Bachelorarbeit
Surface Caching und Lightmapping

Markus Pawellek markuspawellek@gmail.com

Inhaltsverzeichnis

1 Grundlagen und Methoden

In den folgenden Kapiteln wird eine grundlegende Einführung und Definition der Verfahren gegeben, die für den weiteren Verlauf dieser Arbeit von Belang sind. Viele dieser Themen können hier nur angerissen werden, da ihre komplette Behandlung den Rahmen und das Ziel des Themas verfehlen würde. Für eine genauere Einführung in die einzelnen Themengebiete, wird dem Leser geraten sich mit den genannten Quellen auseinander zu setzen.

1.1 Szenenrepresentation

Um realistische Bilder zu generieren, benötigen wir als Eingabe gewisse Daten der physikalischen Objekte, die durch den Beobachter oder auch die Kamera betrachtet werden. Für die später betrachteten Algorithmen ist dabei vor allem das Verhalten von Licht auf den Oberflächen dieser Objekte wichtig. Der Einfachheit halber wollen wir davon ausgehen, dass das Licht nur in den oberen Schichten eines Körpers mit dessen Material wechselwirkt. Für uns ist es also ausreichend, nur die Oberflächen der gegebenen Objekte und nicht deren Inneres zu beschreiben.

Die Oberflächen realer physikalischer Objekte sind im Allgemeinen beliebig geformt und können nicht in geschlossener Form durch eine Gleichung beschrieben werden. Dennoch lassen sie sich im analytischen Sinne durch einfachere Hyperflächen im Raum approximieren. Für die Bildgenerierung wählt man für solch eine Fläche meist ein Dreieck. Es ist einfach zu beschreiben und flexibel genug um die meisten Objekte beliebig genau zu approximieren. Dabei wollen wir entartete Dreiecke, die nur aus einem Punkt oder einer Strecke bestehen, ausschließen, da sie für die betrachteten Render-Verfahren nicht darstellbar sind.

DEFINITION 1.1: (Dreieck)

Ein Dreieck $\triangle$ wird durch eine Sequenz $(A_\triangle, B_\triangle, C_\triangle)$ von Punkten in $\mathbb{R}^3$, für die die Menge $\{B_\triangle - A_\triangle, C_\triangle - A_\triangle\}$ linear unabhängig ist, charakterisiert. Die Menge $S(\triangle)$ der Punkte des Dreiecks ist definiert durch

$$S(\triangle) \coloneqq \operatorname{im} \varphi_\triangle$$

Dabei stellt $(M, \varphi_\triangle)$ mit den folgenden Definitionen die Standardparametrisierung der Menge $S(\triangle)$ dar.

$$M \coloneqq \left\{(u,v) \in [0,1]^2 \;\middle|\; u+v \leq 1\right\}$$

$$\varphi_\triangle \colon M \to S(\triangle), \qquad \varphi_\triangle(u,v) \coloneqq (1-u-v)A_\triangle + uB_\triangle + vC_\triangle$$

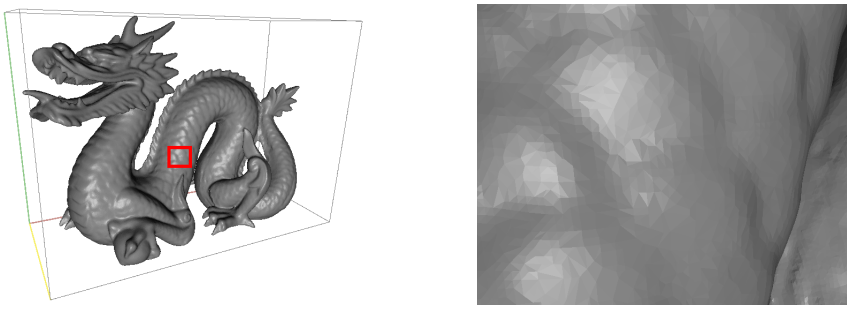
Die baryzentrischen Koordinaten (u,v,w) eines Punktes $x \in S(\triangle)$ sind durch die folgenden Eigenschaften gegeben.

$$(u,v) \in M, \qquad w = 1-u-v, \qquad \varphi_\triangle(u,v) = x$$

Die analytische äußere Normale $\mu_\triangle$ ist gegeben durch den Ausdruck

$$\mu_\triangle \coloneqq \frac{(B_\triangle - A_\triangle) \times (C_\triangle - A_\triangle)}{\|(B_\triangle - A_\triangle) \times (C_\triangle - A_\triangle)\|}$$

Jedes Dreieck besitzt auf seiner gesamten Fläche eine eindeutige konstante äußere analytische Normale. Für die Simulation von globalen Beleuchtungseffekten ist diese Eigenschaft jedoch ein Nachteil, weil die Beleuchtung eines Objektes stark vom Verlauf seiner Normalen abhängt. Nähern



(a) (b)

Abbildung 1: Die Bilder zeigen die gerenderte »Dragon«-Szene. Das rechte Bild entspricht dem roten Bereich des Linken. In dieser Szene werden die Normalen des Objektmodells durch die analytischen Normalen der Dreiecke angenähert. Da das Modell sehr fein trianguliert ist, fällt dies im ersten Bild nicht auf. Zoomt man jedoch mit der Kamera heran werden die Fehler durch die Approximation deutlich und die einzelnen Dreiecke sind mit dem menschlichen Auge auszumachen.

wir ein Objekt nun durch $n \in \mathbb{N}$ Dreiecke an, so nähern wir zur Zeit auch den Normalenverlauf des Objektes durch die stückweise konstanten äußeren Normalen der Dreiecke an. Der Fehler dieser Approximation tritt in Form eines facettenhaften Musters auf, welcher für das menschliche Auge gut erkennbar ist. In Abbildung 1 wird dieser Effekt genauer an einem Beispiel demonstriert. Das Problem kann im Allgemeinen durch eine Interpolation, die die Stetigkeit der Normalen erhält, gelöst werden.

DEFINITION 1.2: (Normalen-Funktion)

Seien $\triangle$ ein Dreieck und $\nu\colon S(\triangle) \to \mathbb{R}^3$. Dann wird ν eine äußere Normalen-Funktion auf $\triangle$ genannt, wenn für alle $x \in S(\triangle)$ die folgenden Eigenschaften gelten.

$$\|\nu(x)\| = 1, \qquad \langle \mu_\triangle, \nu(x) \rangle > 0$$

THEOREM 1.1: (Vertex-Normalen-Funktion)

Sei $\triangle$ ein Dreieck und seien Normalen $\mu_A, \mu_B, \mu_C \in \mathbb{R}^3$ an den Punkten des Dreiecks mit den folgenden Eigenschaften für alle $X \in \{A, B, C\}$ gegeben.

$$\|\mu_X\| = 1, \qquad \langle \mu_X, \mu_\triangle \rangle > 0$$

Dann definiert die Abbildung $\nu\colon S(\triangle) \to \mathbb{R}^3$ eine stetige Normalen-Funktion auf $\triangle$, wenn für alle $x \in S(\triangle)$ mit den baryzentrischen Koordinaten (u, v, w) die folgende Aussage gilt.

$$\nu(x) = \nu \circ \varphi_\triangle(u, v) \coloneqq \frac{w\mu_A + u\mu_B + v\mu_C}{\|w\mu_A + u\mu_B + v\mu_C\|}$$

Wir nennen ν in diesem Falle eine Vertex-Normalen-Funktion.

Durch das Setzen der Normalen an den Eckpunkten eines Dreiecks können wir sicher gehen, dass der Verlauf der Normalen stetig von einem Dreieck zu einem anderen übergeht. Ein beipielhafter

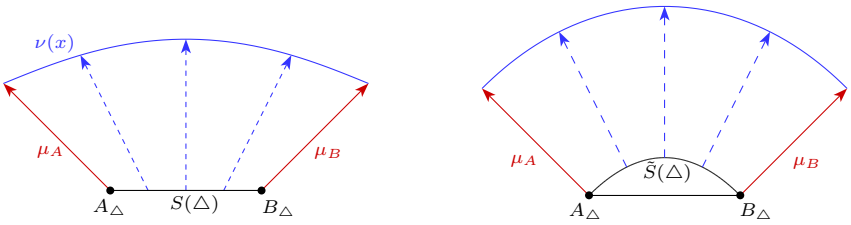


Abbildung 2: Die erste Skizze auf der linken Seite zeigt den Verlauf einer Vertex-Normalen-Funktion ν anhand eines Beispiels. $A_\triangle$ und $B_\triangle$ sind dabei die Eckpunkte eines Dreiecks $\triangle$. μ_A und μ_B sind die jeweilig gegebenen Vertex-Normalen an den Eckpunkten. Im rechten Bereich der Abbildung ist die durch ν approximierte gekrümmte Fläche $\tilde{S}(\triangle)$, auf der die Normalen $\nu(x)$ senkrecht stehen, eingezeichnet.

Verlauf einer Vertex-Normalen-Funktion wird in Abbildung 2 gezeigt. Für die in dieser Arbeit betrachteten Effekte und Verfahren reicht diese Art der Normalen-Interpolation aus. Dennoch gibt es hier weitere Möglichkeiten, wie zum Beispiel »Normal Maps«, die eine noch genauere Approximation zu Lasten des Speicherverbrauchs ermöglichen.

Die kleinste Hyperfläche, die wir durch geschlossene Gleichungen beschreiben können und aus derer wir alle weiteren Objekte zusammensetzen, nennen wir »Primitiv« (engl.: *primitive*). Es soll in unserem Falle nicht nur die Informationen des Dreiecks beinhalten, sondern auch Aussagen über dessen Material oder Oberflächenstruktur treffen können. Insbesondere muss das Primitiv oder dessen Material also die Reflexion, Absorbtion, Transmission und Emission des Lichts an der Oberfläche beschreiben. Für physikalische Materialien gibt es aber je nach den betrachteten Effekten verschiedene verallgemeinerte Varianten der mathematischen Beschreibung. Um keine dieser Varianten auszuschließen, soll hier das Primitiv in einer abstrakteren Form definiert werden.

DEFINITION 1.3: (Primitiv)

Sei $\mathcal{M}$ die Menge der Elemente, die Materialien von Objekten charakterisieren. Dann ist ein Primitiv p gerade ein Tupel $(\triangle_p, \nu_p, m_p)$, bestehend aus einem Dreieck $\triangle_p$, einer Normalen-Funktion ν_p auf $\triangle_p$ und einem Material $m_p \in \mathcal{M}$.

Für die eigentliche Approximation der Oberfläche eines Objektes benötigt man jetzt eine Menge von Primitiven. Solch eine Menge wollen wir auch eine »Szene« (engl.: *scene*) nennen. Häufig benötigt man jedoch neben den Lichtquellen, die durch die Materialien der Primitive gegeben sind, noch weitere anders geartete Lichtquellen, die unabhängig von den eigentlichen Primitven der Szene bestehen. In unserem Falle wollen wir uns speziell auf die Umgebungsbeleuchtungs-Funktion (engl.: *HDR environment map*) beziehen.

DEFINITION 1.4: (Umgebungsbeleuchtungs-Funktion)

Sei $n \in \mathbb{N}$. Dann ist eine Umgebungsbeleuchtungs-Funktion definiert als eine auf Ω integrierbare Abbildung $f\colon \Omega \to [0, \infty)^n$.

Diese Funktion beschreibt das Licht durch $n \in \mathbb{N}$ ausgewählte Koeffizienten, welches von einer Kugeloberfläche mit quasi unendlich großem Radius in die Szene ausgesandt wird. Für einen Punkt der Szene, ist dessen Beleuchtung durch diese Funktion also unabhängig von dessen Position. Diese Tatsache wird in Abbildung 3 wieder an einem Beispiel gezeigt. Die eigentliche Definition soll nun durch eine solche Umgebungsbeleuchtungs-Funktion erweitert werden.

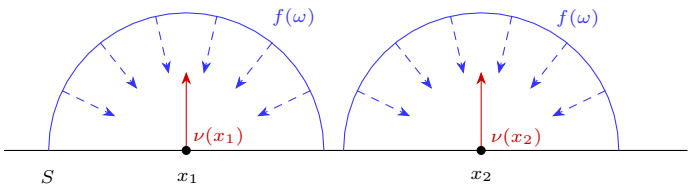


Abbildung 3: Die Darstellung zeigt, dass die Beleuchtung zweier Punkte x_1 und x_2 einer Oberfläche S mit den Normalen $\nu(x_1)$ und $\nu(x_2)$ durch eine Umgebungsbeleuchtungs-Funktion f nur abhängig von der Verdeckung der Punkte und nicht von deren Position ist. Bei x_1 und x_2 sind also in diesem Falle die gleichen Lichtintensitäten zu messen.

DEFINITION 1.5: (Szene)

Eine Szene $\mathcal{S}$ bezeichnet ein Tupel $(P_\mathcal{S}, E_\mathcal{S})$ mit einer endlichen Menge $P_\mathcal{S}$ von Primitiven und einer Umgebungsbeleuchtungs-Funktion $E_\mathcal{S}$.

1.2 Raytracing

Im einfachsten Falle bezeichnet das Wort »Raytracing« (engl.: *ray tracing*) einen Algorithmus zur Ermittlung der Sichtbarkeit von dreidimensionalen Objekten bezüglich eines Ursprungpunktes (engl.: *origin*) im Raum. Häufig versteht man darunter jedoch auch eine Render-Technik für die Generierung eines gesamten Bildes aus einer gegebenen Szene, die auf dem eben genannten Raytracing-Algorithmus basiert.

Grundsätzlich gibt es viele Verfahren, um ein Szene auf ein Bild zu rendern. Die Erfahrung zeigt aber, dass vor allem in Bereichen, in denen die globalen Beleuchtungseffekte realistisch simuliert werden sollen, Raytracing eine wichtige Grundlage darstellt. Der Grund dafür besteht in der Tatsache, dass Raytracing das Sichtbarkeitsproblem löst und durch die Verwendung von Strahlen eine Basis für die Lichtberechnung im Sinne der geometrischen Optik darstellt.

DEFINITION 1.6: (Sichtbarkeitsproblem)

Seien $\mathcal{S}$ eine Szene, $o \in \mathbb{R}^3$ und $x \in S(\mathcal{S})$ gegeben. Dann ist die Sichtbarkeitsfunktion die Abbildung $V_\mathcal{S} \colon \mathbb{R}^3 \times S(\mathcal{S}) \to \{0,1\}$ mit der Eigenschaft.

$$V_\mathcal{S}(o,x) = \begin{cases} 1 & : S(\mathcal{S}) \cap \{(1-\gamma)o + \gamma x \mid \gamma \in (0,1)\} = \emptyset \\ 0 & : \text{sonst} \end{cases}$$

Das Sichtbarkeitsproblem beschreibt dann die Aufgabe diese Funktion zu evaluieren.

Die Sichtbarkeitsfunktion gibt gerade an, ob der Oberflächenpunkt x vom Beobachtungspunkt o aus in gerader Linie gesehen werden kann oder ob zwischen diesen Punkten ein Primitiv den Punkt x verdeckt. Das ist auch der Grund, weshalb Raytracing eine effektive Lösung für diese Aufgabe darstellt. Die Basis des Verfahrens bildet das Aussenden von »Strahlen« (engl.: *ray*) bezüglich eines Ursprungspunktes. Diese werden verfolgt und auf Schnittpunkte mit der Szene getestet. Abbildung 5 zeigt das Verfahren anhand einer Skizze. Mathematisch wollen wir unter einem Strahl Folgendes verstehen.

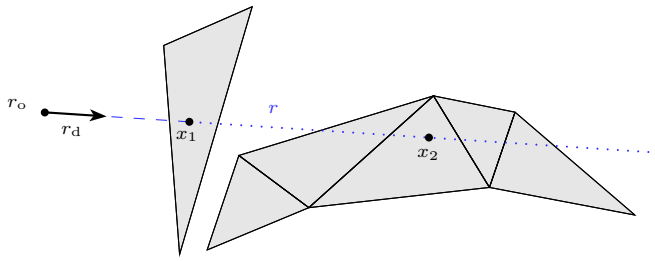


Abbildung 4: Die Abbildung zeigt eine Skizze, welche das Sichtbarkeitsproblem und den Raytracing-Algorithmus verdeutlicht. Der ausgesendete Strahl trifft in der Szene genau zwei Punkte. Dabei wird der Zweite durch den ersten verdeckt. Für den ersten Punkt ergibt die Sichtbarkeitsfunktion also 1 und für den Zweiten gerade 0.

DEFINITION 1.7: (Strahl)

Ein Strahl r sei gerade durch ein Tupel $(r_\mathrm{o}, r_\mathrm{d})$ mit einem Ursprungspunkt $r_\mathrm{o} \in \mathbb{R}^3$ und einer Richtung $r_\mathrm{d} \in \mathbb{R}^3 \setminus \{0\}$ charakterisiert. Die Menge $S(r)$ der Punkte des Strahls ist gegeben durch das Bild der Parametrisierung $([0,\infty), \varphi_r)$

$$S(r) \coloneqq \operatorname{im} \varphi_r, \qquad \varphi_r \colon [0,\infty) \to \mathbb{R}^3, \qquad \varphi_r(t) \coloneqq r_\mathrm{o} + t r_\mathrm{d}$$

DEFINITION 1.8: (Raytracing-Funktion)

Seien $\mathcal{S}$ eine Szene und r ein gegebener Strahl. Dann definiert die Abbildung $\mathrm{rt}_\mathcal{S} \colon R \to (0,\infty]$ mit der folgenden Definition die Raytracing-Funktion der Szene $\mathcal{S}$.

$$\mathrm{rt}_\mathcal{S}(r) \coloneqq \begin{cases} \min\{t \in (0,\infty) \mid r(t) \in S(\mathcal{S})\} & : S(\mathcal{S}) \cap \varphi_r((0,\infty)) \neq \emptyset \\ \infty & : \text{sonst} \end{cases}$$

Häufig erweitert man diesen Algorithmus um die Berechnung eines gesamten Bildes, indem man für jeden Pixel des Bildes einen oder mehrere Strahlen durch einen analogen virtuellen Pixel im Szenenraum schießt und die Sichtbarkeitsfunktion für diese evaluiert. Abbildung 6 zeigt dieses Verfahren anhand einer Skizze. Um gleichzeitig das Shading zu ermöglichen, ermittelt man nicht nur den Schnittpunkt, sondern auch in welchem Primitiv sich dieser befindet und welche baryzentrischen Koordinaten er besitzt. Die Implementierung des Raytracing-Verfahrens soll hier nicht gezeigt werden, da eine einfache naive Implementierung meistens einen zu großen Rechenaufwand darstellt und die hier optimierte Variante weit über das Thema dieser Arbeit hinausgeht.

Diese Definition zeigt diverse Probleme auf, die bei der Modellierung von Objekten entstehen können. Dreiecke der Primitive in einer Szene dürfen sich im Allgemeinen beliebig schneiden. Dadurch ist es im Allgemeinen nicht möglich jedem Oberflächenpunkt eine eindeutige äußere Normale zuzuordnen. Bei der Generierung der Irradiance Maps werden sich diese Punkte als Fehler äußern.

Die abstrakte Definition ist für die Implementierung leider noch nicht ausreichend.

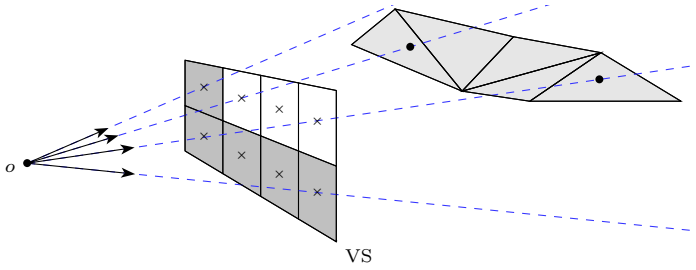


Abbildung 5

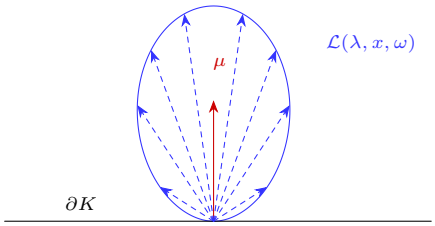


Abbildung 6

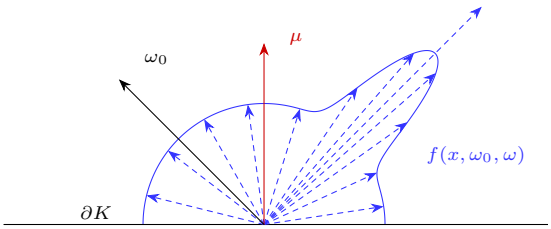


Abbildung 7

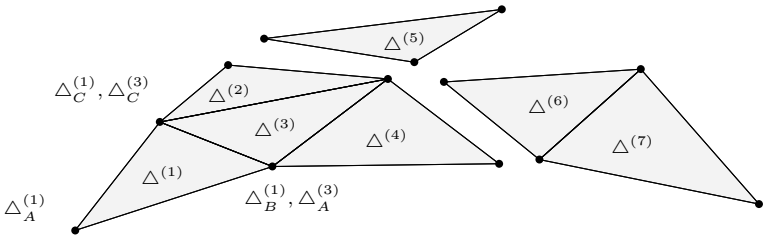


Abbildung 8: Die Abbildung zeigt eine beispielhafte Menge von Dreiecken $\left\{\triangle^{(i)} \;\middle|\; i \in \mathbb{N}, i \leq 7\right\}$. Verschiedene Gruppen von Dreiecken bilden eine Approximation einer Oberfläche im Raum. Dabei werden die Eckpunkte der Dreiecke geteilt und es gilt zum Beispiel $\triangle_B^{(1)} = \triangle_A^{(3)}$ und $\triangle_C^{(1)} = \triangle_C^{(3)}$.