

**Technische Universität Dresden**

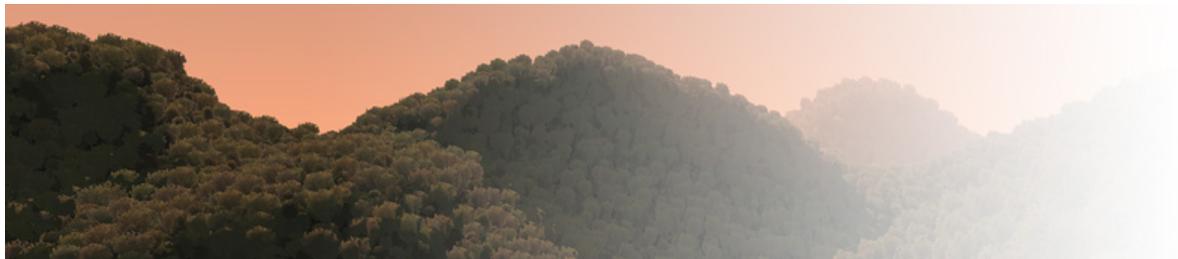
Fakultät Informatik

Institut Software- und Multimediatechnik

Professur Computergrafik und Visualisierung

**Diplomarbeit**

# **Photorealistische Darstellung komplexer Pflanzenmodelle in Level of Detail Repräsentationen**



Bearbeiter: Oliver Franzke  
Matrikelnummer: 2774039  
Betreuer: Prof. Dr. Oliver Deussen  
Hochschullehrer: Prof. Dr. Markus Wacker  
Bearbeitungszeit: 1. Oktober 2004 bis 8. April 2005

---

## **Exposé**

Die Computergrafik, als spezieller Fachbereich der Informatik, beinhaltet viele interessante Forschungsgebiete. Ein Aufgabenbereich stellt die realistische Darstellung von natürlichen Umgebungen dar. In der Regel ist eine solche Landschaft mit vielen verschiedenen Pflanzen bevölkert. Soll ein computergeneriertes Abbild einer natürlichen Szene erstellt werden, dann müssen also auch diese organischen Objekte realistisch visualisiert werden. Doch gerade die Darstellung von virtuellen Pflanzen ist mit vielen Problemen verbunden. Zum Beispiel besteht ein Wald aus tausenden von einzelnen Bäumen, Sträuchern und anderen Objekten. Neben der hohen geometrischen Komplexität einer solchen Szene, gilt es auch die komplizierten Lichtinteraktionen zu behandeln.

Diese Diplomarbeit ist deshalb der photorealistischen Visualisierung komplexer Pflanzenmodelle gewidmet. Dabei spielt sowohl die hochqualitative Darstellung, wie sie zum Beispiel in einer Filmproduktion benötigt wird, als auch die Echtzeitvisualisierung, die in Simulatoren oder Computerspielen Verwendung findet, eine wichtige Rolle. Für beide Anwendungsgebiete werden neue Verfahren eingeführt und bewertet. Mit Hilfe eines Level of Detail-Ansatzes wird es schließlich möglich, umfangreiche natürliche Szenen - in beliebigen Beleuchtungssituationen - in Echtzeit darzustellen.

## **Selbstständigkeitserklärung**

Hiermit versichere ich, die vorliegende Arbeit allein und nur unter Verwendung der angegebenen Quellen angefertigt zu haben.

Dresden, 08.04.2005

Oliver Franzke

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Vorwort . . . . .	1
1.2	Aufgabenstellung . . . . .	4
1.3	Anwendungsgebiete . . . . .	6
<b>2</b>	<b>Grundlagen</b>	<b>9</b>
2.1	Einführung . . . . .	9
2.2	Photorealistische Bildsynthese . . . . .	9
2.2.1	Rendering-Gleichung . . . . .	11
2.2.2	Lösungsmöglichkeiten der Rendering-Gleichung . . . . .	14
2.3	Grundlagen der Pflanzendarstellung . . . . .	26
2.3.1	Biologische Grundlagen . . . . .	26
2.3.2	Pflanzendarstellung durch den Computer . . . . .	31
<b>3</b>	<b>Realistische Pflanzendarstellung</b>	<b>34</b>
3.1	Einführung . . . . .	34
3.1.1	Zielstellung . . . . .	34
3.1.2	Modellierung von Pflanzen . . . . .	36
3.2	Realistisches Rendering von Pflanzenblättern . . . . .	39
3.2.1	Entwicklung eines empirischen Darstellungsmodells . . . . .	40
3.2.2	Erstellung der Texturen . . . . .	53
3.3	Bewertung der Ergebnisse . . . . .	55
3.4	Zusammenfassung und Ausblick . . . . .	59

<b>4</b>	<b>Echtzeitvisualisierung von Pflanzen</b>	<b>61</b>
4.1	Einführung . . . . .	61
4.1.1	Zielstellung . . . . .	62
4.2	Hardwareunterstützte Darstellung . . . . .	63
4.2.1	Architektur aktueller Grafikhardware . . . . .	64
4.2.2	Adaption des empirischen Algorithmus . . . . .	67
4.2.3	Effizienzsteigerung durch Basisprojektion . . . . .	73
4.3	Darstellung in einem Level of Detail Kontext . . . . .	82
4.3.1	Erstellung einer Billboardrepräsentation . . . . .	84
4.3.2	Realistische Darstellung von Billboardpflanzen . . . . .	86
4.3.3	Visualisierung komplexer Landschaften . . . . .	93
4.4	Bewertung der Ergebnisse . . . . .	101
4.5	Zusammenfassung und Ausblick . . . . .	105
<b>5</b>	<b>Zusammenfassung und Ausblick</b>	<b>108</b>
5.1	Resümee . . . . .	108
5.2	Weiterführende Ansätze . . . . .	111
5.3	Schlusswort . . . . .	112
5.4	Danksagung . . . . .	113

# 1 Einleitung

## 1.1 Vorwort

Die Natur - in all ihrer Pracht - hat die Menschen zu jeder Zeit beflügelt. Sie ist nicht nur der edle Lebensspender, der uns mit Nahrung, Luft und weiteren Ressourcen versorgt, sondern bietet ferner ein schier unendliches Spektrum an faszinierenden Impressionen. Nichts ist erholsamer, als ein gemütlicher Spaziergang an einem schönen Strand oder durch einen ruhigen Wald. Dabei stellt die Pflanzenwelt einen besonders wichtigen Teil der Natur dar, ohne sie wäre die Erde unwirtlich, karg und nicht so voller Leben.

Tief beeindruckt von dieser Pracht, versucht der Mensch ein Stück davon einzufangen und über die Zeiten zu transportieren, indem er Gedichte, Lieder oder Gemälde als Medium für seine Sinneseindrücke wählt. Auch die modernen Künste folgen dieser Tradition und so lassen sich viele Fotografien und Filme finden, deren Hintergrund und Vordergrund aus Natur besteht.

Der Grund für diese Faszination kann an dieser Stelle sicherlich nicht vollständig erklärt werden, so hängt er doch von zu vielen Faktoren ab, die von den verschiedensten wissenschaftlichen Disziplinen analysiert werden. Betrachtet man ausschließlich die optischen Eindrücke, so fällt einem augenblicklich die hohe visuelle Komplexität und der große Abwechslungsreichtum auf. Als Beispiel soll ein einzelner Baum dienen. In der Regel besteht sein Blätterwerk aus abertausenden einzelnen Blättern, die sich zwar ähneln, aber letztendlich doch alle verschieden sind. Das Selbe gilt für das darunter liegende Geäst und den Hauptstamm. Zwei artgleiche Bäume sind also niemals exakt identisch. Daraus folgt, dass jede Baumgruppe einmalig und jeder Wald ein Unikat ist.

## 1 Einleitung

---

Insbesondere diese Eigenschaften stellen für die Computergrafik ein großes Problem dar. Selbst mit Hilfe der modernsten Rechentechnik ist es nicht möglich, die Welt in ihrem vollen visuellen Umfang zu modellieren und darzustellen. Auf der anderen Seite scheint aber gerade diese hohe Komplexität den Unterschied zwischen einer realistischen und einer unrealistischen Abbildung auszumachen. In der Vergangenheit wurde deswegen häufig auf die computergenerierte Darstellung der Natur verzichtet. Eine andere Vorgehensweise besteht darin, sinnvolle Vereinfachungen zu finden, die das Problem verarbeitbar machen und dabei den visuellen Eindruck nur wenig vermindern. Diese Diplomarbeit ist in den Kontext des zweiten Lösungsansatzes einzuordnen und konzentriert sich dabei insbesondere auf die Darstellung von virtuellen Pflanzen.



Abbildung 1.1: Insel Rügen

Die Ergebnisse basieren sowohl auf den Erfolgen der vorangegangenen Grundlagenforschung ([JMLH01], [SKS02]), als auch auf den neusten Entwicklungen im Bereich der Hardware ([nvi], [ati]). Dadurch ist das Ziel, eine möglichst photorealistische Darstellung großer Teile der Natur, in greifbare Nähe gerückt. Nicht nur in Filmen, sondern gerade auch in Echtzeitsimulationen werden immer realistischere Landschaften präsentiert. Beide Problemstellungen werden in dieser Arbeit ausführlich behandelt und neue Ansätze eingeführt.

Gerade die Darstellung in Echtzeit stellt eine große Herausforderung dar. Nicht nur wegen der Möglichkeit der Interaktivität, sondern auch auf Grund von gegebenen Geschwindigkeits- und Speichereinschränkungen der aktuellen Grafikhardware. Dennoch sollte die Reaktion auf die Eingabe eines Nutzers ohne eine wahrnehmbare zeitliche Verzögerung erfolgen. Dies ist jedoch nur dann möglich, wenn die speziellen Eigenschaften des menschlichen Auges und der Grafikbeschleunigerkarten berücksichtigt und gezielt ausgenutzt werden. So kann zum Beispiel ein Baum, der eine große Distanz zum Betrachter besitzt, durch eine vereinfachte Repräsentation ersetzt werden, unter der Voraussetzung, dass dadurch kein visueller Eindruck verloren geht. Ist das präsentierte Bild trotzdem photorealistisch, so wird ein hohes Maß an Immersion erreicht. Dabei taucht der Nutzer so zu sagen in die simulierte Welt ein, wobei die Grenze zwischen Realität und Virtualität, während der Benutzung der multimedialen Anwendung, unscharf wird. Und so verkörpert gerade die photorealistische Darstellung der Welt, eines der wichtigsten Ziele der Computergrafik.

Diese Diplomarbeit versucht die Visualisierung von Pflanzen einen Schritt voranzubringen. Eines Tages wird es dann vielleicht möglich sein, die Pracht der Natur - mit Hilfe eines Computers - so realitätsnah nachzubilden, dass Menschen auch dadurch beflügelt werden.

### 1.2 Aufgabenstellung

Der Begriff des Photorealismus bedeutet, dass ein Algorithmus ein computergeneriertes Bild erzeugt, welches von einem menschlichen Betrachter nicht von einem realen Abbild - z.B. von einer Photographie - unterschieden werden kann. Ob eine künstliche Darstellung dieser Qualität entspricht hängt von vielen Faktoren ab. So gilt es neben den Berechnungsmodellen für die Kamera (oder das menschliche Auge) und die Lichtquellen vor allem die Oberflächeneigenschaften der abgebildeten Objekte zu beachten. Deswegen soll im Rahmen dieser Diplomarbeit untersucht werden, in wie weit es möglich ist, ein einfaches und photorealistisches Darstellungsmodell für Pflanzengewebe zu finden, welches auf realen biophysikalischen Effekten basiert. Dabei steht zunächst die Erstellung von sehr hochqualitativen Bildern im Zentrum der Betrachtung. Die Berücksichtigung von möglichst vielen visuellen Eigenschaften ist dabei wichtiger, als eine kurze Berechnungsdauer.

Im Kontext der interaktiven Visualisierung von Pflanzen ändern sich jedoch diese Prioritäten sehr stark, denn für eine flüssige Echtzeitdarstellung müssen viele Bilder pro Sekunde berechnet werden. Deswegen gilt es anschließend, die bereits erlangten Erkenntnisse schrittweise zu adaptieren, so dass sie auch im Kontext einer hardwareunterstützten Darstellung verwendet werden können. Hierbei sollten die vorhandenen Einschränkungen der Grafikkarten analysiert und entsprechende Lösungen präsentiert werden. Besonderes Interesse ist der Frage zu widmen, ob globale Beleuchtungseffekte in dem gegebenen Umfeld überhaupt realisiert werden können. Es sollten verschiedene hardwareunterstützte Darstellungsverfahren für Pflanzen untersucht und anschließend einem Vergleich unterzogen werden.

Aufbauend auf die allgemeinen Untersuchungen im Bereich der photorealistischen Visualisierung und der Echtzeitdarstellung gilt es, im nächsten Teil eine Methode zu entwickeln, mit der auch die vereinfachten Repräsentationen komplexer Pflanzenmodelle realistisch beleuchtet werden können. Da im Laufe einer Echtzeitsimulation zwischen unterschiedlichen geometrischen Darstellungsformen gewechselt werden kann, sollten die verwendeten Visualisierungsverfahren möglichst kohärent sein. Von besonderer Wichtigkeit ist hierbei die Frage, ob die verschiedenen geometrischen Repräsentationen durch den selben Algorithmus dargestellt werden können. Sonst wären ungewollte visuelle Umschaltartefakte die Folge.

## 1 Einleitung

---

Das Gesamtziel dieser Diplomarbeit lässt sich somit folgendermaßen zusammenfassen: Bestehende Methoden zur photorealistischen Darstellung sollen auf ihre Anwendbarkeit in Bezug auf Pflanzenmodelle, die in unterschiedlich komplexen Repräsentationsformen vorhanden sind, untersucht werden. Basierend auf dieser Analyse gilt es, gegebenenfalls neue Darstellungsverfahren zu entwickeln.



## 1.3 Anwendungsgebiete

Die photorealistische Darstellung komplexer Pflanzenmodelle besitzt viele Anwendungsgebiete. Eine grobe Klassifizierung unterscheidet zwischen Offline- und Onlinedarstellung. Nachfolgend werden einige Beispiele aus beiden Bereiche genannt und erläutert.

Das Offline-Rendering wird unter anderen in der Architekturvisualisierung verwendet. Dabei geht es dem Architekten vor allem darum, seine Vorstellung eines neuen Gebäudes möglichst realistisch in Szene zu setzen. Häufig sind photorealistische Abbildungen für die Auftraggeber sehr viel aussagekräftiger, als die eigentlichen Planungsunterlagen. Außerdem lassen sich Effekte wie Schattenwurf und Sonneneinstrahlung sehr genau analysieren. Es wäre schließlich sehr ärgerlich, wenn ein unbeacht gepflanzter Baum irgendwann einen Raum zu sehr beschatten würde. Natürlich wird diese Technik auch in verwandten Anwendungsgebieten verwendet, wie zum Beispiel in der Landschaftsplanung.

Ein weiterer wichtiger Bereich, in der die photorealistische Pflanzendarstellung Anwendung findet, stellt die Filmindustrie dar. Häufig verfügt das Produktionsteam über eine sehr genaue Vorstellung, wie ein spezieller Schauplatz aussehen soll. Ist dieser nicht direkt vorhanden, so existieren nur zwei Lösungsvarianten. Einerseits könnte ein realer Ort gefunden und anschließend solange angepasst werden, bis dieser der Vorstellung des Teams entspricht. Andererseits kann die Szenerie auch im Computer modelliert und berechnet werden. Im letzteren Fall ist eine photorealistische Abbildung der vorhandenen Pflanzen ganz besonders wichtig, weil der Zuschauer sonst relativ schnell bemerkt, dass die präsentierten Bilder aus dem Computer stammen. Eine unrealistische Darstellung der Szenerie führt jedoch dazu, dass auch die Geschichte ein wenig unplausibel erscheint, was wiederum zur Folge hat, dass die Zuschauer nicht so sehr in den Bann des Films geraten.

Im Kontext der Echtzeitdarstellung können die Computerspiele als Anwendungsgebiet genannt werden. Da das Maß an Interaktivität in dieser Art von Applikation ganz besonders hoch ist, spielt hierbei auch der oben bereits genannte Begriff der Immersion eine sehr große Rolle. Das Ziel der Entwickler eines Spiels ist es, eine möglichst glaubwürdige Welt zu präsentieren. Es ist offensichtlich, dass Pflanzen in diesem Kon-

## 1 Einleitung

---

text äußerst wichtig sind, schließlich sind wir in der realen Welt ständig von ihnen umgeben. Eine Landschaft ohne jegliche Vegetation, widerspräche der allgemeinen Vorstellung der Welt, wodurch die Glaubwürdigkeit und damit auch die Immersion sinken würde.

Ein weiteres Anwendungsgebiet für die Echtzeitvisualisierung von großen Landschaften, stellen die Simulationen dar. Konkrete Beispiele sind Flug-, Auto- und Landmaschinensimulationen ([Met05]). Sie werden sowohl zu Aus- und Fortbildungszwecken, als auch im Bereich der Forschung verwendet. In diesem Anwendungsgebiet steht allerdings häufig das physikalische Verhalten der simulierten Objekte im Vordergrund. Die photorealistische Darstellung der Umgebung ist ein sekundäres Ziel.





*Nichts ist einfacher, als sich schwierig auszudrücken,  
und nichts ist schwieriger, als sich einfach auszudrücken.*

Karl Heinrich Waggerl (1897-1973)

## 2 Grundlagen

### 2.1 Einführung

In Rahmen dieses Grundlagenkapitels werden die mathematischen, biologischen und algorithmischen Grundlagen erläutert, welche die Basis für die nachfolgenden Abschnitte darstellen. Der erste Teil beschäftigt sich ganz allgemein mit der Erstellung von photorealistischen Bildern mit Hilfe eines Computers. Dabei wird die so genannte Rendering-Gleichung eingeführt. Anschließend werden unterschiedliche Möglichkeiten beschrieben, wie diese mathematische Formulierung algorithmisch gelöst werden kann. Im Anschluß daran steht speziell die Darstellung von Pflanzen im Mittelpunkt der Betrachtung. Zunächst erfolgt eine kurze Einführung in die notwendigen biologischen Grundlagen, danach werden bisherige Ansätze untersucht, die sich dem Thema des Pflanzen-Renderings gewidmet haben.

### 2.2 Photorealistische Bildsynthese

Ein (digitales) Bild besteht aus einem zweidimensionalen Feld von Bildelementen, die Pixel genannt werden. Jedes Pixel wird durch eine definierte Anzahl von Zahlen repräsentiert, welche die Farbe des gegebenen Bildelements bestimmen. Ein einfaches Beispiel ist das additive RGB-Farbmodell, bei dem jedem Pixel drei Zahlen zugeordnet werden. Hierbei wird davon ausgegangen, dass jede Farbe durch eine Linearkombination der drei Basisfarben Rot, Grün und Blau dargestellt werden kann. In der Praxis funktioniert dieses Modell recht gut, jedoch existieren einige Farbtöne, die nicht mit dem RGB-Farbmodell erzeugt werden können. Ein Bild erzeugender Algorithmus hat nun die Aufgabe, für jeden Pixel eine Farbe zu berechnen. Die Art

und Weise wie dies geschieht, bestimmt letztendlich ob das Ergebnis photorealistisch erscheint oder nicht. Zum Beispiel liefert ein Verfahren, welches auf echten physikalischen Modellen basiert, in der Regel realistischere Ergebnisse, als eine grobe Approximation, die unabhängig von der Realität entwickelt wurde. Ein optimales Vorgehen, würde eine exakte Simulation aller beteiligten physikalischen Prozesse durchführen. Nachfolgend werden ausschließlich photorealistische Verfahren besprochen.

Zunächst gilt es, die virtuelle Welt zu beschreiben, die durch den Bild erzeugenden Algorithmus dargestellt werden soll. Dabei muss für jedes Objekt in der Szene eine geometrische Repräsentation gefunden werden. Eine häufig verwendete Darstellungsform ist das Dreiecksnetz. Dabei wird die Oberfläche eines Objekts durch eine endliche Menge von Dreiecken approximiert. Der Vorteil dieser Repräsentationsform liegt darin, dass sie leicht zu erzeugen und einfach zu verarbeiten ist.

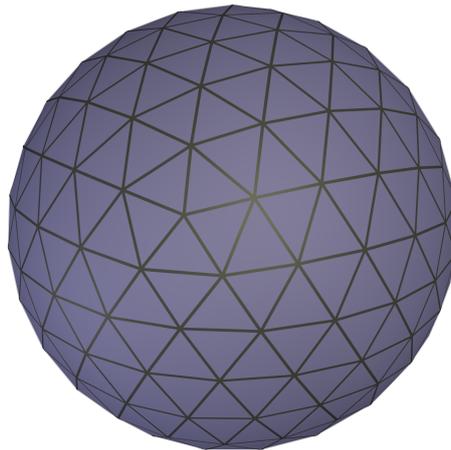


Abbildung 2.1: Triangulierte Kugel

Außerdem müssen die Materialien beschrieben werden, aus denen die Objekte bestehen. Es ist offensichtlich, dass Holz einen anderen visuellen Eindruck erzeugt, als zum Beispiel Metall. Dabei stellt sich die Frage, wie die Oberfläche eines Objekts mit den vorhandenen Photonen interagiert. Einige Interaktionsmöglichkeiten sind zum Beispiel Reflektion, Refraktion oder Absorption.

Die Hauptfrage der photorealistischen Bildsynthese besteht nun darin, für jeden Punkt im Raum die Anzahl (und Wellenlänge), der aus einer gegebenen Richtung einfal-

lenden, Photonen zu bestimmen. Wenn ein entsprechender Algorithmus diese Frage beantworten kann und außerdem für alle Objekte und Materialien ein realistisches mathematisches Modell vorhanden ist, dann wird es möglich ein Bild zu berechnen, dass der Definition von Photorealismus genügt.

### 2.2.1 Rendering-Gleichung

Die Rendering-Gleichung ist ein mathematisches Modell, welches das Interaktionsverhalten von Licht in einer definierten Szene beschreibt. Sie wurde zuerst von Kajiyama ([Kaj86]) vorgestellt. Eine vollständige Herleitung dieser Gleichung würde den Rahmen dieser Arbeit sprengen, deswegen wird es an dieser Stelle lediglich ein kurze Einführung geben, die das allgemeine Verständnis der nachfolgenden Methoden ermöglichen soll. Umfangreichere Einführungen lassen sich in [Kaj86], [CW93] und [Jen01] finden.

Gegeben sei ein beliebiger Punkt  $x$  auf der Oberfläche eines Objekts. Die Normale an dieser Stelle wird durch  $\vec{n}$  repräsentiert. Gesucht ist die in Richtung  $\vec{\omega}$  abgestrahlte Energie des Lichts, die so genannte Radianz (siehe Abb. 2.2). Die Rendering-Gleichung beschreibt nun die Einflussfaktoren, welche die ausgehenden Radianz,  $L_o$ , bestimmen:

$$L_o(x, \vec{\omega}) = L_e(x, \vec{\omega}) + L_r(x, \vec{\omega}) \quad (2.1)$$

Gleichung (2.1) sagt somit aus, dass sich die am Punkt  $x$  in die Richtung  $\vec{\omega}$  ausgehende Radianz aus der Summe von emittierter und reflektierter Radianz ( $L_e$  bzw.  $L_r$ ) zusammensetzt. Objekte die Photonen emittieren, für die also  $L_e > 0$  gilt, werden auch Lichtquellen genannt. Der zweite Summand  $L_r$  beschreibt die Radianz, welche aus der Umgebung in Richtung  $\vec{\omega}$  reflektiert wird:

$$L_r(x, \vec{\omega}) = \int_{\Omega} f_r(x, \vec{\omega}', \vec{\omega}) L_i(x, \vec{\omega}') (\vec{\omega}' \cdot \vec{n}) d\vec{\omega}' \quad (2.2)$$

Bei opaken (lichtundurchlässigen) Materialien können Photonen nur von „oben“ auf die Oberfläche treffen, da sich das Licht ansonsten durch das Objekt hindurch strahlen müsste. Entsprechend dieser Bedingung wird das Integrationsgebiet  $\Omega$  auf die obere Hemisphäre<sup>1</sup> am Punkt  $x$  beschränkt und dabei die einfallende Radianz  $L_i(x, \vec{\omega}')$

<sup>1</sup>Die obere Hemisphäre beinhaltet alle Punkte, die sich im positiv definierten Halbraum bezüglich der, durch  $x$  und  $\vec{n}$  beschriebenen, Tangentialebene befinden.

akkumuliert. Je nach Material werden aber nicht alle eingehenden Photonen in die Richtung  $\vec{\omega}$  reflektiert. Es kann zum Beispiel ein gewisser Teil absorbiert oder in eine andere Richtung gestreut werden. Das Verhältnis zwischen eingehender und ausgehender Radianz wird durch die so genannte bidirektionale Reflektionsverteilungsfunktion  $f_r(x, \vec{\omega}', \vec{\omega})$  beschrieben. Sie ist im wesentlichen für den visuellen Eindruck unterschiedlicher Materialtypen verantwortlich. Nachfolgend wird sie durch BRDF (Bidirectional Reflection Distribution Function) abgekürzt.

Die Kombination aus den Gleichungen (2.1) und (2.2) ergibt die opake Rendering-Gleichung, mit der sich alle nicht transluzenten Materialien Objekte beschreiben lassen:

$$L_o(x, \vec{\omega}) = L_e(x, \vec{\omega}) + \int_{\Omega} f_r(x, \vec{\omega}', \vec{\omega}) L_i(x, \vec{\omega}') (\vec{\omega}' \cdot \vec{n}) d\vec{\omega}' \quad (2.3)$$

Abbildung 2.2 zeigt eine sehr einfache Szene, in der die abgegebene Radianz von einer Lichtquelle (oben rechts) und einer anderen Oberfläche (oben links) abhängt. Die zweite Lichtquelle, die sich unterhalb des betrachteten Objekts befindet, besitzt keinen Einfluss auf  $L_o$ , da ihre Photonen nicht zum Punkt  $x$  vordringen können.

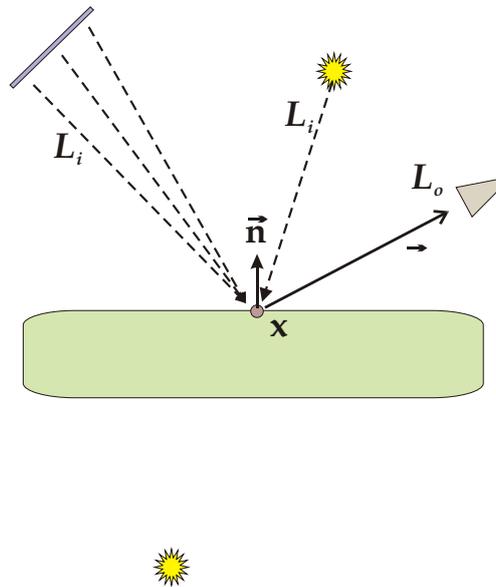


Abbildung 2.2: Rendering-Gleichung

Wie oben bereits erwähnt wurde, können mit Hilfe von Gleichung (2.3) lediglich vollkommen opake Objekte beschrieben werden. Insbesondere bei der Darstellung von



Unabhängig von der mathematischen Beschreibung der Rendering-Gleichung, lassen sich die - durch Gleichung (2.4) - beschriebenen Effekte intuitiv sehr leicht verstehen. Um die Helligkeit und Farbe des von einer Oberfläche abgestrahlten Lichts zu bestimmen, müssen zwei verschiedenartige Effekte berücksichtigt werden:

- Das Material kann selbst Photonen emittieren und damit leuchten (z.B. Lampen).
- Das Material reflektiert Licht, welches von der Umgebung auf die Oberfläche eingestrahlt wurde. Dabei wird in der Regel nur ein gewisser Anteil der eingehenden Photonen in die betrachtete Richtung abgegeben.

Von einem mathematischen Standpunkt aus betrachtet, stellt Gleichung (2.4) eine rekursive Integralgleichung dar. Um den Wert von  $L_o$  an einem Punkt  $x$  zu berechnen, muss über die gesamte eingehende Radianz  $L_i$  integriert werden. Die Bestimmung des konkreten Wertes von  $L_i$  erfordert jedoch abermals eine Evaluierung von Gleichung (2.4), denn die Stelle, welche  $L_i$  abstrahlt, reflektiert ebenfalls das dort eintreffende Licht. Die Rendering-Gleichung gilt natürlich auch an diesem anderen Punkt, wodurch noch eine Integration der eingehenden Radianz benötigt wird, die wiederum weitere (rekursive) Integrationen nach sich zieht, usw. Diese Eigenschaft von Gleichung (2.4) führt dazu, dass sie sich im Allgemeinen nicht analytisch lösen lässt. Man muss also einen numerischen Lösungsansatz finden und anwenden. Nachfolgend werden drei Verfahren erläutert, mit denen die Rendering-Gleichung, durch den Einsatz eines Computers, gelöst werden kann.

### 2.2.2 Lösungsmöglichkeiten der Rendering-Gleichung

Die Eigenschaft der Rekursivität der Rendering-Gleichung hat zur Folge, dass für die Lösung unendlich viele Rechenschritte benötigt werden. Da ein Computer aber lediglich einen endlichen Speicher besitzt und ein Nutzer nicht unendlich lange auf das Ergebnis der Berechnung warten kann, gilt es Gleichung (2.4) zu vereinfachen. Eine Analyse der Rendering-Gleichung zeigt sehr schnell, dass die Rekursionstiefe eingeschränkt werden kann, ohne gravierende visuelle Artefakte zu erhalten. Diese Vereinfachung folgt aus dem Fakt, dass von  $L_i$  nur ein kleiner Teil in die Richtung

$\vec{\omega}$  gestreut wird, da sowohl das Skalarprodukt, als auch die BSSRDF die Menge der reflektierten Photonen verringert. Dadurch wird natürlich auch jener Lichtanteil reduziert, der aus einer weiteren Rekursion folgen würde. Tatsächlich reicht häufig eine Rekursionstiefe von Eins bis Drei aus, um photorealistische Bilder zu berechnen. Jedes der nachfolgend vorgestellten Verfahren, verwendet diese Vereinfachung. In Abbildung 2.4 ist der Einfluss von drei unterschiedlichen Rekursionstiefen (Links: 0, Mitte: 1 und Rechts: 5) zu sehen.

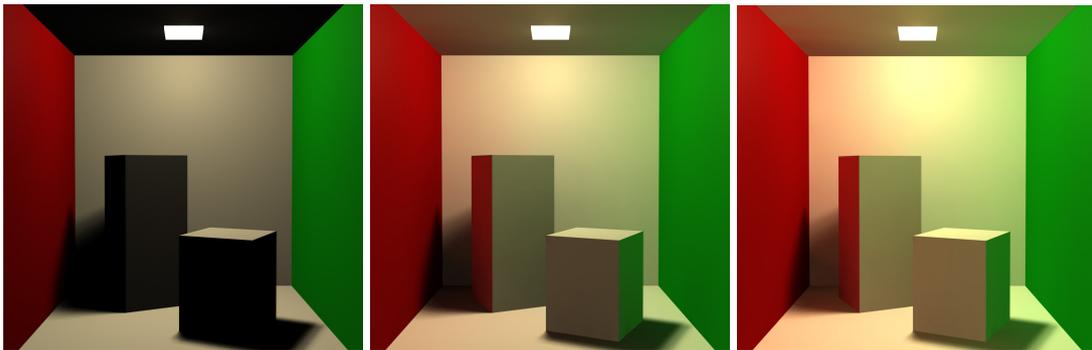


Abbildung 2.4: Einfluss der Rekursionstiefe

### 2.2.2.1 Punktweise Lösung: Raytracing

Selbst nach einer Reduzierung der Rekursivität, würde die Lösung der Rendering-Gleichung unendlich viel Zeit in Anspruch nehmen. Dies folgt aus dem Integral über die gesamte Fläche des Objekts und aus dem Integral über die Hemisphäre an der Stelle  $x'$ . Um die nachfolgenden Erläuterungen ein wenig zu vereinfachen, wird zunächst nur die vereinfachte Variante der Rendering-Gleichung (2.3) betrachtet. Einer endlichen Berechnungszeit steht nun lediglich das Hemisphären-Integral im Wege.

Beim Raytracing wird die Hemisphäre diskretisiert, indem eine Zerlegung in  $N$  endliche große Einzelteile erfolgt. Die Lösung des Integrals wird dann durch den Mittelwert von endlich vielen diskreten Abtastwerten (Samples) approximiert. Damit reduziert sich (2.3) zur folgenden Summenformulierung:

$$L_o(x, \vec{\omega}) = L_e(x, \vec{\omega}) + \frac{1}{N} \sum_{k=1}^N f_r(x, \vec{\omega}_k', \vec{\omega}) L_i(x, \vec{\omega}_k') (\vec{\omega}_k' \cdot \vec{n}) \quad (2.5)$$

Es kann gezeigt werden, dass Gleichung (2.5) zur exakten Lösung konvergiert, wenn die Abtastwerte gleichverteilt sind und  $N \rightarrow \infty$ . Es handelt sich um eine Punktweise Lösung, da die betrachtete Funktion nur an bestimmten Stellen untersucht wird.

Der Wert eines Samples wird bestimmt, indem ein Strahl, von der Stelle  $x$  in die Richtung  $\vec{\omega}_k'$ , durch die Welt verfolgt wird. Am nächsten Auftreffpunkt des Strahls gilt es dann abermals Gleichung (2.5) zu lösen. Das Ergebnis fließt als Faktor  $(L_i(x, \vec{\omega}_k'))$  in die ursprüngliche Berechnung mit ein. Dieses Verfahren wird Raytracing (Strahlenverfolgung) genannt, da bei der Bestimmung von  $L_o$  Strahlen durch die Szene verfolgt werden. Nähere Erläuterungen zum Thema Raytracing lassen sich in [FvDFH97], [WW92], [Fra03] und [Fra04] finden.

Nun stellt sich lediglich noch die Frage, wie die Zerlegung der Hemisphäre vorzunehmen ist. Die einfachste Methode zerlegt die Hemisphäre in Elemente gleicher Größe und wird auch Uniform-Sampling genannt (Abb. 2.5 (a)). Allerdings kann das gleichmäßige Abtasten der Umgebung zu visuellen Artefakten (Aliasing) führen, da eventuell wichtige Teile der Hemisphäre nicht mit einbezogen werden. Zum Beispiel wird in Abbildung 2.5 die Lichtquelle von keinem Strahl getroffen, der berechnete Pixel wird somit dunkel erscheinen.

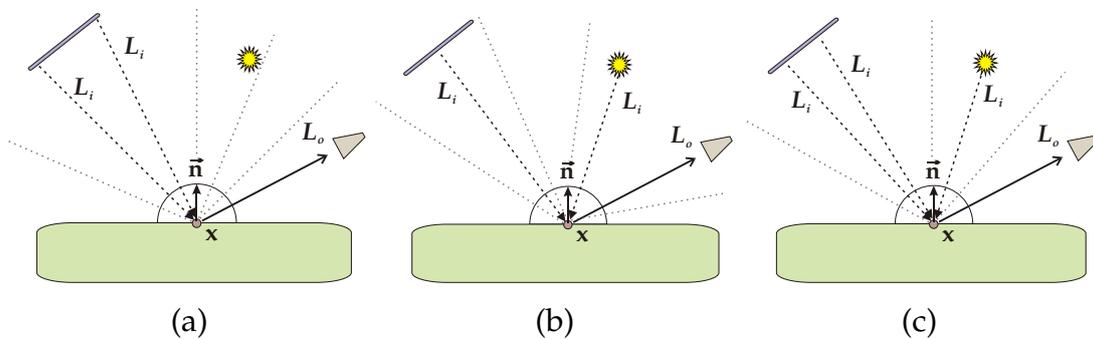


Abbildung 2.5: Diskrete Abtastung der Hemisphäre

Ein weitere Möglichkeit stellt das zufällige Abtasten dar. Dabei wird für jedes  $k$  eine zufällige Richtung  $\vec{\omega}_k'$  gewählt (Abb. 2.5 (b)). Durch dieses Vorgehen verringert sich die Wahrscheinlichkeit, dass wichtige Teile der Hemisphäre nicht mit in die Berechnung einbezogen werden. Selbst wenn in einem bestimmten Pixel eine helle Lichtquelle von keinem Strahl getroffen wird, so kann sie immer noch in einem der

benachbarten Bildelemente berücksichtigt werden. Dadurch wird zwar das Aliasing verringert, jedoch weist das berechnete Bild in der Regel starke Helligkeitsschwankungen (Noise) auf, die optisch als unangenehm empfunden werden. Das Auswerten von Integralen mit Hilfe von zufälligen Abtastwerten wird auch Monte Carlo Methode genannt. Sie spielt nicht nur in der Computergrafik eine wichtige Rolle, sondern zum Beispiel auch in der Physik ([BH02]) und der Wirtschaft ([Gla03]). Eine allgemeine Einführung kann in [Sob91] gefunden werden. Mehr Informationen über Monte Carlo Raytracing bietet [JAD<sup>+</sup>03].

Die beiden Verfahren, der uniformen und zufälligen Abtastung der Hemisphäre, können zu einer quasi-zufälligen Methode kombiniert werden (Abb. 2.5 (c)). Die Richtung  $\vec{\omega}_k'$  wird nun in der Form bestimmt, dass sie einer zufälligen Verteilung ähnelt, jedoch weiterhin deterministisch, dass heißt nicht zufällig, gewählt wird. Dies hat zur Folge, dass zwei benachbarte Pixel fast die selben Richtungen zur Berechnung von Gleichung (2.5) verwenden. Dabei werden die Strahlen so verteilt, dass sie ein optisch angenehmes Ergebnis liefern. Dieses Vorgehen verhindert somit sowohl Aliasing, als auch Noise im Ergebnisbild<sup>2</sup>. Diese Herangehensweise wird auch Quasi Monte Carlo Methode ([JAD<sup>+</sup>03], [Kel01], [Hal64], [HH64]) genannt.

Um die Resultate des Raytracing noch weiter zu verbessern, ist es außerdem möglich das vorhandene Wissen über die Umgebung in die Wahl der Abtastrichtung mit einzubeziehen. Es sollten die Bereiche der Hemisphäre häufiger gesampelt werden, die einen großen Einfluss auf das Ergebnis besitzen. Zum Beispiel könnten mehr Strahlen in die Richtungen der Lichtquellen verfolgt werden, da diese meist mehr Photonen abgeben, als die Umgebung. Diese Optimierung lässt sich sowohl mit dem zufälligen, als auch mit dem quasi-zufälligen Vorgehen kombinieren und wird Importance-Sampling genannt ([JAD<sup>+</sup>03]).

Bis zu diesem Zeitpunkt wurde lediglich die vereinfachte Form der Rendering - Gleichung (2.3) betrachtet. Um die vollständige Gleichung (2.4) zu lösen, muss zusätzlich das Integral über die Fläche des betrachteten Objekts berechnet werden. Grundsätzlich können hierbei die selben Verfahren verwendet werden, die oben bereits erläutert wurden. Der einzige Unterschied besteht darin, dass die Abtastwerte nun über die Fläche des Objekts verteilt werden. Auch dies kann wieder uniform, zufällig oder

---

<sup>2</sup>Die Bilder in Abbildung 2.4 wurden mit Hilfe einer quasi-zufälligen Strahlenverteilung berechnet.

quasi-zufällig geschehen.

Durch Raytracing lassen sich fast alle optischen Phänomene simulieren, deswegen wird es vor allem im Offline Rendering (Filmproduktion) verwendet. Das wichtigste Ziel ist in diesem Kontext ein qualitativ hochwertiges Ergebnis. Dabei ist es meist egal, wie viel Zeit für die Berechnung eines Bildes benötigt wird. Im Anwendungsgebiet der Echtzeit Computergrafik, kann jedoch nicht beliebig lange auf das finale Ergebnis gewartet werden, denn um die Illusion einer flüssigen Bewegung zu erzeugen, müssen mindestens 15 Bilder pro Sekunde erstellt werden. Jedoch genau aus diesem Grund ist Raytracing nicht für die Echtzeitdarstellung geeignet, da sich der benötigte Aufwand im Vorfeld nur schlecht abschätzen lässt und somit auch keine feste Bild-Erstellungsrate angegeben werden kann. Um eine beschleunigte Berechnung zu ermöglichen, muss das Problem vereinfacht und ein anderer Lösungsansatz gefunden werden. Im nachfolgenden Abschnitt wird deswegen ein alternatives Verfahren vorgestellt, welches die Rendering-Gleichung - mit gewissen Einschränkungen - in Echtzeit lösen kann.

### 2.2.2.2 Sphärische Funktion: Spherical Harmonic Lighting

In der Echtzeit Computergrafik existieren - neben den zuvor genannten zeitlichen Bedingungen - weitere Einschränkungen, die zum Teil aus der benutzten Hardware, als auch aus den verwendeten Grundprinzipien<sup>3</sup> folgen. Die aktuellen Grafikkarten besitzen nur einen sehr begrenzten Speicher, deswegen werden nacheinander alle Objekte - eventuell in mehreren Einzelteilen - in den Arbeitsspeicher der Grafikkarte geladen und gerastert. Im Gegensatz zum Raytracing ist - während der Berechnung der Farbe eines Pixels - die Beschreibung der virtuellen Welt nicht vollständig verfügbar, weil sich in der Regel nicht die gesamte Szene im Speicher der Grafikkarte befindet, sondern lediglich ein einzelnes Objekt. Somit ist also der Zugriff auf die Daten eines anderen Szenenobjekts nicht möglich. Daraus folgt jedoch, dass auch keine Strahlen durch die Welt verfolgt werden können, um die eingehende Radianz zu bestimmen, weil dafür auf die geometrische Beschreibung aller Objekte zugegriffen werden muss. Allerdings ist es möglich eine stark begrenzte Anzahl von Zahlenwerten - zusätzlich zu den Informationen über das aktuelle Objekt - in den

---

<sup>3</sup>Eine detailliertere Beschreibung der Arbeitsweise der Grafikkarte erfolgt in Kapitel 4.

Arbeitsspeicher der Grafikkarte zu kopieren, die dann als Eingabedaten für die Bestimmung der Pixelfarbe dienen. Diese Zahlenwerte werden entweder für ein ganzes Dreieck definiert oder mit den Eckpunkten eines Dreiecks - den so genannten Vertices - assoziiert. Es gilt also nunmehr eine geschickte Parametrisierung der nötigen Daten zu finden, um trotz der gegebenen Einschränkungen photorealistische Bilder zu berechnen. Dies erfolgt häufig als Vorverarbeitungsschritt, mit dem Ziel spätere Berechnungen wesentlich zu vereinfachen oder überhaupt erst möglich zu machen. Da während der Bestimmung der Farbwerte aller Pixel keine globalen Informationen über die virtuelle Szene vorhanden sind, wird diese Herangehensweise als lokales Modell bezeichnet.

Eine mögliche Parametrisierung des Integrals über die eingehenden Radianz, stellt die Projektion der Daten in einen anderen Funktionenraum dar. Am besten eignen sich dafür Räume mit orthonormalen Basisfunktionen, da das Ergebnis der Projektion dann lediglich aus einer Menge von Basiskoeffizienten besteht. Bei der Wahl des neuen Funktionenraums sollte eine „kluge“ Entscheidung getroffen werden und ein dem Problem angepasster Zielraum verwendet werden.

Beispielhaft soll zunächst eine einfache Funktion  $f$  (Abb. 2.6 oben), die im zweidimensionalen kartesischen Koordinatensystem definiert ist, in einem anderen Funktionenraum dargestellt werden. Dieser neue Raum wird durch die linearen Basisfunktionen  $B_i$  (Abb. 2.6 Mitte) beschrieben. Um die konkreten Basiskoeffizienten der ursprünglichen Funktion zu bestimmen, welche die Beschreibung von  $f$  im Zielraum ermöglichen, muss jedes Integral aus dem Produkt von  $f$  und  $B_i$  ausgewertet werden. Mit Hilfe der berechneten Koeffizienten kann die ursprüngliche Funktion approximiert werden, indem die Produkte der  $B_i$  mit den jeweiligen Basiskoeffizienten aufsummiert werden. Das Ergebnis ist die approximierte Funktion  $\hat{f}$  (Abb. 2.6 unten). Das in Abbildung 2.6 dargestellte Beispiel zeigt bereits sehr gut, dass sich durch die Projektion in einen anderen Funktionenraum unter Umständen Speicher sparen lässt. Für die Funktion  $f$  müssen im neuen Zielraum lediglich vier reelle Zahlen gespeichert werden. Die erreichte Approximation  $\hat{f}$  ist zwar sehr grob, könnte aber durch die Verwendung einer größeren Anzahl von Basisfunktionen verbessert werden. Der Fehler lässt sich somit sehr gut kontrollieren.

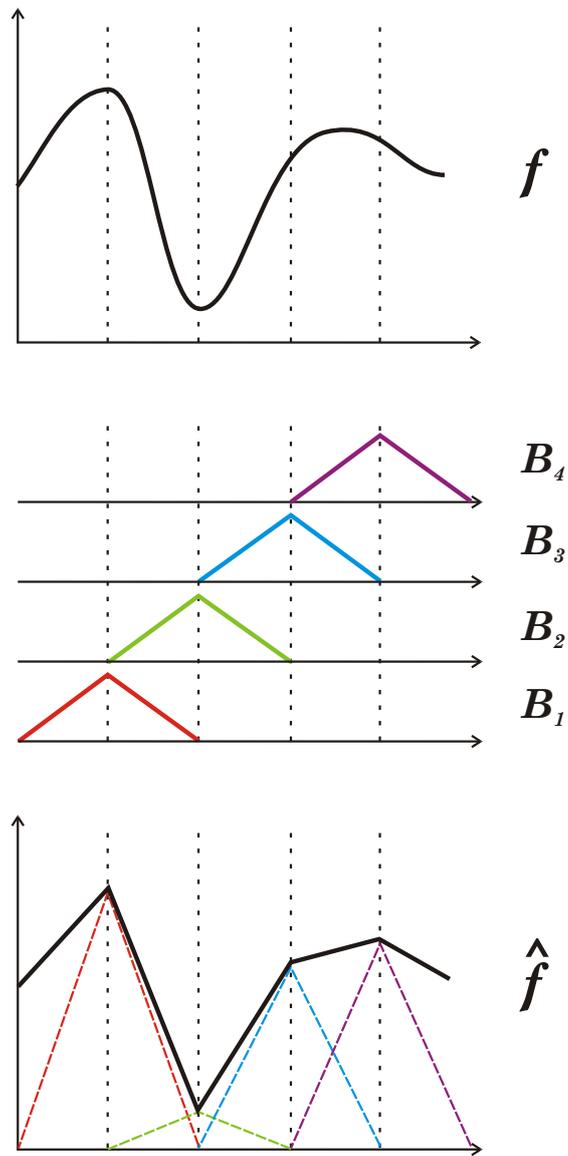


Abbildung 2.6: Approximation einer Funktion

## 2 Grundlagen

---

Im konkreten Anwendungsfall der Rendering-Gleichung liegt eine spezielle Ausgangssituation vor. Das größte Problem stellt hierbei die eingehende Radianz dar, da die Integration über die Hemisphäre Informationen über die Umgebung erfordert und eventuell weitere rekursive Berechnungen nach sich zieht. Da die Hemisphäre ein Teil der Einheitskugel repräsentiert, ist ein optimaler Funktionenraum in einem Kugelkoordinatensystem definiert. In einem solchen Koordinatensystem lässt sich jeder Punkt auf der Einheitskugel in der folgenden Form beschreiben

$$(x, y, z) \rightarrow (\sin \theta \cos \varphi, \sin \theta \sin \varphi, \cos \theta),$$

wobei mit  $\varphi$  der Azimut- und  $\theta$  der Polarwinkel gemeint ist. Als (orthonormale) Basisfunktionen für ein Kugelkoordinatensystem schlugen Sloan *et al.* ([SKS02]) die so genannten (assozierten) Legendre Polynome  $P$  vor. Die vollständige Herleitung der so genannten Spherical Harmonics Transformation, ist sehr kompliziert und würde den Rahmen dieser Arbeit bei weitem sprengen, deswegen werden an dieser Stelle lediglich die entsprechenden Formeln genannt. Eine umfangreiche und leicht verständliche Einführung in das Thema kann in [Gre03] gefunden werden.

Die Bestimmung der Basiskoeffizienten eines beliebigen Elements des Kugelkoordinatensystems erfolgt mit Hilfe der Spherical Harmonics Funktion:

$$y_l^m(\theta, \varphi) = \begin{cases} \sqrt{2}K_l^m \cos(m\varphi)P_l^m(\cos \theta), & m > 0 \\ \sqrt{2}K_l^m \sin(-m\varphi)P_l^{-m}(\cos \theta), & m < 0 \\ K_l^0 P_l^0(\cos \theta), & m = 0 \end{cases} \quad (2.6)$$

Für die Normalisierung der Ergebnisse sorgt der Skalierungsfaktor  $K$ :

$$K_l^m = \sqrt{\frac{(2l+1)(l-|m|)!}{4\pi(l+|m|)!}}$$

Die Parameter  $l$  und  $m$  definieren ein spezielles Band der Spherical Harmonics Funktion, wobei  $l \in \mathcal{N}$  und  $-l \leq m \leq l$  gilt. Die Anzahl der verwendeten Bänder bestimmt, aus wie vielen Elementen ein Basiskoeffizientenvektor besteht. Je mehr Bänder mit in die Projektion einbezogen werden, desto genauer wird die Approximation. Jedoch steigt auch die benötigte Rechenzeit und der verwendete Speicherplatz an. Im konkreten Anwendungsfall wird der Bandindex auf einen Wert  $n \in \mathcal{N}^+$  festgelegt,

## 2 Grundlagen

---

darauhin muss die Spherical Harmonics Funktion für jedes notwendige  $l$  evaluiert werden, wobei  $0 \leq l \leq n - 1$  gilt.

Die assoziierten Legendre Polynome bilden den Kern der Spherical Harmonics Funktion. Sie sind von den Bandindices abhängig und definieren eine Menge von orthogonalen Basisfunktionen. Der Wert der Legendre Polynome lässt sich durch eine rekursive Berechnung ermitteln, welche über die folgenden Gleichungen definiert ist:

$$P_0^0(x) = 1 \quad (2.7)$$

$$P_m^m(x) = (-1)^m (2m - 1)!! (1 - x^2)^{\frac{m}{2}} \quad (2.8)$$

$$P_{m+1}^m(x) = x(2m + 1)P_m^m(x) \quad (2.9)$$

$$(l - m)P_l^m(x) = x(2l - 1)P_{l-1}^m(x) - (l + m - 1)P_{l-2}^m(x) \quad (2.10)$$

Abbildung 2.7 zeigt die Graphen der ersten neun assoziierten Legendre Polynome (Bandindex  $n = 3$ ). Die höheren Bänder zeigen mehr Extremstellen, wodurch Funktionen mit größeren Frequenzen besser approximiert werden können.

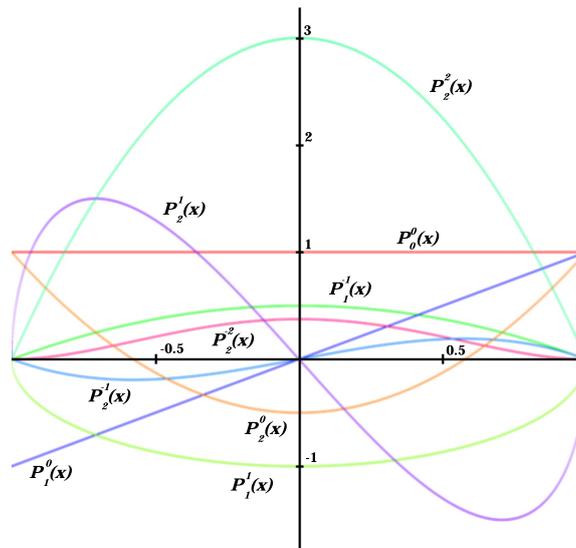


Abbildung 2.7: Graph der Legendre Polynome

Unter der Voraussetzung, dass die BRDF lediglich einen diffusen Anteil besitzt<sup>4</sup>, erfolgt die Projektion des Integrals über die Hemisphäre mit Hilfe der Spherical Har-

<sup>4</sup>Es ist auch möglich nicht diffuse Terme mit einzubeziehen, dann besteht das Ergebnis der Projektion jedoch eher aus einer Koeffizientenmatrix, als aus einem Vektor.

monics Funktion folgendermaßen:

$$c_l^m = \int_{\Omega} y_l^m(\vec{\omega}') f_r(x, \vec{\omega}') L_i(x, \vec{\omega}') (\vec{\omega}' \cdot \vec{n}) d\vec{\omega}' \quad (2.11)$$

Mit  $c_l^m$  ist ein Element des Koeffizientenvektors gemeint. Das Integral kann dabei durch den Einsatz von Raytracing (siehe Abschnitt 2.2.2.1) ausgewertet werden. Der inverse Prozess zur Projektion wird auch Rekonstruktion genannt und ist in der folgenden Form definiert:

$$L_r(x, \vec{\omega}) = \sum_{l=0}^{n-1} \sum_{m=-l}^l c_l^m \times y_l^m(\vec{\omega}) \quad (2.12)$$

Dadurch vereinfacht sich die rein diffuse Variante von (2.3) zur folgenden Gleichung:

$$L_o(x, \vec{\omega}) = L_e(x, \vec{\omega}) + \sum_{l=0}^{n-1} \sum_{m=-l}^l c_l^m \times y_l^m(\vec{\omega}) \quad (2.13)$$

Bei genauerer Betrachtung von Gleichung (2.13) fällt sofort auf, dass diese Version der Rendering - Gleichung keine (expliziten) Informationen über die Umgebung benötigt. Diese sind implizit in den Basiskoeffizienten  $c$  enthalten. Sind also die Koeffizientenvektoren erst einmal berechnet, lässt sich die ausgehende Radianz sehr schnell bestimmen. Der große Vorteil der Spherical Harmonics Repräsentation liegt vor allem darin, dass die Projektion in die neue Basis nur einmal durchgeführt werden muss. Danach kann die Lösung immer wieder verwendet werden.

Zum besseren Verständnis der Spherical Harmonics Transformation, soll diese nun noch einmal anhand eines einfachen Beispiels visualisiert werden. Abbildung 2.8 zeigt eine einfache sphärische Funktion, welche die eingestrahelte Radianz zweier Lichtquellen darstellt. In der oberen Zeile ist ein Abbild der Funktion auf der Einheitskugel zu sehen, wobei hellere Bereiche eine höhere Intensität als dunkle Stellen besitzen. Die untere Zeile zeigt stattdessen die Intensitätsverteilung anhand des Betrags. Dabei stellt die Größe einer „Ausbuchtung“ die Leuchtstärke der Funktion an der entsprechenden Stelle dar. In der linken Spalte (Abb. 2.8 (a)) ist die Originalfunktion zu sehen. Die zweite Spalte (Abb. 2.8 (b)) zeigt die Rekonstruktion einer Spherical Harmonics Repräsentation, die lediglich zwei Bänder benutzt. Die Approximation ist dementsprechend ungenau, benötigt aber lediglich vier reelle Zahlen pro Farbband

(also insgesamt zwölf Werte). In der mittleren rechten Spalte (Abb. 2.8 (c)) ist eine Rekonstruktion dargestellt, die vier Bänder verwendet. Die ursprüngliche Funktion wird sichtbar besser approximiert, erscheint jedoch abgerundet. Der Speicherbedarf besteht aus sechzehn Zahlen pro Farbband. Die Repräsentation in der rechten Spalte (Abb. 2.8 (d)) benutzt zwölf Bänder. Die Qualität der Originalfunktion wird fast erreicht, es müssen jedoch 144 reelle Zahlen pro Farbband gespeichert werden.

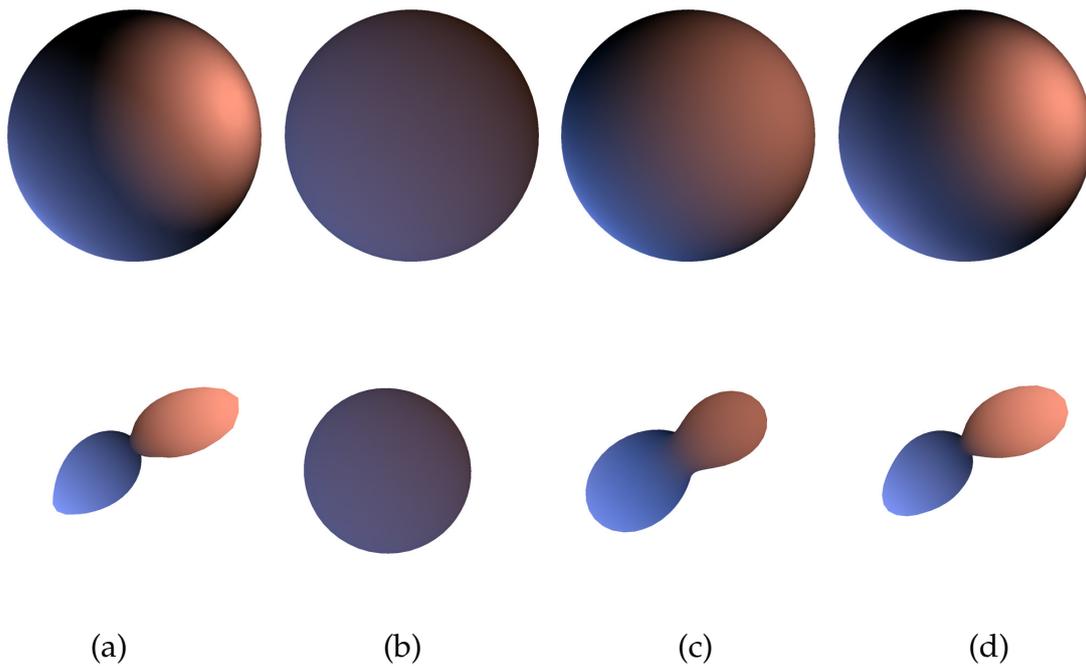


Abbildung 2.8: Approximation einer sphärischen Funktion

Auf Grund des besseren Verständnisses wurde zunächst lediglich die vereinfachte Variante der Rendering-Gleichung (2.3) betrachtet. Die Lösung der vollständigen Version (2.4) erfolgt im Prinzip in der selben Art und Weise. Bei der Projektion muss jedoch die komplette Sphäre mit einbezogen werden, da die eingehende Radianz nicht nur aus der oberen Hemisphäre eingestrahlt wird.

### 2.2.2.3 Finite Elemente: Radiosity

Eine weitere Möglichkeit, die Rendering-Gleichung (2.3) mit Hilfe eines Computers zu lösen, bietet die Finite Elemente Radiosity Methode. Um die Lösung zu verein-

fachen, wird von vornherein auf die Betrachtung der komplizierten, nicht-diffusen Lichtinteraktionen verzichtet. Daraus folgt eine wichtige Vereinfachung: Die BRDF ist sowohl von der ausgehenden, als auch von der eingehenden Strahlungsrichtung des Lichts unabhängig und kann somit vor das Integral über die Hemisphäre gezogen werden. Dadurch wird die Rendering-Gleichung jedoch unabhängig von der Abstrahlungsrichtung  $\vec{\omega}$ . Gleichung (2.3) vereinfacht sich zur so genannten Radiosity-Gleichung:

$$L_o(x) = L_e(x) + f_r(x) \int_{\Omega} L_i(x, \vec{\omega}') (\vec{\omega}' \cdot \vec{n}) d\vec{\omega}' \quad (2.14)$$

Um eine numerische Lösung zu ermöglichen, wird die Szene in eine Menge von - zu- meist rechteckigen - Einzelflächen (Patches) zerlegt und es wird vorausgesetzt, dass die BRDF über ein Patch konstant ist. Die geometrische Beziehung zwischen zwei Einzelflächen wird durch einen so genannten Formfaktor beschrieben, der letztendlich angibt, wie viel Licht von einem Patch zu einer anderen Einzelfläche gestrahlt wird. Befindet sich also zum Beispiel ein anderes Objekt zwischen zwei Flächen, so können diese auch keine Photonen austauschen. Die Formfaktoren werden während der Lösung der Radiosity-Gleichung benötigt, um die (diffuse) Lichtreflektion zwischen den Patches zu beschreiben. Durch eine Transformation in einen anderen Funktionenraum, wird das Problem in der Regel in ein lineares Gleichungssystem transformiert, welches dann durch Standardverfahren, wie der Jacobi-Iteration, gelöst werden kann.

Der Nachteil der Radiosity liegt vor allem in der hohen Berechnungszeit und der mangelnden Flexibilität. Nachdem die Lösung berechnet wurde, sind kaum noch Änderungen an der Beleuchtungssituation oder der Szene möglich. Wird also zum Beispiel eine Lichtquelle oder ein Objekt verschoben, dann muss die Lösung erneut bestimmt werden. Außerdem benötigt Radiosity sehr viel Speicherplatz, da alle Formfaktoren gespeichert werden müssen. Ein großer Vorteil folgt allerdings aus der erreichten Blickpunktunabhängigkeit, denn die berechnete Lösung der Radiosity-Gleichung lässt sich aus einer beliebigen Perspektive betrachten. Auf Grund der genannten Nachteile, stellt das Radiosity - im Bereich des photorealistischen Renderings - keine Alternative zum Raytracing oder der Spherical Harmonics Repräsentation dar. Allerdings wird es sehr erfolgreich in Pflanzenwachstumssimulationen eingesetzt ([SSBdR01]). Eine umfangreiche Einführung in die Radiosity Methode kann in [CW93] gefunden werden.

### 2.3 Grundlagen der Pflanzendarstellung

Pflanzen zeichnen sich - gegenüber anderen Objekten - insbesondere durch ihre interessanten optischen Eigenschaften aus. Viele Pflanzen bestehen aus tausenden Einzelteilen, wie Blätter oder Äste. Diese geometrische Komplexität hat zumeist auch eine vielfältige und komplizierte Lichtinteraktion zur Folge. Um diese Effekte korrekt zu simulieren, gilt es zunächst deren Ursachen zu untersuchen. Dann können die natürlichen Prozesse so weit vereinfacht werden, dass sie sich mit Hilfe eines Computers nachahmen lassen. Problematisch ist dabei allerdings der Fakt, dass es schier unendlich viele verschiedene Arten von Pflanzen gibt. Sie bilden mehr als 90 Prozent der gesamten Biomasse der Erde, wobei ihr Größenspektrum von winzig (einzellige Pflanzen) bis hin zu riesig (Mammutbäume) reicht. Jede Art sieht dabei ein wenig anders aus, was eine umfassende Untersuchung der optischen Eigenschaften von Pflanzen an dieser Stelle unmöglich macht. Deswegen werden nachfolgend hauptsächlich Laubbäume betrachtet, die in den gemäßigten Klimazonen anzutreffen sind. Dies ist jedoch keine zu starke Einschränkung, da sich die präsentierten Erkenntnisse ohne große Probleme auf andere Arten übertragen lassen.

#### 2.3.1 Biologische Grundlagen

In der Regel besitzt eine Pflanze eine Wurzel, deren Hauptaufgabe darin besteht die Pflanze im Boden zu verankern und mit Nährstoffen zu versorgen. Die verzweigte - zumeist unterirdische - Struktur der Wurzel mündet in einer oder mehreren Sprossachsen, welche die Basis für die folgenden Elemente, wie Blätter, Blüten oder weiter verzweigte Sprossachsen, bildet. Die jeweiligen Bestandteile der Pflanzen erfüllen äußerst spezialisierte Aufgaben, so absorbiert die Wurzel Wasser und Mineralien. Diese Stoffe werden anschließend von den Sprossachsen zu anderen Pflanzenteilen weitergeleitet. Währenddessen wird von den Blättern die Photosynthese durchgeführt, welche die - zum Überleben benötigte - Energie erzeugt. Dabei wird Licht, Wasser und Kohlendioxid in chemische Energie umgewandelt und unter anderem Sauerstoff freigesetzt. Durch diesen Nebeneffekt der Photosynthese werden Pflanzen zur wichtigsten Lebensgrundlage für andere Lebensformen, wie Tiere und natürlich Menschen.

Bäume unterscheiden sich von anderen Pflanzen häufig dadurch, dass sie lediglich einen Hauptstamm besitzen, der in der Regel in weitere Nebenstämme und Äste mündet. Letztendlich bildet das Geäst die Basis für die Blätter, welche die Photosynthese durchführen. Ein weiterer Unterschied zu anderen Pflanzenarten besteht außerdem darin, dass Bäume mehrjährige Gewächse sind. Einige Exemplare werden weit über 100 Jahre alt.

Die - für die Pflanzendarstellung - interessantesten Elemente eines Baums sind die Blätter, da sie im Gegensatz zum Stamm und Geäst nicht aus verholztem Gewebe bestehen. Es wird zwischen immergrünen und sommergrünen Bäumen unterschieden. Hiesige Laubbäume sind der zweiten Kategorie zuzuordnen, da sie im Herbst ihre Blätter abwerfen. Im Frühling wird neues Laub ausgebildet. Nachfolgend werden exemplarisch Blätter eben dieser sommergrünen Laubbäume betrachtet.

### 2.3.1.1 Anatomie ausgewählter Blätter

Bevor eine Analyse der optischen Eigenschaften erfolgen kann, sollte zunächst der Aufbau eines Blattes betrachtet werden. In der Regel bestehen die Blätter von sommergrünen Laubbäumen aus einem Stiel (Petiolus), mit dem sie an einem Zweig befestigt sind und aus einem flachen Teil (Blattspreite). Der Blattstiel leitet die Nährstoffe in die Spreite, wo sie als Ausgangsbasis für die Photosynthese dienen. In Abbildung 2.9 sind zwei verschiedene Blattarten zu sehen. Deutlich erkennbar ist sowohl der Stiel, als auch die Blattspreite.



Abbildung 2.9: Ausgewählte Pflanzenblätter

## 2 Grundlagen

Die grüne Färbung rührt von dem vorhandenen Chlorophyll her, welches sich in den Chloroplasten befindet. Dieser grüne Farbstoff ist dem Hämoglobin ähnlich, welches das Blut rot färbt. Die Hauptaufgabe des Chlorophylls besteht darin Licht zu absorbieren und der Photosynthese in Form von gespeicherter Energie zur Verfügung zu stellen.

Wird der Querschnitt eines Blattes betrachtet so sind verschiedene Schichten zu sehen. Die Ober- und Unterseite wird jeweils durch eine wachsartige Schicht (Cuticula) geschützt. Danach folgt die Epidermis des Blattes, die mit Poren durchsetzt ist, um Kohlendioxid aufzunehmen und Sauerstoff abzugeben. Unterhalb der oberen Epidermis befinden sich in der Regel die so genannten Palisadenzellen, die ein Teil des Mesophylls bilden. Sie sind mit Chloroplasten durchsetzt, die das einfallende Licht absorbieren. Darunter ist ein grobgranulares Gewebe zu erkennen, das viele Leerräume enthält. Abbildung 2.10 zeigt ein solches Blatt im Querschnitt und stellt dabei die genannten Elemente dar.

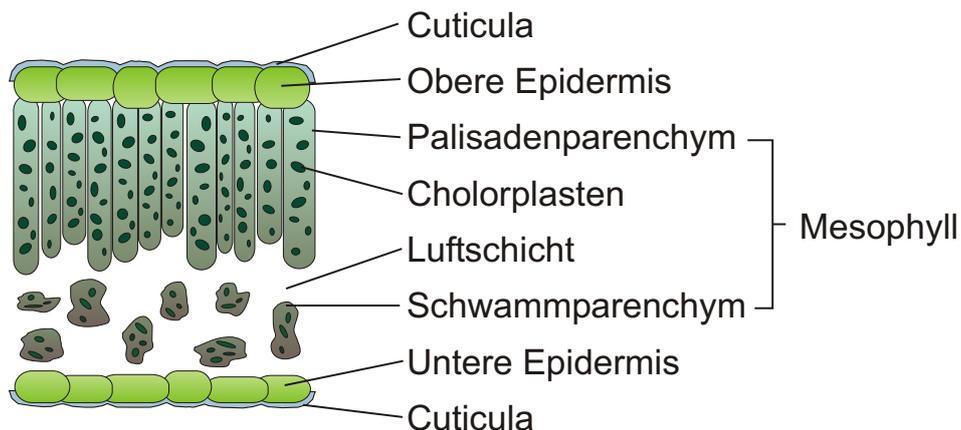


Abbildung 2.10: Querschnitt eines Blattes

Natürlich befinden sich noch viele weitere Zellen in einem „normalen“ Blatt. Sie wurden an dieser Stelle nicht erwähnt, da sie für die nachfolgende Betrachtung der optischen Eigenschaften nur eine untergeordnete Rolle spielen. Für die Darstellung von Blättern ist jedoch wichtig, dass alle Zellen eines Blattes lichtdurchlässig sind. Diese Eigenschaft ist deswegen von entscheidender Bedeutung, da somit das - für die Pflanze lebenswichtige - Licht auch zu den Blättern eines Baums vordringt, die sich

im inneren des Blätterdachs befinden und somit über keine oder nur eingeschränkte direkte Lichteinstrahlung verfügen.

### 2.3.1.2 Optische Eigenschaften von Pflanzen

Die Anordnung der Zellen innerhalb eines Blattes hat sich - durch die Evolution - in der Form gebildet, dass eine optimale Energieversorgung der Pflanze garantiert werden kann. Dabei werden die eintreffenden Photonen durch zahlreiche Lichtinteraktionen zum Chlorophyll und danach zu anderen Blättern weitergeleitet. Gerade an den Grenzen zwischen einzelnen Zellen kommt es zu vielfältigen Reflektionen und Refraktionen. Diese Effekte können mit Hilfe der Gesetze von Snellius (1580 - 1626) und Fresnel (1788 - 1827) beschrieben werden, die sich mit der Natur des Lichts auseinander gesetzt haben.

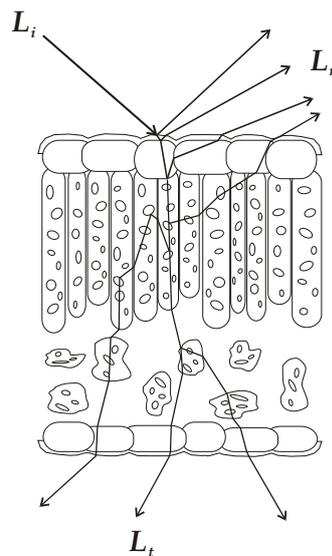


Abbildung 2.11: Lichtinteraktion innerhalb eines Blattes

Wird die Interaktion der Photonen mit einem Blatt auf einem abstrakten Niveau betrachtet, so lässt sich die ausgehende Radianz in einen reflektierten und transmittierten Anteil zerlegen. Mit dem reflektierten Teil werden hierbei die Photonen bezeichnet, die von der selben Seite des Blattes wieder abgestrahlt werden, wie die eingehende Radianz (Abb. 2.11: reflektierte Radianz ( $L_r$ )). Dementsprechend verlässt der

transmittierte Anteil des Lichts das Blatt auf der gegenüberliegenden Seite (Abb. 2.11: transmittierten Radianz ( $L_t$ )). Während sich ein Photon innerhalb des Blattes befindet, kann sich seine Wellenlänge und Energie durch Effekte wie Absorption, Difraktion und Beugung ändern. Zum Beispiel erscheint ein Blatt in einem grünen Farbton, weil durch das Chlorophyll mehr rote und violette als grüne Photonen absorbiert werden.

Wird das Streuverhalten von Blättern analysiert, so fällt sofort auf, dass der transmittierte Anteil des eingestrahlt Lichts eine diffuse Natur besitzt. Dieses Verhalten folgt aus den komplizierten und verzweigten Lichtpfaden innerhalb eines Blattes, es wird keine spezielle Richtung bevorzugt. Im Gegensatz dazu verlassen die reflektierten Photonen, das Blatt hauptsächlich in die Richtung des ideal reflektierten Strahls. Die Bevorzugung einer speziellen Richtung ist also zu erkennen.

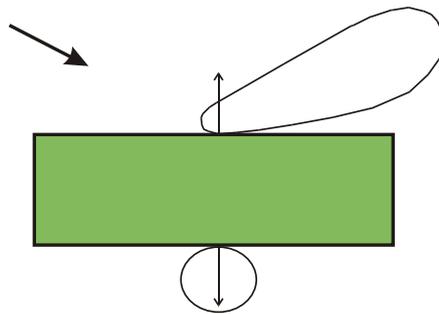


Abbildung 2.12: Schematische BSSRDF eines Blattes

Dieses Verhalten kann sehr gut durch ein einfaches Experiment nachvollzogen werden. Benötigt wird lediglich eine Pflanze, die Blätter (im obigen Sinne) besitzt und eine Lampe. Wird ein einzelnes Blatt zwischen Lichtquelle und Beobachter gehalten, so erscheint das transmittierte Licht sehr diffus (Abb. 2.13 rechts). Befinden sich Betrachter und Lampe auf der selben Seite des Blattes, so kann häufig ein starker Glanzpunkt (Abb. 2.13 links) auf der Blattoberfläche gesehen werden, der von dem reflektierten Licht herrührt. Diese Eigenschaft hat jedoch zur Folge, dass die vollständige Rendering-Gleichung (2.4) gelöst werden muss, wenn Pflanzenblätter dargestellt werden sollen. Im Gegensatz zu den Blättern, verhält sich der Stamm und das Geäst eines Baums fast vollständig opak, das heißt lichtundurchlässig. Somit reicht hier die vereinfachte Variante der Rendering-Gleichung (2.3) für eine realistische Darstellung



Abbildung 2.13: Reflektion (Links) und diffuse Transmission (rechts)

aus.

### 2.3.2 Pflanzendarstellung durch den Computer

Die Darstellung von natürlichen Szenen liegt seit den Anfängen der Computergrafik im Interessensfokus der Wissenschaft. Jede noch so schön modellierte Umgebung wirkt leblos und damit unrealistisch, wenn organische Objekte fehlen. Anfänglich wurden für die Simulation der optischen Eigenschaften von Pflanzenblättern vor allem stochastische Modelle eingesetzt. Zum Beispiel verwendeten Tucker und Garratt ([TG76]) eine Markov - Kette, um die Zustandsänderung von Photonen, die auf ein Blatt auftreffen, zu modellieren. Das Licht wurde dazu auf einem Random Walk durch ein abstrahiertes Schichtenmodell verfolgt. Am Ende befand sich ein Photon entweder im reflektierten, absorbierten oder transmittierten Zustand. Um repräsentative Ergebnisse zu erhalten, musste die Simulation jedoch sehr oft durchgeführt werden, was eine hohe Berechnungszeit mit sich brachte. Das Modell war für die Berechnung von Bildern zwar nicht optimal, wurde aber im Bereich der Fernüberwachung von Pflanzenpopulationen erfolgreich eingesetzt. Dabei wurden Photographien mit den berechneten Werten verglichen, aus Abweichungen konnten Rückschlüsse auf eventuelle Erkrankungen gezogen werden.

Später wurden im Bereich der Visualisierung von lichtdurchlässigen Materialien vor allem Raytracingmethoden eingesetzt. Hanrahan und Krüger ([HK93]) schlugen ein Modell vor, welches eine transluzente Oberfläche in mehrere unterschiedliche Schichten zerlegt. Trifft ein Lichtstrahl auf ein solches mehrschichtiges Objekt, so wird dieser bei der Interaktion mit dem Material beobachtet. Da das Verfahren, bei den internen Reflektionen und Refraktionen die Umgebung nicht mit einbezog, konnten Bilder sehr schnell berechnet werden. Allerdings wurden transluzente Materialien sehr allgemein betrachtet, was dazu führte, dass bei der Darstellung von Blättern wichtige pflanzenspezifische Effekte keine Berücksichtigung fanden.

Das Anwendungsgebiet des abstrahierten Schichtenmodells war hauptsächlich in der photorealistischen Bildsynthese zu suchen. Im Bereich der Fernüberwachung wurde dieses Vorgehen jedoch als zu ungenau angesehen und verworfen. Anstelle der Schichten wurde häufig ein sehr viel detaillierteres Modell verwendet, um die optischen Eigenschaften möglichst genau nachbilden zu können. Zum Beispiel bildeten Govaerts *et al.* ([GJVU96]) die einzelnen Zellen eines idealen Blattes (siehe Abb. 2.10) durch echte dreidimensionale Primitive nach. Dabei wurden die Palisadenzellen durch Zylinder mit abgerundeten Kappen modelliert. Jede Zelle bestand zudem aus mehreren Schichten, wie zum Beispiel Zellwand, Chlorophyll und Wasser. Zur Untersuchung der Lichtinteraktion wurde auch hier ein Monte - Carlo - Raytracingansatz eingesetzt: Trifft ein Lichtstrahl auf das Blatt, so wird dieser so lange bei der Interaktion mit den Zellen verfolgt, bis es entweder zur Absorption oder zum Verlassen des Lichtstrahls kommt. Die Simulation lieferte sehr realistische Ergebnisse, sowohl das spektrale Verhalten, als auch die BSSRDF konnten korrekt nachgebildet werden. Allerdings erwies sich das Modell als zu kompliziert, um es bei der Berechnung von Bildern einzusetzen. Die Simulation benötigte sehr viel Rechenzeit, was vor allem aus der korrekten Modellierung der Zellen folgte.

Später wurden die oben beschriebenen Verfahren von Hanrahan und Govaerts von den zwei Wissenschaftlern Baranoski und Rokne ([BR97]) kombiniert. Sie schlugen das so genannte „Algorithmische BSSRDF<sup>5</sup> Modell“ (ABM) vor, welches das Ziel verfolgte, realistische und biologisch motivierte Bilder zu berechnen. Die vollständige dreidimensionale Modellierung der Zellen wurde in diesem Ansatz verworfen und (abermals) durch ein abstrahiertes Schichtenmodell ersetzt. Doch anders als bei dem

---

<sup>5</sup>Die Autoren bezeichnen die BSSRDF als BDF, gemeint ist jedoch die selbe Funktion.

Vorgehen von Hanrahan und Krüger, fand die Anatomie von Pflanzenblättern dabei eine stärkere Berücksichtigung, um die Ergebnisse zu optimieren und somit realistischere Bilder zu berechnen. Im ABM besteht ein virtuelles Blatt aus drei (abstrahierten) Schichten. Die erste Schicht repräsentiert die obere Epidermis und die Palisadenzellen und ist deswegen auch hauptsächlich für die Reflektion und Absorption des eintreffenden Lichts zuständig. Entsprechend der Anatomie des idealen Pflanzenblattes befindet sich unterhalb der Palisadenzellen ein grobgranulares Gewebe, das zu einem großen Anteil aus Luft besteht. Auf Grund des anatomischen Unterschieds, welcher zwischen diesem Gewebe und den Palisadenzellen vorherrscht, definierten Baranoski und Rokne die zweite Schicht des ABM als eine aus Luft bestehende Ebene. Da die Zellen der unteren Epidermis ein anderes optisches Verhalten als Luft besitzen, bilden diese schließlich die dritte Ebene des abstrahierten Schichtenmodells. Die Interaktion der eintreffenden Photonen mit den einzelnen Schichten des ABM kann mit Hilfe von wenigen - biologisch motivierten - Parametern gesteuert werden. Ein Photon, welches auf ein Blatt trifft, wird entweder von der ersten Schicht absorbiert oder es verlässt das Gewebe nach einigen Reflektionen und Refraktionen wieder. Dadurch lässt sich das konkrete Reflektions- und Absorptionsverhalten der darzustellenden Pflanzenblätter sehr gut kontrollieren. Neben dieser einfachen Parametrisierbarkeit verfügt das ABM über einen weiteren Vorteil, denn es kann problemlos in jeden Monte - Carlo - Raytracer integriert werden. Der Nachteil dieses Ansatzes besteht allerdings darin, dass sehr viel Rechenzeit für die Berechnung eines Bildes benötigt wird. Deswegen schlugen Baranoski und Rokne später eine optimierte Variante des ABM vor ([BR01]), indem möglichst viele Faktoren vorberechnet wurden. Allerdings ist der Aufwand immer noch immens<sup>6</sup>, was die Berechnung von Animationen, welche Pflanzen enthalten, unpraktisch macht.

---

<sup>6</sup>Um ein Bild mit einigen Blättern zu berechnen, werden zirka 95 Minuten auf einer SGI R 10000 benötigt.

*Die Wissenschaft fängt eigentlich erst da an  
interessant zu werden, wo sie aufhört.*

Justus von Liebig (1803-73)

# 3 Realistische Pflanzendarstellung

## 3.1 Einführung

Dieses Kapitel beinhaltet die Entwicklung eines schnellen, photorealistischen und empirischen Darstellungsmodells für Pflanzenblätter und -gewebe, welches in Bereich des Offline-Renderings, also zum Beispiel in der Filmindustrie, eingesetzt werden kann. Zunächst werden die zu erreichenden Ziele näher betrachtet. Außerdem wird untersucht, warum die bisherigen Ansätze nur bedingt für das Rendering von Pflanzen geeignet sind und in wie weit die geometrischen Repräsentationen von Pflanzen von anderen Objekten abweichen. Auf Basis dieser Analyse wird anschließend ein Verfahren vorgestellt, welches die speziellen Voraussetzungen berücksichtigt und die gegebene Zielstellung erfüllt. Nachfolgend werden die erreichten Ergebnisse untersucht und bewertet. Eine Zusammenfassung mit einem Ausblick auf mögliche Erweiterungen rundet diese Kapitel ab.

### 3.1.1 Zielstellung

Bisherige Darstellungsmethoden können nur eingeschränkt für das Rendering von Pflanzen benutzt werden. Entweder berücksichtigen diese Verfahren überhaupt keine Pflanzen-spezifischen Eigenschaften oder sie versuchen möglichst viele physikalischen Effekte mit einzubeziehen, was zu einer sehr hohen Berechnungszeit führt. Hier gilt es einen Mittelweg zu finden. Zum Beispiel ist für das Offline Rendering ein korrektes spektrales Verhalten nur indirekt interessant. Viel wichtiger ist allerdings, dass die Objekte real wirken. Ob dafür eine exakte physikalische Modellierung oder eine grobe Vereinfachung der Lichtinteraktionen verwendet wird, ist letztendlich egal. Anstatt also zum Beispiel alle Zellen explizit nachzubilden, lässt sich der

Aufbau eines Blattes auch durch ein abstrahiertes Modell beschreiben. Bei dieser Vereinfachung muss allerdings sichergestellt werden, dass die Interaktion der eintreffenden Photonen mit dem Gewebe trotzdem realistisch simuliert wird. Somit ist - in diesem Kontext - ein vereinfachtes Modell, welches auf realistischen Parametern und Vorgängen basiert, optimal.

Eine weitere wichtige Voraussetzung, die zum Teil aus dem ersten Ziel folgt, ist die Zeitdauer, die benötigt wird, um ein Bild zu berechnen. Ein computergenerierter Film(-abschnitt) besteht im Allgemeinen aus Tausenden von Einzelbildern. Es ist offensichtlich, dass ein Darstellungsverfahren eine möglichst geringe Berechnungszeit besitzen sollte, um überhaupt in der Lage zu sein, in einem solchen Kontext eingesetzt zu werden. Häufig folgt aus der Verwendung eines vereinfachten Modells jedoch automatisch eine geringere Berechnungszeit. Somit sind diese beiden Ziele eng miteinander verknüpft.

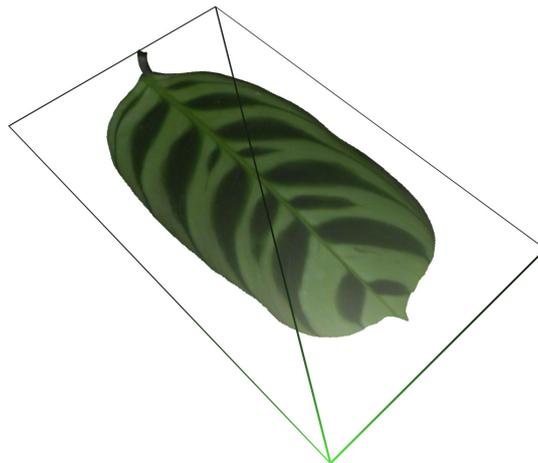


Abbildung 3.1: Geometrische Repräsentation eines Blattes

Eine weitere Besonderheit, die bei der Darstellung von Pflanzen berücksichtigt werden muss, stellt die Art der geometrischen Modellierung dar. Auf Grund der hohen Komplexität, die organische Objekte in der Regel besitzen, werden auch bei der Erstellung ihrer Geometrie Vereinfachungen verwendet. So wird ein Blatt häufig lediglich als eine endliche Ebene (siehe Abb. 3.1) modelliert, die ihre spezifische Form durch eine geschickte Wahl von Materialparametern erhält (Alpha Maskierung). Dieses Vorgehen fällt aber im Allgemeinen nicht auf, da reale Blätter - im Vergleich zu

ihrer Grundfläche - sehr dünn sind. Viele Darstellungsverfahren ([JMLH01], [JB02]) versagen jedoch, wenn das betrachtete Objekt über kein Volumen verfügt. Um Pflanzen realistisch zu rendern, gilt es somit auch diese geometrischen Bedingungen zu beachten. Im nächsten Abschnitt wird deshalb die Modellierung von Pflanzen näher betrachtet.

Ein optimales Modell, zur photorealistischen Darstellung von Pflanzen, sollte somit folgende Bedingungen erfüllen:

- **Parametrisierbarkeit:** Das Verfahren lässt sich über wenige, biologisch motivierte und intuitiv verständliche Parameter steuern.
- **Effizienz:** Da in der Regel sehr viele Bilder mit dieser Methode erstellt werden, gilt es den Zeitaufwand so gering wie möglich zu halten.
- **Geometrische Approximation:** Das Darstellungsmodell sollte auch auf Basis einer stark vereinfachten Geometrie arbeiten.
- **Flexibilität:** Es ist darauf zu achten, dass ein hohes Maß an künstlerischer Freiheit erreicht wird.

#### 3.1.2 Modellierung von Pflanzen

In diesem Abschnitt soll ein kurzer Einblick in die Art und Weise gegeben werden, wie ein Künstler die geometrische Form einer (virtuellen) Pflanze erstellen kann. Dieser Vorgang wird häufig als „Modellierung“ bezeichnet und bildet damit die Basis für die spätere Berechnung von Bildern. Wie bereits in den vorherigen Abschnitten erwähnt wurde, besitzen künstliche Pflanzen in vielen Bereichen eine Sonderstellung in der Computergrafik. Dies gilt auch im Bereich der (geometrischen) Modellierung. „Standardobjekte“ werden von einem Künstler aus einzelnen geometrischen Primitiven zusammengesetzt und auf verschiedenen Verfeinerungsebenen bearbeitet. Zunächst wird die grobe Form des gewünschten Objekts aus Quadern, Zylindern, Kugeln usw. zusammengesetzt. Danach wechselt der Künstler zumeist in eine detailliertere Darstellung. Dabei können zum Beispiel einzelne Dreiecke oder gar Eckpunkte modifiziert werden. Dieses Vorgehen funktioniert aber lediglich bei Modellen

mit moderater Komplexität. Pflanzen können aber nur sehr selten in diese Kategorie eingeordnet werden. Ein einzelner Baum besteht aus abertausenden Elementen, wie Blättern oder Äste. Dies führt jedoch dazu, dass ein Künstler nicht mehr in der Lage ist, jedes einzelne Detail per Hand zu modifizieren.

Tatsächlich ist dieses Problem schon seit den Anfängen der Computergrafik bekannt. Viele Verfahren wurden vorgeschlagen, um die Modellierung von Pflanzen zu ermöglichen. Die entwickelten Vorgehensweisen lassen sich in eine der drei folgenden Kategorien einordnen: Prozedurale Methoden, Regel basierte Verfahren und kombinierte Vorgehensweisen.

Bei der **prozeduralen Modellierung** wird die Geometrie der Objekte mit Hilfe einer Prozedur erzeugt. Das Ergebnis kann dabei in der Regel durch einen Satz von Parametern gesteuert werden. Somit ist der Künstler nur noch teilweise im Modellierungsprozess involviert, er legt die konkreten Werte der Parameter fest und bewertet das erzeugte Ergebnis. Dabei stellt die prozedurale Modellierung die älteste Methode zur Erstellung von organische Strukturen dar. So erzeugte Stanislaw Ulam bereits 1966 wachsende Verzweigungsstrukturen mit Hilfe eines zellulären Automaten. Nur wenig später beschrieb der Biologe Dan Cohen den Wachstumsvorgang von Pflanzen auf der Basis von sehr einfachen Regeln. Über die Jahre wurden die verwendeten Methoden immer weiter ausgebaut, die Ergebnisse ähnelten dabei immer mehr echten Pflanzen. In der Praxis setzten deswegen viele Künstler die prozedurale Modellierung ein. Die große Beliebtheit dieser Methode stellt jedoch kein Zufall dar, denn diese Vorgehensweise hat den zusätzlichen Vorteil, dass sie sich sehr leicht parametrisieren lässt. Nach der Änderung eines Parameterwertes wird die Prozedur erneut ausgeführt und wenig später kann das erzeugte Ergebnis begutachtet werden. Ein Künstler kann somit sehr einfach das Aussehen der beschriebenen Objekte anpassen. Allerdings folgen auch einige Nachteile aus der Verwendung von prozeduralen Verfahren. Die größte Einschränkung stellt wohl die mangelnde Flexibilität dar. Häufig repräsentiert eine Prozedur nämlich nur eine einzelne Kategorie von Objekten. Ein Algorithmus der einen Apfelbaum generiert, kann nur sehr schwer dazu verwendet werden, um Rosen zu erzeugen. Außerdem kann ein Künstler nur die Details anpassen, die zuvor vom Programmierer der Prozedur vorgesehen wurden. Hat dieser einen Parameter vergessen, so lässt sich dies im Nachhinein nicht mehr än-

dern. Nähere Informationen zum Thema der prozeduralen Modellierung lassen sich in [Deu03] finden.

Die **Regel basierte Modellierung** geht einen ganz anderen Weg als die prozedurale Methode. Ihr Ursprung liegt in der theoretischen Informatik. Dabei wird ein Textersetzungs-system mit Hilfe einer Menge von Regeln gesteuert. In einem System werden Zeichen durch andere Zeichenketten ersetzt, wobei das ursprüngliche Wort in der Regel an Länge gewinnt. Um mit einem solchen Vorgehen Geometrie zu erzeugen, werden die einzelnen Zeichen des Wortes geometrisch interpretiert. Dafür wird die so genannte Turtle - Methapher verwendet. Während sich die Turtle bewegt, hinterlässt sie eine „Spur“, die zum Beispiel aus Eckpunkten eines Polygons besteht. Dabei wird die Bewegungsrichtung der Turtle durch die einzelnen Zeichen des Wortes verändert. Auch andere Parameter lassen sich auf diese Art und Weise steuern, so könnte zum Beispiel ein spezielles Zeichen die Farbe, der von der Turtle hinterlassenen, Spur variieren, der Phantasie sind dabei kaum Grenzen gesetzt. Regelbasierte Systeme sind sehr flexibel. Die Änderung einer einzelnen Regel kann ein ganz anderes Ergebnis erzeugen. Mit ein und dem selben System lässt sich sowohl ein Apfelbaum als auch eine Rose beschreiben. Doch genau in dieser Eigenschaft liegt der größte Nachteil dieser Methode. Sie ist lässt sich nur schwer kontrollieren und zeigt teilweise ein chaotisches Verhalten: Bereits kleinste Änderungen können das Ergebnis sehr stark verändern. Auf der anderen Seite eignet sich dieses Verfahren scheinbar sehr gut, um die komplexen Verzweigungsstrukturen von Pflanzen zu modellieren. Mit der regelbasierten Modellierung von Pflanzen haben sich vor allem die Wissenschaftler Prusinkiewicz und Lindenmayer beschäftigt. Eine umfangreichere Einführung in dieses Thema kann in [PL90] und [Deu03] gefunden werden.

Um die einfache Modellierung von möglichst vielen verschiedenen Pflanzen zu ermöglichen, empfiehlt es sich die prozedurale Methode mit einem regelbasierten System zu verbinden. Diese **kombinierte Vorgehensweise** kann dann die jeweiligen Vorteile der beiden Modellierungsmethoden benutzen. Ein solches Verfahren wurde von Deussen und Lintermann ([DL97], [DHL<sup>+</sup>98], [Deu03]) vorgeschlagen. Das Ziel bestand vor allem darin, eine interaktive Methode zur Verfügung zu stellen, um ein zügiges Modellieren zu ermöglichen. Dafür ist es jedoch notwendig, dass sich

ein solches System leicht steuern lässt, denn nur so kann eine bestimmte Pflanze in einer relativ kurzen Zeit von einem Künstler erstellt werden. Aus diesem Grund basiert die Erstellung der Geometrie auf der prozeduralen Modellierung. Im Bezug auf die allgemeine Struktur der beschriebenen Objekte muss dabei ein hohes Maß an Flexibilität garantiert werden, um möglichst viele Pflanzenarten repräsentieren zu können. In diesem Bereich ist die Verwendung einer regelbasierten Methode optimal. Die Vereinigung der beiden Ansätze erfolgt durch ein Komponenten-basiertes Vorgehen. So existieren zum Beispiel Komponenten die Geometrie erzeugen und andere die eine Verzweigungsstruktur erstellen. Somit wird es möglich, Pflanzen als ein Graph von Komponenten darzustellen. Vor der Traversierung wird zunächst die Expansion der Rekursions- und Multiplikationskomponenten durchgeführt, danach existiert die vollständige Struktur der Pflanze. Schließlich werden alle Geometrieerzeugenden Komponenten ausgeführt. Ein Künstler hat nun die Möglichkeit neue Komponenten einzufügen und die vorhandenen Parameter zu verändern. Die Darstellung einer Pflanze durch einzelne Komponenten ermöglicht dabei eine lokale und damit Problem bezogene Modellierung. Die Änderungen können interaktiv gesteuert werden, was die Arbeit eines Künstlers wesentlich vereinfacht. Auf der anderen Seite ist die Komponenten-basierte Repräsentation nicht so abstrakt wie ein regelbasiertes Verfahren. Die Struktur kann über intuitive Parameter wie die Rekursionstiefe oder die Vervielfältigungsanzahl gesteuert werden, wobei ein Künstler nie mit der vollständigen Komplexität zu tun hat. Dieses Vorgehen konnte sehr erfolgreich in dem Modellersystem *xfrog* (*[xfr]*) umgesetzt werden. Mit Hilfe dieser Software wurden unzählige verschiedene Pflanzen modelliert, was den Erfolg wohl zweifelsfrei beweist. An dieser Stelle konnte nur eine sehr allgemeine Beschreibung dieser Methode gegeben werden, sie wird sehr detailliert in [Deu03] besprochen.

## 3.2 Realistisches Rendering von Pflanzenblättern

Auf Basis der Erkenntnisse, die in den vorherigen Abschnitten erläutert wurden, soll es an dieser Stelle um die Entwicklung eines einfachen, aber dennoch biologisch motivierten, Algorithmus gehen, mit dem es möglich ist, (lichtdurchlässiges) Pflanzengewebe photorealistisch darzustellen. Nachdem das eigentliche Verfahren erläutert

wurde, wird ein Blick auf die Erstellung der erforderlichen Datensätze geworfen.

Auch wenn nachfolgend hauptsächlich Pflanzenblätter betrachtet werden, so stellen diese nicht die einzige Anwendungsmöglichkeit dar. Die Betrachtungen lassen sich auch auf andere lichtdurchlässige Bestandteile einer Pflanze verallgemeinern. Die opaken Elemente einer Pflanze, wie der Hauptstamm oder das Geäst, lassen sich sehr gut mit einem Standardverfahren nachbilden, weswegen sie an dieser Stelle nicht näher betrachtet werden.

#### 3.2.1 Entwicklung eines empirischen Darstellungsmodells

Wie bereits in der Einführung dieses Kapitels erwähnt wurde, stellt die exakte Modellierung aller optischen Effekte innerhalb eines Blattes nicht das zu erreichende Ziel dar. Vielmehr gilt es, die sichtbaren visuellen Effekte zu analysieren und möglichst effizient nachzubilden. Dazu dienen natürlich die Ergebnisse der vorangegangenen biologischen und optischen Untersuchungen (Kapitel 2).

Um die zwei visuellen Grundzustände - die es bei der Darstellung von Pflanzenblättern nachzubilden gilt - zu veranschaulichen, soll an dieser Stelle das bereits zuvor genannte einfache Experiment noch einmal etwas genauer betrachtet werden. Dafür wird lediglich ein frisches Pflanzenblatt und eine Lichtquelle benötigt.

1. **Frontallichtsituation** - Beobachter und Lichtquelle befinden sich auf der selben Seite des Blattes. Der visuelle Eindruck besteht in diesem Fall aus der Kombination von diffus und spekulär reflektiertem Licht (Abb. 3.2 links).
2. **Durchlichtsituation** - Das Pflanzenblatt ist zwischen Lichtquelle und Betrachter lokalisiert. Dieser Grundzustand wird durch die diffuse Transmission dominiert (Abb. 3.2 rechts).

Der erste Grundzustand - die Frontallichtsituation - ist zudem von der betrachteten Seite abhängig. Dieses Verhalten folgt aus der Anatomie des Blattes (siehe Abb. 2.10). Auf der Oberseite befindet sich eine stark ausgeprägte Wachsschicht, was zu einem großen spekulären Anteil führt. Darunter ist das Mesophyll zu finden, indem sich die Chloroplasten befinden. Treffen Photonen auf die Oberfläche, so wird ein Teil davon

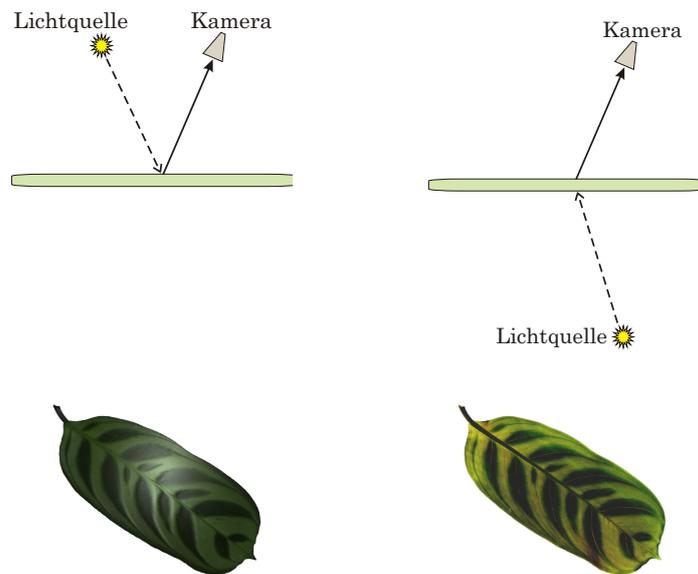


Abbildung 3.2: Visuelle Grundzustände

in das Innere des Blattes weitergeleitet. Durch den Einfluss des Chlorophyll kommt es zur Absorption eines bestimmten Anteils des Lichts (Nicht-Grünes Spektrum). Durch die Streuung verlässt ein Teil des Lichts das Pflanzenblatt wieder auf der Oberseite (siehe Abb. 2.11).

Auf der Unterseite des Pflanzenblattes ist zwar ebenfalls eine Wachsschicht vorhanden, diese ist aber nicht so stark - wie auf der Oberseite - ausgeprägt. Außerdem lassen sich direkt unter der Oberfläche keine Chloroplasten finden, sondern lediglich eine grobgranulare Schicht, die das einfallende Licht stark streut. Viele Photonen erreichen die Palisadenzellen gar nicht mehr, sondern werden vorher absorbiert oder zurück reflektiert. Daraus folgt, dass die Unterseite häufig blasser erscheint, als die Oberseite.

Der zweite visuelle Grundzustand - die Durchlichtsituation - ist nicht von der betrachteten Seite abhängig. Es spielt keine Rolle ob die Ober- oder die Unterseite beobachtet wird. Der Grund für dieses Verhalten ist offensichtlich, denn bei der Transmission müssen die Photonen das gesamte Gewebe durchqueren. Ob das Licht nun von oben oder von unten eingestrahlt wird ist letztendlich egal.

Nachdem die zwei visuellen Grundzustände und deren Ursachen geklärt sind, kann

nach einem Weg gesucht werden, wie sich diese Effekte effizient nachbilden lassen. Da die diffuse Transmission nur mit Hilfe der BSSRDF beschrieben werden kann, gilt es die vollständige Rendering-Gleichung (Gleichung 2.4) zu lösen. Da dies jedoch analytisch nur schwer möglich ist, wird ein numerischer Ansatz benötigt. Die Idee besteht darin, durch das Objekt zu „wandern“ und dabei die reflektierte Radianz, mit Hilfe einer stückweise linearen Approximation, zu berechnen.

Im Kontext des Pflanzenrenderings ist damit jedoch ein Problem verbunden, welches aus der geometrischen Repräsentation folgt: In der Regel verfügen die Repräsentation der Blätter über kein Volumen, durch das geschritten werden könnte. Wie in Abbildung 3.1 zu sehen ist, könnte ein solches Objekt - im Extremfall - lediglich aus einer (endlichen) Ebene bestehen. Um diese Ungenauigkeit des geometrischen Modells zu umgehen, kann eine virtuelle parallele Begrenzungsebene (Abb. 3.3 unten) eingeführt werden. Dabei wird jedem Punkt  $(x, \vec{n})$  auf der Oberfläche ein weiterer virtueller Punkt  $(x', \vec{n}')$  auf der gegenüberliegenden Seite zugeordnet:

$$x' = x - d(x) \cdot \vec{n} \quad (3.1)$$

$$\vec{n}' = -\vec{n} \quad (3.2)$$

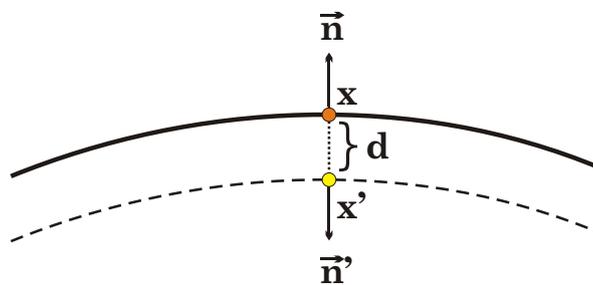


Abbildung 3.3: Virtuelle parallele Ebene

Durch diese neue Begrenzungsebene, wird ein virtuelles Volumen definiert, in dem sich die Zellen des Pflanzenblattes befinden. Dabei bietet es sich an, gleichartige Zellen zusammenzufassen und als eine kombinierte Gewebeschicht zu beschreiben (vgl. Abschnitt 2.3.1.2). Da sich in einer solchen idealisierten Schicht nur Elemente der gleichen Kategorie befinden, ist auch das Interaktionsverhalten der Photonen mit den Zellen dieser Schicht fast identisch. Anstatt also während des Renderings explizit

einzelne Zellen zu simulieren, macht es durchaus Sinn lediglich die abstrahierten Gewebeschichten, die sich im virtuellen Volumen befinden, zu betrachten und bei der Berechnung der reflektierten Radianz einzusetzen. Der Vorteil dieses Vorgehens besteht vor allem darin, dass sehr viel Rechenzeit gespart wird, da nun keine exakte physikalische Modellierung der einzelnen Zellen mehr nötig ist. Ein ähnliches Vorgehen wurde auch von Baranoski und Rokne ([BR97], [BR01]) vorgeschlagen.

Beim Entwurf des konkreten Schichtenmodells muss - neben der Anatomie des idealen Pflanzenblattes (siehe Abb. 2.10) - auch das Brechungsgesetz berücksichtigt werden. Schließlich findet die Refraktion von Licht lediglich an der Grenze zwischen zwei Medien statt, die über verschiedene Brechungsindices verfügen. Da das grobgranulare Gewebe, welches sich zwischen den Palisadenzellen und der unteren Epidermis befindet, zu 50% bis 80% aus Luft besteht, besitzt es ungefähr den Refraktionsindex von Luft (ca. 1). Das umliegende Gewebe ist jedoch anders aufgebaut, denn es wird aus dicht gepackten pflanzlichen Zellen gebildet und verfügt somit auch über einen anderen Brechungsindex (ca. 1,33). Es sind also mindestens drei (abstrahierte) Schichten notwendig, um das Brechungsverhalten innerhalb des idealen Pflanzenblattes zu modellieren. Da die obere Epidermis und die Palisadenzellen ein ähnliches optisches Verhalten besitzen, können sie zusammengefasst werden und bilden dadurch die erste Schicht des Modells. Das bereits erwähnte grobgranulare Gewebe stellt die zweite Ebene dar, während die dritte Schicht durch die untere Epidermis gebildet wird. In Abbildung 3.4 ist das vorgeschlagene Schichtenmodell schematisch dargestellt.

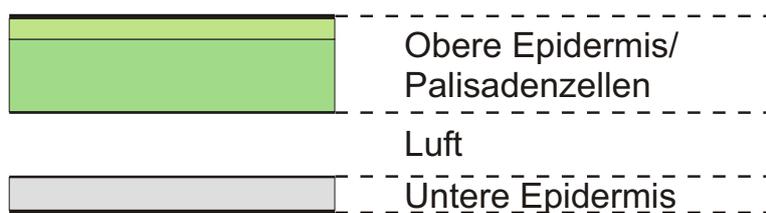


Abbildung 3.4: Idealisiertes Schichtenmodell

Die einzelnen Schichten beschreiben aber nicht nur das Brechungsverhalten, des modellierten Gewebes. Durch die Variation der Dicke der Ebenen wird die ausgehende Radianz direkt beeinflusst. Wenn zum Beispiel die Höhe der Palisadenzellen-Schicht

verkleinert wird, dann existieren auch weniger Chloroplasten im Gewebe, was wiederum dazu führt, dass mehr Photonen des Nicht-Grünen Spektrums das Blatt wieder verlassen können. Um eine einfache Editierbarkeit zu gewährleisten, wird die Größe der Schichten relativ zur Blattdicke  $d$  definiert, wobei die Dicke der einzelnen Ebenen durch die Werte  $e_1$ ,  $e_2$  und  $e_3$  gegeben ist. Auf Grund der relativen Beschreibung muss somit stets die folgende Bedingung gelten:

$$e_1 + e_2 + e_3 = 1.$$

Im Fall der Frontallichtsituation gilt es zusätzlich die aktuell betrachtete Seite zu erkennen. Sei  $c$  die Position des Betrachters,  $x$  ein Punkt auf der Oberfläche und  $\vec{n}$  die entsprechende Normale, dann lässt sich die normierte Blickrichtung folgendermaßen bestimmen:

$$\vec{v} = \frac{x - c}{|x - c|}$$

Auf Basis dieser Daten lässt sich nun die folgende Funktion definieren:

$$F(\vec{n}, \vec{v}) = \begin{cases} 0, & \vec{n} \cdot \vec{v} < 0 \\ 1, & \vec{n} \cdot \vec{v} \geq 0 \end{cases} \quad (3.3)$$

Die Treppenfunktion  $F(\vec{n}, \vec{v})$  entscheidet, ob die Ober- oder die Unterseite des Pflanzenblattes betrachtet wird. Ist der Wert von  $F$  gleich Null, so wird die Vorderseite beobachtet, bei einem Wert von Eins die Rückseite.

Damit sind alle Informationen vorhanden, die benötigt werden, um die Rendering-Gleichung zu lösen und damit die ausgehende Radianz zu bestimmen. Trifft ein Sehstrahl auf ein Pflanzenblatt, so gilt es zunächst zu entscheiden, ob das Objekt von oben oder von unten betrachtet wird, dies geschieht mit Hilfe der Funktion  $F$ . Dementsprechend kann der refraktierte Strahl, ausgehend von der ersten (Vorderseite) bzw. dritten (Rückseite) Schicht, in das Innere des Gewebes verfolgt werden. Die durchquerten Zellen bestimmen, in welchem Umfang die eintreffenden Photonen absorbiert bzw. in die Richtung des Betrachters gestreut werden (siehe Abb. 3.5).

Diese detaillierte Modellierung der Lichtinteraktionen innerhalb des Blattes zieht allerdings einige Komplikationen nach sich. Zuerst einmal ist es notwendig, das Verhalten der Zellen sehr genau zu beschreiben. Dies führt aber in der Regel zu vielen

unintuitiven Parametern, wie zum Beispiel dem Durchmesser der Zellen. Weiterhin kommt es durch die feinen Strukturen des Blattskeletts zusätzlich zu einer räumliche Variationen dieser Parameter, was die Situation weiter verkompliziert. Soll jedoch eine photorealistische Darstellung erreicht werden, dann sind diese filigranen Details von großer Bedeutung, denn ein Betrachter bemerkt sehr schnell ob die feinen Strukturen des Blattes vorhanden sind oder nicht. Eine Lösung für dieses Problem stellt die Verwendung einer Textur dar, welche vollständig die - von der Oberfläche - diffus reflektierte Radianz (diffuse BSSRDF) beschreibt. Dies erspart dem Künstler das aufwendige Anpassen der Parameter und vereinfacht zudem den Algorithmus, was wiederum die Berechnungszeit verkürzt.

Allerdings existieren auch einige Nachteile bei diesem Vorgehen. Zunächst wird jeweils für die Vorder- und die Rückseite eine separate Textur benötigt. Außerdem gilt es einen Weg zu finden, wie die Durchlichtsituation nachgebildet werden kann. Neben dem reflektierten Lichtanteil kann auch die transmittierte Radianz durch eine Textur beschrieben werden. Allerdings ist die Erstellung dieses Datensatzes, im ersten Augenblick, nicht trivial. Die präsentierten Nachteile lassen sich aber akzeptieren, weil die erreichten Vorteile in diesem Fall dominieren. Mehr Informationen darüber, wie die einzelnen Texturen erstellt werden können, lassen sich im nächsten Abschnitt finden. Die drei genannten Texturen (Vorder-, Rückseite und Durchlichtsituation) sind in Abbildung 3.7 zu sehen.

Nachfolgend wird vorausgesetzt, dass die oben genannten Texturen zur Verfügung stehen. Dadurch ist eine vollständige Beschreibung des reflektierten bzw. transmittierten Lichtanteils vorhanden und zwar ohne das ein detailliertes Interaktionsmodell der Zellen nötig wäre. Um die Rendering-Gleichung zu lösen, muss also lediglich die korrekte eingehende Radianz bestimmt werden. Dafür ist es notwendig schrittweise entlang des aktuellen Sehstrahls durch das Volumen des Pflanzenblattes zu wandern und das einfallende Licht zu analysieren. Da auch für den Sehstrahl die Gesetze der Optik gelten, wird dieser ebenfalls an den Grenzen zwischen den abstrahierten Schichten gebrochen, was denn Wert der resultierenden Radianz beeinflusst. In Abbildung 3.5 kann dieses Verhalten betrachtet werden.

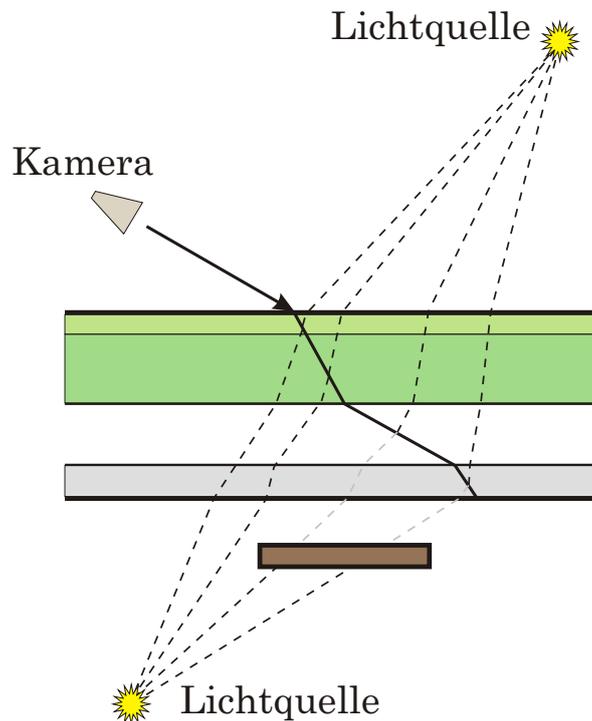


Abbildung 3.5: Lösung der Rendering-Gleichung in Pflanzengewebe

Wie aus der Abbildung erkenntlich wird, kann der Weg des Sehstrahls innerhalb des Blattes - unter gewissen Umständen - relativ lang werden. Dies gilt insbesondere dann, wenn die Oberfläche in einem sehr kleinen Winkel betrachtet wird. Dadurch entsteht jedoch die Möglichkeit, dass die - in der aktuellen Szene enthaltenen - Lichtquellen partiell verdeckt werden. Dieser Effekt ist im gerenderten Ergebnisbild als Penumbra-Region zu erkennen. Zum Beispiel wird der Einfluss, der unteren Lichtquelle in Abbildung 3.5, teilweise von einem anderen Objekt (Abb. 3.5: unteres Rechteck) blockiert. Dieses visuelle Verhalten ist sehr wichtig, denn es verhilft dem dargestellten Material zu einem weichen Aussehen.

Die Beschattung stellt zwar ein wichtigen Einflussfaktor auf die ausgehenden Radianz dar, sie ist aber nicht als der alleinige Effekt. Die Länge des Weges, der von den Photonen innerhalb des Gewebes zurückgelegt werden muss, repräsentiert eine weitere wichtige Größe. Es ist leicht einzusehen, dass ein längerer Lichtpfad zu einer geringeren Helligkeit führen muss. Schließlich werden auf dem Weg durch das Gewebe beständig Photonen absorbiert oder in eine andere Richtung gestreut. Dieses

Verhalten wurde zuerst von Lambert (1728 - 1777) und Beer (1825 - 1863) untersucht. Das - nach ihnen benannte - Lambert-Beersche-Gesetz lautet folgendermaßen:

$$I(s) = I_0 e^{-\alpha s} \quad (3.4)$$

Dabei ist mit  $s$  die Länge des zurückgelegten Weges gemeint. Der Absorptionskoeffizient  $\alpha$  sagt aus, wie stark die Abschwächung der beschriebenen Intensität  $I$  entlang des Pfades ausfällt. Wie aus der Formel (3.4) erkenntlich wird, erfolgt der Intensitätsabfall exponentiell, weswegen der Wert  $s$  einen sehr großen Einfluss auf das Ergebnis besitzt. Doch genau diese Eigenschaft stellt im Kontext der Pflanzendarstellung ein Problem dar. Die exakte Berechnung eines Photonenpfades ist nämlich mit einem sehr großem Aufwand verbunden. Werden die - in Abbildung 3.5 dargestellten - refraktierten Lichtvektoren betrachtet, so ist dieses Problem im ersten Augenblick gar nicht offensichtlich. Schließlich stellt sich der Pfad von einer Lichtquelle zum betrachteten Punkt als ein stückweise linearer Weg dar. Allerdings ist es nicht trivial die korrekten Schnittpunkte auf der Oberfläche zu finden. Zur Veranschaulichung soll hierbei Abbildung 3.6 gelten.

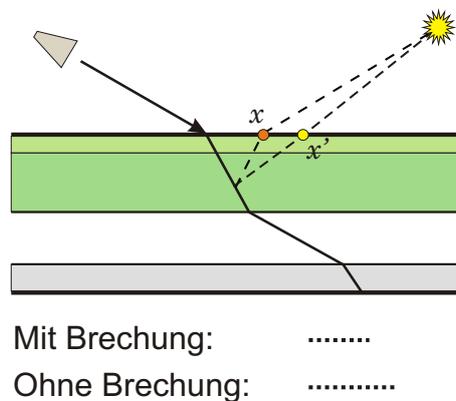


Abbildung 3.6: Inkorrekte Refraktion

Der korrekte Pfad von der Lichtquelle zum Interaktionspunkt in der ersten Schicht wird durch die obere gestrichelte Linie repräsentiert. Die Länge des Weges innerhalb des Gewebes ist im unteren Teil der Abbildung - in Form einer gepunkteten Linie - zu sehen. Um den korrekten Pfad der Photonen innerhalb des Gewebes und damit auch die reale Länge des Lichtpfades zu berechnen, muss der Schnittpunkt  $x$  (Abb.

3.6: Punkt  $x$ ) gefunden werden. Das Problem besteht jedoch darin, dass sich der Zielpunkt des Lichtstrahls, der nötig wäre um diesen speziellen Schnittpunkt zu bestimmen, nur sehr schlecht berechnen lässt. Wird jedoch lediglich der (direkte) Strahl von der Lichtquelle zum aktuellen Interaktionspunkt mit der Oberfläche geschnitten, so lässt sich nur die „falsche“ Stelle  $x'$  (Abb. 3.6: Punkt  $x'$ ) finden. Die Bestimmung des Zielpunkts des Strahls ist dabei sehr leicht, denn dafür wird einfach der momentane Interaktionspunkt benutzt. Allerdings stimmt in diesem Fall die Länge dieses (inkorrekten) Photonenpfads nicht mit dem gesuchten Wert überein (Abb. 3.6 untere gepunktete Linien). Je flacher der Auftreffwinkel des Lichts wird, desto größer wird auch die Deviation der Pfade.

Da die Abweichung zwischen  $x$  und  $x'$  - im Vergleich zu anderen Objekten in der Welt - sehr klein ist, kann zumindest bezüglich der Schattenberechnung eine Vereinfachung getroffen werden. Dabei stellt nämlich der inkorrekte Strahl eine hinreichend genaue Approximation dar. Wenn der Punkt  $x'$  im Schatten liegt, so gilt dies fast immer auch für die Stelle  $x$ . Die gleiche Annahme kann jedoch nicht für die Länge des Lichtpfads innerhalb des Pflanzengewebes eingesetzt werden, denn der exponentielle Charakter der Intensitätsabschwächung führt auch bei einer kleinen Variationen von  $s$  zu starken Änderungen im Ergebnis. Deswegen schlugen Jensen *et al.* eine Korrekturformel vor, welche - auf Basis des leicht zu berechnenden Schnittpunkts  $x'$  - die reale Länge des Lichtpfads innerhalb des Gewebes bestimmt.

$$s = s' \frac{|\vec{\omega} \cdot \vec{n}'|}{\sqrt{1 - \left(\frac{1}{n}\right)^2 \left(1 - |\vec{\omega} \cdot \vec{n}'|^2\right)}} \quad (3.5)$$

Mit  $s$  ist dabei die korrigierte Länge gemeint, während  $s'$  die „falsche“ Länge bezeichnet. Der normalisierte Lichtvektor wird  $\vec{\omega}$  genannt und die Normale an der Stelle  $x'$  erhält die Bezeichnung  $\vec{n}'$ . Das Verhalten dieser Formel lässt intuitiv sehr leicht verstehen. Je flacher der Einstrahlungswinkel des Lichts wird, desto mehr wird die inkorrekte Pfadlänge verkürzt. Treffen die Photonen senkrecht auf die Oberfläche, so erfolgt keine Veränderung von  $s'$ , denn in diesem Fall gilt immer  $s = s'$ .

Mit Hilfe der Korrekturformel (3.5) lässt sich nun also die Lichtabschwächung - innerhalb von Pflanzengewebe - durch das Lambert-Beersche-Gesetz (Gleichung 3.4) bestimmen. Denn für die Berechnung der Lichtintensität an einem beliebigen Punkt

im Gewebe, müssen korrekte Werte für die Länge des (internen) Lichtpfads und für den Absorptionskoeffizient vorhanden sein. In realen Pflanzenblättern ist jedoch weder die Dicke des Gewebes  $d$  noch der Parameter für die Absorption  $\alpha$  ein konstanter Wert. Vielmehr variieren diese Attribute innerhalb des Blattes. Beispielsweise ist das Gewebe von einem Blattskelett durchsetzt, wobei die Adern in der Regel eine höhere optische Dichte besitzen, wodurch mehr Photonen absorbiert bzw. gestreut werden. Außerdem ist das Skelett häufig dicker als das umgebende Gewebe. Da die geometrische Repräsentation diese feinen Details aber nicht modelliert (siehe Abb. 3.1), muss ein anderer Weg gefunden werden, um diese Variationen darzustellen. Um die größtmögliche Flexibilität und eine gute Parametrisierbarkeit zu gewährleisten, bietet sich auch bei diesem Problem die Verwendung von Texturen an. So kann die konkrete Dicke eines Blattes durch einen Maximalwert  $d_{max}$  und eine Modulationstextur  $T$  beschrieben werden. Der Absorptionskoeffizient wird ebenfalls mit Hilfe einer Variationstextur  $A$  und einem Minimalwert  $\alpha_{min}$  und Maximalwert  $\alpha_{max}$  dargestellt.

$$d(x) = d_{max} \cdot T(x) \quad (3.6)$$

$$\alpha(x) = \alpha_{min} \cdot (1 - A(x)) + \alpha_{max} \cdot A(x) \quad (3.7)$$

Die Modulationstexturen  $T$  und  $A$  lassen sich aus der bereits vorhandenen Durchlichttextur extrahieren. Eine detailliertere Erläuterung des Erstellungsprozesses wird es im nächsten Abschnitt (3.2.2) geben. Nachfolgend wird deshalb davon ausgegangen, dass diese beiden Modulationstexturen bereits zur Verfügung stehen. Somit lässt sich an jedem Punkt des beschriebenen Gewebes ein konkreter Wert für den Absorptionskoeffizienten und die Dicke bestimmen. Die Variation der Dicke besitzt jedoch - neben dem oben beschriebenen Einfluss auf die Lichtintensität - noch eine weitere Auswirkung auf die Berechnung der reflektierten Radianz. Sie führt nämlich außerdem zu einer Ortsabhängigkeit der Oberflächennormale, denn durch die neue Textur  $T$  wird ein Relief beschrieben. Das darzustellende Pflanzengewebe erhält somit eine unregelmäßige Oberfläche, auf der es (kleine) Hügel und Täler gibt. Da die Normale per Definition senkrecht auf einer Oberfläche steht, muss sie entsprechend dieses Reliefs angepasst werden. Dabei kommt es gerade zwischen Hügeln und Tälern zu einer relativ starken Verwerfung der Oberflächennormale, da dort der Anstieg des Reliefs am Größten ist. Diese Eigenschaft ist insbesondere für die spekulare Reflexion von entscheidender Bedeutung, weil diese sehr stark von der Richtung des reflektierten Lichtstrahls abhängt. Da die Oberflächennormale einen großen Einfluss

auf die Berechnung der Reflektionsrichtung besitzt, ist eine realistische Modellierung des Reliefs sehr wichtig. Die Verwerfung der Normale äußert sich visuell vor allem dadurch, dass die filigranen Details auf der Oberfläche hervortreten und besser sichtbar werden.



Abbildung 3.7: Texturen (von Links: Vorderseite, Rückseite, Durchlicht)

In Abbildung 3.7 sind die notwendigen (Radianz-)Texturen zu sehen. Wie bereits weiter oben erwähnt wurde, repräsentieren diese Datensätze lediglich den diffus reflektierten Anteil der BSSRDF. Der zweite wichtige Einflussfaktor stellt die spekulare Reflektion des einfallenden Lichts dar. Unter bestimmten Bedingungen überwiegt der spekulare Anteil sogar die diffuse Reflektion. Dies geschieht genau dann, wenn sich der Betrachter entlang der reflektierten Lichteinstrahlungsrichtung befindet. Ein Beispiel für diese - so genannte - spekulare Reflektionskeule ist in Abbildung 2.12 (oben rechts) zu sehen. Es ist zwar grundsätzlich möglich, den spekularen Anteil der BSSRDF ebenfalls durch eine Textur zu beschreiben, dies ist jedoch meist zu kompliziert und speicheraufwändig. Über die Jahre wurden deswegen viele Modelle vorgeschlagen, welche die Berechnung der spekularen Reflektion ermöglichen. Die bekanntesten - und wohl auch am häufigsten eingesetzten - Verfahren sind die Reflektionsmodelle von Phong ([Pho73]) und Blinn ([Bli77]). Tatsächlich können diese Methoden auch im Kontext der Pflanzendarstellung eingesetzt werden. Trotz ihrer algorithmischen Einfachheit, approximieren sie den spekularen Anteil der BSSRDF relativ gut. Deswegen macht es Sinn die BSSRDF in Einzelterme aufzuspalten, die dann unabhängig von einander berechnet werden können. Durch das schrittweise Durchwandern der Schichten des Gewebes, wird - mit Hilfe der Texturen - der diffuse Anteil der BSSRDF berechnet. Unabhängig davon kann das spekulare reflektierte

Licht durch ein approximatives Verfahren dargestellt werden. Ein weiterer angenehmer Nebeneffekt, der durch die Zerlegung der BSSRDF in Einzelterme entsteht, ist die erhöhte Flexibilität und die verbesserte Parametrisierbarkeit. Schließlich kann der Künstler nun die einzelnen Effekte unabhängig voneinander bearbeiten und so zum Beispiel zwischen verschiedenen spekularen Modellen wählen.

Damit sind alle Einzelheiten, die es beim Rendering von Pflanzenblättern zu beachten gilt, hinreichend geklärt und der eigentliche Darstellungsalgorithmus kann formuliert werden. Nachfolgend ist (siehe nächste Seite) der gesamte Algorithmus - in Form von Pseudocode - dargestellt. Die Funktion „AusgehendeRadianz“ berechnet die von der Oberfläche reflektierte Radianz. Dabei werden alle bisher vorgestellten visuellen und optischen Effekte berücksichtigt. Intuitiv kann der vorgestellte Algorithmus folgendermaßen verstanden werden: Zunächst wird das Ergebnis initialisiert. Die nachfolgende Berechnung der Oberflächenvariation bestimmt die konkreten Werte für die Dicke, die Absorption und die Normale. Anhand der Blickrichtung kann nun entschieden werden, ob das Pflanzenblatt von oben oder von unten betrachtet wird. Durch die Verwendung der Treppenfunktion,  $F$ , erfolgt die konkrete Wahl der aktuellen Textur und der Reihenfolge der Schichten. Daraufhin kann die reflektierte Radianz berechnet werden, indem der Einfluss jeder Lichtquelle in der Szene berücksichtigt wird. Bevor die eigentliche Berechnung erfolgen kann, muss jedoch entschieden werden, ob eine Frontal- oder eine Durchlichtsituation vorliegt. Im ersteren Fall inkrementiert der spekulär reflektierte Lichtanteil das Ergebnis. Globale Beleuchtungseffekte werden durch die Berechnung der ambienten Radianz unterstützt, wobei an dieser Stelle sowohl eine konstante Approximation, als auch eine vollständige Simulation der globalen Lichtinteraktion verwendet werden kann. Danach wird die direkte Beleuchtung der aktuellen Lichtquelle bestimmt. Dazu wird die eingehende Radianz, in den einzelnen Schichten, jeweils an einigen Testpunkten berechnet. Diese Stellen befinden sich auf dem Pfad entlang der Blickrichtung, wobei es an den Grenzen zwischen zwei Ebenen zu einer Refraktion der Richtung kommt. Liegt ein solcher Testpunkt nicht im Schatten eines anderen Objekts, so kann die korrigierte Länge des Lichtpfads bestimmt werden. Auf Basis der optischen Tiefe, gilt es nachfolgend die Lichtabschwächung zu berechnen. Daraus ergibt sich schließlich die reflektierte Radianz, die nachfolgend auf das Ergebnis addiert wird.

### 3 Realistische Pflanzendarstellung

---

```
AusgehendeRadianz( Punkt x, Normale n, Blickrichtung v )

Ergebnis = Schwarz

BerechneOberflächenVariationen( x, n )

Falls F( v, n ) < 0
    Blatttextur = Vorderseite
    Gewebeschichten = { 1, 2, 3 }
Sonst
    Blatttextur = Rückseite
    Gewebeschichten = { 3, 2, 1 }

Für jede Lichtquelle Li in der Szene

    omega = x - Li.Position / |x - Li.Position|
    Falls F( omega, n ) * F( v, n ) < 0
        Basisfarbe = Durchlichttextur( x )
    Sonst
        Basisfarbe = Blatttextur( x )
        Ergebnis += SpekulareBeleuchtung( omega, v, n )

Ergebnis += Basisfarbe * AmbienteRadianz( Li, Ti )

t = 0
v' = v
Für jede Schicht Si in Gewebeschichten

    v' = BerechneRefraktion( Si, v', n )
    Wichtungsfaktor = 1 / Anzahl der Testpunkte
    Für jeden Testpunkt Ti entlang v'

        t += dv'
        Falls EingehendeRadianz( Li, Ti ) > 0

            Beleuchtung = EingehendeRadianz( Li, Ti )
            t' = BerechneLichtpfadlänge( Li, Ti )
            Beleuchtung *= BerechneAbsorption( t, t', v, omega )

Ergebnis += Basisfarbe * Beleuchtung * Wichtungsfaktor
```

### 3.2.2 Erstellung der Texturen

Der im letzten Abschnitt beschriebene Algorithmus, verfügt über die gewünschten Eigenschaften der Parametrisierbarkeit, Flexibilität, Effizienz und funktioniert dabei auch auf der Basis einer starken geometrischen Approximation. Diese vorteilhaften Eigenschaften werden vor allem durch die intelligente Nutzung von Texturen anstelle der sonst sehr aufwendigen Simulationen erreicht. Dafür müssen passende Datensätze erzeugt werden. Der Vorgang der Texturerstellung erfolgt in zwei Schritten. Zunächst gilt es die Radianztexturen manuell zu akquirieren, daraus werden im zweiten Schritt automatisch weitere Texturen erstellt.

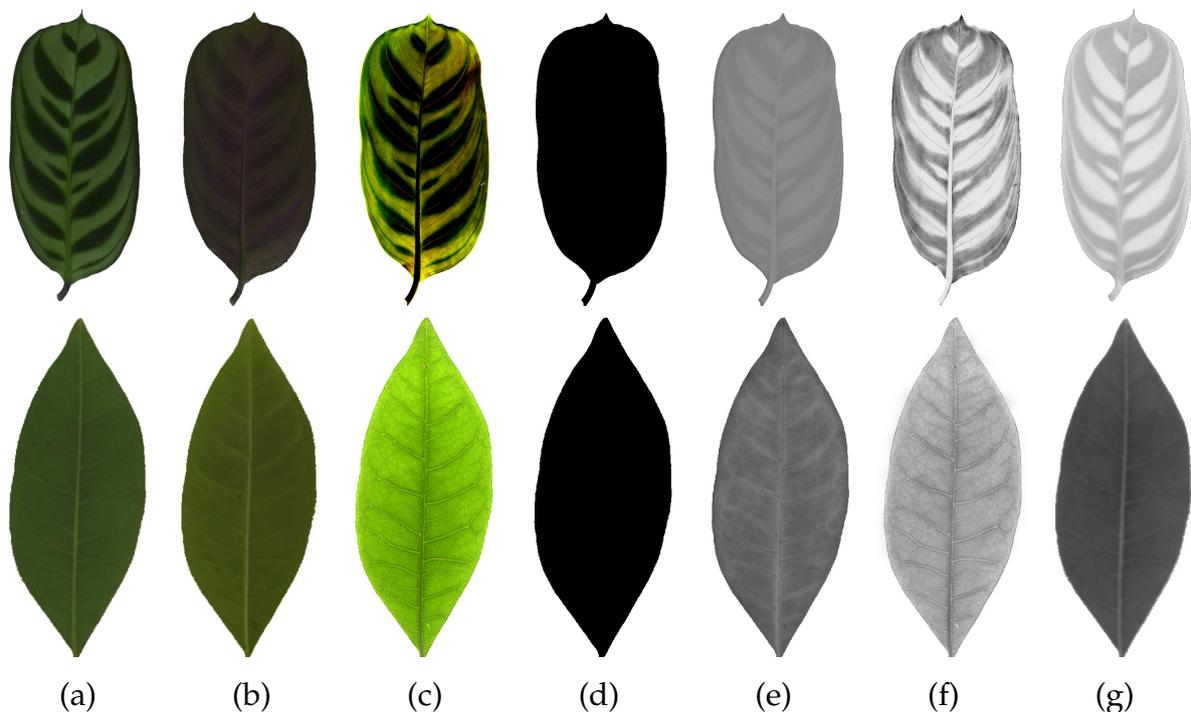


Abbildung 3.8: Beispielt Texturen zweier Pflanzenblätter

Wie bereits oben erwähnt wurde, beschreiben die Radianztexturen den diffusen Anteil der BSSRDF eines Blattes. Sie sind auf der linken Seite von Abbildung 3.8 zu sehen. Die Vorderseiten- (a) und Rückseitentextur (b) decken die zwei Fälle der Frontallichtsituation ab, während das Transmissionsbild (c) die Durchlichtsituation repräsentiert. Diese drei Texturen bilden die Basis für alle weiteren Datensätze.

Zunächst soll kurz die Akquirierung der Vorderseiten- und Rückseitentextur erläutert werden. Rein theoretisch können diese Bilder einen beliebigen Inhalt besitzen, häufig soll jedoch eine reale Pflanze nachgebildet werden. In diesem Fall können Fotografien - der zwei Seiten eines echten Pflanzenblattes - verwendet werden. Steht jedoch keine professionelle Aufnahmetechnik zur Verfügung, so kann es passieren, dass in dem Bild nicht nur der diffuse, sondern auch der spekulare Anteil zu sehen ist. Um dies zu verhindern, kann das Pflanzenblatt in einer mit Wasser gefüllten Schale fixiert werden. Da die Wasseroberfläche höher als das Blatt ist, wird das spekulare Glanzlicht in der Ergebnistextur verschoben. Die Wassermenge kann nun so lange angepasst werden, bis fast keine spekularen Effekte mehr zu sehen sind. Es ist allerdings wichtig, dass die Photokamera orthogonal zur Wasseroberfläche ausgerichtet wird, weil es sonst zu ungewollten Refraktionen kommen könnte, wodurch die Texturen eine Verzerrung erleiden würden. Die Durchlichttextur kann nur sehr schwer mit einer Standard-Photokamera erstellt werden, weil das Blatt direkt vor einer starken diffusen Lichtquelle fixiert sein müsste. Dadurch würde das Ergebnisbild jedoch überstrahlt werden. Von vielen Scannerherstellern wird deswegen ein Durchlichtaufsatz angeboten, welcher über eine stärkere Lampe verfügt. Mit einer solchen Erweiterung lässt sich das Problem optimal lösen. Bei der Akquirierung der Radianztexturen ist es sehr wichtig, dass ein frisches Pflanzenblatt verwendet wird und das zwischen den einzelnen Aufnahmen nicht zu viel Zeit vergeht, sonst trocknet das Blatt aus, wodurch jedoch das typische Reflektions- und Transmissionsverhalten zerstört wird. Die Texturen in Abbildung 3.8 wurden mit den hier beschriebenen Methoden erstellt.

Damit der Darstellungsalgorithmus auch auf Basis der geometrischen Approximationen (siehe Abb. 3.1) funktioniert, ist eine weitere Textur nötig, mit deren Hilfe an jedem Punkt auf der Oberfläche bestimmt werden kann, ob diese Stelle zum Pflanzenblatt gehört oder nicht. Diese Art von Textur wird auch (binäre) Maske (d) genannt.

Die konkrete Dicke des Blattes, an einem Punkt  $x$  auf der Oberfläche, wird mit Hilfe einer Variationstextur (e) bestimmt (siehe Gleichung 3.7). Sie kann folgendermaßen aus der Durchlichttextur extrahiert werden: Zunächst gilt es die Luminanz des Transmissionsbilds zu berechnen. Anschließend wird diese neue Textur Pixelweise invertiert. Danach kann noch ein Hochpassfilter auf das Bild angewendet werden, um zu starke Variationen zu vermeiden. Der Absorptionskoeffizient wird ebenfalls durch ei-

ne Modulationstextur (f) bestimmt. Die Extraktion aus dem Durchlichtbild, erfolgt ebenfalls durch die Berechnung der Luminanz der Quelltextur.

Wie bereits weiter oben erklärt wurde, führt die Variation der Dicke zusätzlich zu einer Veränderung der Oberflächennormale. Um eine ideale Parametrisierung des Darstellungsmodells zu gewährleisten, sollte diese Verwerfung der Normale aber nicht direkt aus der Variationstextur der Dicke bestimmt werden. Häufig möchte ein Künstler nämlich explizit bestimmten, welche Oberflächendetails zu sehen sind und welche nicht. Deswegen macht es Sinn für diesen Zweck eine zusätzliche Textur (g) einzuführen. Diese so genannte „Bumpmap“ könnte zum Beispiel auch aus der Vorderseitentextur (a) erstellt werden.

## 3.3 Bewertung der Ergebnisse

Die Bewertung der Ergebnisse des vorgeschlagenen Algorithmus soll auf der Basis der geforderten Bedingungen (Parametrisierbarkeit, Effizienz, Geometrische Approximation, Flexibilität) erfolgen. Die Vor- und Nachteile werden dabei sowohl anhand von Bildern, als auch durch Vergleiche mit anderen Verfahren erläutert.

Bei der Parametrisierung wurde ein Mittelweg zwischen einem sehr allgemeinen und sehr speziellen Verfahren gewählt. In der Regel müssen von einem Künstler nur sehr wenige Parameter angepasst werden, wie zum Beispiel die maximale Dicke und die obere bzw. untere Grenze des Absorptionskoeffizienten (Gleichung 3.7). Für die relative Dicke der einzelnen Schichten und die Brechungsindices können - im Allgemeinen - Standardwerte verwendet werden. Die selbe Aussage gilt auch für das spekulare Verhalten der Oberfläche. Wenn ein Künstler aber einen sehr speziellen visuellen Eindruck erzeugen möchte, so könnte er auch diese erweiterten Parameter anpassen. Der Vergleich mit einem Standardmodell (Abb. 3.10 unten) der Computergrafik zeigt, dass dieses zwar die Frontallichtsituation relativ gut approximiert, jedoch bei der Durchlichtsituation komplett versagt. Auf der anderen Seite ist der vorgeschlagene Algorithmus (Abb. 3.10 oben) einfacher zu steuern, als ein rein biologisches Modell. So lässt sich das Verfahren von Baranoski und Rokne ([BR97], [BR01]) nur mit Hilfe von relativ unintuitiven Parametern kontrollieren. Zum Beispiel muss die Form der Epidermiszellen („Oblateness“) und die Konzentration der Pigmente in der Me-

sophyllschicht angegeben werden. Natürlich folgt daraus auch ein Nachteil des in dieser Diplomarbeit vorgeschlagenen Verfahrens, so lässt es sich nur äußerst eingeschränkt bei der Untersuchung von biologischen und physikalischen Phänomenen verwenden. Andererseits war dies auch nicht das angestrebte Anwendungsgebiet, welches eher im Kontext der Filmproduktion einzuordnen ist.



Abbildung 3.9: Einzelnes Blatt - Neuer Algorithmus (600x800; 11,5 Sekunden)

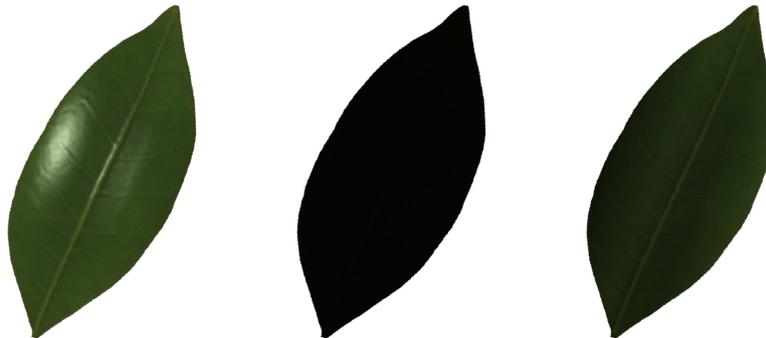


Abbildung 3.10: Einzelnes Blatt - Standardalgorithmus (600x800; 11 Sekunden)

Der wohl wichtigste Vorteil des neuen Algorithmus stellt die hohe Effizienz dar. Durch die Verwendung von Texturen anstelle der komplizierten biologischen und physikalischen Modelle konnte sehr viel Rechenzeit gespart werden. Die Darstellung mit einem Standardmodell erfolgt zwar schneller, ist jedoch nicht in der Lage alle visuellen Effekte nachzubilden, wodurch der Vergleich hinfällig wird. Im Gegensatz

dazu modellieren biologisch motivierte Darstellungsverfahren zwar viele der vorgestellten Phänomene, benötigen dafür aber häufig sehr viel Rechenzeit. In der Praxis (z.B. Filmproduktion) ist eine sehr genaue Simulation aller optischen Eigenschaften häufig gar nicht wichtig, denn die dargestellten Objekte sollen in erster Linie lediglich realistisch wirken. Beispielsweise ist vollständige Beschreibung des spektralen Verhaltens in diesem Anwendungsfall kaum von Bedeutung. Viele biologisch motivierte Ansätze versuchen jedoch genau solche Eigenschaften zu simulieren. So benötigte der Algorithmus von Baranoski und Rokne 95 Minuten Rechenzeit auf einer SGI R 10000, um ein Bild mit zwei Pflanzenblättern zu berechnen. Im Vergleich dazu ist der Aufwand des neuen Verfahrens sehr gering. Die Berechnungsdauer der einzelnen Bilder wurde jeweils direkt unter den Abbildungen vermerkt. Die Produktion der Ergebnisse fand auf einem AMD Athlon mit 1GHz Taktrate und 512MB RAM statt.



Abbildung 3.11: Zimmerpflanze (800x800; 100 Sekunden)

Häufig liegt die darzustellende Pflanze in einer sehr groben geometrischen Approximation vor. So werden die Blätter häufig lediglich als endliche Ebenen modelliert (siehe Abb. 3.1). Der vorgeschlagene Algorithmus funktioniert auch auf der Basis eines solchen Modells, wobei sich in diesem Fall die Zellen eines Pflanzenblattes innerhalb eines virtuellen Volumens befinden, welches durch eine weitere (virtuelle) Begrenzungsebene beschrieben wird. Natürlich ist es auch möglich, eine genauere geometrische Repräsentation zu verwenden, dann fällt die Definition der Gewebedicke und damit die neue Begrenzungsebene weg. Andere Algorithmen zur Darstellung von transluzenten Objekten ([JMLH01], [JB02]), benötigen immer ein echtes Volumen.



Abbildung 3.12: Zimmerpflanze (800x800; 150 Sekunden)

Die große Flexibilität, die aus der Verwendung von Texturen folgt, ist ein weiterer wichtiger Vorteil gegenüber anderen Verfahren. Da ein Künstler die Einzeltexturen frei editieren kann, verfügt er über die Möglichkeit das Aussehen der dargestellten Objekte stark zu verändern. So könnte beispielsweise die Farbe der Durchlichttextur bearbeitet werden, wodurch die gerenderten Pflanzen sicherlich ein sehr individuelles Aussehen besitzen würden. Ein weiteres Beispiel stellt die Anpassung der Absorptionstextur dar, wodurch Buchstaben in das Gewebe eingepreßt werden könnten. Biologisch motivierte Ansätze bieten diese Art von Flexibilität häufig nicht. In der Filmproduktion ist eine große künstlerische Freiheit allerdings stets gewünscht.



Abbildung 3.13: Zimmerpflanze (800x800; 124 Sekunden)

Der vorgeschlagene Algorithmus lässt sich leicht in professionelle Renderer integrieren. Neben Programmen wie mental ray ([men]) und Brazil ([bra]), die Raytracing

vollständig unterstützen, kann das Verfahren, nach einigen kleineren Änderungen, auch in Renderman ([ren]) umgesetzt werden.



Abbildung 3.14: Zimmerpflanze (Busch) (800x800; 10 Minuten)

## 3.4 Zusammenfassung und Ausblick

In diesem Kapitel wurde ein neuer Algorithmus ([FD03b], [FD03a]) vorgestellt, der es ermöglicht Pflanzengewebe photorealistisch darzustellen. Auf Basis einer Analyse, der biologischen und optischen Eigenschaften von echten Blättern, konnte ein vereinfachtes Modell formuliert werden. Dieses Verfahren besitzt einige vorteilhafte Eigenschaften, wie eine einfache Parametrisierung, hohe Effizienz und große Flexibilität. Dabei ist es nicht unbedingt notwendig, dass ein exaktes geometrisches Modell der Pflanzen vorliegt, da der Algorithmus auch auf Basis einer stark vereinfachten geometrischen Repräsentation arbeiten kann. Diese Vorteile werden hauptsächlich durch den geschickten Einsatz von Texturen anstelle der komplizierten physikalischen Simulationen erreicht.

Der neue Algorithmus kann - rein theoretisch - in beliebig komplexen Szenen verwendet werden. Allerdings kommt es bei steigender Komplexität immer mehr zu Hardware-basierten Einschränkungen, wie ein zu geringer Arbeitsspeicher. Um trotzdem einen ganzen Wald realistisch darzustellen, gilt es weitere Vereinfachungen einzuführen. Ein möglicher Lösungsansatz, wird im nächsten Kapitel erläutert.

### 3 Realistische Pflanzendarstellung

---

Trotz der hohen Effizienz des Verfahrens, ist es jedoch - zum aktuellen Zeitpunkt - nicht möglich den Algorithmus in Echtzeit auszuführen. Mit gewissen Einschränkungen, lässt sich auch diese Hürde bewältigen. Das nächsten Kapitel behandelt insbesondere die realistische Darstellung von Pflanzen in einem Echtzeitkontext.



Abbildung 3.15: Der Hof (30 Minuten)

*The world is changed;  
I can feel it in the water, I can feel it in the earth, I can smell it in the air.*

Galadriel, Lord of the Rings von J.R.R. Tolkien

# 4 Echtzeitvisualisierung von Pflanzen

## 4.1 Einführung

Im Rahmen einer Filmproduktion werden ausschließlich Offline Darstellungsverfahren benutzt, um die nötigen Bildsequenzen (Frames) zu erstellen. Dabei gilt eine hohe Qualität der Ergebnisse als das vorrangige Ziel, während die Berechnungsdauer der Bilder eine untergeordnete Rolle spielt. Mitunter können mehrere Minuten und manchmal sogar Stunden Rechenzeit pro Frame in Kauf genommen werden. Diese Situation ändert sich jedoch schlagartig, wenn es um die Echtzeitvisualisierung von Pflanzen geht, denn in diesem Anwendungsgebiet müssen sehr strikte Zeitbeschränkungen bei der Berechnung der Einzelbilder eingehalten werden. Um nämlich die Illusion einer flüssigen Bewegung zu erzeugen, sollten mindestens 15 Bilder pro Sekunde erstellt werden. Diese hohen Bildberechnungsraten lassen sich nur mit Hilfe von spezielle Hardwarekomponenten realisieren, welche die Darstellung beschleunigen sollen. Dafür müssen jedoch einige Einschränkungen in Kauf genommen werden, so dass es auch mit Hilfe der aktuellen Grafikkarten nicht möglich ist, den oben vorgestellten Algorithmus in Echtzeit auszuführen. Dieses Ziel lässt sich nur erreichen, wenn weitere Vereinfachungen gefunden und ausgenutzt werden. Die Qualität der Bilder, welche sich mit einem solchen vereinfachten Algorithmus berechnen lassen, ist zwar in der Regel nicht so hoch wie die der Ergebnisse des Offlinedarstellungsverfahrens, doch dieser Nachteil lässt sich auf Grund der erreichten Interaktivität häufig akzeptieren.

Dieses Kapitel fasst deswegen das Problem der Echtzeitvisualisierung auf und beschreibt die Entwicklung von neuen Verfahren, um eine realistische und interaktive Darstellung von Pflanzen zu gewährleisten. Zunächst werden die zu erreichenden Ziele festgelegt. Danach erfolgt eine Einführung in die Architektur und Arbeitswei-

se der aktuellen Grafikhardware, wobei diese Betrachtung die Grundlage für alle nachfolgenden Abschnitte bildet. Unter Ausnutzung der Fähigkeiten der Beschleunigungshardware, wird anschließend die direkte Adaption des Offlinedarstellungsverfahrens untersucht. Danach erfolgt eine Analyse der Möglichkeit, die Lösung der Rendering-Gleichung in einem anderen Funktionenraum durchzuführen. Doch selbst mit weiteren Vereinfachungen lassen sich damit keine komplexen Szenen wie Baumgruppen oder Wälder darstellen, denn in einem solchen Fall stellt die extrem große geometrische Komplexität der Landschaft das Hauptproblem dar. Dieses Problem lässt sich nur durch die Verwendung von sehr stark vereinfachten Pflanzenmodellen beheben. Um in diesem Fall trotzdem eine möglichst realistische Darstellung der Pflanzen zu garantieren, wird anschließend ein neues Visualisierungsverfahren entwickelt, welches auch auf Basis der geometrisch extrem vereinfachten Modelle funktioniert. Mit Hilfe dieser neuen Vorgehensweise lassen sich dann auch sehr komplexe Landschaften in Echtzeit darstellen. Nachdem die erreichten Ergebnisse bewertet wurden, schließt das Kapitel mit einer Zusammenfassung und einem weiterführenden Ausblick ab.

### 4.1.1 Zielstellung

Die realistische Darstellung von natürlichen Szenen stellt insbesondere dann ein Problem dar, wenn diese in einem Echtzeitkontext erfolgen soll. Dabei unterliegt sowohl die geometrische Repräsentation der Szene als auch die algorithmische Komplexität des Darstellungsverfahrens gewissen Einschränkungen, die vor allem aus der Verwendung von Beschleunigungshardware folgen. Trotz dieser begrenzenden Eigenschaften sollten sich auch komplexe Landschaften darstellen lassen. Deswegen besteht das Ziel dieses Kapitels darin ein Darstellungsverfahren zu entwickeln, welches die photorealistische Visualisierung von Pflanzen in Echtzeit ermöglicht. Ausgehend von dem in Kapitel 3 vorgeschlagenen Offline-Algorithmus sollten weitere Vereinfachungen gefunden und eingesetzt werden. Dabei gilt es insbesondere die Fähigkeiten der aktuellen Grafikhardware auszunutzen, um eine erhöhte Performanz zu garantieren. Eine äußerst effizientes Darstellungsverfahren ist gerade bei der Visualisierung von umfangreichen Szenen von großer Bedeutung, wobei in einem solchen Fall häufig sehr stark vereinfachte Pflanzenmodelle eingesetzt werden, um die geometrische

Komplexität der Landschaft so weit wie möglich zu reduzieren. Deswegen gilt die realistische Darstellung dieser vereinfachten Objektrepräsentationen als ein weiteres Ziel dieses Kapitels. Auf Basis der erzielten Ergebnisse, sollte anschließend eine umfangreiche Untersuchung des Problemfelds der Visualisierung von komplexen Landschaften erfolgen.

### 4.2 Hardwareunterstützte Darstellung

Im Grundlagenkapitel wurden bereits einige Einschränkungen genannt, die es bei der Echtzeitvisualisierung von Pflanzen zu beachten gilt (vgl. Abschnitt 2.2.2.2). So ist zum Beispiel - neben dem stark begrenzten Speicher der Grafikkarten - kein wahlfreier Zugriff auf andere Elemente der Szene möglich, weil die Objekte nacheinander einzeln verarbeitet werden. Da diese Begrenzungen aus der Vorgehensweise der Beschleunigungshardware folgen, wird im nachfolgenden Abschnitt zunächst der Aufbau und die Wirkungsweise der aktuellen Grafikkarten erläutert. Auf Basis dieser Informationen kann anschließend der empirische Algorithmus, welcher im vorherigen Kapitel eingeführt wurde, für die Hardwareunterstützte Darstellung angepasst werden. Im Prinzip lassen sich damit zwar alle Einzelpflanzen in Echtzeit rendern, allerdings reicht die Performanz häufig nicht aus, um komplexe Landschaften darzustellen. Wie bereits erwähnt wurde, werden in einem solchen Fall geometrisch extrem vereinfachte Modelle verwendet, um die Komplexität der Szene zu reduzieren. Durch diese geometrischen Vereinfachungen sind jedoch während des Renderings keine korrekten Informationen mehr über die darzustellende Oberfläche vorhanden. Da der empirischen Algorithmus aber diese Daten benötigt, um richtige Ergebnisse zu liefern, kann er also in diesem Fall nicht eingesetzt werden. Deswegen wird anschließend ein weiteres Darstellungsverfahren untersucht, welches auch ohne Informationen über die Oberfläche korrekt funktioniert und sich somit auch bei der Visualisierung der stark vereinfachten Pflanzenmodelle einsetzen lässt.

### 4.2.1 Architektur aktueller Grafikhardware

Neben den vielen algorithmischen Innovationen der letzten Jahre im Bereich des schnellen photorealistischen Renderings ([Jen01], [SKS02], [JMLH01]), hat sich vor allem auch die Grafikkartentechnologie ([nvi], [ati]) weiterentwickelt. Der rasante Anstieg der Performanz macht es mittlerweile möglich, dass sich auch sehr detaillierte Objekte in Echtzeit verarbeiten und darstellen lassen. Doch diese Leistungssteigerung stellt nur einen Teil des Fortschritts dar, denn gerade in der letzter Zeit wurde die Flexibilität der Hardware ebenfalls stark erhöht. Vor ein paar Jahren war nämlich die Art und Weise der Visualisierung von dreidimensionalen Objekten noch fest vorgegeben, weil die Verarbeitung der Szenendaten durch spezielle Hardwarekomponenten durchgeführt wurde. Dabei ließ sich das erstellte Ergebnis bestenfalls durch einen vorgegebenen Parametersatz kontrollieren. Dieses Vorgehen führte jedoch dazu, dass sich die dargestellten Objekte häufig sehr ähnlich sahen. Um die Individualität der visualisierten Szenen zu erhöhen, wird jedoch eine bessere Kontrolle über die einzelnen Verarbeitungsschritte benötigt. Deswegen ist es mittlerweile möglich, das Verhalten der aktuellen Grafikkarten durch die Verwendung von speziellen Programmen (Shader) zu steuern, die dann auf der Hardware ausgeführt werden.

Die Grundprinzipien - auf dessen Basis die Hardwareunterstützte Darstellung aufbaut - haben sich währenddessen allerdings nicht geändert. So ist das Dreieck immer noch das geometrische Basiselement, welches bei der Darstellung verwendet wird. Es wird durch drei Eckpunkte (Abb. 4.1 rechts) beschrieben, die auch Vertices (Singular: Vertex) genannt werden. Ein solcher Vertex besitzt mindestens eine Position im Raum, es können allerdings auch noch weitere Parameter mit einem Eckpunkt assoziiert werden. Beispielsweise lässt sich an jedem Vertex explizit eine (individuelle) Normale angeben, um die Illusion von beliebig gekrümmten Objekten zu erzeugen. Dabei erfolgt die Bestimmung einer beliebigen Oberflächennormale innerhalb des Dreiecks, durch eine bilineare Interpolation der Normalen der Eckpunkte. Eine solche Interpolation wird auch bei der Beschreibung der Grundfarbe eines Dreiecks benutzt, wobei sich an jedem Vertex eine andere Farbe (Vertexcolor) festlegen lässt. Außerdem können für jeden Eckpunkt bis zu acht Texturkoordinaten angegeben werden, um eine umfangreiche Verwendung von (bis zu acht verschiedenen) Texturen zu ermöglichen. Dabei legen diese Vertexparameter die Position des Dreiecks in den

jeweiligen Texturräumen fest. Die Texturkoordinaten werden - genau wie die Position des Eckpunktes im Raum und die Farbe des Vertex - durch vier-elementige Vektoren aus reellen Zahlen beschrieben, während für die Normale lediglich ein dreielementiger Vektor benötigt wird. Da jedes Dreieck einen eigenen individuellen Satz von Vertexparametern besitzt, werden diese Parameter als lokale Daten eines Dreiecks bezeichnet (siehe Abb. 4.1).

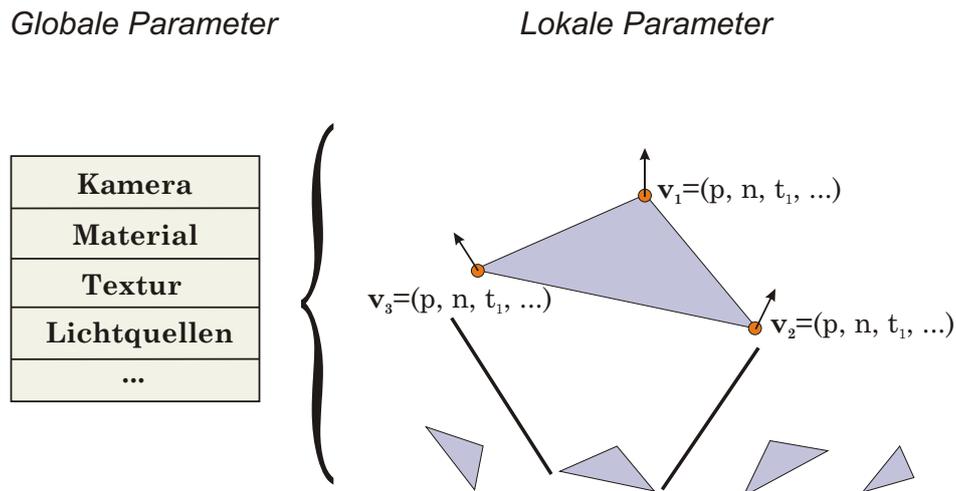


Abbildung 4.1: Lokaler und globaler Parameterraum der Grafikhardware

Für die Darstellung eines Objekts sind allerdings noch weitere Daten nötig, die jedoch für alle (aktuellen) Dreiecke in der selben Art und Weise gelten. So lassen sich in diese Kategorie von Attributen zum Beispiel die Materialparameter oder auch die Beschreibung der Lichtquellen einordnen. Da diese Daten für mehrere Dreiecke gleichzeitig definiert sind, werden sie globale Parameter (Abb. 4.1 links) genannt, wobei sich für die Beschreibung dieser Dreiecks-übergreifenden Attribute unterschiedliche Datentypen verwenden lassen. So können dafür zum Beispiel einfache reelle Zahlen, Vektoren oder sogar Matrizen benutzt werden. Aufgrund des begrenzten Speichers der Grafikkarten lassen sich allerdings nicht beliebig viele lokale und globale Parameter definieren. Die konkrete Anzahl der unterstützten Werte hängt dabei von der benutzten Hardwaregeneration ab.

Die Grafikkarte lässt sich als ein (endlicher) Zustandsautomat ansehen, denn sind die globalen Parameter erst einmal festgelegt, so gilt der aktuelle Zustand für alle Drei-

ecke in der selben Art und Weise. Da aber zum Beispiel nur ein (globaler) Parametersatz für die Materialinformationen zur Verfügung steht, kann zu jeden Zeitpunkt nur eine einzige Oberflächenart dargestellt werden. Deswegen müssen alle geometrischen Modelle in einzelne Teile zerlegt werden, die jeweils ein und das selbe Material besitzen. Diese Einzelteile werden dann nacheinander von der Grafikkarte verarbeitet. Eine solche Dreiecksliste darf nur eine bestimmte Maximalanzahl von Elementen enthalten, weil sonst der Speicher der Hardwarekomponente nicht ausreichen würde. Im Allgemeinen ist also während eines Verarbeitungsschrittes nicht die gesamte Szene im Arbeitsspeicher der Grafikkarte, sondern lediglich ein einzelnes Bestandteil eines bestimmten Objekts. Aus diesem Grund kann bei der Darstellung eines Dreiecks nur auf die aktuellen lokalen Daten (und die globalen Attribute) zugegriffen werden, aber niemals auf die lokalen Parameter anderer Dreiecke. Diese Art von Visualisierung wird deswegen auch als lokales Darstellungsmodell bezeichnet. Somit lässt sich die realistische Darstellung von Pflanzen in diesem Kontext jedoch nicht mit Hilfe einer Raytracinglösung - wie sie im Kapitel 3 vorgestellt wurde - realisieren. Im nächsten Abschnitt werden deshalb weitere Verfahren vorgestellt, welche auf Basis eines lokalen Darstellungsmodells arbeiten.

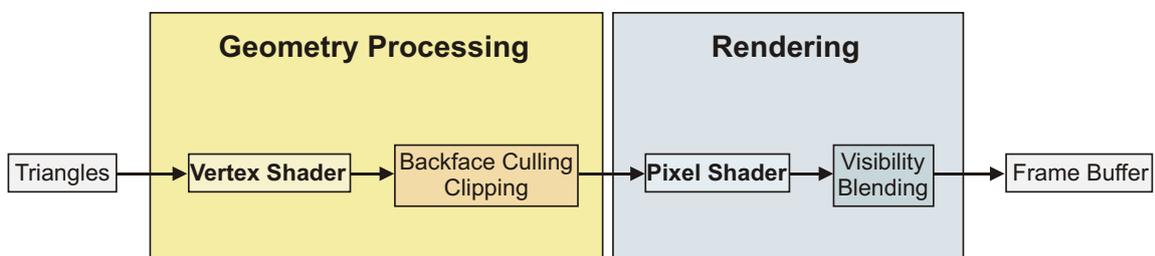


Abbildung 4.2: Vereinfachte Verarbeitungspipeline der Grafikkarte

Zuvor sollte allerdings noch ein genauerer Blick auf die Verarbeitungspipeline der Grafikkarte geworfen werden. Die Transformation der Dreiecke in ein Bild erfolgt in zwei Schritten: geometrische Verarbeitung (Abb. 4.2 linke Box) und Rendering (Abb. 4.2 rechte Box). Der erste Schritt erhält die (lokalen) Vertexdaten der darzustellenden Dreiecke (Triangles) und führt darauf eine Reihe von geometrischen Operationen aus. So werden zunächst alle Eckpunkte in das Koordinatensystem der Bildebene projiziert, woraufhin alle - von der virtuellen Kamera aus - nicht sichtbaren Dreiecke

entfernt (Backface Culling, Clipping) werden. Im zweiten Verarbeitungsschritt erfolgt anschließend das Rendering der restlichen Dreiecke. Hierbei wird für jeden Pixel des Ergebnisbildes (Frame Buffer) ein Farbwert berechnet, wobei die Sichtbarkeit (Visibility) des aktuellen Bildelements berücksichtigt wird.

Um eine hohe Flexibilität bei Transformation der Dreiecke in ein Bild zu gewährleisten, kann das Verhalten der zwei Pipelineschritte durch die Verwendung von kleinen Programmen (Shadern) sehr genau gesteuert werden. Diese Shader werden auf der Grafikkarte ausgeführt und ermöglichen somit eine sehr individuelle (und schnelle) Verarbeitung der Daten. Dabei kann für eine zu verarbeitende Dreiecksliste jeweils ein Programm für die geometrische Verarbeitung (Vertex Shader, Vertex Program) und ein Shader für das Rendering (Pixel Shader, Fragment Program) definiert werden. Die Art und Weise wie ein solcher Shader programmiert wird, hängt stark von dem verwendeten Anwendungskontext (DirectX: HLSL [dx], OpenGL: GLSL [og], CG [cg] ...) ab. Unabhängig von der benutzten Programmierschnittstelle gilt jedoch, dass es - trotz der individuellen Verarbeitungsmöglichkeit der Dreiecke - auch mit Hilfe der Shader-Technologie nicht möglich ist, frei auf die Szene zuzugreifen. Während der Ausführung der Programme stehen nämlich weiterhin nur die globalen und lokalen Parameter zur Verfügung. Allerdings wird von der Grafikkarte die konkrete Semantik dieser Attribute nicht vorgeschrieben, so dass ein Shader die übergebenen Werte sehr frei interpretieren kann. Diese Eigenschaft wird später noch wichtig werden, wenn es um die Optimierung der Performanz der Pflanzendarstellung gehen wird.

### 4.2.2 Adaption des empirischen Algorithmus

Die Programmierbarkeit der aktuellen Grafikkarten macht es somit grundsätzlich möglich, den Darstellungsalgorithmus für Pflanzen auch in einem Echtzeitkontext zu verwenden. Aufgrund der oben genannten Einschränkungen, sind dafür jedoch einige Änderungen an der ursprünglichen Formulierung (vgl. Abschnitt 3.2.1) nötig. Dabei stellt der stark begrenzte Zugriff auf die Daten der Szene das größte Hindernis dar, weil sich dadurch die eingehende Radianz nicht mehr exakt bestimmen lässt. Denn für die Entscheidung ob ein Punkt im Schatten eines anderen Objekts liegt oder nicht, wurde zuvor ein Strahl von der Lichtquelle zum betrachteten Punkt gelegt und

auf Schnitt mit anderen Szenenelementen getestet. Konnte ein solcher Schnittpunkt gefunden werden, so lag die untersuchte Stelle im Schatten, andernfalls wurde sie mit voller Intensität beleuchtet. Die Verwendung von Raytracing bei die Schattenberechnung ist in diesem Fall jedoch nicht möglich, da der wahlfreie Zugriff auf die lokalen Daten anderer Dreiecke verboten ist. Es muss also eine alternative Vorgehensweise ausfindig gemacht werden, die auch unter den gegebenen Bedingungen funktioniert.

Tatsächlich wurde 1978 ein solcher Lösungsansatz vorgeschlagen ([Wil78]), welcher die Berechnung von Schatten trotz dieser starken Einschränkungen ermöglicht. Der von Lance Williams eingeführte Algorithmus wird „Shadow Mapping“ genannt und besteht aus zwei Stufen. Vor der Berechnung des eigentlichen Ergebnisbildes, wird für jede Lichtquelle - die einen Schatten werfen soll - ein weiteres Bild erstellt, wobei die Pixel dieser so genannten Shadow Map jedoch keine herkömmlichen Farbwerte enthalten, sondern den Abstand  $t(\vec{\omega})$  von der aktuellen Lichtquelle zur nächsten Oberfläche in der Szene entlang der Richtung  $\vec{\omega}$ . In der zweiten Stufe dieses Ansatzes wird schließlich das gewünscht Ergebnisbild berechnet, wobei sich die Entscheidung, ob ein Punkt  $x$  beschattet ist oder nicht, nun relativ leicht treffen lässt. Dabei sei  $l$  die Position der betrachteten Lichtquelle und  $x$  die zu untersuchende Stelle auf einer beliebigen Oberfläche, dann kann die folgende Funktion definiert werden:

$$\begin{aligned}
 t_{akt} &= |x - l| \\
 \vec{\omega} &= \frac{x - l}{|x - l|} \\
 S(\vec{\omega}, t_{akt}) &= \begin{cases} 0, & t_{akt} > t(\vec{\omega}) \\ 1, & t_{akt} \leq t(\vec{\omega}) \end{cases} \quad (4.1)
 \end{aligned}$$

Liefert die Schattenfunktion  $S$  den Wert Null, so liegt  $x$  im Schatten eines anderen Objekts. Diese Entscheidung kann deswegen getroffen werden, weil der nächste Punkt - in der aktuellen Richtung  $\vec{\omega}$  - einen geringeren Abstand zur Lichtquelle besitzt, als die Stelle  $x$  (siehe Abb. 4.3: Punkt  $x_1$ ). Ist der Wert der Shadow Map  $t(\vec{\omega})$  jedoch mit dem Abstand  $t_{akt}$  identisch<sup>1</sup>, so kann sich kein anderes Dreieck zwischen der Oberfläche und der Lichtquelle befinden, wodurch  $x$  nicht beschattet sein kann (siehe Abb. 4.3: Punkt  $x_2$ )!

---

<sup>1</sup>Wäre die Shadow Map unendlich genau aufgelöst, so könnte der Test, in Gleichung 4.1, ausschließlich auf Gleichheit erfolgen. Die endliche Natur dieses Bildes führt jedoch zu Ungenauigkeiten, wodurch ein „Kleiner oder Gleich“ Test nötig wird.

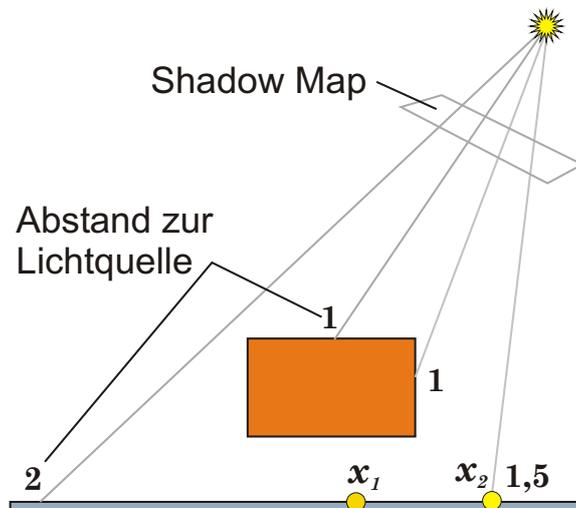


Abbildung 4.3: Shadow Map Algorithmus

Das Shadow Mapping Verfahren besitzt allerdings auch einige Nachteile, denn die Berechnung der Tiefentextur - für jede schattenwerfende Lichtquelle - kostet relativ viel Rechenzeit. Je höher dieses Bild aufgelöst ist, desto exakter lassen sich zwar die Schatten berechnen, desto mehr Aufwand muss allerdings auch in die Erstellung der Schattentextur investiert werden. Da in der Praxis eine Textur lediglich aus endlich vielen Pixeln besteht, kann es außerdem zu Beleuchtungsartefakten (Blockbildung, Shadow Leaking) im Ergebnisbild kommen, die von einer ungenauen Abtastung der Schattentextur herrühren. Trotz dieser Nachteile wird diese Vorgehensweise sehr häufig eingesetzt und auch bei der realistischen Darstellung von Pflanzen macht die Verwendung des Shadow Mapping Verfahrens Sinn.

Da sich nun im Rahmen der Hardwareunterstützten Darstellung auch Schatten berechnen lassen, kann der - in Kapitel 3 vorgeschlagene - Darstellungsalgorithmus für Pflanzen prinzipiell auch in einem Echtzeitkontext benutzt werden. Allerdings stellen die vielen verschiedenen Texturen und die Berechnung der eingehenden Radianz ein Performanzproblem dar. So gilt nämlich das Kopieren eines Bildes aus dem Hauptspeicher des Computers in den Arbeitsspeicher der Grafikkarte als eine sehr Zeitaufwendige Operation, was vor allem daran liegt, dass die Übertragungsrate des dafür zuständigen Datenbusses relativ gering ist. Doch dies ist nur ein Problem, denn weiterhin stellen die im Algorithmus vorhandenen Schleifenkonstrukte und die häu-

figen Zugriffe auf unterschiedliche Texturen ein großes Performanzproblem dar. Somit werden also weitere Änderungen an dem Darstellungsalgorithmus notwendig, um eine hohe Bildberechnungsrate zu garantieren. Dabei kann das Schichtenmodell zunächst einmal in der Form vereinfacht werden, dass die eingehende Radianz nur noch an einem Punkt auf der Oberfläche bestimmt wird. Dadurch fällt die aufwendigste Schleife (vgl. Abschnitt 3.2.1) einschließlich aller damit verbundenen Texturzugriffe weg, die anderenfalls für die Schattenbestimmung nötig wären. Natürlich können nun keine Penumbraeregion mehr dargestellt werden, wobei sich dieser Nachteil jedoch zugunsten einer höheren Performanz in Kauf nehmen lässt. Aus dieser Optimierung folgen weitere Vereinfachungen, denn es kann nun zum Beispiel ebenfalls auf die Modulationstexturen für die Dicke und die Absorption verzichtet werden, da die Radianz nur noch auf der Oberfläche bestimmt wird und somit auch kein virtuelles Volumen mehr nötig ist. Wird weiterhin davon ausgegangen, dass fast immer die Oberseite der Blätter zu sehen ist, dann lässt sich zudem auch die Rückseitentextur ignorieren. Im Gegensatz dazu darf jedoch das Maskierungsbild, welches die Silhouette der Blätter beschreibt, nicht verworfen werden, denn die korrekte Kontur der Pflanzenelemente ist auch in einem Echtzeitkontext sehr wichtig. Da Texturen bei der Hardwareunterstützten Darstellung über vier Farbkanäle verfügen, muss jedoch für die Beschreibung der Silhouette keine eigenes Bild benutzt werden, denn die Maske kann problemlos in den vierten Farbkanal ( $\alpha$ ) einer andere Textur transferiert werden. Durch diese Vereinfachungen lässt sich ein signifikanter Performanzgewinn erreichen, während die Qualität der Ergebnisbilder in einem akzeptierbaren Rahmen sinkt. Auf der nächsten Seite ist der - auf die Hardwareunterstützten Darstellung - angepasste Algorithmus zur Pflanzenvisualisierung in Form von Pseudocode zu sehen, wobei lediglich die erläuterten Vereinfachungen auf die ursprüngliche Formulierung des Darstellungsverfahrens (vgl. Abschnitt 3.2.1) angewendet wurden. Dieses Verfahren lässt sich sehr leicht in Form eines Vertex- und Pixel-Shaders umsetzen.

## 4 Echtzeitvisualisierung von Pflanzen

---

```
AusgehendeRadianz( Punkt x, Normale n, Blickrichtung v )

    Ergebnis = Schwarz

    BerechneOberflächenVariationen( x, n )

    Für jede Lichtquelle Li in der Szene

        t_akt = |x - Li.Position|
        omega = x - Li.Position / t_akt
        Falls F( omega, n ) * F( v, n ) < 0
            Basisfarbe = Durchlichttextur( x )
        Sonst
            Basisfarbe = Blatttextur( x )
            Ergebnis += SpekulareBeleuchtung( omega, v, n )

    Ergebnis += Basisfarbe * AmbienteRadianz( Li, x )

    EingehendeRadianz = SkalarProdukt( omega, n )
    EingehendeRadianz *= S( omega, t_akt )
    Ergebnis += Basisfarbe * EingehendeRadianz
```

Abbildung 4.4 zeigt einige Beispielbilder, die diesem vereinfachten Algorithmus berechnet wurden. Dabei konnten im Durchschnitt 100 Bilder pro Sekunde gerendert werden. Dieses Ergebnis ist zwar vielversprechend, aber die Performanz reicht immer noch nicht aus, um größere Szenen zu visualisieren, denn - wie oben bereits erwähnt wurde - stellt dabei vor allem die hohe Anzahl von zu verarbeitenden Dreiecken das Hauptproblem dar. Außerdem lässt sich dieses Verfahren auch nicht auf die extrem vereinfachten geometrischen Repräsentationen anwenden, die später benutzt werden, um komplexe Landschaften zu visualisieren. Dabei stehen nämlich während der Berechnung der Pixelfarbe keine korrekten räumlichen Informationen über die darzustellende Oberfläche zur Verfügung. Im nächsten Abschnitt wird deswegen ein weiteres Verfahren vorgestellt, welches für die realistische Darstellung von Pflanzen verwendet werden kann.



Abbildung 4.4: Angepasster Darstellungsalgorithmus (ca. 100 FpS)

### 4.2.3 Effizienzsteigerung durch Basisprojektion

Bei der Suche nach Effizienzoptimierungen, lassen sich verschiedene Wege einschlagen. Eine Möglichkeit bestünde zum Beispiel darin, den benutzten Algorithmus immer weiter zu vereinfachen. Das würde letztendlich aber zu einem Standardbeleuchtungsmodell führen, welches die pflanzenspezifischen Eigenschaften nur noch sehr schlecht repräsentiert. Eine weitere Vorgehensweise stellt die Lösung der Rendering-Gleichung in einem anderen Funktionenraum dar. Diese Herangehensweise wurde im Grundlagenkapitel bereits prinzipiell behandelt (vgl. Abschnitt 2.2.2.2) und soll nachfolgend ein wenig genauer untersucht werden. Tatsächlich stellt sich die - bereits weiter oben eingeführte - Spherical Harmonics Transformation als äußerst vorteilhaft heraus, wenn es um die realistische Echtzeitdarstellung von Pflanzen geht. So lässt sich nämlich das Integral über die Hemisphäre durch eine Menge von Basisoeffizienten darstellen, die in einer Vorberechnungsphase bestimmt werden können. Wenn die Reflektionsfunktion eines Materials ausschließlich eine diffuse Lichtinteraktion beschreibt, dann ist dafür lediglich ein Vektor mit einigen reellen Zahlen nötig.

Bei erneuter Betrachtung des ursprünglichen Darstellungsalgorithmus aus Kapitel 3 fällt sofort auf, dass gerade die diffusen Beleuchtungseffekte (z.B. Penumbraeffekte) eine Vielzahl von komplizierten Berechnungen nach sich ziehen. Wäre es also möglich diesen Bestandteil des Verfahrens in einer anderen Form darzustellen, dann ließe sich die ausgehende Radianz sehr einfach bestimmen und damit eine effiziente Visualisierung erreichen. Dabei lässt sich leicht einsehen, dass die diffuse Beleuchtung nur sehr langsam innerhalb eines Dreiecks variiert<sup>2</sup>. Diese Eigenschaft macht es möglich, die Rendering-Gleichung lediglich an den Eckpunkten eines Dreiecks zu lösen und bilinear über die Fläche zu interpolieren. Ältere Grafikkarten nutzten ausschließlich dieses so genannte Gouraud Shading zur Berechnung ihrer Bilder.

Nachfolgend wird davon ausgegangen, dass sich die (diffuse) Lösung der Rendering-Gleichung an den Eckpunkten mit Hilfe der Spherical Harmonics Funktion in einen anderen (Kugel-basierten) Funktionenraum transformieren lässt. Diese Überführung des Problems durch eine Basisprojektion (vgl. Abschnitt 2.2.2.2) liefert einen zusätz-

---

<sup>2</sup>Tatsächlich lässt sich sogar zeigen, dass wenn die Normalen - innerhalb eines Dreiecks - mit Hilfe einer bilinearen Interpolation der Eckpunktnormalen bestimmt werden, dann auch die diffuse (unbeschattete) Beleuchtung in der selben Art und Weise von den Lösungen an den Vertices abhängt.

lichen Zahlenvektor, der pro Vertex des Objekts gespeichert werden muss. Dabei lassen sich diese berechneten (reellen) Basiskoeffizienten während der Darstellung dazu benutzen, um die ausgehende Radianz an jedem Eckpunkt zu bestimmen. Dafür ist es jedoch zuvor nötig, dass die Lichtquellen ebenfalls in den selben Funktionenraum projiziert werden, denn um die Gleichung (2.13) zu evaluieren müssen alle notwendigen Eingabedaten in der selben Darstellungsform vorliegen. Die Bestimmung der - von einem beliebigen Vertex - ausgehenden Radianz, lässt sich dann durch den Einsatz eines sehr einfachen Vertex Shaders auf der Grafikkarte berechnen. Dazu ist es natürlich notwendig, die Basiskoeffizienten als lokale Parameter eines Dreiecks zu beschreiben. Dies stellt jedoch kein Problem dar, da - wie bereits weiter oben erklärt wurde - die Semantik der Vertexdaten keinesfalls festgeschrieben ist. So lassen sich zum Beispiel die Texturkoordinatenvektoren in Basiskoeffizientenvektoren umdefinieren. Da der Shader diese Assoziation „kennt“, kann er sie richtig interpretieren. Dabei gilt es allerdings zu beachten, dass nur eine begrenzte Anzahl von lokalen Parametern zur Verfügung steht. Daraus folgt jedoch wiederum, dass nur eine eingeschränkte Anzahl von Bändern verwendet werden kann, um das Integral der Hemisphäre zu repräsentieren (vgl. Abschnitt 2.2.2.2). Wenn  $n$  Bänder unterstützt werden sollen, dann müssen pro Farbkanal  $n^2$  Basiskoeffizienten gespeichert werden. Somit benötigt eine trichromatische Lösung mit zwei Bändern zwölf reelle Zahlen ( $2^2 * 3$ ) für jeden Basiskoeffizientenvektor. Sollen drei Bänder benutzt werden, sind bereits 27 Basiskoeffizienten ( $3^2 * 3$ ) nötig usw. Da die aktuellen Grafikkarten bis zu acht (vierelementige) Texturkoordinatenvektoren und eine Vertexfarbe unterstützen, stellt eine Repräsentation mit drei Bändern ( $\sqrt{(9 * 4)}/3$ ) die Obergrenze für eine trichromatische (Shader-basierte) Lösung der Rendering-Gleichung dar. Für die Basiskoeffizientenvektoren einer Repräsentation mit mehr als drei Bändern reicht nämlich der vorhandene Speicherplatz der lokalen Parameter der Eckpunkte nicht mehr aus. Wird lediglich eine monochromatische Lösung benötigt, dann lassen sich bis zu sechs Bändern ( $\sqrt{9 * 4}$ ) speichern und verarbeiten. Dabei repräsentieren diese Basiskoeffizienten vollständig die reflektierte Radianz, wobei sowohl die BSSRDF als auch die Umgebung beliebig gewählt werden kann. Auch globale Beleuchtungseffekte, wie Schatten oder von anderen Objekten (indirekt) reflektiertes Licht vergrößern den Berechnungsaufwand während der Darstellung nicht. Denn egal wie kompliziert die Einflüsse auf den betrachteten Punkt auch sind, letztendlich werden sie in einen Vek-

tor mit  $(n^2)$  Basiskoeffizienten zusammengefasst.

Je stärker die Werte der ursprüngliche Funktion fluktuieren, desto mehr Koeffizienten müssen verwendet werden, um eine hinreichend genaue Approximation mit Hilfe einer Basisprojektion herzustellen. Diese Eigenschaft lässt sich sehr gut in Abbildung 2.8 beobachten. Glücklicherweise variiert die diffuse Beleuchtung nur sehr schwach über der Hemisphäre, wodurch häufig eine Spherical Harmonics Lösung mit nur wenigen Bändern (Zwei bis Vier) ausreicht, um eine gute Approximation der Lösung der Rendering-Gleichung zu erhalten. Zwar kommt es durch die Verwendung einer solchen groben Näherung unter Umständen zu einigen Abweichungen bezüglich des korrekten Ergebnisses, dennoch lässt sich dieser Nachteil häufig akzeptieren, da dadurch eine äußerst effiziente Darstellung möglich wird.

Die nun folgende Betrachtung der Berechnung der Basiskoeffizienten setzt voraus, dass ein geometrisches Modell einer Pflanze in Form eines Dreiecksnetzes vorliegt. Dabei werden die Eckpunkte der Dreiecke in einer Liste  $V$  zusammengefasst, wobei die Daten der Vertices jeweils aus einer Position  $x$  und einer Normale  $\vec{n}$  bestehen. Nun können die Basiskoeffizienten  $\vec{c}$  für alle Eckpunkte bestimmt werden. Dazu wird jeweils das Integral über die Sphäre in eine Spherical Harmonics Repräsentation projiziert. Eine analytische Lösung ist dabei nur selten möglich, weshalb - auch an dieser Stelle - die Auswertung des Integrals mit einem numerischen Ansatz erfolgen muss. Hierbei kann zum Beispiel ein Raytracingansatz (vgl. Abschnitt 2.2.2.1) genutzt werden, der das Sphärenintegral durch eine endliche Anzahl von Abtaststrahlen auswertet. Sei  $R$  deshalb eine Menge mit optimal<sup>3</sup> gewählten Strahlrichtungen, die jeweils durch ein Winkelpaar  $(\theta, \varphi)$  beschrieben werden. Nun wird für jedes Element in  $R$  ein Strahl in die aktuelle Szene (also die Pflanze) verfolgt. Trifft dieser auf kein anderes Dreieck, dann können die Photonen der Umgebung ungehindert auf die betrachtete Stelle fallen und damit beleuchten. Dabei liefert die Auswertung der diffusen BSSRDF den Anteil des Lichts, der reflektiert (oder transmittiert) wird. Durch die erhaltene Intensität der reflektierten Photonen lässt sich nun - mit Hilfe der Spherical Harmonics Funktion - der Basiskoeffizientenvektor des verwendeten Strahls bestimmen, indem die aktuelle Strahlrichtung zunächst mit der Lichtintensität multipliziert und anschließend in die Kugelkoordinaten-basierte Beschreibung

---

<sup>3</sup>In der Praxis hat sich gezeigt, dass eine Hammersley Punktmenge ([Kel01]) diese Aufgabe hinreichend gut bewältigt.

überführt wird. Um das vollständige Integral über die Sphäre zu repräsentieren, müssen natürlich alle Richtungen aus  $R$  in der selben Art und Weise transformiert werden. Wenn ein Strahl dabei auf ein anderes Dreieck trifft, dann erhält der aktuell betrachtete Punkt keine direkte Beleuchtung aus der Umgebung, denn er liegt im Schatten eines anderen Objekts. Allerdings können hierbei - je nach BSSRDF des getroffenen Dreiecks - andere Effekte einen Einfluss besitzen, wobei ein Beispiel durch lichtdurchlässiges Pflanzengewebe dargestellt wird. Auf Grund der Anatomie von Pflanzenblättern durchquert ein gewisser Anteil des Lichts das Gewebe und verlässt es auf der gegenüberliegenden Seite wieder. Wird somit bei der Abtastung der Sphäre ein Blatt getroffen, so kann berechnet werden, wie viel Licht aus der Umgebung trotzdem - aufgrund der Lichtdurchlässigkeit des anderen Objekts - auf die betrachtete Stelle fällt. Weiterhin ist es möglich indirekte Beleuchtungseffekte, wie diffuse Lichtinterreflektion zu berücksichtigen<sup>4</sup>. Nachdem für jede Strahlrichtung in  $R$  die beschriebene Transformation durchgeführt wurde, lässt sich der finale Koeffizientenvektor des aktuellen Vertex bestimmen, indem der Mittelwert der Basiskoeffizientenvektoren aller Strahlen aus  $R$  gebildet wird. Nachfolgend ist die Berechnung der Basiskoeffizientenvektoren noch einmal in Form von Pseudocode dargestellt:

```
BerechneBasiskoeffizienten( Vertices V, Strahlrichtungen R )
```

```
  Für jeden Vertex  $v_i$  in V
```

```
     $v_i.c = ( 0, 0, \dots, 0 )$ 
```

```
  Für jede Richtung (theta, phi) in R
```

```
    Beleuchtung = EingehendeRadianz(  $v_i.x$ , theta, phi )
```

```
    Intensität = BSSRDF(  $v_i.x$ ,  $v_i.n$ , theta, phi )
```

```
     $c = \text{BerechneSphericalHarmonics}( \text{theta}, \text{phi} )$ 
```

```
     $c *= \text{Intensität} * \text{Beleuchtung}$ 
```

```
     $v_i.c += c$ 
```

```
   $v_i.c *= (4 * \text{PI}) / |R|$ 
```

---

<sup>4</sup>In einer realen Implementierung werden die Beleuchtungseinflüsse nacheinander in mehreren einzelnen Schritten berechnet. Zunächst erfolgt die Bestimmung der direkten Beleuchtung, danach wird für jede weitere Lichtreflektion ein neuer Berechnungsschritt ausgeführt.

Abbildung 4.5 veranschaulicht die Transformation der Rendering-Gleichung mit Hilfe der Spherical Harmonics Funktion noch einmal. Links ist die ursprüngliche Szene zu sehen, wobei die Abtastung der Umgebung eine sphärische Funktion beschreibt, welche die aktuelle Intensitätsverteilung (Abb. 4.5 Mitte) repräsentiert. Diese lässt sich auch durch einen Vektor aus Basiskoeffizienten darstellen (siehe Abb. 4.5 rechts).

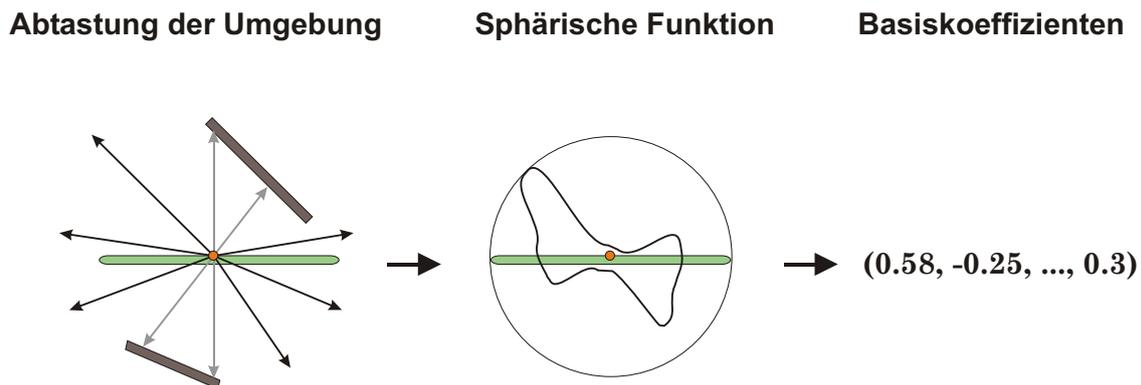


Abbildung 4.5: Spherical Harmonics Transformation der Rendering-Gleichung

Nachdem die Transformation der Rendering-Gleichung mit Hilfe der Spherical Harmonics Funktion ausgeführt wurde, kann die ausgehende (diffuse) Radianz an den Eckpunkten sehr leicht berechnet werden. Wie bereits weiter oben erwähnt wurde, ist es dafür jedoch notwendig die Lichtquellen ebenfalls in das neue Koordinatensystem zu projizieren. Die Frage ist nun, an welcher Stelle diese Projektion berechnet wird. Für eine korrekte Lösung der Rendering-Gleichung müsste die Transformation der Lichtquellen an jedem Vertex durchgeführt werden, denn schließlich wurden die Basiskoeffizienten der Eckpunkte an ihren jeweiligen Positionen bestimmt. Diese lokalen Lichtprojektionen sind aber im Kontext der Echtzeitvisualisierung zu kompliziert, da sie in der Regel nur von der CPU ausgeführt werden können. Deswegen muss eine weitere Vereinfachung eingeführt werden, welche dieses Problem effizient behebt. Wird dabei eine natürliche Szene betrachtet, zum Beispiel ein Baum in der freien Natur, so fällt auf, dass sich die (natürlichen) Lichtquellen im Vergleich zur Größe des Baums sehr weit weg befinden. Zum Beispiel befindet sich die Sonne in einer Distanz von zirka 150 Millionen Kilometern. Da ein durchschnittlicher Baum aber

lediglich die Größe von einigen Metern erreicht, kann davon ausgegangen werden, dass die von der Sonne eingestrahlte Radianz über die Fläche der Pflanze nur sehr geringfügig variiert, da alle Strahlen fast parallel auf den Baum treffen. Dieser Fakt lässt sich nun ausnutzen, indem die Lichtquellen lediglich an einem Punkt innerhalb der Pflanze in das neue Koordinatensystem transformiert werden. Dafür kann zum Beispiel der Mittelpunkt des darzustellenden Objektes verwendet werden<sup>5</sup>. Dabei lässt sich dieses Vorgehen intuitiv folgendermaßen verstehen: Jeder Eckpunkt wird von exakt der selben Umgebung beleuchtet. Da die einzelnen Vertices jeweils einen eigenen Basiskoeffizientenvektor besitzen, erfolgt die Reflektion des eingestrahlten Lichts in einer sehr individuellen Art und Weise. Liegen also nun sowohl die Basiskoeffizienten der Vertices als auch die der Beleuchtungsumgebung im selben Koordinatensystem vor, so kann die Gleichung (2.13) gelöst werden, um den konkreten Wert der ausgehenden Radianz zu berechnen. Um dabei die Summen leichter auswerten zu können, ist es möglich die zwei Bandindices  $l$  und  $m$  in eine lineare Form zu bringen, wobei sich der lineare Bandindex  $i$  folgendermaßen berechnet lässt:

$$i = l * (l + 1) + m \quad (4.2)$$

Durch diese Umformung ist nun lediglich eine Summe nötig, um die Integration über die Sphäre auszuwerten und damit die Rendering-Gleichung zu lösen. Mathematisch betrachtet wird dabei das Skalarprodukt zwischen den Basiskoeffizientenvektoren des Eckpunktes und der Lichtquelle berechnet. Durch die folgende einfache Funktion wird die ausgehende Radianz an einem Eckpunkt bestimmt:

```
AusgehendeRadianz( Vertex v, Beleuchtungsumgebung l )
```

```
    Ergebnis = SkalarProdukt( v.c, l.c )
```

Diese Berechnung lässt sich auch mit Hilfe eines einfachen Vertex Shaders ausführen, wodurch eine sehr effiziente Darstellung von Pflanzen (und anderen Objekten) möglich wird, wobei die Komplexität der diffusen BSSRDF und der Beleuchtungsumgebung keinen Einfluss auf die erreichte Laufzeit-Performanz besitzen.

---

<sup>5</sup>In der Praxis hat sich sogar gezeigt, dass dieses Vorgehen auch für sehr viel nähere Lichtquellen verwendet werden kann. Die entstandene Abweichung vom exakten Ergebnis wird häufig nicht erkannt.

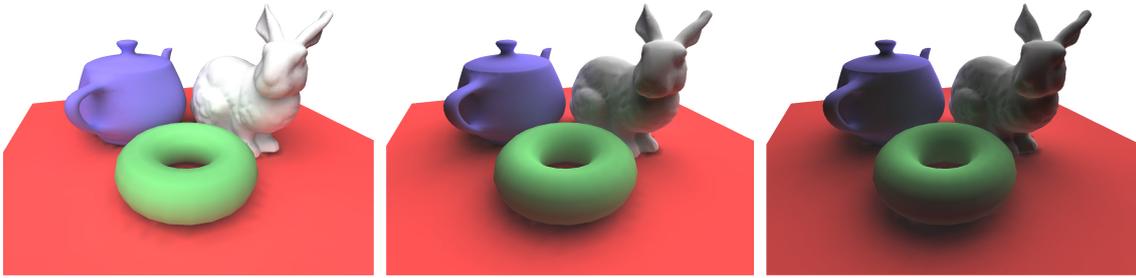


Abbildung 4.6: Spherical Harmonics Darstellung mit einem Vertex Shader (300 FpS)

Während der Berechnung der Basisoeffizienten muss darauf geachtet werden, dass nur der diffuse - sich langsam über die Fläche ändernde - Anteil der BSSRDF mit einbezogen wird. Andernfalls würde die ursprüngliche Funktion zu stark variieren, wodurch mehr Werte für die Spherical Harmonics Repräsentation nötig wären. Deshalb sollte die Textur bei der Basisprojektion noch keinen Einfluss besitzen, sondern erst beim Rendering. Dieses Vorgehen wird aus dem selben Grund auch beim Gouraud Shading benutzt und häufig als Modulation bezeichnet. In der Praxis wird deshalb die Grundfarbe des Dreiecks zunächst im Vertex-Shader bestimmt und dann an das Fragment Programm „weitergeleitet“. Dort kann anschließend die Farbe des Ergebnis-Pixels durch eine Komponentenweise Multiplikation von Grund- und Texturfarbe berechnet werden. Dabei fügt die Textur dem Ergebnis höherfrequente (d.h. sich häufig ändernde) Informationen hinzu. Aus dieser Trennung von Basisfarbe und Textur, kann eine weitere Vereinfachung geschlussfolgert werden, denn unter gewissen Umständen reicht auch für farbige Materialien eine monochromatische Approximation der Rendering-Gleichung aus. Nämlich immer genau dann, wenn die Lichtquellen einfarbig sind und die Grundfarbe vollständig mit in der Textur enthalten ist. Diese Eigenschaft wird im nächsten Abschnitt benutzt, um die stark vereinfachten Pflanzenmodelle zu beleuchten.

Bis zu dieser Stelle wurde ausschließlich von einer diffusen BSSRDF ausgegangen. Natürlich stellt auch das spekulare reflektierte Licht einen wichtigen Anteil bei der Darstellung von Pflanzengewebe dar. Dabei ist es grundsätzlich möglich, auch die spekularen Effekte durch eine Spherical Harmonics Transformation zu beschreiben. Dafür schlugen Sloan *et al.* ([SKS02]) die Verwendung einer Basisoeffizientenma-

trix anstelle der Koeffizientenvektoren vor. In diesem Fall besteht die Lösung der Rendering-Gleichung aus mehreren einzelnen Basisvektoren, die unterschiedliche Betrachtungsrichtungen repräsentieren. Allerdings besitzt dieses Vorgehen auch einige Nachteile. Da die ausgehende Radianz lediglich an den Eckpunkten berechnet wird und die spekulare Reflektion die Eigenschaft besitzt relativ stark innerhalb einer Fläche zu variieren, muss die Vertexdichte aller Objekte stark erhöht werden, um einen solchen Effekt überhaupt darstellen zu können. Außerdem müssen pro Basiskoeffizientenmatrix mehr Werte gespeichert werden, weil die BSSRDF bereits hochfrequente Informationen besitzt. Dadurch steigt jedoch auch der Speicherverbrauch der Lösung an, was die Echtzeitvisualisierung erschwert.

Wie bereits weiter oben erläutert wurde, lässt sich die BSSRDF in Einzelteile aufspalten. Somit wird es möglich, die diffuse und spekulare Reflektion getrennt von einander zu behandeln. Während also die langsam variierenden Anteile der ausgehenden Radianz mit einer Spherical Harmonics Approximation beschrieben werden, kann für die spekulare Reflektion ein Standardmodell ([Pho73], [Bli77]) verwendet werden<sup>6</sup>. Dieses Vorgehen bietet zusätzlich den Vorteil, dass es sich optimal mit Shadern umsetzen lässt. Dabei bestimmt das Vertex Programm zunächst die (niederfrequente) Grundbeleuchtungsfarbe und der Pixel-Shader fügt dieser Lösung danach hochfrequente Informationen hinzu, indem das Ergebnis mit der Texturfarbe moduliert und anschließend um den Betrag der spekularen Reflektion inkrementiert wird.

Die Verwendung einer Spherical Harmonics Repräsentation führt jedoch zu einem großen Nachteil, denn es wird nämlich vorausgesetzt, dass die Welt vollkommen statisch ist. Wurden die Basiskoeffizienten erst einmal berechnet, so sind keine weiteren Änderungen an der geometrischen Beschreibung der Szene möglich. Kein Objekt darf verschoben, rotiert oder skaliert werden. Diese Einschränkung folgt vor allem aus der Art und Weise der Berechnung der Basiskoeffizienten. Dabei wird nämlich die (Hemi-) Sphäre um die Eckpunkte der Dreiecke abgetastet und als Koeffizientenvektor abgespeichert. Dadurch darf sich jedoch die Umgebung nicht mehr verändern, denn andernfalls würden die vorhandenen Basiskoeffizienten nicht mehr mit der aktuellen Situation übereinstimmen. Dieser Nachteil ist zwar sehr stark, in der Praxis hat sich jedoch gezeigt, dass diese Einschränkung nicht so restriktiv beachtet wer-

---

<sup>6</sup>Die Beschattung durch andere Objekte lässt sich approximieren, indem der spekulare Anteil mit der Intensität der Spherical Harmonics Lösung moduliert wird.

den muss, denn es ist sehr häufig möglich die einzelnen Szenenelemente unabhängig voneinander in den anderen Funktionenraum zu überführen. Die dadurch fehlende Lichtinteraktion zwischen den Objekten, fällt einem Betrachter im Allgemeinen gar nicht auf. Lediglich wenn sich zwei Szenenelemente so nah kommen, dass sie Schatten aufeinander werfen würden, kann dies zu Problemen führen. Für diesen speziellen Problemfall wird in einem späteren Abschnitt jedoch eine approximative Lösungsmöglichkeit eingeführt, die in natürlichen Szenen relativ gut funktioniert.

Abschließend sollen noch einmal die Vor- und Nachteile dieser Darstellungsart abgewogen werden.

Nachteile:

- Erhöhter Speicherverbrauch durch die Basiskoeffizienten
- BSSRDF mit geringer Varianz wird vorausgesetzt
- Statische Welt wird vorausgesetzt
- Vorberechnungsphase zur Bestimmung der Koeffizientenvektoren

Vorteile:

- Hohe Laufzeitperformanz
- Unterstützung beliebiger BSSRDFs
- Darstellung aller Beleuchtungseffekte (Schatten, Globale Beleuchtung, ...)
- Beliebige Anzahl von Lichtquellen

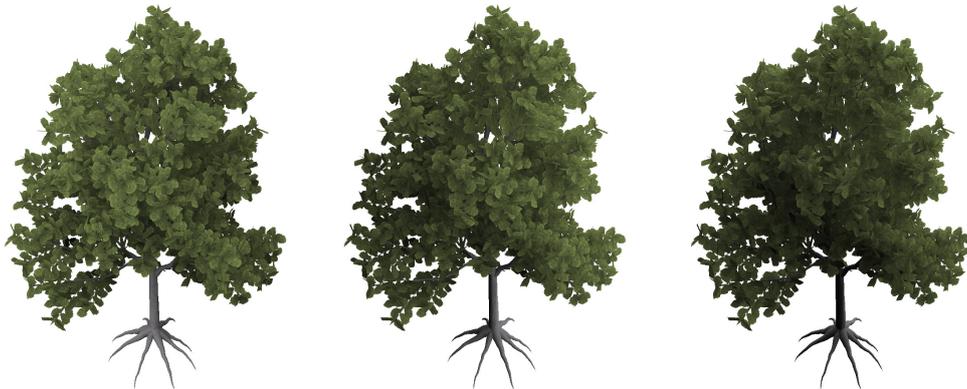


Abbildung 4.7: Ein Baum in verschiedenen Beleuchtungssituationen (300 FpS)

### 4.3 Darstellung in einem Level of Detail Kontext

Nachdem der Darstellungsalgorithmus für Pflanzen so weit verändert wurde, dass er sich sehr effizient mit der aktuellen Grafikhardware verarbeiten lässt, stellt lediglich die geometrische Komplexität der darzustellenden Modelle ein Performanzproblem dar. Dies kommt insbesondere in umfangreichen Szenen zum tragen, so besteht beispielsweise eine Baumgruppe oder gar ein kleiner Wald aus Abertausenden einzelnen Bäumen. Die bisher verwendeten geometrischen Vereinfachungen, welche zum Beispiel ein Pflanzenblatt als eine endliche Ebene beschreiben, reichen jedoch immer noch nicht aus, um entsprechend hohe Bildberechnungsraten zu erreichen. Zur Verdeutlichung dieses Problems soll hierbei ein sehr einfacher Baum als Beispiel dienen, der lediglich mit 10000 Dreiecken beschrieben wurde. Wenn weiterhin davon ausgegangen wird, dass zirka 15000 Einzelbäume verwendet werden und dabei eine Framerate von 15 Bildern pro Sekunde eingehalten wird, dann müsste die Grafikkarte mehr als zwei Milliarden (!) Dreiecke pro Sekunde verarbeiten und darstellen. Die Entwicklung der Grafikhardware geht zwar sehr schnell voran, aber es dürfte wohl noch einige Jahre dauern, bis sich eine solche Szene in der vollen geometrischen Komplexität in Echtzeit visualisieren lässt.

Es wird somit eine andere Vorgehensweise benötigt, um komplexe Landschaften darzustellen. Dabei lässt sich die begrenzte Ortsauflösung des menschlichen Auges aus-

nutzen, denn es ist offensichtlich, dass mit zunehmender Distanz immer weniger Details gesehen werden können. Der Text auf einem A4 Blatt, welches sich in einem Kilometer Entfernung vom Betrachter befindet, kann (ohne Hilfsmittel) nicht mehr gelesen werden. Dies bedeutet jedoch, dass sich auch die Details einer Pflanze ab einer bestimmten Distanz nicht mehr wahrnehmen lassen. Alles was nicht gesehen wird, sollte aber auch nicht gerendert werden und somit ist es möglich die geometrische Komplexität der Modelle mit wachsendem Abstand zu reduzieren. Da der Detailgrad der darzustellenden Objekte somit variabel ist, wird dieser Ansatz auch Level of Detail genannt.

Wie diese vereinfachten geometrischen Repräsentationen erstellt werden, hängt voll und ganz von dem darzustellenden Objekt ab, denn Landschaften werden zum Beispiel anders behandelt als Häuser oder Autos. Diese Ansätze funktionieren allerdings alle sehr ähnlich, denn häufig werden mehrere Dreiecke zusammengefasst und durch ein (neues) Dreieck repräsentiert. Dieses Vorgehen lässt sich für Pflanzen jedoch nur schlecht verwenden, da diese fast immer mit Hilfe von (semi-transparenten) Texturen dargestellt werden. Dabei ist es jedoch nicht offensichtlich, wie die verschiedenen Texturkoordinaten bei der Kombination mehrere Dreiecke behandelt werden sollten. Der nächste Abschnitt beschreibt deswegen einen Ansatz, mit dem sich geometrisch extrem vereinfachte Modelle für Pflanzen erzeugen lassen.

Ein weiteres Problem, welches aus der Verwendung eines Level of Detail Ansatzes folgt, sind Umschaltartefakte, denn meist sind die geometrischen Vereinfachungen nicht subtil genug, so dass ein Betrachter die Änderung des Detailgrades wahrnehmen kann. Auch die Beleuchtung stellt in diesem Kontext ein großes Problem dar, denn durch das Zusammenfassen mehrerer Dreiecke verändert sich auch die Topologie eines Objekts mehr oder weniger schlagartig, wobei die geänderten Normalen im Allgemeinen ruckartig zu einer anderen reflektierten Radianz führen. Dieses Problem ist im Fall einer Level of Detail Darstellung von Pflanzen ganz besonders schwerwiegend, da die Topologie der vereinfachten Repräsentationsformen fast überhaupt nicht mehr mit dem ursprüngliche Modell übereinstimmt. Deshalb sollte also ein kohärentes Beleuchtungsmodell verwendet werden, welches für alle geometrischen Vereinfachungen ein korrektes Ergebnis liefert und zwar unabhängig von der Topologie der Modelle. Die Entwicklung eines solchen Darstellungsverfahrens wird ebenfalls Thema der nachfolgenden Abschnitte sein.

### 4.3.1 Erstellung einer Billboardrepräsentation

Pflanzen unterscheiden sich auch geometrisch sehr stark von anderen Objekten, denn sie besitzen häufig eine sehr unregelmäßige Topologie. Im Gegensatz dazu lassen sich Häuser und Autos in der Regel durch eine geschlossene konvexe oder konkave Oberfläche beschreiben. Werden in diesem Fall zwei benachbarte Dreiecke zusammengefasst, so ändert sich die Silhouette des Objekts (z.B. des Autos) nur geringfügig. Pflanzenmodelle zeigen dieses Verhalten jedoch nicht, denn sie besitzen große durchsichtige Lücken, wodurch eine Veränderung der Topologie sehr schnell zu einer anderen Kontur führt. Bei einer solchen Kombination mehrerer Blätter, muss deswegen insbesondere auf die benutzten Texturen geachtet werden, denn die ursprünglichen Bilder (siehe Abb. 3.8) lassen sich für eine zusammengefasste Blattgruppe nicht mehr direkt verwenden. Vielmehr muss eine spezielle Textur erzeugt werden, welche die Charakteristik der kombinierten Blätter repräsentiert. Diese neue Textur kann dann der vereinfachten Geometrie der Blattgruppe zugeordnet werden.

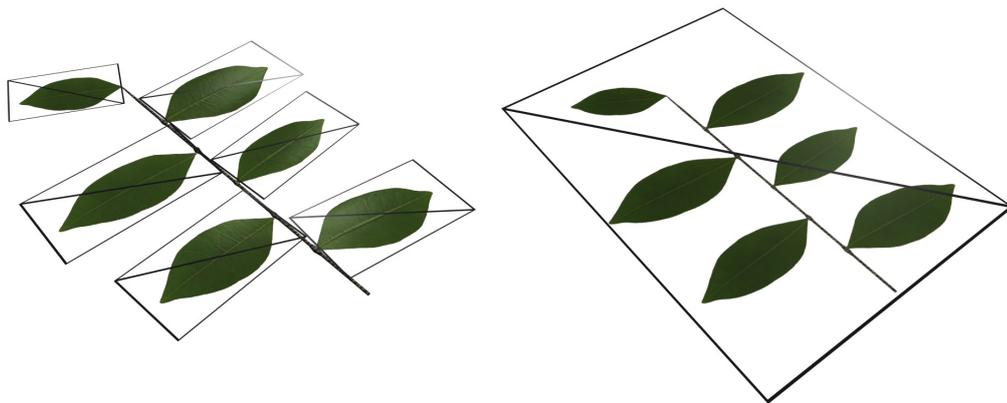


Abbildung 4.8: Approximation eines Astes durch ein Billboard

Aus diesem Vorgehen folgt nun die grundsätzliche Idee, Pflanzen durch eine Menge von texturierten Ebenen - den so genannten Billboards - darzustellen, wobei benachbarte koplanare Elemente einer Pflanze zu einer endlichen Ebene (Billboard) zusammengefasst werden. Wird die Textur dieser neuen Ebene „geschickt“ gewählt, so lässt sich der Unterschied zum ursprünglichen Ensemble nur schwer ausmachen. Dies gilt insbesondere dann, wenn sich der Betrachter in einer relativ großen Entfernung zur Pflanze befindet. Abbildung 4.8 veranschaulicht das Zusammenfassen der Elemente

eines Astes (Links) zu einem Billboard (Rechts). Selbst in dieser einfachen Szene wird der Vorteil dieser Vorgehensweise offensichtlich, denn die Anzahl der Dreiecke des ursprünglichen Astes (ca. 200) ist 100 mal höher als die des Billboards (2).

Nun stellt sich lediglich die Frage, wie sich die Billboards bestimmen lassen, welche eine bestimmte Pflanze approximieren sollen. Dazu wurden in den letzten Jahren einige interessante Ansätze ([DDSD03]) vorgeschlagen, die aber alle das selbe Grundprinzip benutzen. Dabei „suchen“ iterative Clusteringansätze nach ähnlichen Dreiecken, die dann zusammenfasst und durch ein neues Oberflächenelement ersetzt werden. Die Definition von Ähnlichkeit und Verschiedenheit, variiert jedoch von Ansatz zu Ansatz. Die Basis für diese Diplomarbeit wird durch die Masterarbeit ([Beh04]) von Stephan Behrendt gebildet, wobei unterschiedliche Optimierungen des k-means Clusteringalgorithmus mit pflanzenspezifischen Heuristiken kombiniert wurden. Das Ergebnis dieses Verfahrens besteht aus einer Menge von Billboards, denen jeweils eine Liste mit Dreiecken der ursprünglichen Geometrie zugeordnet ist. Dieser Teil der Originalpflanze wird dann in ein Bild gerendert und anschließend als Textur des neuen Billboards festgelegt. Dabei kommt die Billboardrepräsentation einer Pflanze in der Regel mit relativ wenigen Ebenen (50 - 200) aus, wodurch die geometrische Komplexität der Pflanzen erheblich sinkt. Gerade das Blattwerk lässt sich sehr gut durch Billboards approximieren, wobei kleinere Abweichungen und Diskontinuitäten kaum auffallen und somit toleriert werden können. Anders verhält sich dies jedoch beim Geäst und dem Hauptstamm, wobei selbst kleine Abweichungen sofort auffallen, denn sie lassen sich häufig als unrealistische Lücken erkennen. Dabei wirken sich auch die verwendeten Heuristiken kaum positiv auf das erstellte Ergebnis aus. Deswegen bietet es sich wahrscheinlich an, den Hauptstamm und Teile des Geästs als niedrig aufgelöste Zylinder darzustellen. Leider konnte diese These - auf Grund von mangelnder Zeit - nicht mehr im Rahmen dieser Diplomarbeit überprüft werden.

Abbildung 4.9 zeigt eine Billboardrepräsentation (Rechts) eines Baums (Links), wobei die ursprüngliche Pflanze aus zirka 61000 Dreiecken besteht, während die vereinfachte Darstellung lediglich 200 Dreiecke benötigt. Trotz dieser starken Vereinfachungen kann das Objekt deutlich als (der selbe) Baum identifiziert werden.

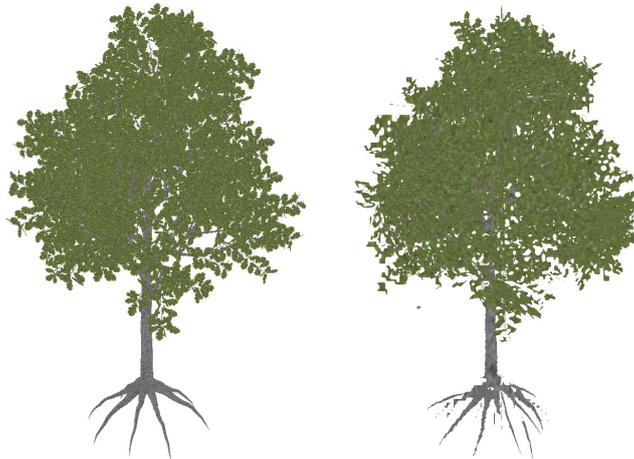


Abbildung 4.9: Billboardrepräsentation eines Baums

### 4.3.2 Realistische Darstellung von Billboardpflanzen

Damit sich die vereinfachten Billboardrepräsentationen realistisch in ihre Umgebung integrieren, wird ein entsprechendes Beleuchtungsmodell benötigt, um diese Objekte realistisch darzustellen. Die geometrischen Vereinfachungen führen in diesem Rahmen jedoch zu einigen Problemen, denn durch die Kombination mehrerer Einzelteile zu einer endlichen Ebene werden nämlich geometrische Details zerstört. Dies folgt aus der Projektion der einzelnen Elemente auf die Billboardebene, weil dabei sowohl die Normalen- als auch die Positionsdaten der kombinierten Elemente verändert werden. Denn die Einzelteile liegen nunmehr alle auf der selben Billboardebene und nehmen dabei auch ihre Normale an. In Abbildung 4.10 wird dieses Problem veranschaulicht, wobei auf der linken Seite die ursprüngliche Szene und auf der rechten Seite die Billboardrepräsentation zu sehen ist.



Abbildung 4.10: Informationsverlust durch die Projektion

Dieser Informationsverlust führt gerade bei der realistischen Darstellung zu Problemen, denn dafür ist sowohl der exakte Punkt im Raum als auch eine korrekte Oberflächennormale von größter Bedeutung. Sind diese Daten jedoch nicht vorhanden, dann kann weder die diffuse noch die spekulare Reflektion hinreichend genau bestimmt werden. Eine Möglichkeit dieses Problem zu beheben, wäre die Verwendung einer Verwerfungstextur mit deren Hilfe sich die verloren gegangenen Daten rekonstruieren lassen, wobei dieses zusätzliche Bild bei der Erstellung der Billboardtextur mit erzeugt werden könnte. Dabei reicht eine Textur mit vier Kanälen aus, um später alle Informationen wiederherstellen zu können, denn die ursprüngliche Normale kann in den ersten drei Kanälen abgespeichert werden, während der vierte Kanal eine Verwerfungsdistanz relativ zur neuen Ebene enthält. Damit wird sozusagen die inverse Billboard-Projektion (siehe Abb. 4.10) beschrieben. Diese Vorgehensweise funktioniert sehr gut, so lange lediglich ein lokales Beleuchtungsmodell ohne Schatten benutzt werden soll. Allerdings lässt sich dieses Verfahren nur sehr schwer mit dem Shadow Mapping Ansatz kombinieren, so dass in den Ergebnisbildern teilweise Schattenlücken sichtbar werden. Da korrekt berechnete Schatten jedoch für eine realistische Darstellung sehr wichtig sind, kann dieser Ansatz nicht verwendet werden, um die Billboardrepräsentationen der Pflanzen darzustellen. Es muss somit eine weitere Vorgehensweise ausfindig gemacht werden.

Um das Problem des Informationsverlustes zu umgehen, sollte das Visualisierungsverfahren während des Renderings keine Positions- und Normalendaten mehr benötigen. Diese Forderung scheint zunächst jedoch unerfüllbar, denn die reflektierte Radianz hängt vollkommen von der örtlichen Beschreibung der darzustellenden Oberfläche ab. Durch die Verwendung einer vorberechneten Spherical Harmonics Lösung der Rendering-Gleichung lässt sich diese Abhängigkeit jedoch auflösen und somit ein Verfahren entwickeln, welches die ausgehende Radianz vollkommen unabhängig von den aktuellen Positions- und Normalendaten bestimmt. Diese Informationen über die darzustellende Oberfläche sind nämlich implizit in den Basiskoeffizienten enthalten und werden somit explizit nicht mehr benötigt. Ein solcher Ansatz würde also das aktuelle Problem der realistischen Darstellung von Billboardpflanzen vollkommen lösen. Allerdings muss zuvor der ursprüngliche Spherical Harmonics Darstellungsalgorithmus abgeändert werden, denn die Eckpunkte stellen im Fall einer Billboarrepräsentation keine akzeptablen Vorberechnungspunkte mehr dar. Schließ-

lich gehört ein Dreieck nun nicht mehr lediglich zu einem Teil der Pflanze, sondern es kann viele verschiedene Elemente enthalten. Zum Beispiel beschreiben die Dreiecke des Billboards in Abbildung 4.8 (Rechts) mehrere Blätter und den Ast.

Es bleibt also nur die Benutzung einer zusätzlichen Textur übrig, um eine Pixelgenaue Lösung des Problems zu ermöglichen. Dieses neue Bild enthält dabei die Basiskoeffizienten der zusammengefassten Elemente der ursprünglichen Pflanze. Während des Renderings des Ergebnisbildes, kann dann der jeweils zugehörige Koeffizientenvektor aus dieser neuen Textur ausgelesen und zur Bestimmung der reflektierten Radianz verwendet werden. Dieses Vorgehen stellt dabei sicher, dass die auf der Billboardtextur sichtbaren Pflanzenteile in der Art und Weise beleuchtet werden, wie dies auch im ursprünglichen Modell erfolgt wäre. Somit wird der Informationsverlust, welcher durch die Erstellung der Billboards entsteht, fast vollständig ausgeglichen. Da die Beleuchtung nun individuell für jeden Pixel berechnet wird, kann allerdings kein Vertex-Shader mehr zur Darstellung der Pflanzen verwendet werden, sondern es wird ein erweitertes Fragment Programm benötigt, welches nun das Skalarprodukt der Basiskoeffizientenvektoren der neuen Textur und der Lichtquelle berechnet. Somit bestehen die neuen Operationen also lediglich aus einem zusätzlichen Texturzugriff und einigen Additionen und Multiplikationen, wodurch sich die Komplexität des Pixel-Shaders nur geringfügig erhöht und weiterhin eine hohe Performanz erreicht wird. Nachfolgend ist diese Vorgehensweise noch einmal in Form einer Pseudocode-Darstellung zu sehen:

```
AusgehendeRadianz( Texturkoordinat t, Beleuchtungsumgebung l )
```

```
Grundfarbe = Billboardtextur( t )  
Basiskoeffizienten = Beleuchtungstextur( t )  
Beleuchtung = SkalarProdukt( Basiskoeffizienten, l.c )  
Ergebnis = Grundfarbe*Beleuchtung
```

Die Berechnung des zusätzlichen Beleuchtungsbildes erfolgt ebenfalls bei der Erstellung der Billboardtextur. Dabei hängt die Anzahl der benötigten Bildkanäle von der Menge ( $n$ ) der verwendeten Bänder ab. Hierbei gilt die selbe Eigenschaft wie bei der Lösung pro Eckpunkt und somit müssen für jeden Spherical Harmonics Farbkanal  $n^2$

Werte abgespeichert werden. Daraus folgt allerdings auch an dieser Stelle eine Einschränkung, denn die aktuellen Grafikkarten unterstützen maximal acht verschiedene Texturen. Da alle Billboards auf jeden Fall ein Bild mit ihrer Basisfarbe benötigen, verringert sich die Anzahl der verfügbaren Texturen auf Sieben. Das heißt eine trichromatische Lösung darf maximal drei Bänder ( $\sqrt{(7 * 4)/3}$ ) verwenden, während eine monochromatische Beschreibung immerhin fünf Spherical Harmonics Bänder ( $\sqrt{7 * 4}$ ) benutzen kann.

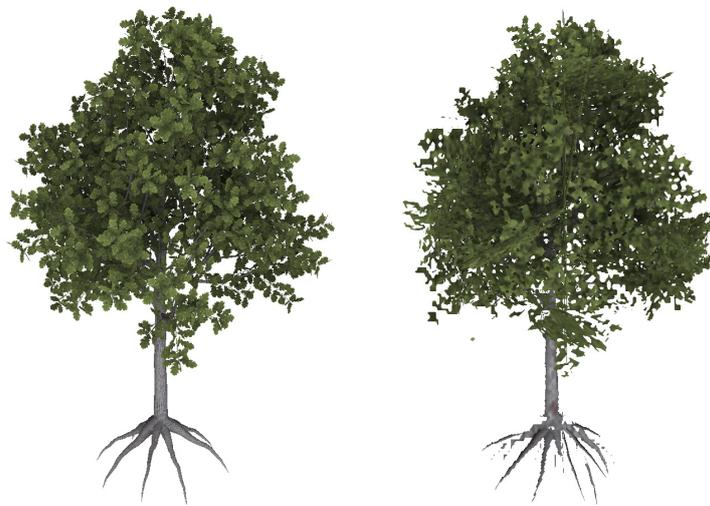


Abbildung 4.11: Beleuchtung einer Billboardrepräsentation

Für den konkreten Anwendungsfall der Pflanzendarstellung wirkt sich diese Einschränkung allerdings kaum aus, da häufig eine monochromatische Lösung mit wenigen Bändern für eine realistische Visualisierung ausreicht. Diese Eigenschaft folgt vor allem aus der Struktur der ursprünglichen Geometrie, denn das Blattwerk eines Baums besteht beispielsweise aus vielen einzelnen planaren Elementen, die jeweils eine relativ geringe Fläche besitzen. Dadurch ist jedoch die Varianz der Beleuchtung innerhalb eines Blattes von vornherein sehr klein, wodurch bereits relativ wenige Bänder in der Lage sind eine hinreichend gute Darstellung zu ermöglichen. Als Beispiel soll der Baum in Abbildung 4.7 dienen, welcher lediglich zwei monochromatische Spherical Harmonics Bänder verwendet. Wird also eine Billboardrepräsentation von dieser Pflanze berechnet, so reicht ein Bild mit vier Kanälen ( $2^2$ ) für die Beleuch-

tungstextur aus. In Abbildung 4.11 ist deswegen ein Vergleich zwischen diesem Baum und seiner Billboardapproximation zu sehen. Trotz der extremen Vereinfachungen der Geometrie ist eine realistische Darstellung möglich, was zum großen Teil auf die korrekte Beleuchtung zurückzuführen ist.

Da die Beleuchtungstextur auf Basis der ursprünglichen Spherical Harmonics Koeffizienten erstellt wurde, folgt neben der sehr realistischen Darstellung ein weiterer Vorteil aus dieser Vorgehensweise, denn es wird ein hohes Maß von Flexibilität bezüglich der benutzten geometrischen Vereinfachungsmethode erreicht. Diese muss lediglich eine Voraussetzung erfüllen, dass sie nämlich texturierte Oberflächen erzeugt. Die Erstellung einer Billboardwolke mit Hilfe eines Clusteringverfahrens genügt dieser Bedingung, stellt aber nur eine spezielle Vorgehensweise dar, denn noch stärkere geometrische Vereinfachungen lassen sich durch einen Kreuzebenen-Ansatz erzielen. Dabei wird eine Pflanze lediglich durch ein, zwei oder drei Billboards dargestellt, welche in den Koordinatenebenen des kartesischen Raums liegen (siehe Abb. Abbildung 4.12). Allerdings sind bei diesem Vorgehen die Parallaxenfehler (siehe Abb. 4.13) sehr stark sichtbar und zwar insbesondere dann, wenn eine sehr nahe Betrachtung der vereinfachten Modelle erfolgt. Somit eignen sich diese Modelle nur für den Hintergrund einer Szene, denn aufgrund der großen Distanz der Pflanzen zum Betrachter fallen diese Ungenauigkeiten dann nicht mehr auf. Die Verwendung von unterschiedlichen Repräsentationen im Nah- und Fernbereich der Szene führt jedoch zu Problemen beim Wechsel der aktuellen Level of Detail Stufe. Deswegen wird im nächsten Abschnitt ein Verfahren erläutert, wie sich die verschiedenen Repräsentationsformen ineinander überblenden lassen.

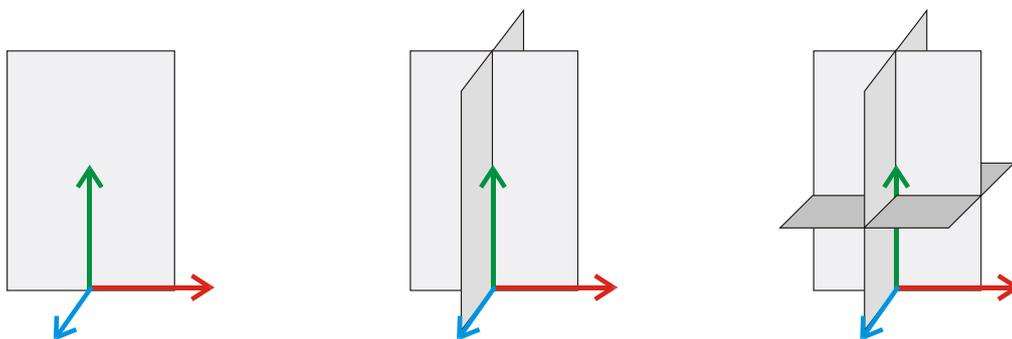


Abbildung 4.12: Kreuzebenen Billboards

Abschließend soll noch ein Blick auf die Texturen geworfen werden. Um Aliasing, das heißt die Unterabtastung dieser Bilder zu vermeiden, werden in der Echtzeit-Computergrafik so genannte Mip Map Stufen während des Renderings von texturierten Dreiecken verwendet. Dabei handelt es sich um verkleinerte, vorgefilterte Versionen der ursprünglichen Textur. Eine Oberfläche, die sich sehr weit entfernt befindet nutzt dabei eine kleinere und damit stärker gefilterte Mip Map Stufe als ein Fragment, welches dem Betrachter sehr nah ist. Diese Mip Map Bilder werden im Allgemeinen erzeugt, indem ihre Größe - ausgehend von der Originaltextur - immer wieder halbiert wird, wobei der Wert eines neuen Pixels aus dem Mittelwert von vier ursprünglich benachbarten Pixeln bestimmt wird. Diese rekursive Berechnung wird erst dann beendet, wenn die kleinste Mip Map Textur (Auflösung 1x1) erzeugt wurde. Aus dieser Vorgehensweise folgt jedoch ein Problem, denn die Berechnung der vorgefilterten Texturen erfolgt ohne Berücksichtigung der Silhouette einer Billboardpflanze, weil die Mip Map Stufen des zugehörigen Maskierungsbildes ebenfalls durch eine normale Mittelwertberechnung erstellt werden. Dadurch wird allerdings die Kontur immer mehr verwischt, was jedoch offensichtlich zu einem inkorrekten Ergebnis führt, da nun auch die Teile der Billboardtextur sichtbar werden, die eigentlich gar keine validen Daten enthalten. Deswegen muss bei der Erstellung der Mip Map Bilder unbedingt eine konturerhaltenden Filtermethode benutzt werden, wobei der Mittelwert nur noch von den Quellpixeln bestimmt wird, die überhaupt sichtbar sind und somit die einen Maskierungswert (Alpha) größer als Null besitzen.



Abbildung 4.13: Kreuzbillboardrepräsentation einiger Bäume

Um das Aliasing noch weiter zu reduzieren, bezieht die Grafikkarte bei einem Texturzugriff nicht nur den direkt adressierten Pixel, sondern auch dessen Nachbarn mit in die Farbbestimmung ein. Dabei wird entsprechend der aktuellen Texturcoordinate eine bilineare Interpolation der vier nächsten Werte durchgeführt. Doch auch hierbei findet die Kontur der Pflanze keine Beachtung, was jedoch gerade bei der Beleuchtungstextur zu einem großen Problem führt, weil durch die Interpolation auch die Pixel mit einbezogen werden, die keine validen Basiskoeffizienten enthalten. Der zurückgegebene Spherical Harmonics Koeffizientenvektor ist nun ebenfalls inkorrekt, wodurch eine falsche reflektierte Radianz bestimmt wird. Dieser Fehler ist im Ergebnisbild durch übermäßig dunkle oder helle Pixel sichtbar. Zwar lässt sich die Texturinterpolation generell deaktivieren, allerdings sinkt dadurch auch die Qualität der gerenderten Bilder, weil nun auf der Fläche innerhalb der Pflanze Blockartefakte zu sehen sind. Eine Lösungsmöglichkeit für dieses Problem besteht darin, für alle Pixel der Beleuchtungstextur einen (möglichst) plausiblen Koeffizientenvektor zu bestimmen. Dafür kann zum Beispiel so lange eine Expansion der validen Spherical Harmonics Werte durchgeführt werden, bis die Beleuchtungstextur mit genügend plausiblen Vektoren gefüllt ist. Auf der linken Seite von Abbildung 4.14 ist die Basisfarbentextur (a) eines einfachen Kreuzbillboards (vgl. Abb. 4.12 (b)) zu sehen. Rechts daneben wird die ursprüngliche Beleuchtungstextur (b) dargestellt, die expandierte Version dieses Bildes (c) lässt sich schließlich auf der rechten Seite von Abbildung 4.14 betrachten.

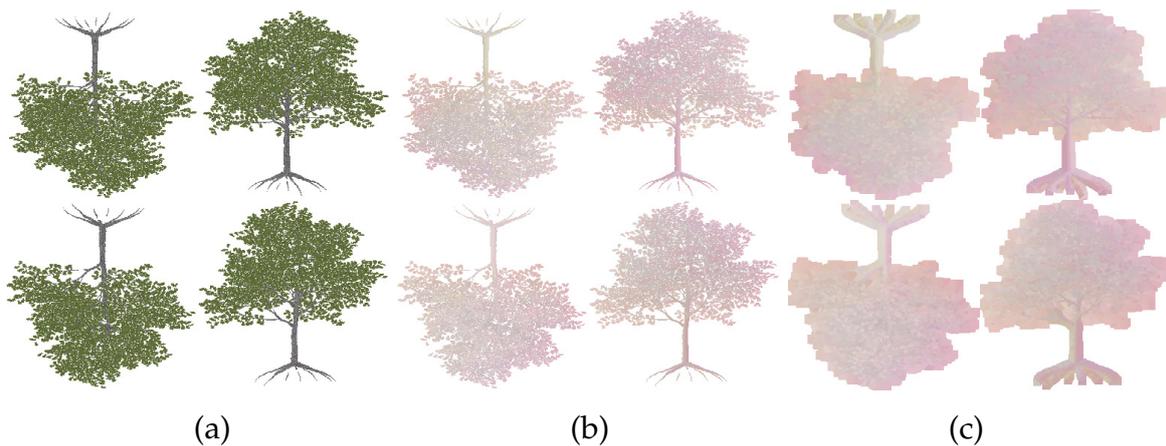


Abbildung 4.14: Texturen eines Kreuzbillboards

### 4.3.3 Visualisierung komplexer Landschaften

Die Visualisierung von größeren Teilen der Natur stellt ein kompliziertes Problem in der Computergrafik dar, denn neben der hohen geometrischen Komplexität einer solchen Landschaft, bereiten auch die komplizierten Beleuchtungseffekte einige Schwierigkeiten. So ist es - selbst mit der modernsten Rechentechnik - nicht möglich einen vollständigen Wald unter Berücksichtigung aller vorhandenen Lichtinteraktionen darzustellen. Deswegen gilt es, dass komplexe Problem in leichter zu lösende Einzelteile zu zerlegen. Aus diesem Grund werden die einzelnen Pflanzen häufig unabhängig von einander und ohne Berücksichtigung weiterer Objekte in der Umgebung verarbeitet und dargestellt. Dieses Vorgehen führt jedoch dazu, dass zum Beispiel keine Schatten zu sehen sind, wodurch jedoch die realistische Wirkung der visualisierten Szene verloren geht. Für die Lösung solcher Probleme wurden über die Jahre einige interessante Ansätze vorgeschlagen, welche eine realistische Darstellung von komplexen Landschaften ermöglichten. An dieser Stelle sollen deswegen zunächst zwei der wichtigsten Methoden kurz vorgestellt werden. Anschließend wird ein neues Verfahren eingeführt, welches auf die Erkenntnisse der vorherigen Abschnitte aufbaut, um natürliche Landschaften realistisch in Echtzeit zu visualisieren.

Bereits 1985 schlugen Reeves und Blau ([Ree83], [RB85]) die Verwendung von beleuchteten Partikelsystemen vor, um das Blattwerk eines Waldes darzustellen, wobei die Blätter dabei lediglich durch eine farbige Kugel modelliert wurden. Diese sehr einfache geometrische Beschreibung der Pflanzen hatte zudem den Vorteil, dass sich die Berechnung der Beleuchtung stark vereinfachte. Die Ergebnisse waren zum Zeitpunkt der Veröffentlichung spektakulär und bildeten einen wichtigen Anteil des Films „The adventures of André and Wally B.“. Interessant ist dabei, dass die finalen Bilder - trotz der starken biologischen und physikalischen Ungenauigkeiten - so realistisch wirken.

Später konnten Qin *et al.*[QNTN03] eine Reihe von sehr beeindruckenden Bilder berechnen, indem sie die vorhandenen Pflanzen in eine Menge von achsenparallelen, texturierten Ebenen zerlegten. Weiterhin wurden einem Baum einige Schattenbilder und ein Volumengitter zugeordnet, auf deren Basis es nun möglich wird die Bäume sehr realistisch zu beleuchten. Grundsätzlich werden dabei die Pflanzen jedoch unab-

hängig von einander betrachtet, lediglich zur Bestimmung der Beschattung wird auf die Umgebung zugegriffen. Obwohl dieser Ansatz hochqualitative Ergebnisse liefert, sind die Berechnungszeiten doch weit von einer Echtzeitdarstellung entfernt.

Um bei der Visualisierung von größeren Landschaften eine hohe Performanz zu erreichen, sollte die geometrische und algorithmische Komplexität so weit wie möglich gesenkt werden. Deswegen stellen die - in den letzten Abschnitten vorgestellten - Billboardpflanzen zusammen mit einer Spherical Harmonics Beleuchtungstextur die ideale Voraussetzung für die Darstellung solcher Szenen dar. Denn die geometrische Komplexität wird durch die Verwendung der Billboards stark verringert und der Aufwand für die realistische Beleuchtung ist - auf Grund der Spherical Harmonics Transformation - ebenfalls sehr gering. Allerdings existieren noch zwei offene Probleme, die zunächst gelöst werden müssen. Zum einen gilt es einen Weg zu finden, die gegenseitige Beschattung der Bäume mit in die Darstellung zu integrieren. Außerdem ist noch die Frage offen, wie sich verschiedene Level of Detail Repräsentationen eines Modells ineinander überblenden lassen.

Bevor für die genannten Probleme jeweils eine Lösung präsentiert werden kann, muss zunächst jedoch ein genauerer Blick auf Darstellung mehrerer - in einer Landschaft verteilter - Billboardpflanzen geworfen werden. Dies erweist sich nämlich im Kontext einer Hardwareunterstützten Visualisierung als schwierig. Das Problem besteht vor allem darin, dass die einzelnen Billboards während ihrer Darstellung eine Alphamaske benutzen. Daraus folgt jedoch, dass erst nach der Bestimmung der Farbe und Transparenz eines Pixels durch ein Fragment Programm feststeht, ob das Ergebnis überhaupt sichtbar ist oder nicht. Da im Rahmen der Echtzeit-Computergrafik im Normalfall der Z-Puffer die Entscheidung trifft, ob ein Fragment sichtbar ist, dass heißt von keinem anderen Objekt verdeckt wird, müssten alle vorhandenen Billboard Ebenen sortiert vorliegen. Denn nur durch die Verwendung von einer „Hinten-nach-Vorne“ Sortierung lässt sich sicherstellen, dass die entferntesten Objekte zuerst gezeichnet werden, um dann anschließend eventuell von näheren Modellen verdeckt zu werden. Der große Nachteil dieser Vorgehensweise besteht darin, dass die Sortierung nur durch die CPU durchgeführt werden kann, was jedoch zu einer sehr niedrigen Performanz führt, weil die Ergebnisse der Sortierung ständig über den langsamen Datenbus zur Grafikkarte übertragen werden müssen. Im schlechtesten Fall, wenn sich nämlich der Betrachter ununterbrochen bewegt, müsste die Sortierung für

jeden Frame erneut erfolgen. Außerdem bereiten überlappende Ebenen große Probleme, weil hierbei überhaupt keine sinnvolle Reihenfolge mehr angegeben werden kann. Die Billboardpflanzen setzen aber gerade auf eine solche Anordnung, um ein möglichst voluminöses Aussehen zu erzielen. Die Sichtbarkeitsbestimmung kann also nicht ausschließlich durch die Verwendung des Z-Puffers erfolgen. Auf Grund dieser Problematik, bieten die Grafikkarten einen so genannten Alphatest an. Dieser wird nach der Berechnung der Pixelfarbe, aber vor dem Verdeckungstest des Z-Puffers durchgeführt. Dabei werden alle Fragmente verworfen, die einen angegebenen Alphareferenzwert unterschreiten. Daraus folgt jedoch wiederum, dass keine Sortierung der einzelnen Billboardebenen mehr nötig ist, denn es werden nur die potentiell sichtbaren Pixel zum Z-Puffer geschickt. Dadurch muss die Grafikkarte zwar einen Test mehr pro berechneten Fragment evaluieren, die verlorene Performanz wird aber vollkommen durch den Verzicht einer CPU-basierten Sortierung ausgeglichen. Somit können die Pflanzen in einer beliebigen Reihenfolge gerastert werden, wobei sich die Anzahl der Datenübertragungen (Contextswitches) vom Hauptspeicher zum Arbeitsspeicher der Grafikkarte minimieren lässt. Wie weiter oben bereits erläutert wurde, ist diese Eigenschaft für eine hohe Performanz sehr wichtig.



Abbildung 4.15: Farbtextur und Level of Detail - Blendingtextur

Die Verwendung des Alphatests bietet einen weiteren großen Vorteil, denn der Effekt lässt sich nämlich gezielt bei der Überblendung verschiedener Level of Detail Stufen ausnutzen. Dadurch wird es dann letztendlich möglich, die Anzahl der gerenderten Dreiecke in Bezug auf die Distanz zum Betrachter zu regulieren. Je weiter eine Pflanze vom Betrachter entfernt ist, desto weniger Billboards werden für ihre Dar-

stellung benötigt, wobei sich das - oben bereits erwähnte - Problem der Umschaltartefakte durch einen sanften Übergang lösen lässt. Dabei wird das detailliertere Modell schrittweise ein- beziehungsweise ausgeblendet. Dieser Effekt lässt sich zum Beispiel dadurch realisieren, dass sowohl der Maskierungswert der Billboardtextur als auch der Referenzwert des Alphatests „geschickt“ gewählt werden, wodurch langsam immer mehr Elemente (Blätter) der Pflanze sichtbar bzw. versteckt werden. Dabei wäre es gut, wenn das detailliertere Blattwerk von innen nach außen „wachsen“ würde, denn dieses Vorgehen garantiert, dass zunächst nur die dichtesten Stellen der Billboards angezeigt werden. Danach sollten langsam immer mehr Elemente zu sehen sein. Dafür benötigt die Alphamaske der einzelnen Billboards jedoch eine spezielle Struktur, wobei das Zentrum den höchsten Alphawert besitzen muss, damit es den Test als erstes besteht. Die benachbarten Elemente erhalten jeweils einen geringeren Maskierungswert, der in Abhängigkeit vom Abstand zum Zentrum ermittelt wird. Somit werden die äußersten Billboardelemente zuletzt eingeblendet. In Abbildung 4.15 ist ein Vergleich zwischen einer Standardtextur (Links) und einer solchen Blendingtextur (Rechts) zu sehen, dabei ist es deutlich zu erkennen, dass weit vom Zentrum entfernte Pixel transparenter sind als die näheren Elemente.

Um das Blending überhaupt erst zu ermöglichen, müsste der Referenzwert des Alphatests gesteuert werden. Je größer nämlich die Distanz eines Billboards zum Betrachter ist, desto höher müsste auch der Wert dieser Schwelle sein, um mehr Elemente eines Billboards auszublenden. Bewegt sich der Beobachter im Gegensatz dazu näher an ein vereinfachtes Modell heran, dann ist der Referenzwert irgendwann niedrig genug, so dass weitere Einzelteile der Pflanze angezeigt werden. Leider lässt sich jedoch die Alphaschwelle nicht durch einen Shader modifizieren, denn dies kann nur durch die - von der CPU ausgeführte - Applikation geschehen. Da jedes Billboard jedoch potentiell über eine eigene Alphaschwelle verfügt, müssten sie in diesem Fall einzeln gerendert werden. Dieses Vorgehen ist allerdings aus Gründen der Performance nicht möglich, denn um die Anzahl der Contextswitches zu minimieren sollte immer mindestens eine gesamte Pflanze verarbeitet und dargestellt werden. Dieser Zwiespalt lässt sich jedoch auch anders lösen, denn anstatt den Referenzwert selbst zu steuern können auch die Alphawerte der Fragmente modifiziert werden. Es ist schließlich egal, welche Seite der Ungleichung des Alphatests verändert wird. Ob nun mit sinkender Distanz zum Betrachter der Referenzwert geringer oder der Alphawert

der Fragmente erhöht wird, liefert letztendlich das äquivalente Ergebnis. Der Vorteil dieser alternativen Herangehensweise liegt jedoch darin, dass sich der ausgegebene Alphawert durch einen Pixel-Shader steuern lässt, wodurch die Überblendung verschiedener Level of Detail Stufen vollständig von der Grafikkarte ausgeführt werden kann. Diese Vorgehensweise garantiert somit eine optimale Performanz. Abbildung 4.16 zeigt die Überblendung von einem einfachen Kreuzbillboard (Links) zu einer höher aufgelösten Version (Rechts), wobei die beschriebenen Alphamodifizierung eingesetzt wurde.



Abbildung 4.16: Überblendung zweier Level of Detail Stufen

Da die Beleuchtung der einzelnen Pflanzen mit Hilfe der Spherical Harmonics Textur durchgeführt wird, sind noch keine Objektübergreifenden (globalen) Beleuchtungseffekte enthalten, denn bei der Berechnung der ursprünglichen Basiskoeffizienten wurde nur das betrachtete Modell mit einbezogen. Dabei wird davon ausgegangen, dass sich in der Umgebung der Pflanze keine weiteren Objekte befinden. Soll jedoch ein Wald dargestellt werden, dann trifft diese Voraussetzung nicht mehr zu, denn benachbarte Pflanzen können zu einer Beschattung des betrachteten Objekts führen. Eine Möglichkeit dieses Problem zu lösen besteht darin, bei der Berechnung der Basiskoeffizientenvektoren auch die anderen Modelle der Umgebung mit einzubeziehen. Dadurch würde aber jede Pflanze ihre eigene Beleuchtungstextur besitzen, denn die Beschreibung der Umgebung ist ortsabhängig und somit von Objekt zu Objekt unterschiedlich. Das würde jedoch bedeuten, dass für jede Pflanze eine neue Textur

in den Arbeitsspeicher der Grafikkarte geladen werden müsste, wodurch die Performance der Darstellung sehr stark sinken würde. Unter der Voraussetzung, dass sich die Pflanzen in einer natürlichen Umgebung befinden, lässt sich noch ein anderer Lösungsweg einschlagen, denn ein „normaler“ Wald wird nämlich in der Regel nur von zwei verschiedenen Lichtquellen beleuchtet: Der Sonne und der Atmosphäre.

Die Sonne gilt dabei als die wichtigste Lichtquelle, denn an einem unbewölkten Tag liefert sie 80% der Gesamthelligkeit, wobei ihr Licht in Form einer fokussierten, parallelen Bestrahlung auf die Erde trifft. Aus diesen Eigenschaften folgt, dass der dadurch hervorgerufene Schatten sehr hart ist. Wird also davon ausgegangen, dass die Sonne die Hauptlichtquelle der darzustellenden Szene ist, dann lässt sich der fehlende Schatten durch die folgende Approximation nachbilden: Dafür muss zunächst eine Shadow Map (vgl. Abschnitt 4.2.2) entlang der Einfallsrichtung des Sonnenlichtes berechnet werden. Dadurch lässt sich nun entsprechend der Position der darzustellenden Fragmente die Beschattung durch andere Pflanzen bestimmen. Dieser Wert wird anschließend zur Abschwächung der Spherical Harmonics Beleuchtung benutzt, wobei dieses Vorgehen den Schatten der Sonne tatsächlich relativ realistisch nachbildet. Allerdings werden durch diese Approximation keine weiteren globalen Beleuchtungseffekte beschrieben, so fehlt zum Beispiel das zwischen den Pflanzen reflektierte Licht vollständig. Dieser (indirekte) Beleuchtungsanteil lässt sich jedoch relativ gut durch einen kleinen ambienten Faktor nachbilden.

Die zweite wichtige Lichtquelle in natürlichen Szenen stellt die Atmosphäre dar, die eine indirekte und sehr diffuse Beleuchtung spendet. Dabei handelt sich eigentlich um Photonen die hauptsächlich von der Sonne eingestrahlt und dann aber einige Male in der Atmosphäre gestreut wurden. Das Licht wird daher aus keiner bevorzugten Richtung eingestrahlt, denn es scheint vielmehr aus jeder Richtung gleichzeitig zu kommen. Dadurch sind die Schatten des Atmosphärenlichts weich und lokal sehr begrenzt. Da für die Spherical Harmonics Transformation aber von vorne herein nur wenige Bänder benutzt wurden, approximiert die vorhandene Lösung diesen weichen Beleuchtungseffekt bereits hinreichend gut, wobei sich in der Praxis gezeigt hat, dass diesbezüglich keine weiteren Approximationen nötig sind.

Da die zwei genannten Probleme somit gelöst werden konnten, kann nun das Visualisierungssystem zur Darstellung von komplexen Landschaften zusammengesetzt wer-

den. Unter der Voraussetzung, dass der Untergrund bereits vorhanden ist können die einzelnen Pflanzen in der Landschaft platziert werden. Hierbei lassen sich sowohl korrekte biologische Wachstumssimulationen ([Deu03]), als auch einfachere Verteilungsalgorithmen benutzen. Wird der letztere Fall verwendet, so sollte jedoch darauf geachtet werden, dass die Objekte gleichmäßig in einer natürlichen Art und Weise platziert werden. Eine solche Verteilung kann sehr gut durch eine Halton (Abb. 4.17 (b)) oder Hammersley (Abb. 4.17 (c)) Punktmenge ([Hal64], [HH64], [Kel01]) nachgebildet werden.

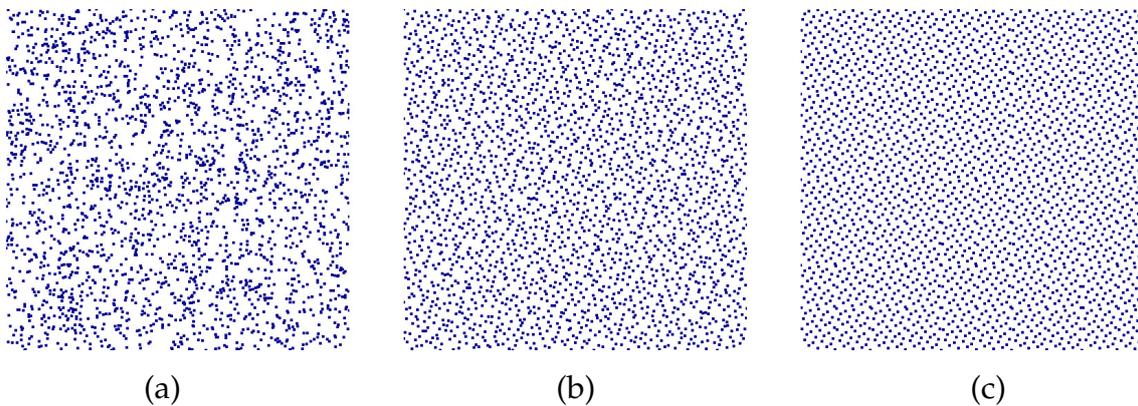


Abbildung 4.17: Punktmenge: Zufällig (a), Halton (b), Hammersley (c)

Um eine hohe Bildberechnungsrate zu erreichen, sollte die Anzahl der gerenderten Dreiecke so klein wie möglich gehalten werden. Deswegen werden verschiedene Level of Detail Stufen verwendet, die in Abhängigkeit der Distanz zum Betrachter ein- bzw. ausgeblendet werden. In der Praxis hat sich gezeigt, dass die Pflanzen im Hintergrund durch eine Kreuzebenenrepräsentation hinreichend gut approximiert werden. Im Nahbereich reicht häufig ein höher aufgelöstes Billboardmodell aus, um die Illusion einer echten Pflanze zu erzeugen, wobei sich die Abweichungen in der Darstellung nur bei einer sehr genauen Betrachtung erkennen lassen (siehe Abb. 4.18).

Die beiden oben genannten Lichtquellen (Sonne, Atmosphärenlicht) können entweder unabhängig voneinander oder zusammen in den alternativen Funktionenraum transformiert werden. Die erstere Vorgehensweise ermöglicht einen zusätzlichen Farbverlauf des Atmosphärenlichts und eine Tageszeit-abhängige Färbung der eingestrahelten Photonen der Sonne. Außerdem muss dabei die Shadow Map immer nur dann

#### 4 Echtzeitvisualisierung von Pflanzen

---

neu berechnet werden, wenn sich entweder die Lichtsituation (Tageszeitwechsel, Bewölkung, usw.) oder die Szene ändert. Sind alle Vorbereitungen getroffen, so kann die Landschaft (siehe Abb. 4.18) endlich dargestellt werden.

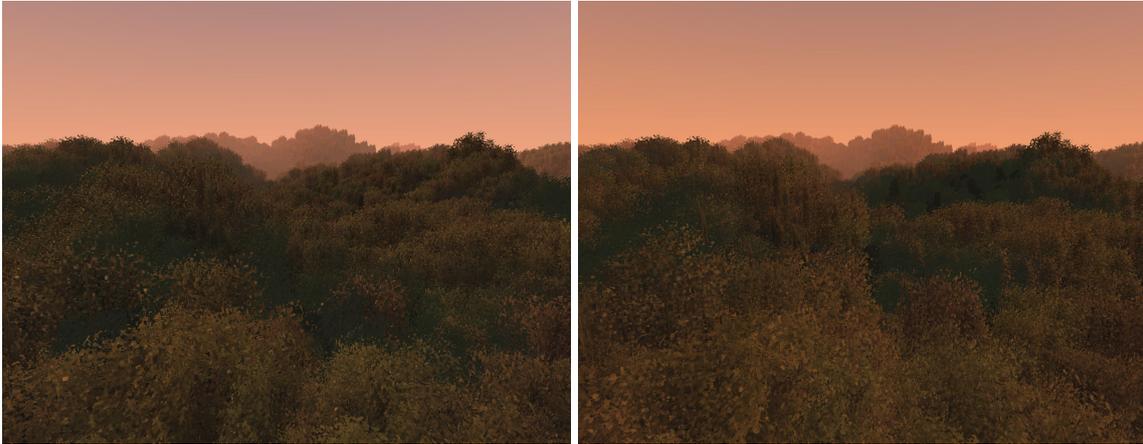


Abbildung 4.18: Auf- und Untergang der Sonne in einem Wald (4-21 FpS)

## 4.4 Bewertung der Ergebnisse

In diesem Kapitel wurden zwei neue Ansätze zur Echtzeit-Darstellung von Pflanzen vorgestellt, die in verschiedene Anwendungsgebiete einzuordnen sind. Sollen kleinere Pflanzen visualisiert werden, wie zum Beispiel ein einzelner Blumentopf, dann bietet sich die Verwendung der direkten Umsetzung des Offline-Algorithmus an, welcher auf die Gegebenheiten der Grafikkarte angepasst wurde. Dabei kann die Implementierung des Verfahrens mit Hilfe der aktuellen Shader-Technologie erfolgen. Auf Grund der Komplexität des Verfahrens dauert die Berechnung der Bilder zwar länger als mit einem Standardbeleuchtungsverfahren (z.B. Phong Shading), allerdings lassen sich nur mit dieser neuen Vorgehensweise die pflanzenspezifischen optischen Effekte beobachten, wie zum Beispiel die diffuse Transmission.



Abbildung 4.19: Vergleich: Baum (271455) und Billboardrepräsentation (652)

Sollen größere Pflanzen gerendert werden, so ist die Verwendung der Spherical Harmonics Transformation vorteilhaft. Dabei wurde die Lösung der Rendering-Gleichung in einen anderen Funktionenraum transformiert, wodurch eine Echtzeitdarstellung von sehr vielen Beleuchtungseffekte (Selbstbeschattung, Lichttransmission, ...) möglich wurde. Der große Vorteil dieses alternativen Darstellungsverfahrens besteht darin, dass die Berechnungsdauer der Bilder vollkommen unabhängig von der optischen Komplexität der darzustellenden Materialien ist, wobei allerdings vorausgesetzt wird, dass die Reflektionsfunktion und die Beleuchtungsumgebung eine geringe Varianz besitzen.

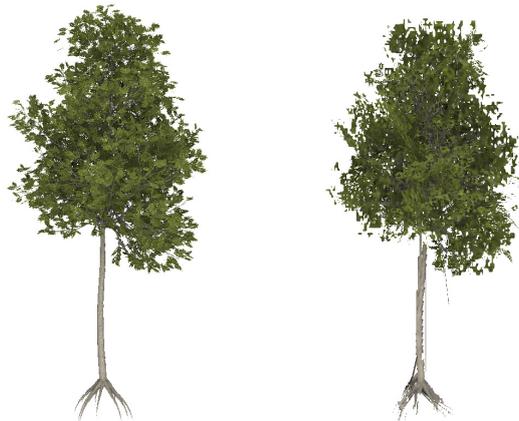


Abbildung 4.20: Vergleich: Baum (88828) und Billboardrepräsentation (160)

Die Spherical Harmonics basierte Visualisierung bietet sich ebenfalls bei der Darstellung von extrem vereinfachten geometrischen Repräsentationen an. In diesem Fall funktioniert nämlich die direkte Umsetzung des Offline-Algorithmus ebenso wenig wie ein Standardbeleuchtungsverfahren, weil Informationen über die darzustellende Oberfläche bei der geometrischen Vereinfachung verloren gegangen sind.



Abbildung 4.21: Vergleich: Sonnenblume (9336) und Billboardrepräsentation (248)

Da die Beleuchtungstexturen für die Billboardrepräsentationen auf Basis der ursprünglichen Geometrie bestimmt wurden, wird es mit dem vorgeschlagenen Verfahren nun möglich, die extrem vereinfachten Pflanzen sehr realistisch darzustellen. Die Bilder in diesem Abschnitt zeigen den Vergleich zwischen den Originalmodellen und den Billboardpflanzen, dabei wird in der Bildbeschreibung jeweils die Anzahl der benutzten Dreiecke angegeben.

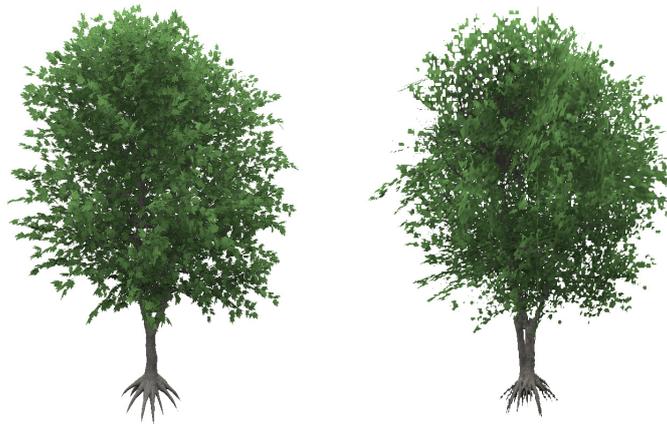


Abbildung 4.22: Vergleich: Baum (407240) und Billboardrepräsentation (120)

Die optische Abweichung zwischen den Repräsentationen folgt aus den Ungenauigkeiten des geometrischen Vereinfachungsalgorithmus und aus der begrenzten Texturauflösung. Trotzdem ist es erstaunlich, dass die extrem vereinfachten Modelle den Originalbeschreibungen so ähnlich sehen.

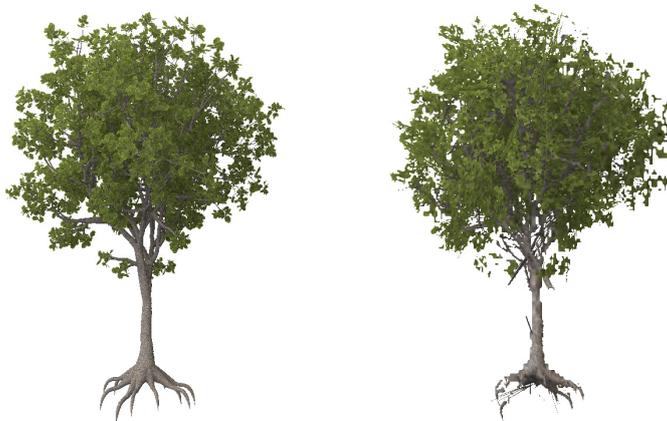


Abbildung 4.23: Vergleich: Baum (166603) und Billboardrepräsentation (388)

Mit Hilfe des neuen Beleuchtungsverfahrens, lassen sich nun auch komplexe Landschaften in einer dynamischen Beleuchtungsumgebung in Echtzeit visualisieren. Abbildung 4.24 zeigt zwei Szenen zu unterschiedlichen Tageszeiten.

## 4 Echtzeitvisualisierung von Pflanzen

---

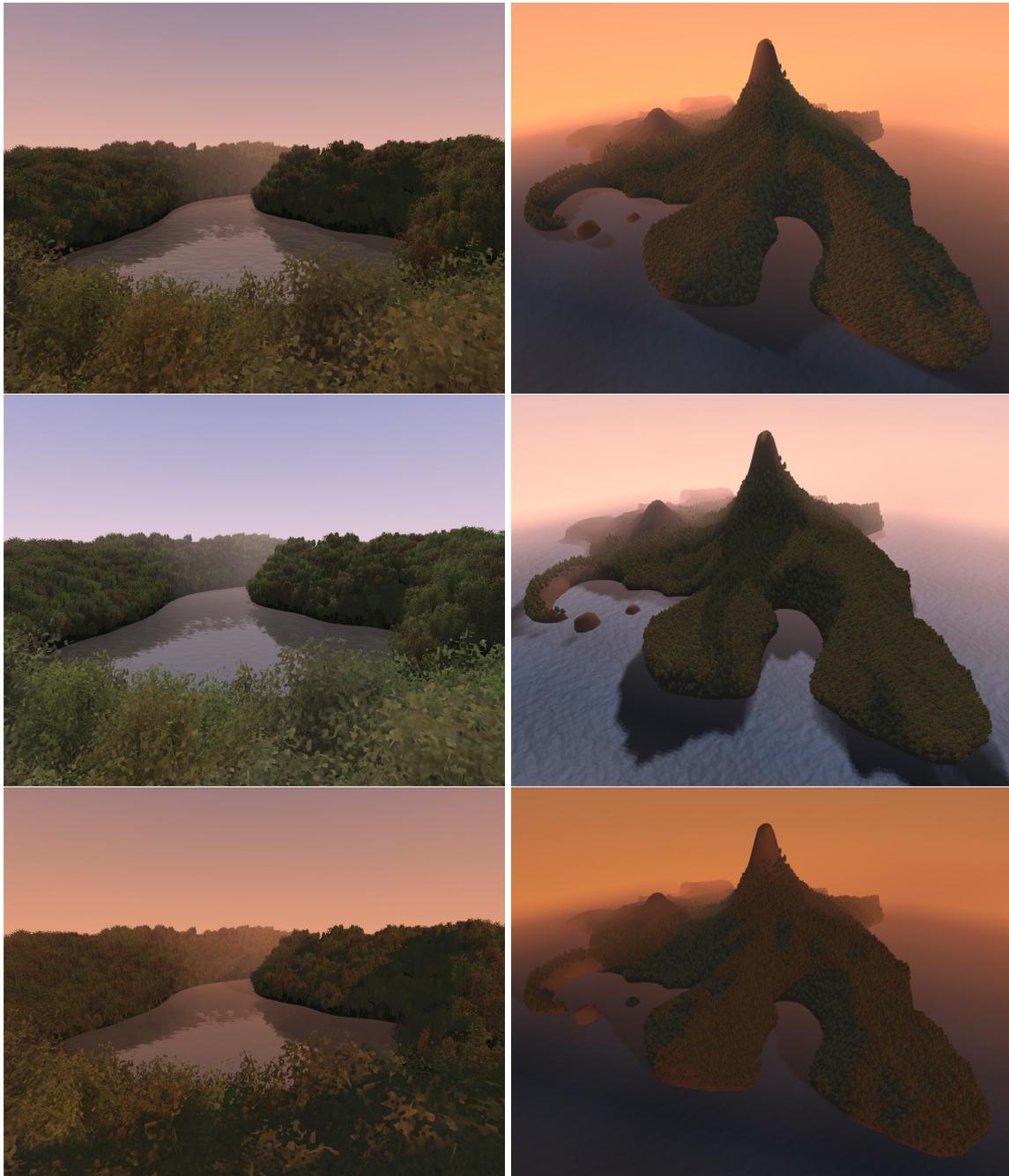


Abbildung 4.24: Links: 6600 Bäume (5-20 FpS), Rechts: 21300 Bäume (3-20 FpS)

Die Spherical Harmonics Textur der Billboardmodelle ermöglicht es unterschiedliche Level of Detail Repräsentationen einer Pflanze in der selben Szene zu verwenden, dabei erfolgt die Überblendung zwischen den Modellen mit minimalen visuellen Artefakten, da alle Objektversionen ein kohärentes Beleuchtungsmodell benutzen.



Abbildung 4.25: Flusstal: 28200 Bäume (3-20 FpS)

### 4.5 Zusammenfassung und Ausblick

In diesem Kapitel stand die Echtzeitdarstellung von Pflanzen im Betrachtungsschwerpunkt. Dabei wurde ausgehend von dem - zuvor vorgestellten - photorealistischen Visualisierungsalgorithmus für Pflanzen eine Menge von Vereinfachungen eingeführt, welche die Berechnungsdauer der Bilder erheblich senken. Je nach Anwendungsgebiet kann zwischen zwei Echtzeitverfahren gewählt werden. Für kleinere Pflanzen, die sehr genau beobachtet werden sollen, bietet sich die Verwendung der direkten Umsetzung des Offline-Darstellungsalgorithmus an. Im Gegensatz dazu werden bei der Visualisierung von größeren Landschaften stark vereinfachte Pflanzenmodelle benötigt, um die geometrische Komplexität der Szene zu reduzieren, wodurch sich jedoch herkömmlichen Darstellungsverfahren nicht mehr anwenden lassen. Deswegen wurde im Rahmen dieses Kapitels eine weitere Vorgehensweise eingeführt, welche die Rendering-Gleichung auf Basis einer Menge von vorberechneten Werten löst. Dadurch lassen sich auch die stark vereinfachten Pflanzenrepräsentationen realistisch

darstellen, wodurch es schließlich möglich wird natürliche Szenen in Echtzeit zu rendern, die Tausende von Pflanzen enthalten.

Ein verbessertes geometrisches Vereinfachungsverfahren und eine optimierte Kodierung der Beleuchtungstextur könnte die erreichte Qualität sicherlich weiter steigern. Dabei sollten vor allem die Anteile der Pflanze durch eine alternative Form nachgebildet werden, welche sich nur schlecht durch Billboards repräsentieren lassen. Zum Beispiel könnte der Hauptstamm eines Baums durch einen niedrig aufgelösten Zylinder approximiert werden.

Die Visualisierung von noch größeren Landschaften könnte dadurch erreicht werden, dass die geometrische Komplexität der einzelnen Modelle weiter gesenkt wird. Dazu hat Johannes Kopf ([Kop04]) ein Verfahren vorgeschlagen, welches die Pflanzen in Form von mehreren Landschafts-parallelen Oberflächen beschreibt. Diese Methode sollte sich sehr gut mit dem in diesem Kapitel präsentierten Beleuchtungsmodell kombinieren lassen.



Abbildung 4.26: Das Flusstal

# 5 Zusammenfassung und Ausblick

## 5.1 Resümee

Die realistische Darstellung von künstlichen Pflanzen unterscheidet sich in vielerlei Hinsicht von Standard-Visualisierungsverfahren. Dies beginnt bereits bei der Erstellung der geometrischen Repräsentationen, denn im Gegensatz zu anderen Objekten, die ein Künstler fast immer aus einzelnen Primitiven zusammensetzt, werden bei der Erstellung von künstlichen Pflanzen sehr häufig prozedurale und regelbasierte Modellierungsmethoden verwendet. Dieser Unterschied folgt vor allem daraus, dass Pflanzen in der Regel aus vielen kleinen Elementen zusammengesetzt sind, wodurch die erstellten Modelle eine sehr hohe geometrische Komplexität besitzen. Dabei ist es nicht möglich, dass ein Künstler jedes winzige Detail manuell bearbeitet. Damit sich die modellierten Pflanzen später überhaupt darstellen lassen, werden die einzelnen Bestandteile dieser Objekte häufig durch eine vereinfachte geometrische Beschreibung repräsentiert. Dabei besitzen zum Beispiel die Blätter eines virtuellen Baums kein Volumen und werden oft lediglich durch nur eine endliche Ebene dargestellt. Diese Abweichungen von der Realität müssen auch von dem verwendeten Visualisierungsverfahren berücksichtigt werden, da andernfalls keine realistischen Bilder erzeugt werden können. Da die Standard-Darstellungsalgorithmen jedoch unter Voraussetzung entwickelt wurden, dass die Objekte ein möglichst realistisches geometrisches Modell besitzen, lassen sie sich in einem solchen Fall nur sehr eingeschränkt benutzen. Zusätzlich dazu finden bei diesen Verfahren die biologischen und optischen Eigenschaften des organischen Gewebes keine Berücksichtigung, wodurch sich jedoch auch keine realistische Darstellung erzeugen lässt. Deswegen behandelte diese Diplomarbeit die realistische Visualisierung von künstlichen Pflanzen, wobei dieses Thema sowohl im Kontext des Offlinerendings (Filmproduktion) als auch im

Bereich der Echtzeitdarstellung (Virtual Reality) betrachtet wurde. In diesen beiden Anwendungsgebieten konnten erfolgreich neue Algorithmen entwickelt und implementiert werden, wodurch eine photorealistische Darstellung von komplexen Pflanzenmodellen ermöglicht wurde.

Das Ziel des Offlinerenderings besteht darin möglichst hochqualitative Bilder zu berechnen, wobei die photorealistische Wirkung der dargestellten Szene eine größere Priorität besitzt, als eine niedrige Berechnungsdauer der Bilder. Trotzdem kann die Komplexität eines Darstellungsverfahrens auch in diesem Bereich nicht vollkommen ignoriert werden, denn die hochaufgelösten Ergebnisbilder müssen nach einer „angemessenen“ Zeitspanne fertig gestellt sein. Nur so lassen sich computergenerierte Filme bzw. Filmsequenzen erzeugen, denn in diesem Fall müssen viele Tausend Einzelbilder erstellt werden. Aus diesem Grund wurde im Rahmen dieser Diplomarbeit ein empirisches Darstellungsmodell vorgeschlagen, welches die schnelle und biologisch motivierte Visualisierung von lichtdurchlässigem Pflanzengewebe ermöglicht. Dabei lag sowohl die realistische Wirkung der dargestellten Objekte als auch die hohe Effizienz des Verfahrens im Fokus der Betrachtung. Um diese beiden Ziele zu erfüllen wurde absichtlich auf eine vollständige biophysikalische Lichtsimulation verzichtet, wobei die Approximation des optischen Verhaltens des visualisierten Pflanzengewebes durch die Verwendung von verschiedenen Texturen erfolgt. Dabei wurde insbesondere auch auf eine möglichst hohe Parametrisierbarkeit und Flexibilität des Darstellungsverfahrens geachtet, wobei die Verwendung von geometrisch vereinfachten Modellen ebenfalls Berücksichtigung fand. Die mit diesem Verfahren erstellten Ergebnisbilder zeigen, dass der präsentierte Algorithmus die geforderten Ziele erfüllt und eine photorealistische Darstellung von Pflanzen ermöglicht.

Steht die Echtzeitvisualisierung komplexer Landschaften im Vordergrund, dann verändern sich die zu erreichenden Ziele sehr stark, denn in diesem Kontext ist es sehr wichtig eine minimale Berechnungsdauer zu garantieren. Nur so lässt sich die Illusion einer flüssigen Bewegung erzeugen, wobei eine entsprechend hohe Bildberechnungsrate jedoch nur auf Basis von Vereinfachungen ermöglicht wird. Dabei sorgt vor allem die große geometrische Komplexität der Pflanzenmodelle für Performanzprobleme während der Visualisierung. Aus diesem Grund werden die Pflanzen durch eine extrem vereinfachte Repräsentation ersetzt. So besteht zum Beispiel das Blattwerk eines Baums in diesem Fall lediglich aus einer geringen Anzahl von endlichen

Ebenen. Die Verwendung solcher stark vereinfachten Modelle besitzt auch Auswirkungen auf das Darstellungsverfahren, denn durch die geometrischen Vereinfachungen gehen Informationen über die darzustellende Oberfläche verloren. Deswegen muss der Visualisierungsalgorithmus auch ohne diese Daten funktionieren und realistische Ergebnisse liefern. Zu diesem Zweck wurde im Rahmen dieser Diplomarbeit eine neue Vorgehensweise eingeführt, welche die Darstellung von Pflanzen mit Hilfe von vorberechneten Werten durchführt, wodurch eine sehr effiziente Visualisierung erreicht wurde. Dabei können sehr viele optische Effekte berücksichtigt werden ohne dadurch einen Einfluss auf die Laufzeitperformanz auszuüben. Allerdings wird dieser Vorteil nur unter der Voraussetzung erreicht, dass sich die Szenenobjekte in einer diffusen Beleuchtungsumgebung befinden.

Die kohärente Beleuchtung von Pflanzen in verschiedenen geometrischen Auflösungen stellt die Voraussetzung für einen Level of Detail Ansatz dar. Dabei wird die begrenzte örtliche Auflösung des menschlichen Auges ausgenutzt, indem weit entfernte Objekte mit weniger Details dargestellt werden. Bei der Echtzeitvisualisierung von Landschaften können Pflanzen, die eine große Distanz zum Betrachter besitzen, durch eine sehr einfache geometrische Repräsentation ersetzt werden. Häufig werden dafür nur ein oder zwei endliche Ebenen verwendet, die mit Texturen versehen sind. Je näher der Beobachter den Modellen kommt, desto mehr Details werden hinzugefügt, wobei auch diese Überblendung zwischen unterschiedlichen Detailstufen in dieser Arbeit analysiert wurde. Damit in einem solchen Anwendungskontext überhaupt eine realistische Darstellung von Pflanzen durchgeführt werden kann, wird insbesondere vorausgesetzt, dass die Modelle aller Detailstufen unabhängig von ihrer konkreten Topologie in der selben Art und Weise beleuchtet werden. Auf Grund der vorherigen Untersuchungen ließ sich diese Voraussetzung jedoch erfüllen, so dass die Überblendung zwischen verschiedenen Repräsentationsformen einer Pflanze mit minimalen visuellen Artefakten durchgeführt werden kann.

Auf Basis der beschriebenen Algorithmen und Vorgehensweisen konnte anschließend sogar gezeigt werden, dass sich damit nun auch komplexe, natürliche Landschaften in einer dynamischen Beleuchtungsumgebung in Echtzeit visualisieren lassen. Die Funktionstüchtigkeit des beschriebenen Verfahrens, wurde anhand von verschiedenen Szenen demonstriert, welche mit Tausenden von Pflanzen bedeckt waren.

Die erreichten Ergebnisse zeigen, dass die angestrebten Ziele dieser Diplomarbeit erfüllt werden konnten, wobei die präsentierten Verfahren erfolgreich in der Praxis umgesetzt und ausgiebig getestet wurden. Die zu dieser Ausarbeitung zugehörige CD beinhaltet eine umfangreiche Sammlung von Ergebnissen und die Dokumentation der Implementierungen.

### 5.2 Weiterführende Ansätze

Da in den jeweiligen Kapiteln bereits einige Anmerkungen zu konkreten Erweiterungen gemacht wurden, soll an dieser Stelle die Pflanzendarstellung im Allgemeinen betrachtet werden. Dabei stellt die Berücksichtigung aller globalen Beleuchtungseffekte immer noch ein großes Problem dar, denn in der Realität existieren viele Lichtinteraktionen, die sich nur mit einem sehr großen Aufwand nachbilden lassen, wobei die geometrische Komplexität von Pflanzenmodellen das Problem noch weiter verschärft. Aus diesem Grund wäre eine Untersuchung lohnenswert, ob sich in diesem Kontext Volumenrenderingansätze sinnvoll einsetzen lassen. Zum Beispiel könnte die eingehende Radianz auf Basis eines Gitters bestimmt werden, welches die Pflanze umgibt, wobei der Vorteil dieses Verfahrens darin liegt, dass sich die Auflösung des Beleuchtungsgitters beliebig variieren lässt, wodurch je nach geforderter Qualität mehr oder weniger Testpunkte berücksichtigt werden können. Interessant ist in diesem Fall, ob sich trotz der starken Abweichungen realistische Bilder berechnen lassen, denn wäre dies möglich, dann ließe sich dieser Ansatz auch bei der Vorberechnung der Spherical Harmonics Basiskoeffizienten nutzen, wodurch die Effizienz der Vorverarbeitung sehr stark steigen würde.

Weiterführende Untersuchungen könnten auch im Bereich des „Nicht - Photorealistischen Renderings“ (NPR) erfolgen. Diese Darstellungsart visualisiert Objekte gezielt in einer abstrakteren Art und Weise, um zum Beispiel eine verbesserte schematische Übersicht zu ermöglichen. Dabei stellt die Architekturvisualisierung ein wichtiges Anwendungsgebiet des NPR dar. Es existieren zwar bereits schon einige Arbeiten ([WS94], [DS00]) in diesem Gebiet, allerdings wäre eine weitergehende Forschung in Bezug auf die pflanzenspezifischen Beleuchtungseffekte sicherlich interessant. Außerdem gilt es zu untersuchen, ob eine NPR Darstellungsform auch auf Basis der

geometrisch extrem vereinfachten Modelle funktioniert. Darauf aufbauend könnte dann die Visualisierung in einem Level of Detail Kontext betrachtet werden.

Auch in den nächsten Jahren wird die Forschung sicherlich neue Impulse liefern, welche sich aufgreifen und für die Pflanzendarstellung verwenden lassen. Dabei gibt es noch viele Probleme zu überwinden und Optimierungen zu entwickeln, bis dieses Thema in jeder Hinsicht zufrieden stellend behandelt wurde.

### 5.3 Schlusswort

Die Computergrafik ist sicherlich eines der interessantesten und schönsten Forschungsgebiete der Informatik, denn sind erst einmal alle mathematischen und technischen Probleme gemeistert, dann offenbart sich das Ergebnis der Bemühungen häufig als ein faszinierendes computergeneriertes Bild. Neben rein wissenschaftlichen Anwendungen, darf auf keinen Fall die Unterhaltungsindustrie als wichtiger Nutzer vergessen werden, denn in der modernen Informations- und Mediengesellschaft hat sich dieser Wirtschaftszweig zu einer der größten Industrien entwickelt. Die Tage der „Garagenunternehmen“ sind längst vorbei und Begriffe wie Projekt- und Risikomanagement spielen mittlerweile eine große Rolle. Deswegen sind Investitionen mit einem Umfang von mehreren Millionen Euro schon lange keine Ausnahme mehr in der Filmproduktion oder der Spieleentwicklung. Somit wird auch der Ruf nach Spezialisten lauter, wobei auch die Ausbildungsstätten diese Entwicklung erkannt haben und vermehrt auf Medien-bezogene Schwerpunkte setzen. So bieten immer mehr Fachhochschulen und Universitäten die Studienrichtung Medieninformatik an. Die hier präsentierten Ergebnisse gehören zu einer der ersten Diplomarbeiten in diesem Fachgebiet an der Technischen Universität Dresden. Trotz des geringen Alters dieses Studiengangs zeigt diese Arbeit hoffentlich recht gut, was sich alles in dieser Forschungsrichtung erreichen lässt, denn für einige Probleme, die bei der Visualisierung der Natur auftreten, konnten neue bzw. verbesserte Lösungsmöglichkeiten präsentiert werden. Nun liegt es an den nachfolgenden Jahrgängen, diese Ergebnisse weiterzuentwickeln und somit dem Ziel der photorealistische Darstellung der Welt einen Schritt näher zu kommen.

## 5.4 Danksagung

An dieser Stelle möchte ich mich bei allen Personen bedanken, die mir über die Jahre so viel Unterstützung gegeben haben, denn ohne meine Familie und meine Freunde wäre mein Studium wohl nicht so optimal vorangeschritten. Außerdem gilt dieser Dank auch meinen Arbeits- und Forschungskollegen am Lehrstuhl für Computergrafik, die mir stets mit einem guten Rat und konstruktiver Kritik zur Seite standen.

Bei der Erstellung dieser Diplomarbeit haben mir insbesondere die folgenden Personen auf die eine oder andere Weise stark geholfen: Joscha Metze, Benjamin Neidhold, Prof. Dr. Markus Wacker, Prof. Dr. Oliver Deussen und natürlich Regina und Günter Franzke. Vielen Dank!

# Abbildungsverzeichnis

1.1	Insel Rügen . . . . .	2
2.1	Triangulierte Kugel . . . . .	10
2.2	Rendering-Gleichung . . . . .	12
2.3	Vollständige Rendering-Gleichung . . . . .	13
2.4	Einfluss der Rekursionstiefe . . . . .	15
2.5	Diskrete Abtastung der Hemisphäre . . . . .	16
2.6	Approximation einer Funktion . . . . .	20
2.7	Graph der Legendre Polynome . . . . .	22
2.8	Approximation einer sphärischen Funktion . . . . .	24
2.9	Ausgewählte Pflanzenblätter . . . . .	27
2.10	Querschnitt eines Blattes . . . . .	28
2.11	Lichtinteraktion innerhalb eines Blattes . . . . .	29
2.12	Schematische BSSRDF eines Blattes . . . . .	30
2.13	Reflektion (Links) und diffuse Transmission (rechts) . . . . .	31
3.1	Geometrische Repräsentation eines Blattes . . . . .	35
3.2	Visuelle Grundzustände . . . . .	41
3.3	Virtuelle parallele Ebene . . . . .	42
3.4	Idealisiertes Schichtenmodell . . . . .	43
3.5	Lösung der Rendering-Gleichung in Pflanzengewebe . . . . .	46
3.6	Inkorrekte Refraktion . . . . .	47
3.7	Texturen (von Links: Vorderseite, Rückseite, Durchlicht) . . . . .	50
3.8	Beispieltexturen zweier Pflanzenblätter . . . . .	53
3.9	Einzelnes Blatt - Neuer Algorithmus (600x800; 11,5 Sekunden) . . . . .	56

## Abbildungsverzeichnis

---

3.10	Einzelnes Blatt - Standardalgorithmus (600x800; 11 Sekunden) . . . . .	56
3.11	Zimmerpflanze (800x800; 100 Sekunden) . . . . .	57
3.12	Zimmerpflanze (800x800; 150 Sekunden) . . . . .	58
3.13	Zimmerpflanze (800x800; 124 Sekunden) . . . . .	58
3.14	Zimmerpflanze (Busch) (800x800; 10 Minuten) . . . . .	59
3.15	Der Hof (30 Minuten) . . . . .	60
4.1	Lokaler und globaler Parameterraum der Grafikhardware . . . . .	65
4.2	Vereinfachte Verarbeitungspipeline der Grafikhardware . . . . .	66
4.3	Shadow Map Algorithmus . . . . .	69
4.4	Angepasster Darstellungsalgorithmus (ca. 100 FpS) . . . . .	72
4.5	Spherical Harmonics Transformation der Rendering-Gleichung . . . . .	77
4.6	Spherical Harmonics Darstellung mit einem Vertex Shader (300 FpS) . . . . .	79
4.7	Ein Baum in verschiedenen Beleuchtungssituationen (300 FpS) . . . . .	82
4.8	Approximation eines Astes durch ein Billboard . . . . .	84
4.9	Billboardrepräsentation eines Baums . . . . .	86
4.10	Informationsverlust durch die Projektion . . . . .	86
4.11	Beleuchtung einer Billboardrepräsentation . . . . .	89
4.12	Kreuzebenen Billboards . . . . .	90
4.13	Kreuzbillboardrepräsentation einiger Bäume . . . . .	91
4.14	Texturen eines Kreuzbillboards . . . . .	92
4.15	Farbtextur und Level of Detail - Blendingtextur . . . . .	95
4.16	Überblendung zweier Level of Detail Stufen . . . . .	97
4.17	Punktmengen: Zufällig (a), Halton (b), Hammersley (c) . . . . .	99
4.18	Auf- und Untergang der Sonne in einem Wald (4-21 FpS) . . . . .	100
4.19	Vergleich: Baum (271455) und Billboardrepräsentation (652) . . . . .	101
4.20	Vergleich: Baum (88828) und Billboardrepräsentation (160) . . . . .	102
4.21	Vergleich: Sonnenblume (9336) und Billboardrepräsentation (248) . . . . .	102
4.22	Vergleich: Baum (407240) und Billboardrepräsentation (120) . . . . .	103
4.23	Vergleich: Baum (166603) und Billboardrepräsentation (388) . . . . .	103
4.24	Links: 6600 Bäume (5-20 FpS), Rechts: 21300 Bäume (3-20 FpS) . . . . .	104
4.25	Flusstal: 28200 Bäume (3-20 FpS) . . . . .	105
4.26	Das Flusstal . . . . .	107

# Literaturverzeichnis

- [ati] <http://www.ati.de/>. ATI Technologies Inc.
- [Beh04] Stephan Behrendt. Optimale Billboardanordnung zur Approximation von Baummodellen in der Computergrafik, 2004.
- [BH02] Kurt Binder and Dieter W. Heermann. *Monte Carlo Simulation in Statistical Physics: An Introduction*. Springer Verlag, 2002.
- [Bli77] James F. Blinn. Models of light reflection for computer synthesized pictures. In *SIGGRAPH '77: Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, pages 192–198. ACM Press, 1977.
- [BR97] Gladimir V.G. Baranoski and Jon G. Rokne. An Algorithmic Reflectance and Transmittance Model for Plant Tissue. In *Computer Graphics Forum*, 1997.
- [BR01] Gladimir V.G. Baranoski and Jon G. Rokne. Efficiently simulating Scattering of Light by Leaves. In *The Visual Computer*, 2001.
- [bra] <http://www.splutterfish.com/>. SplutterFish LLC.
- [cg] [http://developer.nvidia.com/page/cg\\_main.html](http://developer.nvidia.com/page/cg_main.html). NVIDIA.
- [CW93] Michael F. Cohen and John R. Wallace. *Radiosity and Realistic Image Synthesis*. Morgan Kaufmann, 1993.
- [DDSD03] Xavier Décoret, Frédo Durand, François X. Sillion, and Julie Dorsey. Billboard clouds for extreme model simplification. volume 22, pages 689–696. ACM Press, 2003.

- [Deu03] Oliver Deussen. *Computergenerierte Pflanzen: Technik und Design digitaler Pflanzenwelten*. Springer Verlag, 2003.
- [DHL<sup>+</sup>98] Oliver Deussen, Pat Hanrahan, Bernd Lintermann, Radomir Měch, Matt Pharr, and Przemyslaw Prusinkiewicz. Realistic modeling and rendering of plant ecosystems. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 275–286. ACM Press, 1998.
- [DL97] Oliver Deussen and Bernd Lintermann. A modelling method and user interface for creating plants. In *Proceedings of the conference on Graphics interface '97*, 1997.
- [DS00] Oliver Deussen and Thomas Strothotte. Computer-generated pen-and-ink illustration of trees. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 13–18. ACM Press/Addison-Wesley Publishing Co., 2000.
- [dx] <http://www.microsoft.com/windows/directx/default.aspx>. Microsoft Corporation.
- [FD03a] Oliver Franzke and Oliver Deussen. Fast and Accurate Graphical Representation of Plant Leaves. In *International Symposium on Plant Growth Model, Simulation, Visualization and Applications*. PMA, 2003.
- [FD03b] Oliver Franzke and Oliver Deussen. Rendering Plant Leaves Faithfully. In *SIGGRAPH '03: Proceedings of the SIGGRAPH 2003 conference on Sketches & applications*. ACM Press, 2003.
- [Fra03] Oliver Franzke. Flexible Raytracerarchitekuren zur Berechnung komplexer optischer Phänomene. Technische Universität Dresden, Großer Beleg, 2003.
- [Fra04] Oliver Franzke. Raytracing. Technische Universität Dresden, Einführungsfolien in das Komplexpraktikum Raytracing, 2004.

- [FvDFH97] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley Publishing Company, second edition in c edition, 1997.
- [GJVU96] Yves M. Govaerts, Stéphane Jacquemoud, Michel M. Verstraete, and Susan L. Ustin. Three-dimensional radiation transfer modeling in a dicotyledon leaf. In *Applied Optics*, 1996.
- [Gla03] Paul Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer Verlag, 2003.
- [Gre03] Robin Green. Spherical Harmonic Lighting: The in Gitty Details. In *Proceedings of Game Developers Conference 2003*, 2003.
- [Hal64] John H. Halton. Algorithm 247: Radical-inverse Quasi-random Point Sequence. *Commun. ACM*, 7(12):701–702, 1964.
- [HH64] John M. Hammersley and David C. Handscomb. *Monte Carlo Methods*, 1964.
- [HK93] Pat Hanrahan and Wolfgang Krüger. Reflection from Layered Surfaces due to Subsurface Scattering. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, 1993.
- [JAD<sup>+</sup>03] Henrik Wann Jensen, James Arvo, Phil Dutre, Alexander Keller, Art Owen, Matt Pharr, and Peter Shirley. Course 44: Monte Carlo Ray Tracing. In *SIGGRAPH '03: Proceedings of the 30th annual conference on Computer graphics and interactive techniques*. ACM Press, 2003.
- [JB02] Henrik Wann Jensen and Juan Buhler. A rapid hierarchical rendering technique for translucent materials. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 576–581. ACM Press, 2002.
- [Jen01] Henrik Wann Jensen. *Realistic Image Synthesis Using Photon Mapping*. A K Peters, Ltd., 2001.

- [JMLH01] Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. A practical model for subsurface light transport. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 511–518. ACM Press, 2001.
- [Kaj86] James T. Kajiya. The Rendering Equation. In *SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150. ACM Press, 1986.
- [Kel01] Alexander Keller. Strictly Deterministic Sampling Methods in Computer Graphics. Technical report, mental images, 2001.
- [Kop04] Johannes Kopf. Texturbasierte Approximation für virtuelle Landschaften, November 2004.
- [men] <http://www.mentalimages.de/>. mental images GmbH.
- [Met05] Joscha Metze. Entwicklung eines Visualisierungssystems für Virtual-Reality Simulationen. 2005.
- [nvi] <http://www.nvidia.de/>. NVIDIA.
- [og] <http://www.opengl.org/>. Silicon Graphics, Inc.
- [Pho73] Bui Tuong Phong. Illumination for computer generated pictures. In *Communications of the ACM*. ACM Press, 1973.
- [PL90] Przemyslaw Prusinkiewicz and Aristid Lindenmayer. *The Algorithmic Beauty of Plants*. Springer Verlag, 1990.
- [QNTN03] Xueying Qin, Eihachiro Nakamae, Katsumi Tadamura, and Yasuo Nagai. Fast Photo-Realistic Rendering of Trees in Daylight. volume 22. EG Press, 2003.
- [RB85] William T. Reeves and Ricki Blau. Approximate and probabilistic algorithms for shading and rendering structured particle systems. In *SIGGRAPH '85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 313–322. ACM Press, 1985.

- [Ree83] William T. Reeves. Particle systems, a technique for modeling a class of fuzzy objects. In *SIGGRAPH '83: Proceedings of the 10th annual conference on Computer graphics and interactive techniques*, pages 359–375. ACM Press, 1983.
- [ren] <http://www.pixar.com/renderman/>. Pixar Animation Studios.
- [SKS02] Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 527–536. ACM Press, 2002.
- [Sob91] Ilya M. Sobol. *Die Monte-Carlo-Methode*. Deutscher Verlag der Wissenschaften, 1991.
- [SSBdR01] Cyril Soler, François Sillion, Frédéric Blaise, and Philippe de Reffye. A physiological plant growth simulation engine based on accurate radiant energy transfer. Technical report, Institut National de Recherche en Informatique et en Automatique, 2001.
- [TG76] Compton J. Tucker and Michael W. Garratt. Leaf optical system modeled as a stochastic process. In *Applied Optics*, 1976.
- [Wil78] Lance Williams. Casting curved shadows on curved surfaces. In *SIGGRAPH '78: Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, pages 270–274. ACM Press, 1978.
- [WS94] Georges Winkenbach and David H. Salesin. Computer-generated pen-and-ink illustration. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 91–100. ACM Press, 1994.
- [WW92] Alan Watt and Mark Watt. *Advanced Animation and Rendering Techniques: Theory and Practice*. Addison Wesley Longman Limited, 1992.
- [xfr] <http://www.greenworks.de/>. greenworks organic-software.