

Advanced Raytracing

Eine Ausarbeitung zum Vortrag im Rahmen des Seminars
„3D-Grafik“ im Wintersemester 2002/2003
am Institut für Informatik
der Humboldt Universität Berlin

von

Marcus Richter

[mrichter@informatik.hu-berlin.de]

und

Dörte Schäfer

[dschaefe@informatik.hu-berlin.de]

Inhaltsverzeichnis

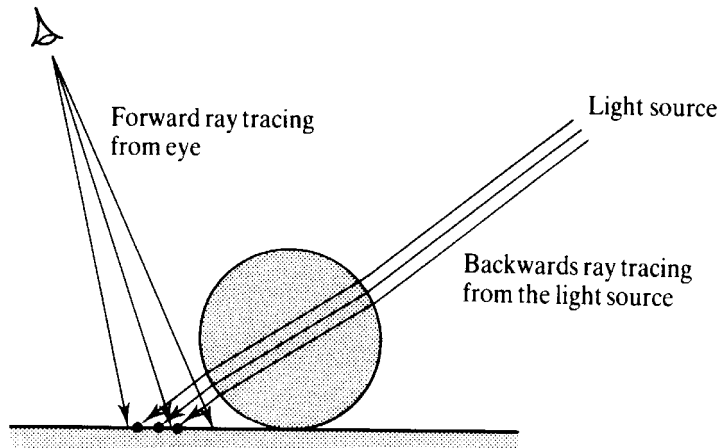
Advanced Raytracing

Backward Raytracing.....	4
Beam Tracing.....	6
Distributed Raytracing.....	8
Grundlagen und Vorteile.....	8
Jitter.....	9
Reflektionsverhalten von Oberflächen.....	10
Nachteile.....	12
Depth of Field (Tiefenschärfe).....	12
(Weiche) Schatten.....	14
Motion Blur (Bewegungsunschärfe).....	15
Globale Beleuchtung.....	17
Radiosity.....	17
Allgemeine Vorgehensweise.....	19
Die Radiosity-Gleichung.....	20
Formfaktoren.....	22
Die Hemicube-Methode.....	23
Steigerung der Genauigkeit der Lösung durch Substructuring.....	26
Zusammenfassung und Fazit.....	27
Globale Beleuchtungsmodelle.....	28
Kombination von Raytracing und Radiosity.....	29
Die erweiterte Two-Pass Methode von Sillion.....	31

Photon Maps.....	32
Das Photon.....	32
Das Leuchtkörper.....	33
Die Projektionskarte.....	34
Russisches Roulette.....	34
Treffer!.....	36
Die Karte bitte.....	36
...und das ist erst der Anfang.....	39
 Literatur	 40

Backward Raytracing

Backward Raytracing oder auch *Light Ray Tracing* geht auf Arvo 1986 zu-

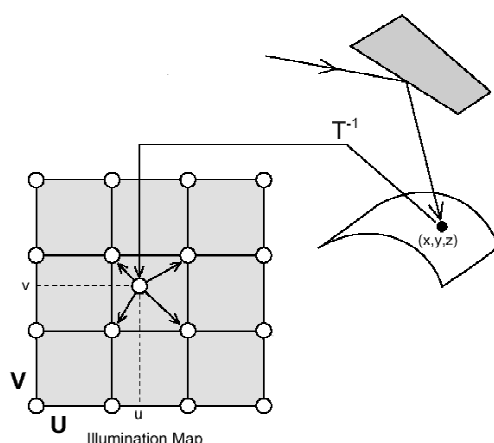


rück und entstand unter der Notwendigkeit indirekte Beleuchtung zu erzeugen, sowie dem Problem der Darstellung von Halbschatten zu begegnen.

In einem ersten Path Tracing werden Strahlen von den Lichtquellen aus in die

Szene geschickt, trifft ein solcher Strahl auf eine zumindest anteilig diffuse Oberfläche wird seine Radiance nach der Reflektion reduziert und der Differenzbetrag der Energie der Oberfläche gutgeschrieben, auf die der Strahl aufgetroffen ist. Diese wird in einer so genannten *Illumination Map* gespeichert; dies ähnelt dem Konzept einer Texture Map.

Im zweiten Schritt wird konventionelles Raytracing durchgeführt, wobei die *Illumination Maps* der Oberflächen in die Beleuchtungsberechnung mit einfließen.



Es wird also ein „Pass“ von jeder Lichtquelle aus und ein weiterer „Pass“ vom Betrachter aus berechnet. Um die Informationen „in der Mitte“, also das Zusammentreffen der berechneten Informationen darzustellen, wird eine *Illumination Map* für jede Oberfläche der Szene berechnet.

Sie besteht aus einem zweidimensionalen Raster von Datenpunkten (skalare Werte für weißes Licht oder RGB Triplets für farbiges Licht).

Über eine Funktion $T(u, v) \rightarrow (x, y, z)$ ist sie mit der Oberfläche eines Objekts verbunden. Die Inverse dieser Funktion stellt die benötigten Informationen über die Orte auf die die Beleuchtungsstrahlen treffen und deren Energie, welche sie beinhalten, dar.

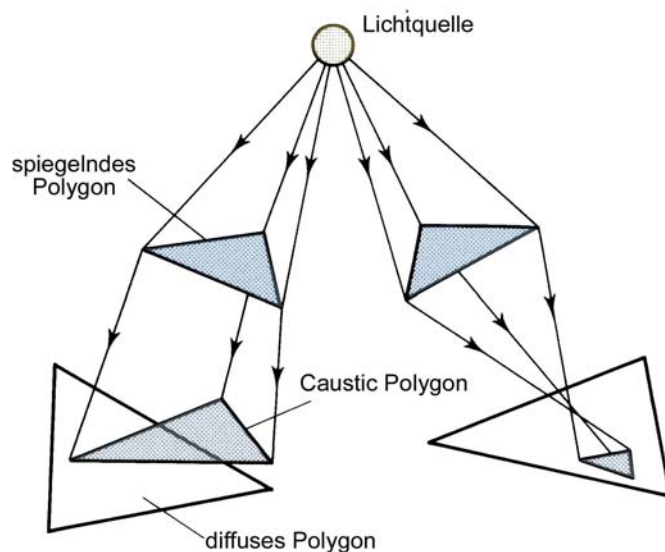
Wenn ein Strahl auf eine Oberfläche trifft, welche mit einer *Illumination Map* verbunden ist, werden die entsprechenden u und v Koordinaten berechnet und ein Teil der Energie des Strahls wird in der *Illumination Map* an diesem Punkt gespeichert. Dies geschieht durch bilineare Teilung des Beitrags verteilt auf die 4 umliegenden Datenpunkte, indem er auf die gegenwärtigen Werte addiert wird. Näher liegende Punkte werden dabei höher gewertet, als weiter entfernte. Die *Illumination Map* enthält zunächst nur Energie, welche in die Intensität umgerechnet werden muss. Dies geschieht durch Dividieren der Energie $E(u, v)$ an einem Datenpunkt durch die korrespondierenden Bereiche. Bis hierhin stellt dieses Verfahren eine betrachterunabhängige Berechnung dar.

Der Aufwand bei diesem Verfahren ist extrem hoch, da die Auflösung der *Illumination Map* hoch und die Anzahl der Strahlen im ersten Path Tracing sehr groß sein muss. In der *Illumination Map* liegt die Energie an dieser Stelle vor, nicht aber die Richtung aus der das Licht einfiel.

Backward Raytracing ist ein Verfahren, das mit Caustics umgehen kann, jedoch ist es vor allem für punktförmige Lichtquellen geeignet, mit großflächigen Beleuchtungsquellen kann es nur schwer umgehen, sie werden lediglich approximiert.

Beam Tracing

Heckbert und Hanrahan überlegten sich, dass es sinnvoll sein könnte statt dünnen einzelnen Strahlen ganze Bündel von Strahlen durch eine Szene zu schicken. Mit diesem Verfahren ist es nun erstmals möglich Schatten wirklichkeitsgetreuer darzustellen. Objektoberflächen werden in diesem



Verfahren in spiegelnde und diffuse unterteilt. Für jedes Polygon eines Objekts mit spiegelnder Oberfläche wird ein Lichtbündel erzeugt, indem von jeder Ecke aus ein Strahl in Richtung der Lichtquelle geschickt wird. Als nächstes wird ein Strahlenbündel erzeugt, welches aus Reflektion oder Brechung

entsteht. Dieses "gleitet" durch die Szene und testet, ob ein diffuses Polygon getroffen wird. Tritt eine solche Kreuzung auf, wird das transmittierte Lichtstrahlbündel auf das diffuse Polygon projiziert. In der Abbildung ist zu sehen, dass das eine Bündel nach der Brechung größer streut (links), als beim rechten Bündel, welches die Lichtstrahlen auf das diffuse Polygon fokussiert. Diese Projektion versieht das Caustic Polygon mit einer Intensität und verweist auf das diffuse Polygon, welches für die Erzeugung dieser Werte verantwortlich ist. Dieses Caustic Polygon wird den Daten der Polygone als ein Oberflächendetail zugefügt. Dies wirkt sich nicht auf die Gestalt eines Objekts aus, sondern auf seine Schattenwirkung. Diese Vorgehensweise hat den Vorteil der Blickunabhängigkeit, solche Caustic Polygone müssen nur einmal berechnet und in den Daten gespeichert werden, solange die Lichtquellen nicht verändert werden.

In einer zweiten Phase wird die Szene vom Blickpunkt aus gerendert, mit nur einem Unterschied; Wird ein diffuses Polygon als sichtbar erkannt, wird getestet, ob es von einem Caustic Polygon beeinflusst wird. Sind mehrere Caustic Polygone an der entsprechenden Stelle vorhanden, werden ihre Werte addiert und mit den Werten des diffusen Polygons ausgewertet.

Es bleibt die Frage offen, welche Intensität den Caustic Polygonen zugewiesen werden. Die Energie E , die das spiegelnde Polygon mit der Leuchtkraft I von der Lichtquelle aus erreicht errechnet sich aus der Leuchtkraft des Polygons mal der Fläche des Polygons, welche von der Lichtquelle beleuchtet wird.

$$E = I(N \cdot L)$$

Wobei N die Oberflächennormale des spiegelnden Polygons ist.

Angenommen das spiegelnde Polygon absorbiert keine Energie, dann wird das Caustic Polygon die Energie E besitzen. Sind die Polygone klein genug gewählt so dass der Abstand zwischen Punkten auf dem spiegelnden Polygon und dem Caustic Polygon nebensächlich ist, dann wird sie Energie gleichmäßig auf das Caustic Polygon verteilt, dass heißt:

$$I_{caustic} = \frac{E}{Fläche_{caustic}}$$

$$= I(N \cdot L) \cdot \frac{Fläche_{spiegelnd}}{Fläche_{caustic}}$$

Diese Gleichung ähnelt dem Formfaktor, auf den später unter dem Punkt Radiosity näher eingegangen wird.

Raytracing kann Spiegelung und Transparenzen von Objekten berechnen, berücksichtigt dabei jedoch nur direktes Licht, reflektiertes wird nicht oder nur annähernd wiedergegeben. Lichtquellen werden als Punktquellen modelliert, Beleuchtung durch Bildschirme, Neonröhren etc. wird nicht erfasst. Raytracing ist außerdem nicht in der Lage natürliche Schatten zu erzeugen, meist wirken sie zu hart.

Distributed Raytracing

Grundlagen

Im Gegensatz zu den 'klassischen' Raytracing-Verfahren in denen die Berechnung eines Bildes mit Hilfe einzelner Strahlen bzw. geometrischer Figuren (Beam- und Conetracing), die durch die Szenerie verfolgt werden, erfolgt, wird bei dem verteilten Raytracing ein Bündel von Strahlen benutzt, deren Anzahl pro Bildpunkt jedoch durch statistische Verfahren eingeschränkt wird. Der Name "*verteilt*" Raytracing bezieht sich auf die Verteilung dieser Strahlen und hat nichts mit verteiltem Rechnen zu tun. Diese Methode heißt - da ein stochastisches Anti-Aliasing ein Teil der Verfahrens ist - auch stochastisches Raytracing und eignet sich besonders zur Darstellung von

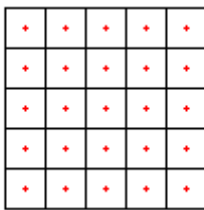
- unscharfen Reflektionen und Brechungen
- weichen Schatten
- Tiefenschärfe
- Bewegungsunschärfe.

Beim Raytracing mittels geometrischer Figuren (Kegel and Pyramiden) statt Strahlen kommt es sehr schnell zu Aliasing-Problemen: Um eine annähernd korrekte Darstellung einer dreidimensionalen Szenerie zu erreichen, müssen die Kegel und Pyramiden sehr schmal sein, was dazu führt, das sich ihre Anzahl erhöht, was letztendlich zu einem immensen Berechnungsaufwand führt. Andernfalls führt die die Regelmäßigkeit des Strahlenbündels zu Aliasing-Artefakten.

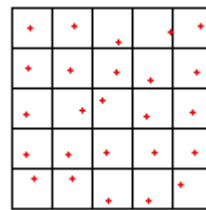
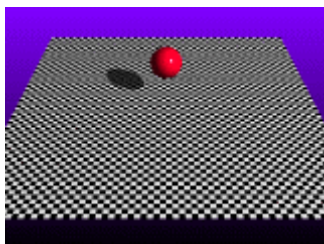
Jitter

Die hier vorgestellte und dem Buch "*Advanced Animation and Rendering Techniques*" entnommene Methode von Cook (1984) verwendet für das Distributed Raytracing 16 Strahlen pro Pixel, die in einem 4x4 Raster zufällig angeordnet sind und deren Position innerhalb dieses Rasters wiederum zufällig verteilt werden. Der Mittelwert, der aus allen Strahlen errechnet wird, ergibt die darzustellenden Werte für den betreffenden Pixel. Damit können alle oben genannten Effekte dargestellt werden, der Rechenaufwand beträgt jedoch "lediglich" den eines 4x4 Supersamplings. Da das aber immer noch sehr viel ist, wird dieses Verfahren in einigen Quellen auch als "Brute-Force-Methode" bezeichnen. Die Zahlen der verwendeten Strahlen je Pixel variiert in verschiedenen Quellen von 9 bis 25, bleibt aber während eines Raytracing-Vorgangs gleich.

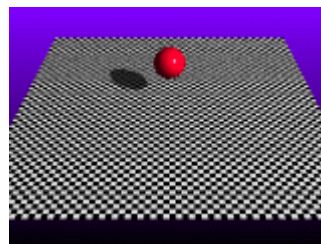
Beispiel für Strahlenverteilung mit 25 Strahlen per Pixel



Regelmäßiges Gitter



Jitter-Gitter



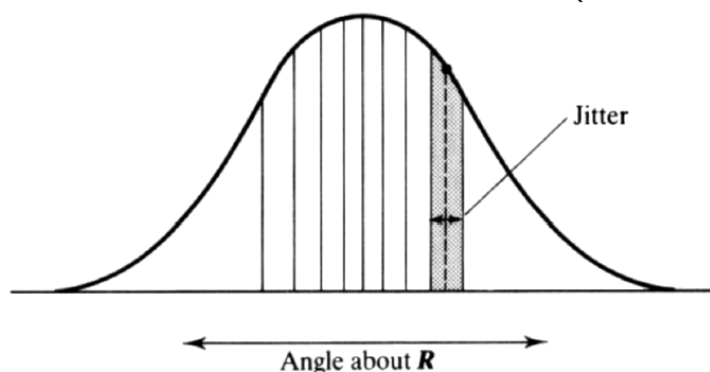
Reflektionsverhalten von Oberflächen

Das Reflektionsverhalten eines Objekts lässt sich mittels der "Bidirectional Reflectance Distribution Function" (BRDF) beschreiben, welche die Anteile der Leuchtkraft die auf einen Punkt x trifft berechnet, der wieder - in einem bestimmten Winkel - reflektiert wird. Ist das Reflektionsverhalten (BRDF) eines Objektes bekannt, können die Reflektionsstrahlen danach verteilt werden.

(Analog dazu lassen sich auch die transmissiven Eigenschaften einer Oberfläche beschreiben.

Mehr dazu hier: <http://geography.bu.edu/brdf/brdfexpl.html>)

Im Gegensatz zum Stochastischen Anti-Aliasing werden nicht nur die Primärstrahlen (Pixel bis zum ersten Auftreffen) berechnet, sondern auch jede darauffolgende Reflektion und Brechung. Für die Berechnung der Re-

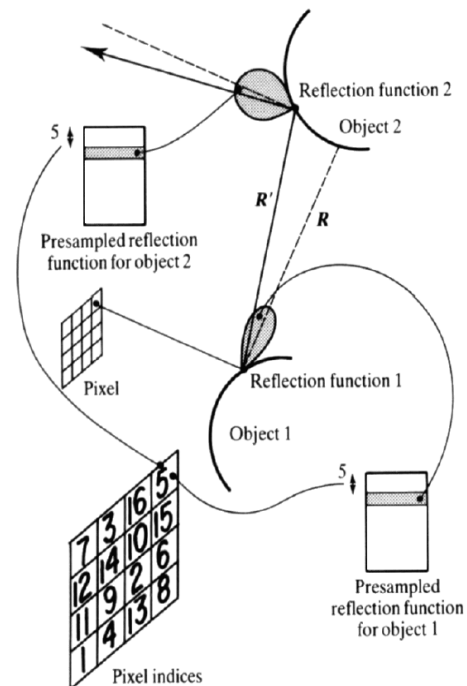


flexionen werden im Voraus berechnete Reflektionsfunktionen verwendet, auf die so genanntes "Importance Sampling" angewendet wird. Dabei werden die

Sampling Intervalle flächenmäßig gleich über die Kurve der Reflektionsfunktion verteilt, was dazu führt, dass bei den größten Werten der Funktion mehr Samples liegen als bei den kleineren Werten.

Ausgehend vom Winkel des einfallenden Strahls und der Oberflächen-Normalen werden in einer vorher berechneten Tabelle Reflektionswinkel und Jittergröße ausgelesen. Jede Region in dem Graphen entspricht einem einzelnen Reflektionswinkel und jeder einzelne Sample-Punkt wird in der Mitte positioniert und dann entsprechend dem zufälligen Jitter verschoben.

Der Index der Nachschlag-Tabelle entspricht dem Index, den der Strahl bei der Pixel-Unterteilung zufällig zugewiesen bekam. Dabei haben der primäre Strahl und alle seine Nachfolger denselben Index, das bedeutet, dass die relative Verschiebung des errechneten Reflektionswinkels R' zum perfekten, tatsächlichen Reflektionswinkel R immer gleich bleibt. Außerdem ist so sichergestellt, dass zwei Strahlen mit statistischer Sicherheit für einen Einfallswinkel nie denselben Reflektionswinkel haben, da die Jitterwerte für die beiden Strahlen unterschiedlich sein sollten.



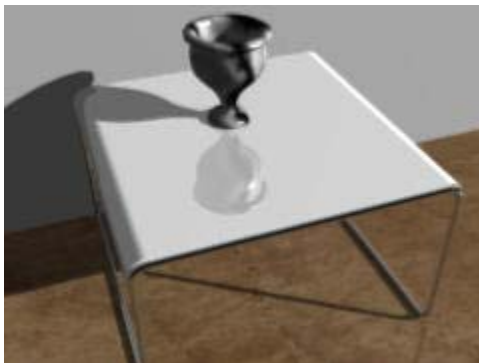
Jeder Strahl wird an jedem Reflektionspunkt nach dem vorberechneten Wert aus der Tabelle weiter durch die Szenerie geschickt. Erst nachdem der Strahl auf eine nichtreflektierende/durchlässige Oberfläche getroffen ist, wird der tatsächliche Wert des Strahls ausgerechnet und zur Mittelwertbildung für den Bildpunkt benutzt.

Transmission/Brechung wird analog zur Reflektion implementiert, nur dass dabei die verwendeten Funktionen andere sind.

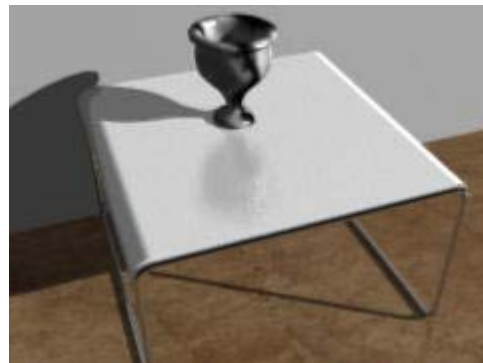
Nachteile

Natürlich hat auch Distributed Raytracing Nachteile:

- Durch die Mittelwertbildung ist kein Pixel wirklich korrekt berechnet. Dieses ist aber eher ein Argument für Perfektionisten. Ähnlich wie bei dem MP3-Audio-Dateien, entsprechen die Bilder nicht dem 'tatsächlichen' Original, die Verluste bzw. Unschärfen sind für das menschliche Auge bei einer genügend hohen Anzahl an Samplingstrahlen nicht wahrnehmbar. Technisch ist der Kritikpunkt berechtigt: Eine "klare" Trennung von Körper und Umwelt kann nur dann erfolgen, wenn alle Strahlen eines Pixels auf den Körper treffen.
- demzufolge können Bilder verrauscht sein
- relativ hoher Rechenaufwand



ideal spiegelnd

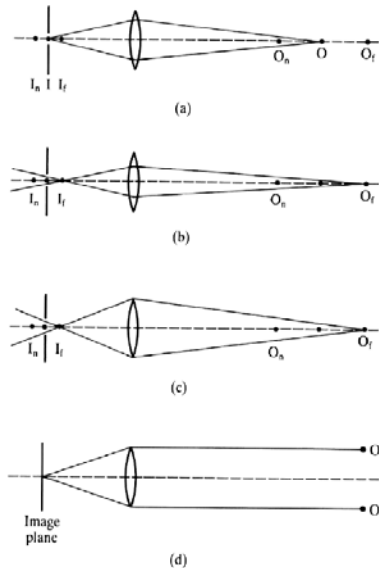


matt spiegelnd

Depth of Field (Tiefenschärfe)

Normales Rendering oder Raytracing wird auf eine Bildfläche projiziert, die der einer Lochkamera gleicht. Deswegen sind alle Objekte – egal in welcher Entfernung sie sich befinden gleich scharf. In den Anfangstagen der 3D-Grafik wurde dieser Effekt als "ultra-realistisch" bezeichnet. Doch schon bald wollte man lieber einen realistischen Realismus und dazu war die Darstellung von Tiefenschärfe nötig – d.h. nur Objekte in einer be-

stimmten Entfernung vom Betrachter sind scharf, alle davor und dahinter sind unscharf.

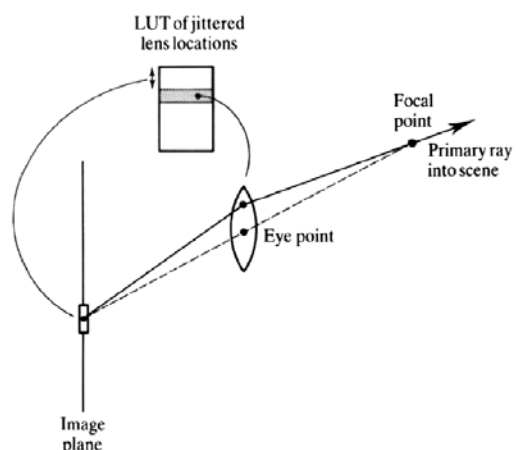


Der erste Ansatz um Tiefenschärfe darzustellen - ein Effekt dem übrigens auch das menschliche Auge unterliegt obwohl wir es nicht wahrnehmen, ist von Potmesil. Nach einem herkömmlichen Render-Vorgang wendete er einen Unschärfe-Filter auf jedes Objekt an. Der Radius des Filters wird durch eine Funktion der Entfernung des Objektes von der Linse bestimmt. Dieses Verfahren erfordert zusätzlichen Rechenaufwand, da es sich um einen zusätzlichen Arbeitsschritt handelt.

Laut Cook ist das Ergebnis jedoch inkorrekt, da Tiefenschärfe nicht nur die Verzerrung der Schärfe sondern auch die Sichtbarkeit eines Objektes beeinflusst. (siehe Abbildung)

Da der Vorgang jedoch nach dem Rendern erfolgt, wird die Sichtbarkeit von Objekten nicht mehr beeinflusst. Rendert man eine Szenerie normal, in der drei gleich große Objekte hintereinander liegen, und der Fokus auf dem mittleren Objekt liegt, ist nur das vorderste Objekt im Bild zu sehen. Bei einer korrekten Darstellung müsste jedoch das hintere Objekt als verschwommener Kreis auf der Bildfläche zu sehen sein.

Die hier vorgestellte Lösung, die vor der eigentlichen Bildfläche die Auswirkungen einer Linse berechnet, ist wiederum von Cook. Ein Raytracing-Verfahren mit mehreren Strahlen pro Pixel ist für solch ein Vorgehen unbedingt notwendig, da es sonst



passieren kann, dass ein Pixel auf den mehrere Strahlen konvergieren falsch dargestellt wird, da in diesem Fall nur ein Objekt abgebildet würde.

Für jeden Pixel wird ein Strahl zum "Augenpunkt" in der Mitte der Linse gezogen. Dann wird der Brennpunkt dieses Strahls errechnet. Durch Nachschlagen in einer gejitterten Linsen-Lokalisations-Tabelle wird ein Punkt (beziehungsweise alle 16 zu einem Pixel gehörende) auf der Linse als Funktion der Position des Strahls innerhalb des Pixels ausgewählt. Alle Primär-Strahlen, die zu dem Augenpunkt-Strahl gehören, werden durch die Linie von dem nachgeschlagenen Punkt zum Brennpunkt gegeben. Dieser Strahl wird bis zum ersten Auftreffen übertragen und erzeugt dort - ganz normal - einen Reflektions- oder Brechungsstrahl. Der Effekt der Tiefenschärfe ist abhängig von der Brennweite und der Blende der Linse. Dabei ist die Brennweite die Entfernung zwischen Linsenmitte und Brennpunkt und Blende die Einheit, die das eintretende Licht misst. Je höher der Blenden-Wert desto weniger Licht fällt in die Linse ein und desto größer ist die Tiefenschärfe.

(Weiche) Schatten

Hier werden – ganz ähnlich dem normalen Raytracing – Schattenstrahlen verwendet, die jedoch wieder verteilt werden.

Wieder wird Importance-Sampling verwendet und die Ziele einer verteilten Lichtquelle werden auf Basis der Helligkeit der verschiedenen Teile der Lichtquelle – ihrer Verteilungsfunktion – und der projizierten Fläche der Lichtquelle wie sie von der Oberfläche die gerade betrachtet wird, zu sehen ist, ausgewählt.

Der hauptsächliche Vorteil von Schatten, die mit Distributed Raytracing erzeugt wurden, ist gegenüber Cone-Traced-Schatten das keine Bänder auftreten, die beim Cone-Tracing als Aliasing-Artefakte auftreten können.

Motion Blur (Bewegungsunschärfe)

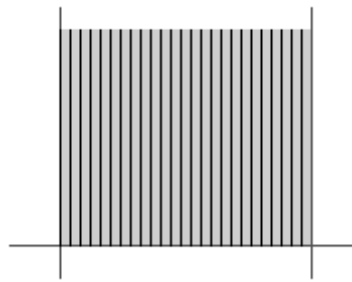
Animierte Sequenzen sind normalerweise – ähnlich wie bei fehlender Tiefenschärfe – immer gestochen scharf. Bei einem Realfilm entspräche das einer Kamera mit einer unendlich kurzen Verschlusszeit. Die menschliche Wahrnehmung liefert jedoch für bewegende Szenerien oder Objekte weniger Detail. Bei Zeichnungen wird dieser Umstand durch so genannte Speedlines realisiert, beim Raytracing durch Einsatz des Motion-Blur-Verfahrens.



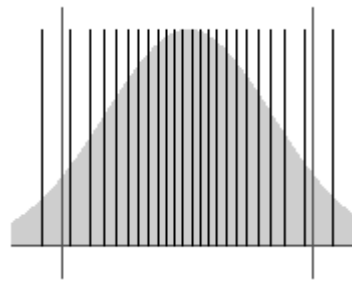
Hier gibt es wieder einen Postprocessingansatz von Potmesil, bei dem jedes einzelne Frame nach dem Raytracing aufgrund der Bewegungsrate der Kamera weichgezeichnet wird. Auch bei diesem Ansatz hat Cook festgestellt, dass die Lösung nicht korrekt ist, da sich während einer Verschlusszeit Sichtbarkeits- und Schattenberechnungen ändern können. (Ein vor stationärem Hintergrund bewegendes Objekt kann mehr vom Hintergrund freigeben. Ein drehendes Objekt, das man von oben sieht, behält scharfes Highlight, Schatten und Umriss.)

Um Motion Blur zu erzeugen, wird zusätzlich zu den normalen Pixel-Samples auch die Zeit gesampelt. Dabei wird die Dauer einer Verschlusszeit (Fernsehkamera: 1/25 Sekunde) in Abschnitte unterteilt (Im Blizzard-Raytracer 128 – 512). Dann wird Importance-Sampling auf eine Funktion angewendet, die das Verschlusszeitverhalten der Linse beschreibt. Jeder Strahl wird dann einem Zeit-Sample zugeordnet.

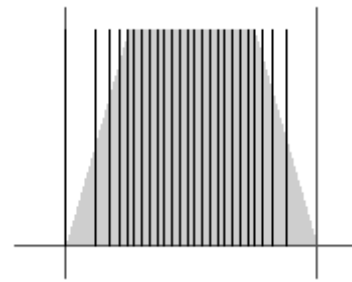
Beispiele für verschieden Verschlussverhalten



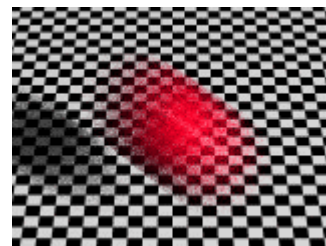
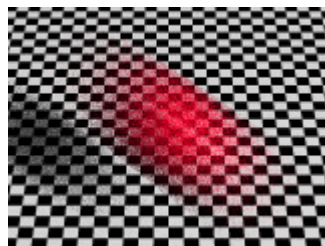
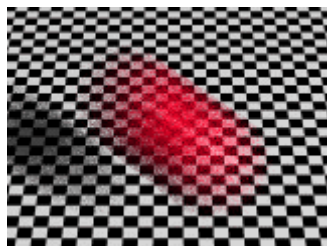
Box (ideal)



Gauss (Filmkamera)



Shutter (Spiegelreflex)



Globale Beleuchtung

Als *lokale Beleuchtung* bezeichnet man das Licht, das direkt von den Lichtquellen auf den schattierten Punkt fällt. Bisher wurde die globale Beleuchtung durch einen Term für ambiente Beleuchtung modelliert, der für alle Punkte auf allen Objekten konstant war. Dieser Term berücksichtigte weder die Positionen von Objekt und Betrachter noch Objekte, die das Umgebungslicht blockieren könnten.

Ein globales Beleuchtungsmodell berücksichtigt bei der Berechnung der Farbe für einen Punkt das von den Lichtquellen direkt abgegebene Licht und das Licht, das den Punkt nach Reflexion und Transmission durch seine eigene und andere Flächen erreicht. Dieses indirekt reflektierte und hindurchgelassene Licht heißt *globale Beleuchtung*.

Radiosity

Raytracing kann spiegelnde Reflexionen und Transparenz mit Brechung (jedoch ohne Streuung) modellieren, alle anderen globalen Lichter lassen sich jedoch nur durch einen ambienten Beleuchtungsterm approximieren.

Radiosity beruht auf der Theorie der Wärmeausbreitung, es geht von der Erhaltung der Lichtenergie in einer geschlossenen Szene aus.

Radiosity ist also ein Verfahren, dass beim Rendern einer Szene nicht nur die Szenen-Lichtquellen berücksichtigt, sondern auch das Licht, welches wiederum von Gegenständen der Szene reflektiert wird. Eine indirekte Beleuchtung, wie eine angestrahlte Wand, welche diffuses Licht in den Raum wirft und so Gegenstände indirekt beleuchtet, ist hiermit möglich.

Die Gesamthelligkeit einer Szene wird berechnet, indem die Helligkeit der Lichtquelle und die Helligkeit des reflektierten Lichts addiert werden. Daher ist die Gesamthelligkeit einer Szene, welche mit Radiosity-Verfahren

berechnet wurde immer heller, als mit herkömmlichen Raytracing-Verfahren.



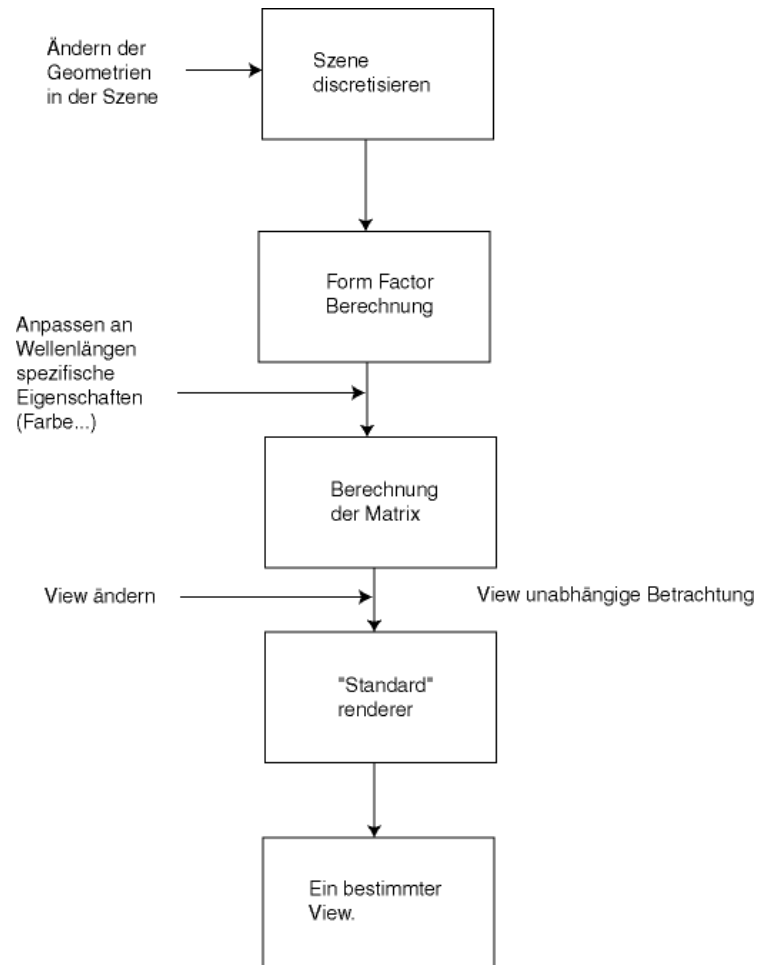
eine Szene konventionell beleuchtet ...



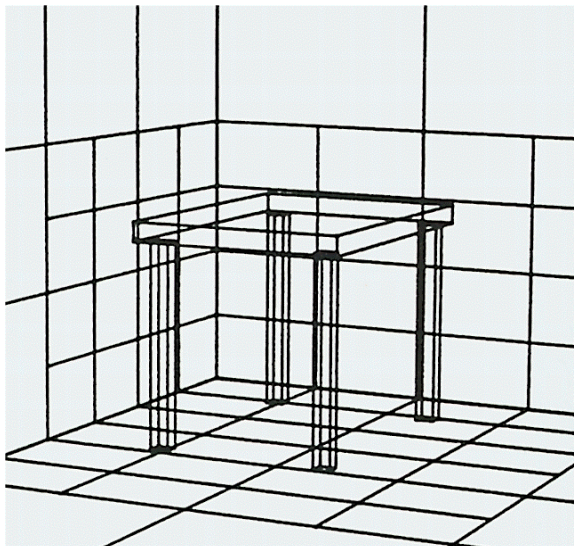
und mit Radiosity

Im Gegensatz zu herkömmlichen Verfahren berechnet Radiosity erst unabhängig vom Blickpunkt alle Lichtinteraktionen einer Szene.

Allgemeine Vorgehensweise



Die Szene wird in eine endliche Anzahl diskreter Flächenelemente zerlegt (Patches).



Es wird angenommen, dass jedes dieser Elemente über die gesamte Fläche gleichmässig Licht abstrahlt und Licht reflektiert. Jedes dieser Flächenelemente wird als opaken, diffusen Lambertschen Strahler und Reflektierer betrachtet.

Seien F_i und F_j zwei Flächenelemente, so kann der Tausch von

Licht innerhalb dieser Flächenelemente der Szene durch ein Gleichungssystem ausgedrückt werden, indem eine geometrische Modellierung durch Formfaktoren vorgenommen wird. Die Lichtenergie zwischen diesen Elementen stellt ein Wechselspiel aus der geometrischen Beziehung der beiden, sowie der Entfernung und der relativen Orientierung dar. So würde viel Energie übertragen werden, lägen die Patches nah beieinander und parallel zueinander. Wichtig dabei ist, dass Verdeckungen beachtet werden müssen.

Die Radiosity – Gleichung

Der Ansatz des Radiosityverfahrens bezieht sich wie oben erwähnt auf ein Berechnungsverfahren für den Wärmeaustausch zwischen Oberflächen. In Anlehnung an dieses Verfahren wurde eine Gleichung für den Strahlungsaustausch entwickelt.

Die Strahlung der oben genannten Flächenelemente ist über diesen konstant und ist die Strahlung, die ein Flächenstück von seiner Umgebung erhält plus die Energie, die es selbst erzeugt.

$$\text{Strahlung} = \text{reflektierteStrahlung} + \text{ausgesendeteStrahlung}$$

Wobei das reflektierte Licht sich aus der Summe des einfallenden Lichts multipliziert mit dem Reflexionsvermögen dieser Fläche errechnet.

$$\text{ReflektiertesLicht} = \text{einfallendesLicht} \cdot \text{Reflexionsvermögen}$$

Das einfallende Licht wird aus der Summe des Lichts, dass alle anderen Flächen der Szene verlässt multipliziert mit dem Lichtanteil, der eine Flächeneinheit des empfangenden Patches errechnet:

*einfallendes Licht = Summe des Lichts, dass alle Patches einer Szene verlässt *
Lichtanteil, der die Flächeneinheit dieses Patches erreicht*

Für zwei Flächenstücke ergibt sich daher folgende Gleichung:

$$B_i A_i = E_i A_i + \rho_i \sum_{j=1}^n B_j A_j F_{ji}$$

Wobei

$B_i A_i$ die gesamte Energie ist, die ein Element verlässt,

E_i ist die Rate mit der das Flächenelement Licht abstrahlt,

ρ_i ist das Reflexionsvermögen des Elements i

F_{ij} ist der dimensionslose Formfaktor, wie oben erklärt beschreibt er den Anteil der Energie, die das Element j insgesamt verlässt und an am Element i ankommt. Er berücksichtigt hierbei die Form und Orientierung beider Flächenelemente sowie eventuell vorhandene blockierende Elemente.

A_i und A_j sind die Flächeninhalte von den Patches i und j .

Eine solche Beziehung existiert für jeden Patch n , daher kann man den Austausch von Licht innerhalb der Flächenelemente der Szene durch ein Gleichungssystem ausdrücken:

$$\begin{bmatrix} 1 - \rho_1 F_{11} & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1 - \rho_2 F_{22} & \cdots & -\rho_2 F_{2n} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ -\rho_n F_{n1} & -\rho_n F_{n2} & \cdots & 1 - \rho_n F_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \cdot \\ \cdot \\ \cdot \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \cdot \\ \cdot \\ \cdot \\ E_n \end{bmatrix}$$

Diese Gleichung muss für jeden Wellenlängenbereich gelöst werden, der im Beleuchtungsmodell betrachtet wird, da p_i und E_i von der Wellenlänge abhängen.

Für ein konvexes Objekt wird F_{ii} immer gleich null sein, da es keinen Beitrag zu seiner eigenen reflektierten Energie tragen wird.

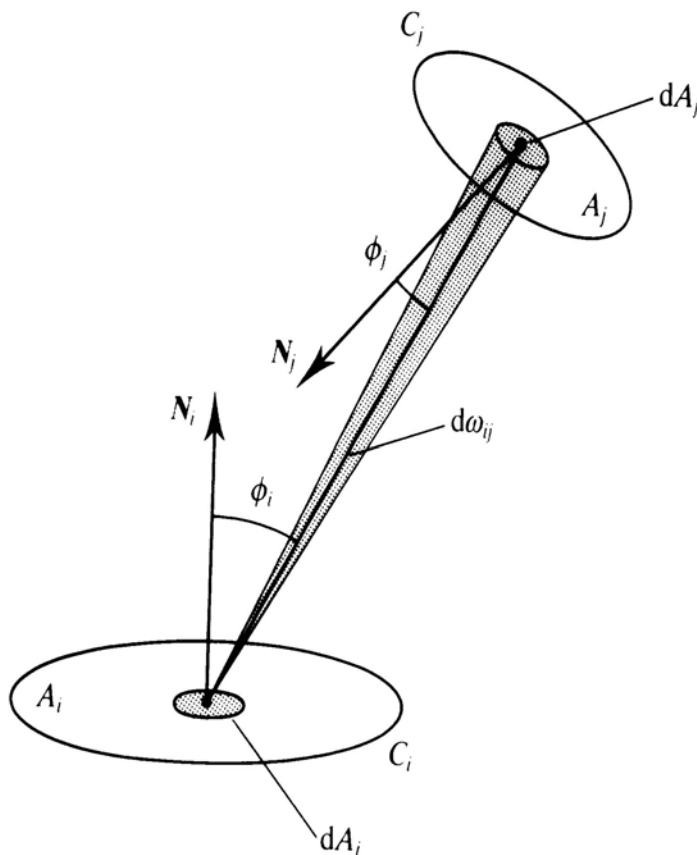
Löst man diese Gleichung nach dem Gauss-Seidel Verfahren, erhält man für jedes Flächenelement einen Strahlungswert pro Wellenlänge, dies ist eine Blickunabhängige Lösung. In einem Standard Renderer kann B_i zur Ermittlung der Intensitäten sichtbarer Flächen benutzt werden.

(Definition von) **Formfaktoren**

Eine entscheidende Bedeutung bei der Berechnung wieviel abgestrahlte Energie des einen Oberflächenpatches ein anderes erreicht, haben die Formfaktoren. Mit den Formfaktoren wird der Strahlungsaustausch zwischen zwei Oberflächen ausgedrückt. Die Berechnung stellt den Kompliziertesten und rechenintensivsten Teil des Radiosityverfahrens dar. Der Wert des Formfaktors hängt im Wesentlichen von der relativen Orientierung und der Entfernung zwischen zwei Patches ab. Der Formfaktor ist definiert durch die abgestrahlte Energie, die das Oberflächenstück verlässt und auf das andere Stück auftritt, geteilt durch die abgestrahlte Energie, welche das Oberflächenstück in alle Richtungen verlässt, also

F_{ij} = Anteil des Lichts, dass von A_i ausgehend A_j trifft

Mit Hilfe des Raumwinkels wird dieser Anteil bestimmt.



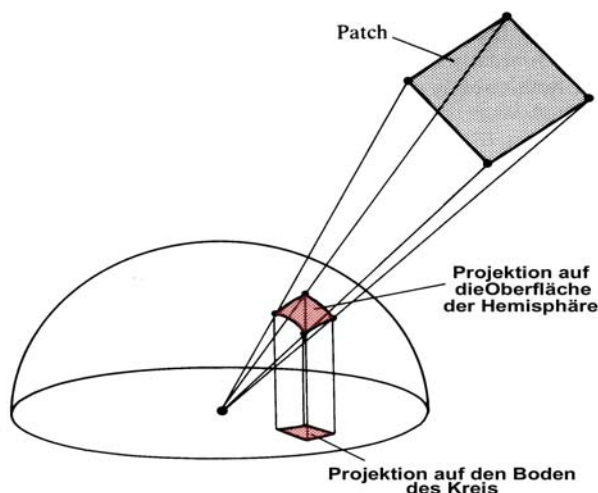
Der Formfaktor wird durch ein hässliches und schwer zu lösendes Doppelintegral beschrieben:

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos \phi_j \cos \phi_i}{r^2} dA_j dA_i$$

Selbst bei einfachen geometrischen Verhältnissen ist die Berechnung dieser integrale sehr aufwendig, außerdem setzen sie die Sichtbarkeit der Flächen voraus.

Die Hemicube-Methode

wegen oben genannter Schwierigkeiten greift man auf eine Näherung zurück, welche das Integral der Formfaktorgleichung auf geometrische Weise löst.



Es wird angenommen, dass bei einem großen Abstand zwischen den Oberflächenstücken, der Formfaktor des betrachteten Oberflächenstücks über seine gesamte Fläche konstant ist. Für die weitere Berechnung wird die

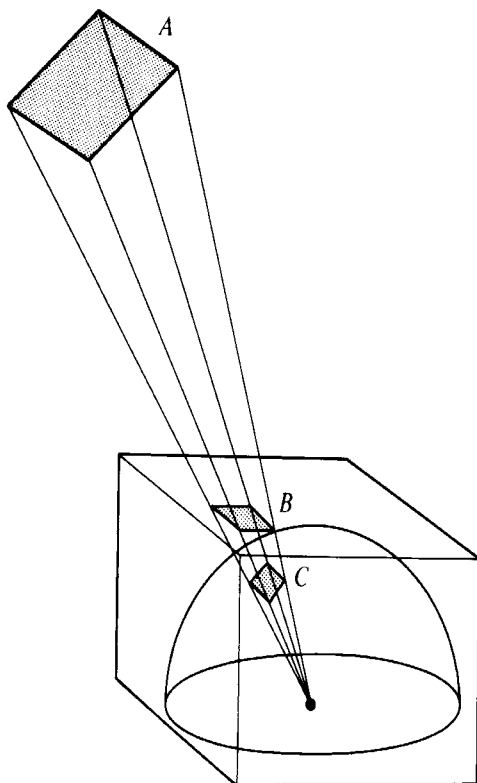
Tatsache ausgenutzt, dass es keinen Unterschied macht, ob man das andere Oberflächenstück oder lediglich seine Projektion auf eine Halbkugel, welche man über den Mittelpunkt des betrachteten Oberflächenstücks gespannt hat, betrachtet.

Das Element wird auf die Halbkugel projiziert und dann orthogonal auf die Grundfläche der Halbkugel; Der Anteil dieser Projektion an der Gesamtgrundfläche der Halbkugel entspricht dem Formfaktor.

Da nun flache Projektionsebenen für den Computer einfacher zu be-

rechnen sind, wird die Halbkugel durch einen Halbwürfel angenähert.

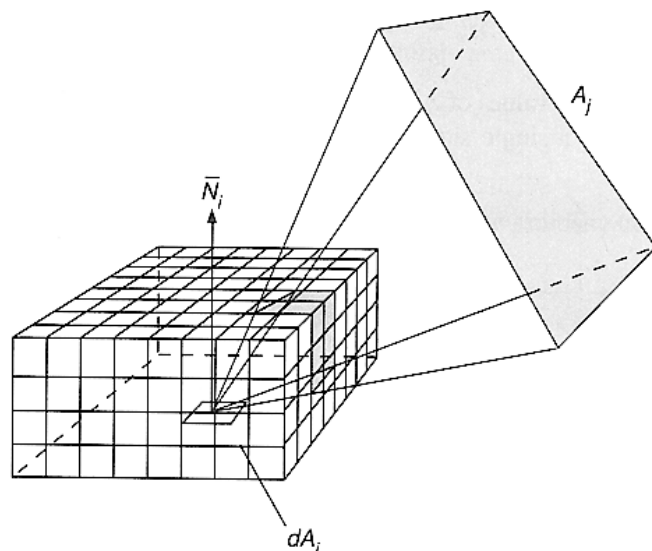
Die Flächenstücke A, B und C in der Abbildung links haben denselben Formfaktor.



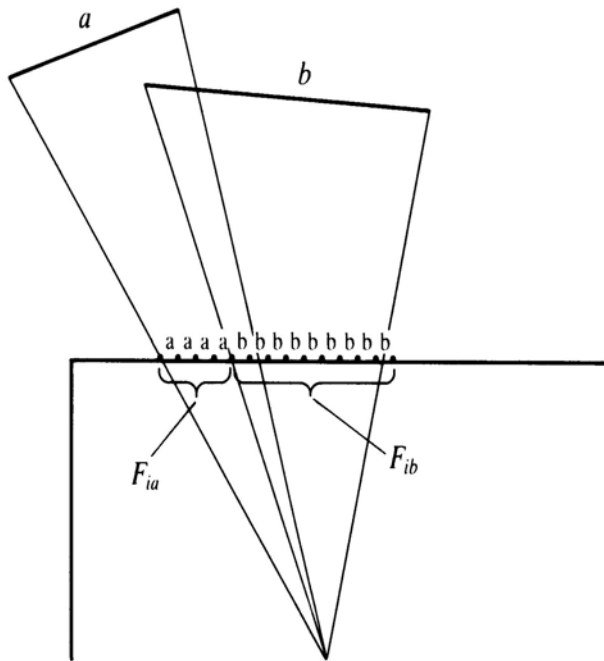
Die Flächen des Halbwürfels werden nun noch zusätzlich in Pixel aufgeteilt (s. Abbildung unten). Die Pixel des Halbwürfels können nun wiederum als kleine Flächen betrachtet werden, somit ist es möglich auch für diese Pixel einen Formfaktor, den so genannten Deltaformfaktoren zu berechnen.

Die Effektivität dieser Methode besteht darin, dass man diese Deltaformfaktoren im Voraus berechnen kann und in einer Look-Up Tabelle ablegt.

Anschließend braucht man nur noch die Projektionen



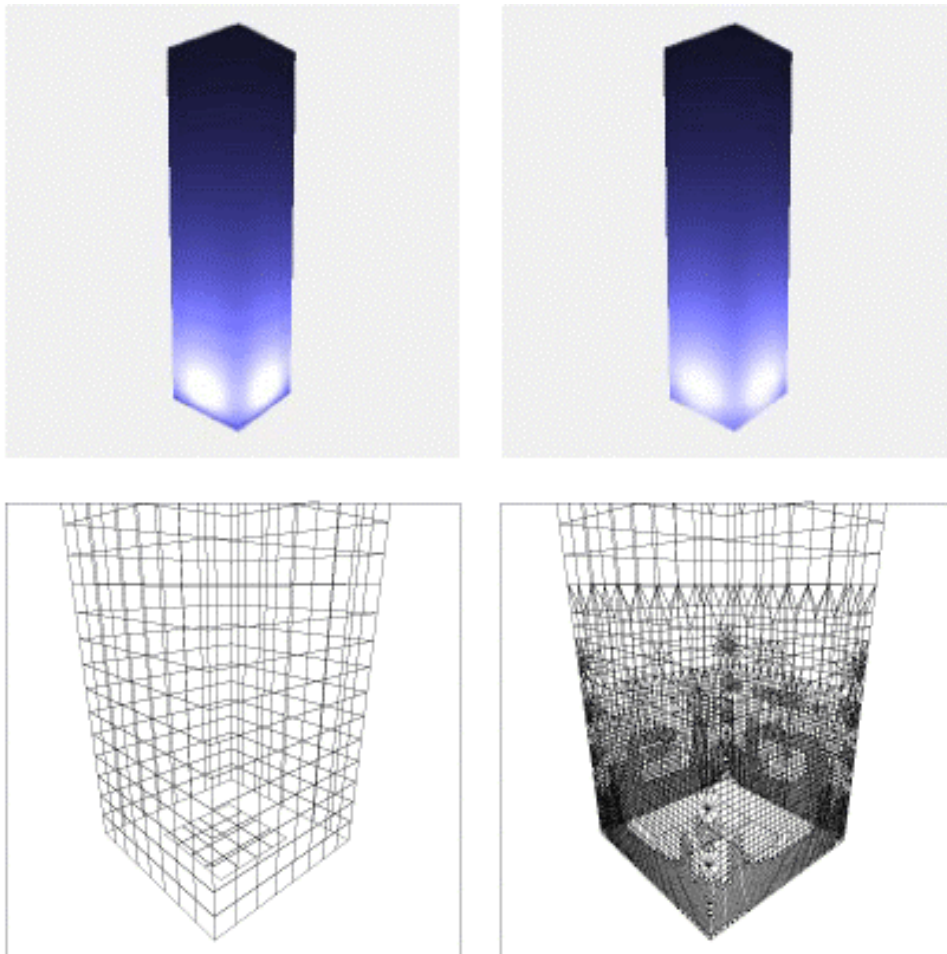
der anderen Patches auf diese Pixel bestimmen und deren Deltaformfaktoren aufsummieren, die durch die Projektion überdeckt werden.



Mit dieser Methode ist es nun auch möglich verdeckte Oberflächenstücke zu berücksichtigen. Man merkt sich hierfür, welches Pixel durch welche Projektion überdeckt wird. Wird bei einer Projektion ein Pixel von einer näher liegenden Projektion überdeckt, so wird sein Deltaformfaktor bei der Berechnung nicht berücksichtigt

Steigerung der Genauigkeit der Lösung durch Substructuring

Die Qualität der Darstellung der bisher betrachteten Lösung hängt stark von der Größe der einzelnen Patches ab. Deshalb werden Bereiche mit hohem Radiosity Gradienten unterteilt, bis eine anzugebende Untergrenze der Kantenlängen eines Patches erreicht ist.



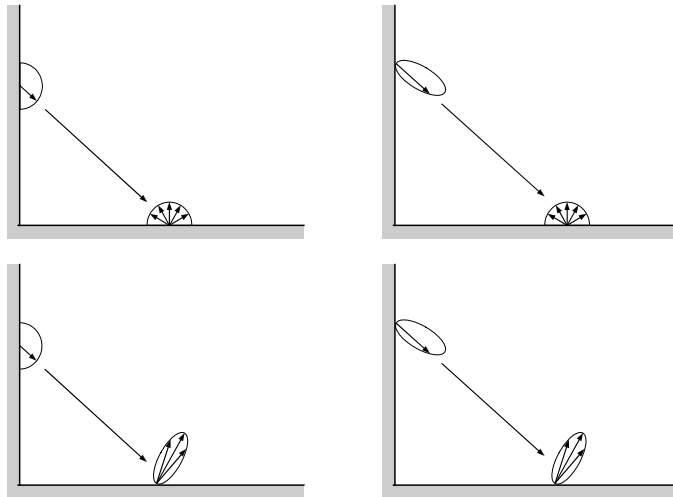
Die Idee dabei ist, eine genauere Lösung durch Unterteilung der Flächenstücke in Elemente vorzunehmen. Bereiche, die eine Unterteilung benötigen sind erst nach der Berechnung einer "grobe" Lösung bekannt.

Zusammenfassung und Fazit

Das Radiosity-Verfahren bietet als erstes die Möglichkeit, indirekte Beleuchtung und die Wirkung von flächenhaften diffusen Lichtquellen effizient zu berechnen. Insbesondere in Innenräumen wird dadurch eine Szene wesentlich realistischer darstellbar als mit anderen Verfahren. Es liefert weiche Helligkeitsübergänge und Schatten. Es ist ein beobachtungsunabhängiges Verfahren, was ein Durchwandern einer Szene in Echtzeit möglich macht. Sowohl vom Speicheraufwand als auch von der Rechenzeit ist es jedoch wesentlich aufwendiger, als Raytracing-Verfahren. Es ist keine Berechnung von Spiegelungen oder Glanzpunkten auf Oberflächen möglich. Die Beschränkung auf rein diffuse Lichtausbreitung kann durch Kombination mit Raytracing aufgehoben werden.

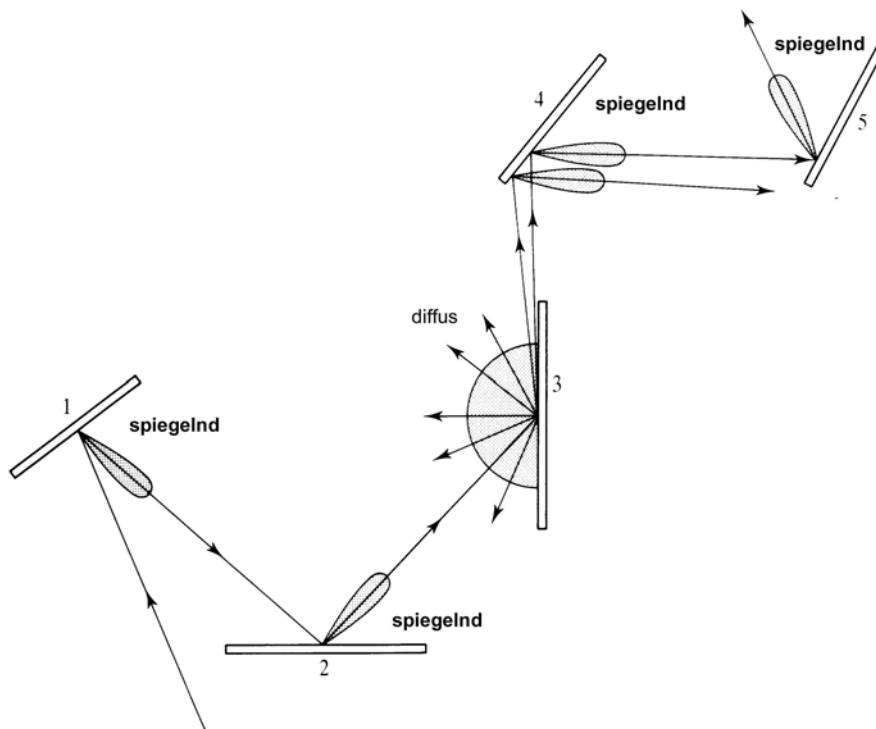
Globale Beleuchtungsmodelle

Es gibt vier verschiedene Arten wie Licht reflektiert werden kann:



- a) diffus zu diffus
- b) spiegelnd zu diffus
- c) diffus zu spiegelnd
- d) spiegelnd zu spiegelnd

Globale Beleuchtung kann nicht durch einen dieser Transportmechanismen simuliert und dargestellt werden, es ist die Kombination dieser 4 Mechanismen entlang einer Kette von Beleuchtungsreaktionen, die Schwierigkeiten mit sich bringt.



Obige Abbildung zeigt, dass der Postprozess des Raytracing nach einer Radiosityberechnung nicht korrekt sein kann, da spiegelnd- spiegelnde

Interaktion von diffuser-diffuser Interaktion, welche näher an der Lichtquelle liegt, auftreten kann. Ebenso wenig kann spiegelnd zu diffuser Interaktion mit der Radiosity Methode nicht dargestellt werden.

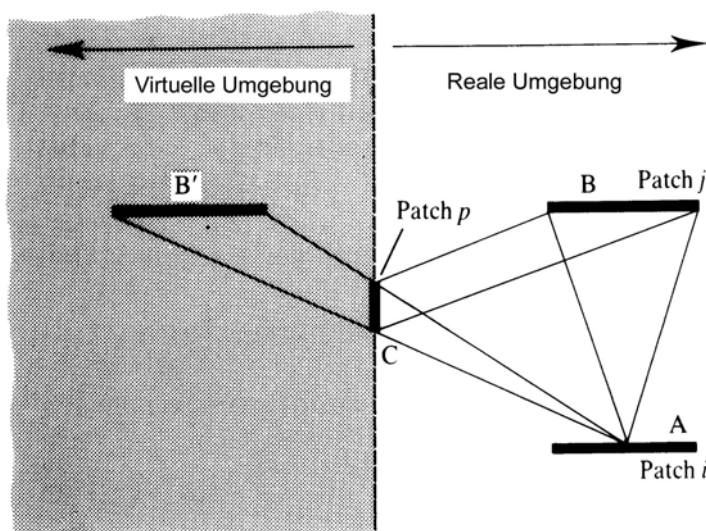
Radiosity kann perfekt mit den Problemen von diffuser zu diffuser Reaktion umgehen, Raytracing beherrscht die Berechnung von spiegelnd zu spiegelnder perfekt, empirisch geht es mit diffuser zu spiegelnder um. Der versuch spiegelnd zu spiegelnd Interaktionen in Radiosity zu integrieren stellt sich als problematisch heraus.

Kombination von Raytracing und Radiosity

John Wallace hat erstmals versucht, das Problem der globalen Beleuchtung in zwei Schritten zu lösen. Der erste Schritt (Prä-Prozess) besteht aus einer erweiterten Radiosity-Lösung, der zweite Schritt (Post-Prozess) aus einem erweiterten Raytracing verfahren.

Hierbei wird Fall a) mit der klassischen Radiosity-Gleichung behandelt, d) mit normalem Raytracing. Für die Fälle b) und c) wurden eigene Methoden entwickelt.

Fall b) – spiegelnd nach diffus- kann man auch als indirekte Beleuchtung



einer Fläche betrachten, bei der diffuses Licht den Umweg über eine spiegelnde Fläche nimmt.

A wird durch B direkt diffus beleuchtet

Wenn C eine spiegelnde Fläche ist, dann muss auch die indirekte diffuse

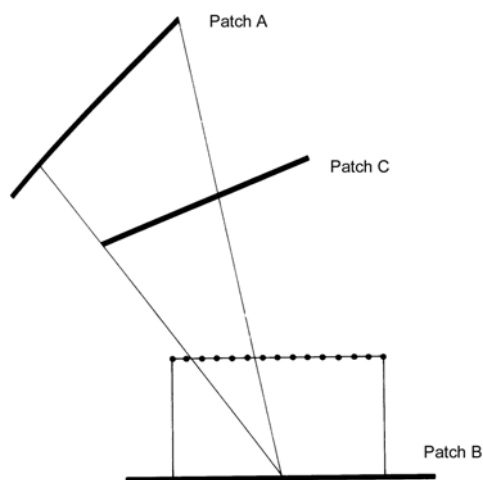
Beleuchtung von B über C nach A berechnet werden. Die Idee besteht nun daraus sich mit einer virtuellen (gespiegelten) Umgebung zu helfen. Das virtuelle Patch B' hat die gleichen Eigenschaften wie B, somit kann

man das Problem der indirekten diffusen Beleuchtung von B über C nach A durch eine direkte Diffuse Beleuchtung von B' nach A umgehen. Dieser Fall kann also im Präprozess mittels Radiosity gelöst werden.

Fall C) diffus nach spiegelnd wird im Postprozess behandelt. Dort wird beim Verfolgen der Lichtstrahlen nicht ein einzelner unendlich dünner Strahl behandelt, sondern ein pyramidenförmiges Bündel. Der Boden der Pyramide liegt dabei auf dem Patch von welchem die diffuse Strahlung ausgeht.

Durch den Präprozess, der Radiosity-Berechnung, ist die Helligkeit der einzelnen Elemente eines Patch bekannt, diese Helligkeiten werden gemittelt, aufsummiert und als einzelner Wert innerhalb des Raytracing verwendet.

Auch Lichtdurchlässigkeit und Transparenz wird unterstützt, sie treten auf, wenn ein Patch A eine anderes Patch B durch ein drittes, transparentes Patch C hindurchsieht.



In diesem Fall benutze Wallace für die Formfaktorberechnung einen umgedrehten Hemicube.

Jedoch steigt durch die Betrachtung einer virtuellen Umgebung die Komplexität der Formfaktorberechnung an, außerdem werden nur planare Spiegelflächen unterstützt, Lichtbrechung wird nicht berechnet.

Wallace et al. Unterteilten bei dieser Two-Pass Methode die Lichtverteilung in vier Typen, welche dann in einem ausschließlich diffusen Präprozess und einen ausschließlich spiegelnden Postprozess behandelt werden. Diese strikte Trennung ist problematisch, denn Licht an sich ist niemals nur spiegelnd oder nur diffus. Frank Sillion und Claude Puech bauen daher die Berücksichtigung von spiegelnden Komponenten des Lichts in die Berechnung der Formfaktoren mit ein.

Die erweiterte Two-Pass Methode von Sillion

Es wurden erweiterte Formfaktoren eingeführt, welche eine beliebige Anzahl von Spiegelungen und Brechungen mit einbezieht. Dadurch wird im Gegensatz zur oben genannten Two-Pass Methode nicht mehr nur der direkte Energietransfer zwischen Patches berechnet, sondern auch der Energietransfer von einem zum anderen Patch über beliebig viele Spiegelungen und Brechungen hinweg. Von jedem Patch aus werden Strahlen in endlich viele Richtungen geschossen, zu jeder Richtung wird ein Strahlenbaum erzeugt, der Formfaktor wird nun in jedem dieser Strahlbäume verteilt.

Das nun noch ausstehende Raytracing gestaltet sich jetzt sehr einfach, da aufgrund der vorausgegangenen Berechnungen keine Schatten mehr berechnet werden müssen und das Shading trivial ist, der Wert eines Punktes kann direkt „nachgelesen“ werden. Das Einbringen sowohl diffuser als auch spiegelnder Lichtverteilungen in die Formfaktorberechnung verhilft diesem Verfahren zu exakteren Resultaten. Die dazu verwendeten Strahlenbäume erlauben Spiegelungen und Brechungen an beliebigen Flächen. Spiegelnd reflektierende Flächen müssen während des Radiosity Verfahrens nicht kostspielig unterteilt werden.

Photon Maps

Ein Großteil der Photonentechniken [vor allem Grundlagen] sind von oder unter Mitarbeit von Henrik Wann Jensen entstanden. Mittlerweile gibt es von ihm auch ein Buch zu Photon Maps. Das einzige in Berlin an öffentlichen Bibliotheken verfügbare Exemplar an der TU Berlin war jedoch ausgeliehen. Die nun folgenden Erklärungen stammen deshalb zum Großteil aus der Diplomarbeit von Stephen Maaß und Folien, die aus dem Netz stammen.

Um eines vorweg zu nehmen: bei diesem Verfahren handelt es sich nicht um ein weiteres Raytracing-Verfahren, sondern um die Unterstützung eines [herkömmlichen] Distributed Raytracers. Grundsätzlich funktioniert diese Verfahren so, dass statt Lichtstrahlen kleine Energiepakete von den Lichtquellen im Raum verteilt werden. Treffen diese auf eine Oberfläche, die sie absorbiert, werden sie in einer Photon Map abgespeichert und später beim Rendern für die Berechnung der Leuchtstärken benutzt.

Das Photon

Aber der Reihe nach: Photonen werden von einer Lichtquelle ausgesandt. Im einfachsten Fall ist das eine Punktquelle, welche die „Lichtpäckchen“ gleichmäßig verteilt in alle Richtungen ausstrahlt. Die Energieladung eines einzelnen Photons berechnet sich dabei anteilmäßig aus der Leistungskraft der Lichtquelle und dem Anteil der Photonen. Ein Photon einer Quelle, die mit 60 Watt 100.000 Energiepakete aussendet, hätte einen Flux von $\Delta\Phi = 0,6 \text{ mW}$. An dieser Stelle kommen dann gleich erste Erweiterungen und Verbesserungen zum tragen.

$$\text{Ladung} / \text{Flux} / \Delta\Phi = \frac{\text{Leistung der Lichtquelle}}{\text{Anzahl der Photonen}}$$

Der Leuchtkörper

Theoretisch ist eine äußerst realistische Darstellung von Leuchtkörpern möglich, da der Abstrahlort und -winkel für jedes einzelne Photon festgelegt werden könnte. In einer Glühbirne wäre in solch einem Fall der Austrittspunkt der Photonen die Glühwendel. Die Teilchen treten dann durch eine beinahe transparente Oberfläche [den Kolben] in die Szenerie ein. Meist werden aber mathematische Funktionen verwendet um das Abstrahlverhalten zu steuern. Die wohl gebräuchlichste Art der Abstrahlung [laut der Diplomarbeit von Stephen Maaß] ist die Lambert-Charakteristik. Dabei richtet sich die Größe der abgestrahlten Energie [das entspricht der Anzahl der Photonen] nach dem Kosinus zwischen der Oberflächennormale des Leuchtkörpers und dem möglichen Abstrahlwinkel zwischen denen die Photonen mittels Importance Sampling verteilt werden.

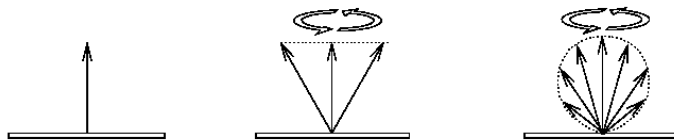


Abbildung 3.2 Querschnitte verschiedener Strahlungscharakteristiken: directional, spot, lambert

Damit verringert sich die Anzahl der Photonen, je mehr der Winkel der Abstrahlrichtung von der Normalen abweicht. Noch einmal: Natürlich kann man an durch Verwendung einer beliebigen Funktion nahezu jede Art von Lampe oder Scheinwerfer darstellen.

Die Projektionskarte

Schnell fällt vor allem in Szenerien mit wenigen Objekten auf, dass viele Photonen unnütz ausgeschiedt werden, da sie niemals auf ein Objekt treffen. Um die Anzahl nutzloser Berechnungen einzudämmen, werden sogenannte Projektionskarten benutzt. Diese beschreiben mit Hilfe eines binä-

ren Rasterbildes die Sichtbarkeit von Objekten (von der Lichtquelle aus). Die Färbung der Pixel wird in einem Vorverarbeitungsschritt mittels Projektion der Objekte [oder einfacher: Objekt(gruppen)hüllen] errechnet. Während der Ausstrahlung treten nur die durch die markierten Zellen des Rasterbildes Photonen aus.

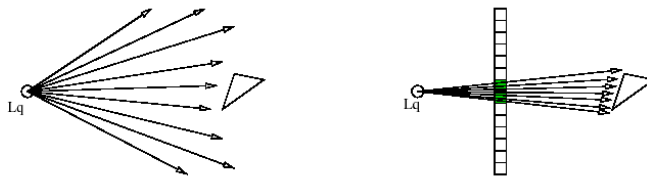


Abbildung 3.3 Aussendung der Photonen: naiv und unter Verwendung einer Projektionskarte

Um zu verhindern das mehr Energie ausgestrahlt wird, als dem Teil der Lichtquelle entspricht, der durch das Raster fällt, wird die die Energie der einzelnen Photonen an diesen Umstand angepasst. [Das ist notwendig, da alle verfügbaren Photonen durch das Raster geschickt werden und nicht nur ein entsprechender Teil.] Der Energiegehalt eines Photons entspricht nach der Anpassung dem ursprünglichen Wert multipliziert mit dem Verhältnis zwischen insgesamt vorhandenen und ausgefüllten Zellen des Rasterbildes.

Russisches Roulette

Die eigentliche Stärke des Photonmap-Verfahrens liegt in der Berechnung von [mehrfachen] Interaktionen des Lichtes mit Oberflächen jeglicher Art. Bei einer tatsächlichen Betrachtung von Lichtverläufen, müsste die Energieladung eines Photons aufgespalten werden. Dies würde schnell zu einem exponentiellen Wachstum der Anzahl von Photonen führen, die verfolgt werden müssen. Um das zu verhindern, wird Russisches Roulette angewendet, eine Methode, welche die Photonenanzahl niedrig hält und trotzdem dafür sorgt, das die annähernd korrekte Energiemenge in alle

erforderlichen Richtungen übertragen wird. Verfügt eine Fläche über alle 3 Eigenschaften [diffus, spiegelnd, absorbierend], sieht das Russische Roulette wie folgt aus:

$\xi \in [0 \dots d]$ → **diffuse Reflexion**

$\xi \in [d \dots d+s]$ → **spiegelnde Reflexion**

$\xi \in [d+s \dots 1]$ → **Absorption**

Wobei ξ ein zufällig ermittelter Wert ist. Dadurch das die jeweiligen Photonen *vollständig* mit Fläche interagieren, nimmt die Photonenzahl mit zunehmender Reflektion nur ab, während der Energiehaushalt durch die korrekt gewählten Wahrscheinlichkeiten richtig wieder gegeben wird.

Durch die Berücksichtigung der Energielevel jeder einzelnen RGB-Farbe während einer Photon-Bahn, kann eine sehr farbgetreue Lichtverteilung errechnet werden. [Extrembeispiel: Ein Photon, das bereits von einer roten Wand reflektiert wurde, wird von einer blauen Wand definitiv absorbiert.]

Die Photonenverfolgung endet, wenn ein Photon die Szene verlässt, absorbiert wird oder die Anzahl der Interaktionen einen vorgegebenen Wert überschreitet. Handelt es sich um ein Photon zur Konstruktion einer Kauistik-Photonmap, so endet der Pfad zusätzlich bei der Interaktion mit einer diffusen Oberfläche.

Treffer!

Trifft ein solches Photon auf eine Oberfläche wird es entweder absorbiert oder reflektiert [oder gebrochen, das macht bei Photon Maps keinen Unterschied]. Ein Treffer wird mit folgenden Daten in einem kd-Baum abgespeichert:

- Position im Raum
- Energiegehalt
- Einschlagswinkel
- Caustic / Regular - Flag
- Daten zur Speicherung im kd-Baum (Dimension)

Die Daten werden nicht komprimiert, da sie im Verlaufe des Verfahrens so oft benötigt werden, dass der zusätzliche Aufwand der Dekodierung den Nutzen durch den Speicherplatzgewinn übersteigt.

Anmerkung zu kd-Bäumen:

„Der kd-Baum ist eine raumunterteilende Datenstruktur. Das Kürzel „kd“ steht hierbei für „k-dimensional“... Der Ausgangsraum wird durch den kleinsten, achsenparallelen Quader (*Bounding-Box*), welcher alle Photonen enthält, definiert. Beginnend mit dieser Boundingbox wird der Raum durch jeden Knoten des kd-Baumes an einer Koordinatenachse geteilt. Pro Knoten wird dabei die Dimension und Position der Teilung gespeichert. Alle Knoten im linken und rechten Unterbaum liegen demnach auf verschiedenen Seiten der unterteilenden (Hyper-) Ebene.“

Die Karte bitte

Während des Photonmap-Vorgangs werden bis zu drei Karten angefertigt:

- eine globale Photonmap, die relativ grob angelegt ist und nur mit ein „paar“ Photonen realisiert wird
- eine eigene Map für Caustics. Am Anfang des Photonmappings oder während des Durchgangs, welcher die globale Photonmap liefert, werden die Flächen oder Objekten, die Caustics erzeugen können, markiert und in gesonderte Projektionskarten eingetragen. Mit deren Hilfe wird dann ein weiterer Durchgang gestartet, bei dem das 25 – 100fache an Photonen verwendet wird. Zur Berechnung der C-Map wird ein Pre-Tracing durchgeführt. Das ist unter anderem notwendig, da so der geringste Abstand zweier Strahlen, die auf einen Körper [nicht zwangsweise denselben] treffen herausgefunden werden kann. Die Dichte der Elemente muss größer sein als dieser Abstand.

- Die Shadow-Map wird erzeugt, wenn ein Photon auf eine nichttransparente Oberfläche trifft. Auf ihr sind Schattenphotonen eingetragen, die ähnlich den normalen Schattenstrahlen funktionieren.

Während die Caustic-Map direkt visualisiert wird, dienen globale und Schatten-Map lediglich der Unterstützung des Distributed Raytracers, der im zweiten Schritt benutzt wird, um die Szenerie tatsächlich zu rendern.

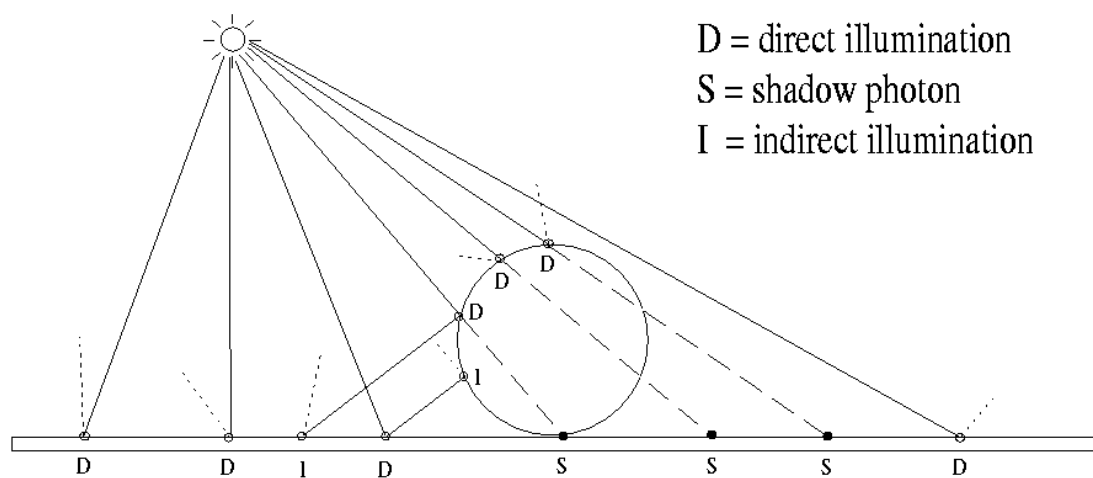
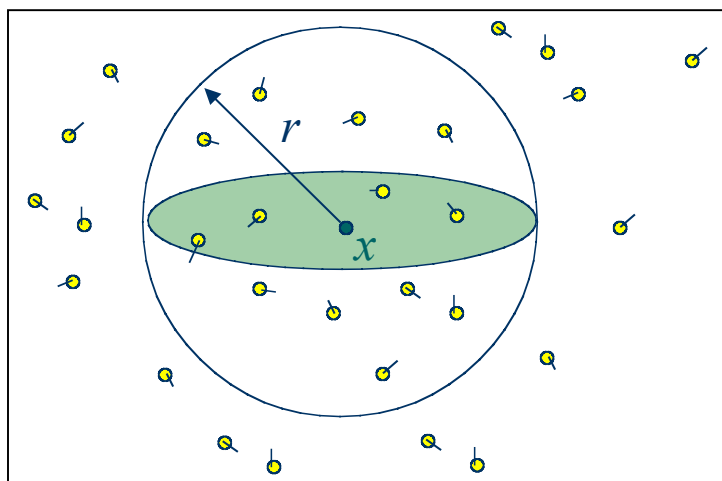


Fig. 1. The photons in the global photon map are classified to optimize the rendering of shadows

Nachdem alle Photonen gespeichert sind, lässt sich für jeden Punkt x im Raum dessen „Leuchtwert“ feststellen. Dazu betrachtet man eine Kugel um diesen Punkt x mit einem Radius r derart, dass diese Kugel die n nächsten Photonen enthält. Dann lässt sich die Leuchtkraft wie folgt be-

rechnen:



$$E_x = \frac{\sum_{i=1}^n e_i}{\pi r^2}$$

Mittels einer kleinen mathematischen Überprüfung wird ausgeschlossen, dass falsche Protonen verwendet werden: Ist das Skalarprodukt der Normale des Punktes und der Normale des Photons negativ, wird das Photon nicht verwendet. Das entspricht $\cos\{\alpha\}$ des Winkels der Vektoren. Dieser Wert wird negativ, wenn der Winkel größer als 90° bzw. kleiner als 270° ist. Diese Rechnung verhindert also, dass Photonen um die Ecke leuchten.

Caustics und Diffus-Second-Hits werden direkt aus der Map abgelesen. Das bedeutet: Trifft der Strahl auf einen Körper, wird der Leuchtwert aus der Map direkt übernommen. Damit dieser Wert korrekt ist, muss die Caustic-Photon-Map sehr dicht sein.

Sind alle Berechnungen für das Photon-Mapping abgeschlossen, wird ein Distributed Raytracer benutzt, um das Bild zu berechnen. Der dargestellte Leuchtwert eines Punktes der Szenerie wird dabei durch folgende Formel berechnet:

$$L_p = L_e + (L_{r,s} + (L_{r,d,l} + L_{r,d,s} + L_{r,d,d}))$$

Die Teil-Leuchtwerte haben dabei folgende Bedeutung:

Art	Erklärung	Berechnung
L_e	ausgestrahltes Licht	Errechnung aus Photon Map
$L_{r,s}$	spiegelnd reflektiertes Licht	Distribution Raytracing
$L_{r,d,l}$	diffus reflektiertes Licht, das von einer Lichtquelle kommt	Distribution Raytracing
$L_{r,d,s}$	diffus reflektiertes Licht, das von einer spiegelnden Reflektion/Refraktion kommt	Kaustiken werden direkt aus der Photon Map abgelesen / sonst wie $L_{r,d,d}$
$L_{r,d,d}$	mehrfache diffuse Reflektion/Refraktion	Berechnung aus Photon Map und BRDF und "Irradiance Gradients"; ab dem zweiten Hit : Photon Map

Der Schatten wird durch ähnliche Abschätzungen berechnet, dabei werden natürlich die Shadow-Photonen verwendet. Das Raytracing wird bis zu einer gewissen Grenze (Threshold) durchgeführt, das heißt, so bald der

Leuchtwert eines Auftreffpunkts eines Strahls einen gewissen Wert unterschreitet, wird der Strahl nicht weiter verfolgt.

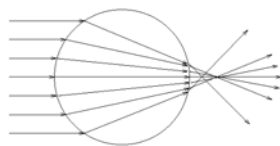
...und das ist erst der Anfang.

Kombiniert man Photon-Maps mit Volumen Rendering/Raycasting, lässt sich die Interaktion zwischen Photonen und Materialien (Cooper nennt sie Medien) darstellen. Dabei wird ein Volumen diskretisiert*. Bei jedem Punkt wird gewürfelt, ob das Photon mit dem Medium interagiert und ob es seine Energie verteilt (scattered), was zum Beispiel dazu führt, dass man Lichtstrahlen im Nebel sehen kann, da ja die Energie nicht nur in Richtung des Lichtstrahls geht, sondern in alle Richtungen verteilt wird.

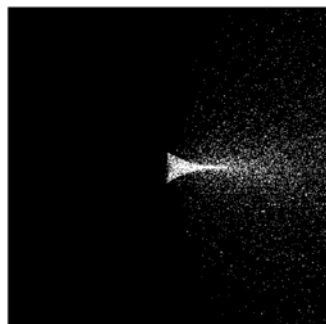
***Ein kurzer Ausflug ins Volume-Raycasting:**

Um eine Szenerie zu diskretisieren, wird eine als Raycasting bekannte Technik benutzt. Ein Raum wird dazu in kleinste Datenpunkte zerlegt, die analog zum Pixel Voxel heißen. Jeder dieser Punkte hat dabei bestimmte Werte. In der Medizin werden diese Daten durch Messwerte gegeben, die der Dichte an dem jeweiligen Punkt bestimmen. Für die Verwendung bei Photon Maps werden diese Punkte durch mathematische Formeln beschrieben und enthalten Werte für Dichte, Radiosity oder andere. Durch eine Transfer-Funktion wird diese Zahl in einen RGBA-Wert umgewandelt, der die Sichtbarkeit in semi-transparenten Volumen (Flüssigkeiten, Gase, Wolken) beschreibt. Beim Raycasting - richtig müsste es hier eigentlich Photoncasting heißen, werden die Photonen durch das Volumen verfolgt und an jedem Voxel die Helligkeit bestimmt. Durch russisches Roulette wird bestimmt, ob ein Photon im Volumen trifft oder durchgeht, so "dämpft" das Volumen mit dem Verlauf die Energie der Photonen. Auch hier wird wieder Scattering verwendet - durch die wiederum zufällige Verteilung der Energie eines Photons in alle Richtungen lassen

sphere in fog



light path



photon map



Literatur

- Watt, Alan Watt, Mark: Advanced Animation and Rendering Techniques. Addison Wesley; 1994
- James Arvo: Backward Raytracing Apollo Computer, Inc Chelmsford, MA, August 1986 (in Developments in Raytracing, Siggraph 86 Course Notes, Volume 12)
- Matthias Röhm: "Radiosity"; Universität Ulm; Proseminar Computergrafik SS2002
- Monika Geppert; "Ein globales Beleuchtungsmodell für diffuse Reflexion", Diplomarbeit ; Technische Universität München
- "Global Illumination using Photon Maps"; Henrik Wann Jensen; 1996:
<http://graphics.stanford.edu/~henrik/papers/>
- "Efficient Importance Sampling Techniques for the Photon Map"; Alexander Keller und Ingo Wald:
<http://www.uni-kl.de/AG-Heinrich/Importance.pdf>
- "Photon Maps in Bidirectional Monte Carlo Ray Tracing of Complex Objects"; Henrik Wann Jensen und Niels Jörgen Christensen; 1994:
<http://graphics.stanford.edu/~henrik/papers/>
- "Effiziente Photonmap-Techniken"; Stefan Maaß; 2001:
<http://graphics.tu-bs.de/PubArc/dipl/>
- "Das Photonmap-Verfahren"; Stefan Maaß; 2001:
<http://graphics.tu-bs.de/lvcg01-02/Vorlesung1/>
- "Photon Mapping"; John C. Hart; 2003:
<http://www-courses.cs.uiuc.edu/~cs319/>
- "Efficient Simulation of Light Transport in Scenes with Participating Media using Photon Maps"; Scott Cooper; 2000:
<http://www.cs.unc.edu/~scooper/comp238/>
- Fachhochschule Wedel :
<http://www.fh-wedel.de/~ko/Galerie/2001-WS-Seminar/>

- Braunschweig University of Technologie; "Radiosity Teil 1", "Radiosity Teil 2", "Radiosity Teil 3", "Radiosity Teil 4"; S. Schäfer:
<http://www.cg.cs.tu-bs.de/lvcg99-00/Vorlesungsmaterial.html>
- Universität des Saarlandes:
"Beleuchtung und Raytracing", "Radiosity"; Phillip Slusallek:
<http://graphics.cs.uni-sb.de/Courses/ws9900/cg/>
- Universität Marburg: "Hybride Methoden"; Adam Sosnowski;
"Das hierarchische Radiosity-Verfahren"; Juri Bender:
<http://www.mathematik.uni-marburg.de/~schroedr/sem/sem.html>
- "BRDF explained"; Wolfgang Lucht; 2000:
<http://geography.bu.edu/brdf/brdfexpl.html>
- "Blizzard – Distributed Raytracing"; Steffen A. Mork; unbekannt:
http://ls7-www.cs.uni-dortmund.de/~mork/blizzard_distr.html
- "Computergrafik"; Vorlesung von Winfried Kurth; 2002:
http://www-gs.informatik.tu-cottbus.de/~wwwgs/cg2_vorles.htm
(insbesondere http://www-gs.informatik.tu-cottbus.de/~wwwgs/cg_v12a.pdf)
- "How do you like your anime – smooth or chunky?"; Chad Kime; unbekannt:
http://www.ex.org/4.8/18-column_pov1.html