
Bachelorarbeit

Surface Caching und Lightmapping

Markus Pawellek

markuspawellek@gmail.com

Inhaltsverzeichnis

1 Grundlagen und Methoden	2
1.1 Szenenrepresentation	2
1.2 Raytracing	4

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 würden.

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

$$S(\triangle) := \text{im } \varphi_\triangle$$

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

$$M := \{(u, v) \in [0, 1]^2 \mid u + v \leq 1\}$$

$$\varphi_\triangle: M \rightarrow S(\triangle), \quad \varphi_\triangle(u, v) := (1 - u - v)A_\triangle + uB_\triangle + vC_\triangle$$

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

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

Die analytische äußere Normale $\mu(\triangle)$ ist gegeben durch den Ausdruck

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

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 wir ein Objekt nun durch $n \in \mathbb{N}$ Dreiecke an, so nähern wir den Normalenverlauf des Objektes

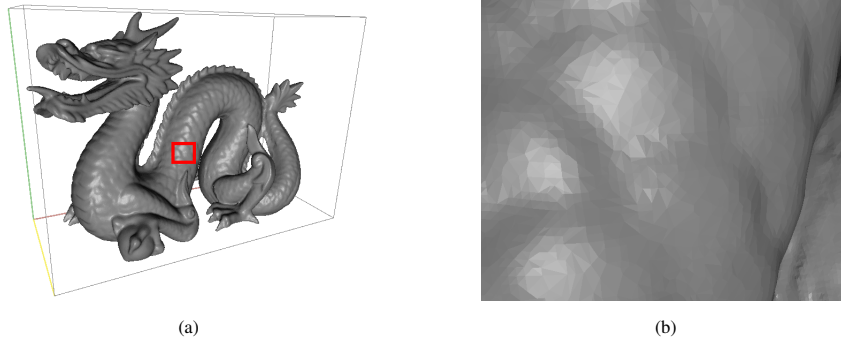


Abbildung 1: Die Abbildungen zeigen das gerenderte Bild der »Dragon«-Szene. Das zweite Bild entspricht dem roten Bereich im ersten Bild. In dieser Szene werden die Normalen des Drachenmodells durch die analytischen Normalen der Dreiecke angenähert. Da das Modell sehr fein trianguliert ist, fällt dies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

durch die stückweise konstanten Normalenverläufe der Dreiecke an. Diese Tatsache tritt in Form eines facettenhaften Musters auf. In den Bildern der Abbildung 1 wird dies genauer an einem Beispiel demonstriert. Dieses Problem kann durch eine Interpolation, die die Stetigkeit der Normalen erhält, gelöst werden.

DEFINITION 1.2: (Normalen-Funktion)

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

$$\|\nu(x)\| = 1, \quad \langle \mu(\triangle), \nu(x) \rangle > 0$$

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.

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

Dann definiert die Abbildung $\nu: S(\triangle) \rightarrow \mathbb{R}^3$ eine stetige Normalen-Funktion auf \triangle .

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

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

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

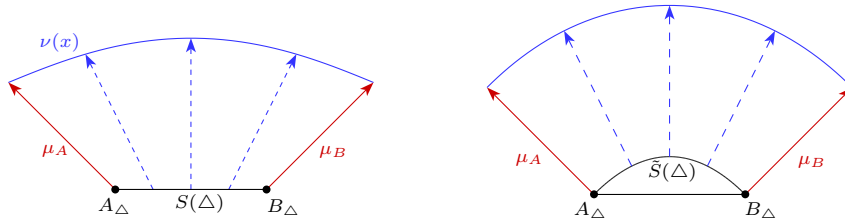


Abbildung 2: Die erste Skizze auf der linken Seite zeigt den Verlauf einer Vertex-Normalen-Funktion ν anhand eines Beispiels. A_Δ und B_Δ sind dabei die Eckpunkte eines Dreiecks Δ . μ_A und μ_B sind die jeweilig gegebenen Vertex-Normalen an den Eckpunkten. Im rechten Bereich der Abbildung ist die durch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

das Material oder die Oberflächenstruktur treffen können. Dazu gehört vor allem auch, ob und wie das Primitiv Licht aussendet. Für physikalische Materialien gibt es nun je nach den betrachteten Effekten verschiedene verallgemeinerte Varianten. Um keine dieser Varianten auszuschließen, soll hier das Primitiv in einer abstrakteren Form definiert werden.

DEFINITION 1.3: (Primitiv)

Sei X die Menge der Elemente, die Materialien von Objekten charakterisieren. Dann ist ein Primitiv P gerade ein Tupel (Δ_P, ν_P, x_P) , bestehend aus einem Dreieck Δ_P , einer Normalen-Funktion ν_P auf Δ_P und einem Material $x \in X$.

Für die eigentliche Bildgenerierung werden als Eingabe alle verwendeten Primitive benötigt. Häufig werden auch verschiedene Diese Eingabe wollen wir auch »Szene« (engl.: *scene*) nennen.

DEFINITION 1.4: (Szene)

Eine Szene \mathcal{S} bezeichnet eine endliche Menge von Primitiven. Die Menge $S(\mathcal{S})$ der Punkte der Szene ist definiert durch

$$S(\mathcal{S}) := \bigcup_{(\Delta, \nu, f) \in \mathcal{S}} S(\Delta)$$

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 1.5: (Sichtbarkeitsproblem)

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

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

Das Sichtbarkeitsproblem beschreibt dann die Aufgabe diese Funktion zu evaluieren.

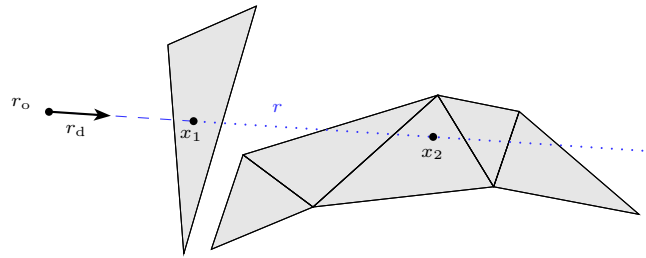


Abbildung 3: Die Abbildung zeigt eine Skizze, welche das Sichtbarkeitsproblem und den Raytracing-Algorithmus verdeutlicht. Der ausgesendete Strahl trifft in der Szene genau zwei Punkte. Dabei wird der Zweite durch den ersten verdeckt. Für den ersten Punkt ergibt die Sichtbarkeitsfunktion also 1 und für den Zweiten gerade 0.

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 3 zeigt das Verfahren anhand einer Skizze. Mathematisch wollen wir unter einem Strahl Folgendes verstehen.

DEFINITION 1.6: (Strahl)

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

$$S(r) := \text{im } \varphi_r, \quad \varphi_r: [0, \infty) \rightarrow \mathbb{R}^3, \quad \varphi_r(t) := r_o + tr_d$$

DEFINITION 1.7: (Raytracing-Funktion)

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

$\mathcal{S}.$

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

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 ?? 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.

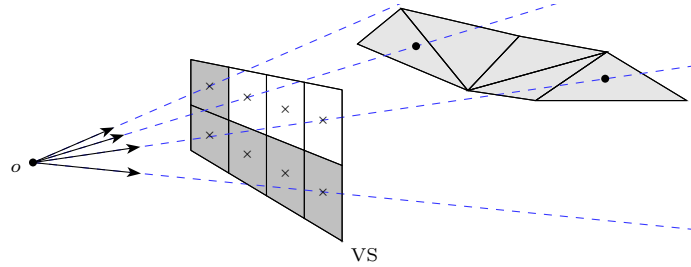


Abbildung 4

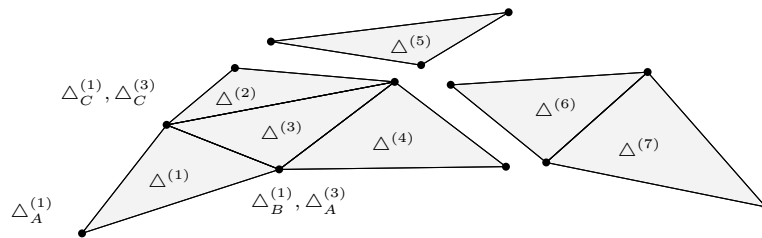


Abbildung 5: Die Abbildung zeigt eine beispielhafte Menge von Dreiecken $\{\Delta^{(i)} \mid i \in \mathbb{N}, i \leq 7\}$. Verschiedene Gruppen von Dreiecken bilden eine Approximation einer Oberfläche im Raum. Dabei werden die Eckpunkte der Dreiecke geteilt und es gilt zum Beispiel $\Delta_B^{(1)} = \Delta_A^{(3)}$ und $\Delta_C^{(1)} = \Delta_C^{(3)}$.

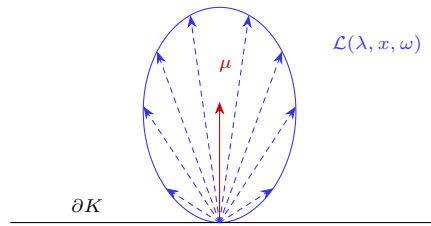


Abbildung 6

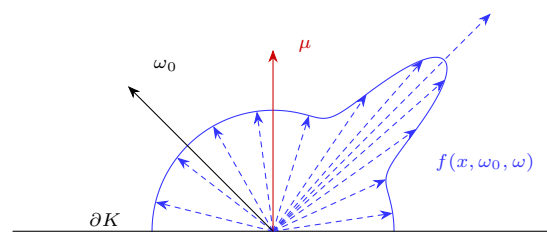


Abbildung 7