

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, 20. Mai 2015

Aufgabenstellung

Die Darstellung von Partikeldaten mittels Kugelglyphen ist in 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 20. Mai 2015

Josef Schulz

Inhaltsverzeichnis

1	Einleitung	3
1.1	Motivation	3
2	Verwandte Arbeiten	5
3	Grundlagen	7
3.1	Strahl und Kugelgleichung	7
3.2	Die perspektivische Kamera	8
4	Darstellungsmöglichkeiten	11
5	Beleuchtungsmodelle	13
5.1	Einfaches Beleuchtungs-Modell	14
5.2	Erweitertes Beleuchtungs-Modell	15
6	Implementierung	17
7	Evaluation	19
7.1	Qualitative Evaluation	19
7.2	Quantitative Evaluation	19
7.2.1	SSIM	19
7.2.2	Beispiel	20
7.3	Fazit	22
	Literaturverzeichnis	23
	Abbildungsverzeichnis	25
	Tabellenverzeichnis	27

1 Einleitung

1.1 Motivation

Ein Partikel ist ein beliebiges geometrisches Objekt, das mindestens durch eine Position beschrieben wird. Neben der Position sind beliebige weitere Attribute zur Beschreibung eines Partikels vorstellbar. Mengen von Partikeln lassen sich zu Datensätzen zusammenfassen. In einen solchen, sind die einzelnen Partikel von einander unabhängig. Anders verhält es sich bei Simulationen, in diesen werden Partikel zur Beschreibung physikalischer Vorgänge verwendet und bedingen sich sehr wohl. Neben Simulationen entstehen partikelbasierte Datensätze in einer Vielzahl von Messvorgängen.

Erst durch eine geeignete Darstellung lassen sich diese optimal auswerten und analysieren. Das Ziel von Visualisierungen ist die Unterstützung der menschlichen Wahrnehmung bei der Auswertung der erhobenen Daten. Dabei geht es nicht um eine bloße optische Verschönerung der Darstellung, sondern darum bestimmte Eigenschaften in den Fokus des Betrachters zu rücken und Tendenzen hervorzuheben. Die Darstellung von dreidimensionalen Partikeln auf einer zweidimensionalen Ebene ist mit Problemen verbunden. Diese sollen im Folgenden näher betrachtet werden.

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 besteht in der Schätzung der Entfernung über die Epipolarometrie. Bei der Betrachtung eines zweidimensionalen Bildes kann diese Variante lediglich die Bildschirmenebene erfassen, jedoch nicht die Tiefe der abgebildeten Objekte. Die Objektiefen 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 sich 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 Strukturformen, 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. Eine weitere vom Menschen, zur Bestimmung von Objektiefen genutzte Möglichkeit, ist die der Rekonstruktion von Lichtwegen.

Lokale Beleuchtungsmodelle eignen sich zur Berechnung von Schattierung, unterstützen die Wahrnehmung der Tiefe jedoch nur bedingt. Es existiert eine Reihe von Beleuchtungsmodellen die die Szene global beleuchten. Der Einsatz solcher Modelle stellt eine bessere Unterstützung der Wahrnehmung dar. Existierenden 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 Vorgelegt worden, wird das Verfahren Qualitativ und Quantitativ evaluiert und es wird auf mögliche Erweiterungen im Schlusswort eingegangen.

2 Verwandte Arbeiten

Die Arbeit [JSYR14] fasst verschiedenen 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 wird die Arbeit mit der Vorstellung anwendbarer Lösungsverfahren.

In der Publikation [JPP98] wird ein zum Teil analytisches Verfahren vorgestellt. Der Algorithmus wird zur Darstellung von Voxeligittern eingesetzt. Hierbei wird jedem Voxel eine Dichte zugeordnet, aus welcher ein Wert für die Transparenz abgeleitet wird. Hinter dem Voxeligitter 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 Strahlen 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⁺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⁺09] dar, welche hier nur der Vollständigkeit halber erwähnt werden soll.

3 Grundlagen

Der verwendete Algorithmus basiert auf dem Raycast-Verfahren. Grundlage 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 Strahlsegment 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 sind 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 einem Richtungsvektor \vec{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 p auf der Oberfläche O , wenn sie den 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 Schnittpunkte 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 wird spiegelverkehrt aufgenommen wird.

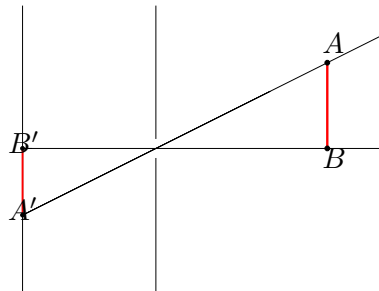


Abbildung 3.1: Schematische Darstellung des Abbildungsvorgangs der *camera obscura*

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 Darstellungsmöglichkeiten

Es gibt eine Reihe von Verfahren um mit Hilfe eines Computers geometrische Objekte auf eine Bildfläche abzubilden. Die wohl am häufigsten in der Computergrafik eingesetzte Technik ist der Rasterisierung. Um eine beliebige beschriebene geometrische Form mit Hilfe dieser Technik abzubilden ist eine Zerlegung in ein polygonales Netz erforderlich. Das bedeutet, dass die eigentliche Geometrie in Primitiven, wie Punkte, Linien oder Dreiecke zerlegt wird. Diese werden dann auf die Bildschirmfläche abgebildet. Im Anschluss werden die sogenannten Fragmente, die auf die Pixel abgebildeten Polygone, nach einem beliebigen Verfahren zur eigentlichen Pixelfarbe verrechnet. Ein großer Vorteil dieser Methode ist die Geschwindigkeit, mit der die Bilder aus der Geometrie erzeugt werden können.

5 Beleuchtungsmodelle

Der Ausgangspunkt dieses Kapitels 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 in dieser Arbeit vernachlässigt. Die Gleichung der Beleuchtungsberechnung unterteilt sich in zwei Summanden:

$$I = I_A + I_E \quad (5.1)$$

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:

$$I_A = I_B \cdot T(D) \quad (5.2)$$

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

$$T(s) = \exp\left(-\int_0^s \tau(t) dt\right) \quad (5.3)$$

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 κ abgebildet. Zur Vereinfachung der Formeln wird eine weitere Funktion definiert, welche die Länge aus der Differenz einer hinteren und einer vorderen Position berechnet:

$$T'(t_n, t_f) = \exp\left(-\int_{t_n}^{t_f} (\lambda\kappa) dt\right) \quad (5.4)$$

In der Funktion 5.4 steht die Variable t_n für den Eintrittspunkt in das Volumen und t_f für den Austrittspunkt. Da das Produkt von λ und κ ein konstanter Wert ist, kann es aus dem Integral herausgezogen werden. Somit existiert eine analytisch Lösung:

$$T'(t_n, t_f) = \exp\left(-\lambda\kappa \cdot \int_{t_n}^{t_f} dt\right) = e^{-\lambda\kappa \cdot (t_f - t_n)} \quad (5.5)$$

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

$$I_E = \int_0^D g(s) \cdot T'(s, D) ds \quad (5.6)$$

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 Volumen abgeschwächt. Die Formeln für die Absorption und die Emission wird in die Gleichung 5.1 eingesetzt und es ergibt sich das komplette Modell für die Beleuchtungsberechnung in der Form:

$$I = I_B \cdot T(D) + \int_0^D g(s) \cdot T'(s, D) ds \quad (5.7)$$

Die Gleichung 5.7 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.

5.1 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.

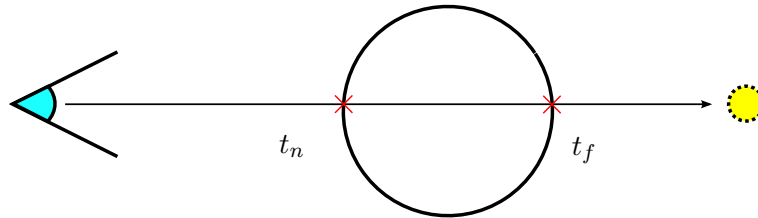


Abbildung 5.1: Die Abbildung zeigt einen primären Sichtstrahl, der ein Partikel schneidet. Auf der rechten Seite der Abbildung symbolisiert eine kleine Sonne die Hintergrundbeleuchtung und auf der linken ein Auge den Sensor.

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 κ abbildet. Zunächst wird die Emissionsgleichung mit den Eintritts- und Austrittspunkten parametrisiert:

$$I_E = \int_{t_n}^{t_f} T'(t, t_f) \cdot g(t) dt \quad (5.8)$$

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

$$I_E = \int_{t_n}^{t_f} \exp(-\lambda\kappa(t_f - t)) \cdot \lambda\kappa I_c dt \quad (5.9)$$

Für die Gleichung 5.9 lässt sich mit Hilfe der Substitutionsregel eine analytische Lösung finden:

$$\begin{aligned} I_E &= I_C \cdot (1 - e^{-\lambda\kappa(t_f - t_n)}) = I_C \cdot (1 - T'(t_n, t_f)) \\ I_E &= I_C \cdot \Theta(t_n, t_f) \end{aligned} \quad (5.10)$$

Wird diese Lösung und die Lösung 5.5 in die Gleichung 5.7 eingesetzt, dann ergibt sich die gesamte Gleichung zur Beleuchtung eines Partikels:

$$I = I_B \cdot T'(t_n, t_f) + I_C \cdot \Theta(t_n, t_f) \quad (5.11)$$

5.2 Erweitertes Beleuchtungs-Modell

Kern dieser Arbeit ist eine Erweiterung des eben vorgestellten Beleuchtungsmodells. Neben der Hintergrundbeleuchtung wird die Beleuchtungsberechnung für eine beliebige Anzahl von Punkt- und Richtungslichtquellen erweitert. Zur Herleitung der Gleichung werden die Lichtquellen auf die Hintergrundbeleuchtung und eine Punktlichtquelle beschränkt. Anschließend wird die gefundene Lösung wieder für beliebig viele Lichtquellen verallgemeinert. Die folgende Abbildung soll zunächst die Vereinfachung veranschaulichen.

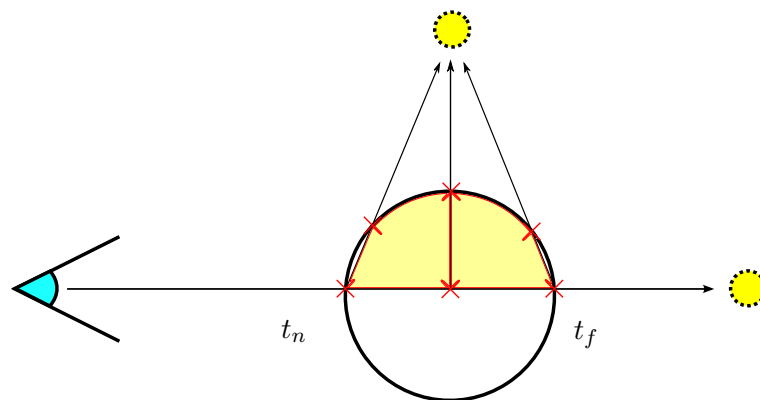


Abbildung 5.2: Die Notation ist die selbe wie in der Abbildung 5.1. Diese Abbildung ergänzt das Modell um eine Punktlichtquelle und es werden drei sekundäre Strahlen zu dieser dargestellt.

Die Grundlage für das erweiterte Beleuchtungs-Modell ist selbe Gleichung 5.7, wie für das vorherige. Diese wird zur besseren Übersicht an dieser Stelle mit den Notationen aus dem vorhergehenden Abschnitt wiederholt:

$$I = I_B \cdot T'(t_n, t_f) + \int_{t_n}^{t_f} g(t) \cdot T'(t, t_f) dt \quad (5.12)$$

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 Punktlichtquelle 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:

$$g(t) = I_L \cdot T'(0, l(t)) + I_c \cdot \Theta(0, l(t)) \quad (5.13)$$

Die Variable I_L entspricht der Intensität der Punktlichtquelle. 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 Gleichung 5.13 in die Gleichung 5.12 eingesetzt, ergibt sich daraus ein Integral für das keine analytische Lösung existiert:

$$I = I_B \cdot T'(t_n, t_f) + \int_{t_n}^{t_f} (I_L \cdot T'(0, l(t)) + I_c \cdot \Theta(0, l(t))) \cdot T'(t, t_f) dt \quad (5.14)$$

Die Lösung der Gleichung wird mit Hilfe der Riemann Summe numerisch approximiert. 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:

$$\int_0^D f(x) dx \approx \sum_{i=1}^n f(x_i) \Delta x \quad (5.15)$$

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 Δx skaliert und auf das Ergebnis aufsummiert.

Demnach lässt sich die Formel 5.12 zu der folgenden Gleichung vereinfachen:

$$I = I_B \cdot T'(t_n, t_f) + \sum_{t=t_n}^{t_f} g(t) \cdot T'(t, t_f) \Delta t \quad (5.16)$$

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 Formelzeichen 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:

$$I = I_B \cdot T'(t_n, t_f) + \sum_{t=t_n}^{t_f} \left(\frac{1}{m} \cdot \sum_{i=1}^m I_{L_i} \cdot T'(0, l(t)) + I_c \cdot \Theta(0, l(t)) \right) \cdot T'(t, t_f) \Delta t \quad (5.17)$$

6 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 Durchlässigkeit zusammensetzt. Die maximale Durchlässigkeit wird dabei durch das Symbol $\Theta_{max} \in [0, 1]$ repräsentiert. Aus Θ_{max} wird für jede Kugel der Wert für κ nach der folgenden Vorschrift berechnen:

$$\kappa = -\frac{1}{\lambda \cdot 2r} \ln(1 - \Theta_{max}) \quad (6.1)$$

Der Parameter λ wird für alle Kugeln global gewählt und liegt im Intervall $(0, 1]$. Neben der Hintergrundbeleuchtung gibt es die Möglichkeit beliebig viele Punkt- und Richtungslichtquellen zu definieren. Eine Punktlichtquelle besitzt neben ihrer Position eine Farbe. Eine Richtungslichtquelle besitzt keine Position, sondern einen normierten Richtungsvektor und eine Farbe als Parameter.

Die Auflösung eines Bildes entspricht $W \times H$. 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 umgesetzt.

Für die Sichtstrahlen wird die Beleuchtung durch das erweiterte Beleuchtungsmodell berechnet. Dazu muss für jeden Strahl ein Schnittest mit allen Kugeln durchgeführt 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. 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. Diese werden als Sekundärstrahlen bezeichnet. Für jeden Sekundärstrahl wird das Licht, der entsprechenden Lichtquelle, als Hintergrundbeleuchtung betrachtet. Die Intensität dieses Lichtes wird entlang des Strahls wiederum durch Partikel, aber dieses mal nach dem vereinfachten Beleuchtungsmodell abgeschwächt. Um dies zu bewerkstelligen ist es nötig für jeden Sekundärstrahl abermals den Schnitt mit allen Kugeln zu bestimmen und es wieder ist eine Sortierung der Ergebnismenge entlang des Strahls erforderlich.

Die Geschwindigkeit des Algorithmus hängt entscheidend von der Auflösung des Bildes und der damit

verbunden Anzahl von Primärstrahlen ab. Entscheidend für Berechnungsdauer ist die Wahl der Stützpunktzahl, da für jeden ein Sekundärstrahl ein Schnitt mit allen Kugeln geprüft werden muss.

An dieser Stelle wird ein weiterer 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. Im nächsten Kapitel wird der Einfluss, den diese Vereinfachung auf das Ergebnisbild hat evaluiert.

7 Evaluation

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. Das einfache Modell kann komplett analytisch gelöst werden. Für das erweiterte ist eine numerische Approximation von Nöten, dazu wurde das Integral in Riemann Summen zerlegt. Es wurde eine Vereinfachung zur Beschleunigung der Bildsynthese vorgestellt. Diese wird im Folgenden mit Ergebnissen verglichen, die ohne Vereinfachung erzeugt worden sind.

7.1 Qualitative Evaluation

7.2 Quantitative Evaluation

7.2.1 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 Gütewert, zur Abschätzung der Ähnlichkeit zweier Bilder kann auf verschiedene Weisen erhoben werden. Viele Ansätze basieren auf einem Quantitativen Vergleich der Bilder. Für jedes Paar von Pixeln wird ein Abstand bestimmt und der Betrag oder das Quadrat von diesem wird als Fehler aufsummiert. Das Problem bei einem solchen Vorgehen besteht darin, dass ein solches Verfahren Kontext unabhängig sind.

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 μ und die Standardabweichung σ benötigt. Der Mittelwert μ_x eines Signals x ergibt sich dabei als

$$\mu_x = \frac{1}{N} \sum_{i=1}^N x_i \quad (7.1)$$

und die Standardabweichung σ_x eines Signals entspricht dem folgenden Term:

$$\sigma_x = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2} \quad (7.2)$$

Zusätzlich wird zur Berechnung des SSIM ein weiterer Term σ_{xy} benötigt:

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y) \quad (7.3)$$

$$SSIM(x, y) = l(x, y) \cdot c(x, y) \cdot s(x, y) \quad (7.4)$$

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (7.5)$$

7.2.2 Beispiel

Zur Auswertung wurden zwei Datensätze in verschiedenen Szenen gezeichnet und die Rechenzeiten erhoben. Die zur Evaluation erhobenen Daten wurden mit einem Core i7-4700HQ CPU (2.40GHz x 8) berechnet und die Bilder mit einer Auflösung von 512×512 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 Stützpunktzahl 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.

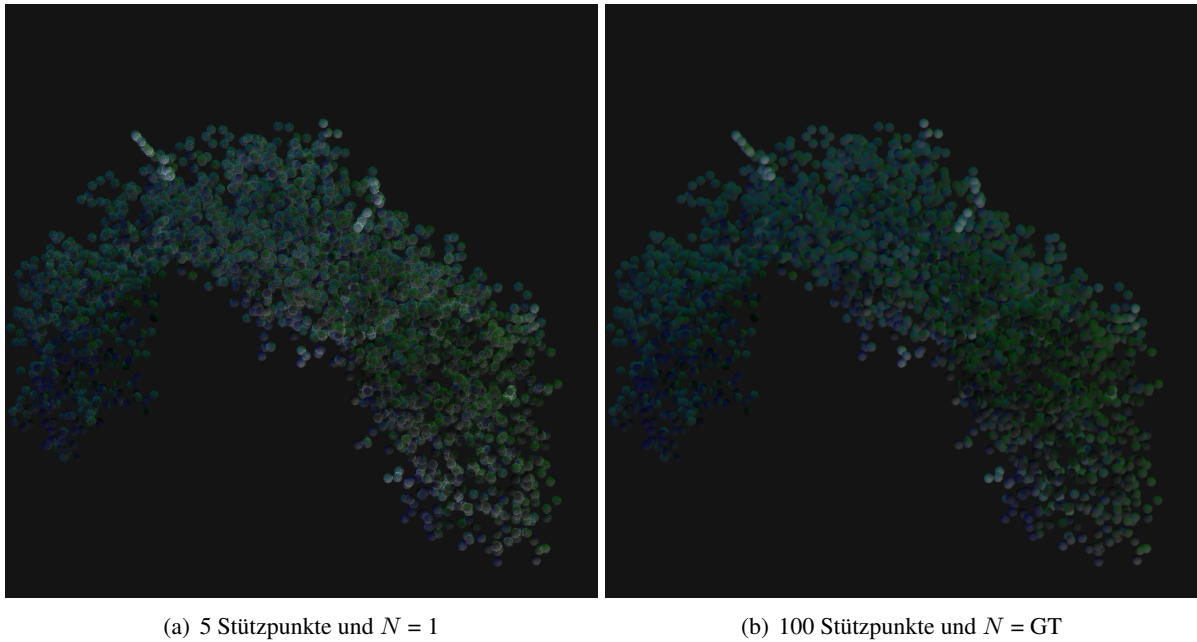


Abbildung 7.1: Abbildungen des 1UUN-Proteins, die mit verschiedenen Parametern erzeugt wurden.

Es wird das 1UUN-Protein betrachtet, die Abbildung 7.1 zeigt 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:

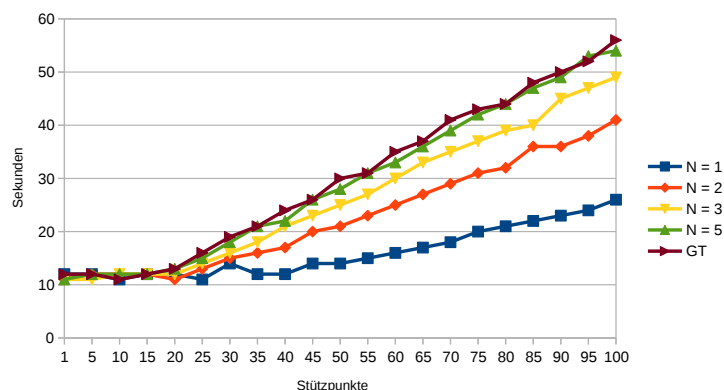


Abbildung 7.2: Das Diagramm stellt die Renderzeiten in Abhängigkeit der gewählten Parameter 1UUN-Proteins dar.

In der Abbildung 7.2 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 Stützpunkanzahl gezeichnet. Die Abbildung zeigt einen linearen Anstieg der Rechenzeiten in Abhängigkeit der Stützpunkanzahl. 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.

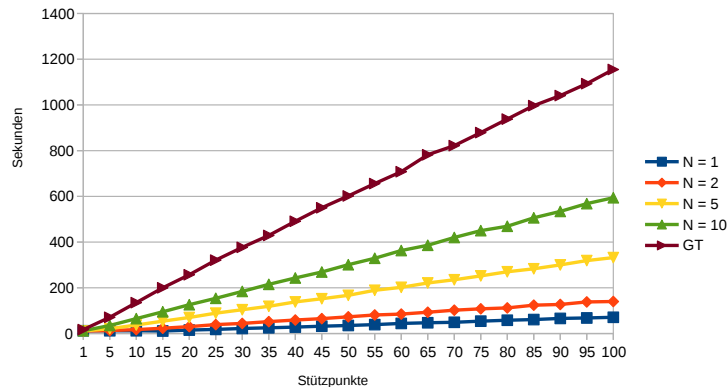


Abbildung 7.3: Das Diagramm stellt die Renderzeiten in Abhängigkeit der Stützpunkanzahl und N , des Datensatzes pdb.1UUN dar.

Die Abbildung 7.3 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 7.2 vermutlich auf Grund der wesentlich kleineren Zeitskala der Y-Achse mehr zur Geltung, als bei der Abbildung 7.3.

Zur Evaluation der beschleunigten Variante, wurden die erzeugten Bilder hinsichtlich ihrer Ähnlichkeit untersucht. Diese wurde mit Hilfe des *structural similarity*-Index, kurz SSIM ermittelt. Dieser wurden nur für Bilderpaare mit 100 Stützpunkten bestimmt. Bei niedrigeren Anzahlen von Stützpunkten würde der Fehler, der Approximation das Ergebnis zu sehr verfälschen. 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 7.1: 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 7.2: Die Tabelle stellt die ermittelten SSIM-Indizes für den zufällig generierten Datensatz dar.

Die Tabellen 7.1 und 7.2 zeigen deutlich, dass die Ergebnisse der beschleunigten Variante des Algorithmus, ab einen gewissen N sich kaum noch von den Ground-Truth Bildern unterscheiden. Ein großer Unterschied besteht jedoch in der Geschwindigkeit der Beleuchtungsberechnung.

7.3 Fazit

Literaturverzeichnis

- [ASW13] AMENT, Marco ; SADLO, Filip ; WEISKOPF, Daniel: Ambient Volume Scattering. In: *IEEE Trans. Vis. Comput. Graph.* 19 (2013), Nr. 12, S. 2936–2945
- [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⁺15] GROTTTEL, 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
- [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⁺03] KNISS, 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
- [SPH⁺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
5.1	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.	14
5.2	Die Notation ist die selbe wie in der Abbildung 5.1. Diese Abbildung ergänzt das Modell um eine Punktlichtquelle und es werden drei sekundäre Strahlen zu dieser dargestellt. . .	15
7.1	Abbildungen des 1UUN-Proteins, die mit verschiedenen Parametern erzeugt wurden. . .	20
7.2	Das Diagramm stellt die Renderzeiten in Abhängigkeit der gewählten Parameter 1UUN-Proteins dar.	21
7.3	Das Diagramm stellt die Renderzeiten in Abhängigkeit der Stützpunkanzahl und N , des Datensatzes pdb.1UUN dar.	21

Tabellenverzeichnis

7.1	Die Tabelle stellt die ermittelten SSIM-Indizes für das 1UUN-Protein dar.	22
7.2	Die Tabelle stellt die ermittelten SSIM-Indizes für den zufällig generierten Datensatz dar.	22

Danksagung

Die Danksagung...

Erklärungen zum Urheberrecht

Hier soll jeder Autor die von ihm eingeholten Zustimmungen der Copyright-Besitzer angeben bzw. die in Web Press Rooms angegebenen generellen Konditionen seiner Text- und Bildübernahmen zitieren.

