wird ein Punkt gewählt, auf den die Kamera blicken soll. Dieser Blickpunkt kann auch als Massenzentrum gewählt werden, sodass die Kamera auf den Schwerpunkt der Szene ausgerichtet wird.

Um die Szene mit Kugeln anzureichern, existieren zwei Möglichkeiten. Zum einen können die Kugeln direkt in der Textdatei definiert werden und zum anderen wird das laden von SIFF-Dateien unterstützt. In SIFF-Dateien können die Kugeln durch ihrer Position, ihren Radius und ihrer Farbe ebenfalls Zeilenweise beschrieben werden.

Um die Anzahl der Kollisionstests auf ein Minimum zu reduzieren, wird eine Beschleunigungsdatenstruktur in Form eines KD-Baums eingesetzt. Dieser wird direkt nach dem Laden der Szene konstruiert

und es lässt sich eine maximale Baumtiefe angeben. Dieser Wert, sollte in Abhängigkeit von der Anzahl der gesetzten Partikel gewählt werden.

Weitere globale Parameter, wie  $\lambda$  und die Anzahl von Stützpunkten lassen sich ebenfalls setzen. Der Parameter  $\Theta_{max}$  lässt sich für jede Kugel einzeln setzen, aber auch global für alle Kugeln festlegen. Das gleiche gilt für die Farbe und die Größe der Kugeln. Es existiert ein weiterer globaler Parameter, welcher den Approximationsgrad bestimmt. Was es mit diesen Parameter auf sich hat, wird später beschrieben.

Neben der Hintergrundbeleuchtung gibt es die Möglichkeit beliebig viele Punkt- und Richtungslichtquellen zu definieren. Beide Typen von Lichtquellen besitzen einen Farbvektor als Parameter. Die Punktlichtquellen besitzen zusätzlich eine Position und Richtungslichtquellen besitzt anstatt der Position einen normierten Richtungsvektor.

Die Beleuchtungsbrechung lässt sich neben dem erläuterten Emission- und Absorptionsmodell auch mit dem Phong-Beleuchtungsmodell berechnen. Beide Varianten, können entweder mit oder ohne globale Schatteneffekte berechnet werden. Transparenzeffekte werden mit dem Phong-Beleuchtungsmodell nicht unterstützt.

## 5.2 Details zum Algorithmus

Im folgenden wird beschrieben, wie der Raycast-Algorithmus arbeitet und wann welches Beleuchtungsmodell zum Einsatz kommt. Grundlegend gilt, das die Auflösung, des zu erzeugenden Bildes  $W \times H$  entspricht. Unabhängig vom Beleuchtungsmodell, wird für jeden Bildpunkt  $(x, y) \in W \times H$  wird ein primärer Strahl erzeugt, dessen Ursprung der Kameraposition entspricht. Der Richtungsvektor lässt sich anhand der Vorschrift 3.7 bestimmen. Das Resultat dieses Schrittes ist eine Liste der primären Strahlen.

Ein großer Vorteil der Raycast-Technik besteht in der Möglichkeit die Berechnungen zu parallelisieren, da die Beleuchtungsberechnung für jeden Strahl unabhängig voneinander durchgeführt werden kann. In der eigentlichen Implementierung wird die Parallelisierung mit Hilfe der OpenMP-Bibliothek<sup>1</sup> umgesetzt.

Für jeden Primärstrahl, Sichtstrahl wird der Schnitt mit allen in der Szene befindlichen Kugeln berechnet und die Beleuchtungsberechnung durchgeführt.

Im Fall des Phong-Beleuchtungsmodells ist allerdings nur der erste Schnittpunkt, ausgehend vom Sensor von Interesse. An der Schnittposition wird die Normale berechnet, um einen Wert für die diffusen und für die spekularen Reflektionsanteile zu bestimmen. Das genaue Modell wird in der Arbeit [Bli77] erläutert. Zur Erzeugung der globalen Schatten, müssen ausgehend des Schnittpunktes Sekundärstrahlen zu jeder Lichtquelle geschossen werden und wenn ein Sekundärstrahl dabei eine andere Kugel schneidet, dann wird die Farbe der Lichtquelle mit dem Nullvektor ersetzt.

Die Bildsynthese für das Emission- und Absorptionsmodell ist etwas aufwendiger. Für die Sichtstrahlen wird die Beleuchtung durch das erweiterte Beleuchtungsmodell berechnet. Dazu muss ebenfalls für jeden Strahl ein Schnitttest mit allen Kugeln durchgeführt werden. Sobald alle Kugeln bestimmt worden sind, die von einem Sichtstrahl geschnitten worden, müssen diese entlang des Strahls sortiert werden. Anschließend wird für jede von diesen Kugeln, die erweiterte Beleuchtungsberechnung durchgeführt. Dabei wird mit der Kugel begonnen, die am weitesten vom Ursprung des Strahls entfernt liegt. Das Resultat der Beleuchtungsberechnung, für diese Kugel, wird als neue Hintergrundbeleuchtung für die nächste eingesetzt. Dieser Vorgang wird wiederholt, bis alle Kugeln entlang des Strahls abgearbeitet wurden. Am Sensor wird die Intensität der letzten Beleuchtungsberechnung gespeichert und anschließend normiert.

Da sich für die erweiterte Beleuchtungsberechnung keine analytische Lösung finden ließ, muss für jeden Stützpunkt ein neuer Strahl in Richtung jeder Lichtquelle geschossen werden. Für jeden Sekundärstrahl

---

<sup>1</sup><http://openmp.org/wp/>

wird das Licht, der entsprechenden Lichtquelle, als Hintergrundbeleuchtung betrachtet. Werden globale Schatten verwendet werden, dann wird dieses Licht entlang des Strahls wiederum durch die vom Sekundärstrahl geschnittenen Partikel abgeschwächt. Diese Abschwächung wird mit vereinfachten Beleuchtungsmodell berechnet. Für jeden Sekundärstrahl ist ein erneuter Schnitttest mit allen Kugeln nötigt, sowie eine Sortierung der Ergebnismenge entlang des Sekundärstrahls.

Die Geschwindigkeit des Algorithmus hängt von der Auflösung des Bildes und der damit verbunden Anzahl von Primärstrahlen ab. Entscheidend für Berechnungsdauer ist die Wahl der Stützpunktanzahl, da für jeden ein Sekundärstrahl Szene geschossen werden muss und für jeden ein Schnitttest und eine anschließende Sortierung erforderlich ist.

An dieser Stelle wird bereits angesprochene Approximationsgrad als ein neuer Parameter  $N$  eingeführt. Der Wert von  $N$  bestimmt, für wie viele Kugeln entlang des Primärstrahls, aber dieses mal ausgehend von dem Ursprung des Strahls, die Beleuchtung mit der erweiterten Formel bestimmt werden soll. Mit Hilfe dieser Approximation lässt sich die Anzahl der zu erzeugenden Sekundärstrahlen drastisch reduzieren und dadurch die Dauer der Berechnung verringern. Eine weitere Möglichkeit die Anzahl der Sekundärstrahlen zu reduzieren, lässt sich durch eine adaptive Wahl der Stützpunktanzahl realisieren. Das bedeutet, dass für jede verdeckte Kugel, in Abhängigkeit des Verdeckungsgrades Stützpunktanzahl reduziert wird, bis schließlich das Vereinfachte Modell zum Einsatz kommt. In diesen Programm wurde die Anzahl nicht adaptiv gewählt, sondern nur die einfache Approximation implementiert. Der Einfluss und die Güte dieser Approximation wird im nachfolgenden Kapitel anhand von Beispielen evaluiert.

## 5.3 Aufbau des Programms

Im folgenden wird der Aufbau des eigentlichen Programms anhand des UML-Diagramms 5.1 erläutert. Die Hauptklasse hat den selben Namen wie das Programm, sie heißt *zikade* und besteht im wesentlichen aus zwei Methoden. Die Methode *loadScene* liest die Textdatei ein, welche die Szene beschreibt und initialisiert damit alle für den Zeichenvorgang nötigen Variablen. In der zweite Methode *render* ist der eigentliche Algorithmus zum Berechnen des Bildes implementiert.

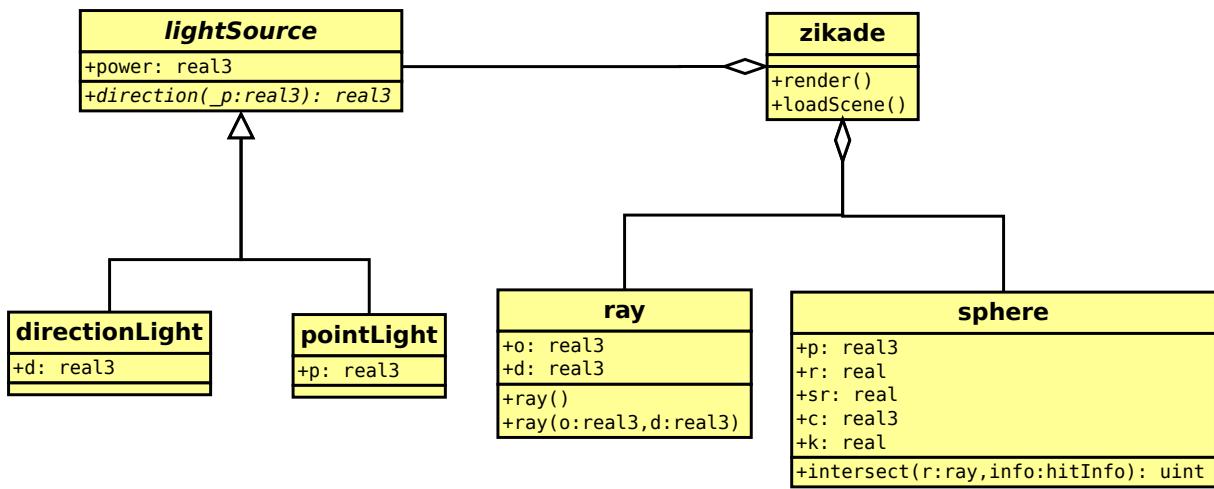


Abbildung 5.1: UML-Diagramm des Programms, die Hauptklasse heißt *zikade*.

Im Diagramm tauchen Variablen vom Typ *real* auf. Dieser Bezeichner kapselt den dahinter liegenden Gleitkommazahl-Typ und entspricht in dieser Implementierung dem Typ *double*. Variablen vom *real3* entsprechen dreidimensionalen Vektoren, deren einzelne Einträge dem Typ *real* entsprechen. Die Klasse *lightSource* ist eine abstrakte Klasse, von der sich die spezialisierten Klassen *directionLight* und *pointLight* ableiten. Die Variable *power* entspricht der Farbe des Lichtes und die Methode *direction* liefert die

Richtung des Lichtes zurück. Die Klasse *ray* repräsentiert den Strahl aus dem Grundlagenkapitel und die Klasse *sphere* entsprechend die Kugel. Weiterhin existiert eine Klasse die den KD-Baum abbildet. Da diese die Berechnung zwar beschleunigt, aber für den Algorithmus nicht erforderlich ist, wird auf ihre Repräsentation im Diagramm verzichtet.

## 6 Auswertung und Fazit

Zu Beginn der Arbeit wurden die mathematischen Grundlagen der Raycast-Technik hergeleitet und im Anschluss zwei Lösungen für die Volumengleichung besprochen. Die Idee bestand darin die Gleichung, für die Beleuchtung, weitestgehend analytisch zu lösen. Für das einfache Beleuchtungsmodell existiert eine analytische Lösung, aber zur Berechnung des erweiterten Modells ist eine numerische Approximation von Nötien. Dazu wird das Integral in Riemann-Summen zerlegt, damit die Lösung näherungsweise berechnet werden kann. Aufgrund der Komplexität des Verfahrens, wurde eine Vereinfachung zur Beschleunigung der Bildsynthese vorgestellt.

In diesem Kapitel werden zunächst qualitative Unterschiede in Abhängigkeit der gewählten Parameter erläutert und im Anschluss wird die beschleunigte Variante des Algorithmus evaluiert.

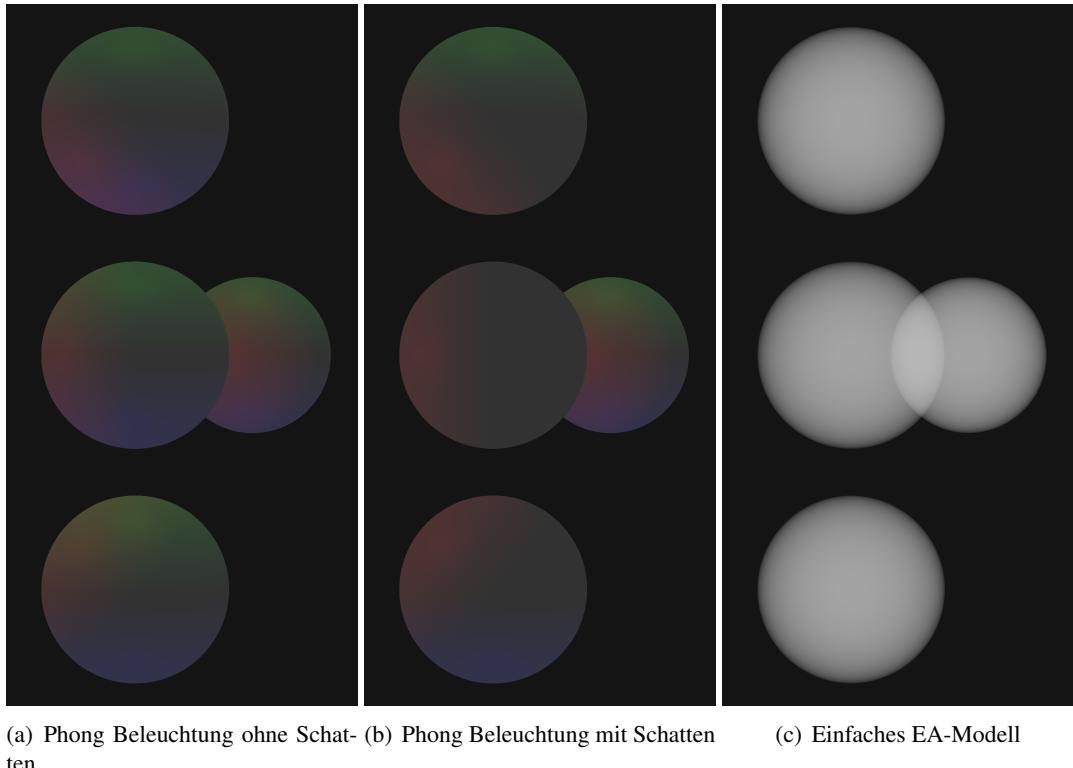


Abbildung 6.1: Abbildungen einer einfachen Szene gezeichnet mit dem Phong-Beleuchtungsmodell, einmal ohne und einmal mit globalen Schatteneffekten. Zusätzlich, wird die selbe Szene, gezeichnet mit dem einfachen Emissions- und Absorptionsmodell dargestellt.

Die Abbildungen 6.1 und 6.2 zeigen eine sehr einfache Szene bestehend aus vier Kugeln. Drei dieser Kugeln sind vertikal verschoben, aber in der gleichen Tiefe platziert. Die vierte Kugel ist etwas weiter nach hinten verschoben und so positioniert, dass eine Verdeckung durch die Kugel in der Mitte entsteht, damit die Transparenzeffekte deutlich sichtbar werden. Die Szene wird von drei Lichtquellen beleuchtet. Von oben-nach unten strahlt eine grüne Richtungslichtquelle und von unten-nach oben wird die Szene von einer blauen Richtungslichtquelle beleuchtet. Auf der linken Seite und auf der Höhe der beiden mittleren Kugeln, ist eine rot leuchtende Punktlichtquelle platziert.

In den Abbildungen 6.1 (a) und (b) wurde die Szene mit dem Phong-Beleuchtungsmodell beleuchtet. Wobei die linke Abbildung keine Schatteneffekte zeigt, das Licht wird im Gegensatz zur mittleren Abbildung nicht durch die Kugeln verdeckt. Ein Problem dieses Modells wird besonders in der mittleren Abbildung deutlich. Denn die Kugel in der Mitte, der Abbildung 6.1.c wird nur durch die rote Punktlichtquelle beleuchtet, da die anderen Lichtquellen verdeckt werden. Dort wo kein Licht auf die Kugel fällt entsteht keine Tiefenwirkung, da die Oberfläche der Kugel nicht schattiert wird. Anders verhält es sich im dritten Bild (c) der Abbildungen 6.1, dort wird die gleiche Szene durch das einfachen Emissions- und Absorptionsmodell dargestellt. Dieses unterstützt Transparenzeffekte, so dass der verdeckte Teil der nach hinten verschobenen Kugel ebenfalls sichtbar wird. Das einfache Emissions- und Absorptionsmodell berechnet keine Wechselwirkungen mit anderen Lichtquellen, außer dem Hintergrundlicht. Dieses wird durch die Kugeln abgeschwächt und gleichzeitig emittiert jede Kugel entsprechend ihrer Kugelfarbe eigenes Licht. Die Kugeln wirken auf Grund der unterschiedlichen langen Strecken, die das Licht durch die Kugeln zurücklegen muss weitaus plastischer als jene, die mit dem Phong-Beleuchtungsmodell gezeichnet wurden.

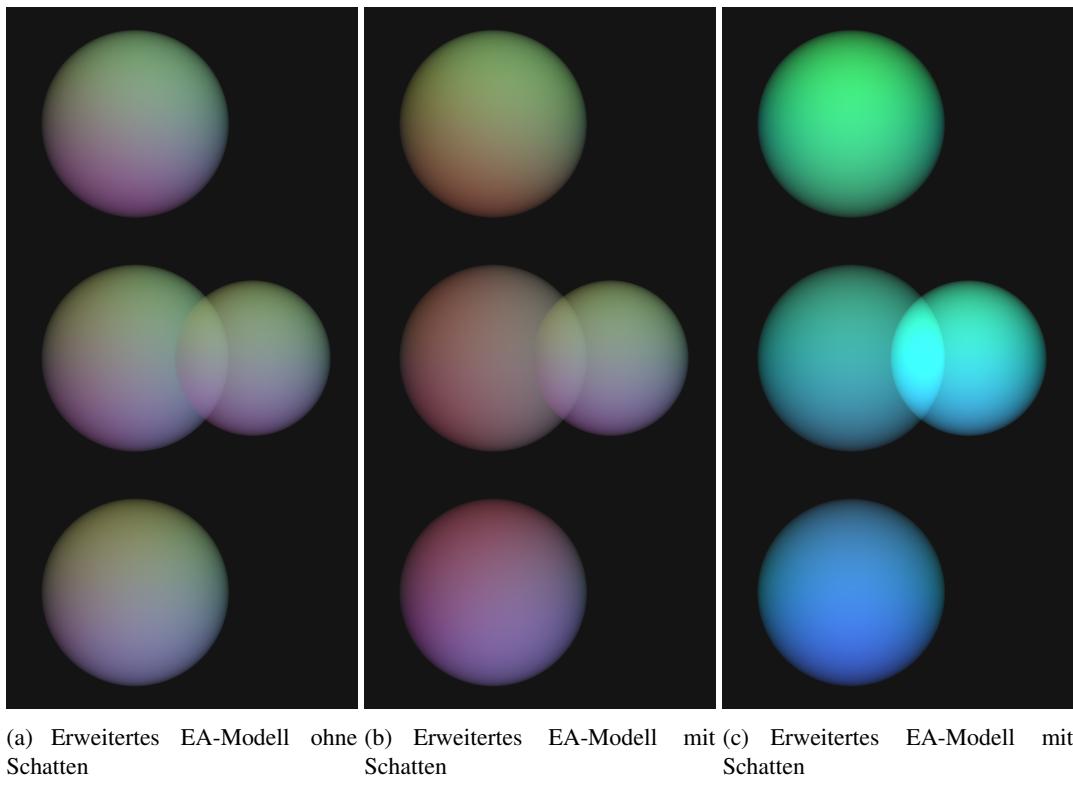


Abbildung 6.2: Die Abbildung zeigt die selbe Szene wie die Abbildung 6.1, gezeichnet mit dem erweiterten Emissions- und Absorptionsmodell

Die Bilder (a) und (b) in der Abbildung 6.2 zeigen die gleiche Szene wie zuvor, dieses mal wurde die Szene mit dem erweiterten Emissions- und Absorptionsmodell beleuchtet. Transparenzeffekte werden ebenso wie weitere Lichtquellen unterstützt, so dass verdeckte Inhalte für den Betrachter weiterhin sichtbar bleiben. Das Licht aller Lichtquellen flutet durch das innere der Kugeln, dadurch sind die Schatteneffekte weitaus weniger endgültig als bei dem Phong-Beleuchtungsmodell mit Schatten. Bei dem Bild (c) aus der Abbildung 6.2 wurde auf die rote Punktlichtquelle verzichtet und  $\Theta_{max} = 0.4$  gewählt. Das Licht der beiden Richtungslichtquellen wird durch die Kugeln abgeschwächt, trotzdem reicht die Lichtintensität aus um die mittlere Kugel zu beleuchten.

Bevor dir Geschwindigkeit des Algorithmus betrachtet wird, werden numerische Grenzfälle betrachtet.

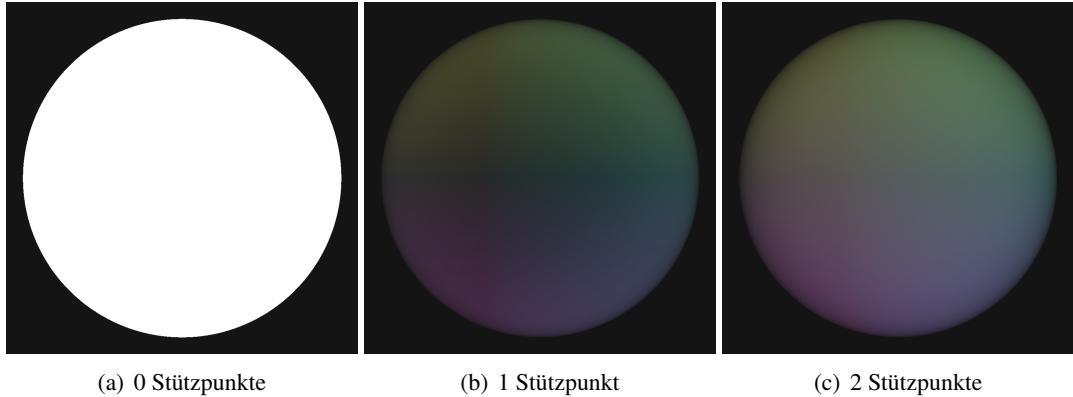


Abbildung 6.3: Die Abbildung zeigt jeweils eine Kugel, mit sehr niedrigen Anzahlen von Stützpunkten.

Dazu wurden in der Abbildung 6.3 wesentlich weniger Stützpunkte gewählt, als in den bereits gezeigten Abbildungen. Im Fall des Bildes (a) aus der Abbildung 6.3 wird gar kein Stützpunkt verwendet, was zur Folge hat, dass das Licht aus den Lichtquellen keinen Einfluss auf die Farbe der Kugel hat. In den Bildern (a) und (b) der gleichen Abbildung, zeigt sich eine deutliche Verbesserung der Lichtverteilung in der Kugel. Als Lichtquellen wurden wieder die gleichen Lichtquellen verwendet wie zuvor.

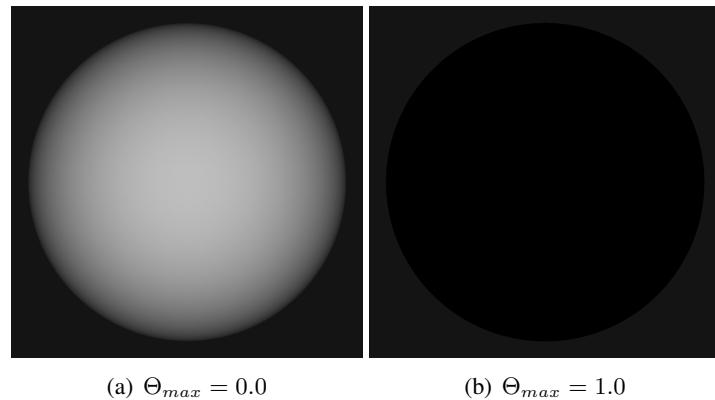


Abbildung 6.4: Die Abbildung zeigt jeweils eine Kugel, mit sehr niedrigen Anzahlen von Stützpunkten.

Die Abbildung 6.4 zeigt, was passiert wenn der Wert für die maximale Lichtdurchlässigkeit die Grenzfälle annimmt. Im Bild (a) wird  $\Theta_{max} = 0.0$  gesetzt und im Bild (b) auf 1.0, so dass die Kugel überhaupt kein Licht mehr durchlässt.

Bevor der Algorithmus quantitativ untersucht werden kann ist es nötig, zwei Bilder miteinander vergleichen zu können. Ein Gütwert muss die Übereinstimmung der beiden Bilder widerspiegeln. Wie ein solcher Gütwert erhoben wird, stellt der folgende Abschnitt vor.

## 6.1 SSIM

Der *structural similarity* Index, kurz *SSIM* ist eine Metrik, eine Abstandsfunktion. Diese weist zwei Bildern einen nicht negativen reellen Wert zu, der als Ähnlichkeit der Bilder interpretiert wird.

Ein solcher Gütwert, zur Abschätzung der Ähnlichkeit zweier Bilder kann auf verschiedene Weisen erhoben werden. Ein möglicher Ansatz besteht darin, eine Abstandsfunktion auf jedes Pixelpaar anzu-

wenden. Für jedes dieser Paare wird ein Abstand bestimmt und der Betrag oder das Quadrat von diesem wird als Fehler aufsummiert. Das Problem besteht darin, dass diese Verfahren Kontext unabhängig sind. Als Beispiel, wird eines der Bilder um wenige Pixel auf der X- oder Y-Achse verschoben, dann entsprechen die Pixelpaare keinen wirklichen Korrespondenzpunkten mehr. Der Fehler steigt drastisch an und die beiden Bilder als vollkommen unterschiedlich klassifizieren, obwohl sie sich im Prinzip nur geringfügig unterscheiden.

Beim SSIM dagegen wird der Fehler in einen Beleuchtungsterm  $l(x, y)$ , einen Kontrastterm  $c(x, y)$  und in einen Term  $s(x, y)$  zum Vergleich von Strukturen unterteilt. Zur Berechnung dieser Terme wird nicht ein einzelner Pixel, sondern ein kleines Fenster um einen Pixel zur Auswertung herangezogen. Diese drei Komponenten sind verhältnismäßig unabhängig voneinander. Eine Veränderung der Helligkeit des Bildes, beeinflusst die Struktur dieses beispielsweise nicht.  $x$  und  $y$  entsprechen dabei keinen Pixeln, sondern den beiden zu vergleichenden Signalen, in diesem Fall den Bildern.

Für die Berechnung des SSIM wird für jedes Signal der Mittelwert  $\mu$  und die Standartabweichung  $\sigma$  benötigt. Der Mittelwert  $\mu_x$  eines Signals  $x$  ergibt sich dabei als

$$\mu_x = \frac{1}{N} \sum_{i=1}^N x_i \quad (6.1)$$

und die Standartabweichung  $\sigma_x$  eines Signals entspricht dem folgendem Term:

$$\sigma_x = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2} \quad (6.2)$$

Zusätzlich wird zur Berechnung des SSIM eine weiterer Term  $\sigma_{xy}$  benötigt:

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y) \quad (6.3)$$

Das letzliche Maß lässt durch die folgende Gleichung errechnen:

$$\begin{aligned} SSIM(x, y) &= l(x, y) \cdot c(x, y) \cdot s(x, y) \\ SSIM(x, y) &= \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \end{aligned} \quad (6.4)$$

Der SSIM wird in dieser Arbeit zur Evaluation der verwendeten Approximationen genutzt.

## 6.2 Quantitative Evaluation

Zur Auswertung wurden zwei Datensätze in verschiedenen Szenen gezeichnet und die Rechenzeiten erhoben. Die zur Evaluation erhoben Daten wurden mit einem Core i7-4700HQ CPU (2.40GHz x 8) berechnet und die Bilder mit einer Auflösung von  $512 \times 512$  erzeugt. Bei dem ersten Datensatz handelt es sich um das 1UUN-Protein, das mit 2758 Partikeln dargestellt wird. Der zweite Datensatz wurde mit einem Zufallsgenerator generiert und besteht aus 73642 Partikeln. Geschwindigkeit und Güte des Algorithmus werden in Abhängigkeit der Stützpunktanzahl und dem Parameter  $N$ , für die beschleunigte Variante evaluiert. Die Abkürzung GT steht im weiteren Textverlauf für Ground-Truth und bezeichnet eine Bildsynthese ohne das Beschleunigungsverfahren.

Es wird das 1UUN-Protein betrachtet, die Abbildung 6.6 zweigt zwei Abbildungen des Proteins, gezeichnet mit zwei unterschiedlichen Parameter Einstellungen. Das Bild auf der linken Seite wurde mit einer sehr niedrigen Anzahl von Stützpunkten generiert und verwendet nur für die erste, vom Sichtstrahl geschnittene Kugel, das erweiterte Beleuchtungsmodell. Das zweite Bild auf der rechten Seite wurde mit 100 Stützpunkten je geschnittener Kugel generiert und verwendet für alle Kugeln, geschnitten vom Sichtstrahl, das erweiterte Beleuchtungsmodell. Die Anzahl der Lichtquellen ist ein Multiplikator der Sekundärstrahlanzahl, da für jeden Stützpunkt zu jeder Lichtquelle ein Sekundärstrahl in die Szene geschossen wird. Die Testszenen wurden mit jeweils zwei Richtungslichtquellen gezeichnet. Die erste Strahlt grünes Licht von oben nach unten in die Szene und die zweite blaues Licht von unten nach oben. Die zu diesen Datensatz erhobenen Rechenzeiten sind im Folgenden Diagramm abgebildet:

In der Abbildung 6.7 stellt die Y-Achse benötigte Rechenzeit in Sekunden dar und die X-Achse bildet die Anzahl der verwendeten Stützpunkte ab. Die Szene wurde für verschiedene Stufen der Beschleunigungsvariante, in Abhängigkeit der Stützpunktanzahl gezeichnet. Die Abbildung zeigt einen linearen Anstieg der Rechenzeiten in Abhängigkeit der Stützpunktzahl. Zudem steigt der Anstieg mit wachsendem  $N$ . Der Datensatz des 1UUN-Proteins umfasst nur wenige Partikel, deshalb sind die Ground-Truth Ergebnisse mit  $N = 5$  nahezu identisch.

Die Abbildung 6.8 zeigt ein ähnliches Ergebnis für den Zufällig generierten Datensatz. In diesem wurden wesentlich mehr Partikel betrachtet, weshalb die Unterschiede zwischen den einzelnen Parameter-einstellungen stärker zur Geltung kommen. Die auftretenden Schwankungen bei den Messungen des 1UUN-Proteins kommen in der Abbildung 6.7 vermutlich auf Grund der wesentlich kleineren Zeitskala der Y-Achse mehr zur Geltung, als bei der Abbildung 6.8.

Zur Evaluation der beschleunigten Variante, wurden die erzeugten Bilder hinsichtlich ihrer Ähnlichkeit untersucht. Diese wurde mit Hilfe des bereits vorgestellten *structural similarity*-Index, kurz SSIM ermittelt. Dabei wird die Übereinstimmung der Ergebnisse, im Verhältnis zum Ground-Truth Bildes für jeden Farbkanal einzeln in Prozent angeben. Die nachfolgenden Tabellen wurden um den Mittelwert der Übereinstimmung aller Farbkanäle ergänzt:

1UUN-Protein				
$N$	Rot	Grün	Blau	Mittelwert
1	89.8914%	96.0094%	95.1042%	93.6683%
2	98.8771%	99.6152%	99.505%	99.3324%
3	99.9335%	99.9758%	99.9708%	99.9601%
5	99.9996%	99.9998%	99.9998%	99.9997%

Tabelle 6.1: Die Tabelle stellt die ermittelten SSIM-Indizes für das 1UUN-Protein dar.

cool random				
$N$	Rot	Grün	Blau	Mittelwert
1	77.0101%	84.1127%	70.3878%	77.1702%
2	92.3301%	95.1143%	88.0297%	91.8247%
5	99.9073%	99.9413%	99.9012%	99.9166%
10	99.9995%	99.9997%	99.9997%	99.9996%

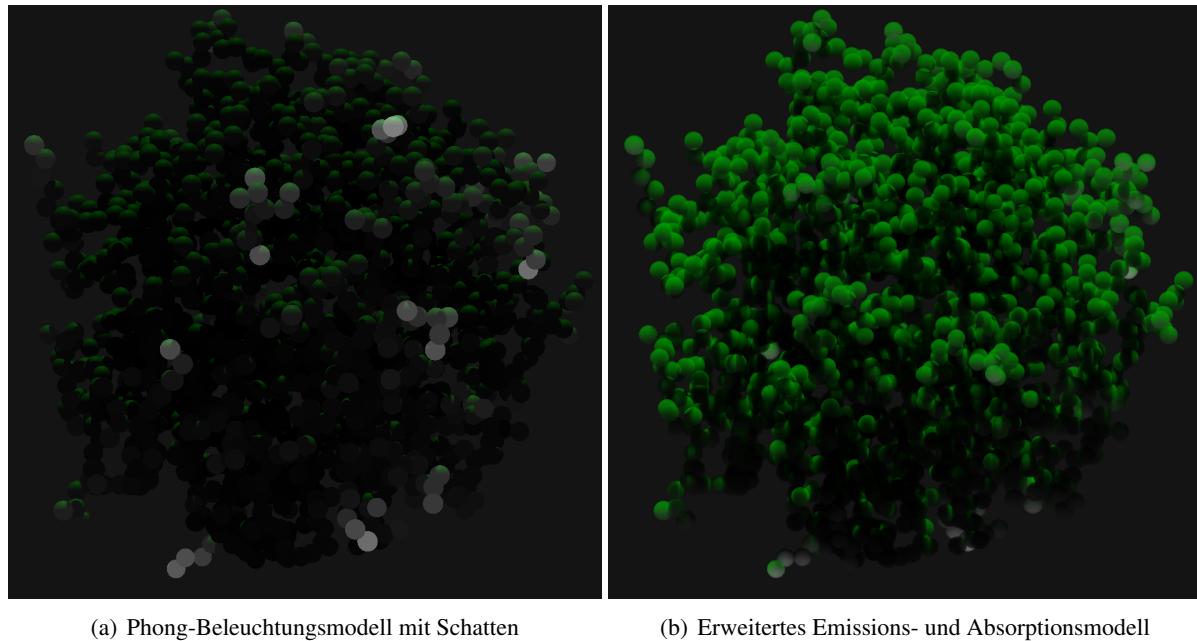
Tabelle 6.2: Die Tabelle stellt die ermittelten SSIM-Indizes für den zufällig generierten Datensatz dar.

Die Tabellen 6.1 und 6.2 zeigen deutlich, dass die Ergebnisse der beschleunigten Variante des Algorithmus, ab einem gewissen  $N$  sich kaum noch von den Ground-Truth Bildern unterscheiden. Ein großer Unterschied besteht jedoch in der Geschwindigkeit der Beleuchtungsberechnung.

### 6.3 Fazit

Ein Problem der Implementierung besteht darin, dass es zu Artefakten kommt, wenn sich Kugeln überlappen. Um dieses Problem zu lösen, könnten alle Überlappendenden Kugeln zu Objekten zusammengefasst werden. Bei einem Schnitt mit einem solchen Objekt, müsste die Berechnung an den Schnittebenen der Kugeln einzeln durchgeführt werden.

Auf die Implementierung von Brechungseffekten wurde aus zwei folgenden Gründen bewusst verzichtet.



(a) Phong-Beleuchtungsmodell mit Schatten

(b) Erweitertes Emissions- und Absorptionsmodell

Abbildung 6.5: Abbildungen des 2BT9-Proteins, links mit dem Phong und rechts mit dem erweiterten Emissions- und Absorptionsmodell beleuchtet.

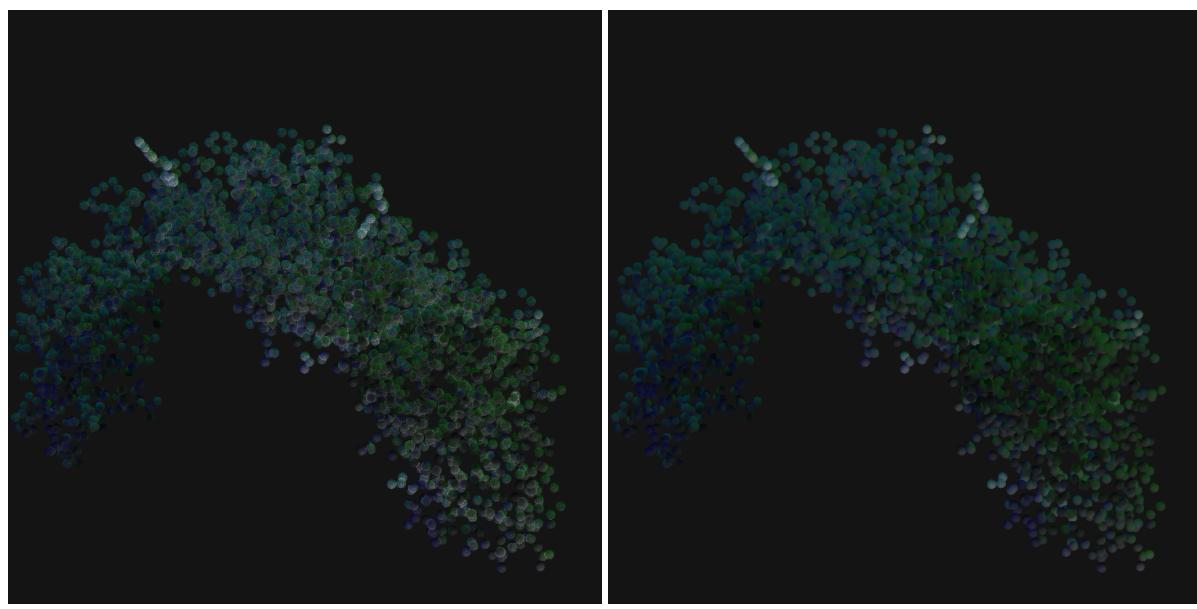
(a) 5 Stützpunkte und  $N = 1$ (b) 100 Stützpunkte und  $N = GT$ 

Abbildung 6.6: Abbildungen des 1UUN-Proteins, die mit verschiedenen Parametern erzeugt wurden.

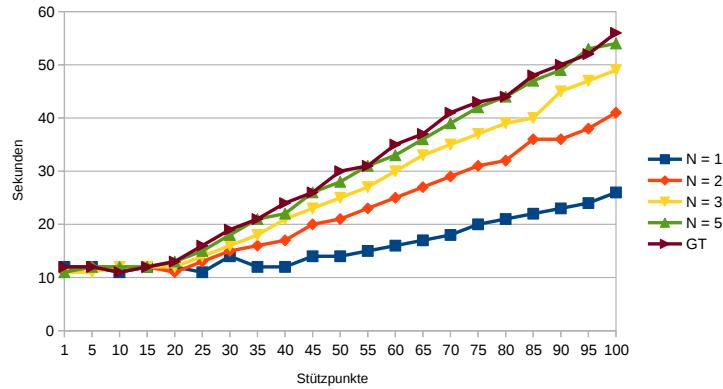


Abbildung 6.7: Das Diagramm stellt die Renderzeiten in Abhängigkeit der Stützpunktzahl und  $N$ , des Datensatzes 1UUN-Proteins dar.

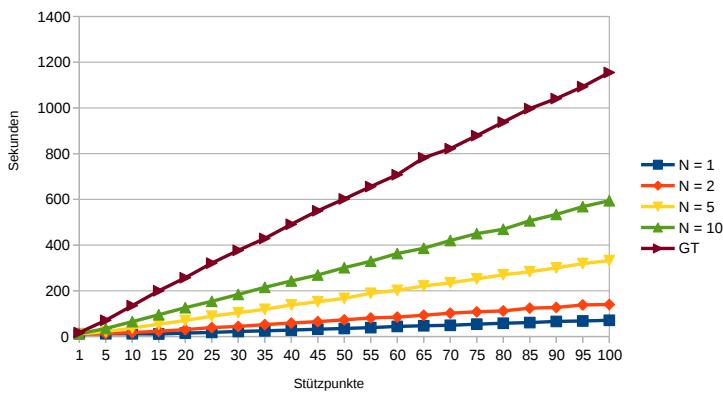


Abbildung 6.8: Das Diagramm stellt die Renderzeiten in Abhängigkeit der Stützpunktzahl und  $N$ , des Datensatzes cool\_random dar.

## Literaturverzeichnis

- [ASW13] AMENT, Marco ; SADLO, Filip ; WEISKOPF, Daniel: Ambient Volume Scattering. In: *IEEE Trans. Vis. Comput. Graph.* 19 (2013), Nr. 12, S. 2936–2945
- [Bli77] BLINN, James F.: Models of Light Reflection for Computer Synthesized Pictures. In: *SIGGRAPH Comput. Graph.* 11 (1977), Juli, Nr. 2, S. 192–198. – ISSN 0097–8930
- [Bli82] BLINN, James F.: Light Reflection Functions for Simulation of Clouds and Dusty Surfaces. In: *SIGGRAPH Comput. Graph.* 16 (1982), Juli, Nr. 3, S. 21–29. – ISSN 0097–8930
- [GKM<sup>+</sup>15] GROTTEL, S. ; KRONE, M. ; MULLER, C. ; REINA, G. ; ERTL, T.: MegaMol—A Prototyping Framework for Particle-based Visualization. In: *Visualization and Computer Graphics, IEEE Transactions on* 21 (2015), Feb, Nr. 2, S. 201–214. – ISSN 1077–2626
- [Hub96] HUBER, Martin: Warum ist denn  $\exp(x^2)$  nicht elementar integrierbar? / Universität Zürich. 1996. – Forschungsbericht
- [JKRY12] JÖNSSON, Daniel ; KRONANDER, Joel ; ROPINSKI, Timo ; YNNERMAN, Anders: Historygrams: Enabling Interactive Global Illumination in Direct Volume Rendering using Photon Mapping. In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 18 (2012), Nr. 12, S. 2364–2371
- [JPP98] JUNG, Moon-Ryul ; PARK, Hyunwoo ; PAIK, Doowon: An Analytical Ray Casting of Volume Data. In: *Pacific Conference on Computer Graphics and Applications*, IEEE Computer Society, 1998. – ISBN 0–8186–8620–0, S. 79–86
- [JSYR14] JÖNSSON, Daniel ; SUNDÉN, Erik ; YNNERMAN, Anders ; ROPINSKI, Timo: A Survey of Volumetric Illumination Techniques for Interactive Volume Rendering. In: *Computer Graphics Forum* 33 (2014), Nr. 1, S. 27–51
- [Kaj86] KAJIYA, James T.: The Rendering Equation. In: *SIGGRAPH Comput. Graph.* 20 (1986), August, Nr. 4, S. 143–150. – ISSN 0097–8930
- [KPB12] KROES, Thomas ; POST, Frits H. ; BOTHA, Charl P.: Exposure render: an interactive photo-realistic volume rendering framework. In: *PLoS ONE* 7 (2012), 07, Nr. 7. – 10.1371/journal.pone.0038586
- [KPH<sup>+</sup>03] KNİSS, Joe ; PREMOZE, Simon ; HANSEN, Charles D. ; SHIRLEY, Peter ; MCPHERSON, Allen: A Model for Volume Lighting and Modeling. In: *IEEE Trans. Vis. Comput. Graph.* 9 (2003), Nr. 2, S. 150–162
- [Max95] MAX, Nelson: Optical Models for Direct Volume Rendering. In: *IEEE Transactions on Visualization and Computer Graphics* 1 (1995), Juni, Nr. 2, S. 99–108. – ISSN 1077–2626
- [SGG15] STAIB, Joachim ; GROTTEL, Sebastian ; GUMHOLD, Stefan: Visualization of Particle-based Data with Transparency and Ambient Occlusion, 2015, S. to appear
- [SPH<sup>+</sup>09] SCHOTT, Mathias ; PEGORARO, Vincent ; HANSEN, Charles ; BOULANGER, KÃ©vin ; BOUATOUCH, Kadi: A Directional Occlusion Shading Model for Interactive Direct Volume Rendering. In: *Computer Graphics Forum* (2009). – ISSN 1467–8659
- [WBSS04] WANG, Zhou ; BOVIK, Alan C. ; SHEIKH, Hamid R. ; SIMONCELLI, Eero P.: Image Quality Assessment: From Error Visibility to Structural Similarity. In: *IEEE TRANSACTIONS ON*

*IMAGE PROCESSING* 13 (2004), Nr. 4, S. 600–612

# Abbildungsverzeichnis

3.1	Schematische Darstellung des Abbildungsvorgangs der <i>camera obscura</i> . . . . .	8
4.1	Die Abbildung zeigt einen primären Sichtstrahl, der ein Partikel schneidet. Auf der rechten Seite der Abbildung symbolisiert eine kleine Sonne die Hintergrundbeleuchtung und auf der linken ein Auge den Sensor. . . . .	13
4.2	Die Notation ist dieselbe wie in der Abbildung 4.1. Diese Abbildung ergänzt das Modell um eine Punktlichtquelle und es werden drei sekundäre Strahlen zu dieser dargestellt. . . . .	15
5.1	UML-Diagramm des Programms, die Hauptklasse heißt <i>zikade</i> . . . . .	21
6.1	Abbildungen einer einfachen Szene gezeichnet mit dem Phong-Beleuchtungsmodell, einmal ohne und einmal mit globalen Schatteneffekten. Zusätzlich, wird die selbe Szene, gezeichnet mit dem einfachen Emissions- und Absorptionsmodell dargestellt. . . . .	23
6.2	Die Abbildung zeigt die selbe Szene wie die Abbildung 6.1, gezeichnet mit dem erweiterten Emissions- und Absorptionsmodell . . . . .	24
6.3	Die Abbildung zeigt jeweils eine Kugel, mit sehr niedrigen Anzahlen von Stützpunkten. . . . .	25
6.4	Die Abbildung zeigt jeweils eine Kugel, mit sehr niedrigen Anzahlen von Stützpunkten. . . . .	25
6.5	Abbildungen des 2BT9-Proteins, links mit dem Phong und rechts mit dem erweiterten Emissions- und Absorptionsmodell beleuchtet. . . . .	29
6.6	Abbildungen des 1UUN-Proteins, die mit verschiedenen Parametern erzeugt wurden. . . . .	29
6.7	Das Diagramm stellt die Renderzeiten in Abhängigkeit der Stützpunkanzahl und $N$ , des Datensatzes 1UUN-Proteins dar. . . . .	30
6.8	Das Diagramm stellt die Renderzeiten in Abhängigkeit der Stützpunkanzahl und $N$ , des Datensatzes cool_random dar. . . . .	30



## Tabellenverzeichnis

6.1 Die Tabelle stellt die ermittelten SSIM-Indizes für das 1UUN-Protein dar. . . . .	27
6.2 Die Tabelle stellt die ermittelten SSIM-Indizes für den zufällig generierten Datensatz dar.	27



## Danksagung

Die Danksagung...

